

Pre-Proceedings of INEX 2007

**Edited by
Norbert Fuhr
Mounia Lalmas
Andrew Trotman**

December 17-19, 2007
Schloss Dagstuhl
International Conference and Research
Center for Computer Science
<http://inex.is.informatik.uni-duisburg.de/2007/>

TABLE OF CONTENTS

Organizers	vii
Preface	ix
Acknowledgements	x
Schloss Dagstuhl	xi

AD HOC TRACK

Overview of the INEX 2007 Ad Hoc Track	1
N. Fuhr, J. Kamps, M. Lalmas, S. Malik, A. Trotman	
INEX 2007 Evaluation Measures	23
J. Kamps, J. Pehcevski, G. Kazai, M. Lalmas, S. Robertson	
The Role of Shallow Features in XML Retrieval	33
F. Huang	
The Simplest XML Retrieval Baseline That Could Possibly Work	39
P. Dopichaj	
ENSM-SE at INEX 2007: Scoring with Proximity	53
M. Beigbeder	
The Garnata Information Retrieval System at INEX'07	56
L. M. de Campos, J. M. Fernandez-Luna, J. F. Huete, C. Martín-Dancausa, A. E. Romero	
Preliminary Work on XML Retrieval	70
Q. Wang, Q. Li, S. Wang	
Indian Statistical Institute at INEX 2007 Ad Hoc Track: VSM Approach	77
S. Pal and M. Mitra	
Using Topic Models in XML Retrieval	82
F. Huang	
TopX @ INEX 2007	87
A. Broschart, R. Schenkel, M. Theobald, G. Weikum	
LIG at INEX 2007 Ad Hoc Track : Using Collectionlinks as Context	94
D. Verbyst, P. Mulhem	
CSIR at INEX 2007	105
W. Lu, D. Liu, J. Jiang	
Document Order Based Scoring for XML Retrieval	111
P. Arvola	
An XML Information Retrieval using RIP List	117
H. Tanioka	
How well does Best in Context reflect Ad Hoc XML retrieval?	124
J. A. Thom, J. Pehcevski	

Dynamic Element Retrieval in the Wikipedia Collection	126
C. J. Crouch, D. B. Crouch, N. Kamat, V. Malik, A. Mone	
Phrase Detection in the Wikipedia	128
M. Lehtonen, A. Doucet	
Ranking Ad-Hoc Retrieval using Summary Models and Structural Relevance	133
M. S. Ali, M. P. Consens, S. Khatchadourian	
Probabilistic Document Model Integrating XML Structure	139
M. Gery, C. Largeton, F. Thollard	
Semi-Supervised Learning of Ranking Functions for Structured Information Retrieval	150
D. Buffoni, J.-N. Vittaut, P. Gallinari	
Ranking and Presenting Search Results in an RDB-based XML Search Engine	156
K. Hatano, T. Shimizu, J. Miyazaki, Y. Suzuki, H. Kinutani, M. Yoshikawa	
Study on Reranking XML Retrieval Elements Based on Combining Strategy and Topics Categorization	170
J. Liu, H. Lin, B. Han	
BOOK SEARCH	
<hr/>	
BookSearch'07: INEX 2007 Book Search Track Overview	177
G. Kazai, A. Doucet	
Logistic Regression and EVIs for XML Books and the Heterogeneous track	185
R. R. Larson	
CMIC at INEX 2007: Book Search Track	197
W. Magdy, K. Darwish	
DOCUMENT MINING	
<hr/>	
XML Document Classification using Extended VSM	200
J. Yang, F. Zhang	
A Categorization Approach for Wikipedia Collection Based on Negative Category Information and Initial Descriptions	212
M. S. Murugesan, K. Lakshmi, S. Mukherjee	
Document Clustering using Incremental and Pairwise Approaches	215
T. Tran, R. Nayak	
Rare Patterns to Improve Path-Based Clustering of Wikipedia Articles	224
J. Yao, N. Zerida	
Probabilistic Methods for Structured Document Classification at INEX'07	232
L. M. de Campos, J. M. Fernandez-Luna, J. F. Huete, A. E. Romero	
Clustering XML Documents using Closed Frequent Subtrees-A Structure-Only Based Approach	246
S. Kutty, T. Tran, R. Nayak, Y. Li	

Efficient Clustering of Structured Documents using Graph Self-Organizing Maps	257
M. Hagenbuchner, A.C. Tsoi, A. Sperduti, M. Kc	

ENTITY RANKING

Multitype-Topic Models for Entity Ranking	261
H. Shiozaki, K. Eguchi	
An n-gram and Description-Checking Based Approach for Entity Ranking Track	269
M. S. Murugesan, S. Mukherjee	
Structured Document Retrieval, Multimedia Retrieval, and Entity Ranking Using PF/Tijah	273
T. Tsikrika, P. Serdyukov, H. Rode, T. Westerveld, R. Aly, D. Hiemstra, A. P. de Vries	
Experiments on Category Expansion at INEX 2007	287
J. Jämsen, T. Näppilä, P. Arvola	
Using Wikipedia Categories and Links in Entity Ranking	297
A.-M. Vercoistre, J. Pehcevski, J. A. Thom	
Integrating Document Features for Entity Ranking	312
J. Zhu, D. Song, S. Rüger	
L3S Research Center at the INEX Entity Ranking Track	317
G. Demartini, C. S. Firan, T. Iofciu	
Entity Ranking using XML Retrieval Techniques	326
M. S. Ali, M. P. Consens, S. Khatchadourian	

HETEROGENEOUS COLLECTIONS

Retrieval of Document Parts using Bayesian Networks and Entropy as a Degree of (Dis)organization	327
C. Estombelo-Montesco, D. Chiodi, T. Kudo, A. Serra-Neto, F. P. de Almeida Prado, A. A. Macedo	

INTERACTIVE EXPERIMENTS

How Task Affects Information Search	337
E. G. Toms, T. MacKenzie, C. Jordan, H. O'Brien, L. Freund, S. Toze, E. Dawe, A. MacNutt	
A Comparison of Interactive and Ad-Hoc Relevance Assessments	342
B. Larsen, S. Malik, A. Tombros	

LINK-THE-WIKI

Overview of INEX 2007 Link the Wiki Track	350
W. C. Huang, Y. Xu, S. Geva	
Wikipedia Ad Hoc Passage Retrieval and Wikipedia Document Linking	365
D. Jenkinson, A. Trotman	

University of Waterloo at INEX2007: Ad Hoc and Link-the-Wiki Tracks K. Y. Itakura, C. L. A. Clarke	380
The University of Amsterdam at INEX 2007 K. N. Fachry, J. Kamps, M. Koolen, J. Zhang	388
GPX@INEX2007: Ad-Hoc Queries and Automated Link Discovery in the Wikipedia S. Geva	403
<hr/> MULTIMEDIA <hr/>	
Report on the INEX 2007 Multimedia Track T. Tsikrika, T. Westerveld	410
MM-XFIRM at INEX Multimedia track 2007 M. Torjmen, K. Pinel-Sauvagnat, M. Boughanem	423
An XML Fragment Retrieval Method with Image and Text using Textual Information Retrieval Techniques Y. Suzuki, M. Mitsukawa, K. Hatano, T. Shimizu, J. Miyazaki, H. Kinutani	433
<hr/> APPENDIX <hr/>	
AD HOC	
INEX 2007 Guidelines for Topic Development A. Trotman, B. Larsen, <i>et al.</i>	436
INEX 2007 Retrieval Task and Result Submission Specification C. L. A. Clarke, J. Kamps, M. Lalmas	445
INEX 2007 Relevance Assessment Guide M. Lalmas, B. Piwowarski	454
BOOK SEARCH	
INEX 2007 Book Search Track Topic Development Guidelines G. Kazai	464
INEX 2007 Book Search Track Tasks and Submission Guidelines G. Kazai	473
ENTITY RANKING	
INEX 2007 Entity Ranking Track Guidelines A. P. de Vries, J. A. Thom, A.-M. Vercoustre, N. Craswell, M. Lalmas	481
LINK-THE-WIKI	
INEX 2007 Link the Wiki Task and Result Submission Specification S. Geva, A. Trotman	487

MULTIMEDIA

INEX 2007 Multimedia Track: Guidelines for Topic Development for the MMimages Task	491
T. Westerveld, T. Tsikrika, <i>et al.</i>	
INEX 2007 Multimedia Track: Specification of Retrieval Tasks and Result Submissions	501
T. Tsikrika T. Westerveld	

ORGANIZERS

PROJECT LEADERS

Norbert Fuhr (University of Duisburg-Essen)
Mounia Lalmas (Queen Mary University of London)
Andrew Trotman (University of Otago)

CONTACT PEOPLE

Saadia Malik (University of Duisburg-Essen)
Zoltán Szlávik (Queen Mary University of London)

WIKIPEDIA DOCUMENT COLLECTION AND EXPLORATION

Ludovic Denoyer (Université Paris 6)

DOCUMENT EXPLORATION

Ralf Schenkel (Max-Planck-Institut für Informatik)
Martin Theobald (Stanford University)

TOPIC FORMAT SPECIFICATION

Birger Larsen (Royal School of Library and Information Science)
Andrew Trotman (University of Otago)

TASK DESCRIPTION

Jaap Kamps (University of Amsterdam)
Charlie Clarke (University of Waterloo)

ONLINE RELEVANCE ASSESSMENT TOOL

Benjamin Piwowarski (Yahoo! Research Latin America)

EFFECTIVENESS MEASURES

Gabriella Kazai (Microsoft Research Cambridge)
Benjamin Piwowarski (Yahoo! Research Latin America)
Jaap Kamps (University of Amsterdam)
Jovan Pehcevski (INRIA-Rocquencourt)
Stephen Robertson (Microsoft Research Cambridge)
Paul Ogilvie (Carnegie Mellon University)

DOCUMENT MINING TRACK

Ludovic Denoyer (Université Paris 6)
Patrick Gallinari (Université Paris 6)

MULTIMEDIA TRACK

Thijs Westerveld (CWI)
Theodora Tsirikas (CWI)

ENTITY RANKING TRACK

Arjen de Vries (CWI)
Nick Craswell (Microsoft Research Cambridge)
James A. Thom (RMIT University)
Anne-Marie Vercoustre (INRIA-Rocquencourt)
Mounia Lalmas (Queen Mary University of London)

LINK-THE-WIKI TRACK

Shlomo Geva (Queensland University of Technology)
Andrew Trotman (University of Otago)

BOOK SEARCH TRACK

Gabriella Kazai (Microsoft Research Cambridge)
Antoine Doucet (INRIA – IRISA)

PREFACE

Welcome to the 6th workshop of the Initiative for the Evaluation of XML Retrieval (INEX)!

Now, in its sixth year, INEX is an established evaluation forum for XML information retrieval (IR), with over 90 participating organizations worldwide. Its aim is to provide an infrastructure, in the form of a large XML test collection and appropriate scoring methods, for the evaluation of XML IR systems.

XML IR plays an increasingly important role in many information access systems (e.g. digital libraries, web, intranet) where content is more and more a mixture of text, multimedia, and metadata, formatted according to the adopted W3C standard for information repositories, the so-called eXtensible Markup Language (XML). The ultimate goal of such systems is to provide the right content to their end-users. However, while many of today's information access systems still treat documents as single large (text) blocks, XML offers the opportunity to exploit the internal structure of documents in order to allow for more precise access, thus providing more specific answers to user requests. Providing effective access to XML-based content is therefore a key issue for the success of these systems.

2007 was an exciting year for INEX, and brought a lot of changes. In total eight research tracks were included, which studied different aspects of XML information access: Ad-hoc, Document Mining, Multimedia, Entity Ranking, Link-the-Wiki, and Book Search. The Heterogeneous Track and Interactive track were run as extensions of the 2006 tracks. The Link-the-Wiki and Book Search tracks were new for 2007. The consolidation of the existing tracks, and the expansion to new areas offered by the two new tracks, allows INEX to grow in reach.

The aim of the INEX 2007 workshop is to bring together researchers in the field of XML IR who participated in the INEX 2007 campaign. During the past year participating organizations contributed to the building of a large-scale XML test collection by creating topics, performing retrieval runs and providing relevance assessments. The workshop concludes the results of this large-scale effort, summarizes and addresses encountered issues and devises a work plan for the future evaluation of XML retrieval systems.

ACKNOWLEDGEMENTS

INEX is funded by the DELOS Network of Excellence on Digital Libraries, to which we are very thankful. We would also like to thank the Wikipedia and Microsoft for providing us the XML document collections.

We gratefully thank organizers of the various tracks for their great work in setting up the new tracks, and carrying on and refining the existing tracks. Thanks also to those involved in running and coordinating the *ad hoc* track which each year involves a major effort.

As always, special thanks go to the participating organizations and people for their contributions and hard work throughout the year! The first point of contact of many of us is either Saadia Malik or Zoltán Szlávik for whom we, and we are sure every participant, give thanks.

At the conclusion of the 2007 campaign, Mounia Lalmas and Norbert Fuhr will step down from the project leader role. Without a doubt INEX would not be the success it is without the commitment of these two founders. They were responsible for securing DELOS funding between 2002 and 2007, for securing Schloss Dagstuhl as a workshop venue, and for attracting the now more than 90 participants. To the founders INEX will always be thankful.

We hope you have enjoyed the INEX 2007 campaign and have fruitful and stimulating discussions at the workshop.

Norbert Fuhr, University of Duisburg-Essen
Mounia Lalmas, Queen Mary, University of London
Andrew Trotman, University of Otago

December 2007

SCHLOSS DAGSTUHL



Schloss Dagstuhl or Dagstuhl manor house was built in 1760 by the then reigning prince Count Anton von Öttingen-Soetern-Hohenbaldern. After the French Revolution and occupation by the French in 1794, Dagstuhl was temporarily in the possession of a Lorraine ironworks.

In 1806 the manor house along with the accompanying lands was purchased by the French Baron Wilhelm de Lasalle von Louisenthal.

In 1959 the House of Lasalle von Louisenthal died out, at which time the manor house was then taken over by an order of Franciscan nuns, who set up an old-age home there.



In 1989 the Saarland government purchased the manor house for the purpose of setting up the International Conference and Research Center for Computer Science.

The first seminar in Dagstuhl took place in August of 1990. Every year approximately 2,000 research scientists from all over the world attend the 30-35 Dagstuhl Seminars and an equal number of other events hosted at the center.



<http://www.dagstuhl.de/>

Overview of the INEX 2007 Ad Hoc Track

Norbert Fuhr¹, Jaap Kamps², Mounia Lalmas³, Saadia Malik¹, and Andrew Trotman⁴

¹ University of Duisburg-Essen, Duisburg, Germany
{norbert.fuhr,saadia.malik}@uni-due.de

² University of Amsterdam, Amsterdam, The Netherlands
kamps@science.uva.nl

³ Queen Mary, University of London, London, UK
lalmas@dcs.qmul.uk.ac

⁴ University of Otago, Dunedin, New Zealand
andrew@cs.otago.ac.nz

Abstract. This paper gives an overview of the INEX 2007 Ad Hoc Track. The main purpose of the Ad Hoc Track was to investigate the value of the internal document structure (as provided by the XML mark-up) for retrieving relevant information. For this reason, the retrieval results were liberalized to arbitrary passages and measures were chosen to fairly compare systems retrieving elements, ranges of elements, and arbitrary passages. The INEX 2007 Ad Hoc Track featured three tasks: For the *Focused Task* a ranked-list of non-overlapping results (elements or passages) was needed. For the *Relevant in Context Task* non-overlapping results (elements or passages) were returned grouped by the article from which they came. For the *Best in Context Task* a single starting point (element start tag or passage start) for each article was needed. We discuss the results for the three tasks, examine the relative effectiveness of element and passage retrieval. This is examined in the context of content only (CO, or Keyword) search as well as content and structure (CAS, or structured) search.

1 Introduction

This paper gives an overview of the INEX 2007 Ad Hoc Track. The main research question underlying the Ad Hoc Track is that of the value of the internal document structure (mark-up) for retrieving relevant information. That is, does the document structure help in identify where the relevant information is within a document? This question has recently attracted a lot of attention. Trotman and Geva [13] argued that, since INEX relevance assessments are not bound to XML element boundaries, retrieval systems should also not be bound to XML element boundaries. Their implicit assumption is that a system returning passages is at least as effective as a system returning XML elements. This assumption is based on the observation that elements are of a lower granularity than passages and so all elements can be described as passages. The reverse, however is not

true and only some passages can be described as elements. Huang et al. [6] implement a fixed window passage retrieval system and show that a comparable element retrieval ranking can be derived. In a similar study, Itakura and Clarke [7] show that although ranking elements based on passage-evidence is comparable, a direct estimation of the relevance of elements is superior. Finally, Kamps and Koolen [8] study the relation between the passages highlighted by the assessors and the XML structure of the collection directly, showing reasonable correspondence between the document structure and the relevant information.

Up to now, element and passage retrieval approaches could only be compared when mapping passages to elements. This may significantly affect the comparison, since the mapping is non-trivial and, of course, turns the passage retrieval approaches effectively into element retrieval approaches. To study the value of the document structure through direct comparison of element and passage retrieval approaches, the retrieval results for INEX 2007 were liberalized to arbitrary passages. Every XML element is, of course, also a passage of text.

The evaluation measures are now based directly on the highlighted passages, or arbitrary best-entry points, as identified by the assessors. As a result it is now possible to fairly compare systems retrieving elements, ranges of elements, or arbitrary passages. These changes address earlier requests to liberalize the retrieval format to ranges of elements [1] and later requests to liberalize to arbitrary passages of text [13].

The INEX 2007 Ad Hoc Track featured three tasks:

1. For the *Focused Task* a ranked-list of non-overlapping results (elements or passages) must be returned. It is evaluated at early precision relative to the highlighted (or believed relevant) text retrieved.
2. For the *Relevant in Context Task* non-overlapping results (elements or passages) must be returned, these are grouped by document. It is evaluated by mean average generalized precision where the generalized score per article is based on the retrieved highlighted text.
3. For the *Best in Context Task* a single starting point (element's starting tag or passage offset) per article must be returned. It is also evaluated by mean average generalized precision but with the generalized score (per article) based on the distance to the assessor's best-entry point.

The *Thorough Task* as defined in earlier INEX rounds is discontinued. We discuss the results for the three tasks, giving results for the top 10 participating groups and discussing the best scoring approaches in detail. We also examine the relative effectiveness of element and passage runs, and with content only (CO) queries and content and structure (CAS) queries.

The rest of the paper is organized as follows. First, Section 2 describes the INEX 2007 Ad Hoc retrieval tasks and measures. Section 3 details the collection, topics, and assessments of the INEX 2007 Ad Hoc Track. In Section 4, we report the results for the Focused Task (Section 4.2); the Relevant in Context Task (Section 4.3); and the Best in Context Task (Section 4.4). Section 5 details particular types of runs (such as CO versus CAS, and element versus passage),

and on particular subsets of the topics (such as topics with a non-trivial CAS query). Finally, in Section 6, we discuss our findings and draw some conclusions.

2 Ad Hoc Retrieval Track

In this section, we briefly summarize the ad hoc retrieval tasks and the submission format (especially how elements and passages are identified). We also summarize the metrics used for evaluation. For more detail the reader is referred to the formal specification documents [2] and [10].

2.1 Tasks

Focused Task The scenario underlying the Focused Task is the return, to the user, of a ranked list of elements or passages for their topic of request. The Focused Task requires systems to find the most focused results that satisfy a information need, where by focused we mean without returning “overlapping” elements (shorter is preferred in the case of equally relevant elements). Since ancestors elements and longer passages are always relevant (to a greater or lesser extent) it is a challenge to chose the correct granularity.

The task has a number of assumptions:

Display the results are presented to the user as a ranked-list of results.

Users view the results top-down, one-by-one.

Relevant in Context Task The scenario underlying the Relevant in Context Task is the return of a ranked list of articles and within those articles the relevant information (captured by a set of non-overlapping elements or passages). A relevant article will likely contain relevant information that could be spread across different elements. The task requires systems to find a set of results that corresponds well to all relevant information in each relevant article. The task has a number of assumptions:

Display results will be grouped per article, in their original document order, access will be provided through further navigational means, such as a document heat-map or table of contents.

Users consider the article to be the most natural retrieval unit, and prefer an overview of relevance within this context.

Best in Context Task The scenario underlying the Best in Context Task is the return of a ranked list of articles and the identification of a best-entry-point from which a user should start reading each article in order to satisfy the information need. Even an article completely devoted to the topic of request will only have one best starting point from which to read (even if that is the beginning of the article). The task has a number of assumptions:

Display a single result per article.

Users consider articles to be natural unit of retrieval, but prefer to be guided to the best point from which to start reading the most relevant content.

2.2 Submission Format

Since XML retrieval approaches may return arbitrary results from within documents, a way to identify these nodes is needed.

XML element results are identified by means of a file name and an element (node) path specification. File names in the Wikipedia collection are unique so that (with the .xml extension removed), for example:

```
<file>9996</file>
```

identifies 9996.xml as the target document from the Wikipedia collection. Element paths are given in XPath, but only fully specified paths are allowed. For example:

```
<path>/article[1]/body[1]/section[1]/p[1]</path>
```

identifies the first “article” element, then within that, the first “body” element, then the first “section” element, and finally within that the first “p” element. Importantly, XPath counts elements from 1 and counts element types. For example if a section had a title and two paragraphs then their paths would be: title[1], p[1] and p[2].

A result element, then, is identified unambiguously using the combination of file name and element path, for example:

```
<result>
  <file>9996</file>
  <path>/article[1]/body[1]/section[1]/p[1]</path>
  <rsv>0.9999</rsv>
</result>
```

Passages are given in the same format, but extended for optional character-offsets. As a passage need not start and end in the same element, each is given separately. The following example is equivalent to the element result example above since it starts and ends on an element boundary.

```
<result>
  <file>9996</file>
  <passage start="/article[1]/body[1]/section[1]/p[1]"
    end="/article[1]/body[1]/section[1]/p[1]"/>
  <rsv>0.9999</rsv>
</result>
```

In the next passage example the result starts 85 characters after the start of the paragraph and continues until 106 characters after a list item in list. The end location is, of course, after the start location.

```
<result>
  <file>9996</file>
  <passage start="/article[1]/body[1]/section[1]/p[1]/text()[1].85"
    end="/article[1]/body[1]/section[1]/normallist[1]/item[2]/text()[2].106"/>
  <rsv>0.6666</rsv>
</result>
```


The result can start anywhere in any text node. Character positions count from 0 (before the first character) to the *node-length* (after the last character). A detailed example is provided in [2].

2.3 Measures

We briefly summarize the main measures used for the Ad Hoc Track (see Kamps et al. [10] for details). The main change at INEX 2007 is the inclusion of arbitrary passages of text. Unfortunately this simple change has necessitated the deprecation of element-based metrics used in prior INEX campaigns because the “natural” retrieval unit is no longer an element, so elements cannot be used as the basis of measure. We note that properly evaluating the effectiveness in XML-IR remains an ongoing research question at INEX.

The INEX 2007 measures are solely based on the retrieval of highlighted text. We simplify all INEX tasks to highlighted text retrieval and assume that systems return all, and only, highlighted text. We then compare the characters of text retrieved by a search engine to the number and location of characters of text identified as relevant by the assessor. For best in context we use the distance between the best entry point in the run to that identified by an assessor.

Focused Task Recall is measured as the fraction of all highlighted text that has been retrieved. Precision is measured as the fraction of retrieved text that was highlighted. The notion of rank is relatively fluid for passages so we use an interpolated precision measure which calculates interpolated precision scores at selected recall levels. Since we are most interested in what happens in the first retrieved results, the INEX 2007 official measure is interpolated precision at 1% recall (iP[0.01]). We also present interpolated precision at other early recall points, and (mean average) interpolated precision over 101 standard recall points (0.00, 0.01, 0.02, ..., 1.00) as an overall measure.

Relevant in Context Task The evaluation of the Relevant in Context Task is based on the measures of generalized precision and recall [11], where the per document score reflects how well the retrieved text matches the relevant text in the document. Specifically, the per document score is the harmonic mean of precision and recall in terms of the fractions of retrieved and highlighted text in the document. We are most interested in overall performances so the main measure is mean average generalized precision (MAgP). We also present the generalized precision scores at early ranks (5, 10, 25, 50).

Best in Context Task The evaluation of the Best in Context Task is based on the measures of generalized precision and recall where the per document score reflects how well the retrieved entry point matches the best entry point in the document. Specifically, the per document score is a linear discounting function

of the distance d (measured in characters)

$$\frac{n - d(x, b)}{n}$$

for $d < n$ and 0 otherwise. We use $n = 1,000$ which is roughly the number of characters corresponding to the visible part of the document on a screen. We are most interested in overall performance, and the main measure is mean average generalized precision (MAGP). We also show the generalized precision scores at early ranks (5, 10, 25, 50).

3 Ad Hoc Test Collection

In this section, we discuss the corpus, topics, and relevance assessments used in the Ad Hoc Track.

3.1 Corpus

The document collection was the Wikipedia XML Corpus based on an XML'ified version of the English Wikipedia in early 2006 [3]. The Wikipedia collection contains 659,338 Wikipedia articles. On average an article contains 161 XML nodes, where the average depth of a node in the XML tree of the document is 6.72.

The original Wiki syntax has been converted into XML, using both general tags of the layout structure (like *article*, *section*, *paragraph*, *title*, *list* and *item*), typographical tags (like *bold*, *emphatic*), and frequently occurring link-tags. For details see Denoyer and Gallinari [3].

3.2 Topics

The ad hoc topics were created by participants following precise instructions given elsewhere [14]. Candidate topics contained a short CO (keyword) query, an optional structured CAS query, a one line description of the search request, and narrative with a details of the topic of request and the task context in which the information need arose. Figure 1 presents an example of an Ad Hoc topic. Based on the submitted candidate topics, 130 topics were selected for the INEX 2007 Ad Hoc track, there were given INEX topic numbers 414–543.

The INEX 2007 Multimedia track also had an ad hoc search task and 19 topics were used for both the Ad Hoc track and the Multimedia track. They were designated topics 525–543. Table 1 presents the topics shared between the Ad Hoc and Multimedia tracks. Six of these topics (527, 528, 530, 532, 535, 540) have an additional `<mmtitle>` field, a multimedia query.

The 12 INEX 2006 iTrack topics were also inserted into the topic set (as topics 512-514, and 516-524) as these topics were not assessed in 2006. Table 2 presents the 12 INEX 2006 iTrack topics, and their corresponding Ad Hoc track topic numbers.

```

<inex_topic topic_id="414" ct_no="3">
  <title>hip hop beat</title>
  <castitle>/**[about(., hip hop beat)]</castitle>
  <description>what is a hip hop beat?</description>
  <narrative>
    To solve an argument with a friend about hip hop music and beats, I
    want to learn all there is to know about hip hop beats. I want to know
    what is meant by hip hop beats, what is considered a hip hop beat,
    what distinguishes a hip hop beat from other beats, when it was
    introduced and by whom. I consider elements relevant if they
    specifically mention beats or rythm. Any element mentioning hip hop
    music or style but doesn't discuss abything about beats or rythm is
    considered not relevant. Also, elements discussing beats and rythm,
    but not hip hop music in particular, are considered not relevant.
  </narrative>
</inex_topic>

```

Fig. 1. INEX Ad Hoc Track topic 414.

Table 1. Topics shared with the INEX 2007 Multimedia track.

Topic	Title-field
525	potatoes in paintings
526	pyramids of egypt
527	walt disney land world
528	skyscraper building tall towers
529	paint works museum picasso
530	Hurricane satellite image
531	oil refinery or platform photographs
532	motor car
533	Images of phones
534	Van Gogh paintings
535	japanese garden old building -chapel
536	Ecuador volcano climbing quito
537	pictures of Mont Blanc
538	photographer photo
539	self-portrait
540	war map place
541	classic furniture design chairs
542	Images of tsunami
543	Tux

Table 2. iTrack 2006 topics.

iTrack	Ad hoc	Title-field	Type	Structure
1	519	types of bridges vehicles water ice	Decision making	Hierarchical
2	512	french impressionism degas monet renoir impressionist movement	Decision making	Hierarchical
3	520	Chartres Versailles history architecture travelling	Decision making	Parallel
4	516	environmental effects mining logging	Decision making	Parallel
5	521	red ants USA bites treatment	Fact finding	Hierarchical
6	513	chanterelle mushroom poisonous deadly species	Fact finding	Hierarchical
7	522	April 19th revolution peaceful revolution velvet revolution quiet revolution	Fact finding	Parallel
8	517	difference fortress castle	Fact finding	Parallel
9	523	fuel efficient cars	Info gathering	Hierarchical
10	514	food additives physical health risk grocery store labels	Info gathering	Hierarchical
11	524	home heating solar panels	Info gathering	Parallel
12	518	tidal power wind power	Info gathering	Parallel

3.3 Judgments

Topics were assessed by participants following precise instructions [12]. The assessors used Piwowarski’s X-RAI assessment system that assists assessors in highlight relevant text. Topic assessors were asked to mark all, and only, relevant text in a pool of documents. The granularity of assessment was roughly a sentence. After assessing each article a separate best entry point decision was made by the assessor. The Focused and Relevant in Context Tasks were evaluated against the text highlighted by the assessors, whereas the Best in Context Task was evaluated against the best-entry-points.

The relevance judgments were frozen on October 29, 2007, at 11:56. At this time 99 topics had been fully assessed. Moreover, 7 topics were judged by two separate assessors, each without the knowledge of the other. All results in this paper refer to the 99 topics for which judgments had been completed on October 29.

- The 99 assessed topics were: 414-431, 433-436, 439, 440, 444-450, 453, 454, 458, 459, 461-463, 465, 467, 468, 470-475, 477, 479-491, 498-500, 502, 503, 505, 507-509, 511, 515-523, and 525-543.
- All 19 Multimedia topics, 525-543, were assessed.
- Only 8 of the 12 iTrack 2006 topics, 516-523, were assessed.

Table 3 presents statistics of the number of judged and relevant articles, and passages. In total 60,536 articles were judged. Relevant passages were found in 6,014 articles. The mean number of relevant articles per topic is 60, but the distribution is skewed with a median of 36. There were 10,818 highlighted passages. The mean was 109 passages and the median was 62 passages per topic.

Table 3. Statistics over judged and relevant articles per topic.

	total		per topic				
	topics	number	min	max	median	mean	st.dev
judged articles	99	60,536	600	671	609	611	10.50
articles with relevance	99	6,014	2	479	36	60	72.66
highlighted passages	99	10,818	2	832	62	109	155.07

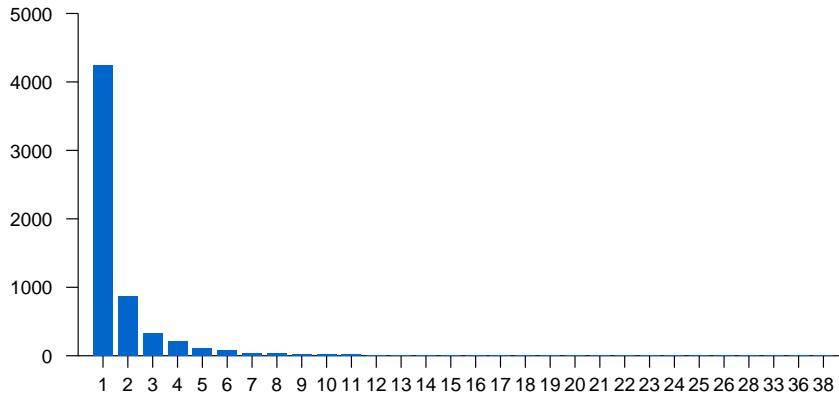


Fig. 2. Distribution of passages over articles.

Figure 2 presents the number of articles with the given number of passages. The vast majority of relevant articles (4,247 out of 6,014) had only a single highlighted passage, and the number of passages quickly tapers off.

4 Ad Hoc Retrieval Results

In this section, we discuss, for the three ad hoc tasks, the participants and their results.

4.1 Participation

216 runs were submitted by 27 participating groups. Table 4 lists the participants and the number of runs they submitted, also broken down over the tasks (Focused, Relevant in Context, or Best in Context); the used query (Content-Only or Content-And-Structure); and the used result type (Element or Passage). Participants were allowed to submit up to three CO-runs per task and three CAS-runs per task (for all three tasks). This totaled to 18 runs per participant.¹ The submissions are spread well over the ad hoc retrieval tasks with 79 submissions

¹ As it turns out, three groups submitted more runs than allowed: *mines* submitted 1 extra CO-run, and both *lip6* and *qutau* submitted 6 extra CO-runs each. At this moment, we have not decided on any repercussions other than mentioning them in this footnote.

Table 4. Participants in the Ad Hoc Track.

Participant	Full name	Foc	RiC	BiC	CO	CAS	Ele	Pas	Total
cmu	Language Technologies Institute, School of Computer Science, Carnegie Mellon University	1	0	0	1	0	1	0	1
eurise	Laboratoire Hubert Curien - Universit de Saint-Etienne	2	0	0	2	0	2	0	2
indstainst	Indian Statistical Institute	2	0	0	2	0	2	0	2
inria	INRIA-Rocquencourt- Axis	3	3	3	9	0	9	0	9
irit	IRIT	0	0	2	1	1	2	0	2
justsystem	JustSystems Corporation	6	6	6	9	9	18	0	18
labcsiro	Information Engineering lab, ICT Centre, CSIRO	1	0	0	1	0	1	0	1
lip6	LIP6	5	5	5	15	0	15	0	15
maxplanck	Max-Planck-Institut fuer Informatik	4	4	4	6	6	12	0	12
mines	Ecoles des Mines de Saint-Etienne, France	3	4	3	10	0	10	0	10
qutau	Queensland University of Technology	7	7	7	15	6	21	0	21
rmit	RMIT University	1	1	1	3	0	3	0	3
uamsterdam	University of Amsterdam	6	6	6	9	9	18	0	18
udalian	Dalian University of Technology	6	6	6	9	9	18	0	18
udoshisha	Doshisha University	2	0	0	1	1	2	0	2
ugrenoble	CLIPS-IMAG	3	3	3	9	0	9	0	9
uhelsinki	University of Helsinki	2	0	0	2	0	2	0	2
uminnesota	University of Minnesota Duluth	1	2	2	5	0	5	0	5
uniKaislau	University of Kaiserslautern, AG DBIS	3	3	0	6	0	6	0	6
unigordon	Information Retrieval and Interaction Group, The Robert Gordon University	3	3	3	9	0	9	0	9
unigranada	University of Granada	3	3	5	8	3	11	0	11
unitoronto	University of Toronto	2	0	0	0	2	2	0	2
uotago	University of Otago	3	3	3	9	0	0	9	9
utampere	University of Tampere	3	3	3	9	0	9	0	9
utwente	Cirquid Project (CWI and University of Twente)	3	2	1	6	0	6	0	6
uwaterloo	University of Waterloo	2	0	4	6	0	6	0	6
uwuhan	Center for Studies of Information Resources, School of Information Management, Wuhan University, China	2	2	4	8	0	8	0	8
Total	runs	79	66	71	170	46	207	9	216

Table 5. Top 10 Participants in the Ad Hoc Track Focused Task.

Participant	iP[0.00]	iP[0.01]	iP[0.05]	iP[0.10]	MAiP
udalian-5	0.4380	0.4259	0.3457	0.3162	0.1401
maxplanck-3	0.4744	0.4149	0.3211	0.2902	0.1115
udoshisha-0	0.4257	0.3988	0.3204	0.2762	0.1154
uamsterdam-2	0.4780	0.3938	0.3236	0.2974	0.1326
uwaterloo-0	0.4118	0.3853	0.3257	0.2928	0.1318
qutau-20	0.4086	0.3842	0.3433	0.3208	0.1541
inria-2	0.3955	0.3794	0.3464	0.3152	0.1775
rmit-0	0.3955	0.3788	0.3446	0.3175	0.1804
unigordon-1	0.4073	0.3786	0.3271	0.3054	0.1552
mines-2	0.4595	0.3762	0.2477	0.2100	0.0865

for Focused, 66 submissions for Relevant in Context, and 71 submissions for Best in Context.

4.2 Focused Task

We now discuss the results of the Focused Task in which a ranked-list of non-overlapping results (elements or passages) was required. The official measure for the task was (mean) interpolated precision at 1% recall (iP[0.01]). Table 5 shows the best run of the top 10 participating groups. The first column gives the participant, see Table 4 for the full name of group, and see Appendix 6 for the precise run label. The second to fifth column give the interpolated precision at 0%, 1%, 5%, and 10% recall. The sixth column gives mean average interpolated precision over 101 standard recall levels (0%, 1%, . . . , 100%).

Here we briefly summarize what is currently known about the experiments conducted by the top five groups (based on official measure for the task, iP[0.01]).

Dalian University of Technology Using the CAS query. Only index the content contained by the tags often occur or retrieved by users. Use the BM25 retrieval model and pseudo-relevance feedback. Both document retrieval and document parts retrieval, and then combine the document score and document parts score. Further special handlings on the category of topics finding images, by removing the returned elements whose structural paths contained “image” or “figure” tags to the top one by one. Overlap was removed in the order of the resulting run.

Max-Planck-Institut für Informatik Using the CAS query: the basis for this run is an ad hoc CAS run were the target tag was evaluated strictly, i.e., a result was required to have the tag specified as target in the query and match at least one of the content conditions, whereas support conditions were optional; phrases and negations in the query were ignored. To produce the focused run, elements were removed in case they overlap with a higher scoring element for the same topic.

Doshisha University Using the CO query. Used a term-weighting approach like the *tf.ipf* (term frequency times inverted path frequency) scoring proposed by Grabs and Schek [5] to get ranked search result, where the log of the *tf* is taken. Small-sized XML fragments were removed. The smaller the size an XML fragments is, the smaller the scores of the XML fragment in our scoring method.

University of Amsterdam Using the CO query. Having an index containing all elements, a language model was used with a standard length prior and an incoming links prior. The focused run was created by list-based removal of overlapping elements.

University of Waterloo Using the CO query. Query terms were formed by transforming each topic title into a disjunctive form, less negative query terms. Wumpus [15] was used to obtain positions of query terms and XML elements. The most frequently occurring XML elements in the corpus were listed and ranked using Okapi BM25. Nested results were removed for the Focused task.

Based on the information from these and other participants:

- The two best scoring teams used the CAS query. Hence using the structural hints, even strict adherence to the target tag, seemed to promote early precision
- More generally, limiting the retrieved types of elements, either at indexing time (by selecting elements based on tag type or length) or at retrieval time (by enforcing CAS target elements, or using length-priors), seems to promote early precision.
- The system at rank eight, *rmit-0*, is retrieving only full articles.

4.3 Relevant in Context Task

We now discuss the results of the Relevant in Context Task in which non-overlapping results (elements or passages) need to be returned grouped by the article they came from. The task was evaluated using generalized precision where the generalized score per article was based on the retrieved highlighted text. The official measure for the task was mean average generalized precision (MAGP).

Table 6 shows the top 10 participating groups (only the best run per group is shown) in the Relevant in Context Task. The first column lists the participant, see Table 4 for the full name of group, and see Appendix 6 for the precise run label. The second to fifth column list generalized precision at 5, 10, 25, 50 retrieved articles. The sixth column lists mean average generalized precision.

Here we briefly summarize the information available about the experiments conducted by the top five groups (based on MAGP).

Dalian University of Technology Using the CO query. See the description for the Focused Task above. Cluster the returned elements per document, and remove overlap top-down.

Table 6. Top 10 Participants in the Ad Hoc Track Relevant in Context Task.

Participant	gP[5]	gP[10]	gP[25]	gP[50]	MAgP
udalian-16	0.1735	0.1513	0.1242	0.0985	0.1013
qutau-18	0.1879	0.1522	0.1136	0.0890	0.0975
rmit-1	0.1698	0.1554	0.1152	0.0878	0.0884
uamsterdam-4	0.1732	0.1487	0.1086	0.0831	0.0860
unigordon-7	0.1650	0.1421	0.1087	0.0810	0.0812
utwente-5	0.1424	0.1211	0.0978	0.0767	0.0784
inria-5	0.1698	0.1554	0.1208	0.0873	0.0752
maxplanck-8	0.1491	0.1252	0.0890	0.0701	0.0747
justsystem-14	0.1230	0.1074	0.0854	0.0645	0.0734
mines-9	0.1406	0.1195	0.0836	0.0628	0.0656

Queensland University of Technology Using the CO query: Plural/singular expansion was used on the query, as well as removal of words preceded by a minus sign. GPX [4] was used to rank elements, based on a leaf-node index and $tf \cdot icf$ (term frequency times inverted collection frequency) weighting modified by i) the number of unique terms, ii) the proximity of query-term matches, and iii) boosting of query-term occurrences in the name field. All leaf-node-scores were normalized by their length, and the overall article’s similarity score was added. The score of elements was calculated directly from the content of the nodes, obviating the need for score propagation with decaying factors.

RMIT University Using the CO query. This is a baseline article run using Zettair [16] with the Okapi similarity measure with default settings. The title from each topic was automatically translated as an input query to Zettair. The similarity of an article to a query determines its final rank.

University of Amsterdam Using the CO query. Having an index with only the “container” elements – elements that frequently contain an entire highlighted passage at INEX 2006 – basically corresponding to the main layout structure. A language model was used with a standard length prior and an incoming links prior, after list-based removal of overlapping elements the final results are clustered per article on a first-come, first-served basis. See the description for the Focused Task above.

Robert Gordon University Using the CO query. An element’s score was computed by a mixture language model combining estimates based on element full-text and a “summary” of it (i.e., extracted titles, section titles, and figure captions nested inside the element). A prior was used according to an element’s location in the original text, and the length of its path. For the post-processing, they filter out redundant elements by selecting the highest scored element from each of the paths. Elements are reordered so that results from the same article are grouped together.

Based on the information from these and other participants:

Table 7. Top 10 Participants in the Ad Hoc Track Best in Context Task.

Participant	gP[5]	gP[10]	gP[25]	gP[50]	MAgP
rmit-2	0.3564	0.3296	0.2566	0.1950	0.1951
uwaterloo-3	0.2651	0.2513	0.2194	0.1722	0.1842
qutau-19	0.3246	0.2710	0.2104	0.1711	0.1823
udalian-7	0.2504	0.2443	0.1995	0.1575	0.1771
unigordon-2	0.3481	0.2953	0.2299	0.1765	0.1759
uamsterdam-16	0.3311	0.2906	0.2266	0.1775	0.1736
inria-8	0.3564	0.3296	0.2616	0.1960	0.1655
justsystem-7	0.2844	0.2655	0.1994	0.1561	0.1624
utwente-2	0.2546	0.2234	0.1794	0.1419	0.1338
maxplanck-6	0.2039	0.2060	0.1729	0.1320	0.1326

- Solid article ranking seems a prerequisite for good overall performance, with third best run retrieving only full articles.
- The use of the structured query does not appear to promote overall performance: all five groups submitting a CAS query run had a superior CO query run.

4.4 Best in Context Task

We now discuss the results of the Best in Context Task in which documents were ranked on topical relevance and a single best entry point into the document was identified. The Best in Context Task was evaluated using generalized precision but here the generalized score per article was based on the distance to the assessor’s best-entry point. The official measure for the task was mean average generalized precision (MAgP).

Table 7 shows the top 10 participating groups (only the best run per group is shown) in the Best in Context Task. The first column lists the participant, see Table 4 for the full name of group, and see Appendix 6 for the precise run label. The second to fifth column list generalized precision at 5, 10, 25, 50 retrieved articles. The sixth column lists mean average generalized precision.

Here we briefly summarize the information available about the experiments conducted by the top five groups (based on MAgP).

RMIT University Using the CO query. This is the exact same run as the article run for the Relevant in Context Task. See the description for the Relevant in Context Task above.

University of Waterloo Using the CO query. See the description for the Focused Task above. Based on the Focused run, duplicated articles were removed in a post-processing step.

Queensland University of Technology Using the CO query. See the description for the Relevant in Context Task above. The best scoring element was selected.

Table 8. Statistical significance (t-test, one-tailed, 95%).

(a) Focused Task										(b) Relevant in Context Task										(c) Best in Context Task													
	1	2	3	4	5	6	7	8	9	10		1	2	3	4	5	6	7	8	9	10		1	2	3	4	5	6	7	8	9	10	
udalian-5	-	-	-	-	-	-	-	-	-	-	udalian-16	-	*	*	*	*	*	*	*	*	*	*	rmit-2	-	-	-	*	*	*	*	*	*	*
maxplanck-3	-	-	-	-	-	-	-	-	-	-	qutau-18	-	-	*	*	*	*	*	*	*	*	*	uwaterloo-3	-	-	-	-	-	*	*	*	*	*
udoshisha-0	-	-	-	-	-	-	-	-	-	-	rmit-1	-	*	*	*	*	*	*	*	*	*	*	qutau-19	-	-	-	-	-	*	*	*	*	*
uamsterdam-2	-	-	-	-	-	-	-	-	-	-	uamsterdam-4	-	-	-	-	-	*	*	*	*	*	*	udalian-7	-	-	-	-	*	*	*	*	*	*
uwaterloo-0	-	-	-	-	-	-	-	-	-	-	unigordon-7	-	-	-	-	-	-	-	-	-	-	*	unigordon-2	-	-	-	*	*	*	*	*	*	*
qutau-20	-	-	-	-	-	-	-	-	-	-	utwente-5	-	-	-	-	-	-	-	-	-	-	*	uamsterdam-16	-	-	*	*	*	*	*	*	*	*
inria-2	-	-	-	-	-	-	-	-	-	-	inria-5	-	-	-	-	-	-	-	-	-	-	*	inria-8	-	-	*	*	*	*	*	*	*	*
rmit-0	-	-	-	-	-	-	-	-	-	-	maxplanck-8	-	-	-	-	-	-	-	-	-	-	*	justsystem-7	-	-	*	*	*	*	*	*	*	*
unigordon-1	-	-	-	-	-	-	-	-	-	-	justsystem-14	-	-	-	-	-	-	-	-	-	-	-	utwente-2	-	-	*	*	*	*	*	*	*	*
mines-2	-	-	-	-	-	-	-	-	-	-	mines-9	-	-	-	-	-	-	-	-	-	-	-	maxplanck-6	-	-	*	*	*	*	*	*	*	*

Dalian University of Technology Using the CO query. See the description for the Focused Task and Relevant in Context above. Return the element which has the largest score per document.

Robert Gordon University Using the CO query. See the description for the Relevant in Context Task above. For the best-in-context task, the element with the highest score for each of the documents is chosen.

Based on the information from these and other participants:

- As for the Relevant in Context Task, we see again that solid article ranking is very important. In fact, the full article run *rmit-2* is the most effective system.
- Using the start of the whole article as a best-entry-point, as done by the top scoring article run, appears to be a reasonable strategy.
- With the exception of *uamsterdam-16*, which used a filter based on all CAS target elements in the topic set, all best runs per group use the CO query.

4.5 Significance Tests

We tested whether higher ranked systems were significantly better than lower ranked system, using a t-test (one-tailed) at 95%. Table 8 shows, for each task, whether it is significantly better (indicated by “*”) than lower ranked runs. For example, For the Focused Task, we see that early precision is a rather unstable measure and none of the runs are significantly different. Hence we should be careful when drawing conclusions based on the Focused Task results. For the Relevant in Context Task, we see that the top run is significantly better than ranks 3 through 10, the second and third ranked systems better than ranks 5 through 10, and the fourth ranked system better than rank 10. For the Best in Context Task, we see that the top run is significantly better than ranks 5 through 10, the second ranked system better than ranks 8 through 10, and the third to eighth ranked system better than those at rank 9 and 10.

Table 9. Ad Hoc Track: Passage runs.

(a) Focused Task					
Participant	iP[0.00]	iP[0.01]	iP[0.05]	iP[0.10]	MAiP
uotago-5	0.3651	0.3617	0.2380	0.1782	0.0649

(b) Relevant in Context Task					
Participant	gP[5]	gP[10]	gP[25]	gP[50]	MAgP
uotago-2	0.1099	0.1000	0.0797	0.0611	0.0653

(c) Best in Context Task					
Participant	gP[5]	gP[10]	gP[25]	gP[50]	MAgP
uotago-8	0.1407	0.1467	0.1247	0.1031	0.1082

5 Analysis of Run and Topic Types

In this section, we will discuss relative effectiveness of element and passage retrieval approaches, and on the relative effectiveness of systems using the keyword and structured queries.

5.1 Elements versus passages

We received some, but few, submissions using passage results. We will look at the relative effectiveness of element and passage runs.

As we saw above, in Section 4, for all three tasks the best scoring runs used elements as the unit of retrieval. All nine official passage submissions were from the same participant. Table 9 shows their best passage runs for the three ad hoc tasks. As it turns out, the passage run *otago-5* would have been the 12th ranked participant (out of 26) for the Focused Task; *otago-2* would have been the 11th ranked group (out of 18) for the Relevant in Context Task; and *otago-8* would have been the 12th ranked group (out of 19) for the Best in Context Task.

This outcome is consistent with earlier results using passage-based element retrieval, where passage retrieval approaches showed comparable but not superior behavior to element retrieval approaches [6, 7].

It is hard to draw any conclusions for several reasons. First, the passage runs took no account of document structure with passages frequently starting and ending mid-sentence. Second, with only a single participant it is not clear whether the approach is comparable or the participant’s runs are only comparable. Third, this is the first year passage retrieval has run at INEX and so the technology is less mature than element retrieval.

We hope and expect that the test collection and the passage runs will be used for further research into the relative effectiveness of element and passage retrieval approaches.

5.2 CO versus CAS

We now zoom in on the relative effectiveness of the keyword (CO) and structured (CAS) queries. As we saw above, in Section 4, the best two runs for the Focused

Table 10. CAS query target elements over all 130 topics.

Target Element	Frequency
*	51
article	29
section	28
figure	9
p	5
image	5
title	1
(section p)	1
body	1

task used the CAS query, and one of the top 10 runs for the Best in Context Task used the CAS query.

All topics have a CAS query since artificial CAS queries of the form

```
/**[about(., keyword title)]
```

were added to topics without CAS title. Table 10 show the distribution of target elements. In total 111 topics had a CAS query formulated by the authors. Some authors already used the generic CAS query above. There are only 86 topics with a non-trivial CAS query.² The CAS topics numbered 415, 416, 418-424, 426-432, 434-440, 442-448, 454, 459, 461, 463, 464, 466, 470, 472, 474, 476-491, 493-498, 500, 501, 507, 508, 511, 515, and 525-543. 72 of these CAS topics were assessed. The results presented here are restricted to only these 72 CAS topics.

Table 11 lists the top 10 participants measured using just the 72 CAS topics and for the Focused Task (a), the Relevant in Context Task (b), and the Best in Context Task (c). For the Focused Task the best two CAS runs outperform the CO runs, as they did over the full topic set. For the Relevant in Context Task, the best CAS run would have ranked fourth among CO runs. For the Best in Context Task, the best two CAS runs would rank sixth and seventh among the CO runs.

We look in detail at the Focused Task runs. Overall, the CAS submissions appear to perform similarly on the subset of 72 CAS topics to the whole set of topics. This was unexpected as these topics do contain real structural hints. The 72 CAS topics constitute three-quarters of the full topic set, making it reasonable to get such a result. However, there are some notable performance characteristics among the CO submissions:

- Some runs (like *udoshisha-0*) perform equally well as over all topics.
- Some runs (like *rmit-0* and *unigordon-1*) perform much better than over all topics. A possible explanation is the larger number of article-targets among the CAS queries.

² Note that some of the wild-card topics (using the “*” target) in Table 10 had non-trivial about-predicates and hence have not been regarded as trivial CAS queries.

Table 11. Ad Hoc Track CAS Topics: CO versus CAS.

(a) Focused Task											
Participant	iP[0.00]	iP[0.01]	iP[0.05]	iP[0.10]	MAiP	Participant	iP[0.00]	iP[0.01]	iP[0.05]	iP[0.10]	MAiP
udoshisha-0	0.4354	0.4087	0.3265	0.2731	0.1273	udalian-5	0.4289	0.4247	0.3304	0.3050	0.1446
rmit-0	0.3941	0.3923	0.3496	0.3218	0.1868	maxplanck-3	0.4805	0.4141	0.3118	0.2837	0.1201
unigordon-1	0.4081	0.3916	0.3231	0.3103	0.1603	udoshisha-1	0.4463	0.3819	0.2858	0.2505	0.1066
inria-2	0.3933	0.3916	0.3508	0.3244	0.1860	justsystem-3	0.3802	0.3558	0.2444	0.2150	0.0826
qutau-17	0.4289	0.3869	0.3193	0.2670	0.1049	uamsterdam-10	0.3976	0.3554	0.2923	0.2645	0.1266
uwaterloo-0	0.4085	0.3835	0.3326	0.2983	0.1444	unitoronto-0	0.3793	0.3051	0.2343	0.2117	0.0820
cmu-0	0.4757	0.3819	0.2791	0.2506	0.0999	qutau-9	0.2926	0.2886	0.2823	0.2597	0.1342
udalian-0	0.3969	0.3816	0.3209	0.3000	0.1415	unigranada-3	0.3600	0.2264	0.0836	0.0524	0.0182
uamsterdam-2	0.4487	0.3757	0.2928	0.2704	0.1298						
mines-2	0.4572	0.3699	0.2362	0.1952	0.0827						

(b) Relevant in Context Task											
Participant	gP[5]	gP[10]	gP[25]	gP[50]	MAgP	Participant	gP[5]	gP[10]	gP[25]	gP[50]	MAgP
qutau-18	0.2000	0.1581	0.1149	0.0884	0.1081	udalian-8	0.1704	0.1445	0.1169	0.0891	0.0987
udalian-4	0.1775	0.1553	0.1138	0.0905	0.1039	uamsterdam-13	0.1638	0.1419	0.1008	0.0761	0.0844
rmit-1	0.1650	0.1554	0.1126	0.0834	0.0951	qutau-10	0.1538	0.1257	0.0997	0.0765	0.0792
unigordon-7	0.1748	0.1478	0.1059	0.0761	0.0870	maxplanck-5	0.1702	0.1410	0.1080	0.0731	0.0762
uamsterdam-4	0.1717	0.1440	0.1036	0.0777	0.0870	justsystem-15	0.1162	0.1040	0.0775	0.0619	0.0726
inria-5	0.1650	0.1554	0.1192	0.0861	0.0829						
utwente-5	0.1347	0.1142	0.0916	0.0686	0.0817						
maxplanck-8	0.1534	0.1240	0.0843	0.0670	0.0801						
justsystem-14	0.1230	0.1061	0.0823	0.0600	0.0799						
uotago-0	0.1097	0.0992	0.0762	0.0567	0.0692						

(c) Best in Context Task											
Participant	gP[5]	gP[10]	gP[25]	gP[50]	MAgP	Participant	gP[5]	gP[10]	gP[25]	gP[50]	MAgP
rmit-2	0.3634	0.3345	0.2484	0.1888	0.2057	udalian-17	0.2568	0.2501	0.2136	0.1730	0.1847
uwaterloo-3	0.2925	0.2701	0.2263	0.1746	0.1994	uamsterdam-16	0.3218	0.2896	0.2221	0.1735	0.1772
qutau-0	0.3369	0.2828	0.2314	0.1830	0.1963	justsystem-9	0.3003	0.2695	0.2037	0.1682	0.1611
udalian-7	0.2600	0.2547	0.2112	0.1670	0.1916	qutau-3	0.2735	0.2309	0.1626	0.1232	0.1460
unigordon-2	0.3708	0.3019	0.2269	0.1723	0.1881	maxplanck-1	0.2724	0.2458	0.1967	0.1388	0.1273
uamsterdam-7	0.2771	0.2675	0.2117	0.1664	0.1762	unigranada-6	0.1871	0.1793	0.1519	0.1231	0.1084
inria-8	0.3634	0.3345	0.2560	0.1910	0.1757	irit-4	0.0310	0.0326	0.0322	0.0224	0.0168
justsystem-7	0.3083	0.2900	0.2143	0.1649	0.1755						
maxplanck-6	0.2134	0.2177	0.1761	0.1383	0.1418						
utwente-2	0.2526	0.2144	0.1596	0.1224	0.1369						

- Some runs (like *udalian-0* and *uamsterdam-2*) perform less well than over all topics.

We should be careful to draw conclusions based on these observations, since the early precision differences between the runs tend not to be significant.

Finally, for the Relevant in Context Task over the CAS topics, the passage run *uotago-0* is ranked at the tenth best CO submission, even though it ignored both the structural hints in the topics and in the documents!

6 Discussion and Conclusions

In this paper we provided an overview of the INEX 2007 Ad Hoc Track that contained three tasks: For the *Focused Task* a ranked-list of non-overlapping results (elements or passages) was required. For the *Relevant in Context Task* non-overlapping results (elements or passages) grouped by the article that they belong to were required. For the *Best in Context Task* a single starting point (element’s starting tag or passage offset) per article was required. We discussed the results for the three tasks, and analysed the relative effectiveness of element and passage runs, and of keyword (CO) queries and structured queries (CAS).

When examining the relative effectiveness of CO and CAS we found that the best Focused Task submissions use the CAS query, showing that structural hints can help promote initial precision. This provides further evidence that structured queries can be a useful early precision enhancing device [9]. Although, when restricting to non-trivial CAS queries, we see no real gain for the CAS submissions relative to the CO submissions.

An unexpected finding is that article retrieval is a reasonably effective at XML-IR: an article-only run scored the eighth best group for the Focused Task; the third best for the Relevant in Context Task; and the top ranking group for the Best in Context Task. This demonstrates the importance of the article ranking in the “in context” tasks. The chosen measures were also not unfavorable towards article-submissions:

- For the Relevant in Context Task, the F-score per document equally rewards precision and recall. Article runs have excellent recall, and in the case of Wikipedia, where articles tend to be focused on a single topic, acceptable precision.
- For the Best in Context Task, the window receiving scores was 1,000 characters which, although more strict than the measures at INEX 2006, remains too lenient.

Given the efforts put into the fair comparison of element and passage retrieval approaches, the number of passage submissions was disappointing. The passage runs that were submitted ignored document structure—perhaps the identification based on the XML structure turned out to be difficult, or perhaps the technology is just not yet mature. Although we received only passage results from a single participant, and should be careful to avoid hasty conclusions, we

saw that the passage based approach was better than average, but not superior to element based approaches. This outcome is consistent with earlier results using passage-based element retrieval [6, 7]. The comparative analysis of element and passage retrieval approaches was the aim of the track, hoping to shed light on the value of the document structure as provided by the XML mark-up. Although few official submissions used passage retrieval approaches, we hope and expect that the resulting test collection will prove its value in future use. After all, the main aim of the INEX initiative is to create bench-mark test-collections for the evaluation of structured retrieval approaches.

Acknowledgments

Eternal thanks to Benjamin Piwowarski for completely updating the X-RAI tools to ensure that all passage offsets can be mapped exactly.

Jaap Kamps was supported by the Netherlands Organization for Scientific Research (NWO, grants # 612.066.513, 639.072.601, and 640.001.501), and by the E.U.'s 6th FP for RTD (project MultiMATCH contract IST-033104).

Bibliography

- [1] C. L. A. Clarke. Range results in XML retrieval. In *Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology*, pages 4–5, Glasgow, UK, 2005.
- [2] C. L. A. Clarke, J. Kamps, and M. Lalmas. INEX 2007 retrieval task and result submission specification. In *This Volume*, 2007.
- [3] L. Denoyer and P. Gallinari. The Wikipedia XML Corpus. *SIGIR Forum*, 40:64–69, 2006.
- [4] S. Geva. GPX – gardens point XML IR at INEX 2005. In *Advances in XML Information Retrieval and Evaluation: INEX 2005*, volume 3977 of *LNCS*, pages 204–253, 2006.
- [5] T. Grabs and H.-J. Schek. ETH Zürich at INEX: Flexible information retrieval from XML with PowerDB-XML. In *Proceedings of the First Workshop of the INitiative for the Evaluation of XML retrieval (INEX)*, pages 141–148. ERCIM Publications, 2003.
- [6] W. Huang, A. Trotman, and R. A. O’Keefe. Element retrieval using a passage retrieval approach. In *Proceedings of the 11th Australasian Document Computing Symposium (ADCS 2006)*, pages 80–83, 2006.
- [7] K. Y. Itakura and C. L. A. Clarke. From passages into elements in XML retrieval. In *Proceedings of the SIGIR 2007 Workshop on Focused Retrieval*, pages 17–22. University of Otago, Dunedin New Zealand, 2007.
- [8] J. Kamps and M. Koolen. On the relation between relevant passages and XML document structure. In *Proceedings of the SIGIR 2007 Workshop on Focused Retrieval*, pages 28–32. University of Otago, Dunedin New Zealand, 2007.

- [9] J. Kamps, M. Marx, M. de Rijke, and B. Sigurbjörnsson. Articulating information needs in XML query languages. *Transactions on Information Systems*, 24:407–436, 2006.
- [10] J. Kamps, J. Pehcevski, G. Kazai, M. Lalmas, and S. Robertson. INEX 2007 evaluation measures. In *This Volume*, 2007.
- [11] J. Kekäläinen and K. Järvelin. Using graded relevance assessments in IR evaluation. *Journal of the American Society for Information Science and Technology*, 53:1120–1129, 2002.
- [12] M. Lalmas and B. Piwowarski. INEX 2007 relevance assessment guide. In *This Volume*, 2007.
- [13] A. Trotman and S. Geva. Passage retrieval and other XML-retrieval tasks. In *Proceedings of the SIGIR 2006 Workshop on XML Element Retrieval Methodology*, pages 43–50. University of Otago, Dunedin New Zealand, 2006.
- [14] A. Trotman and B. Larsen. INEX 2007 guidelines for topic development. In *This Volume*, 2007.
- [15] Wumpus. The Wumpus search engine, 2007. <http://www.wumpus-search.org>.
- [16] Zettair. The Zettair search engine, 2007. <http://www.seg.rmit.edu.au/zettair/>.

Appendix: Full run names

Run	Label
cmu-0	p40_nophrasebase
inria-2	p11.ent-ZM-Focused
inria-5	p11.ent-ZM-RiC
inria-8	p11.ent-ZM-BiC
irit-4	p49_xfirm.cos.01_BIC
justsystem-14	p41_VSM_CO_09
justsystem-15	p41_VSM_CAS_10
justsystem-3	p41_VSM_CAS_04
justsystem-7	p41_VSM_CO_14
justsystem-9	p41_VSM_CAS_16
maxplanck-1	p25_TOPX-CAS-exp-BIC
maxplanck-3	p25_TOPX-CAS-Focused-all
maxplanck-5	p25_TOPX-CAS-RIC
maxplanck-6	p25_TOPX-CO-all-BIC
maxplanck-8	p25_TOPX-CO-all-exp-RIC
mines-2	p53_EMSE.boolean.Prox200NF.0012
mines-9	p53_EMSE.boolean.Prox200NRs.0011
qutau-0	p9_BIC_00
qutau-10	p9_RIC_05
qutau-17	p9_FOC_06
qutau-18	p9_RIC_07
qutau-19	p9_BIC_07
qutau-20	p9_FOC_07
qutau-3	p9_BIC_04
qutau-9	p9_FOC_04
rmit-0	p32_zet-okapi-Focused
rmit-1	p32_zet-okapi-RiC
rmit-2	p32_zet-okapi-BiC
uamsterdam-10	p36_inex07_contain_beta1_focused_clp_10000_cl_cas_pool_filter
uamsterdam-13	p36_inex07_contain_beta1_focused_clp_10000_cl_cas_pool_filter_ric_hse
uamsterdam-16	p36_inex07_contain_beta1_focused_clp_10000_cl_cas_pool_filter_bic_hse
uamsterdam-2	p36_inex07_element_beta1_focused_clp_10000_cl
uamsterdam-4	p36_inex07_contain_beta1_focused_clp_10000_cl_ric_hse
uamsterdam-7	p36_inex07_contain_beta1_focused_clp_10000_cl_bic_hse
udalian-0	p26_DUT_06_Focused
udalian-16	p26_DUT_01_Relevant
udalian-17	p26_DUT_03_Best
udalian-4	p26_DUT_02_Relevant
udalian-5	p26_DUT_04_Focused
udalian-7	p26_DUT_02_Best
udalian-8	p26_DUT_05_Relevant
udoshisha-0	p22_Kikori-CO-Focused
udoshisha-1	p22_Kikori-CAS-Focused
unigordon-1	p35_Focused-LM
unigordon-2	p35_BestInContext-LM
unigordon-7	p35_RelevantInContext-LM
unigranada-3	p4_CID_pesos_15
unigranada-6	p4_CID_pesos_15_bic
unitoronto-0	p60_4-sr
uotago-0	p10_DocsNostem-PassagesStem-StdDevNo
uotago-2	p10_DocsNostem-PassagesStem-StdDevYes
uotago-5	p10_DocsNostem-PassagesStem-StdDevYes-Focused
uotago-8	p10_DocsNostem-PassagesStem-StdDevYes-BEP
utwente-2	p45_articleBic
utwente-5	p45_star_logLP_RinC
uwaterloo-0	p37_FOER
uwaterloo-3	p37_BICERGood

INEX 2007 Evaluation Measures

Jaap Kamps¹, Jovan Pehcevski², Gabriella Kazai³, Mounia Lalmas⁴, and
Stephen Robertson³

¹ University of Amsterdam, The Netherlands
kamps@science.uva.nl

² INRIA Rocquencourt, France
jovan.pehcevski@inria.fr

³ Microsoft Research Cambridge, United Kingdom
{gabkaz, ser}@microsoft.com

⁴ Queen Mary, University of London, United Kingdom
mounia@dcs.qmul.ac.uk

Abstract. This paper describes the official measures of retrieval effectiveness that are employed for the Ad Hoc Track at INEX 2007. Whereas in earlier years all, but only, XML elements could be retrieved, the result format has been liberalized to arbitrary passages. In response, the INEX 2007 measures are based on the amount of highlighted text retrieved, leading to natural extensions of the well-established measures of precision and recall. The following measures are defined: The Focused Task is evaluated by interpolated precision at 1% recall (iP[0.01]) in terms of the highlighted text retrieved. The Relevant in Context Task is evaluated by mean average generalized precision (*MAgP*) where the generalized score per article is based on the retrieved highlighted text. The Best in Context Task is also evaluated by mean average generalized precision (*MAgP*) but here the generalized score per article is based on the distance to the assessor’s best-entry point.

1 Introduction

Focused retrieval investigates ways to provide users with direct access to relevant information in retrieved documents, and includes tasks like question answering, passage retrieval, and XML element retrieval [17]. Since its launch in 2002, INEX has studied different aspects of focused retrieval by mainly considering XML element retrieval techniques that can effectively retrieve information from structured document collections [6]. The main change in the Ad Hoc Track at INEX 2007 is allowing retrieval of arbitrary document parts, which can represent XML elements or passages [3]. That is, a retrieval result can be either an XML element (a sequence of textual content contained within start/end tags), or an arbitrary passage (a sequence of textual content that can be either contained within an element, or can span across a range of elements). In this paper, we will use the term “document part” to refer to both XML elements and arbitrary passages. These changes address requests to liberalize the retrieval format to ranges of elements [2] and to arbitrary passages [15]. However, this simple change has

deer consequence for the measures as used up to now at INEX [5, 8, 9, 12, 13]. By allowing arbitrary passages, we loose the “natural” retrieval unit of elements that was the basis for earlier measures. At INEX 2007 we have adopted an evaluation framework that is based on the amount of highlighted text in relevant documents (similar to the HiXEval measures [14]). In this way we build directly on highlighting assessment procedure used at INEX, and define measures that are natural extensions of the well-established measures of precision and recall used in traditional information retrieval [1].

This paper is organised as follows. In Section 2, we briefly describe the ad hoc retrieval tasks at INEX 2007, and the resulting relevance assessments. Then in three separate sections, we discuss the evaluation measures used for each of the INEX 2007 tasks: the Focused Task (Section 3); the Relevant in Context Task (Section 4); and the Best in Context Task (Section 5).

2 Ad Hoc Retrieval Track

In this section, we briefly summarize the ad hoc retrieval tasks, and the resulting relevance judgments.

2.1 Ad hoc retrieval tasks

The INEX 2007 Ad Hoc Track investigates the following three retrieval tasks as defined in [3]. First, there is the Focused Task.

Focused Task This task asks systems to return a ranked list of non-overlapping, most focused document parts that represent the most appropriate units of retrieval. For example, in the case of returning XML elements, a paragraph and its container section should not both be returned. For this task, from all the estimated relevant (and possibly overlapping) document parts, systems are required to choose those non-overlapping document parts that represent the most appropriate units of retrieval.

The second task corresponds to an end-user task where focused retrieval answers are grouped per document, in their original document order, providing access through further navigational means. This assumes that users consider documents as the most natural units of retrieval, and prefer an overview of relevance in their original context.

Relevant in Context This task asks systems to return non-overlapping relevant document parts clustered by the unit of the document that they are contained within. An alternative way to phrase the task is to return documents with the most focused, relevant parts highlighted within.

The third task is similar to Relevant in Context, but asks for only a single best point to start reading the relevant content in an article.

Best in Context This task asks systems to return a single document part per document. The start of the single document part corresponds to the best entry point for starting to read the relevant text in the document.

Given that passages can be overlapping in sheer endless ways, there is no meaningful equivalent of the *Thorough Task* as defined in earlier years of INEX.

Note that there is no separate passage retrieval task, and for all the three tasks arbitrary passages may be returned instead of elements. For all the three tasks, systems could either use the title field of the topics (content-only topics) or the castitle field of the topics (content-and-structure topics). Trotman and Larsen [16] provide a detailed description of the format used for the INEX 2007 topics.

2.2 Relevance Assessments

Since 2005, a highlighting assessment procedure is used at INEX to gather relevance assessments for the INEX retrieval topics [11]. In this procedure, assessors from the participating groups are asked to highlight sentences representing the relevant information in a pooled set of documents of the Wikipedia XML document collection [4]. After assessing an article with relevance, a separate best entry point judgment is also collected from the assessor, marking the point in the article that represents the best place to start reading.

The Focused and Relevant in Context Tasks will be evaluated against the text highlighted by the assessors, whereas the Best in Context Task will be evaluated against the best-entry-points.

3 Evaluation of the Focused Task

3.1 Assumptions

In the Focused Task, for each INEX 2007 topic, systems are asked to return a ranked list of the top 1,500 non-overlapping most focused relevant document parts. The retrieval systems are thus required not only to rank the document parts according to their estimated relevance, but to also decide which document parts are the most focused non-overlapping units of retrieval.

We make the following evaluation assumption about the Focused Task: *The amount of relevant information retrieved is measured in terms of the length of relevant text retrieved.* That is, instead of counting the number of relevant documents retrieved, in this case we measure the amount of relevant (highlighted) text retrieved.

3.2 Evaluation measures

More formally, let p_r be the document part assigned to rank r in the ranked list of document parts L_q returned by a retrieval system for a topic q (at INEX 2007, $|L_q| = 1,500$ elements or passages). Let $rsize(p_r)$ be the length of highlighted

(relevant) text contained by p_r in characters (if there is no highlighted text, $rsize(p_r) = 0$). Let $size(p_r)$ be the total number of characters contained by p_r , and let $Trel(q)$ be the total amount of (highlighted) relevant text for topic q . $Trel(q)$ is calculated as the total number of highlighted characters across all documents, i.e., the sum of the lengths of the (non-overlapping) highlighted passages from all relevant documents.

Measures at selected cutoffs Precision at rank r is defined as the fraction of retrieved text that is relevant:

$$P[r] = \frac{\sum_{i=1}^r rsize(p_i)}{\sum_{i=1}^r size(p_i)} \quad (1)$$

To achieve a high precision score at rank r , the document parts retrieved up to and including that rank need to contain as little non-relevant text as possible.

Recall at rank r is defined as the fraction of relevant text that is retrieved:

$$R[r] = \frac{\sum_{i=1}^r rsize(p_i)}{Trel(q)} \quad (2)$$

To achieve a high recall score at rank r , the document parts retrieved up to and including that rank need to contain as much relevant text as possible.

An issue with the precision measure $P[r]$ given in Equation 1 is that it can be biased towards systems that return several shorter document parts rather than returning one longer part that contains them all (this issue has plagued earlier passage retrieval tasks at TREC [19]). Since the notion of ranks is relatively fluid for passages, we opt to look at precision at recall levels rather than at ranks. Specifically, we use an interpolated precision measure $iP[x]$, which calculates interpolated precision scores at selected recall levels:

$$iP[x] = \begin{cases} \max_{1 \leq r \leq |L_q|} (P[r] \wedge R[r] \geq x) & \text{if } x \leq R[|L_q|] \\ 0 & \text{if } x > R[|L_q|] \end{cases} \quad (3)$$

where $R[|L_q|]$ is the recall over all documents retrieved. For example, $iP[0.01]$ calculates interpolated precision at the 1% recall level for a given topic.

Over a set of topics, we can also calculate the interpolated precision measure, also denoted by $iP[x]$, by calculating the mean of the scores obtained by the measure for each individual topic.

Overall performance measure In addition to using the interpolated precision measure at selected recall levels, we also calculate overall performance scores

based on the measure of average interpolated precision AiP . For an INEX topic, we calculate AiP by averaging the interpolated precision scores calculated at 101 standard recall levels (0.00, 0.01, ..., 1.00):

$$AiP = \frac{1}{101} \cdot \sum_{x=0.00,0.01,\dots,1.00} iP[x] \quad (4)$$

Performance across a set of topics is measured by calculating the mean of the AiP values obtained by the measure for each individual topic, resulting in mean average interpolate precision ($MAiP$). Assuming there are n topics:

$$MAiP = \frac{1}{n} \cdot \sum_t AiP(t) \quad (5)$$

3.3 Results reported at INEX 2007

For the Focused Task we report the following measures over all INEX 2007 topics:

- Mean interpolated precision at four selected recall levels: $iP[x]$, $x \in [0.00, 0.01, 0.05, 0.10]$; and
- Mean interpolated average precision over 101 recall levels ($MAiP$).

The official evaluation for the Focused Task is an early precision measure: interpolated precision at 1% recall ($iP[0.01]$).

4 Evaluation of the Relevant in Context Task

4.1 Assumptions

The Relevant in Context Task is a variation on document retrieval, in which systems are first required to rank documents in a decreasing order of relevance and then identify a set of non-overlapping, relevant document parts. We make the following evaluation assumption: *All documents that contain relevant text are regarded as (Boolean) relevant documents.* Hence, at the article level, we do not distinguish between relevant documents.

4.2 Evaluation measures

The evaluation of the Relevant in Context Task is based on the measures of generalized precision and recall [10], where the per document score reflects how well the retrieved text matches the relevant text in the document. The resulting measure was introduced at INEX 2006 [7, 12].

Score per document For a retrieved document, the text identified by the selected set of non-overlapping retrieved parts is compared to the text highlighted by the assessor. More formally, let d be a retrieved document, and let p be a document part in d . We denote the set of all retrieved parts of document d as \mathcal{P}_d . Let $Trel(d)$ be the total amount of highlighted relevant text in the document d . $Trel(d)$ is calculated as the total number of highlighted characters in a document, i.e., the sum of the lengths of the (non-overlapping) highlighted passages.

We calculate the following for a retrieved document d :

- Document precision, as the fraction of retrieved text (in characters) that is highlighted (relevant):

$$P(d) = \frac{\sum_{p \in \mathcal{P}_d} rsize(p)}{\sum_{p \in \mathcal{P}_d} size(p)} \quad (6)$$

The $P(d)$ measure ensures that, to achieve a high precision value for the document d , the set of retrieved parts for that document needs to contain as little non-relevant text as possible.

- Document recall, as the fraction of highlighted text (in characters) that is retrieved:

$$R(d) = \frac{\sum_{p \in \mathcal{P}_d} rsize(p)}{Trel(d)} \quad (7)$$

The $R(d)$ measure ensures that, to achieve a high recall value for the document d , the set of retrieved parts for that document needs to contain as much relevant text as possible.

- Document F-Score, as the combination of the document precision and recall scores using their harmonic mean [18], resulting in a score in $[0,1]$ per document:

$$F(d) = \frac{2 \cdot P(d) \cdot R(d)}{P(d) + R(d)} \quad (8)$$

For retrieved non-relevant documents, both document precision and document recall evaluate to zero.

We may choose either precision, recall, the F-score, or even other aggregates as document score ($S(d)$). For the Relevant in Context Task, we use the F-score as the document score:

$$S(d) = F(d) \quad (9)$$

The resulting $S(d)$ score varies between 0 (document without relevant text, or none of the relevant text is retrieved) and 1 (all relevant text is retrieved without retrieving any non-relevant text).

Scores for ranked list of documents Given that the individual document scores ($S(d)$) for each document in a ranked list \mathcal{L} can take any value in $[0,1]$, we employ the evaluation measures of generalized precision and recall [10].

More formally, let us assume that for a given topic there are in total $Nrel$ relevant documents, and let $IsRel(d_r) = 1$ if document d at document-rank r contains highlighted relevant text, and $IsRel(d_r) = 0$ otherwise. Let $Nrel$ be the total number of document with relevance for a given topics.

Over the ranked list of documents, we calculate the following:

- generalized precision ($gP[r]$), as the sum of document scores up to (and including) document-rank r , divided by the rank r :

$$gP[r] = \frac{\sum_{i=1}^r S(d_i)}{r} \quad (10)$$

- generalized Recall ($gR[r]$), as the number of relevant documents retrieved up to (and including) document-rank r , divided by the total number of relevant documents:

$$gR[r] = \frac{\sum_{i=1}^r IsRel(d_i)}{Nrel} \quad (11)$$

Based on these, the average generalized precision AgP for a topic can be calculated by averaging the generalized precision scores obtained for each natural recall points, where generalized recall increases:

$$AgP = \frac{\sum_{r=1}^{|\mathcal{L}|} IsRel(d_r) \cdot gP[r]}{Nrel} \quad (12)$$

For non-retrieved relevant documents a generalized precision score of zero is assumed.

The mean average generalized precision ($MAgP$) is simply the mean of the average generalized precision scores over all topic.

4.3 Results reported at INEX 2007

For the Relevant in Context Task we report the following measures over all topics:

- Non-interpolated mean generalized precision at four selected ranks: $gP[r]$, $r \in [5, 10, 25, 50]$; and
- Non-interpolated mean average generalized precision ($MAgP$).

The official evaluation for the Relevant in Context Task is the overall mean average generalized precision ($MAgP$) measure, where the generalized score per article is based on the retrieved highlighted text.

5 Evaluation of the Best in Context Task

5.1 Assumptions

The Best in Context Task is another variation on document retrieval where, for each document, a single best entry point needs to be identified. We again assume that all documents with relevance are equally desirable.

5.2 Evaluation measures

The evaluation of the Best in Context Task is also based on the measures of generalized precision and recall [10], where the per document score reflects how well the retrieved entry point matches the best entry point in the document. Note that at INEX 2006 a different, and more liberal, distance measure was used [12].

Score per document The document score $S(d)$ for this task is calculated with a distance similarity measure, $s(x, b)$, which measures how close the system-proposed entry point x is to the ground-truth best entry point b given by the assessor. Closeness is assumed to be an inverse function of distance between the two points. The maximum value of 1 is achieved when the two points match, and the minimum value is zero.

We use the following formula for calculating the distance similarity measure:

$$s(x, b) = \begin{cases} \frac{n-d(x,b)}{n} & \text{if } 0 \leq d(x, b) \leq n \\ 0 & \text{if } d(x, b) > n \end{cases} \quad (13)$$

where the distance $d(x, b)$ is measured in characters, and n is the number of characters representing the visible part of the document that can fit on a screen (typically, $n = 1000$ characters).

We use the $s(x, b)$ distance similarity score as the document score for the Best in Context Task:

$$S(d) = s(x, b) \quad (14)$$

The resulting $S(d)$ score varies between 0 (non-relevant document, or the distance between the system-proposed entry point and the ground-truth best entry point is more than n characters) and 1 (the system-proposed entry point is identical to the ground-truth best entry point).

Scores for ranked list of documents Completely analogous to the Relevant in Context Task, we use generalized precision and recall to determine the score for the ranked list of documents. For details, see the above discussion of the Relevant in Context Task in Section 4.

5.3 Results reported at INEX 2007

For the Best in Context Task we report the following measures over all topics:

- Non-interpolated mean generalized precision at four selected ranks: $gP[r]$, $r \in [5, 10, 25, 50]$; and
- Non-interpolated mean average generalized precision ($MAgP$).

The official evaluation for the Best in Context Task is the overall mean average generalized precision ($MAgP$) measure with the generalized score per article is based on the distance to the best-entry point.

Acknowledgements

We thank Benjamin Piwowarski and James A. Thom for their valuable comments on earlier drafts of this paper.

Jaap Kamps was supported by the Netherlands Organization for Scientific Research (NWO, grants # 612.066.513, 639.072.601, and 640.001.501), and by the E.U.'s 6th FP for RTD (project MultiMATCH contract IST-033104).

Bibliography

- [1] R. Baeza-Yates and B. Ribeiro-Neto, editors. *Modern Information Retrieval*, 1999. ACM Press, New York and Addison Wesley Longman, Harlow.
- [2] C. L. A. Clarke. Range results in XML retrieval. In *Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology*, pages 4–5, Glasgow, UK, 2005.
- [3] C. L. A. Clarke, J. Kamps, and M. Lalmas. INEX 2007 retrieval task and result submission specification. In *This Volume*, 2007.
- [4] L. Denoyer and P. Gallinari. The Wikipedia XML corpus. *SIGIR Forum*, 40(1):64–69, 2006.
- [5] N. Gövert and G. Kazai. Overview of the INitiative for the Evaluation of XML retrieval (INEX) 2002. In *Proceedings of the First Workshop of the INitiative for the Evaluation of XML retrieval (INEX)*, pages 1–17. ERCIM Publications, 2003.
- [6] INEX. INitiative for the Evaluation of XML Retrieval, 2007. <http://inex.is.informatik.uni-duisburg.de/>.
- [7] J. Kamps, M. Lalmas, and J. Pehcevski. Evaluating Relevant in Context: Document retrieval with a twist. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 723–724. ACM Press, New York NY, USA, 2007.
- [8] G. Kazai. Report of the INEX 2003 metrics work group. In *INEX 2003 Workshop Proceedings*, pages 184–190, 2004.
- [9] G. Kazai and M. Lalmas. INEX 2005 evaluation measures. In *Advances in XML Information Retrieval and Evaluation: Fourth Workshop of the INitiative for the Evaluation of XML Retrieval (INEX 2005)*, volume 3977 of *Lecture Notes in Computer Science*, pages 16–29. Springer Verlag, Heidelberg, 2006.
- [10] J. Kekäläinen and K. Järvelin. Using graded relevance assessments in IR evaluation. *Journal of the American Society for Information Science and Technology*, 53(13):1120–1129, 2002.
- [11] M. Lalmas and B. Piwowarski. INEX 2007 relevance assessment guide. In *This Volume*, 2007.
- [12] M. Lalmas, G. Kazai, J. Kamps, J. Pehcevski, B. Piwowarski, and S. Robertson. INEX 2006 evaluation measures. In N. Fuhr, M. Lalmas, and A. Trotman, editors, *Comparative Evaluation of XML Information Retrieval Systems: Fifth Workshop of the INitiative for the Evaluation of XML Retrieval*

- (*INEX 2006*), volume 4518 of *Lecture Notes in Computer Science*, pages 20–34. Springer Verlag, Heidelberg, 2007.
- [13] S. Malik, M. Lalmas, and N. Fuhr. Overview of INEX 2004. In *Advances in XML Information Retrieval. Third Workshop of the INitiative for the Evaluation of XML Retrieval, INEX 2004*, volume 3493 of *Lecture Notes in Computer Science*, pages 1–15. Springer Verlag, Heidelberg, 2005.
 - [14] J. Pehcevski and J. A. Thom. HiXEval: Highlighting XML retrieval evaluation. In *Advances in XML Information Retrieval and Evaluation: Fourth Workshop of the INitiative for the Evaluation of XML Retrieval, INEX 2005*, volume 3977 of *Lecture Notes in Computer Science*, pages 43–57, 2006.
 - [15] A. Trotman and S. Geva. Passage retrieval and other XML-retrieval tasks. In *Proceedings of the SIGIR 2006 Workshop on XML Element Retrieval Methodology*, pages 43–50, Seattle, USA, 2006.
 - [16] A. Trotman and B. Larsen. INEX 2007 guidelines for topic development. In *This Volume*, 2007.
 - [17] A. Trotman, S. Geva, and J. Kamps, editors. *Proceedings of the SIGIR 2007 Workshop on Focused Retrieval*, 2007. University of Otago, Dunedin New Zealand.
 - [18] C. J. van Rijsbergen. *Information Retrieval*. London: Butterworths, 1979.
 - [19] C. Wade and J. Allan. Passage retrieval and evaluation. Technical report, CIIR, University of Massachusetts, Amherst, 2005.

The Role of Shallow Features in XML Retrieval

Fang Huang

School of Computing, The Robert Gordon University, Scotland
f.huang@rgu.ac.uk

Abstract. This paper describes the retrieval approach based on language models used by Robert Gordon University in the INEX 2007 ad hoc track. We focused on the question of how shallow features of text display information in an XML document can be used to enhance retrieval effectiveness. We employed a mixture language model combining estimates based on element full-text and the compact representation of the element. We also used non-content priors, including the location the element appears in the original document, and the length of the element path, to boost retrieval effectiveness.

1 Introduction

In this paper, we describe our experiments of using language models in the INEX 2007 ad hoc track. With the rapidly widespread use of the eXtensible Markup Language (XML) on the internet, XML information retrieval (XML-IR) has been receiving growing research interest. A variety of approaches have been exploited to score XML elements' relevance to a user's query. Geva [1] described an approach based on the construction of a collection sub-tree that consists of all elements containing one or more of the query terms. Leaf nodes are assigned a score using a *tf.idf* variant, and scores are propagated upwards in the document XML tree, so that all ancestor elements are ranked. Ogilvie and Callan [5] proposed using hierarchical language models for ranking XML elements. An element's relevance is determined by weighted combining of several language models estimated, respectively, from the text of the element, its parent, its children, and the document. In our participation of INEX 2006, we [2] investigated which parts of a document or an XML element are more likely to attract a reader's attention, and proposed using these "attractive" parts to build a compact form of a document (or an XML element). We then used a mixture language model combining estimates based on element full-text, the compact form of it, as well as a range of non-content priors. The retrieval model presented in this paper is mainly based on our previous approach [2], but we made a few modifications to improve retrieval effectiveness.

The remainder of this paper is organized as follows: Section 2 describes the mixture language model we used. Our INEX experiments and submitted runs are presented in section 3. Section 4 discusses our results in the INEX 2007 official evaluation. The final part, section 5, concludes with a discussion and possible directions for future work.

2 the Retrieval Model

While current work in XML information retrieval focuses on exploiting the hierarchical structure of XML elements to implement more focused retrieval strategies, we believe that text display information together with some shallow features (e.g., an XML element’s location in the original document) could be used to enhance retrieval effectiveness. This is based on the fact that when a human assessor reads an article, he (or she) usually can judge its relevance by skimming over certain parts of the documents. Intuitively, the titles, section titles, figures, tables, words underlined, and words emphasized in bold, italics or larger fonts are likely to be the most representative parts. In [2], we proposed to extract and put together all those most representative words to build a compact form of a document (or an XML element), and employed retrieval models that emphasized the importance of the compact form in identifying the relevance of an XML element. However, our results in the INEX 2006 evaluation showed that it did not achieve good performances as we expected. One reason might be that a compact form built like that contained some noise, as in the large, heterogeneous collection we used, not all the features we used are related to texts’ importances. Based on this consideration, in this work, the compact form was generated by words only from titles, section titles, and figure captions. For the remainder of the paper, when we refer to the compact form of an XML element, we mean a collection of words extracted from the titles, section titles, and figure captions nested within that element.

The retrieval model we used is based on the language model, i.e., an element’s relevance to a query is estimated by

$$P(e|q) \propto P(e) \cdot P(q|e) \quad (1)$$

where e is an XML element; q is a query consisting of the terms t_1, \dots, t_k ; the prior, $P(e)$, defines the probability of element e being relevant in absence of a query; $P(q|e)$ is the probability of the query q , given element e .

2.1 Element priors

The Prior $P(e)$ defines the probability that the user selects an element e without a query. Elements are not equally important even though their contents are ignored. Several previous studies[3, 7] reported that a successful element retrieval approach should be biased towards retrieving large elements. In INEX 2006, we conducted a preliminary experiment to investigate potential non-content features that might be used to boost retrieval effectiveness, and concluded that relevant elements tend to appear in the beginning parts of the text, and they are not likely to be nested in depth[2].

Based on these considerations, we calculate the prior of an element according to its location in the original document, and the length of its path.

$$P(e) = \frac{1}{5 + |e_{location}|} \cdot \frac{1}{3 + |e_{path}|} \quad (2)$$

where, $e_{location}$ is the location value of element e ; and e_{path} is the path length of e . Location was defined as the local order of an element ignoring its path. The path length of an element e equals to the number of elements in the path including e itself and those elements nesting e . For example, for an element `/article[1]/body[1]/p[1]` (the first paragraph in the document), the location value is 1 (the first paragraph), and the path length is 3.

2.2 Probability of the query

Assuming query terms to be independent, $P(q|e)$ can be calculated according to a mixture language model:

$$P(q|e) = \prod_{i=1}^k (\lambda \cdot P(t_i|C) + (1 - \lambda) \cdot P(t_i|e)) \quad (3)$$

where λ is the so-called smoothing parameter; C represents the whole collection. $P(t_i|C)$ is the estimate based on the collection used to avoid sparse data problem.

$$P(t_i|C) = \frac{doc_freq(t_i, e)}{\sum_{t' \in C} doc_freq(t', C)} \quad (4)$$

The element language model, $P(t_i|e)$, defines where our method differs from other language models. In our language model, $P(t_i|e)$ is estimated by a linear combination of two parts:

$$P(t_i|e) = \lambda_1 \cdot P(t_i|e_{full}) + (1 - \lambda - \lambda_1) \cdot P(t_i|e_{compact}) \quad (5)$$

where λ_1 is a mixture parameter; $P(t_i|e_{full})$ is a language model for the full-text of element e ; $P(t_i|e_{compact})$ is the estimate based on the compact representation of element e . Parameter λ and λ_1 play important roles in our model. Previous experiments[3, 8] suggested that there was a correlation between the value of the smoothing parameter and the size of the retrieval elements. Smaller average sizes of retrieved elements require more smoothing than larger ones. In our experiments, the retrieval units, which are XML elements, are relatively small. We set the smoothing parameter $\lambda = 0.6$. And λ_1 was set to 0.3. In summary, the probability of a query is calculated by

$$P(q|e) = \prod_{i=1}^k (0.6(t_i|C) + 0.3(t_i|e_{full}) + 0.1(t_i|e_{compact})) \quad (6)$$

3 INEX Experiments

In this section, we present our experiments in participating the INEX 2007 ad hoc track.

3.1 Index

We created inverted indexes of the collection using Lucene[4]. Indexes were word-based. All texts were lower-cased, stop-words removed using a stop-word list, but no stemming. We considered paragraph elements to be the lowest possible level of granularity of a retrieval unit. And indexed text segments consisting of paragraph elements and of elements containing at least one paragraph element as a descendant element. For the remainder of the paper, when we refer to the XML elements considered in our investigation, we mean the segments that correspond to paragraph elements and to their ancestors. For each XML element, all text nested inside it was indexed. In addition to this, we added an extra field which corresponded to the compact representation of the element. As some studies[3, 7] have already concluded that a successful element retrieval approach should be biased toward retrieving large elements, in the experiments, we indexed only those elements that consist of more than 200 characters (excluding stop words). The decision to measure in characters instead of words was based on the consideration that smaller segments such as “I like it.” contains little information, while a sentence with three longer words tends to be more informative.

3.2 Query processing

Our queries were created using terms only in the <title> parts of topics. Like the index, queries were word-based. The text was lower-cased and stop-words were removed, but no stemming was applied. ‘+’, ‘-’ and quotes in queries were simply removed. The modifiers “and” and “or” are ignored.

3.3 Submissions

We submitted 3 runs based on the language model, one for each of the three tasks: Focused-LM for the Focused task, RelevantInContext-LM for the Relevant-in-Context task, and BestInContext-LM for the Best-in-Context task.

In our experiments, the top ranked elements were returned for further processing. For the Focused task, overlaps were removed by applying a post-filtering on the retrieved ranked list by selecting the highest scored element from each of the paths. In case of two overlapping elements with the same relevance score, the child element was selected. For the Relevant-in-Context task, we simply took the results for the Focused task, reordered the elements in the list such that results from the same article were grouped together. In the Best-in-Context task, the element with the highest score was chosen for each document. If there were two or more elements with the same highest score, the one that appears first in the original document was selected. For each of the runs, the top 1,500 ranked elements were returned as answers.

4 Evaluation and results

The system’s performance was evaluated against the INEX human relevance assessments. Details of the evaluation metrics can be found in [6]. Table 1 lists

the result of our Focused run, where $iP@j$, $j \in [0.00, 0.01, 0.05, 0.10]$, is the interpolated precision at j recall level cutoffs, and MAiP is the mean average interpolated precision. Evaluation results of Relevant-in-Context runs and Best-in-Context runs are listed in table 2 and table 3, respectively. Here, $g[r]$, $r \in [5, 10, 25, 50]$, is non-interpolated generalized precision at r ranks; and MAgP is non-interpolated mean average generalized precision.

Table 1. Results of Focused runs (totally 79 submissions)

RunID	$iP@0.00$		$iP@0.01$		$iP@0.05$		$iP@0.10$		MAiP	
	score	rank	score	rank	score	rank	score	rank	score	rank
Focused-LM	0.4073	28	0.3786	19	0.3271	11	0.3054	9	0.1552	5

Table 2. Results of Relevant-in-Context runs (totally 66 submissions)

RunID	gP[5]		gP[10]		gp[25]		gp[50]		MAgP	
	score	rank	score	rank	score	rank	score	rank	score	rank
RelevantInContext-LM	0.1650	20	0.1421	17	0.1087	15	0.0810	17	0.0812	15

Table 3. Results of Best-in-Context runs (totally 71 submissions)

RunID	gP[5]		gP[10]		gp[25]		gp[50]		MAgP	
	score	rank	score	rank	score	rank	score	rank	score	rank
BestInContext-LM	0.3481	5	0.2953	3	0.2299	3	0.1765	4	0.1759	8

Due to the pressure of time, we did not submit baseline runs for retrieval models based on full-text solely or without priors for comparison.

5 Conclusions and Future Work

We have presented, in this paper, our experiments of using shallow structural features for the INEX 2007 evaluation campaign. We assumed important words could be identified according to the ways they were displayed in the text. We proposed to generate a compact representation of an XML element by extracting words appearing in titles, section titles, and figure captions the element nesting. Our retrieval methods emphasized the importance of these words in identifying relevance. We also integrated non-content priors that emphasized elements

appeared in the beginning part of the original text, and elements that are not nested deeply. We used a mixture language model combining estimates based on element full-text, the compact form of it, as well as the non-content priors. In general, our system performed well compared to other submissions. However, due to the pressure of time, we could not submit baseline runs for comparisons of exactly how these priors and compact forms improve performances.

Our future work will focus on refining the retrieval models. Currently, the compact representation of an element is generated by words from certain parts of the text. However, the effectiveness of this method depends on the type of the documents. For example, in scientific articles, section titles (such as introduction, conclusion, etc) are not very useful for relevance judgment, whereas section titles in news reports are very informative. In the future, we will explore different patterns for generating compact representations depending on types of texts. This might involve genre identification techniques. We will investigate different priors' effectiveness and how different types of evidence can be combined to boost retrieval effectiveness.

6 Acknowledgments

The Lucene-based indexer used this year was partly based on the indexing code developed for RGU INEX'06 by Stuart Watt and Malcolm Clark.

References

1. Geva G. Gardens point XML IR at INEX 2005. *Proceedings of Fourth Workshop of the INitiative for the Evaluation of XML Retrieval (INEX 2005), 2006*
2. Huang,F., Watt, S., Harper, D., Clark, M.: Compact representations in XML retrieval. *Comparative Evaluation of XML Information Retrieval Systems: Fifth International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX 2006), LNCS, Vol 4518, 2007*
3. Kamps J., Marx M., de Rijke M. and Sigurbjornsson B. XML retrieval: What to retrieve? *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2003*
4. Lucene. The Lucene search engine, 2005. <http://jakarta.apache.org/lucene>
5. Ogilvie P. and Callan J. Parameter estimation for a simple hierarchical generative model for XML retrieval. *Proceedings of Fourth Workshop of the INitiative for the Evaluation of XML Retrieval (INEX 2005), 2006*
6. Pehcevski, J., Kamps, J., Kazai, G., Lalmas, M., Ogilvie, P., Piwowarski, B., Robertson, S.: INEX 2007 Evaluation Measures. INEX2007.
7. Sigurbjornsson B., Kamps J. and de Rijke M. An element-based approach to XML retrieval. *INEX 2003 Workshop Proceedings, 2004*
8. Zhai C. and Lafferty J. A study of smoothing methods for language models applied to ad hoc information retrieval. *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2001*

The Simplest XML Retrieval Baseline That Could Possibly Work ^{*}

Philipp Dopichaj

dopichaj@informatik.uni-kl.de
University of Kaiserslautern
Gottlieb-Daimler-Str.
67663 Kaiserslautern
Germany

Abstract Five years of INEX have produced many competing XML element retrieval methods that make use of the document structure. So far, no clearly best method has been identified, and there is even no clear evidence what parts of the document structure can be used to improve retrieval quality. Little research has been done on simply using standard information retrieval techniques for XML retrieval. This paper aims at addressing this; it contains a detailed analysis of the BM25 similarity measure in this context, revealing that this can form a viable baseline method.

1 Introduction

In the five years since the inception of INEX, much research on XML element retrieval methods has been done by the participants. Through the use of the INEX test collections, it was possible to determine the retrieval quality of the competing retrieval engines. One thing all retrieval engines participating in INEX have in common is that they make use of the XML document structure in some way, based on the reasonable assumption that retrieval engines that use more of the information that is available can yield better results.

To our knowledge, this assumption has never been tested in detail. To close this gap, we provide a detailed analysis of the retrieval quality that can be achieved by simply using the standard BM25 similarity measure with minimum adaptations to XML retrieval.

1.1 Evaluation Metrics

Over the years, the evaluation metrics and retrieval tasks used for INEX have changed considerably. In this paper, we will only evaluate the *thorough* retrieval task; this task is the simplest of all INEX tasks, and the results for the other tasks are typically created by applying a postprocessing step to the *thorough* results.

^{*} ... and it does!

We use the standard nxCG measure as used for INEX 2005 and 2006 [], and the official assessments from the corresponding workshop web sites¹.

We do not use the official evaluation software EvalJ², but our own reimplementation of the official measures; this was necessary because the overhead of calling an external process would have been too high. We made sure that our version of the evaluation gives the same results (although at a slightly higher numerical accuracy).

1.2 Test Collections

The INEX workshops used a collection of IEEE computer society³ journal and transactions articles through 2005, where later versions of the collection are supersets of earlier versions (new volumes were added). From 2006 on, a conversion of the English version of Wikipedia was used [2]. The evaluations in this thesis will be based on the collections from 2004, 2005, and 2006. Figure 1 gives an overview of various characteristics of the document collections.

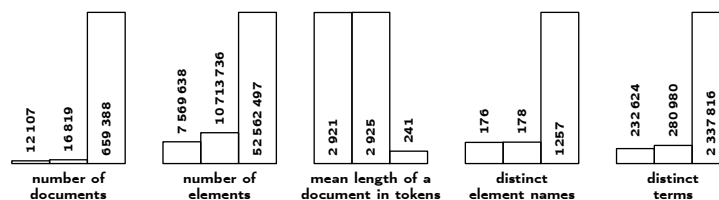


Figure 1: Test collections statistics. The bars in each group are, from left to right, the IEEE 1.4 collection (2004), the IEEE 1.9 collection (2005), and the Wikipedia collection (2006). The token count excludes stop words.

For each year of the workshop, a new set of topics was created by the participants, consisting of a longer description of the information need and a query in NEXI format. The number of topics varied: in 2004, there were 40 CO topics (34 have been assessed), in 2005, there were 40 topics (29 assessed), and in 2006, there were 130 topics (114 assessed). For our evaluations, we will only use content-only topics.

The assessment procedure has changed against the years: In 2004, the assessors had to manually select both specificity and exhaustiveness on a scale from 0 to 2 for each element in the recall base. In 2005, a highlighting approach was introduced; the assessor used a virtual highlighter to mark relevant passages in the documents to denote specificity. In the next step, the exhaustiveness had to be set for each element as in 2004. From 2006 on, exhaustiveness was dropped from the assessments, only the highlighting approach to selectivity was retained.

¹ see <http://inex.is.informatik.uni-duisburg.de/>

² see <http://evalj.sourceforge.net>

³ see <http://www.computer.org>

Note that we use nxCG for the evaluations on the INEX 2004 test collection, even though nxCG was not the official evaluation measure at the time. This is possible because the data that was collected for the assessments is compatible, and it makes the results presented in this paper more consistent and comparable. The results may not be as meaningful as the results for the other collections, but it is still interesting to see differences of behavior compared to the 2005 results, which are based on almost the same document collection.

1.3 Standard Similarity Measures

As mentioned before, we use the BM25 similarity measure as introduced by the Okapi project, as described by Robertson and Walker [10]. The core idea is the notion of *eliteness*, which denotes to what degree a document d is “elite” for term t . As with most information retrieval measures, eliteness is derived from the term frequency $\text{tf}(t, d)$, and each term has a *global weight* w_i , which is derived from the term’s document frequency $\text{df}(t)$ and the total number of documents.

The conversion from the plain term frequency to the term eliteness probability can be adapted with the global parameter k_1 ; the formula ensures that the term eliteness is 0 if the term frequency is 0, and it asymptotically approaches 1 as the term frequency increases. This implies that the first few occurrences of a term make the greatest contribution to term eliteness – the function is steep close to 0. The eliteness of term t for document d , using a document-length normalization constant K (see below) is defined as:

$$\text{eliteness}(t, d) = \frac{(k_1 + 1) \text{tf}(t_i, d)}{K + \text{tf}(t_i, d)} \cdot \underbrace{\log \frac{N - \text{df}(t_i) + 0.5}{\text{df}(t_i) + 0.5}}_{w_i} \quad (1)$$

An important feature of BM25 is *document-length normalization*. Based on the assumption that document length is caused either by needless verbosity – this implies normalization – or a more thorough treatment of the subject – this implies no normalization –, BM25 uses partial length normalization. The degree of normalization is controlled by a global parameter b .

$$K = k_1 \left((1 - b) + b \cdot \frac{\text{len}(d)}{\text{avg}(\text{len}(d))} \right) \quad (2)$$

The final similarity of document d to the query q consisting of terms $t_1 \dots t_m$ is then accumulated as follows (we assume that there are no weights attached to query terms):

$$\text{sim}(q, d) = \sum_{i=1}^m \text{eliteness}(t_i, d) \quad (3)$$

For completeness, we will also examine the similarity measure used by the Apache Lucene project⁴. This similarity measure proved to be effective for our INEX 2005 submissions, with minor adaptations [3].

⁴ see <http://lucene.apache.org>

$$\text{sim}(q, d) = \text{coord}(q, d) \sum_{t \in q} \left[\sqrt{\text{tf}(d, t)} \left(1 + \log \left(\frac{N}{\text{df}(t) + 1} \right) \right) \text{lnorm}(d) \right] \quad (4)$$

$$\text{lnorm}(d) = \frac{1}{\sqrt{\text{len}(d)}} \quad (5)$$

$$\text{coord}(q, d) = |\{t \in q : \text{tf}(d, t) > 0\}| \quad (6)$$

The coordination factor $\text{coord}(q, d)$ is the number of query terms in q that also occur in d . The intention is to reward documents that contain more of the query terms. The result is that documents that contain all the query terms will usually end up in the first ranks in the result list, which is usually the right thing to do.

1.4 Adaptation for XML Retrieval

The standard information retrieval similarity measures are based on the assumption that a document is atomic, that is, documents cannot be decomposed into sub-documents. This assumption is not valid for element retrieval, so minor adaptations have to be performed.

In particular, each document is split into its elements, and *every* element is stored in the index. The cost for indexing *all* elements may appear to be prohibitive, but with appropriate index structures, the overhead can be kept at an acceptable rate [5].

```
<section><title>Example document</title>
<p>A paragraph.</p>
<p>A paragraph with <it>inline</it> markup.</p>
</section>
```

(a) Input XML document.

XPath	Indexed contents
/section	Example document A paragraph. A paragraph with inline markup.
/section/title	Example document
/section/p[1]	A paragraph.
/section/p[2]	A paragraph with inline markup.
/section/p[2]/it	inline

(b) Indexed “documents”.

Figure 2: Example of XML document indexing.

One change that this entails is the choice of the global frequency (in the original formulas, document frequency). Of course, it is still possible to use document frequency in element retrieval, but this is not the only option. In fact, if every element is indexed as if it were a document, the new concept of *element frequency* might well be a more logical choice.

There are other options [12, 8], but they require larger changes to the standard information retrieval techniques and index structures, so we will not consider them here.

2 Parameter Tuning for the Baseline Retrieval Engine

For both similarity measures, BM25 and Lucene, we will tune the parameters to suit XML retrieval; the default parameters are good for standard information retrieval, but will probably have to be adapted for this new scenario. The results for these similarity measures will then be compared to the best submitted results of the corresponding INEX workshop to put things in context.

2.1 Lucene Similarity Measure

The Lucene similarity measure gave good results at least in 2005. In this section, we will evaluate two global weighting methods – element and document frequency – and a parameterizable version of Lucene’s length normalization function:

- Standard length normalization:

$$\text{lnorm}_{\text{luc}}(d) = \frac{1}{\sqrt{\text{len}(d)}} \quad (7)$$

- Standard length normalization with a constant value up to length l :

$$\text{lnorm}_{\text{const}}(d) = \frac{1}{\sqrt{\max(\text{len}(d), l)}} \quad (8)$$

The following parameter combinations have to be tested, using $\text{lnorm}_{\text{const}}$ (for 0, $\text{lnorm}_{\text{const}}$ is effectively $\text{lnorm}_{\text{luc}}$):

$$\underbrace{\{\text{df}, \text{ef}\}}_{\text{gf}} \times \underbrace{\{0, 5, 10, \dots, 195, 200\}}_{\text{lnorm}}$$

In our INEX submissions, we used a non-linear adaptation of Lucene’s function [3] – elements shorter than about 50 tokens basically get an RSV of 0. This length normalization function leads to inferior results in all experiments (in particular at higher ranks), so it is not included in the evaluation.

Tuning the length normalization is crucial to good performance, and what version is the best depends on the document collection. As figure 3 shows, for the IEEE collection, a soft threshold of 65 tokens yields the best results, whereas for the Wikipedia collection, a lower value of about 50 is better. This can be

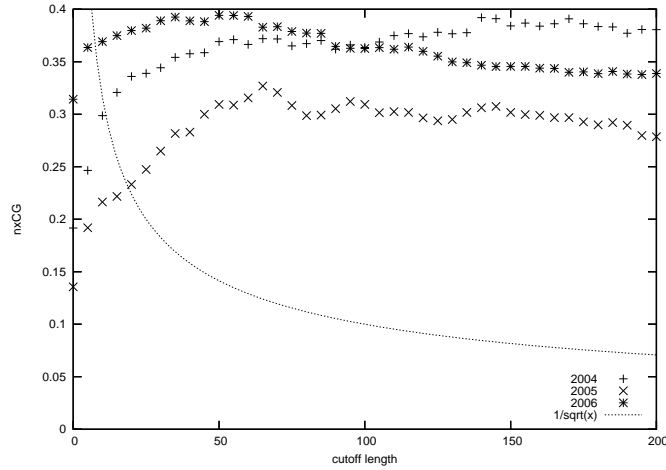


Figure 3: Lucene retrieval quality (nxCG@10), using document frequency. For reference, a plot of the Lucene length normalization function is included in the plot.

explained by the different typical lengths of the documents in the collections: IEEE articles are much longer than Wikipedia articles, so the relevant parts are also longer (but this might also be a side effect of the assessment procedure).

For INEX 2004, the results are significantly worse than the best official results; it is unclear what the reason is. For INEX 2005, the Lucene similarity measure can exceed the best official submission at rank 10 (our own submission also using Lucene with a different length normalization function). For INEX 2006, the best Lucene results are about 10 percent worse than the best submitted results.

The results for the different global weighting functions are close to one another. This indicates that it does not matter whether document or element frequency is used with the Lucene similarity measure.

2.2 BM25 Similarity Measure

For BM25, length normalization is controlled by the parameters b and k_1 . Permissible values for b are in the range $0 \dots 1$, where 0 means “no length normalization” and 1 means “maximum influence of length normalization”. The larger k_1 gets, the closer the local term weight gets to the raw term frequency.

According to Spärck Jones et al. [11], $b = 0.75$ and k between 1.2 and 2 work well on the TREC data, but it is unlikely that these parameter combinations can be transferred unchanged to XML retrieval. Theobald [12] uses $k_1 = 10.5$ and $b = 0.75$, but the TopX approach is sufficiently different from mine to warrant further exploration.

The following parameter combinations should be tested (the full range for b and a reasonable range for k_1):

$$\underbrace{\{0.0, 0.1, \dots, 1.0\}}_b \times \underbrace{\{1, 1.5, 2, \dots, 4.5, 5\}}_{k_1}$$

Figure 4 shows the results for the three test collections. It is obvious that a good choice of parameter b is much more critical than a good choice of k_1 . In general, lower values of b work better than higher values, with the exception of $b = 0$ (that is, no length normalization). Compared to the best parameter values for traditional information retrieval ($b = 0.75$ and $k_1 = 1.2$), the best value of b for element retrieval is much lower (somewhere between 0.1 and 0.2), so the influence of length normalization is reduced.

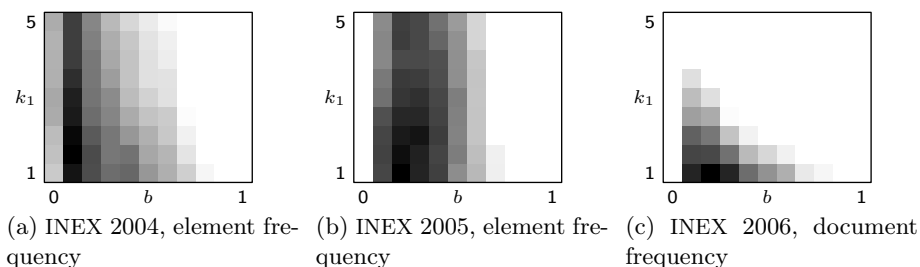


Figure 4: Parameter tuning for BM25; the darkness of each field corresponds to nxCG at cutoff rank 10. In each map, black corresponds to the maximum and white corresponds to 10 percent more than the minimum. The horizontal axis corresponds to b , from 0 to 1, and the vertical axis corresponds to k_1 , from 1 to 5.

Each parameter space has a global maximum; the parameters for this maximum are close for the different test collections, but not identical. In particular, it is surprising to see that the best parameters for 2004 and 2005 differ noticeably.

The reason is that in our usage scenario, length normalization also fulfills the purpose of selecting the right result granularity (should a chapter or a paragraph be ranked higher?). What happens is that for maximum length normalization ($b = 1$), very short elements are pushed to the front of the result lists, typically leading to a list of section titles or titles of cited works. This is obviously a bad result. With length normalization completely disabled ($b = 0$), there is a strong bias towards the longest elements, that is, complete articles or their bodies. For values of b between the extremes, the results are much more balanced; they are a mixture of sections, complete articles, and other elements. Although an occasional title does occur in the top ranks, this is the exception rather than the rule and does not do much harm. In fact, if all elements of fewer than ten terms are removed from the results, retrieval quality drops dramatically.

The best choice for the global frequency function depends on the document collection: Element frequency is best for the IEEE collection, whereas document frequency is better for the Wikipedia collection.

Using element frequency as the global frequency consistently leads to better results than using document frequency for the IEEE collection (2004 and 2005). Although this is consistent with the original formula, this result is somewhat surprising: Element frequency is not simple to interpret – terms that occur in deeply nested elements have a higher element frequency than terms that do not.

The explanation lies in a peculiarity of the BM25 formula: For terms that occur in more than half of all documents, the term weight w_i is negative so that the presence of these terms actually decreases the RSV:

$$w_i = \log \frac{N - \text{df}(t_i) + 0.5}{\text{df}(t_i) + 0.5} \quad (9)$$

To circumvent this problem, the term weight is generally set to 0 if it is negative, which means that these terms are treated as stop words.

In the IEEE collection, there are many terms that occur in more than half of the documents, so they cannot contribute to the RSV. There are, however, no terms for which the element frequency is high enough to obtain a negative weight, so this particular problem does not occur.

One might argue that terms that occur so frequently are useless for retrieval, but this is not necessarily the case for element retrieval: The terms “IEEE”, “volume”, and “computer” basically occur in all documents, so they have no discriminatory power at the document level. On the other hand, they may well be useful for element retrieval. For example, if a user searches for “IEEE conferences”, elements that mention both terms are likely to be relevant, but elements that only mention “conferences” will have a high rate of false positives.

For the 2006 data, the behavior of element and document frequency is roughly identical, with document frequency being slightly better. This discrepancy is somewhat puzzling: what characteristic affects this? In the Wikipedia collection, the topics of the documents are more diverse, so there are no terms (apart from stop words) that occur in more than half of the documents, so the problem of negative term weights does not occur. The only outlier in this respect is the term “0”, which occurs in almost all documents’ header.

Figure 5 illustrates the effect of the global frequency for all tested combinations of b and k_1 .

2.3 Comparison with the Official Submissions

So far, we have obtained the best BM25 parameter combinations for the various test collections, but it is still unclear how the results compare to the results of XML retrieval systems. It is hard to determine a single best official run, so we will compare the quality of the base retrieval engine with the maximum of all official submissions to that year’s workshop. That is, for each rank, the nxCG value averaged over all topics for each submission is calculated, and we use the

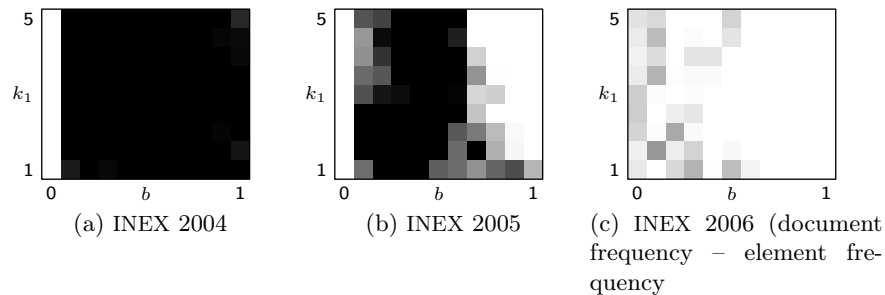


Figure 5: Choice of global frequency for BM25. The heat maps show the difference between the results for element frequency and the results for document frequency; each square corresponds to one combination of b and k_1 . White squares denote no change or better results for document frequency, all other shades of gray denote the degree of improvement when using element frequency.

maximum as the comparison run; the resulting curve does not correspond to a real run, but it gives us an indication of where the baseline stands with respect to the others. Lucene results are excluded because they are exceeded in all cases by BM25 results.

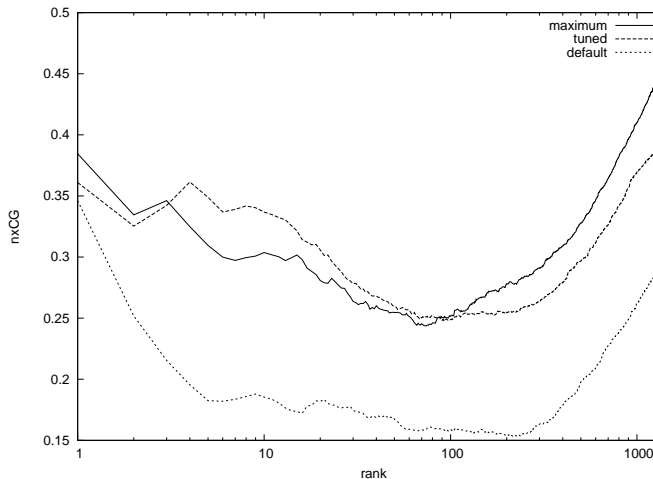
From the INEX 2005 results, one can see that unmodified BM25 already yields high-quality results, even compared to the official submissions. This is somewhat alarming, as it shows that the methods tailored to XML retrieval fail to better the general-purpose algorithms.

Further tuning resulted in the values presented in table 1. For INEX 2005, there is a noticeable increase in retrieval quality, whereas for INEX 2006, the increase is less pronounced. For INEX 2004, the optimum result of the base retrieval engine is significantly worse than the best submitted run. This is surprising, considering that the 2004 and 2005 collections basically use the same document collection. It should be noted, however, that the assessment procedure has changed between these rounds of INEX. Figure 6 shows the results for the 2005 and 2006 collections compared to the maximum of the submissions for all ranks and shows that the good quality at rank 10 is not completely isolated.

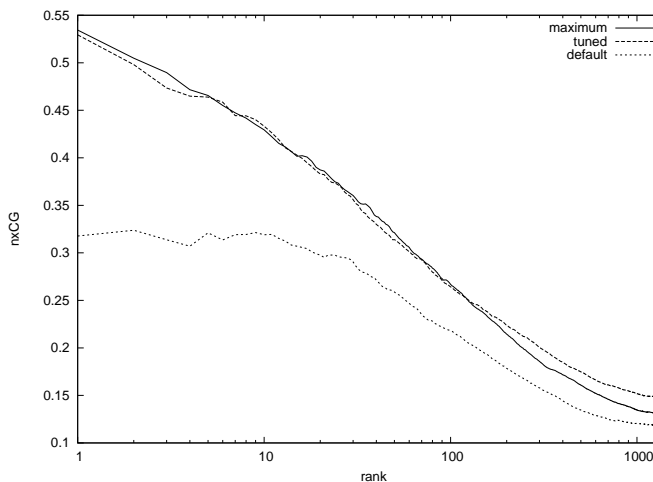
In a real-world scenario, there are usually no relevance assessments available, so it is impossible to find the optimal parameter values. However, the values for the 2005 and 2006 test collections are close in magnitude although the collections are very different; thus, one can assume that these values are good starting points for other collections.

3 Discussion

It is surprising to see how well a simple adaptation of standard information retrieval techniques can work for XML retrieval. Simply indexing all elements as if they were documents and applying BM25 with the right parameters can lead to



(a) INEX 2005, tuned is $b = 0.2, k_1 = 1$. The base retrieval engine is better than the best submissions up to about rank 100 (with the exception of the top ranks). Below that rank, performance gets significantly worse, possibly due to the pooling problems.



(b) INEX 2006, tuned is $b = 0.18, k_1 = 0.8$. Although the baseline does not quite reach the top status, it is close.

Figure 6: Two of the base BM25 runs compared with the maximum run (“maximum”). The BM25 run with $b = 0.75$ and $k_1 = 1.2$ (“default”) shows what can be achieved without parameter tuning and the “tuned” BM25 run shows the best parameter combination for the test collection.

Table 1: Best parameters and evaluation results for the different test collections. In all cases, the Lucene similarity measure yielded worse results. The “base” column displays the value for the base engine, the “max” column displays the maximum of all official submissions in that year. The maximum from 2005 is our own submission.

Test collection	Parameters			nxCg@10	
	b	k_1	gf	base	max
INEX 2004	0.08	1.5	ef	0.4669	0.5099
INEX 2005	0.20	1.0	ef	0.3368	0.3037
INEX 2006	0.18	0.8	df	0.4332	0.4294

better results than the best official submissions. One should keep in mind that the optimal parameters were determined after the fact by evaluating a large range of combinations on the assessed test data; the real submissions do not have the advantage of this fine-tuning.

On the other hand, the best parameters are very similar for the INEX 2005 IEEE collection and the Wikipedia collection, and minor deviations from the optimal results do not decrease retrieval quality much. Considering that these collections are very different from one another, it seems plausible to assume that using $b = 0.2$ and $k_1 = 1$ will work reasonably well in other situations. It is surprising that the best parameters are different for the INEX 2004 collection, which is almost identical to the 2005 collection. It is not clear what the reason is, but it should be kept in mind that we used an evaluation measure that was not official back then.

3.1 Realism of the Experiments

Keep in mind, however, that the test collections and evaluation metrics that are used at the INEX workshops do not entirely reflect the intended application area, and other potential problems may affect the results:

- Both the IEEE articles and the Wikipedia articles are rather short and self-contained so that it is unlikely that a fragment of such an article is more relevant than the article itself.
- The two collections differ in so many aspects that it is impossible to attribute the difference in retrieval quality to a single difference.
- The assessment process is not the same in different years, which makes it hard to do a comparison.
- Relevance assessments are generally subjective; in the cases where several people assessed the same topic, the assessments were quite different [13, 9].
- Runs that are evaluated, but were not included in the pooling process may suffer if they retrieve elements that are not in the pool. Whereas this effect has been shown to be minimal in the context of TREC [15], no study has been made in the context of INEX, but problems have been reported [14].

- The assessment interface differs from what a user of the retrieval system would see; it does not use ranking and is document-based, so the relation to real-world scenarios is unclear.

The last point needs further explanation: The unranked presentation of the results is inherent to the pooling approach that has successfully been used for traditional information retrieval evaluation for years. In the context of element retrieval, however, there is the problem that the pool does not reflect the retrieval results. Even if the pooled results only contain a single paragraph from a document, the assessor must assess the complete document. This in itself is a minor technical problem, but it seems likely that the assessment can be different from the assessment that would be obtained if the isolated paragraph were presented; if the paragraph is shown in the context of the document, the assessor may – consciously or not – use this context to rate the element’s relevance.

3.2 Evaluation Metrics

It is clear that even the INEX organizers and participants have not yet reached consensus on how to evaluate the effectiveness of XML retrieval systems: Through the years, various metrics were adopted and abandoned, and even the basic retrieval tasks for the ad-hoc track are far from being fixed (INEX 2007 dropped the *thorough* task, which previously was the only task that had been done in every year). This is not avoidable, considering that XML retrieval is still a relatively young research area, but the lack of clear definitions makes it hard to do meaningful comparisons between systems.

In general, it is questionable whether the results from batch evaluations – as done in the INEX ad-hoc track – contribute to user satisfaction. Hersh et al. [7] compare several systems’ performance on TREC data in batch and interactive experiments and come to the conclusion that there are significant differences in the results. In XML retrieval, the differences are likely to be even more pronounced, because the assessment user interface displays the results in a different fashion than an XML retrieval system would – the element results are shown in the context of the complete document. This is likely to affect the assessment: the users can take the surrounding material into account when judging the relevance of an element.

Buckley and Voorhees [1] discuss what it takes to draw conclusions with a sufficiently low error rate. The retrieval scenarios in this thesis are closest to their notion of web retrieval – it is very difficult to know how many relevant documents exist in total, so precision at a cutoff level of 10 to 20 should be used. In this scenario, precision is replaced by $nxCG$, but the reasoning is the same. To achieve a reasonable error rate, they suggest using 100 queries, which implies that only INEX 2006 data can be used to obtain reasonable conclusions (2004 and 2005 together have only 63 queries); unfortunately, the IEEE collection more closely matches the assumptions made in this thesis.

Overall, even document-based retrieval evaluation has problems, despite having a rather long tradition. For INEX, the problems are amplified by a number

of new problems, partly specific to XML, partly due to the resources being much more limited than for TREC. Evaluations in INEX data are certainly far from worthless, but they should be interpreted with care.

4 Conclusions

We have shown that standard information retrieval techniques can yield surprisingly good results for XML element retrieval, even compared to techniques specifically designed for XML retrieval. This does not imply that the existing XML retrieval methods are inferior; this paper only examined retrieval quality as determined by the standard measures, storage size and speed have not been addressed. It is conceivable that other methods yield comparable retrieval quality with less overhead, or are less sensitive to parameter changes; this should definitely be examined in future research. It is hard to say what exactly the reasons are, but we hope that future research will reveal techniques for exploiting the document structure to achieve greater retrieval quality.

We propose that BM25 with suitable parameters should be used as a baseline to compare XML retrieval systems against. This may lead to painful conclusions at first – for example, we found that our work on structural patterns [4] does not work as well as previously though [6] –, but in the long run, we believe that it will lead to a higher acceptance of XML retrieval in the standard information retrieval community.

Note that the results reported in this paper only pertain to content-only retrieval and the *thorough* retrieval task. It is obviously impossible to directly use standard techniques for content-and-structure retrieval, because the standard methods do not support structural queries. For the other content-only tasks, like *focused* and *in context*, postprocessing steps on the baseline results can be used; in fact, most INEX participants already derive the results for the advanced tasks from the *thorough* results. Thus, the next logical step for further research is to combine existing approaches for the advanced tasks with the baseline retrieval methods presented here and examine what the results are.

References

- [1] Chris Buckley and Ellen M. Voorhees. Evaluating evaluation measure stability. In *SIGIR 2000 proceedings*, pages 33–40. ACM, 2000.
- [2] Ludovic Denoyer and Patrick Gallinari. The Wikipedia XML corpus. *SIGIR Forum*, 40(1):64–69, 2006.
- [3] Philipp Dopichaj. The University of Kaiserslautern at INEX 2005. In *INEX 2005 proceedings*, pages 196–210. Springer, 2006.
- [4] Philipp Dopichaj. Improving content-oriented XML retrieval by applying structural patterns. In *ICEIS 2007 proceedings*, pages 5–13. INSTICC, 2007.
- [5] Philipp Dopichaj. Space-efficient indexing of XML documents for content-only retrieval. *Datenbank-Spektrum*, 7(23), November 2007.

- [6] Philipp Dopichaj. *Content-oriented retrieval on document-centric XML*. PhD thesis, University of Kaiserslautern, 2007. submitted.
- [7] William Hersh, Andrew Turpin, Susan Price, Benjamin Chan, Dale Kramer, Lynetta Sacherek, and Daniel Olson. Do batch and user evaluations give the same results? In *SIGIR 2000 proceedings*, pages 17–24. ACM, 2000.
- [8] Yosi Mass, Matan Mandelbrod, Einat Amitay, Yoelle Maarek, and Aya Soffer. JuruXML – an XML retrieval system at INEX '02. In *INEX 2002 proceedings*, pages 73–80, 2002.
- [9] Jovan Pehcevski and James A. Thom. HiXEval: Highlighting XML retrieval evaluation. In *INEX 2005 proceedings*, pages 43–57. Springer, 2006.
- [10] Stephen E. Robertson and Steve Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR 1994 proceedings*, pages 232–241. ACM, 1994. URL <http://portal.acm.org/citation.cfm?id=188490.188561>.
- [11] Karen Spärck Jones, Steve Walker, and Stephen E. Robertson. A probabilistic model of information and retrieval: development and status. Technical report, Computer Laboratory, University of Cambridge, 1998. URL <http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-446.html>.
- [12] Martin Theobald. *TopX – Efficient and Versatile Top-k Query Processing for Text, Structured, and Semistructured Data*. PhD thesis, Universität des Saarlandes, 2006.
- [13] Andrew Trotman. Wanted: Element retrieval users. In Andrew Trotman, Mounia Lalmas, and Norbert Fuhr, editors, *Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology*, pages 63–69, 2005. URL <http://www.cs.otago.ac.nz/inexmw/>. see <http://www.cs.otago.ac.nz/inexmw/>.
- [14] Andrew Trotman, Nils Pharo, and Dylan Jenkinson. Can we at least agree on something? In Andrew Trotman, Shlomo Geva, and Jaap Kamps, editors, *Proceedings of the SIGIR 2007 Workshop on Focused Retrieval*, pages 49–56, 2007. URL <http://www.cs.otago.ac.nz/sigirfocus/papers.html>.
- [15] Justin Zobel. How reliable are the results of large-scale information retrieval experiments? In *SIGIR 1998 proceedings*, pages 307–314. ACM, 1998. doi: <http://doi.acm.org/10.1145/290941.291014>.

ENSM-SE at INEX 2007: Scoring with Proximity

Extended abstract

Michel Beigbeder

École Nationale Supérieure des Mines de Saint-Étienne
michel.beigbeder@emse.fr

1 Introduction

The experiments conducted by the ENSM-SE in the INEX 2007 campaign for ad'hoc structured retrieval are based on the use of the proximity of the query terms in the documents. We will first present the notion of proximity between two terms. Then we will show how this notion can be extended to boolean queries. Given a proximity function mapping the positions in a textual document to $[0,1]$, a scoring function will be presented. Then we will present how these ideas are extended to structured documents.

2 Fuzzy proximity

2.1 Fuzzy proximity to a position

Given a position p_0 in a textual document it is easy to define a fuzzy proximity to p_0 with a function, $p \mapsto prox(p, p_0)$, that maps any position p in the document to $[0,1]$. Any function with the three following properties is acceptable and modelizes the proximity idea:

- symmetric around p_0 ,
- decreasing with the distance to p_0 ,
- maximum (value 1) reached at p_0 .

The simplest one is a linearly decreasing function centered around p_0 : $prox(p, p_0) = \max(\frac{k-|p-p_0|}{k}, 0)$ where k is a controlling parameter. When the distance between p and p_0 is greater than k , the fuzzy proximity is zero – that's to say that p is far from p_0 .

2.2 Fuzzy proximity to a term

Measuring the (fuzzy) proximity of a position in a document d to a query term t consists in measuring its fuzzy proximity to the nearest occurrence of the term t . As stated in section 2.1, proximity is decreasing with the distance so the proximity to the nearest occurrence is the maximum of the proximity to any occurrence:

$$prox_d(p, t) = \max_{p_0 \in Occ(d, t)} prox(p, p_0)$$

where $Occ(d, t)$ is the set of the positions of the occurrences of t in the document d .

2.3 Fuzzy proximity to two terms

Given one occurrence of a term t at position p_t and one occurrence of a term t' at position $p_{t'}$, we define the proximity to these two occurrences of t and t' by the minimum of the proximity to p_t and the proximity to $p_{t'}$. Again as the proximity function is decreasing with the distance, this minimum function reaches its maximum value at the middle of p_t and $p_{t'}$. Moreover the closer the positions p_t and $p_{t'}$, the higher the maximum.

We generalize to the proximity to the terms t and t' in a document with:

$$\min(\text{prox}_d(p, t), \text{prox}_d(p, t'))$$

This measures how far a position is from t and t' . So we can rewrite:

$$\text{prox}_d(p, t \wedge t') = \min(\text{prox}_d(p, t), \text{prox}_d(p, t'))$$

2.4 Fuzzy proximity to a query

Again it is easy to generalize the latter formula to any boolean query q with $\text{prox}_d(p, q)$. As a boolean query, the query q is a tree with conjunctive and disjunctive nodes. To define the proximity on a conjunctive node the minimum is taken over the proximity functions of its sons. Similarly, the proximity on a disjunctive node is defined as the maximum over the proximity functions of its sons.

2.5 Scoring a document by summation

Given the proximity function of a document d to a query q that maps the positions in the document d to $[0,1]$ with $\text{prox}_d(p, q)$, there are two basic ways to compute a score for the document: either by considering the maximum value of $\text{prox}_d(p, q)$, the second one is by summing this function over all the positions. We prefer the second one because it embeds the *tf* idea of the vector and probabilistic models. On another hand the first one could give the best entry point in the document.

3 Structured retrieval

3.1 Proximity in structured documents

To extend the proximity model to structured retrieval, we have to define proximity functions that take into account the structure.

The most simple and most used structure in document is the hierarchical one with sections, subsections, etc. where each instance at each level has got a title. With this kind of structure, we define the proximity to a position in a title as 1 (maximum value) over all the positions in the corresponding section. The idea is that if a term appears in a title, it is near every occurrence of every term that

appear in the corresponding section. For the terms that appear in the text of a section, their proximity is limited to the boundary of the section itself.

To take into account the much more complex structure of the actual documents found in the Wikipedia collection, we classified the XML elements in four categories. Two categories are related to the proximity introduced for hierarchical documents:

- title-like elements (`name`, `template`, `title` and `caption`)
- and section-like elements (`article`, `body`, `section`, `figure`, `image`, `page div`).

The two other classes are:

- soft elements: elements whose tags are ignored but their content is kept (*e.g.* `item`, `emph3`, `collectionlink`)
- deleted elements: elements whose content is deleted from indexation (*e.g.* `conversionwarning`, `math`, `aaa`, `aboutus`).

When a term appears in the content of a title-like element, the proximity function is set to one over all the extent of the immediately surrounding section-like element. When a term appears in the content of a section-like element its proximity function is limited to the extent of this element.

3.2 Scoring the elements of structured documents

Given the proximity function that maps the positions in a structured document to $[0,1]$, each XML element can be scored by summation of this function over the range of this element – again, maximizing this function is an alternative. However such scores are only computed for section-like elements and soft elements.

Finally a normalization is applied, and the sum is divided by the length of the element.

The Garnata Information Retrieval System at INEX'07

Luis M. de Campos, Juan M. Fernández-Luna, Juan F. Huete,
Carlos Martín-Dancausa, and Alfonso E. Romero

Departamento de Ciencias de la Computación e Inteligencia Artificial
E.T.S.I. Informática y de Telecomunicación, Universidad de Granada,
18071 – Granada, Spain
{lci,jmfluna,jhg,cmdanca,aeromero}@decsai.ugr.es

Abstract. This paper exposes the results of our participation in INEX07 in the AdHoc track and the comparison of these results with respect to the ones obtained last year. Three runs were submitted to each of the Focused, Relevant In Context and Best In Context tasks, all of them obtained with Garnata, our Information Retrieval System for structured documents. As is the past year, we use a model based on Influence Diagrams, the CID model. The result of our participation has been better than the last year so we have reached an acceptable position in the ranking for the three tasks. In the paper we describe the model, the system and we show the differences between our systems in INEX'06 and in INEX'07 which make possible to get a better performance.

1 Introduction

This is the second year that members of the research group “Uncertainty Treatment in Artificial Intelligence” at the University of Granada submit runs to the INEX official tasks, although before 2006 we also contributed to INEX with the design of topics and the assessment of relevance judgements. Like in the past year, we have participated in the Ad hoc Track with an experimental platform to perform structured retrieval using Probabilistic Graphical Models [5–7], called Garnata [4].

This year we have improved the version of Garnata that we used at INEX'06 in two ways, and we have also adapted it to cope with the three, non thorough tasks proposed this year, namely focused, relevant in context and best in context. For each of these tasks, we have submitted three runs, all of them using Garnata with a different set of parameters. The results of this second participation are considerably better than those of the past year, where we were in the last positions of the ranking. Nevertheless, we are still quite far from the first positions, so there is still room for improvement, and more research and experimentation need to be carried out.

The paper is organised as follows: the next section describes the probabilistic graphical models underlying Garnata. Sections 3 and 4 give details about the new characteristics/improvements incorporated into the system and the adaptation

of Garnata to generate outputs valid for the three tasks, respectively. In Section 5 we discuss the experimental results. The paper ends with the conclusions and some proposals for future work with our system.

2 Probabilistic Graphical Models in the Garnata System

The Garnata IRS is based on probabilistic graphical models, more precisely an influence diagram and the corresponding underlying Bayesian network. In this section we shall describe these two models and how they are used to retrieve document components from a document collection through probabilistic inference (see [2, 3] for more details). We assume a basic knowledge about graphical models.

2.1 The Underlying Bayesian Network

We consider three different kinds of entities associated to a collection of structured documents, which are represented by the means of three different kinds of random variables: *index terms*, *basic structural units*, and *complex structural units*. These variables are in turn represented in the Bayesian network through the corresponding *nodes*. Term nodes form the set $\mathcal{T} = \{T_1, T_2, \dots, T_l\}$; $\mathcal{U}_b = \{B_1, B_2, \dots, B_m\}$ is the set of basic structural units, those document components which only contain terms, whereas $\mathcal{U}_c = \{S_1, S_2, \dots, S_n\}$ is the set of complex structural units, that are composed of other basic or complex units. For those units containing both text and other units, we consider them as complex units, and the associated text is assigned to a new basic unit called *virtual unit*, see the example in Figure 1¹. The set of all structural units is therefore $\mathcal{U} = \mathcal{U}_b \cup \mathcal{U}_c$.

The binary random variables associated with each node T , B or S take its values from the sets $\{t^-, t^+\}$, $\{b^-, b^+\}$ or $\{s^-, s^+\}$ (the term/unit is not relevant or is relevant), respectively. A unit is considered relevant for a given query if it satisfies the user's information need expressed by this query. A term is relevant in the sense that the user believes that it will appear in relevant units/documents.

Regarding the arcs of the model, there will be an arc from a given node (either term or structural unit) to the particular structural unit the node belongs to. The hierarchical structure of the model determines that each structural unit $U \in \mathcal{U}$ has *only one* structural unit as its child: the unique structural unit containing U (except for the leaf nodes, i.e. the complete documents, which have no child). We shall denote $U_{hi(U)}$ the single child node associated with node U (with $U_{hi(U)} = \text{null}$ if U is a leaf node).

To assess the numerical values for the required probabilities $p(t^+)$, $p(b^+|pa(B))$ and $p(s^+|pa(S))$, for every node in \mathcal{T} , \mathcal{U}_b and \mathcal{U}_c , respectively, and every configuration $pa(X)$ of the corresponding parent sets $Pa(X)$, we use the canonical

¹ Of course this type of unit is non-retrievable and it will not appear in the XPath route of its descendants, is only a formalism that allows us to clearly distinguish between units containing only text and units containing only other units.

model proposed in [1], which supports a very efficient inference procedure. These probabilities are defined as follows:

$$\forall B \in \mathcal{U}_b, \quad p(b^+ | pa(B)) = \sum_{T \in R(pa(B))} w(T, B), \quad (1)$$

$$\forall S \in \mathcal{U}_c, \quad p(s^+ | pa(S)) = \sum_{U \in R(pa(S))} w(U, S), \quad (2)$$

where $w(T, B)$ is a weight associated to each term T belonging to the basic unit B and $w(U, S)$ is a weight measuring the importance of the unit U within S . In any case $R(pa(U))$ is the subset of parents of U (terms for B , and either basic or complex units for S) relevant in the configuration $pa(U)$, i.e., $R(pa(B)) = \{T \in Pa(B) \mid t^+ \in pa(B)\}$ and $R(pa(S)) = \{U \in Pa(S) \mid u^+ \in pa(S)\}$. These weights can be defined in any way with the only restrictions that

$$w(T, B) \geq 0, \quad w(U, S) \geq 0, \quad \sum_{T \in Pa(B)} w(T, B) \leq 1, \quad \text{and} \quad \sum_{U \in Pa(S)} w(U, S) \leq 1.$$

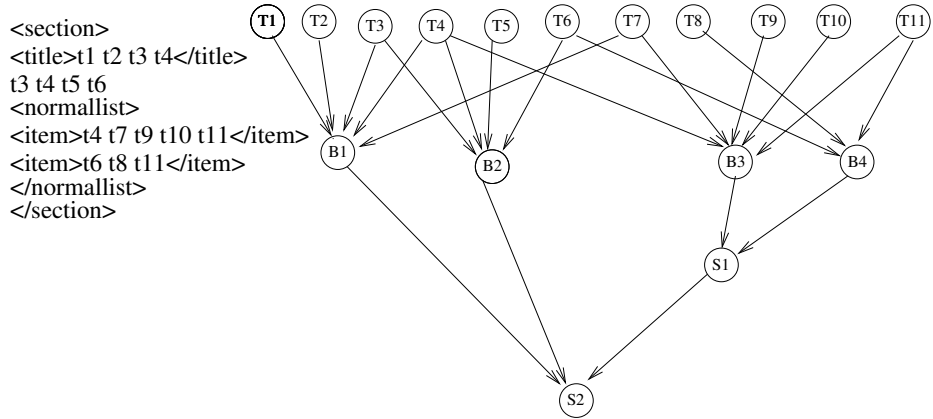


Fig. 1. Sample XML text and the corresponding Bayesian network. T_i represent index terms; the basic unit B_1 corresponds with the tag `<title>`, and B_3 and B_4 with the tag `<item>`; the complex units S_1 and S_2 correspond with the tags `<normallist>` and `<section>` respectively; B_2 is a virtual unit used to store the text within S_2 which is not contained in any other unit inside it.

2.2 The Influence Diagram Model

The Bayesian network is now enlarged by including decision nodes, representing the possible alternatives available to the decision maker, and utility nodes, thus

transforming it into an influence diagram. For each structural unit $U_i \in \mathcal{U}$, R_i represents the decision variable related to whether or not to return U_i to the user (with values r_i^+ and r_i^- , meaning ‘retrieve U_i ’ and ‘do not retrieve U_i ’, respectively), and the utility node V_i measures the value of utility for the corresponding decision. We shall also consider a *global utility node* Σ representing the joint utility of the whole model (we assume an additive behavior of the model).

In addition to the arcs between the nodes present in the Bayesian network, a set of arcs pointing to utility nodes are also included, employed to indicate which variables have a direct influence on the desirability of a given decision. In order to represent that the utility function of V_i obviously depends on the decision made and the relevance value of the structural unit considered, we use arcs from each structural unit node U_i and decision node R_i to the utility node V_i . Moreover, we include also arcs going from $U_{hi(U_i)}$ to V_i , which represent that the utility of the decision about retrieving the unit U_i also depends on the relevance of the unit which contains it (of course, for those units U where $U_{hi(U)} = \text{null}$, this arc does not exist). The utility functions associated to each utility node V_i are therefore $v(r_i, u_i, u_{hi(U_i)})$, with $r_i \in \{r_i^-, r_i^+\}$, $u_i \in \{u_i^-, u_i^+\}$, and $u_{hi(U_i)} \in \{u_{hi(U_i)}^-, u_{hi(U_i)}^+\}$.

Finally, the utility node Σ has all the utility nodes V_i as its parents. These arcs represent the fact that the joint utility of the model will depend on the values of the individual utilities of each structural unit. Figure 2 displays the influence diagram corresponding to the previous example.

2.3 Inference and Decision Making

Our objective is, given a query, to compute the expected utility of retrieving each structural unit, and then to give a ranking of those units in decreasing order of expected utility (at this moment we assume a thorough task, i.e. structural units in the output may overlap. In Section 4 we shall see how overlapping may be removed). Let $\mathcal{Q} \subseteq \mathcal{T}$ be the set of terms used to express the query. Each term $T_i \in \mathcal{Q}$ will be instantiated to t_i^+ ; let q be the corresponding configuration of the variables in \mathcal{Q} . We wish to compute the expected utility of each decision given q . As we have assumed a global additive utility model, and the different decision variables R_i are not directly linked to each other, we can process each one independently. The expected utilities for retrieving each U_i can be computed by means of:

$$EU(r_i^+ | q) = \sum_{\substack{u_i \in \{u_i^-, u_i^+\} \\ u_{hi(U_i)} \in \{u_{hi(U_i)}^-, u_{hi(U_i)}^+\}}} v(r_i^+, u_i, u_{hi(U_i)}) p(u_i, u_{hi(U_i)} | q) \quad (3)$$

Although the bidimensional posterior probabilities $p(u_i, u_{hi(U_i)} | q)$ in eq. (3) could be computed exactly, it is much harder to compute them than the unidimensional posterior probabilities $p(u_i | q)$, which can be calculated very efficiently due to

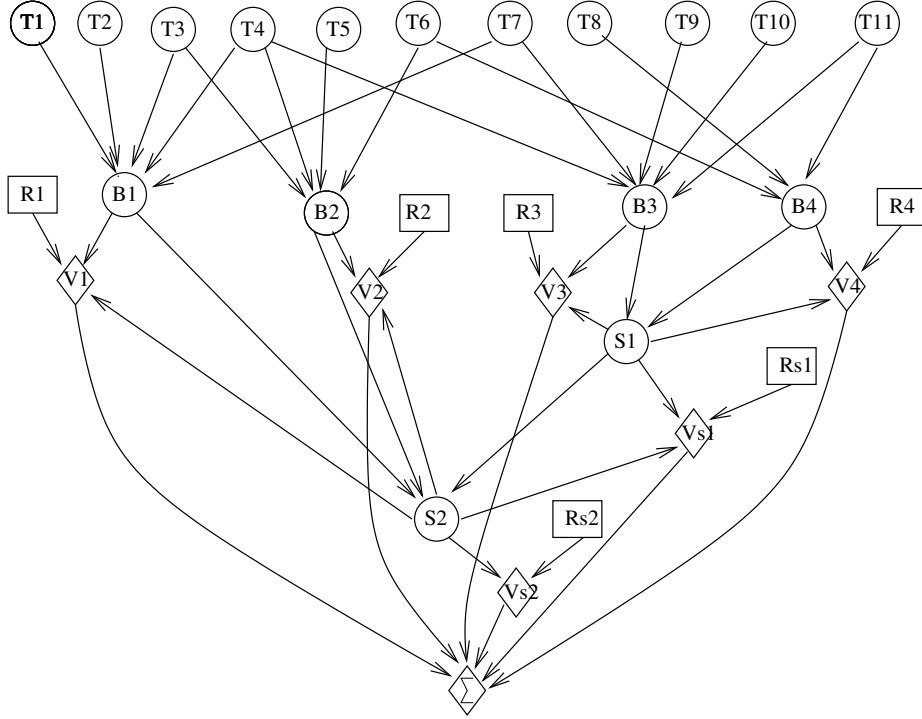


Fig. 2. Influence diagram for the example in Figure 1.

the specific characteristics of the canonical model used to define the conditional probabilities and the network topology. So, we approximate the bidimensional probabilities as $p(u_i, u_{hi(U_i)}|q) = p(u_i|q) \times p(u_{hi(U_i)}|q)$. The computation of the unidimensional probabilities is based on the following formulas [2, 3]:

$$\forall B \in \mathcal{U}_b, \quad p(b^+|q) = \sum_{T \in Pa(B) \setminus \mathcal{Q}} w(T, B) p(t^+) + \sum_{T \in Pa(B) \cap R(q)} w(T, B), \quad (4)$$

$$\forall S \in \mathcal{U}_c, \quad p(s^+|q) = \sum_{U \in Pa(S)} w(U, S) p(u^+|q). \quad (5)$$

Figure 3 shows an algorithm that efficiently computes these probabilities, derived from eqs. (4) and (5), traversing only the nodes in the graph that will require updating. It is assumed that the prior probabilities of all the nodes are stored in $\text{prior}[X]$; the algorithm uses variables $\text{prob}[U]$ which, at the end of the process, will store the corresponding posterior probabilities. Essentially, the algorithm starts from the terms in \mathcal{Q} and carries out a width graph traversal until it reaches the basic units that require updating, thus computing $p(b^+|q)$. Then,

starting from these modified basic units, it carries out a depth graph traversal to compute $p(s^+|q)$, only for those complex units that require updating.

```

for each item  $T$  in  $\mathcal{Q}$ 
  for each unit  $B$  child of  $T$ 
    if ( $\text{prob}[B]$  exists)
       $\text{prob}[B] += w(T,B) * (1 - \text{prior}[T]);$ 
    else { create  $\text{prob}[B];$ 
           $\text{prob}[B] = \text{prior}[B] + w(T,B) * (1 - \text{prior}[T]);$  }
for each basic unit  $B$  s.t.  $\text{prob}[B]$  exists {
   $U = B; \text{prod} = \text{prob}[B] - \text{prior}[B];$ 
  while ( $U_{hi(U)}$  is not NULL) {
     $S = U_{hi(U)};$ 
     $\text{prod} *= w(U,S);$ 
    if ( $\text{prob}[S]$  exists)
       $\text{prob}[S] += \text{prod};$ 
    else { create  $\text{prob}[S];$ 
           $\text{prob}[S] = \text{prior}[S] + \text{prod};$  }
     $U = S; }$ 
  }

```

Fig. 3. Computing $p(b^+|q)$ and $p(s^+|q)$.

The algorithm that initialises the process by computing the prior probabilities $\text{prior}[U]$ (as the terms $T \in \mathcal{T}$ are root nodes, the prior probabilities $\text{prior}[T]$ do not need to be calculated, they are stored directly in the structure) is quite similar to the previous one, but it needs to traverse the graph starting from all the terms in \mathcal{T} .

3 Changes from the Model Presented at INEX 2006

As the two changes with respect to the model used at INEX'06 refers to the parametric part of the model, first we are going to describe in some detail which are these parameters and how they were computed, and next to explain the proposed changes.

3.1 Parameters in Garnata

The parameters that need to be fixed in order to use Garnata are the prior probabilities of relevance of the terms, $p(t^+)$, the weights $w(T, B)$ and $w(U, S)$ used in eqs. (4) and (5), and the utilities $v(r_i^+, u_i, u_{hi(U_i)})$.

For the prior probabilities Garnata currently uses an identical probability for all the terms, $p(t^+) = p_0, \forall T \in \mathcal{T}$, with $p_0 = \frac{1}{|\mathcal{T}|}$.

The weights of the terms in the basic units, $w(T, B)$, follow a normalized tf-idf scheme:

$$w(T, B) = \frac{tf(T, B) \times idf(T)}{\sum_{T' \in Pa(B)} tf(T', B) \times idf(T')} \quad (6)$$

The weights of the units included in a complex unit, $w(U, S)$, measure, to a certain extent, the proportion of the content of the unit S which can be attributed to each one of its components:

$$w(U, S) = \frac{\sum_{T \in An(U)} tf(T, An(U)) \times idf(T)}{\sum_{T \in An(S)} tf(T, An(S)) \times idf(T)} \quad (7)$$

where $An(U) = \{T \in \mathcal{T} \mid T \text{ is an ancestor of } U\}$, i.e., $An(U)$ is the set of terms that are included in the structural unit U .

The utilities which are necessary to compute the expected utility of retrieving structural units, $EU(r_i^+ \mid q)$, namely $v(r_i^+, u_i, u_{hi(U_i)})$, are composed of a component which depends on the involved unit and another component independent on the specific unit and depending only on which one of the four configurations, $(u_i^-, u_{hi(U_i)}^-)$, $(u_i^-, u_{hi(U_i)}^+)$, $(u_i^+, u_{hi(U_i)}^-)$ or $(u_i^+, u_{hi(U_i)}^+)$, is being considered:

$$v(r_i^+, u_i, u_{hi(U_i)}) = nidf_Q(U_i) \times v(u_i, u_{hi(U_i)}) \quad (8)$$

with $v(u_i^-, u_{hi(U_i)}^-) = v^{--}$, $v(u_i^-, u_{hi(U_i)}^+) = v^{-+}$, $v(u_i^+, u_{hi(U_i)}^-) = v^{+-}$ and $v(u_i^+, u_{hi(U_i)}^+) = v^{++}$.

The part depending on the involved unit is defined as the sum of the inverted document frequencies of those terms contained in U_i that also belong to the query Q , normalized by the sum of the idfs of the terms contained in the query (a unit U_i will be more useful, with respect to a query Q , as more terms indexing U_i also belong to Q):

$$nidf_Q(U_i) = \frac{\sum_{T \in An(U_i) \cap Q} idf(T)}{\sum_{T \in Q} idf(T)} \quad (9)$$

Regarding the other component of the utility function independent on the involved unit, at INEX 2006 we used the following values

$$v^{--} = v^{-+} = v^{+-} = 0, \quad v^{++} = 1$$

3.2 Changing Weights

We have modified the weights of the units included in a complex unit, $w(U, S)$, in order to also take into account, not only the proportion of the content of S which is due to U , but also some measure of the importance of the type (tag) of unit U within S . For example, the terms contained in a `collectionlink` (generally proper nouns and relevant concepts) or `emph2` should be quantified higher than terms outside those units. Units labeled with `title` are also very informative, but units with `template` are not.

So, we call I_U the *importance of the unit* U , which depends of the type of tag associated to U . These values constitute a global set of free parameters, specified at indexing time. The new weights $nw(U, S)$, are then computed from the old ones in the following way:

$$nw(U, S) = \frac{I(U) \times w(U, S)}{\sum_{U' \in Pa(S)} I(U') \times w(U', S)} \quad (10)$$

Then, we show the three different importance schemes used in the official runs. Unspecified importance values are set to 1 (notice that by setting $I_U = 1, \forall U \in \mathcal{U}$, we get the old weights).

“Pesos 8:”

```
conversionwarning 0
emph2 10
emph3 10
name 20
title 20
caption 10
collectionlink 10
language link 0
template 0
```

“Pesos 11:”

```
conversionwarning 0
emph2 30
emph3 30
name 100
title 50
caption 10
collectionlink 10
language link 0
template 0
```

“Pesos 15:”

```
conversionwarning 0
emph2 30
emph3 30
name 200
title 50
caption 30
collectionlink 30
language link 0
template 0
```

3.3 Changing Utilities

This year the formula of the utility values for a unit U is computed by considering another factor called *relative utility value*, $RU(U)$, which depends only on the kind of tag associated to that unit, so that:

$$v(r_i^+, u_i, u_{hi(U_i)}) = nidf_Q(U_i) \times v(u_i, u_{hi(U_i)}) \times RU(U_i) \quad (11)$$

It should be noticed that this value $RU(U)$ is different from the importance $I(U)$: a type of unit may be considered very important to contribute to the relevance degree of the unit containing it and, at the same time, is considered not very useful to retrieve this type of unit itself. For example, this may be the case of units having the tag `<title>`: in general a title alone may be not very useful for a user as the answer to a query, probably the user would prefer to get the content of the structural unit having this title; however, terms in a title tends to be highly representative of the content of a document part, so that the importance of the title should be greater than the importance derived simply of the proportion of text that the title contains (which will be quite low).

The sets of utility values used in the official runs are:

No utilities:

All the units are given a relative utility value equal to 1

“Util 1:”

```
conversionwarning 0
name 0.75
title 0.75
collectionlink 0.75
language link 0
article 2
section 1.5
p 1.5
body 1.5
```

“Util 2:”

```
conversionwarning 0
emph2 1.5
emph3 1.5
name 0.75
title 0.75
collectionlink 1.5
language link 0
article 2.5
```

“Util 3:”

conversionwarning 0
name 0.85
title 0.85
collectionlink 0.75
language link 0
article 2.5
section 1.25
p 1.5
body 2

In all the cases, the default value for the non-listed units is 1.0.

4 Adapting Garnata to the INEX 2007 Ad Hoc Retrieval Tasks

For each query, Garnata generates a list of document parts or structural units, ordered by relevance value (expected utility), as the output. So, this output is compatible with the thorough task used in previous editions but not with the three adhoc tasks for INEX 2007, *focused*, *relevant in context* and *best in context*. To cope with these tasks, we still use Garnata but after we filter its output in a way which depends on the kind of task:

Focused task: The output must be an ordered list of structural units where overlapping has been eliminated. So, we must supply some criterion to decide, when we find two overlapping units in the output generated by Garnata, which one to preserve in the final output. The criterion we have used is to keep the unit having the greatest relevance value and, in case of tie, we keep the more general unit (the one containing a larger amount of text).

Relevant in context task: In this case the output must be an ordered list of documents and, for each document, a set of non-overlapping structural units, representing the relevant text within the document (i.e., a list of non-overlapping units clustered by document). Therefore, we have to filter the output of Garnata using two criteria: how to select the non-overlapping units for each document, and how to rank the documents. To manage overlapping units we use the same criterion considered for the focused task. To rank the documents, we have considered three criteria to assign a relevance value to the entire document: the relevance value of a document is equal to: (1) the maximum relevance value of its units; (2) the relevance value of the `"/article[1]"` unit; (3) the sum of the relevance values of all its units. Some preliminary experimentation pointed out that the maximum criterion performed better, so we have used it in the official runs.

Best in context task: The output must be an ordered list composed of a single unit per document. This single document part should correspond to the best entry point for starting to read the relevant text in the document. Therefore, we have to provide a criterion to select one structural unit for each document

and another to rank the documents/selected units. This last criterion is the same considered in the relevant in context task (the maximum relevance value of its units). Regarding the way of selecting one unit per document, the idea is to choose some kind of *centroid* structural unit: for each unit U_i we compute the sum of the distances from U_i to each of the other units U_j in the document, the distance between U_i and U_j being measured as the number of links in the path between units U_i and U_j in the XML tree times the relevance value of unit U_j ; then we select the unit having minimum sum of distances. In this way we try to select a unit which is nearest to the units having high relevance values.

5 Results of our model at INEX'07

We have obtained the following results in the three tasks, using the combinations of weight and utility configurations displayed in the tables:

Focused:

Weight file	Utility file	Ranking
8	3	67/79
15	No	69/79
15	2	71/79

Relevant in Context:

Weight file	Utility file	Ranking
15	3	44/66
8	3	45/66
11	1	49/66

Best in Context:

Weight file	Utility file	Ranking
8	3	45/71
15	No	46/71
15	2	50/71

As we can see in these results, the configuration of utilities with the value 3 is the most appropriate to get the best results in the different tasks, although we can not fix a specific configuration of weights that obtain the same results.

Finally, we show the graphics of the different tasks, where we can see the comparison of our results (red lines) with the results of the other organizations.

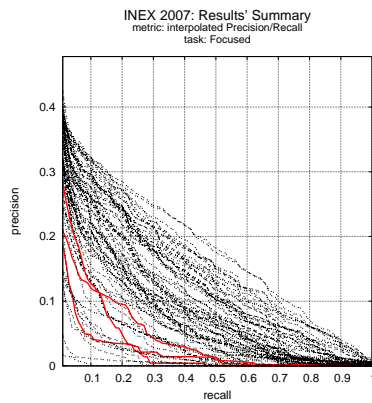


Fig. 4. Results on the Focused task

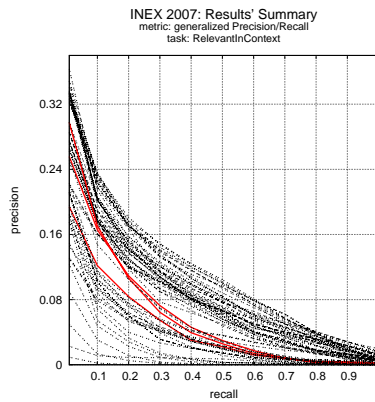


Fig. 5. Results on the Relevant In Context task

We have come to the conclusion that our system gets better results than the year before, so we have reached a middle position in the ranking (except for the focused task, where the results are worse) as we can see in the graphics and in the tables.

6 Concluding Remarks

In this year, our participation in the AdHoc track has been more productive than the one presented last year. In 2006, we only applied for one of the four

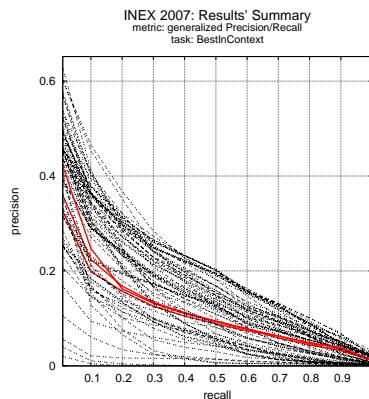


Fig. 6. Results on the Best In Context task

AdHoc tasks (Thorough), and in 2007 we have sent results for all the tasks of the track. Besides, on 2006 we got a very bad ranking (lying on the percentile 91). The best runs of this year are clearly better than the one obtained last year (corresponding to percentiles **84** [Focused], **66** [Relevant in Context] and **63** [Best in Context]).

Results in the “Relevant in Context” and “Best in Context” tasks are at the end of the second-third of the ranking, but in “Focused” they are in a very low position. So, the filter used for “Focused” should be improved much more.

On the other hand, we have not done yet a deep experimentation of different configurations for both the importance and the utility values. The used values are randomly selected configurations that obtained good results with the queries and the judgements of the wikipedia collection at INEX 2006. We think that the behaviour of our model could be clearly improved with a more systematic experimentation finding an optimal configuration of the parameters. We hope to include this experimentation in the final version of the paper.

Acknowledgments. This work has been jointly supported by the Spanish Ministerio de Educación and Ciencia, and Junta de Andalucía, under projects TIN2005-02516 and TIC-276, respectively.

References

1. L.M. de Campos, J.M. Fernández-Luna and J.F. Huete. The BNR model: foundations and performance of a Bayesian network-based retrieval model. *Int. J. Appr. Reason.*, **34**: 265–285, 2003.
2. L.M. de Campos, J.M. Fernández-Luna and J.F. Huete. Using context information in structured document retrieval: An approach using Influence diagrams. *Inform. Process. Manag.*, **40**(5): 829–847, 2004.

3. L.M. de Campos, J.M. Fernández-Luna and J.F. Huete. Improving the context-based influence diagram for structured retrieval. *Lect. Notes Comput. Sc.*, **3408**: 215–229, 2005.
4. L.M. de Campos, J.M. Fernández-Luna, J.F. Huete and A.E. Romero. Garnata: An information retrieval system for structured documents based on probabilistic graphical models. Proceedings of the Eleventh International Conference of Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU), 1024–1031, 2006
5. F.V. Jensen. *Bayesian Networks and Decision Graphs*, Springer Verlag, 2001.
6. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan and Kaufmann, San Mateo, 1988.
7. R. Shachter. Probabilistic inference and influence diagrams. *Oper. Res.*, **36**(5): 527–550, 1988.

Preliminary Work on XML Retrieval

(Extended Abstract)

Qiuyue Wang^{1,2}, Qiushi Li^{1,2}, Shan Wang^{1,2},

¹ School of Information, Renmin University of China,

² Key Laboratory of Data Engineering and Knowledge Engineering, MOE,
Beijing 100872, P. R. China

{qiuyuew, liqiushi, swang}@ruc.edu.cn

Abstract. As our preliminary work on XML retrieval, we conducted a series of experiments to investigate and analyze different XML retrieval strategies within a single framework, in order to detect key factors affecting the performance and get better knowledge about the problems. We use INEX as the test bed, and implement all the strategies in the Lemur/Indri system.

1 Introduction

As the de-facto standard for data representation and exchange on the Web, XML is being widely used in many applications. The need of full-text search and relevance ranking is called for when looking for information in large amounts of heterogeneous or document-centric XML data.

In contrast with traditional information retrieval systems, XML-IR systems aim to retrieve the document fragments (e.g. elements in XML documents), rather than the whole documents, relevant to user queries. The straightforward approach of applying the existing information retrieval models to XML element retrieval is to adapt the granularity of statistics from documents as retrieval units to elements as retrieval units. For example, the term frequency in a document is changed into term frequency in an element. Each element is viewed as “bag of words” consisting all the terms contained in the subtree rooted at the element, and scored individually. The direct application of the traditional flat-text IR models does not fully exploit the structural information in XML documents. How to exploit the structural information to enhance the effectiveness in XML retrieval, however, remains a major challenge and unresolved problem.

A well-accepted idea to exploit the hierarchical structure in XML documents is to score the leaf elements that directly contain terms and propagate the scores up to their ancestors. Thus the scores of elements up in the tree are calculated as weighted combinations of their descendants’ scores. The weights are usually less than 1 as the lower elements are considered as more specific than the upper elements [1]. Such a score propagation strategy can reflect the hierarchical level of the elements and also the weights can be set to reflect the importance of different element types.

As our preliminary work on XML retrieval, we attempt to evaluate and compare the different strategies and state-of-art scoring models for XML retrieval in a single framework, in order to identify how various factors affect the performance and gain better understanding into the problem. In this paper, we describe and analyze the work on comparing different XML retrieval strategies.

The paper is organized as follows. Section 2 describes the basic approach of XML retrieval by applying the conventional IR models directly to retrieve elements instead of documents. Section 3 discusses the hierarchical retrieval approach of considering structure in XML documents by applying score propagation in computing scores. In Section 4, we present the series of experiments for evaluating the different retrieval strategies. We conclude the paper in Section 5.

2 Basic Retrieval Models

The basic approach for XML element retrieval is to apply the existing IR models directly with the statistics collected at the element level. The content of each element consists of all the terms contained in all the descendants of the element along with itself, referred as the *full content* of the element. Each element is scored independently by any existing scoring models, and a ranked list of elements is returned. In this paper, we evaluate three classes of IR models for scoring elements using the basic approach, i.e. vector space model, probabilistic model, and language model.

For all the models that we evaluated, we collect the statistics with the notations as follows.

$tf(t, col)$: term frequency in the collection;
 $tf(t, d)$: term frequency in the document “d”;
 $tf(t, e)$: term frequency in the element “e”;
 $len(col)$: size of the collection (number of terms);
 $len(d)$: size of the document “d” (number of terms);
 $len(e)$: size of the element “e” (number of terms);
 N_d : total number of documents in the collection;
 N_e : total number of elements in the collection;
 $df(t)$: number of documents containing the term;
 $ef(t)$: number of elements containing the term;

In the following subsections, we give the formulas of all the evaluated models. Note that all formulas are given as the scoring function of element “e” on a single term “t”. For a query with multiple terms, the score of “e” is averaged over its scores on all query terms.

2.1 Vector Space Model

For the vector space model, we choose the basic TFIDF formula as follows:

$$score(e, t) = tf(t, e) \cdot \log \frac{N_d}{df(t)}. \quad (1)$$

$$score(e, t) = tf(t, e) \cdot \log \frac{N_e}{ef(t)}. \quad (2)$$

As elements are of various lengths and distributions while documents are roughly more homogeneous, we use two ways to measure the specialty of term “t”. One is the inverse document frequency (idf) in equation (1) [2], and the other is the inverse element frequency (ief) in equation (2) [3]. Their effects on performance are evaluated in experiments in Section 4.

2.2 Probabilistic Model

The most widely used and highly successful probabilistic model is Okapi BM25 [4], which are given by the following formulas:

$$score(e, t) = \frac{(k_1 + 1) \cdot tf(t, e)}{k_1 \cdot ((1 - b) + b \cdot \frac{len(e)}{avel}) + tf(t, e)} \cdot \log \frac{N_d - df(t) + 0.5}{df(t) + 0.5}. \quad (3)$$

$$score(e, t) = \frac{(k_1 + 1) \cdot tf(t, e)}{k_1 \cdot ((1 - b) + b \cdot \frac{len(e)}{avel}) + tf(t, e)} \cdot \log \frac{N_e - ef(t) + 0.5}{ef(t) + 0.5}. \quad (4)$$

avel is the average length of the elements in the collection, which can be computed from $len(col)/N_e$. As in vector space model, we test both idf and ief cases.

2.3 Language Model

Language modeling is a newly developed and promising approach to information retrieval. The basic idea is to estimate a language model for each document/element, and then rank the document/element by the likelihood of generating the query with the language model. There are different smoothing methods to estimate the language model, i.e. the probability of generating each term [5].

Dirichlet priors smoothing method.

$$score(e, t) = \frac{tf(t, e) + \mu \cdot \frac{tf(t, col)}{len(col)}}{len(e) + \mu}. \quad (5)$$

Jelinek-Mercer smoothing method.

$$score(e, t) = (1 - \lambda) \frac{tf(t, e)}{len(e)} + \lambda \frac{tf(t, col)}{len(col)}. \quad (6)$$

Two-Stage smoothing method.

$$score(e, t) = (1 - \lambda) \frac{tf(t, e) + \mu \cdot \frac{tf(t, d)}{len(d)}}{len(e) + \mu} + \lambda \frac{tf(t, col)}{len(col)}. \quad (7)$$

In the above formula, element “e” is contained in the document “d”. That is, the element language model is first smoothed with the document language model using a Dirichlet prior, and then it is further smoothed with the collection language model using Jelinek-Mercer method [7].

3 Hierarchical Retrieval Models

In the basic retrieval strategy, elements are scored independently. To capture the hierarchical relationship among elements, a common approach is to propagate the scores along the tree, that is, the scores of elements up in the tree are calculated as the weighted combination of scores of its children. The propagation is done recursively from the leaf nodes till the root of the tree [8][9].

Leaf-content vs. full-content. We can use any scoring model presented in Section 2 to give the initial score for each element before propagation. As non-leaf elements will gain scores from its descendants, the initial scores for each element can be based on two options of element content: one is all the terms directly contained in the scored element, referred as *leaf content*; the other is the full content of the element. We evaluate both strategies in Section 4.

Propagation weights. There are many different ways to define the weights of propagating the score of an element to its parent. One basic approach is to assign equal weight to each element, and the accumulated score of an element is calculated as the average of all its children’s accumulated scores as well as its initial score. Another approach is to assign the weights proportional to the lengths of elements. So the propagation weight of each element is equal to the length of the element divided by the length of its parent element. Weights can also be defined to reflect the importance of specific element types or the degree of the dependence between the element and its parent. However, such more sophisticated approaches require some knowledge about the schema of XML documents. In our experiments, we evaluate the first two basic approaches. The first one is referred as *average* and the second one is referred as *length* strategies respectively in this paper.

4 Experiments

We conducted a series of experiments to evaluate the different strategies presented in the previous sections, to gain better knowledge about structured document retrieval.

4.1 Experimental Setup

We implemented all the scoring models and different retrieval strategies inside the Lemur/Indri IR system [10]. It is based on language modeling approaches, and uses the full-content scoring strategy; by default, no hierarchical structure is exploited. We added scoring functions of TFIDF and Okapi models to the system, extended the index with statistics at the element level, and exploited the structure to do score propagation.

We use the data and queries from INEX 2006 as the test bed. The data collection consists of more than 4G bytes of Wikipedia documents. We choose 15 topics from INEX 2006 topics, and test only the CO queries. All the elements in XML documents are indexed, and the index is built with the Krovetz stemmer.

The metrics used in the experiments are the INEX 2007 metrics for focused retrieval tasks, i.e. interpolated precisions at selected recall levels (0.0, 0.01, 0.05, 0.1), and the mean average interpolated precision (MAiP) computed from the interpolated average precisions at 101 recall levels. For each run, the system returns the top 1500 elements. To apply the evaluation measures, overlap in the result list has to be removed first. We adopt the simplest strategy of removing overlap, i.e. just keeping the highest ranked element on each path.

4.2 Results

The experiment results are shown in Table 1 and Table 2. All the presented scoring models under different retrieval strategies are evaluated. For Okapi scoring models, we set the parameters $k_1=1.0$ and $b=0.5$; for the Dirichlet method, the parameter μ is set to be 2500; for the Jenilek-Mercer method, we set parameter $\lambda=0.4$; the parameters

in the two-stage method are set as $\mu=2500$ and $\lambda=0.4$. In the tables, “full-noStruct” denotes the basic approach of scoring each element independently based on its full content; “leaf-struct” and “full-struct” denote the strategies of taking into account the hierarchical structure of XML documents by propagating scores along paths based on the leaf content and full content of an element respectively, while the propagation weights can be set to be proportional to the length of an element---“length” or the average to all children elements---“avg”.

Different scoring models. Among the three classes of scoring models, language modeling approaches, especially the two stage smoothing method, performed better than others both in terms of the MAiP and the iP at 0.01 recall level.

idf vs. ief. As for the TFIDF and Okapi BM25 models, it can be observed that inverse element frequency (ief) had slightly better discriminating power among elements than the inverse document frequency (idf).

Full vs. leaf. According to the measurements of MAiPs, for most scoring models, retrieval strategies based on full content of elements, especially those without considering structural information, performed better than those based on leaf content of elements with score propagation. Two-stage method is an exception however. It performs better using “leaf-struct” strategy. But in terms of the iP at 0.01 recall level, “leaf-struct” strategy performs better than or roughly the same as those based on full content of elements.

Table 1. Mean Average Interpolated Precisions (MAiP) for different strategies.

Scoring Models		full-noStruct	leaf-struct		full-struct	
			length	avg	length	avg
tfidf	idf	0.0934	0.0567	0.0433	0.0698	0.0895
	ief	0.1123	0.0555	0.0359	0.0701	0.1080
okapi	idf	0.0450	0.0361	0.0216	0.0408	0.0318
	ief	0.0479	0.0458	0.0274	0.0539	0.0335
LM	dirichlet	0.1804	0.1039	0.0420	0.0984	0.1715
	jenilek	0.1078	0.0691	0.0631	0.0605	0.0767
	two-stage	0.2001	0.2751	0.2830	0.2385	0.2161

Table 2. Interpolated Precisions (iP) at 0.01 recall level for different strategies.

Scoring Models		full-noStruct	leaf-struct		full-struct	
			length	avg	length	avg
tfidf	idf	0.1832	0.3039	0.3297	0.3245	0.1802
	ief	0.2496	0.3310	0.4163	0.3307	0.2436
okapi	idf	0.4547	0.4415	0.3250	0.4578	0.3258
	ief	0.5084	0.5106	0.3744	0.5282	0.3475
LM	dirichlet	0.4170	0.5107	0.3712	0.3840	0.4581
	jenilek	0.6139	0.5538	0.4269	0.5606	0.5009
	two-stage	0.7216	0.8133	0.8124	0.7914	0.7642

Length vs. average. For propagating scores up in the tree, using length-proportional weights is better than using average weights in most test cases.

5 Conclusions and Future Work

In this paper, we describe our preliminary work on XML retrieval. We did a series of experiments to analyze different retrieval strategies and scoring models in a single framework, attempting to identify the key factors affecting the performance and get better knowledge about the problem. We use INEX as the test bed, and implement the retrieval strategies in the Lemur/Indri system.

At the time of submitting this extended abstract, we are still working on more extensive experiments with much more queries, tuning the parameters in different models, and etc. More detailed presentation and analysis is expected in the final version of this paper.

As our future work, we are going to develop the scoring approach with the following issues in mind:

1. How to score the elements to determine the appropriate portion of the document to return?

2. How to interpret structural conditions or to combine the structural and content statistics in the scoring model?

We are also interested in studying the problem of evaluating top-k queries efficiently.

Acknowledgments. The research work is funded by the National Natural Science Foundation of China under Grant Nos. 60473069, 60496325, and the Key Project of Chinese Ministry of Education under Grant No. 106006.

References

1. N. Fuhr, K. Grossjohann, "XIRQL: a query language for information retrieval in XML documents", SIGIR 2001.
2. D. Carmel, Y.S. Maarek, M. Mandelbrod, et al., "Searching XML documents via XML fragments", SIGIR 2003.
3. T. Grabs, H.-J. Schek, "Flexible Information Retrieval on XML Documents", Intelligent Search on XML data, H. Blanken et al. (Eds.), 2003.
4. M. Theobald, R. Schenkel, G. Wiekum, "An Efficient and Versatile Query Engine for TopX Search", VLDB 2005.
5. C. Zhai, J. Lafferty, "A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval", SIGIR 2001.
6. D. Hiemstra, "Statistical Language Models for Intelligent XML Retrieval", Intelligent Search on XML data, H. Blanken et al. (Eds.), 2003.
7. C. Zhai, J. Lafferty, "Two-Stage Language Models for Information Retrieval", SIGIR 2002.
8. P. Ogilvie, J. Callan, "Hierarchical Language Models for XML Component Retrieval", INEX 2004.
9. P. Ogilvie, J. Callan, "Parameter Estimation for a Simple Hierarchical Generative Model for XML Retrieval", INEX 2005.
10. Lemur/Indri. <http://www.lemurproject.org>.

Indian Statistical Institute at INEX 2007 Adhoc track: VSM Approach

Sukomal Pal and Mandar Mitra

Information Retrieval Lab, CVPR Unit,
Indian Statistical Institute, Kolkata
India
{sukomal_r, mandar}@isical.ac.in

Abstract. This paper describes the work that we did at Indian Statistical Institute towards XML retrieval for INEX 2007. As a continuation of our INEX 2006 work, we applied the Vector Space Model and enhanced our text retrieval system (SMART) to retrieve XML elements against the INEX Adhoc queries. Like last year, we considered Content-Only(CO) queries and submitted two runs for the FOCUSED sub-task. The baseline run does retrieval at the document level; for the second run, we submitted our first attempt at element level retrieval. This run uses a very naive approach and performs poorly, but the relative performance of the baseline run was respectable. Our next step will be to explore ways to improve element-level retrieval.

1 Introduction

Traditional Information Retrieval systems return whole documents in response to queries, but the challenge in XML retrieval is to return the most relevant parts of XML documents which meet the given information need. INEX 2007 [1] marks a paradigm shift as far as retrieval granularity is concerned. This year, arbitrary passages are also permitted as retrievable units, besides the usual XML elements. A retrieved passage can be a sequence of textual content either from within an element or spanning a range of elements. INEX 2007 also classified the adhoc retrieval task into three sub-tasks: a) the FOCUSED task which asks systems to return a ranked list of elements or passages to the user; b) the RELEVANT in CONTEXT task which asks systems to return relevant elements or passages grouped by article; and c) the BEST in CONTEXT task which expects systems to return articles along with one best entry point to the user.

Each of the three subtasks can be based on two different query variants: Content-Only(CO) and Content-And-Structure(CAS) queries. In the CO task, the user poses the query in free text and the retrieval system is supposed to return the most relevant elements/passages. A CAS query can provide explicit or implicit indications about what kind of element the user requires along with a textual query. Thus, a CAS query contains structural hints expressed in XPath [2] along with an *about()* predicate.

Our retrieval approach this year was based on the Vector Space Model which sees both the document and the query as bags of words, and uses their *tf-idf* based weight-vectors to measure the inner product *similarity* between the document and the query. The documents are retrieved and ranked in decreasing order of the similarity-value.

We used the SMART system for our experiments at INEX 2007 and submitted two runs for the *FOCUSED* sub-task of the Adhoc track considering CO queries only. In the following section we describe our approaches for these two runs, and discuss results and further work in Section 3.

2 Approach

To extract the useful parts of the given documents, we shortlisted about thirty tags that contain useful information: `<p>`, `<ip1>`, `<it>`, `<st>`, `<fnm>`, `<snm>`, `<atl>`, `<ti>`, `<p1>`, `<h2a>`, `<h>`, `<wikipedialink>`, `<section>`, `<outsidelink>`, `<td>`, `<body>`, etc. Documents were parsed using the LIBXML2 parser, and only the textual portions included within the selected tags were used for indexing. Similarly, for the topics, we considered only the *title* and *description* fields for indexing, and discarded the *inex-topic*, *castitle* and *narrative* tags. No structural information from either the queries or the documents was used.

The extracted portions of the documents and queries were indexed using single terms and a controlled vocabulary (or pre-defined set) of statistical phrases following Salton's blueprint for automatic indexing [3]. Stopwords were removed in two stages. First, we removed frequently occurring common words (like *know*, *find*, *information*, *want*, *articles*, *looking*, *searching*, *return*, *documents*, *relevant*, *section*, *retrieve*, *related*, *concerning*, etc.) from the INEX topic-sets. Next, words listed in the standard stop-word list included within SMART were removed from both documents and queries. Words were stemmed using a variation of the Lovin's stemmer implemented within SMART. Frequently occurring word bi-grams (loosely referred to as phrases) were also used as indexing units. We used the N-gram Statistics Package (NSP)¹ on the English Wikipedia text corpus and selected the 100,000 most frequent word bi-grams as the list of candidate phrases. Documents and queries were weighted using the *Lnu.ltn* [4] term-weighting formula. For each of 130 adhoc queries(414-543), we retrieved 1500 top-ranked XML documents or non-overlapping elements.

2.1 Baseline Run

For the baseline run, *VSMfb*, we retrieved whole documents only. We had intended to use blind feedback for this run, but ended up inadvertently submitting the results of simple, inner-product similarity based retrieval.

¹ <http://www.d.umn.edu/~tpederse/nsp.html>

2.2 Element-level Run

This year, we also attempted element-level retrieval for the first time. Since Smart does not support the construction of inverted indices at the element-level, we adopted a 2-pass strategy. In the first pass, we retrieved 1500 documents for each query. In the second pass, only the retrieved documents were analysed at the element level, and the best-matching elements constituted the final ranked list.

More specifically, for the first pass, we applied automatic query expansion. To reduce query drift, we first re-ranked the top 50 retrieved documents from the baseline run using proximity constraints and term correlation information [5]. After the reranking step, queries were expanded via blind feedback using the top 20 documents. The expansion parameters are given below:

$$\begin{aligned}\text{number of words} &= 20 \\ \text{number of phrases} &= 5 \\ \text{Rocchio } \alpha &= 4 \\ \text{Rocchio } \beta &= 4 \\ \text{Rocchio } \gamma &= 2.\end{aligned}$$

For each topic, 1500 documents were retrieved using the expanded query. These documents were then parsed using the libXML2 parser, and leaf nodes having textual content were identified. The total set of leaf-level textual elements obtained from the 1500 top-ranked documents were then indexed and compared to the query as before to obtain the final list of 1500 retrieved elements. Since we considered only the leaf-nodes, the retrieved elements are automatically non-overlapping.

3 Results

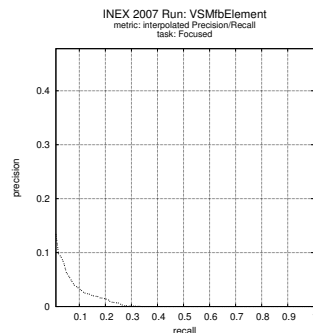
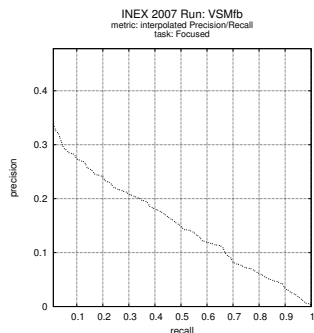
The results reported for the two runs are shown in Table 1. Overall ranks are out of 79 runs, and CO-ranks are out of 58 runs published on the INEX 2007 web-site.

The first run or the baseline, if not satisfactory, was certainly promising. Since this run returns only whole documents, it compares unfavourably with other runs when evaluated using precision-oriented measures such as $P@0.00$ or $P@0.01$, but looks respectable in terms of $P@0.10$ and ends up at 7th position in terms of MAiP. It remains to be seen whether further improvements are achieved when blind feedback is actually used (as originally intended).

The element-level run proved to be a damp squib. In hindsight, this is not very surprising since our present system does not consider elements at intermediate (non-leaf) levels. Leaf nodes are very often too small to contain any meaningful information. However, this needs to be thoroughly investigated. We intend to complete these investigations once we obtain the updated EVALJ package incorporating the new official metrics.

Table 1. Metric: Interpolated Precision/Recall task: FOCUSED, COretrieval unit:Element

Run Id	P@0.00			P@0.01			P@0.05			P@0.10			MAiP		
	Score	Overall rank	CO rank	Score	Overall rank	CO rank	Score	Overall rank	CO rank	Score	Overall rank	CO rank	Score	Overall rank	CO rank
VSMfb	0.3539	50	34	0.3376	39	27	0.2933	28	20	0.2739	22	15	0.1528	7	7
VSMfbElement	0.1684	78	57	0.1345	75	54	0.0632	76	55	0.0326	77	56	0.0110	77	56
BEST run		0.4780			0.4259			0.3482			0.3238			0.1804	



4 Conclusion

This was our second year at INEX. Our main objective this year was to incorporate element-level retrieval within Smart. We started with retrieval only at the leaf-level, but this obviously needs to be extended to enable retrieval of elements at any level within the XML tree. We will be particularly interested in effective term-weighting and normalization strategies for element retrieval. We hope this will be an exciting exercise which we plan to continue in the coming years.

References

1. INEX, Initiative for the Evaluation of XML Retrieval. (2007) <http://inex.is.informatik.uni-duisburg.de/2007>.
2. W3C: XPath-XML Path Language(XPath) Version 1.0 <http://www.w3.org/TR/xpath>.
3. Salton, G.: A Blueprint for Automatic Indexing. ACM SIGIR Forum **16**(2) (Fall 1981) 22–38
4. Buckley, C., Singhal, A., Mitra, M.: Using Query Zoning and Correlation within SMART: TREC5. In Voorhees, E., Harman, D., eds.: Proc. Fifth Text Retrieval Conference (TREC-5). Volume NIST Special Publication 500-238. (1997)
5. Mitra, M., Singhal, A., Buckley, C.: Improving automatic query expansion. In: SIGIR 98, Melbourne, Australia, ACM (1998) 206–214

Using Topic Models in XML Retrieval

Fang Huang

School of Computing, The Robert Gordon University, Scotland
f.huang@rgu.ac.uk

Abstract. This paper describes Robert Gordon University’s experiments of using probabilistic topic models in the INEX 2007 ad hoc track. We looked at a recent statistical model called Latent Dirichlet Allocation[1], and explored how it could be applied to XML retrieval.

1 Introduction

XML retrieval aims to return relevant document components (e.g., XML elements) rather than whole documents. A variety of approaches have been exploited to score XML elements’ relevance to a user’s query[4, 6]. In this work, we experimented on how the topic model, a recent unsupervised learning technique, can be use in XML retrieval. The specific model at the heart of this study is the Latent Dirichlet Allocation (LDA) model[1], a hierarchical Bayesian model employed previously to analyze text corpora and to annotate images[2]. The basic idea of a topic model is that documents are mixtures of topics, where a topic is a probability distribution over words. We used LDA to discover topics in the Wikipedia collection. Documents, XML elements, user queries and words were all represented as mixtures of probabilistic topics, and were compared to each other to calculate their relevance.

The remainder of this paper is organized as follows: Section 2 briefly introduces the LDA model and explains how LDA is used to model the relationships of documents in the Wikipedia collection. Our experimental setup is described in section 3. In section 4, we discuss our submitted runs and our results in the INEX official evaluation. The final part, section 5, concludes with a discussion and possible directions for future work.

2 Using the Latent Dirichlet Allocation Model on Wikipedia Collection

Latent dirichlet allocation[1] is a generative probabilistic model for collections of discrete data such as text corpora. It assumes that each word of each document is generated by one of several “topics”; each topic is associated with a different conditional distribution over a fixed vocabulary. The same set of topics is used to generate the entire set of documents in a collection but each document reflects these topics with different relative proportions. Specifically, for a collection

consists of words $w = w_1, w_2, \dots, w_n$, where $w_i (1 \leq i \leq n)$ belongs to some documents, as in a word-document co-occurrence matrix. For each document d_i , we have a multinomial distribution over k topics, with parameters $\theta^{(d_i)}$, so for a word in document d_i , $P(z_i = j) = \theta_j^{(d_i)}$. The $j^{th} (1 \leq j \leq k)$ topic is represented by a multinomial distribution over the n words in the vocabulary, with parameters $\alpha^{(j)}$, so $P(w_i | z_i = j) = \alpha_{w_i}^{(j)}$. A Dirichlet prior is introduced for the topic distribution with parameters $\alpha_i (1 \leq i \leq k)$:

$$p(\theta | \alpha) = \frac{\Gamma(\sum_{i=1}^k \alpha_i)}{\prod_{i=1}^k \Gamma(\alpha_i)} \theta_1^{\alpha_1 - 1} \dots \theta_k^{\alpha_k - 1} \quad (1)$$

where the parameter α is a k -vector with components $\alpha_i > 0$, and $\Gamma(x)$ is the Gamma function. Thus, the probability of observing a document d_i is:

$$p(d_i | \alpha, \beta) = \int p(\theta | \alpha) \left(\prod_{n=1}^N \sum_{z_n} p(z_n | \theta) p(w_n | z_n, \beta) \right) d\theta \quad (2)$$

where document d_i contains N words $w_n (1 \leq n \leq N)$. The number of parameters to estimate in this model is k parameters for the Dirichlet distribution and $n - 1$ parameters for each of the k topic models. The estimation of parameters is done by variational inference algorithms.

We applied the LDA on the Wikipedia collection. All texts in the collection were lower-cased, stop-words removed using a stop-word list. After the pre-processing, each document was represented in a form of a word frequency vector. A Gibbs sampling algorithm was then used to estimate parameters of LDA in our implementation. As the LDA model assumes that the dimensionality of the Dirichlet distribution (and thus the dimensionality of the topic variable z) is known and fixed, two topic models were learned in our experiments. The dimensionalities of them were 200 and 50, respectively. The content of words, documents, any XML elements, and user queries were then represented as vectors of topic probabilities.

3 Experimental Setup

We created inverted indexes of the collection using Lucene[3]. Indexes were word-based. All texts were lower-cased, stop-words removed using a stop-word list, but no stemming. For each XML element, all text nested inside it was indexed. We considered paragraph elements to be the lowest possible level of granularity of a retrieval unit. And indexed text segments consisting of paragraph elements and of elements containing at least one paragraph element as a descendant element. For the remainder of the paper, when we refer to the XML elements considered in our investigation, we mean the segments that correspond to paragraph elements and to their ancestors.

Our queries were created using terms only in the <title> parts of topics. Like the index, queries were word-based. The text was lower-cased and stop-words

were removed, but no stemming was applied. ‘+’, ‘-’ and quotes in queries were simply removed. The modifiers “and” and “or” were ignored.

As described in section 2, we learned two topic models. The dimensionalities (number of topics) of them were 200 and 50, respectively. For each of the topic models, XML elements, and user queries were represented as vectors in the topic space. The similarity of a user query and an XML element were determined by cosine similarity between the two corresponding vectors.

4 Submissions and Results

In this section, we describe the runs submitted to the INEX 2007 ad-hoc track. We totally submitted 6 runs based on topic models, two for each of the 3 tasks (Focused, Relevant-in-Context, and Best-in-Context). Table 1 lists a brief description of the runs. In our experiments, the top ranked elements were returned

Table 1. Ad-hoc runs based on topic models

RunID	Approach	INEX task
Focused-TM-1	topic model with 200 topics	Focused
Focused-LDA	topic model with 50 topics	Focused
RelevantInContent-TM-1	topic model with 200 topics	Relevant-in-Context
RelevantInContent-LDA	topic model with 50 topics	Relevant-in-Context
BestInContext-TM-1	topic model with 200 topics	Best-in-Context
BestInContext-LDA	topic model with 50 topics	Best-in-Context

for further processing. For the Focused Task, overlaps were removed by applying a post-filtering on the retrieved ranked list by selecting the highest scored element from each of the paths. In case of two overlapping elements with the same relevance score, the child element was selected. For the Relevant-in-Context task, we simply took the results for the Focused task, reordered the elements in the list such that results from the same article were grouped together. In the Best-in-Context task, the element with the highest score was chosen for each document. If there are two or more elements with the same highest score, the one that appeared first in the original document was selected. For each of the runs, the top 1,500 ranked elements were returned as answers.

Table 2 lists the result of our Focused runs in the INEX 2007 official evaluation, where $iP@j$, $j \in [0.00, 0.01, 0.05, 0.10]$, is the interpolated precision at j recall level cutoffs, and MAip is the mean average interpolated precision. Details of the evaluation metrics can be found in [5]. Performance of Focused-LDA is relatively poor. As we used 50 topics to model the collection in this run, the result prompts us that 50 topics are not enough to describe the whole collection. This is reasonable, as the Wikipedia collection we used is a large heterogeneous corpus

Table 2. Results of Focused runs

RunID	$iP@0.00$	$iP@0.01$	$iP@0.05$	$iP@0.10$	MAiP
Focused-TM-1	0.4079	0.3586	0.2845	0.2500	0.0945
Focused-LDA	0.0276	0.0171	0.0137	0.0116	0.0041

containing 659,388 documents with a large number of various topics. Furthermore, when we increased the number of topics, in Focused-TM-1 (which is based on a topic model with 200 topics), the performance is significantly improved.

Table 3. Results of Relevant-in-Context runs

RunID	gP[5]	gP[10]	gP[25]	gP[50]	MAgP
RelevantInContext-TM-1	0.1546	0.1357	0.0993	0.0778	0.0730
RelevantInContext-LDA	0.0034	0.0033	0.0060	0.0058	0.0048

Table 4. Results of Best-in-Context runs

RunID	gP[5]	gP[10]	gP[25]	gP[50]	MAgP
BestInContext-TM-1	0.2244	0.2115	0.1773	0.1382	0.1290
BestInContext-LDA	0.0136	0.0098	0.0123	0.0105	0.0099

Evaluation results of Relevant-in-Context runs and Best-in-Context runs are listed in table 3 and table 4, respectively. Here, $g[r]$, $r \in [5, 10, 25, 50]$, is non-interpolated generalized precision at r ranks; and MAgP is non-interpolated mean average generalized precision. Again, results show that runs based on the topic model with 200 topics (i.e., RelevantInContext-TM-1, BestInContext-TM-1) perform significantly better than runs based on the topic model with 50 topics (i.e., RelevantInContext-LDA, BestInContext-LDA). This is not surprising as we explained above. It indicates that the collection is much better described with 200 topics than 50 topics. As the topics dimensionalities were randomly set as 50 and 200 in our experiments, we expect that retrieval results will be significantly improved given that we know the actually number of topic underlying the collection.

5 Conclusions

We have presented, in this paper, our experiments of using topic models for the INEX 2007 evaluation campaign. We participated in all the three ad hoc track tasks. The LDA model is used to detect topics underlying the collection. We learned two topic models with topic numbers of 50 and 200, respectively. The evaluation results showed that runs based on the topic model with 200 topics achieved significantly better performances than runs based on a lower-dimensional topic space (50 topics). One assumption of the LDA model is that the dimensionality of the topic is known and fixed. In our experiments, dimensionalities were randomly set as 50 and 200. We expect the results will be better if we learn the number of topics underlying the collection. Our future work will focus on integrating text mining techniques to learn the number of topics before applying LDA model.

6 Acknowledgments

The Lucene-based indexer used this year was partly based on the indexing code developed for RGU INEX'06 by Stuart Watt and Malcolm Clark.

References

1. Blei, D., Ng, A., Jordan, M.: Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993-1022, 2003.
2. Blei, D., Jordan, M.: Modeling annotated data. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* ACM press; 2003: 127-134.
3. Lucene. The Lucene search engine, 2005. <http://jakarta.apache.org/lucene>
4. Mass, Y., Mandelbrod, M.: Using the inex environment as a test bed for various user models for XML retrieval. In: Fuhr, N., Lalmas, M., Malik, S., Kazai, G. (eds.) *INEX2005*. LNCS, vol. 3977, pp. 187-195. Springer, Haidelberg (2006)
5. Pehcevski, J., Kamps, J., Kazai, G., Lalmas, M., Ogilvie, P., Piwowarski, B., Robertson, S.: *INEX 2007 Evaluation Measures*. INEX2007.
6. Sigurbjornsson B., Kamps J. and de Rijke M. An element-based approach to XML retrieval. *INEX 2003 Workshop Proceedings, 2004*

TopX @ INEX 2007

Andreas Broschart¹, Ralf Schenkel¹, Martin Theobald², and Gerhard Weikum¹

¹ Max-Planck-Institut für Informatik, Saarbrücken, Germany

<http://www.mpi-inf.mpg.de/departments/d5/>

{abrosch,schenkel,weikum}@mpi-inf.mpg.de

² Stanford University

<http://infolab.stanford.edu/>

theobald@stanford.edu

Abstract. This paper describes the setup and results of the Max-Planck-Institut für Informatik's contributions for the INEX 2007 AdHoc Track task. The runs were produced with TopX, a search engine for ranked retrieval of XML data that supports a probabilistic scoring model for full-text content conditions and tag-term combinations, path conditions as exact or relaxable constraints, and ontology-based relaxation of terms and tag names.

1 System Overview

TopX [2, 5] aims to bridge the fields of database systems (DB) and information retrieval (IR). From a DB viewpoint, it provides an efficient algorithmic basis for top- k query processing over multidimensional datasets, ranging from structured data such as product catalogs (e.g., bookstores, real estate, movies, etc.) to unstructured text documents (with keywords or stemmed terms defining the feature space) and semistructured XML data in between. From an IR viewpoint, TopX provides ranked retrieval based on a relevance scoring function, with support for flexible combinations of mandatory and optional conditions as well as text predicates such as phrases, negations, etc. TopX combines these two aspects into a unified framework and software system, with emphasis on XML ranked retrieval.

Figure 1 depicts the main components of the TopX system. The *Indexer* parses and analyzes the document collection and builds the index structures for efficient lookups of tags, content terms, phrases, structural patterns, etc. TopX currently uses Oracle10g as a storage system, but the JDBC interface would easily allow other relational backends, too. An *Ontology* component manages optional ontologies with various kinds of semantic relationships among concepts and statistical weighting of relationship strengths.

At query run-time, the *Core Query Processor* decomposes queries (which can be either NEXI or XPath Full-Text) and invokes the top- k algorithms. It maintains intermediate top- k results and candidate items in a priority queue, and it schedules accesses on the precomputed index lists in a multi-threaded architecture. Several advanced components provide means for run-time acceleration:

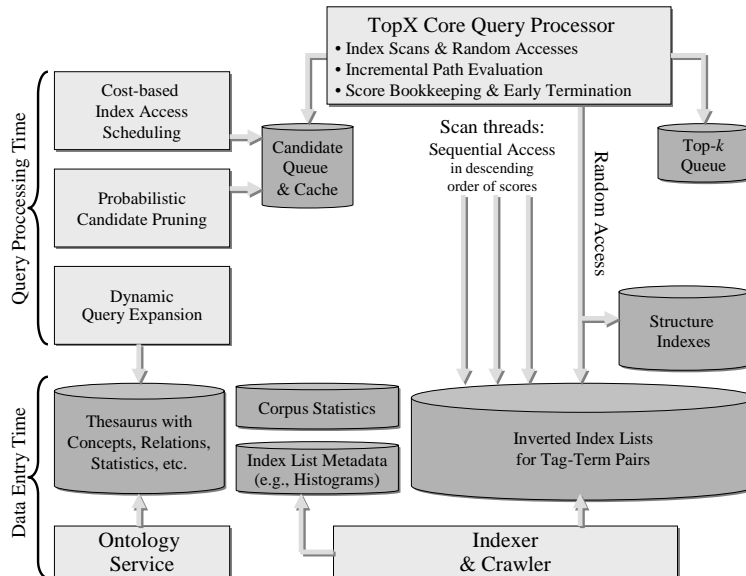


Fig. 1. TopX architecture.

- The *Probabilistic Candidate Pruning* component [6] allows TopX to drop candidates that are unlikely to qualify for the top- k results at an early stage, with a controllable loss and probabilistic result guarantees.
- The *Index Access Scheduler* [1] provides a suite of scheduling strategies for sorted and random accesses to index entries.
- The *Incremental Path Evaluation* uses additional cost models to decide when to evaluate structural conditions like XML path conditions, based on specialized indexes for XML structure.
- The *Dynamic Query Expansion* component [4] maps the query keywords and/or tags to concepts in the available ontology and incrementally generates query expansion candidates.

As our INEX runs focused on result quality, not on efficiency, they were produced using only the Index Access Scheduler and Incremental Path Evaluation. Topx supports three different front-ends: a servlet with an HTML end-user interface (that was used for the topic development of INEX 2006 and 2007), a Web Service with a SOAP interface (that was used by the Interactive track), and as a Java API (that was used to generate our runs).

2 Data Model and Scoring

We refer the reader to [2] for a thorough discussion of the scoring model. This section shortly reviews important concepts.

2.1 Data Model

We consider a simplified XML data model, where idref/XLink/XPointer links are disregarded. Thus every document forms a tree of nodes, each with a *tag* and a related *content*. We treat attributes nodes as children of the corresponding element node. The content of a node is either a text string or it is empty. With each node, we associate its *full-content* which is defined as the concatenation of the text contents of all the node’s descendants in document order.

2.2 Content Scores

For content scores we make use of element-specific statistics that view the full-content of each element as a bag of words:

- 1) the *full-content term frequency*, $ftf(t, n)$, of term t in node n , which is the number of occurrences of t in the full-content of n ;
- 2) the *tag frequency*, N_A , of tag A , which is the number of nodes with tag A in the entire corpus;
- 3) the *element frequency*, $ef_A(t)$, of term t with regard to tag A , which is the number of nodes with tag A that contain t in their full-contents in the entire corpus.

The score of an element e with tag A with respect to a content condition of the form $T[\text{about}(\cdot, \tau)]$ (where T is either e ’s tag A or the tag wildcard operator $*$) is then computed by the following BM25-inspired formula:

$$\text{score}(e, T[\text{about}(\cdot, \tau)]) = \frac{(k_1 + 1) ftf(t, e)}{K + ftf(t, n)} \cdot \log \left(\frac{N_A - ef_A(t) + 0.5}{ef_A(t) + 0.5} \right) \quad (1)$$

$$\text{with } K = k_1 \left((1 - b) + b \frac{\sum_{t'} ftf(t', e)}{\text{avg}\{\sum_{t'} ftf(t', e') \mid e' \text{ with tag } A\}} \right)$$

For a query content condition with multiple terms, the score of an element satisfying the tag constraint is computed as the sum of the element’s content scores for the corresponding content conditions, i.e.:

$$\text{score}(e, T[\text{about}(\cdot, t_1 \dots t_m)]) = \sum_{i=1}^m \text{score}(e, T[\text{about}(\cdot, t_i)]) \quad (2)$$

TopX provides the option to evaluate queries either in conjunctive mode or in “andish” mode. In the first case, all terms (and, for content-and-structure queries, all structural conditions) must be met by a result candidate, but still different matches yield different scores. In the second case, a node is already considered a match if it satisfies at least one content condition in the target dimension specified in the NEXI/XPath query.

Orthogonally to this, TopX can be configured to return two different granularities as results: in *document mode*, TopX returns the best documents for a query, whereas in *element mode*, the best target elements are returned, which may include several elements from the same document. For the INEX experiments in this year’s AdHoc track, we used element mode with some additional postprocessing for the Focused task, and document mode for the RelevantInContext and BestInContext tasks.

2.3 Structural Scores

Given a query with structural and content conditions, we transitively expand all structural query dependencies. For example, in the query `//A//B//C[about(. , t)]` an element with tag C has to be a descendant of both A and B elements. Branching path expressions can be expressed analogously. This process yields a *directed acyclic graph* (DAG) with tag-term conditions as leaves, tag conditions as inner nodes, and all transitively expanded descendant relations as edges.

Our structural scoring model essentially counts the number of navigational (i.e., tag-only) conditions that are completely satisfied by a result candidate and assigns a small and constant score mass c for every such condition that is matched. This structural score mass is combined with the content scores. In our setup we have set $c = 1$, whereas content scores are normalized to $[0, 1]$, i.e., we emphasize the structural parts.

3 AdHoc Track Results

As the recent development of TopX has focused on efficiency issues, its scoring function used to rank results did not change from the experiments reported last year [3]. The discussion of the experimental results in this section therefore focuses on differences introduced by the new metrics used for INEX 2007.

For each subtask, we submitted the following four runs:

- `CO-{subtask}-all`: a CO run that considered the terms in the title of a topic without phrases and negations, allowing all tags for results.
- `CO-{subtask}-ex-all`: a CO run that considered terms as well as phrases and negations (so-called *expensive predicates*), again without limiting tags of results.
- `CAS-{subtask}-all`: a CAS run that considered the castitle of a topic if it was available, and the title otherwise. The target tag was evaluated strictly, whereas support conditions were optional; phrases and negations were ignored.
- `CAS-{subtask}-ex-all`: a CAS run that additionally considered phrases and negations.

3.1 Focused Task

Our runs for the focused task were produced by first producing a run with all results (corresponding to the *Thorough* task in previous years) and then postprocessing the run to remove any overlap. For each such run, we kept an element e if there was no other element e' from the same document in the run that had a higher score than e and had a path that overlapped with e' 's path. This simple, syntactic postprocessing yielded good results for the CAS runs (shown in Table 1). Especially for the early recall levels, TopX performed extremely well with peak rank 2 in the official result. Interestingly, the CAS run that considered phrases and negation did slightly worse than its counterpart without expensive predicates, whereas the CO run with phrases and negation did better than the plain CO run. Compared to 2006, the results are surprising as our CO runs were much better than our CAS runs then; we assume that limiting the result tags for CO queries to 'article', 'section' and 'p' as we did in 2006 would have helped to improve the CO results.

run	iP[0.00]	iP[0.01]	iP[0.05]	iP[0.10]	MAiP
TOPX-CAS-Focused-all	0.4744 (2)	0.4149 (2)	0.3211 (16)	0.2902 (17)	0.1115 (28)
TOPX-CAS-Focused-ex-all	0.4364 (9)	0.3938 (7)	0.2981 (25)	0.2640 (25)	0.1036 (30)
TOPX-CO-Focused-all	0.4200 (17)	0.3621 (28)	0.2848 (32)	0.2549 (31)	0.1010 (32)
TOPX-CO-Focused-ex-all	0.4379 (8)	0.3758 (23)	0.3001 (23)	0.2709 (23)	0.1021 (31)

Table 1. Results for the Focused Task: iterpolated precision at different recall levels (ranks are in parentheses) and mean average interpolated precision

3.2 RelevantInContext Task

To produce the runs for the RelevantInContext task, we ran TopX in document mode. This yielded a list of documents ordered by the highest score of any element within the document, together with a list of elements and their scores for each document.

The results (Table 2) are reasonably good for CAS queries with peak rank of 12 at 25 documents. For CO queries, results are much worse than 2006; again we attribute this to the fact that we did not limit the tags of result elements.

3.3 BestInContext Task

To compute the best entry point for a document, we postprocessed the RelevantInContext runs by simply selecting the element with highest score from each document and ordered them by score. The results (Table 3) show that this did not work as well as 2006, with a peak rank of 25 this year (compared to a peak rank of 1 for 2006). Especially CO runs performed much worse than

run	gP[5]	gP[10]	gP[25]	gP[50]	MAgP
TOPX-CO-all-RIC	0.1393 (30)	0.1261 (27)	0.0930 (28)	0.0740 (26)	0.0710 (29)
TOPX-CO-ex-all-RIC	0.1491 (26)	0.1252 (28)	0.0890 (31)	0.0701 (30)	0.0747 (23)
TOPX-CAS-RIC	0.1654 (19)	0.1436 (16)	0.1111 (12)	0.0784 (22)	0.0735 (24)
TOPX-CAS-ex-RIC	0.1270 (35)	0.1207 (31)	0.0924 (29)	0.0664 (34)	0.0664 (31)

Table 2. Results for the RelevantInContext Task: generalized precision/recall at different ranks and mean average generalized precision (ranks are in parentheses)

expected in general, even though they performed better than our CAS runs or mean average generalized precision. We attribute this to the fact that we evaluated target tags strictly in CAS runs, so we limited our choice of best entry points to elements with these tags.

run	gP[5]	gP[10]	gP[25]	gP[50]	MAgP
TOPX-CO-all-BIC	0.2039 (44)	0.2060 (42)	0.1729 (38)	0.1320 (37)	0.1326 (32)
TOPX-CO-ex-all-BIC	0.2097 (42)	0.1936 (43)	0.1637 (42)	0.12461 (39)	0.1299 (33)
TOPX-CAS-BIC	0.2604 (25)	0.2309 (25)	0.1892 (28)	0.1330 (36)	0.1225 (37)
TOPX-CAS-ex-BIC	0.2368 (35)	0.2197 (39)	0.1800 (32)	0.1294 (38)	0.1153 (38)

Table 3. Results for the BestInContext Task: generalized precision/recall at different ranks and mean average generalized precision (ranks are in parentheses)

4 Conclusion

This paper the results of the runs produced for the INEX 2007 AdHoc Track with the TopX search engine. This year, runs using CAS topics performed better than runs with CO topics, and TopX performed especially well for the Focused task. We need to further investigate why the results for CO runs were not as good as expected, especially compared to the results from INEX 2006.

References

1. Holger Bast, Debapriyo Majumdar, Martin Theobald, Ralf Schenkel, and Gerhard Weikum. IO-Top- k : Index-optimized top- k query processing. In *Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB 2006)*, pages 475–486, 2006.
2. Martin Theobald, Holger Bast, Debapriyo Majumdar, Ralf Schenkel, and Gerhard Weikum. Topx: efficient and versatile top- k query processing for semistructured data. *VLDB J.*, accepted for publication, 2008.

3. Martin Theobald, Andreas Broschart, Ralf Schenkel, Silvana Solomon, and Gerhard Weikum. Topx – adhoc track and feedback task. In *Proceedings of the 5th International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX 2006)*, pages 233–242, 2006.
4. Martin Theobald, Ralf Schenkel, and Gerhard Weikum. Efficient and self-tuning incremental query expansion for top- k query processing. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 242–249, 2005.
5. Martin Theobald, Ralf Schenkel, and Gerhard Weikum. An efficient and versatile query engine for TopX search. In *Proceedings of the 31st International Conference on Very Large Data Bases (VLDB 2005)*, pages 625–636, 2005.
6. Martin Theobald, Gerhard Weikum, and Ralf Schenkel. Top- k query evaluation with probabilistic guarantees. In *Proceedings of the 30th International Conference on Very Large Data Bases (VLDB 2004)*, pages 648–659, 2004.

LIG at INEX 2007 Ad Hoc Track : Using Collectionlinks as Context

Delphine Verbyst¹ and Philippe Mulhem²

¹ LIG - Université Joseph Fourier, Grenoble, France

Delphine.Verbyst@imag.fr

² LIG - CNRS, Grenoble, France

Philippe.Mulhem@imag.fr

Abstract. We present in this paper the work of the Information Retrieval Modeling Group (MRIM) of the Computer Science Laboratory of Grenoble (LIG) at the INEX 2007 Ad Hoc Track. We study here the impact of non structural relations between structured document elements (doxels) on structured documents retrieval. We use existing links between doxels of the collection, encoded with the *collectionlink* tag, to integrate link and content aspects. We characterize the relation induced by the *collectionlink* tag with relative exhaustivity and specificity scores. As a consequence, the matching process is based on doxels content and these features. Results of experiments on the test collection are presented. Runs using non structural links overperform a baseline without such links.

1 Introduction

This paper describes the approach used for the Ad Hoc Track of the INEX 2007 competition. Our goal here is to show that the use of non structural links can increase the quality of the results provided by an information retrieval system on XML documents. We consider that handling links between documents in a smart way may help an information retrieval system, not only to provide better results, but also to organize the results in a way to overcome the usual simple list of documents. For INEX 2007, we only show that our approach impacts in a positive way the quality of the results provided.

The use of non structural links, such as Web links or similarity links has been studied in the past. Well known algorithms such as Pagerank [1] or HITS [3] do not integrate in a seamless way the links in the matching process. Savoy, in [6], showed that the use of non structural links may provide good results, without qualifying the strength of the inter-relations. In [7], Smucker and Allan show that similarity links may help navigation in the result space. We want, with the work described here, to go further in this direction.

In the following, the non structural relations between doxels will be referred to as the *context* of the doxels. Our assumption is that document parts are not only relevant because of their content, but also because they are related to other

document parts that answer the query. In some way, we revisit the *Cluster Hypothesis* of van Rijsbergen [8], by considering that the relevance value of each document is impacted by the relevance values of related documents.

In our proposal, we first build inter-relations between doxels, and then characterize these relations using relative exhaustivity and specificity at indexing time. These elements are used later on by the matching process.

The nine officially submitted runs by the LIG for the Ad Hoc track integrate such non structural links. For each of the three tasks (Focused, Relevant in Context, Best in Context) a baseline without using such links was submitted. Taking into account the non structural links outperforms consistently this baseline.

The rest of this paper is organized as follows: we describe the links that were used in our experiments in part 2, the doxel space is described in detail in section 3, in which we propose a document model using the context. Section 4 introduces our *matching in context* process. Results of the INEX 2007 Ad Hoc track are presented in Section 5.

2 Choice of Collectionlinks

The idea of considering neighbours was first proposed in [9], in order to facilitate the exploration of the result space by selecting the relevant doxels, and by indicating potential good neighbours to access from one doxel. For this task, the 4 Nearest Neighbours were computed.

The INEX 2007 collection contains several links between documents, like *unknownlinks*, *language links* and *outsidelinks* for instance. We only considered existing relations between doxels with the *collectionlink* tag, because these links denote links inside the collection. Such links have several attributes, but the important attribute for use here is *xlink : href* that indicates the target of the link. We notice that the targets of such links are only whole documents, and not documents parts; this aspect may negatively impact our expectations compared to our model that supports documents parts as targets. The table 1 shows these relations, with a first document D_1 (file 288042.xml) about “Croquembouche” and a second document D_2 (file 1502304.xml) about “Choux pastry”. The third *collectionlink* tag in D_1 links D_1 to D_2 and also ensures a direct proximity between doxels. For our runs, we only considered :

- for each leaf doxel d : the 4 first collectionlinks of d ,
- for non-leaf doxels d' : the union of 4 first collection links of its leaf doxels direct or indirect components

Overall, there are 17 013 512 collectionlinks in the INEX 2007 collection, and with the restriction above we take into account 12 352 989 of them.

Document D_1 :

```
<article>
<name id="288042">Croquembouche</name>
...
<body>A
<emph3>croquembouche</emph3>is a
<collectionlink ... xlink:href="10581.xml">French</collectionlink>
<collectionlink ... xlink:href="57572.xml">cake</collectionlink>
consisting of a conical heap of cream-filled
<collectionlink ... xlink:href="1502304.xml">choux</collectionlink>
buns bound together with a brittle
<collectionlink ... xlink:href="64085.xml">caramel</collectionlink>
sauce, and usually decorated with ribbons or spun sugar.
...
</body>
</article>
```

Document D_2 :

```
<article>
<name id="1502304">Choux pastry</name>
...
<body>
<emph3>Choux pastry</emph3>
<emph2>(pte choux)</emph2>is a form of light
<collectionlink ... xlink:href="67062.xml">pastry</collectionlink>
used to make
<collectionlink ... xlink:href="697505.xml">profiterole</collectionlink>
s or
<collectionlink ... xlink:href="1980219.xml">eclair</collectionlink>
s. Its
<collectionlink ... xlink:href="198059.xml">raising agent</collectionlink>
is the high water content, which boils during cooking, puffing
out the pastry.
...
</body>
</article>
```

Table 1. Collectionlinks in articles

3 Doxel space

3.1 Doxel content

The representation of the content of doxel d_i is a vector generated from a usual vector space model using the whole content of the doxel: $d_i = (w_{i,1}, \dots, w_{i,k})$. Such a representation has proved to give good results for structured document retrieval [2]. The weighting scheme retained is a simple *tf.idf*, with *idf* based on the whole corpus and with the following normalizations: the *tf* is normalized by the max of the *tf* of each doxel, and the *idf* is log-based, according to the document collection frequency. To avoid an unmanageable quantity of doxels, we kept only doxels having the following tags: article, p, collectionlink, title, section, item. The reason for using only these elements was because, except for the collectionlinks, we assume that the text content for these doxels are not too small. The overall number of doxels considered by us here is 29 291 417.

3.2 Doxel context

Let's consider the two linked by *collectionlink* structured documents D_1 and D_2 proposed in table 1, they share *apriori* information. If a user looks for all the information about "croquembouche", the system should indicate that the link above is a relevant part of the query result. If the user only wants to have general informations about "croquembouche", D_1 is highly relevant, D_2 is less relevant, and moreover, the system should indicate that the link between D_1 and D_2 is not interesting for this query result. To characterize the relations between doxels, we propose to define relative exhaustivity and relative specificity between doxels. These features are inspired from the definitions of specificity and exhaustivity proposed at INEX 2005 [4]. Consider a non-compositional relation from the doxels d_1 to the doxel d_2 :

- The relative specificity of this relation, noted $Spe(d_1, d_2)$, denotes the extent to which d_2 focuses on the topics of d_1 . For instance, if d_2 deals only with elements from d_1 , then $Spe(d_1, d_2)$ should be close to 1.
- The relative exhaustivity of this relation, noted $Exh(d_1, d_2)$, denotes the extent to which d_2 deals with all the topics of d_1 . For instance, if d_2 discusses all the elements of d_1 , then $Exh(d_1, d_2)$ should be close to 1.

The values of these features are in $[0, 1]$. We could think that these features behave in an opposite way: when $Spe(d_1, d_2)$ is high, then $Exh(d_1, d_2)$ is low, and vice versa.

Relative specificity and relative exhaustivity between two doxels are extensions of the overlap function [5] of the index of d_1 and d_2 : these values reflect the amount of overlap between the source and target of the relation. We define relative specificity and relative exhaustivity on the basis of the non normalized doxel vectors $w_{1,i}$ and $w_{2,i}$ (respectively for d_1 and d_2) as follows.

We estimate values of the exhaustivity and the specificity of d_1 and d_2 , based on a vector where weights are *tf.idf*

$$Exh(d_1, d_2) = \frac{\sum_i w_{1,i} \cdot w_{2,i}}{\sum_i w_{\oplus 1/w_{2,i}}^2} \quad (1)$$

$$Spe(d_1, d_2) = \frac{\sum_i w_{1,i} \cdot w_{2,i}}{\sum_i w_{\oplus 2/w_{1,i}}^2} \quad (2)$$

$$\text{where: } w_{\oplus m/w_{n,i}} = \begin{cases} w_{m,i} & \text{if } w_{n,i} \leq 1 \\ \sqrt{w_{m,i} \cdot w_{n,i}} & \text{otherwise.} \end{cases}$$

$w_{\oplus m/w_{n,i}}$ ensures that the scores are in $[0, 1]$.

4 Matching in context model

As we have characterized the doxel context, the matching process should return doxels relevant to the user’s information needs regarding both content and structure aspects, and considering the context of each relevant doxel.

We define the matching function as a linear combination of a standard matching result without context and a matching result based on relative specificity and exhaustivity. The relevant status value $RSV(d, q)$ for a given doxel d and a given query q is thus given by:

$$RSV(d, q) = \alpha * RSV_{content}(d, q) + (1 - \alpha) * RSV_{context}(d, q), \quad (3)$$

where $\alpha \in [0, 1]$ is experimentally fixed, $RSV_{content}(d, q)$ is the score without considering the set of neighbours \mathcal{V}_d of d (i.e. cosine similarity) and

$$RSV_{context}(d, q) = \sum_{d' \in \mathcal{V}_d} \frac{\beta * Exh(d, d') + (1 - \beta) * Spe(d, d')}{|\mathcal{V}_d|} RSV_{content}(d', q) \quad (4)$$

where $\beta \in [0, 1]$ is used to privilege exhaustivity or specificity.

The matching in context model computes scores with both content and context dimensions to complete our model.

5 Experiments and results

The INEX 2007 Adhoc track consists of three retrieval tasks: the Focused Task, the Relevant In Context Task, and the Best In Context Task. We submitted 3 runs for each of these tasks. For all these runs, we used only the *title* of the

INEX 2007 queries as input for our system: we removed the words prefixed by a ‘-’ character, and we did not consider the indicators for phrase search. The vocabulary used for the official runs is quite small (39 000 terms).

First of all, we have experimented our system with INEX 2006 collection to fix α and β parameters (see above). The best results were achieved with a higher value for the exhaustivity than for the specificity. As a consequence, we decide to fix $\alpha = 0.75$ and $\beta = 0.75$ for our expected best results.

5.1 Focused Task

The INEX 2007 Focused Task is dedicated to find the most focused results that satisfy a information need, without returning “overlapping” elements. In our focused task, we experiment with two different rankings.

For the first run, the “default” one, namely *LIG.075075_FOC_FOC* with $\lambda = 0.75$ and $\beta = 0.75$, we rank the result based on matching in context proposed in section 4; overlap is removed by applying a post-processing.

For the second run, we choose to use the results of the Relevant In Context Task to produce our Focused Task results : relevant doxels are ranked by article, and we decide to score the doxels with the score of each corresponding article and list them according to their position in the document, and removing overlapping doxels. This run is called *LIG.075075_FOC_RIC*, and we set $\lambda = 0.75$ and $\beta = 0.75$

The last run, namely *LIG.1000_FOC_RIC* is a baseline run. It is similar to the second run with $\lambda = 1.0$ and $\beta = 0.0$.

We present our results for the focused task in Table 2 showing precision values at given percentages of recall, and in Figure 1 showing the generalized precision/recall curve. These results show that runs based on Relevant In Context approach outperforms the “default” Focused Task run, *LIG.075075_FOC_FOC*: after checking the code, we found a bug that leads to incorrect paths for the doxels, and this bug impacts in a lesser extent the second run. We report the results using the Mean Average Interpolated Precision (first column) and, with the *LIG.1000_FOC_RIC* run as baseline, the *LIG.075075_FOC_RIC* run shows that collectionlinks improve results (+13.6%). Moreover, in Table 2 and in Figure 1, we see that for the results between 0.01 recall and 0.25 recall, the *LIG.075075_FOC_RIC* performs much better than the *LIG.1000_FOC_RIC*.

5.2 Relevant In Context Task

For the Relevant In Context Task, we take “default” focused results and re-ordered the first 1500 doxels such that results from the same document are clustered together. It considers the article as the most natural unit and scores the article with the score of its doxel having the highest RSV.

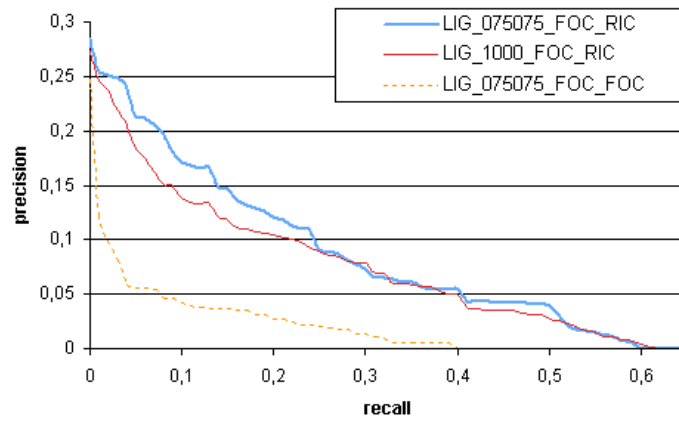
We submitted three runs :

- *LIG.1000_RIC* : a baseline run which doesn’t take into account the inner collectionlinks to score doxels. We set $\lambda = 1.0$ and $\beta = 0.0$;

Table 2. Focused Task for INEX2007 Ad Hoc.

<i>Run</i>	precision at 0.0 recall	precision at 0.01 recall	precision at 0.05 recall	precision at 0.10 recall
<i>LIG_075075_FOC_FOC</i> <i>MAiP</i> = 0.0150	0.2474	0.1215	0.0560	0.0425
<i>LIG_1000_FOC_RIC</i> <i>MAiP</i> = 0.0522	0.2734	0.2465	0.1853	0.1388
<i>LIG_075075_FOC_RIC</i> <i>MAiP</i> = 0.0593(+13.6%)	0.2847 (+4.1%)	0.2554 (+3.6%)	0.2126 (+14.7%)	0.1706 (+22.9%)

Fig. 1. Interpolated Precision/Recall - Focused Task



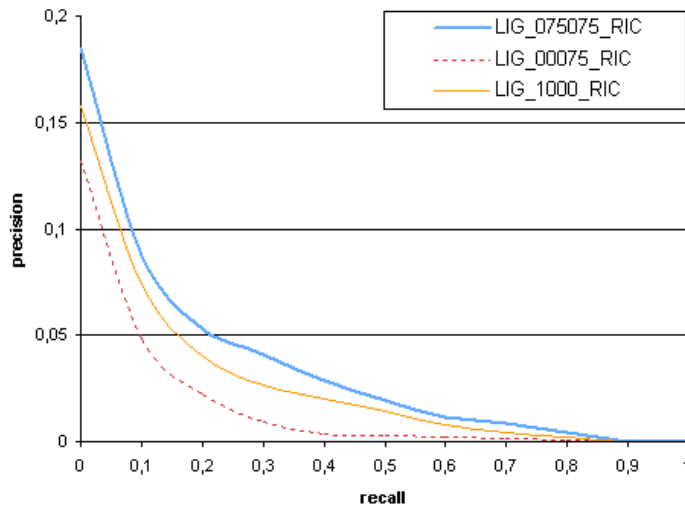
- *LIG_075075_RIC* : a retrieval approach based on the collectionlinks use. We set $\lambda = 0.75$ and $\beta = 0.75$;
- *LIG_00075_RIC* : an approach that consider the RSV of a doxel only considering its context: we set $\lambda = 0.0$ and $\beta = 0.75$.

For the relevant in context task, our results in terms of non-interpolated generalized precision at early ranks $gP[r], r \in \{5, 10, 25, 50\}$ and non-interpolated Mean Average Generalized Precision $MAgP$ are presented in Table 3. Figure 2 shows the generalized precision/recall curve. This shows that using collectionlinks and the doxels content (*LIG_075075_RIC*) improves the baseline by a ratio greater than 15%. The *LIG_00075_RIC* gives bad results, showing that the context of the doxels alone is not relevant. In Figure 2, we see that the *LIG_075075_RIC* run is also above the default run.

Table 3. Relevant In Context Task for INEX2007 Ad Hoc.

<i>Run</i>	gP[5]	gP[10]	gP[25]	gP[50]
<i>LIG_1000_RIC</i> <i>MAgP</i> = 0.0232	0.0678	0.0597	0.0423	0.0307
<i>LIG_075075_RIC</i> <i>MAgP</i> = 0.0305 (+31.5%)	0.0785 (+15.8%)	0.0726 (+21.6%)	0.0501 (+18.4%)	0.0375 (+22.2%)
<i>LIG_00075_RIC</i> <i>MAgP</i> = 0.0122 (-47.4%)	0.0587 (-13.4%)	0.0423 (-29.1%)	0.0290 (-31.4%)	0.0203 (-33.9%)

Fig. 2. Generalized Precision/Recall - Relevant In Context task



5.3 Best In Context Task

For the Best In Context Task, we examine whether the most focused doxel in a relevant document is the best entry point for starting to read relevant articles. We take “normal” focused results and the first 1500 doxels belonging to different files. For this task, we submitted three runs:

- *LIG_1000_BIC* : the baseline run which doesn’t take into account collectionlinks: we set $\lambda = 1.0$ and $\beta = 0.0$;
- *LIG_075075_BIC* : the retrieval approach based on the use of collectionlinks. We set $\lambda = 0.75$ and $\beta = 0.75$;
- *LIG_00075_BIC* : the approach that uses only the context of doxels to compute their RSV: we set $\lambda = 0.0$ and $\beta = 0.75$.

For the best in context task, our results are presented in Table 4 and Figure 3 with the same measures as the Relevant In Context Task results. Conclusions are the same: using collectionlinks and content improves the baseline by a mean average of more than 24%, and the *LIG_00075_BIC* run is consistently below the baseline. There is one result however, the *LIG_00075_BIC* run outperforms the baseline at $gP[5]$ by more than 10% and in Figure 3 we see than the baseline and the *LIG_00075_BIC* are quite close to eachothers. This means that the *a priori* links are really meaningful.

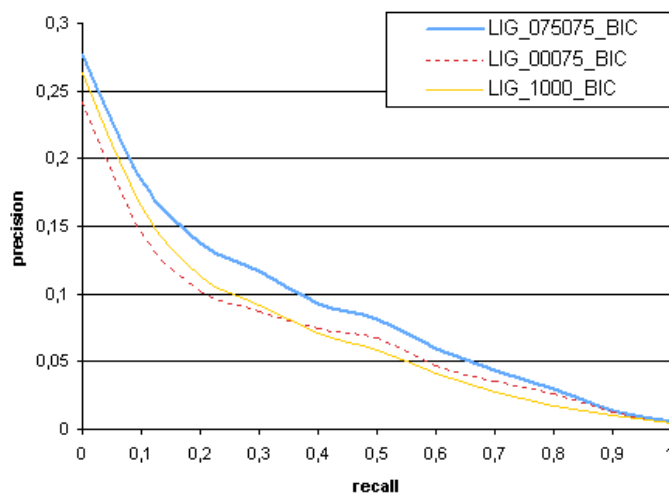
Table 4. BIC for INEX2007 Ad Hoc.

<i>Run</i>	gP[5]	gP[10]	gP[25]	gP[50]
<i>LIG_1000_BIC</i> <i>MAgP</i> = 0.0614	0.1191	0.1165	0.1036	0.0892
<i>LIG_075075_BIC</i> <i>MAgP</i> = 0.0762 (+24.1%)	0.1405 (+18.0%)	0.1268 (+8.8%)	0.1158 (+11.8%)	0.0950 (+6.5%)
<i>LIG_00075_BIC</i> <i>MAgP</i> = 0.0632 (+2.9%)	0.1318 (+10.7%)	0.1123 (-3.6%)	0.0966 (-6.8%)	0.0801 (-10.2%)

6 Summary and Conclusion

We proposed a way to integrate the content of the doxels as well as their context (collectionlinks in INEX 2007 documents). We have submitted runs implementing our theoretical proposals for the different Ad Hoc tasks. For each of the tasks, we showed that combining content and context produce better results than considering content only and context only of the doxels, which is a first step in validating our proposal. According to the official evaluation of INEX 2007, our best runs are ranked in the last third of participants systems, for the Content-Only runs. However, we plan to improve our baseline to obtain better results in the following directions:

Fig. 3. Generalized Precision/Recall - Best In Context task



- As mentioned earlier, the size of the vocabulary used is too small, leading to query terms out of our vocabulary. We are currently extending this vocabulary, so we decide to launch a new indexation and test once again our proposal.
- When submitting our runs for our first participation at INEX competition we found some bugs related to the identifiers of the doxels, so the results were negatively impacted.
- We are working on the integration of negative terms in the query, in a way to get better results.

References

1. S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107-117, 1998.
2. D. H. Fang Huang, Stuart Watt and M. Clark. Robert Gordon University at INEX 2006: Adhoc Track. In *INEX 2006 Workshop Pre-Proceeding*, pages 70-79, 2006.
3. J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604-632, 1999.
4. B. Piwowarski and M. Lalmas. Interface pour l'évaluation de systemes de recherche sur des documents XML. In *Premiere COnference en Recherche d'Information et Applications (CORIA '04)*, Toulouse, France, 2004. Hermes.
5. G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval - Chapter 6, page 203*. McGraw-Hill, Inc., New York, NY, USA, 1986.
6. J. Savoy. An extended vector-processing scheme for searching information in hyper-text systems. *Inf. Process. Manage.*, 32(2):155-170, 1996.

7. M. D. Smucker and J. Allan. Using similarity links as shortcuts to relevant web pages. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 863–864, New York, NY, USA, 2007. ACM Press.
8. C. van Rijsbergen. *Information retrieval, Second edition - Chapter 3*. Butterworths, 1979.
9. D. Verbyst and P. Mulhem. Doxels in context for retrieval: from structure to neighbours. In *SAC '08: Proceedings of the 2008 ACM symposium on Applied computing*, New York, NY, USA, 2008. ACM.

CSIR at INEX 2007

Wei Lu, Dan Liu, Jiepu Jiang

Center for Studies of Information Resources, School of Information Management
Wuhan University, China
reedwhu@yahoo.com.cn

Abstract. In this paper, we describe the Centre for Studies of Information Resources' participation in the INEX 2007 ad-hoc track. For the Focused Task and Relevant in Context Task, our main aim this year is to investigate the affects of the selection of retrievable elements and passages adoption. For the Best in Context Task, we proposed a novel method of choosing the best entry point. Our submission evaluation shows that our method didn't produce ideally effectiveness. The reason is still need to be further investigated.

1. Introduction

This is the third year for us and the first year for the CSIR's participation in INEX. In the previous two years, we used a field-weighted BM25 model for INEX 2005 [1] and a simple BM25 model based on elements cut-off for INEX 2006 [2] respectively. Our results show the field-weighted method is promising while the latter one is obscure.

For INEX 2007, there are 3 ad-hoc subtasks: the Focused Task, the Relevant in Context Task and the Best in Context Task. These 3 tasks are derived from INEX 2006 ad-hoc subtasks. But the tasks' requirement is a little different, that is, not only the elements but also passages are allowed to be retrieved as relevant units. This raises a new question: how to recognize a passage?

The retrieval of passages has been an occasional interest within the document retrieval community for many years. Many of the problems and possibilities of using passage-level evidence are discussed by Callan [3]. Robertson et al [4] point out that passages may be defined more-or-less arbitrarily (for example in terms of fixed word-length windows on the text, or by means of relatively superficial parsing such as sentence or paragraph separation) at the simplest level. Then each document is retrieved on the basis of the score of the best-matching passage within it, rather than on the basis of scoring the entire document. In our experiment, considering the structure of XML document and for simplicity, we take two or more paragraphs as a retrievable passage. The detail method of determining a passage will be introduced in section 2.

The selection of retrievable elements (tags) is investigated in our experiment this year. Section 2 gives more information on the selection of these elements. And the proposed novel method for selecting the best entry point is also discussed in section 2.

In section 3, we discuss our submitted runs. Evaluation results are reported in section 4. A conclusion and further work to be undertaken are given at the end.

2. Our Method

In this section, we firstly briefly introduced the BM25 model, and then discuss the selection of retrievable elements and element cut-off. Further, the method of recognizing passage and method for best entry point search are illustrated.

2.1 BM25 Model

As that in INEX 2006, the BM25 formula used in our experiment is as follows:

$$wf_j(d, C) = \frac{(k_1 + 1)tf_j}{k_1((1 - b) + b \frac{dl}{avdl}) + tf_j} * \frac{N - df_j + 0.5}{df_j - 0.5} \quad (1)$$

where C denotes the document collection, tf_j is the term frequency of the j th term in document d , df_j is the document frequency of term j , dl is the document length, $avdl$ is the average document length across the collection, and k_1 and b are tuning parameters.

From formula (1) we can see that we used a slightly different function for term's collection weight. That is, we avoided using logarithmic functions which produce negative weight values.

2.2 Selection of Elements

In INEX 2005, we only chose <article>, <body>, <section>, <p> as retrievable element. Before this year's participation, we analyzed the element distribution in INEX 2006's relevance assessments. Table 1 shows the top ranked 11 tags (percentage is larger than 2%) in the INEX 06's relevant assessments, and table 2 shows the top ranked best entry point distribution in the INEX 06's relevant assessments. After having examined this, we determined to use all the tags in table 1 as the retrievable elements (tags). The experiment however shows our selection doesn't produce good results. The underlying reason is still need to be further investigated.

2.3 Passage Recognition

As stated in Section 1, considering the structure of XML document and for simplicity, we take two or more paragraphs (tag <p>) as a retrievable passage. Given

some candidate paragraphs, only paragraphs satisfy the following two rules could be treated as passages:

- (1) The paragraphs are in the same section;
- (2) These paragraphs are adjacent.

Table 1: Top ranked relevant element distribution in the INEX 06's relevant assessments

Element tag	Count	Percentage
collectionlink	80589	37.05 %
p	15873	7.29 %
emph2	14926	6.86 %
item	14693	6.75 %
cell	13898	6.39 %
unknownlink	12598	5.79 %
section	9714	4.46 %
emph3	5930	2.72 %
article	5648	2.59 %
body	5646	2.59 %
title	5371	2.46 %

Table 2: Top ranked best entry point distribution in the INEX 06's relevant assessments

Element tag	Count	Percentage
p	1743	30.86%
name	986	17.45%
emph3	626	11.08%
collectionlink	587	10.39%
title	464	8.21%
body	361	6.39%
item	193	3.41%
section	126	2.23%
unknownlink	87	1.54%
caption	71	1.25%
image	71	1.25%
normallist	63	1.11%
template	62	1.09%

For example, given some candidate paragraphs p1, p3, p4, p5 and p6 in Fig. 1, there is only 1 validate passage, which contains p4, p5 and p6.

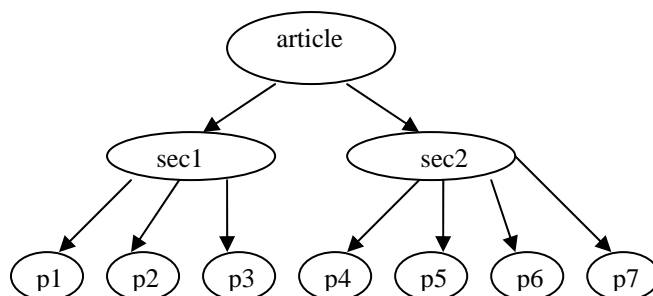


Fig.1 The method of recognizing passages

2.4 Method for Best in Context Task

We used 3 methods for Best in Context Task this year. The first 2 methods are similar to those for last year. That is, in the first method, we just take the element with the highest weight score (best-match element) in each document as the best entry point; in the second method, the distribution of element weight scores in the document is considered. See more in [2].

Our newly proposed method this year for the Best in Context Task is that the adjacent information of paragraphs is taken into consideration. In this method, we first choose the best-match element in each document; then if the best-match element is a paragraph, we'll further investigate relevant paragraphs in context; if there are one or more paragraphs in the same section are adjacent to the best-match paragraph, then the first one will be taken as the best entry point.

For example, in Fig. 1, if p1, p3, p4, p5 and p6 contains relevant information and p5 is the best-match element, then the best entry point for this document is p3. We didn't consider element cross sections, much work still needed to be done on this.

3. Description of the Experiments

For each of the sub-task, we submitted 3 runs respectively. The details of these experiments are as follows:

3.1 FOCUSED TASK

The 3 submitted runs for FOCUSED TASK are as follows:

- FOCU_BM25_BASE_FILTER uses simply basic BM25 model to choose the best weighted elements in each article;

- FOCU_BM25_BASE_FILTER_BSP uses simply basic BM25 model to choose the best weighted elements in each article, and only body, section and p are considered as the retrievable tags;
- FOCU_BM25-PASSAGE-FILTER is similar to the first one, but the passage is considered.

3.2 RELEVANT IN CONTEXT TASK

For this task, we submitted runs REL_BM25_BEST_FILTER, REL_BM25_BEST_FILTER_BSP and REL_BM25_PASSAGE_FILTER. These runs use the same conditions as the ones for FOCUSED TASK. The difference is that the results in the runs are grouped by articles.

3.3 BEST IN CONTEXT TASK

For this task, our submitted 3 runs are BM25_BEST_FILTER, BM25_PARENT_FILTER and BM25_PASSAGE_FILTER.

- BEST_BM25_BEST_FILTER uses the first method talked in section 2.4, which chooses the best weighted element in each article;
- BEST_BM25_PARENT_FILTER uses the second method in section 2.4, which considers the distribution of element weight scores in the document;
- BEST_BM25-BEST-FIRST uses the novel method proposed in this paper, see more in section 2.4.

4. Evaluation

The evaluation results of our runs are shown in Table 3, Table 4 and Table 5. Our two runs in Table 3 and Table 4 haven't been listed in the official results. From the INEX official result reports, our runs don't do well. Only the runs BEST_BM25_PARENT_FILTER for the best entry point search produce relatively better results. Compare with our INEX 2005's submission, we found that the selection of retrievable elements this year produce even worse results. The reason of this needs more experiments.

Table 3: Evaluation results for FOCUSED Task

Runs	Interpolated precision at 0.01 recall
FOCU_BM25_BASE_FILTER	0.2812
FOCU_BM25_BASE_FILTER_BSP	0.2996
FOCU_BM25_PASSAGE_FILTER	-

Table 4: Evaluation results for Relevant in Context Task

Runs	MAgP
REL_BM25_BEST_FILTER	0.0525
REL_BM25_BEST_FILTER_BSP	0.0507
REL_BM25_PASSAGE_FILTER	-

Table 5: Evaluation results for BEST IN CONTEXT Task

Runs	MAgP
BEST_BM25_BEST_FILTER	0.0967
BEST_BM25_PARENT_FILTER	0.1228
BEST_BM25_BEST_FIRST	0.0983

5 Conclusion

For all the three ad-hoc runs, we submitted totally 9 runs. For the Focused Task and Relevant in Context Task, our main aim this year is to investigate the affects of the selection of retrievable elements and passages adoption. For the Best in Context Task, we proposed a novel method of choosing the best entry point. Our submission results show that our method didn't do quite well. The selection of retrievable elements based on the INEX 2006's relevant assessments this year produce even worse results. This needs to be further investigated. We have proposed another novel method for the best entry point location, but more works still need to be done on that.

Acknowledgements

This work is supported in part by National Social Science Foundation of China 06CTQ006.

References

- [1] W. Lu, S. Robertson, A. Macfarlane. Field-Weighted XML Retrieval Based on BM25. Proceedings of INEX 2005. LNCS. 2006 126-137
- [2] W. Lu, S. Robertson, A. Macfarlane. CISR at INEX 2006. Proceedings of INEX 2006. LNCS. 2007 57-63
- [3] J. Callan. Passage-level evidence in document retrieval. Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Springer-Verlag. 1994 302 - 310
- [4] S. Robertson, W. Lu, A. Macfarlane. XML-structured documents: retrievable units and inheritance. FQAS 2006. Springer LNCS. 2006 121-132

Document Order Based Scoring for XML Retrieval

Paavo Arvola
Department of Information Studies, Kanslerinrinne 1,
33014 University of Tampere, Finland
paavo.arvola@uta.fi

Abstract. This study presents a novel matching method based on score propagation for the ancestors and the elements' positions in document order. In addition, it presents a length normalization component substitute, which enables query processing based merely on key locations in the inverted file.

Keywords: Document-order, Dewey, Length normalization, Matching

1 Introduction

1.1 Aims

This study presents a novel, pruned version of TRIX (Tampere Retrieval and Indexing for XML) IR system [2]. The version utilizes DoOrBa (Document-Order Based) scoring for Content Only queries, in which the matching is based solely on the inverted file. This kind of approach enables a genuine schema independent scoring. In other words matching for an element can be done without the knowledge of the common structure of the element, including length normalization.

In terms of retrieved element independency of each other, there are roughly two approaches how to present XML retrieval results to the user [6, 7].

1. elements are independent retrieval units
2. elements are viewed within their context

Let's consider the latter, navigation driven use case, where the elements are not mere returnable units, but rather highlighted within the document. This is intended for the user easier to navigate thorough the relevant content of the whole document. This may include highlighting the relevant text content, starting the browsing from the best entry point [5], link-anchor based browsing between relevant items within the document [e.g. 1, 3]. Accordingly, if the first descendant element(s) of an element is relevant, it is practically unimportant, if the returnable element is an element itself or its first child in document order (shortly ido). Hence, we call elements starting in the same location in the document *navigationally equivalent*.

In addition to navigational equivalence or closeness of elements we make a supposition that the first descendant elements (ido) are better in describing the whole content of an element than the descendant elements further. The first descendant elements mean e.g. titles for the sections and headings, abstracts and keywords for the whole documents. As a result, the importance of these elements should affect more on the retrieval status value of the ancestor. In contrast, if the best matching content is rather in the last descendants (ido) of the element, these descendants should be returned instead of the element.

The DoOrBa scoring method recursively propagates element scores for the ancestors. This is done by giving decreasing values for the descendant elements based on their position (ido), and can thus be used as a substitute for elements length normalization. The substitute crops the tail of an element and reduces also the importance of the elements length in element scoring. This enables matching based solely on the inverted file, which is described in the following sections.

To summarize, there are two factors, which are essential in motivating the TRIX's DoOrBa approach:

1. Text occurring early is essential in element weighting (title)
2. Navigational equivalence or nearness of elements

2 Indexing and Scoring

The query evaluation of TRIX system is based on structural indices (i.e. Dewey labels). Especially the DoOrBa scoring is based solely on this aspect. This section presents the indexing mechanism and scoring based on the mechanism, and the section 3 focuses on the query processing based on structural indices and DoOrBa scoring in more detail.

2.1 Structural indices

In TRIX the management of structural aspects is based on the structural indices (i.e. labels), also called Dewey indices. In Figure 1 there is a tree presentation of an XML document with indices and element names for each node. The idea of Dewey indices in the context of XML is that the topmost (root) element is indexed by $\langle 1 \rangle$ and its children by $\langle 1,1 \rangle$, $\langle 1,2 \rangle$, $\langle 1,3 \rangle$. Further, the children of the element with the index $\langle 1,2 \rangle$ are labelled by $\langle 1,2,1 \rangle$, $\langle 1,2,2 \rangle$ and so on. This kind of indexing enables analyzing of the relationships among elements in a straightforward way. For example, the ancestors of the element labelled by $\langle 1,2,2,1 \rangle$ are associated with the indices $\langle 1,2,2 \rangle$, $\langle 1,2 \rangle$ and $\langle 1 \rangle$. In turn, any descendant related to the index $\langle 1,2 \rangle$ is labelled by $\langle 1,2, \xi \rangle$ where ξ is a non-empty part of the index.

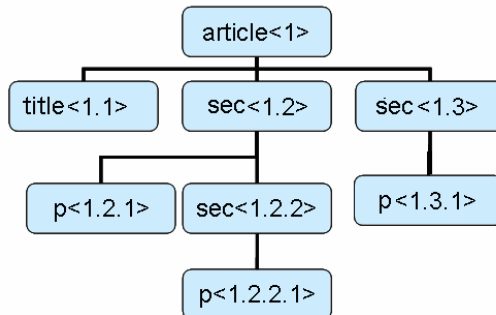


Figure 1: Tree presentation of an XML document with element names and indices

Moreover, because the labelling for the siblings is executed in the document order the indexing works well in figuring out the preceding-following relationship between known indices as well. As an illustration of this, we can say that element $\langle \xi, i \rangle$ is the i :th child of the element ξ , and thus preceding an element $\langle \xi, i+1 \rangle$, if it exists.

2.2 DoOrBa scoring

Similarly to e.g. GPX [4] the DoOrBa (document order based) scoring is calculated separately for leaf elements and branch elements. This is done so that the leaf scores have been delivered upwards to the branch elements. A leaf element is considered here to be an element which contains directly a *text element*. It is worth noting that an element is considered to have no more than one text element directly. In other words the text element means all direct text content of an element. A branch element is an element having children (other than text elements). Due to these definitions an element can be a leaf element, a branch element or even both. For instance the following paragraph contains both text elements and is also a branch element (has a child: *collectionlink*).

<p>

It was rumoured that there was some intra-band tension throughout the latter half of 1996, and at the end of a successful tour of Britain later that year, at Brixton Academy on 16th December 1996, the band told Max they would not renew Gloria's management contract. Max Cavallera left the band (and formed a band called

<collectionlink xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple" xlink:href="221501.xml">Soufly</collectionlink>

), and the others announced that they would continue under the Sepultura name and were searching for a replacement.

</p>

In the example the text presented in italics form the content of the element p . The score of an element is a sum of leaf element and branch element scores. Since the text elements tend to be short, although of varying length and importance, the score of the text element is basically the sum of the *idf* (inverse document frequency) values of query terms in the text element. The leaf score (text score) is calculated with the following equation (1), in which the v is a constant, and m is the number of (unique) terms in the query expression.

$$textScore(q, \xi) = \sum_{t=1}^m idf_t \quad (1)$$

The score for the branch element is calculated recursively as a result of the scores of its child elements. This has been done so that the scores of the child elements are considered in relation to their positions (*ido*). The primary goal is to emphasize scores of child elements appearing early (*ido*) in the child list. This is done by applying a specific *child score vector* (CS) for the element weighting.

The CS is filled with constant values, which are used to express the contribution each child has in branch element weighting. The position of the value in the vector corresponds to the child number (*ido*), and the smaller the value, the more important is the corresponding child. We note $CS[i]$ to denote the i :th component of the vector. For instance applying a $CS=\langle a,b,c \rangle$ for the element ξ , means that a is for $\langle \xi, 1 \rangle$, b for $\langle \xi, 2 \rangle$, c for $\langle \xi, 3 \rangle$. On the basis of this, we get a following general matching formula (2), which combines elements branch score (if any descendants) with elements text score (if any text):

$$score(q, \xi) = \sum_{i=1}^{\min(n, len(CS))} \left(\frac{score(q, \langle \xi, i \rangle)}{v \times (a + CS[i])} \right) + textScore(q, \xi) \quad (2)$$

in which

- $score(q, \xi)$ is the score of the element ξ in relation to the query q
- n is the number of child elements
- $len(CS)$ is the length of CS
- i is the child element position in the element's child list
- v and a are constants for tuning

Decreasing the value of a and v emphasizes the effect of the CS vector. The equation $v \times (a + CS[i])$ is actually used as a substitute of a length normalization component and can be thus called a *length normalization substitute*. The score of the component affects to the elements score by adding the child's weight divided by it.

For instance, if we have a vector $CS=\langle 1,2,3,4,5\rangle$, $a=0$ and $v=1$, the weight of the first child is taken into account as a whole, the score of the second child increases the element's score by $1/2$ of the child's score, the third by $1/3$ and so on.

3 Query processing based on structural indices and DoOrBa scoring

In our approach, the inverted file (IF) contains explicit locations of keys. That means for each key there is a set of indices, for example:

$$IF = \{\langle \text{keya}, \{\langle 1,1,3\rangle, \langle 1,2,5,1\rangle\}\rangle, \langle \text{keyb}, \{\langle 1,2,4\rangle, \langle 1,2,5\rangle, \langle 1,2,6\rangle\}\rangle, \dots\}$$

To be accurate, the inverted file contains the indices of the lowermost (i.e. leaf) elements having text containing the key. An inverse function (3) for individual key weights based on the DoOrBa function (2), aside with the inverted file allows coping with only the explicit indices of keys.

$$w(t, \xi) = \sum_{\xi' \in \text{inds}(t, IF) \wedge \xi' \in \text{descs}(\xi)} \left(\prod_{i=\text{len}(\xi')+1}^{\text{len}(\xi')} \left(\frac{1}{v \times (a + CS[i])} \right) \times \text{idf}_i \right) \quad (3)$$

in which

- $w(t, \xi)$ is the weight of key t for the element ξ
- function $\text{inds}(t, IF)$ returns the indices related to the key t in the IF
- function $\text{descs}(\xi)$ returns the descendants or self of ξ
- function $\text{len}(\xi)$ is the length of index ξ (e.g. $\text{len}(\langle 1,2,5\rangle) = 3$)

As an illustration of the abovementioned formula, let's consider the sample IF in the beginning of this section and calculate $w(\text{keya}, \langle 1\rangle)$ with the constant values $v=1$, $a=0$ and $CS = \langle 1,2,3,4,5\rangle$.

$$w(\text{keya}, \langle 1\rangle) = \frac{1}{1} \times \frac{1}{3} \times \text{idf}_{\text{keya}} + \frac{1}{2} \times \frac{1}{5} \times \frac{1}{1} \times \text{idf}_{\text{keya}}$$

As intermediate results, we get all key weights for the descendants' of the element $\langle 1\rangle$. The final score of the element is the sum of key weights of the query keys.

4 Results and discussion

By the result deadline of INEX, only basic settings have been tested. For every run, we used the CS as an infinite vector $CS = \langle 1,2,3,\dots \rangle$. Even so, the early precision results for the Focussed task were satisfactory. TRIX DoOrBa reached 15th, 17th and 19th positions in the precision at 5% recall, with runs from 8 institutes ahead.

Typically, aside of the inverted file, the length normalization requires additional data structures for query processing [8], which the length normalization substitute does not require. In our approach the size of the content-only inverted file for the Wikipedia collection (4.6 GB) is 739 MB, calculated after stemming and stopword removal.

The structural indices of the key locations carry also some information about element structure; this can be utilized in the estimation of the element length. Consequently, this will probably lead to more accurate matching. Further studies may include this aspect.

Acknowledgements

This study is supported by the Academy of Finland under grant number 115480. The travel and accommodation costs are granted by the Nordic Research School in Library and Information Science (Norslis).

References

1. Arvola, P., Junkkari, M., and Kekäläinen J. Applying XML retrieval methods for result document navigation in small screen devices. Proceedings of MUIA 2006, 2006, 6-10.
2. Arvola, P., Kekäläinen, J., and Junkkari, M. Query evaluation with structural indices. INEX 2005, LNCS 3977, 2005, 134-145.
3. Chiaramella, Y. Information retrieval and structured documents. Proceedings of ESSIR 2000, 2000, 286-309.
4. Geva, S. GPX - Gardens point XML IR at INEX 2006. INEX 2006, LNCS 4518, 2007, 137-150.
5. Lalmas, M. and Reid, J. Automatic identification of best entry points for focused structured document retrieval. Proceedings of CIKM 2003, 2003, 540-543.
6. Larsen, B., Tombros, A., and Malik, S. Is XML retrieval meaningful to users?: searcher preferences for full documents vs. elements. Proceedings of SIGIR 2006, 2006, 663-664.
7. Lehtonen, M., Pharo, N., and Trotman, A. A taxonomy for XML retrieval use cases. INEX 2006, LNCS 4518, 413-422.
8. Zobel, J. and Moffat, A. 2006. Inverted files for text search engines. ACM Comput. Surv. 38, 2, 2006, 6.

An XML Information Retrieval using RIP List

Hiroki Tanioka

Innovative Technology R&D, JustSystems Corporation,
108-4 Hiraishi-Wakamatsu Kawauchi-cho Tokushima-shi Tokushima, Japan
hiroki.tanioka@justsystems.com

Abstract. There are two approaches for XML information retrieval. One is based on the approaches in the database field, and the other is based on the approaches in the information retrieval field. And the vector space model is commonly used in the information retrieval field. In the previous year, we developed an XML information retrieval system with the vector space model. To be more flexible for the query, we also developed the system using unitizing of fragment elements. The system realized searching XML elements for numerous queries without reindexing. However the system took time for unitizing of fragment elements. To solve the problem, our system is composed of an inverted-file list and a relative inverted-path list in this year. Then we have examined the effectiveness of the system in the Initiative for the Evaluation of XML Retrieval (INEX) 2006 Adhoc Track.

1 Introduction

In the research field of document information retrieval (IR), the unit of retrieval results returned by IR systems is a whole document or a document fragment, like a paragraph in passage retrieval. Traditional IR systems based on the vector space model compute feature vectors of the units and calculate the similarities between the units and the query. Our system uses keywords (terms; words) as the query, and separates XML [1] documents into document information and structure information parts. Therefore the system searches fast XML nodes (nodes; sub-documents) which include query terms using an inverted-file list (Section 2.2).

For huge size XML documents, our system indexes all XML nodes with each term. Here the terms are located just below the XML node. At the retrieving phase, the score of retrieved node is merged and calculated from its descendant nodes. To merge scores while identifying parent-child relationships, our system uses a Relative Inverted-Path list (*RIP* list; Section 2.5) which is labeled preorder in order to save the structure information.

The indexing way was already published IR-CADG[13], which are separately divided into document information and structure information. Also, the merging method was proposed as Bottom-UP Scheme (BUS)[12]. In recent years, SIRIUS[15] achieved high precision using a combination of document information and structure information. And GPX [16] used an index for some types of queries by BUS method.

However GPX showed average 7.2 seconds per topic, it took more time than 30 seconds depending on the type of query. Unfortunately it's not yet up to a practical level. Meanwhile, a way of eliminating unwanted part of XML documents was proposed by Hatano[17]. With that system, we can increase XML document size, but it needs to reindex according to the type of query.

For these reasons, after studying we have made the fast XML retrieval system at practical level using a RIP list. Our system is without reindexing while keeping the

```

<? xml version="1.0" ?>
<article>
  <body>
    <sec>
      <p>I am XML.</p>
      <p></p>
      <p>First, Text is here. Here issues XML.</p>
      <p>
        <image>
          <title><p></p></title>
        </image>
      </p>
    </sec>
  </body>
</article>

```

Fig. 1. XML document

```

"t"    → 0: {{3, 1}}
"am"   → 1: {{3, 1}}
"xml"  → 2: {{3, 1}, {5, 1}}
"first" → 3: {{5, 1}}
"text" → 4: {{5, 1}}
"is"   → 5: {{5, 1}}
"here" → 6: {{5, 2}}
"issue" → 7: {{5, 1}}

```

Fig. 2. Inverted list

index size small. Then the system records average 3.94 seconds (in the worst case; 9.95 seconds) and gets a good precision as Focused Task (Overlap=on) on the Initiative for the Evaluation of XML Retrieval (INEX) 2006 Adhoc Track.

2 XML Information Retrieval

XML information retrieval targets XML documents, which retrieves and ranks retrieved results in units of not only XML documents but also XML nodes to queries. With a database-based approach, first, the system narrows the number of the retrieved nodes using XPath[2], XQuery[3] and such. After that, it performs a keyword search though. Current research[17] indicates that the system has low precision and requires considerable time for retrieval time. Because a keyword-based search system can't reduce the number of the results, using queries which consist of entirely keywords.

2.1 Sub-document Retrieval

In the research field of document information retrieval, there is an approach of passage retrieval which replies portions of document such as Chapter, Section and Paragraph. Also, Evans[10] proposed the approach of document retrieval using sub-documents, then it achieved some positive results. The results supported the effectiveness of retrieving portions of document.

A solution to the issue is to score for each hierarchical level of document, and which accomplishes the purpose based on portions of document are uniform in size. However XML nodes have variation in size. Thus we need an indicator of node score with the information of node size. And XML nodes have structure information as well as the size, which also have a great deal of potential in the XML information retrieval.

2.2 Index

Our system has an inverted-file index which manages document information. In the system, word terms and XML nodes become numerical terms, term IDs and node IDs respectively. Then term IDs and node IDs are indexed as bellow.

Term ID: {Node ID, Term Frequency}

Where the term frequency is a frequency of appearance of a word as node ID in a node as node ID. And the inverted-file index for Figure 1 is as shown in Figure 2.

The XML Wikipedia collection in INEX 2006 Adhoc Track has 52,562,497 nodes and 13,903,331 unique terms. When both a node ID and a term ID are in 4 byte integer, the size of a inverted-file list is practically about 1.78 GB.

2.3 Retrieval Model

Our system basically uses TF-IDF score. TF-IDF score is regarded as amount of information, which has additivity. Therefore, an node score is easily calculable, when it consists of its descendant nodes. First, the TF-IDF score of the j th node is composed of the term frequency tf_i of the i th term in the query, the number of nodes including the term and the number of all the nodes in the XML collection.

$$L_j = \sum_{i=1}^t \log(tf_i \cdot \frac{n}{f_i}) \quad (1)$$

Then, the node score is summation of its descendant nodes score, but it is a problem that the root node always has the higher score than its descendant nodes. Therefore, the summation score R_j of the j th node is composed of the summation number T_j of terms contained in the j th node, the summation score L_k of the k th node as the j th node's descendant and the summation number t_k of terms contained in the k th node.

$$R_j = \sum_{k \text{ child of } j} D(k, t_k, T_j) \cdot L_k \quad (2)$$

$$T_j = \sum_{k \text{ child of } j} t_k \quad (3)$$

And the coefficient function $D(k, t_k, T_j)$ is as shown in the following equation,

$$D(k, t, T) = \begin{cases} 0, & \text{if } t > T_1 \cup T > T_2 \\ 1/(\log d_k + 1), & \text{otherwise} \end{cases}$$

where $T_1 (= 100)$ is a threshold for the number of terms contained in the node to merge, $T_2 (= 2,000)$ is a threshold for the number of terms contained in the merged node. According to the above coefficient function, scores decays depending on the difference d_k between j th node and k th node.

Then, let α is the set of terms included in the query, β_j is the set of terms included in the j th node. The conjunction, $\gamma_j = \alpha \cap \beta_j$, is the set of query terms included in the j th node. For every node,

$$s_j = \text{count}(\delta_j), \quad \delta_j = \bigcup_{k \text{ child of } j} \gamma_k, \quad (4)$$

$$S_j = \frac{Q}{q} \cdot s_j$$

where $Q (= 500)$ is a constant number. S_j is one of heuristic scores we called leveling score, which means that the score is the highest, when the number of terms contained in the set is the most while the number of terms contained in the query is the least.

$$V_j = \frac{R_j + S_j}{\log T_j} \quad (5)$$

After that, the score V_j of j th node is composed of the TF-IDF score R_j , the leveling score S_j and the logarithm number of terms T_j . Thus, the retrieved results are chosen from the node list V_j which is sorted in descending order of scores.

	Nord ID (pre-orderd)	XPath list	Frequency of terms	Offset for parent
0		/article	0	0
1		/article/bdy	1	0
2		/article/bdy/sec	1	0
3		/article/bdy/sec/p	1	3
4		/article/bdy/sec/p	2	0
5		/article/bdy/sec/p	3	7
6		/article/bdy/sec/p/image	1	0
7		/article/bdy/sec/p/image/title	1	0
8		/article/bdy/sec/p/image/title/p	1	0
9		/article/bdy/sec/p/image/title/p	2	0

Rip List

Fig. 3. Relative inverted-path list

2.4 Information Granularity

For the XML information retrieval, the formula 5 is intended as follows,

- The size of node: The best result has an appropriate amount of words for each user.
- The granularity of node: The best result has the highest density of terms contained in the query.
- The coverage of query: The best result includes the highest coverage of terms contained in the query.

It is the information granularity issue that the best node as retrieved result is depending on the size of node and the density of the terms contained in the query. In our system, The information granulariy is measured by the means of applying the coefficient function $D(k, t, T)$ and normalizing in the number of terms T . And for the coverage of query, our system uses the leveling method.

Then, the number of terms T of the normalizing method means the base of a logarithm for the amount of information I , where P is the occurrence probability based on the score $R + S$ ($P \propto e^{-(R+S)}$),

$$I = -\log P \propto R + S,$$

$$V \propto \frac{I}{\log T} = -\frac{\log P}{\log T} = -\log_T P \quad (6)$$

Hence, the score V in the formula 6 is the indicator, which is interprets the quantity V of the occurrence probability P coded in surprisal T , in the node as the information source.

2.5 Relative Inverted-Path List

There are various indexing and labeling means for strucuture information [18], Our system labels in preorder of XML nodes, which are traversed in depth first order. As a

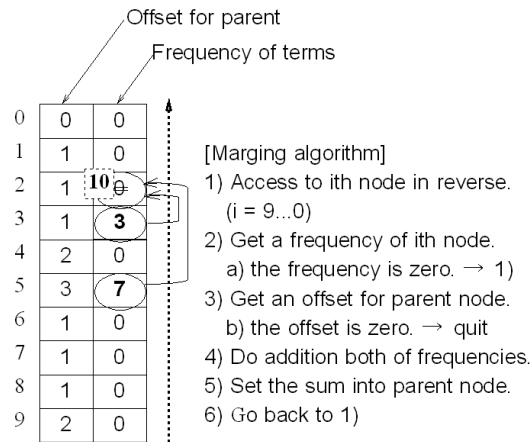


Fig. 4. Merging using a relative inverted-path list

result, the list contains all the structure information and has uniqueness, although the size of list are relatively small. And the list adopts all the distances between nodes and their child node, which we called the relative infreted-path (RIP list).

Node ID: {Distance, Term Frequency}

Figure 3 shows the RIP list has high accessibility to the parent node ID from each node ID. Our system merge the numbers of terms contained in every node, the scores of retrieved nodes and the numbers of query terms contained in each node (in Section 2.3). Figure 4 shows the merging of the numbers of terms contained in every node. Then to merge the numbers, the system operates fast in one-pass.

In the system, the number of terms contained in a node is in 4 byte integer, and the maximum number of nodes contained in a node is in 2 byte integer (65,536 nodes). Therefore the system occupies about 315 MB in memory for the RIP list.

3 Experimental Results

3.1 INEX 2006 Adhoc Track and Indexing

The index of the system is made from the collection of XML 2006 Adhoc Track. First, the system parses all the structures of each XML document with XML parser and parses all the text nodes of each XML document. Then, the size of the index is about 8.32 GB, related to both document information and structure information. After that, the system uses the index in all the experiments.

3.2 Evaluation with INEX 2006

Our experiment targets for CO Task only, the system accepts CO queries, which are terms enclosed in <title> or <ontopic_keywords> tags. Then, there are Thorough Task, Focused Task, All In Context Task and Best In Context Task, in INEX 2006 Adhoc Track, and Focused Task only remains in INEX 2007. Thus the system are evaluated on Focused Task,

Table 1. Focused Task (Overlap=on)

Affiliation	nxCG@5	Rank
JSXIR	0.4057	-
cityuni	0.3944	1/106
lip6	0.3744	2/106
maxplanck	0.3696	3/106
maxplanck	0.3659	4/106
uhebrew	0.3547	5/106
uhebrew	0.3483	6/106

Table 2. Focused Task (Overlap=off)

Affiliation	nxCG@5	Rank
lip6	0.4708	1/106
lip6	0.4292	2/106
cityuni	0.4176	3/106
JSXIR	0.4143	-
uhebrew	0.4066	4/106
uhebrew	0.3900	5/106
uhebrew	0.3890	6/106

*ep-gr (Quantization:gen, Overlap=on).

*ep-gr (Quantization:gen, Overlap=off).

JSXIR means our experimental system. PC:
CPU Celeron 2GHz, RAM 2GB, HDD SATA
300GB; Implementation: Java 1.4.2.06.

3.3 Experimental Results

Table 1 and Table 2 show results of our system on Focused Task. In the results, our system has realized relatively high precisions. Then the system has retrieved in an average 3.92 seconds per a topic, and for fewer than 9.95 seconds per a topic.

4 Conclusions

In this paper, the means of high-speed processing for BUS has realized with the RIP list. The system with the RIP list takes a shorter time to retrieve XML nodes than ever, while the system uses various scores for an index. One reason for the fast search is with downsizing of structure information enough to be on memory. The other reason is the merging algorithm makes the time complexity $O(n)$, because the cost of searching for each parent node is vanishingly low.

In the evaluation of precision, the system has taken first place in ranking of precisions on Focused Task (Overlap=on) in INEX 2006. However, the system has not taken first place on every task in every evaluation measure. In the future, we want to research suitable scores for retrieving XML nodes, and develop a theory of scoring with a probabilistic approach.

References

1. Extensible Markup Language (XML) 1.1 (Second Edition). <http://www.w3.org/TR/xml11/>
2. XML Path Language (XPath) Version 1.0. <http://www.w3.org/TR/xpath>
3. XQuery 1.0: An XML Query Language. <http://www.w3.org/TR/xquery/>
4. Initiative for the Evaluation of XML Retrieval (INEX). <http://inex.is.informatik.uni-duisburg.de/>
5. Clarke, C., Kamps, J. and Lalmas, M.: INEX 2006 Retrieval Task and Result Submission Specification. http://inex.is.informatik.uni-duisburg.de/2006/inex06/pdf/INEX06_Tasks_v1.pdf
6. Kazai, G. and Lalmas, M.: INEX 2005 Evaluation Metrics, INEX 2005, pp. 16–29. <http://www.dcs.qmul.ac.uk/~7Emounia/CV/Papers/inex-2005-metrics.pdf>
7. Pehcevski, J., Kamps, J., Kazai, G., Lalmas, M., Ogilvie, P., Piwowarski, B. and Robertson, S.: INEX 2007 Evaluation Measures (Draft), <http://inex.is.informatik.uni-duisburg.de/2007/inex07/pdf/inex07-measures.pdf>

8. Baeza-Yates, R. and Ribeiro-Neto, B.: *Modern Information Retrieval*, Acm Press Series, Addison-Wesley, pp. 1–69, 141–162 (1999).
9. Salton, G., Wong, A. and Yang, C. S.: *A vector space model for automatic indexing*, Communications of the ACM, **18**, pp. 613–620 (1975).
10. Evans, D. A. and Lefferts, R. G.: Design and evaluation of the clarit-trec-2 system. *TREC*, pp. 137-150 (1993).
11. Amer-Yahia, S. and Lalmas, M.: XML search: languages, INEX and scoring, *SIGMOD Rec.*, ACM Press, Vol. 35 No. 4, pp. 16–23 (2006).
12. Shin, D., Jang, H. and Jin, H.: BUS: an effective indexing and retrieval scheme in structured documents, *DL '98: Proceedings of the third ACM conference on Digital libraries*, pp. 235–243 (1998).
13. Weigel, F., Meuss, H., Schulz, K. U. and Bry, F.: Content and structure in indexing and ranking XML, *WebDB '04 Proceedings of the 7th International Workshop on the Web and Databases*, pp. 67–72 (2004).
14. Tanioka, H.: A Method of Preferential Unification of Plural Retrieved Elements for XML Retrieval Task, *Comparative Evaluation of XML Information Retrieval Systems 5th International Workshop of the Initiative for the Evaluation of XML Retrieval.*, LNCS, Vol. 4518, Springer-Verlag, pp. 45–56 (2007).
15. Eugen, P., M enier, G. and Marteau, P.-F.: SIRIUS XML IR System at INEX 2006: Approximate Matching of Structure and Textual Content, *Comparative Evaluation of XML Information Retrieval Systems 5th International Workshop of the Initiative for the Evaluation of XML Retrieval.*, LNCS, Vol. 4518, Springer-Verlag, pp. 185–199 (2007).
16. Geva, S.: GPX - Gardens Point XML IR at INEX 2006, *Comparative Evaluation of XML Information Retrieval Systems 5th International Workshop of the Initiative for the Evaluation of XML Retrieval.*, LNCS, Vol. 4518, Springer-Verlag, pp. 137–150 (2007).
17. Hatano, K., Kikutani, H., Yoshikawa, M. and Uemura, S.: Determining the Retrieval Targets for XML Fragment Retrieval Systems Based on Statistical Information, *The IEICE transactions on information and systems*, Vol. J89-D, No. 3, pp. 422–431 (2006).
18. Shimizu, T., Onizuka, M., Eda, T. and Yoshikawa, M.: A Survey in Management and Stream Processing of XML Data, *The IEICE transactions on information and systems*, Vol. J99-D, No. 2, pp. 159-184 (2007).

How well does Best in Context reflect ad hoc XML retrieval?

James A. Thom¹ and Jovan Pehcevski²

¹ RMIT University, Melbourne, Australia
james.thom@rmit.edu.au

² MIT – Faculty of Information Technologies, Skopje, Macedonia
jovan.pehcevski@acm.org

Extended Abstract

This extended abstract describes the participation of the RMIT group in the Initiative for the Evaluation of XML retrieval (INEX) ad hoc track in 2007. Of the three tasks in the INEX 2007 XML ad hoc track: Focused, Relevant in Context (RiC), Best in Context (BiC), the RMIT system performed surprisingly well on the last task.

Our Approach

Our approach is limited to retrieval of articles using the Zettair³ search engine. Zettair is an open source search engine developed at RMIT, which we used to index the full text of Wikipedia articles and return complete articles ranked by their similarity score to the query. Zettair is “one of the most complete engines” according to a recent comparison of open source search engines [3]. Within Zettair we used the Okapi BM25 similarity measure which worked well on the INEX 2006 Wikipedia test collection [1].

For each of the Focused, RiC, and BiC tasks, we simply return the same ranked list of whole documents. Thus these Zettair runs can be seen as a baseline against which element or passage retrieval would be expected to do better.

Results

We present our results that investigate the effectiveness of document retrieval when applied to the three tasks in the INEX 2007 ad hoc track.

For the Focused retrieval task the RMIT system had an interpolated average precision at 0.01 recall of 0.3788 (compared with 0.4259 for the best performing system on this task) and was ranked 17 out of the 79 runs.

For the RiC task the RMIT system had a non-interpolated mean average precision (MAgP) of 0.0884 (compared with 0.1013 for the best performing system on this task) and was ranked 10 out of 66 runs.

For the BiC task the RMIT system had a non-interpolated mean average precision (MAgP) of 0.1951 and was surprisingly the top ranked run (out of 71 runs) for this task.

³ <http://www.seg.rmit.edu.au/zettair/>

Discussion

Looking at the results (as compared with other systems), document retrieval (using Zettair) seems to work well on the INEX Wikipedia XML collection. Only relatively small gains are made by the best systems using element or passage retrieval for the Focused and the RiC tasks. For the BiC task, it seems difficult to do better than returning the start of the document as the best entry point.

Why is this the case? Firstly, from the definition of the BiC task we are looking for retrieving relevant documents in the first place. Obviously, Zettair does a good job here (but we already know this from our INEX 2006 ad hoc experiments). Secondly, after locating a relevant document, the task asks systems to find the best entry point (BEP) to start reading the document. In their analysis of the INEX 2006 relevance assessments, Kamps et al. [2] observed that assessors would mainly choose the best entry point to be “some distance” from the start of the document; specifically, they observed the following:

“What we see is that the BEP is a fair distance into the article (median distance 556 [characters], mean distance 3,090 [characters]). The difference between median and mean distance signals that the distribution is skewed toward the start of the article. Comparing the BEP distance and the length of the article, we find a significant correlation of 0.66.”

Judging from the way Zettair performed, we suspect that this skew towards the start of articles is at least as great in the case of INEX 2007 relevance assessments as it was in the case of INEX 2006 relevance assessments. As we retrieve only articles with Zettair, it is therefore of no great surprise that we perform better than any of the other element or passage retrieval systems.

Acknowledgements

Most of this work was completed while James Thom was visiting INRIA and Jovan Pehcevski was working at INRIA.

References

1. D. Awang Iskandar, J. Pehcevski, J. A. Thom, and S. M. M. Tahaghoghi. Social media retrieval using image features and structured text. In *Comparative Evaluation of XML Information Retrieval Systems: Fifth Workshop of the INitiative for the Evaluation of XML Retrieval, INEX 2006*, volume 4518 of *Lecture Notes in Computer Science*, pages 358–372, 2007.
2. J. Kamps, M. Koolen, and M. Lalmas. Where to start reading a textual xml document? In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 723–724, New York, NY, USA, 2007. ACM.
3. C. Middleton and R. Baeza-Yates. A comparison of open source search engines. Technical report, Universitat Pompeu Fabra, Barcelona, Spain, 2007. <http://wrg.upf.edu/WRG/dctos/Middleton-Baeza.pdf>.

Dynamic Element Retrieval in the Wikipedia Collection

Carolyn J. Crouch, Donald B. Crouch, Nachiket Kamat, Vikram Malik, Aditya Mone

Department of Computer Science
University of Minnesota Duluth
Duluth, MN 55812
(218) 726-7607
ccrouch@d.umn.edu

Abstract

Our work for INEX 2007 centers on solving the interesting problems which arose for dynamic element retrieval when the experimental collection changed from IEEE to Wikipedia. Dynamic element retrieval—i.e., the dynamic retrieval of elements at the desired degree of granularity—has been the focus of our investigations at INEX for some time [1, 2]. We have demonstrated that our method works well for structured text and that it in fact produces a result virtually identical to that produced by the search of the same query against the corresponding all-element index [3]. The challenge is to adapt our methods to the particular issues presented by Wiki.

The well structured IEEE collection lends itself quite naturally to representation by Fox’s Extended Vector Space Model. Wikipedia documents, on the other hand, are semi-structured at best. They contain untagged text which is distributed throughout the documents. These documents can be nicely represented within the Vector Space Model; retrieval then takes place against an all-element index composed of articles, sections, and paragraphs (or terminal nodes). But they pose particular problems for dynamic element retrieval, which requires that all the terminal nodes of a document be identifiable. Since the process requires the execution time building of document trees of interest to the query, all of the terminal nodes or text-bearing elements of the tree must be present in order for their parent elements to be generated properly.

The impact of untagged text is twofold. During parsing, it must be identified, so that it may subsequently be used in generating the document schemas utilized by dynamic element retrieval as it builds the document trees. And since the method requires an initial retrieval against the terminal node index to identify the documents of interest to the query (i.e., those whose trees will be built), we must determine the value of untagged text in this context. In other words, is the untagged text distributed throughout a document (or interspersed among tagged elements) important from the retrieval viewpoint?

Experiments to answer this and other, related questions were performed during the past year. Results show that untagged text is absolutely as important as tagged text with respect to content and its impact on retrieval. Using the 2006 INEX test collection and evaluation metrics, we have established that dynamic element retrieval can be effectively applied to semi-structured collections, producing a result identical to that produced by the equivalent all-element retrieval. Moreover, the results produced by our methods (with the inclusion of a final step which expands the terminal node to return the paths of its embedded elements) are highly competitive with respect to both the Thorough and Focused (overlap off) subtasks. We are currently in the process of running the 2007 query set utilizing both all-element retrieval (baseline) and dynamic element retrieval for the Ad Hoc subtasks. It appears that this approach can be also be used to support passage retrieval, but this has yet to be proven.

References

- [1] Crouch, C., Khanna, S., Potnis, P., and Doddapaneni, N. The dynamic retrieval of XML elements. In Fuhr, et. al. (Eds): *Advances in XML Information Retrieval and Evaluation: Fourth Workshop of the Initiative for the Evaluation of XML Retrieval (INEX 2005)*, LNCS 3977, Springer, 2006, 268-281.

- [2] Crouch, C., Crouch, D., Ganapathibhotla, M., Bakshi, V. Dynamic element retrieval in a semi-structured collection. In Fuhr, et. al. (Eds): *Comparative Evaluation of XML Retrieval Systems* (INEX 2006), LNCS 4518, Springer, 2007, 82-88.
- [3] Crouch, C. Dynamic element retrieval in a structured environment. *ACM Transactions on Information Systems*, 24(4), 2006, 437-454.

Phrase detection in the Wikipedia

Miro Lehtonen¹ and Antoine Doucet^{1,2}

¹ Department of Computer Science
P. O. Box 68 (Gustaf Hällströmin katu 2b)
FI-00014 University of Helsinki
Finland

{Miro.Lehtonen, Antoine.Doucet} @cs.helsinki.fi

² GREYC CNRS UMR 6072,
University of Caen Lower Normandy
F-14032 Caen Cedex
France

Antoine.Doucet @info.unicaen.fr

Abstract. The Wikipedia XML collection turned out to be rich of marked-up phrases as we carried out our INEX 2007 experiments. Assuming that a phrase occurs at the inline level of the markup, we were able to identify over 18 million phrase occurrences, most of which were either the anchor text of a hyperlink or a passage of text with added emphasis. As our IR system — EXTIRP — indexed the documents, the detected inline-level elements were duplicated in the markup with two direct consequences: 1) The frequency of the phrase terms increased, and 2) the word sequences changed. Because the markup was manipulated before computing word sequences for a phrase index, the actual multi-word phrases became easier to detect. The effect of duplicating the inline-level elements was tested by producing two run submissions in ways that were similar except for the duplication. According to the official INEX 2007 metric, the positive effect of duplicated phrases was clear.

1 Introduction

In previous years, our INEX-related experiments have included two dimensions to phrase detection, one at the markup level [1] and another in the term sequence analysis [2]. The methods have been tested on plain text corpora and scientific articles in XML format. The Wikipedia XML documents are the first collection of hypertext documents where our phrase detection methods are applied.

Regarding marked-up phrases, the nature of the markup in a hypertext document differs from that in a scientific article. The phrases that are marked in scientific texts are mostly meant to be displayed with a different typeface, e.g. italicised or underlined, whereas hypertext documents have similar XML structures for marking the anchor text related to a hyperlink. Both emphasised passages and anchors are important, but whether they can be treated equally is still an open question.

The initial results support the idea that emphasised phrases and anchors are equal as long as they are marked with similar XML structures — inline-level elements.

2 EXTIRP baseline

The EXTIRP baseline without duplicated phrases is similar to our INEX 2006 submission [4] except for a few major bugs that have been fixed. The results are thus not comparable. First, EXTIRP scans through the document collection and selects disjoint fragments of XML to be indexed as atomic units. Typical fragments include XML elements marking sections, subsections, and paragraphs. In the Wikipedia, typical names for these elements are `article`, `section`, and `p`. The disjoint fragments are treated as traditional documents which are independent of each other. The pros include that the traditional IR methods apply, so we use the vector space model with a weighting scheme based on the $tf*idf$. The biggest of the cons is that the size of the indexed fragments is static, and if bigger or smaller answers are more appropriate for some query, the fragments have to be either divided further or combined into bigger fragments.

Second, two separate inverted indices are built for the fragments. A *word index* is created after punctuation and stopwords are removed and the remaining words are stemmed with the Porter algorithm [5]. The *phrase index* is based on Maximal Frequent Sequences (MFS) [6]. Maximal phrases of two or more words are stored in the phrase index if they occur in seven or more fragments. The threshold of seven comes from the computational complexity of the algorithm. Although lower values for the threshold produce more MFSs, the computation itself would take too long to be practical.

When processing the queries, we compute the cosine similarity between the document and the base term vectors which results in a `Word_RSV` value. In a similar fashion, each fragment vector gets a similarity score `MFS_RSV` for phrase similarity. These two scores are aggregated into a single RSV so that the aggregated $RSV = \alpha * \text{Word_RSV} + \beta * \text{MFS_RSV}$, where α is the number of distinct query terms and β is the number of distinct query terms in the query phrases.

3 Phrase detection and duplication

The steps from the original XML fragment to an intermediate XML format and, finally, the vector representation.

The definition of a *Qualified inline element*: An XML element is considered a qualified inline element when the corresponding element node in the document tree meets the following conditions:

- (1) The text node siblings contain at least n characters after whitespace has been normalised.
- (2) The text node descendants contain at least m characters after normalisation.
- (3) The element has no element node descendants.

- (4) The element content is separated from the text node siblings by word delimiters, e.g. whitespace or commas.

When the whitespace of a text node is normalised, all the leading and trailing whitespace characters are trimmed away.

Defining the lower bounds of n and m improves the quality of detected phrases in the qualified inline elements.

We set the parameters to a minimum of three (3) characters in at least one Text node child and a minimum of five (5) characters in at least one Text node sibling, so that $n = 5$ and $m = 3$.

4 Qualified inline elements in the Wikipedia XML

The most common elements that were duplicated are summarised in Table 1. The exhaustivity of an element type is the percentage of element occurrences duplicated out of all occurrences of that element.

XML Element	Count	Exhaustivity %	Percentage
collectionlink	12,971,384	76.2	69.1
unknownlink	2,372,870	60.0	12.6
emph2	1,339,345	49.2	7.1
emph3	992,373	67.0	5.3
p	282,438	10.3	1.5
outsidelink	230,675	26.8	1.2
title	222,917	14.0	1.2
languagelink	114,828	14.5	0.6
emph5	57,443	70.8	0.3
wikipedialink	42,009	23.8	0.2
All links	15,734,890	68.9	83.8
All emphasis	2,406,372		12.8
Total	18,784,132		100

Table 1. Distribution of the most frequent qualified inline elements by element type.

5 MFS extraction

In this section, we are comparing our runs from the point of view of the MFSs that were extracted. We conjecture that the phrase duplication process facilitates the extraction of the more useful sequences, hereby inducing better retrieval performance. We will try to confirm this by analysing the extracted sequence sets corresponding to our runs.

Statistics are summarized in Table 5. The frequency threshold was always of 7 occurrences, that is, a sequence was considered frequent if it occurred in at least 7 minimal units of a same document cluster.

Run	Clusters	Number of sequences (total freq)	Average length	Average Frequency
UHel-Run1	500	21,009,668	2.248	19.9
UHel-Run2	250	37,252,061	2.184	26.4

Table 2. Per run statistics of the extracted MFS sets (frequency threshold: 7).

The 10 most frequent phrases that were duplicated are shown in Table 5.

Frequency	Phrase
37,474	Native American
37,328	population density
37,047	African American
36,046	married couples
35,926	per capita income
35,829	other races
35,807	poverty line
35,764	Pacific Islander
32,974	United States Census Bureau
26,572	United States

Table 3. The 10 most frequent phrases that were duplicated.

6 Results

We submitted two runs for the adhoc track task of Focused retrieval. The initial results are shown in Table 4.

	Run1		Run2			Best official
Recall level	Rank	Score	Rank	Score	Improvement	Score
0.00	48	0.2641	39	0.3157	19.5%	0.4780
0.01	46	0.2439	36	0.2986	22.3%	0.3988
0.05	40	0.2075	35	0.2476	19.3%	0.3482
0.10	38	0.1751	35	0.1972	12.6%	0.3238

Table 4. Performance of submissions “UHel-Run1” and “UHel-Run2” measured with interpolated precision at four recall levels. A total of 58 submissions are included in the ranking.

7 Conclusion

Analysing the markup did not involve any information about the document type, such as element names or tag names, so the methods can be applied to any XML documents.

References

1. Lehtonen, M.: Preparing heterogeneous XML for full-text search. *ACM Trans. Inf. Syst.* **24** (2006) 455–474
2. Doucet, A., Ahonen-Myka, H.: Probability and expected document frequency of discontinued word sequences, an efficient method for their exact computation. *Traitement Automatique des Langues (TAL)* **46** (2006) 13–37
3. Doucet, A., Aunimo, L., Lehtonen, M., Petit, R.: Accurate Retrieval of XML Document Fragments using EXTIRP. In: *INEX 2003 Workshop Proceedings, Schloss Dagstuhl, Germany* (2003) 73–80
4. Lehtonen, M., Doucet, A.: Extirp: Baseline retrieval from wikipedia. In Malik, S., Trotman, A., Lalmas, M., Fuhr, N., eds.: *Comparative Evaluation of XML Information Retrieval Systems*. Volume 4518 of *Lecture Notes in Computer Science.*, Springer (2007) 119–124
5. Porter, M.F.: An algorithm for suffix stripping. *Program* **14** (1980) 130–137
6. Ahonen-Myka, H.: Finding all frequent maximal sequences in text. In Mladenic, D., Grobelnik, M., eds.: *Proceedings of the 16th International Conference on Machine Learning ICML-99 Workshop on Machine Learning in Text Data Analysis, Ljubljana, Slovenia, J. Stefan Institute* (1999) 11–17

Ranking Ad-hoc Retrieval using Summary Models and Structural Relevance

M. S. Ali, Mariano P. Consens, and Shahan Katchadourian

University of Toronto

{sali, consens, shahan}@cs.toronto.edu

Abstract. At INEX, there have been numerous proposals for how to incorporate structural constraints and hints into ranking. These proposals have introduced novel ways to either boost the score or filter out elements that have desirable structural properties. In this paper, we propose an alternative approach that is able to express user preferences in the scoring of search results, and provides a reasonable way to apply these methods across different collections. The proposal is to use summary graph techniques to describe how a user structurally characterizes a collection, and then, based on the summary, we quantify the relative isolation of elements, in order to score elements that are (i) content-wise relevant to a user, (ii) structurally relevant (*i.e.*, contextualized) to a user, and (iii) isolated within the collection from other elements. Ostensibly, this approach introduces a single, big parameter into scoring. Our results suggest that this approach can improve search effectiveness, and that the methodology developed can be applied to structural scoring across XML collections.

1 Introduction

INEX is a forum dedicated to research in information retrieval from collections of XML documents. The INEX 2007 Ad-hoc Track highlights the comparison of element retrieval to passage retrieval in focused retrieval. The focused task constrains results to relevant elements that are the most focused on the information need. Focused results may not contain overlapping elements. So, the challenges in the task are to identify where relevant text appears in the collection; and then identify the appropriate size of the element to return that contains the text [1].

In this paper, we explore element retrieval using a novel structural approach which combines keyword search with structural boosting to find where the relevant text occurs; and then we apply structural relevance to our candidate retrieval elements to identify the most appropriate elements to return to the user. We show how the structure can be used in content-only search to dynamically boost elements with structurally desirable properties, and then how the overlap in the system output is resolved using a post-processor to find the structurally most relevant ranked list of elements for output.

A number of existing approaches to structural retrieval have relied on rote return structures and ad-hoc tuning parameters to score elements. For instance,

a naive approach assumes that XML documents are structured as articles, and so only logical elements such as articles, sections and paragraphs are returned in the search results. NEXI is a notation for expressing XML queries that includes structural constraints and hints [6]. Another approach is to use XPATH to retrieve strict XML structural paths according to what the user specifies in a NEXI query. More sophisticated approaches to structural retrieval use element weighting schemes in scoring to control overlap based on element structure to re-rank results [2, 3]. By and large, the parameterization of these new methods have involved the development of ad-hoc heuristics based on empirical user studies. The development and use of these methods requires a significant effort to conduct user studies, and it is a challenge to apply a given method across different collections. It has been suggested that the reason that this challenge arises is because users are not very good at specifying structure in their queries. In fact, preliminary work at INEX has suggested that the best structural elements are a function of the document collection and not the user's query [5].

In our approach, we use the document collection to derive a model of the user based on an XML summary of the collection. We quantify the user model in terms of a novel concept called isolation [4], which is a measure of the probability that in a given collection a random reviewer will be browsing in a particular set of XML elements. We generate an XML summary based on a bijection of the collection into partitions, where each partition represents a set of XML path expressions. Then, based on the summary partitions, we approximate the isolation for all elements. In this proposal, we show how the isolation of partitions can be used in the search engine to boost elements with desirable structural properties, and then we show how isolation can be used on the entire ranked list to both find the best ranking of elements and to remove overlap in the results.

2 Post-Processing for Focused Retrieval

2.1 System Overview

Our search engine is based on Apache Lucene. Lucene is also used to index the collection and generate the summary. As tokens are indexed, the payload information associated with each token occurrence contains the summary partition in which the token appears. The payload information of each token is accessible during scoring and is used in conjunction with the boost parameters. The boost parameters are calculated using the isolation of the summary partitions (which is described in the next section 2.2) and is based on the extent size of each partition. A PayloadTokenizer has been to add the partition payload to the indexed tokens obtained from a sequence of Lucene's default tokenizers; namely, LowerCaseFilter, StopFilter, and LetterTokenizer. The LowerCaseFilter makes all tokens lowercase, the StopFilter exclude a set of stop words from indexing, and the LetterTokenizer removes certain punctuation symbols. Our system uses several indexes that represent elements as document units. The elements selected for the document units are taken from the structural hints in the NEXI queries

of each topic. This allows the term frequencies, document frequencies, as well length normalization to be affected on a subgraph level.

2.2 Isolation

The isolation of summary partitions was used to generate the boost parameters as well as to remove the overlap from the system output. The measures used to generate the isolation of the p^* summary, in which each measure is generated for each partition in the summary, were based on the extent size of each partition.

The isolation of a summary partition is the probability of a user being in some summary partition i while browsing the collection. We denote the isolation as π_i , and we calculate it by using the steady-state probabilities of a time-reversible discrete Markovian process applied to the summary,

$$\pi_i = \frac{\sum_j w_{ij}}{\sum_i \sum_j w_{ij}} \quad (1)$$

where $i, j \in S$ are partitions of the summary, and w_{ij} is the size of the extent of the child node among the partitions i and j . We interpret π_i as the fraction of time that a user who uses a description of the document structure (*i.e.* a summary) to browse will spend π_i of their time in partition i of the document.

2.3 Re-ranking Results Based on Isolation

The approach adopted in this proposal was to first search for exhaustive results across all indexes, and then to combine the results across all indexes into a single weakly-ordered, overlapped ranked list R which would be processed in two stages. The first stage of processing involved finding the most structurally relevant strictly-ordered permutation R^* of the ranked list R . The second stage of processing was to produce the final output by removing overlaps from the most structurally relevant ranked list R^* . We refer to the j -th permutation of ranked list R as $R^{(j)}$.

$$\ell = |\Omega| = \prod_{i=1}^m |R_i|! \quad (2)$$

The number of permutations to be evaluated in this first stage is calculated using equation 2, where m is the number of ranks in R and $|R_i|$ is the number of elements in rank i of the list R . Each list is then evaluated for structural relevance in precision (*i.e.*, the isolation of elements based on their order in the list and how they are structurally related to one and other). The highest scoring list is selected for further processing. It should be noted that the highest score could be shared by more than one permutation, and, in those cases, we selected the first permutation found for further processing.

Structural relevance for element u in ranked list R is calculated based on the rank of the element, and the elements in R that are higher-ranked and overlapped

to u [ref]. Equation 3 shows how structural relevance SR is calculated. $R[u]$ is the ranked list up to the rank of element u . $rel(e)$ is the relevance of the element e . $R[u]_{(e)}$ are the set of elements in the same rank as e in the ranked list $R[u]$. $ov(R[u]_{(e)}, e)$ is the set of overlapped elements in the same rank as e in $R[u]$ that are overlapped with element e . m is the number of higher ranked, overlapped elements in $R[u]$ to element e . Finally, every element in the collection belongs to a partition in the summary. It has been shown that the isolation of elements can be approximated using the isolation of summary partitions. Let $\pi_{(e)}$ denote the isolation of the summary partition of element e .

$$SR[u] = \sum_{e \in R[u]} \frac{rel(e)}{|R[u]_{(e)}|} \sum_{n=1}^{|ov(R[u]_{(e)}, e)|} \pi_{(e)}^{n+m-1} \quad (3)$$

In the first stage, using a given summary S of the collection, every strictly ordered permutation of R was evaluated for structural relevance in precision (SRP) with the assumption that all elements were relevant. We calculate SRP for a ranked list R where k is the top- k with,

$$SRP(R) = frac1k \cdot \sum_{u \in R} SR[u] \quad (4)$$

Algorithm 1, below, shows the algorithm that we used to determine the most structurally relevant list R^* . We serially evaluate all permutations of R until we find the highest score for SRP.

Algorithm *FindMostIsolatedList*

Input: Summary of collection (π) and a weakly-ordered overlapped ranked list R .

Output: A non-overlapping strictly-ordered ranked list R^*

- 1: let Ω be the set of strictly-ordered permutations of R .
- 2: let ℓ be the number of permutations of R .
- 3: let R^* be the highest scoring permutation of R .
- 4: let $high$ be the highest SR score found.
- 5: $high = 0$
- 6: **for** $j = 1$ to ℓ **do**
- 7: let $R^{(j)}$ be the j -th permutation of R
- 8: let $score = SRP(R^{(j)})$, /* see eq. 4 */
- 9: **if** $score > high$ **then**
- 10: $R^* = R^{(j)}$
- 11: **end if**
- 12: **end for**

Fig. 1. Find the most isolated list

In our evaluation of structural relevance in precision for post-processing, we have assumed that all returned elements are relevant. If there were other criteria

other than isolation for post-processing a list then it would be desirable to loosen this assumption and allow for a broader range of output lists. Using SRP, we would use thresholds to evaluate SRP with either a maximum desired precision (*i.e.*, $score > constant$), or we would evaluate SRP up to a given rank level (*i.e.*, evaluate SRP for all elements that are ranked higher than some constant rank).

The second stage of post-processing removes the overlap in the most relevant list R^* . The first stage removed tied ranks. In the second stage, we are interested in resolving the overlap for focused element retrieval. Sibling elements are allowed in the results, but ancestor-descendant relationships between elements are not allowed. We implemented a simple rule that would choose the highest ranked ancestor-descendant element for final output, and remove all lower ranked elements from the final output. This is a naive approach, which is based on the best ordering of tied ranks which was established in the first stage of processing.

3 Experimental Results

Our experimental results have shown that boosting at the partition level using isolation does improve results, as compared to simply retrieving elements using keyword search across partitions. We intend to elaborate on the experiment in later versions of this paper.

4 Conclusion

We have presented a general methodology for introducing structural constraints into element retrieval where the parameterization of our model allows for complex modelling of user behaviour based on summary representations of the collection, and quantified with the relative isolation of partitions. Our approach does not make any assumptions about the collection, and can be easily and quickly employed for searching any XML collection. The experimental results suggest that this structural approaches can improve, and it agrees with the observation of Trotman and Lalmas that the effectiveness of structural search is dependent on the collection itself and not the proficiency of users at large to express structural hints and constraints.

References

1. S. G. A. Trotman. Passage retrieval and other xml-retrieval tasks. In *Proc. SIGIR 2006 Workshop on XML Element Retrieval Methodology*, pages 43–50, 2006.
2. C. Clarke. Controlling overlap in content-oriented XML retrieval. In *SIGIR '05: Proc. of the 28th Ann. Intl. ACM SIGIR Conf. on Res. and Dev. in IR*, pages 314–321, New York, NY, USA, 2005. ACM Press.
3. L. Guo, F. Shao, C. Botev, and J. Shanmugasundaram. XRANK: Ranked keyword search over xml documents. In *ACM SIGMOD*, New York, NY, USA, 2003. ACM Press.

4. M. C. M. A. . M. Lalmas. Structural relevance in xml retrieval evaluation. In *SIGIR 2007 Workshop on Focused Retrieval, Amsterdam, The Netherlands, July 27, 2007*, 2007.
5. A. Trotman and M. Lalmas. Why structural hints in queries do not help xml-retrieval. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 711–712, New York, NY, USA, 2006. ACM.
6. A. Trotman and B. Sigurbjornsson. Narrowed extended XPath I (NEXI). In *Proc. INEX Workshop*, pages 16–39, 2004.

Probabilistic document model integrating XML structure

Mathias Géry, Christine Largeron and Franck Thollard

Jean Monnet University,
Laboratoire Hubert Curien, UMR CNRS 5516, Saint-Etienne
{Mathias.Gery,Christine.Largeron,Thollard}@univ-st-etienne.fr

Abstract. While representing textual document, different approaches have been used: models based on boolean model, algebraic models extended from vector space model or probabilistic models.

In text mining as in information retrieval, these models have shown good results about textual documents modeling. They nevertheless do not take into account documents structure. In many applications however, documents are inherently structured (e.g. XML documents).

In this article¹, we propose an extended probabilistic representation of documents in order to take into account a certain kind of structural information: tags that represent logical structure and layout structure of the document. Our approach includes a learning step in which the weight of each tag is estimated. This weight is related to the probability a given tag is able to distinguish the relevant terms. Our model has been evaluated during INEX 2006 & 2007 evaluation campaign.

1 Introduction

In Information Retrieval as in text mining many approaches are used to model documents. As stated in [1], these approaches can be organized in three families: models based on boolean model, for example fuzzy or extended boolean model; models based on vector space model; probabilistic models. The later holds Bayesian networks, inference networks or belief networks. All these models appear to be appropriate to represent textual documents. They were successfully applied in categorization task or in information retrieval task.

However they all present the drawback of not taking into account the structure of the documents. It appears nevertheless that most of the available information either on the Internet or in textual databases are strongly structured. This is for example the case for scientific articles in which a title, an abstract, keywords, introduction, conclusion and other sections do not have the same importance. This is also true for the documents available on the Internet as they are written in languages (e.g. HTML or XML) that explicitly describe the logical structure of the document and a part of the layout structure (e.g. font size, color, ...).

For all these documents, the information provided by structure can be useful to emphasize some particular part of the textual document.

¹ This work has been partly funded by the Web Intelligence project (région Rhône-Alpes).

Consequently a given word does not have the same importance depending on its position in the article (*e.g.* in the title or in the body) or if it is emphasized (bold font, etc.). Indeed, if the author of a web page deliberately writes a given word in a particular font, it could be thought that a particular information can be associated with the term and therefore that the term should be considered differently.

For all these reasons, recent works in information retrieval as in text mining, focused on considering documents structure.

This leads, in particular, to content oriented XML information retrieval (RI) that aims at taking advantage of the structure provided by the XML tree. Taking into account the structure can be done either at the indexing step or at the querying one. In the former [4, 18, 13], a structured document is indexed using a tree of logical textual fragments. The terms weight in a given fragment is propagated through the structural relation, *i.e.* from leaves to the root or from root to leaves. In the later [9], SQL query language has been adapted to the structured context in order to allow queries like "I look for a paragraph dealing with running, included in an article that deals with the New-York marathon and in which a photo of a marathon-man is present". The INEX competition (INitiative for Evaluation of XML Retrieval) provides, since 2002, large collections of structured documents. Systems are evaluated through their ability to find relevant part of documents associated with XML fragment rather than the whole documents.

Taking advantage of the structure has also been studied in supervised and unsupervised document clustering tasks [15, 2, 3, 17, 16]. In such a context, many strategies appear. Among them is the extension of the usual document representations. For example, Doucet and Ahonen-Myka [5] generalized the vector space model by considering terms as well as tags. The results are not yet convincing. The authors nevertheless argued that the way the textual and structural information are combined is responsible for these poor results.

Other approaches have been developed based on the tree-like structure of XML documents. In this structure, leaves contain the textual information and nodes the structural one (tags or XML elements) [15, 7, 2, 3]. The documents can then be modeled by flattening their trees into sets of paths. The structured vector space model of [19] takes advantage of this representation. In this model, the components can be terms or another structured vector. In the same way [17] generalizes the previous model by introducing parameters that constrain (for example) the paths length or the choice of the beginning and the end of the path. CBCS and CBNCS are Bayesian classification models that take into account the tree-like representation structure in a recursive way [10]. The main problem with all these approaches is the number of parameters that need to be tuned, number which increases with collection size and heterogeneity. This limits the application of such models on collections available on the web.

In the context of novelty detection, other works took into account the logical structure of the documents, by associating a weight to each part of the document [8].

In this article, we propose to extend the probabilistic model in order to take into account the document structure (either the logical structure or the layout aspect). Our approach is made up of two steps, the first one being a learning step, in which a weight is computed for each tag. This weight is estimated based on the probability that a given tag distinguishes relevant terms. In the second step, the above weight is used to better estimate the probability for a document to be relevant for a given query.

An overview of our model is presented in the next section. A more formal one follows in section 3. The preliminary results obtained on the INEX 2006 and 2007 collections are then presented in section 4.

2 Integrating tags into document modeling

In Information Retrieval, the probabilistic model [12] aims at estimating the relevance of a document for a given query through two probabilities: the probability of finding a relevant information and the probability of finding a non relevant information.

These estimates are based on the probability for a given term in the document to appear in relevant (or in non relevant) documents. This estimation can be done using a training collection in which the documents relevance according to some query is available. With such a collection, one can estimate the probability for a given term to belong to a relevant (respectively non relevant) document, given its distribution in relevant (respectively non relevant) documents.

This probabilistic model leads to good results in textual information retrieval. Our goal here is to extend this model by taking into account the documents structure. Different kinds of "structure" can be considered. As an example, Fourel defined physical structure, layout structure, linguistic structure, discursive structure and logical structure [6]. In our model, we only consider the structure defined through XML tags: logical structure (title, section, paragraph, ...) and layout structure (bold font, centered text, ...).

Integrating the structure in the probabilistic model is done at two levels :

- In the first one, the logical structure is used in order to select the XML elements (section, paragraph, table, ...) that are considered at the indexing step.
- In the second one, tags describing layout structure are integrated into the classic probabilistic model.

Integrating tags needs a preliminary step in which a weight for each tag is computed. This weight is based on the probability, for a given tag, to distinguish relevant terms from non relevant ones. This is closely related to the classic probabilistic model, in which a weight for each term is estimated, based on the probability for the term to appear in relevant documents. But in our approach, tags are considered instead of terms and terms instead of documents. Moreover, the relevance is not evaluated on the whole document but on its parts (term by term). Accordingly, in the INEX collection, the relevance is defined on structural fragments, i.e. XML elements and parts of them (i.e. sentences). In our model, we do not consider the relevance of sentences, but only the relevance of XML elements.

In the second step, the probability for a document element to be relevant is estimated by taking into account the classic weight of the terms it contains, modified by the weight of the tags included in the element.

A more formal presentation of our model is given in the next section.

3 A probabilistic model for the representation of structured documents

3.1 Notations and examples

Let \mathcal{D} be a set of structured documents.

In practice, XML documents are considered. Each logical element (section, paragraph, etc.) e_j of the XML tree will therefore be represented by a set of terms. For example, we consider the following three documents D_0 , D_1 and D_2 :

D_0	D_1	D_2
<pre><article> <p> t₁t₂t₃ </p> <section> <p> t₁t₄ </p> <p> t₂t₅ </p> </section> </article></pre>	<pre><article> <section> <p> t₂t₄ </p> <p> t₂t₅ </p> </section> <p> t₂t₁ </p> </article></pre>	<pre><article> <section> <p> t₅ </p> <p> t₃t₄ </p> <p> t₃t₅ </p> </section> </article></pre>

Each tag describing logical structure defines elements corresponding to part of document which will be indexed. In the example, document D_2 is indexed by five elements: an article (tag $\langle \text{article} \rangle$), a section (tag $\langle \text{section} \rangle$) and three paragraphs (tag $\langle \text{p} \rangle$).

We note :

- $E = \{e_j, j = 1, \dots, l\}$, the set of the logical elements available in the collection (*article*, *section*, etc.).
- $T = (t_1, \dots, t_i, \dots, t_n)$, a term index built from E .
- $B = \{b_1, \dots, b_k, \dots, b_m\}$, the set of tags.

Let E_j , be a vector of random variables T_{ij} in $\{0, 1\}$:

$$E_j = (T_{10}, \dots, T_{1k}, \dots, T_{1m}, \dots, T_{i0}, \dots, T_{ik}, \dots, T_{im}, \dots, T_{n0}, \dots, T_{nk}, \dots, T_{nm})$$

$$\text{with } \begin{cases} T_{ik} = 1 & \text{if the term } t_i \text{ appears tagged by } b_k \\ T_{ik} = 0 & \text{if the term } t_i \text{ does not appear tagged by } b_k \\ T_{i0} = 1 & \text{if the term } t_i \text{ appears without being tagged by a tag in } B \\ T_{i0} = 0 & \text{if the term } t_i \text{ does not appear without being tagged} \end{cases}$$

We note $e_j = (t_{10}, \dots, t_{1k}, \dots, t_{1m}, t_{i0}, \dots, t_{ik}, \dots, t_{im}, t_{n0}, \dots, t_{nk}, \dots, t_{nm})$ a realization of the random variable E_j .

In the previous example with three documents D_0, D_1 and D_2 , we have $b_1 = \textit{article}$, $b_2 = \textit{section}$, $b_3 = p$, $b_4 = b$ and $T = \{t_1, \dots, t_5\}$.

The element $e_1: \langle p \rangle t_1 t_2 t_3 \langle /p \rangle$ of D_0 can be represented by the vector

$$\{t_{10}, t_{11}, t_{12}, t_{13}, t_{14}, \dots\} = \{0, 1, 0, 1, 0, \dots\}$$

since the term t_1 is tagged by *article* ($t_{11} = 1$), and p ($t_{13} = 1$) but neither by *section* ($t_{12} = 0$) nor by b ($t_{14} = 0$). We have $t_{10} = 0$ since the term does not appear without tag.

Given this representation, the goal is now to propose an extension of the probabilistic model that will take into account the documents structure.

3.2 Term based probability for an XML element to be relevant

The weighting function BM25 [12], is broadly used in probabilistic information retrieval systems to evaluate the weight of a term t_i in an element XML e_j . This weight is noted w_{ij} .

3.3 Tag based probability for an XML element to be relevant

The probabilities estimations are based on the model introduced in [12]. Nevertheless they have to be adapted in order to take into account the documents structure described in section 3.1. So, we consider not only term weights but also tag based weights.

In an information retrieval context, we want to estimate the relevance of an XML element e_j given a query. We thus want to estimate:

$P(R|e_j)$: the probability to find a relevant information in e_j given a query.

$P(NR|e_j)$: the probability of finding a non relevant information in e_j given a query.

Let $f_1(e_j)$ be a document ranking function:

$$f_1(e_j) = \frac{P(R|e_j)}{P(NR|e_j)}$$

The higher $f_1(e_j)$, the more relevant the information presented in e_j . Using Bayes formula, we get:

$$f_1(e_j) = \frac{P(e_j|R) \times P(R)}{P(e_j|NR) \times P(NR)}$$

The term $\frac{P(R)}{P(NR)}$ being constant over the collection for a given query, it will not change the ranking of the documents. We therefore define f_2 – which is proportional to f_1 – as:

$$f_2(e_j) = \frac{P(e_j|R)}{P(e_j|NR)}$$

Using the Binary Independence Model assumption, we have:

$$P(E_j = e_j|R) = \prod_{t_{ik} \in e_j} P(T_{ik} = t_{ik}|R) \quad (1)$$

$$= \prod_{t_{ik} \in e_j} P(T_{ik} = 1|R)^{t_{ik}} \times P(T_{ik} = 0|R)^{1-t_{ik}} \quad (2)$$

In the same way, we get :

$$P(E_j = e_j|NR) = \prod_{t_{ik} \in e_j} (P(T_{ik} = 1|NR))^{t_{ik}} \times (P(T_{ik} = 0|NR))^{1-t_{ik}} \quad (3)$$

For the sake of simplified notations, we note, for a given XML element:

- $p_0 = P(T_{i0} = 0|R)$: the probability that t_i does not appear given a relevant element.
- $p_{ik} = P(T_{ik} = 1|R)$: the probability that t_i appears, tagged by b_k given a relevant element.
- $q_0 = P(T_{i0} = 0|NR)$: the probability that t_i does not appear given a non relevant element.
- $q_{ik} = P(T_{ik} = 1|NR)$: probability that t_i appears tagged by b_k given a non relevant element.

Using these notations in equations 2 and 3, we get:

$$P(e_j|R) = \prod_{t_{ik} \in e_j} (p_{ik})^{t_{ik}} \times (1 - p_{ik})^{1-t_{ik}},$$

$$P(e_j|NR) = \prod_{t_{ik} \in e_j} (q_{ik})^{t_{ik}} \times (1 - q_{ik})^{1-t_{ik}}.$$

The ranking function $f_2(e_j)$ can then be re-written:

$$f_2(e_j) = \frac{\prod_{t_{ik} \in e_j} (p_{ik})^{t_{ik}} \times (1 - p_{ik})^{1-t_{ik}}}{\prod_{t_{ik} \in e_j} (q_{ik})^{t_{ik}} \times (1 - q_{ik})^{1-t_{ik}}}$$

The \log function being monotone increasing, taking the logarithm of the ranking function will not change the ranking. We can then define f_3 as:

$$\begin{aligned} f_3(e_j) &= \log(f_2(e_j)) \\ &= \sum_{t_{ik} \in e_j} (t_{ik} \log(p_{ik}) + (1 - t_{ik}) \log(1 - p_{ik}) - t_{ik} \log(q_{ik}) - (1 - t_{ik}) \log(1 - q_{ik})) \\ &= \sum_{t_{ik} \in e_j} t_{ik} \times \left(\log\left(\frac{p_{ik}}{1 - p_{ik}}\right) - \log\left(\frac{q_{ik}}{1 - q_{ik}}\right) \right) + \sum_{t_{ik} \in e_j} \log\left(\frac{1 - p_{ik}}{1 - q_{ik}}\right) \end{aligned}$$

As before, the term $\sum_{t_{ik} \in e_j} \log\left(\frac{1-p_{ik}}{1-q_{ik}}\right)$ is constant with respect to the collection (independent of t_{ik}). Not considering it will not change the ranking provided by $f_3(e_j)$:

$$f_{tag}(e_j) = \sum_{t_{ik} \in e_j} t_{ik} \log\left(\frac{p_{ik}(1-q_{ik})}{q_{ik}(1-p_{ik})}\right) \quad (4)$$

The weight of a term t_i tagged by b_k will be written $w'_{ik} : w'_{ik} = \log\left(\frac{p_{ik}(1-q_{ik})}{q_{ik}(1-p_{ik})}\right)$

Finally, in our probabilistic model that takes into account the document structure, the relevance of an XML element e_j is defined through $f_{tag}(e_j)$:

$$f_{tag}(e_j) = \sum_{t_{ik} \in e_j} t_{ik} \times w'_{ik}$$

In practice, we have to estimate the probabilities p_{ik} and q_{ik} , $i \in \{1, \dots, n\}$, $k \in \{0, \dots, m\}$ in order to evaluate the element relevance. For that purpose, we used a learning set EA in which elements relevance for a given query is known. Given the set R (respectively NR) that contains the relevant elements (respectively non relevant ones) a contingency table can be built for each term t_i tagged by b_k :

	R	NR	EA
$t_{ik} \in e_j$	r_{ik}	$n_{ik} - r_{ik}$	n_{ik}
$t_{ik} \notin e_j$	$R - r_{ik}$	$N - n_{ik} - R + r_{ik}$	$N - n_{ik}$
Total	R	$N - R$	N

with:

- r_{ik} : the number of relevant terms t_i tagged by b_k in EA;
- $\sum_i r_{ik}$: the number of relevant terms tagged by b_k in EA.
- n_{ik} : the number of terms t_i tagged by b_k in EA;
- $r'_{ik} = n_{ik} - r_{ik}$: the number of non relevant terms t_i tagged by b_k in EA;
- $R = \sum_{ik} r_{ik}$: the number of relevant terms in EA;
- $N-R = \sum_{ik} r'_{ik}$: the number of non relevant terms in EA.

We can now estimate $\begin{cases} p_{ik} = P(t_{ik} = 1|R) = \frac{r_{ik}}{R} \\ q_{ik} = P(t_{ik} = 1|NR) = \frac{n_{ik} - r_{ik}}{N - R} \end{cases}$

Given the unbiased estimators p_{ik} and q_{ik} (the probability that t_i is tagged b_k respectively in a relevant and non-relevant element), we can estimate $p.k$, the probability of having b_k given a relevant element, and $q.k$ the probability of having a tag b_k given a non relevant element.

$$p.k = \sum_i p_{ik} \quad \text{and} \quad q.k = \sum_i q_{ik}$$

3.4 Combining term based and tag based scores

In order to obtain the score $fc(e_j)$ of an element e_j given a query, our first attempt was to multiply the weight w_{ij} of each term in e_j with the average weights w'_{ik} of the tags that label these terms:

$$fc(e_j) = \sum_{t_i \in e_j} w_{ij} * \prod_{k/t_{ik}=1} w'_{ik}$$

We can note that some tags will reinforce the weight of the term ($w'_{ik} > 1$) while other will weaken it ($w'_{ik} \leq 1$).

Once the w'_{ik} are computed, we experiment two ways of considering tags. The first, called RSPM (for Reinforced Structured Probabilistic Model), only considers tags that reinforce the terms $w'_{ik} > 1$. The second, called SPM (for Structured Probabilistic Model, considers all the tags.

These strategies have been evaluated on the INEX collection.

4 Experiments on INEX 2006 & 2007 collection

4.1 INEX collection

We used for our experimentations the INEX (Initiative for Evaluation of XML Retrieval) collection as it contains a significant amount of data together with the availability of relevant assessments.

The corpus contains 659.388 articles in english, from the free Wikipedia encyclopedia. The documents are strongly structured as they are composed of 52 millions XML fragments. Each XML article view as a tree contains, on average, 161 elements for an average deep of 6.72. Moreover, whole articles (textual content + XML structure) represent 4.5 Gb while the textual content weights only 1.6 Gb. The structural information thus represents more than twice the size of the textual one.

In order to evaluate information retrieval systems, a set of queries is submitted by the participants during INEX 2006 and 2007 competition. 114 queries were selected in 2006, and 130 in 2007.

4.2 Experimental protocol

The 2006 INEX campaign made available the relevance assessments of the 114 queries. The corpus enriched by these assessments is used as a training set in order to estimate the w'_{ij} weights.

The second phase then consists in processing the queries. The vector space model using BM25 weighting function is used as the baseline, without stemming nor stoplist. In order to understand the pro and cons of our structured document model, BM25 is also used as the term weighting function before integrating the tags weight.

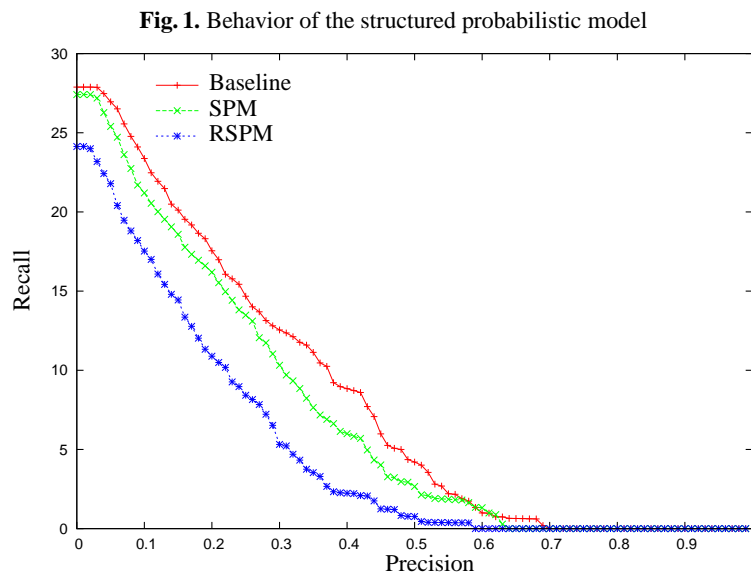
Two sets of evaluations were made: one on the 114 queries of the 2006 campaign, another one on the 130 queries of the 2007 campaign. The evaluation measures used are the *precision* and *recall* measures as defined by [14].

The *interpolated average precision* (iAP), introduced by INEX, combines *precision* and *recall*, and provides an evaluation of the system results for each query. By averaging the iAP values on the set of queries, an overall measure of performance is defined [11]. This average is called *interpolated mean average precision* (iMAP).

4.3 Results and discussion

We now compare the results obtained on the 114 queries of the INEX 2006 evaluation campaign using our baseline and the two variants of our structured probabilistic model. We obtain an iMAP of 2.34% for the baseline (i.e. without the structure). The Reinforced Structured Probabilistic Model, RSPM, obtains an 1.08% iMAP while the simple Structured Probabilistic Model, SPM, obtains an 1.80% iMAP.

These results are confirmed while considering precision and recall independently as seen on figure 1.



During the INEX 2007 campaign only two runs were sent : baseline and SPM. The baseline obtains a 4.44% iMAP, while SPM obtains a 2.19% iMAP.

Table 1 shows interpolated precision at several recall levels.

Even if the evaluated models do not outperform the baseline, we are still convinced that the structural information must be taken into account. Actually, the important information here is that SPM outperforms RSPM. This means that some tags informs us that the terms they contain brings less information than terms in other part. Regarding the fact that the baseline outperforms the two other methods, we think this could come from

Table 1. Result on the 130 queries of the 2007 campaign

	@0	@0.01	@0.05	@0.10	iMAP
Baseline (BM25)	34.90	27.49	17.49	13.39	4.44%
SPM	17.03	14.53	10.51	6.28	2.19%

the way we combine weights shared by the baseline (namely the w_{ij}) and the weights derived from the tag analysis (namely the w'_{ik}). We are thus confident in our model and have already started a deeper analysis of the results on the 2007 evaluation campaign.

5 Conclusion

In this article, we have proposed to extend the probabilistic model for representing documents in order to take the structural information of the documents into account. Our approach divides into two steps: a learning step where part of the collection is considered in order to estimate and quantify the impact of a given tag regarding the relevance of the tagged fragment. A second step in which the weight of a term (computed with a classical BM25 weighting) is combined with the information provided at the first step.

Preliminary results were obtained on the INEX 2006 and 207 evaluation campaign. It appears that the rather naive method used to combine the term weight and the tag information is too rough. Some work is still needed here, as these two pieces of information are not of the same type. We thus have to consider more elaborate combining of the information.

References

1. R. Baeza-Yates and B Ribeiro-Neto. *Modern information retrieval*. Addison-Wesley, 1999.
2. G. Costa, G. Manco, R. Ortale, and A. Tagarelli. A tree-based approach to clustering xml documents by structure. In *PKDD*, pages 137–148, 2004.
3. T. Dalamagas, T. Cheng, K. Winkel, and T. Sellis. Clustering xml documents using structural summaries. In *In Proc. of ClustWeb - International Workshop on Clustering Information over the Web in conjunction with EDBT 04*, 2004.
4. B. Defude. *Etude et réalisation d'un système intelligent de recherche d'informations : Le prototype IOTA*. PhD thesis, Institut National Polytechnique de Grenoble, Janvier 1986.
5. A. Doucet and H. Ahonen-Myka. Naive clustering of a large xml document collection. In *Proceedings of the First Workshop of the Initiative for the Evaluation of XML Retrieval (INEX)*, pages 81–87, Schloss Dagsuhl, Germany, 2002.
6. F. Fourel. *Modélisation, indexation et recherche de documents structurés*. PhD thesis, Université de Grenoble 1, France, 1998.
7. F.D. Francesca, G. Gordano, R. Ortale, and A. Tagarelli. Distance-based clustering of xml documents. In *Proceedings of the first workshop on mining graphs, trees and sequences, ECML/ PKDD'03 Workshop*, pages 75–78, 2003.
8. F. Jacquenet and C. Largeron. Using the structure of documents to improve the discovery of unexpected information. In *SAC*, pages 1036–1042, 2006.
9. D. Konopnicki and O. Schmueli. W3qs : A query system for the world-wide web. In *21ème International Conference on Very Large Data Bases (VLDB95)*, pages 54–65, Septembre 1995.

10. P.F. Marteau, G. Ménier, and L. Ekamby. Apport de la prise en compte du contexte structurel dans les modèles bayésiens de classification de documents semi-structurés. In *Revue des Nouvelles Technologies de l'Information, numéro spécial sur la fouille de données complexes*, 2005.
11. J. Pehcevski, J. Kamps, G. Kazai, M. Lalmas, P. Ogilvie, B. Piwowarski, , and S. Robertson. Inex 2007 evaluation measures. In *INEX 2007 Pre-Proceedings*, 2007.
12. S.E. Robertson and K. Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Information Sciences*, 27(3):129–146, 1976.
13. K. Sauvagnat and M. Boughanem. Propositions pour la pondération des termes et l'évaluation de la pertinence des éléments en recherche d'information structurée. In *CONFérence en Recherche d'Infomations et Applications (CORIA 2005)*, 2005.
14. J.A. Swets. Information retrieval systems. *Science*, 141:245–250, 1963.
15. A. Termier, Rousset M.-C., and Sebag M. Tree finder: a first step towards xml data mining. In *Proc. of Int. Conf. on Data Mining*, pages to–appear, 2002.
16. A.M. Vercoustre, M. Fegas, S. Gul, and Y. Lechevallier. A flexible structured-based representation for xml document mining. *ArXiv Computer Science e-prints*, 2006.
17. A.M. Vercoustre, M. Fegas, Y. Lechevallier, and T. Despeyroux. Classification de documents xml a partir d'une representation lineaire des arbres de ces documents. In *In Actes des 6eme journees Extraction et Gestion des Connaissances (EGC 2006), Revue des Nouvelles Technologies de l'Information (RNTI-E-6)*, pages 433–444, 2006.
18. R. Wilkinson. Effective retrieval of structured documents. In *17th ACM Conference on Research and Development in Information Retrieval (SIGIR'94)*, July 2007.
19. J. Yi and N. Sundaresan. A classifier for semi-structured documents. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 340–344, 2000.

Semi-supervised learning of ranking functions for Structured Information Retrieval

David Buffoni, Jean-Noël Vittaut, and Patrick Gallinari

Laboratoire d'Informatique de Paris 6
104, avenue du Président-Kennedy, F-75016 Paris, France
{buffoni, vittaut, gallinari}@poleia.lip6.fr

Abstract. We present a Retrieval Information system for XML documents using a Machine Learning Ranking approach. This system learns a ranking function using a training of queries and relevance judgments on a subset of the document elements. Classical ranking techniques learn from labeled data only. Besides an adaptation of ranking methods to structured IR, we also introduce a semi-supervised ranking scheme which learns both from labeled and unlabeled data. Our model improves the performance of a baseline Information Retrieval system by optimizing a ranking loss criterion and combining scores computed from doxels and from their local structural context. We analyze the performance of these models on the CO-Focused task.

1 Introduction

Ranking algorithms have been developed in the Machine Learning field for some times. In the field of IR, they have first been used for combining features or preferences relations in tasks such as meta search [1], [2]. Learning ranking functions has also lead to improved performance in a series of tasks such as passage classification, automatic summarization [3]. More recently, they have been used for learning the ranking function of search engines [4], [5], [6], [7].

Ranking algorithms work by combining features which characterize the data elements to be ranked. In our case, these features will depend on the document element (doxel) itself and on its structural context. Ranking algorithms will learn to combine these different features in an optimal way, according to a specific loss function, using a set of examples.

Ranking algorithms are trained in a supervised way, using a set of labeled data. This approach is probably not adapted to Information Retrieval tasks. Data labeling in this context is time consuming. Also, due to the large variability of potential queries and the open nature of the task itself, it is unrealistic to envision the labeling of a representative subset of data for most IR tasks. This is even more sensitive for structured IR where the number of elements to be retrieved is potentially much larger than in traditional IR. A potential solution to this problem is to develop semi-supervised ranking methods which learn from a small set of labeled data and attempt to exploit jointly the information provided by unlabeled data. Semi-supervised techniques have been developed for

classification tasks but not for ranking ones. We propose here a semi-supervised approach to the ranking problem and analyse its performance wrt a baseline model and a supervised ranking model introduced last year [8].

The paper is organized as follows, in section 2 we present the ranking model. We then show how this model can be extended to support semi-supervised learning. In section 3 we comment the results obtained by our semi supervised model and compare them to a supervised model.

2 Ranking model

In this section we briefly describe a probabilistic model of ranking which can be adapted to Information Retrieval or Structured Information Retrieval. A more detailed description of the model can be found in [8].

The main idea behind the Machine Learning Ranking is to learn a total strict order on \mathcal{X} , a set of elements. This allows it to compare any pair of elements in this set.

Given this total order, we are able to order any subset of \mathcal{X} in a ranking list. For Information Retrieval on XML documents, \mathcal{X} will be the set of couples (doxel, query) for all doxels and queries in the document collection and the total order is the natural order on the doxel's scores. In addition, we need a training set of ordered pairs of examples to learn how to rank. This training set will provide us with a partial order on the elements of \mathcal{X} . Our algorithm will use this information to learn a total order on \mathcal{X} and it will then be able to rank new elements.

2.1 Notations

As described above, we assume available a set \mathcal{X} of elements ordered by a partial order noted \prec . This relation will be used when it is possible to compare element pairs of \mathcal{X} . Let \mathcal{D} be the set of all doxels of all documents in the Wikipedia collection and \mathcal{Q} be the set of CO-queries. In the context of structured IR, we will make define $\mathcal{X} = \mathcal{Q} \times \mathcal{D}$. The partial order hypothesis on $\mathcal{X} = \mathcal{Q} \times \mathcal{D}$, means that for a subset of the queries in \mathcal{Q} we know preferences between some of the doxels in \mathcal{D} . For a given query, these preferences will define a partial order on the doxels in \mathcal{D} . The preferences among doxels are provided by manual assessments.

Ranking We represent each element $x \in \mathcal{X}$ by a vector (x_1, x_2, \dots, x_l) where x_i are features needed to rank elements of \mathcal{X} . We denote \mathcal{L} as the set of doxel types given by the DTD of wikipedia collection. For example : *article, section, paragraph,...* The following linear combination of features is used to define the ranking function f_ω , that we will use to learn a total order on \mathcal{X} :

$$f_\omega(x) = \omega_1^l \cdot \omega_2^l \cdot Okapi(x) + \omega_3^l \cdot Okapi(parent(x)) + \omega_4^l \cdot Okapi(document(x)) \quad (1)$$

where ω_i^l are the parameters of the combination to be learned, l is the type of doxel of the element x and Okapi is an Okapi [9] model adapted to Structured Information Retrieval. This combination takes into account both the information provided by the context of the doxel and the structural information given by the node type of the doxel.

More precisely, we have used the following vector representation:

$$x = ((\omega_1^{l_1}, \omega_2^{l_1}, \omega_3^{l_1}, x_4^{l_1}), (\omega_1^{l_2}, \omega_2^{l_2}, \omega_3^{l_2}, x_4^{l_2}), \dots, (\omega_1^{l_{|\mathcal{L}|}}, \omega_2^{l_{|\mathcal{L}|}}, \omega_3^{l_{|\mathcal{L}|}}, \omega_4^{l_{|\mathcal{L}|}})) \quad (2)$$

where $|\mathcal{L}|$ is the total number of doxel types in the collection. In the equation (2), each component of the vector $(\omega_1^{l_i}, \omega_2^{l_i}, \omega_3^{l_i}, \omega_4^{l_i})$ is $(0, 0, 0, 0)$ except for the component which corresponds to the doxel's type, say l_i , which is equal to :

$$(1, Okapi(x), Okapi(parent(x)), Okapi(document(x)))$$

Ranking loss f_ω is said to respect the order $x \prec x'$ if $f_\omega(x) < f_\omega(x')$ for $x, x' \in \mathcal{X}$. In this case, the pair (x, x') is well ordered by the function f_ω . Consequently, the ranking loss will evaluate the number of times f_ω does not respect this condition, in other terms, it will count the number of mis-ordered pairs in \mathcal{X}^2 . This criterion is commonly called Area Under the ROC Curve (AUC) and it can be written as follows :

$$A_{ROC} = \frac{1}{n \cdot p} \sum_{i \in \mathcal{X}^+} \sum_{j \in \mathcal{X}^-} [[f(x_i) - f(x_j) \leq 0]] \quad (3)$$

where \mathcal{X}^- is the set of non relevant element, \mathcal{X}^+ is the set of relevant elements and $n = |\mathcal{X}^-|$ and $p = |\mathcal{X}^+|$. The aim of a ranking algorithm is to learn ω (the combination parameters (1)) by minimising (3). However, equation (3) is not differentiable. Therefore instead of the ROC criterion (3), we will use an upper-bound of (3) which has the form of an exponential loss (4). it is differentiable and convex, and can be minimized by standard optimization techniques like gradient descent:

$$R_e(\mathcal{X}, \omega) = \sum_{\substack{(x, x') \in \mathcal{X}^2 \\ x \prec x'}} e^{f_\omega(x) - f_\omega(x')} \quad (4)$$

In addition, we can use some particularities of INEX, to decrease the complexity of (4). First of all, since comparing doxels from different queries has no sense, we define a partition $\mathcal{X} = \bigcup_{q \in \mathcal{Q}} \mathcal{X}_q$ where

$$\mathcal{X}_q = \{x = (d, q') \in \mathcal{X} / q' = q\}$$

Second, the assessments being described by discrete dimensions on exhaustivity and specificity, there will be no preference (this is denoted (\perp)) among

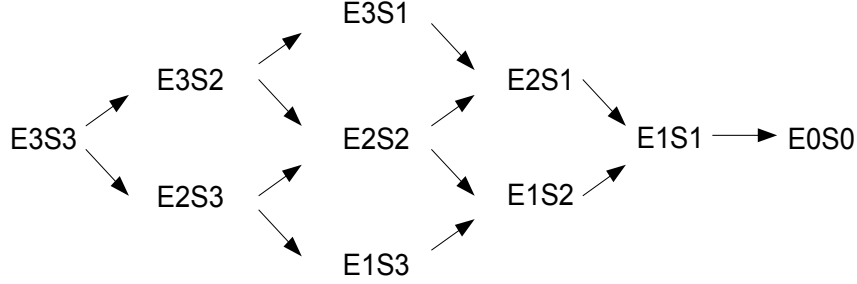


Fig. 1. Graph representing the order between doxels for a given query, according to the two dimensional discrete scale of INEX. Doxels labeled E_3S_3 must be the highest ranked, and doxels labeled E_0S_0 the lowest ranked.

doxels with the same value of exhaustivity and specificity. Since INEX'06, specificity was replaced by two parameters : rsize (amount of relevant highlighted text in the doxel) and size (total number of characters contained by the doxel). Thus, we convert the ratio $\frac{rsize(d)}{size(d)}$ of a doxel d to an integer value between 0 and 3.

Therefore, with \mathcal{A} the set of assessments and $A(x)$ the assessment for an element x , we can write the partition $\mathcal{X}_q = \bigcup_{a \in \mathcal{A}} \mathcal{X}_q^a$ where

$$\mathcal{X}_q^a = \{x \in \mathcal{X}_q / A(x) = a\}$$

According to the two properties above, we obtain a new exponential loss :

$$R_e(\mathcal{X}, \omega) = \sum_{q \in \mathcal{Q}} \sum_{a \in \mathcal{A}} \left\{ \left(\sum_{x \in \mathcal{X}_q^a} e^{f_\omega(x)} \right) \left(\sum_{\substack{b \in \mathcal{A} \\ \mathcal{X}_q^b \prec \mathcal{X}_q^a}} \sum_{x \in \mathcal{X}_q^b} e^{-f_\omega(x)} \right) \right\} \quad (5)$$

where $\mathcal{X}_q^b \prec \mathcal{X}_q^a$ means that \mathcal{X}_q^a is better than \mathcal{X}_q^b . A possible order between assessments is represented in figure 1 according to the couple (exhaustivity, specificity).

The complexity of the algorithm, $O(|\mathcal{X}^2|)$, is reduced to a complexity in $O(K \cdot |\mathcal{Q}| \cdot |\mathcal{X}|)$ where $|\mathcal{K}|$ is the number of sets in the partition of \mathcal{X} .

Gradient descent To minimize the exponential loss (5), we can apply a gradient descent technique. The gradient component is:

$$\begin{aligned} \frac{\partial R_e}{\partial \omega_k}(\mathcal{X}, \omega) = & \sum_{q \in \mathcal{Q}} \sum_{a \in \mathcal{A}} \left\{ \left(\sum_{x \in \mathcal{X}_q^a} x_k e^{f_\omega(x)} \right) \left(\sum_{\substack{b \in \mathcal{A} \\ \mathcal{X}_q^b \prec \mathcal{X}_q^a}} \sum_{x \in \mathcal{X}_q^b} e^{-f_\omega(x)} \right) \right. \\ & \left. + \left(\sum_{x \in \mathcal{X}_q^a} e^{f_\omega(x)} \right) \left(\sum_{\substack{b \in \mathcal{A} \\ \mathcal{X}_q^b \prec \mathcal{X}_q^a}} \sum_{x \in \mathcal{X}_q^b} -x_k e^{-f_\omega(x)} \right) \right\} \end{aligned} \quad (6)$$

Incorporation of unlabeled data With the semi-supervised model, we have to label correctly all unlabeled elements y . Thus, we attribute each new element y to the partition \mathcal{X}_q^a (and not to all the partitions) according to a probability of belonging. In other terms, an element belongs to a group with which it has the maximum indifference probability.

$$P(y \in \mathcal{X}_q^a) = P(\{y\} \perp \mathcal{X}_q^a) = \prod_{x \in \mathcal{X}_q^a} P(y \perp x) = \prod_{x \in \mathcal{X}_q^a} P(y \prec x) P(x \prec y)$$

Next, we choose for y the group \mathcal{X}_q^a which minimizes the ranking loss as follows :

$$e^{f_\omega(y)} \sum_{x \in \mathcal{X}_q^a} e^{-f_\omega(x)} + e^{f_\omega(-y)} \sum_{x \in \mathcal{X}_q^a} e^{f_\omega(x)}$$

The semi supervised model can be summed up by the following algorithm:

Algorithm 1

1. *Minimize the ranking loss on labeled examples.*
2. *Repeat until convergence:*
 3. *Affect each unlabeled example to a group \mathcal{X}_q^a according to the minimum ranking loss:*

$$e^{f_\omega(y)} \sum_{x \in \mathcal{X}_q^a} e^{-f_\omega(x)} + e^{f_\omega(-y)} \sum_{x \in \mathcal{X}_q^a} e^{f_\omega(x)}.$$
 4. *Minimize the ranking loss on labeled and unlabeled examples.*

3 Experiments

3.1 Learning base

The Wikipedia collection [10] has been used with different sets of queries for training and testing. INEX 2006 queries and assessments were used for training and the 2007 collection was used for testing. In order to analyze the behavior of

the ranking and semi-supervised ranking methods, experiments were performed with different labeled training test sizes.

See below an enumeration of the training sets :

1. One of 3 queries
2. One of 10 queries
3. One of 50 queries
4. One of 100 queries

The model learns by taking into account different percentages of labeled data. We focus our experiments on small percentages ($< 10\%$) to be close of the semi-supervised paradigm.

3.2 Results

The runs submitted to the official evaluation were bugged. New results will be presented at the workshop.

References

1. Cohen, W.W., Schapire, R.E., Singer, Y.: Learning to order things. In Jordan, M.I., Kearns, M.J., Solla, S.A., eds.: *Advances in Neural Information Processing Systems*. Volume 10., The MIT Press (1998)
2. Freund, Y., Iyer, R., Schapire, R.E., Singer, Y.: An efficient boosting algorithm for combining preferences. In: *Proceedings of ICML-98, 15th International Conference on Machine Learning*. (1998)
3. Amini, M.R., Usunier, N., Gallinari, P.: Automatic text summarization based on word-clusters and ranking algorithms. In: *ECIR'05: European Conference on Information Retrieval*. (2005) 142–156
4. Craswell, N., Robertson, S., Zaragoza, H., Taylor, M.: Relevance weighting for query independent evidence. In: *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference*. (2005)
5. Vittaut, J-N., Gallinari, P.: Machine Learning Ranking for Structured Information Retrieval. In: *ECIR'06: European Conference on Information Retrieval*. (2006) 338–349
6. Xu, J., Li, H.: AdaRank: A Boosting Algorithm for Information Retrieval. In: *SIGIR '07: Proceedings of the 28th annual international ACM SIGIR conference*. (2007)
7. Tsai, M-F., Liu, T-Y., Qin, T., Chen, H-H., Ma, W-Y. : FRank: A Ranking Method with Fidelity Loss. In: *SIGIR '07: Proceedings of the 28th annual international ACM SIGIR conference*. (2007)
8. Vittaut J.N., Gallinari P. : Supervised and Semi-Supervised Machine Learning Ranking. *INEX 2006 preproceedings*, (2006)
9. Robertson, S.E., Walker, S., Hancock-Beaulieu, M., Gull, A., Lau, M.: Okapi at TREC. In: *Text REtrieval Conference*. (1992) 21–30
10. Denoyer L., Gallinari P.: The Wikipedia XML Corpus *SIGIR Forum* (2006)

Ranking and Presenting Search Results in an RDB-based XML Search Engine

Kenji Hatano¹, Toshiyuki Shimizu², Jun Miyazaki³, Yu Suzuki⁴,
Hiroko Kinutani⁵, and Masatoshi Yoshikawa²

¹ Faculty of Culture and Information Science, Doshisha University
khatano@mail.doshisha.ac.jp

² Graduate School of Informatics, Kyoto University
shimizu@soc.i.kyoto-u.ac.jp, yoshikawa@i.kyoto-u.ac.jp

³ Graduate School of Information Science,
Nara Institute of Science and Technology
miyazaki@is.naist.jp

⁴ College of Information Science and Technology, Ritsumeikan University
suzuki@ics.ritsumeai.ac.jp

⁵ Institute of Industrial Science, The University of Tokyo
kinutani@tkl.iis.u-tokyo.ac.jp

Abstract. Conventional ranking methods for document search have considered content of documents to rank a search result. They have attained some positive results in the research area of document search; however, it has been said that content of not only documents but also queries should be utilized if users want to get a search result accurately. This fact applies to XML search engines. In this paper, therefore, we propose a ranking method for XML search considering content-and-structure conditions of both XML documents and queries. We also propose a method for presenting a search result for XML search, because it is very important for users to grasp and understand the entire search result, too. We implemented our ranking method on top of XRel, a relational database system for XML documents, and found that our proposal allows users to search XML fragments more accurately than previously proposed approaches for XML search.

1 Introduction

Extensible Markup Language (XML) [1] is becoming widely used as a standard document format in many application domains. In the near future, we believe that a greater number of documents will be produced in XML. Therefore, in a similar way to the development of Web search engines, XML search engines will become very important tools for users wishing to explore XML documents.

In the meantime, a search result of current Web search engines is usually a list of Web documents. That is, Web documents are sorted in descending order of their scores. The scores are calculated by content of Web documents, which are quantified based on the number of occurrences of terms extracted from Web

documents like the *tf-idf* scoring [2]. In the case of XML search engines, it is said that XML queries combine conditions on both content and logical structure such as Narrowed-Extended XPath I (NEXI) [3] and XQuery Full-Text queries [4]. When such queries are issued to an XML search engine, the search result is usually a list of XML fragments⁶ as opposed to that of entire documents in current Web search engines. As a result, several approaches have been proposed to extend the well-established content-based scoring in some retrieval with the ability to rank XML fragments.

Conventional approaches for XML search take into consideration both content and logical structure of XML documents in order to rank XML fragments which satisfy query conditions [5]. For example, in the context of *tf-idf* scoring, element scoring precomputes *tf* and *idf* factors for each distinct tag in input XML documents [6, 7], while path scoring precomputes them for distinct paths [8, 9]. These refined scoring approaches led to improvements in the retrieval accuracy of search results consist of scored XML fragments. However, these approaches tended to attach great importance to small XML fragments, so that they caused a problem returning small XML fragments partially satisfied with users' information need in some cases [10–12]. On the other hand, it is also said that it is important for XML search engines to handle overlapping parts of XML fragments. It means that when a user grasps and understand the content of an XML fragment, the user browses ancestor of the XML fragment unconsciously. In short, users can grasp and understand the content of search results if XML search engines can indicate a list of large size of XML fragments containing small ones. Considering this fact, Clarke proposed to control overlapping by re-ranking the descendant and ancestor of search results [13]. In Clarke's approach, however, users have to browse the overlapping parts of XML fragments more than once, so that it increases the burden on users.

In order to overcome two problems described above, we propose ranking and presenting methods of XML fragments as search results for XML search engines. In our ranking method, we advocate the use of two scoring algorithms for content-only (CO) and content-and-structure (CAS) queries. The former is based on the content condition of XML documents like conventional element or path scoring methods, and is utilize statistics extracted from XML documents effectively, though the latter is based on the content-and-structure conditions of both XML documents and queries. This is because the basic idea of our ranking method has been shown to improve retrieval accuracies of search results in the research area of traditional document search. At the same time, we also insist that we devise ways of effectively presenting search results to handle overlapping parts of answer XML fragments, because XML search engines just have to decide and present one XML fragment in one way or another if there is an ancestor-descendant relationship among XML fragments in a search result. In order to verify the effectiveness of our proposals, we implemented two scoring methods on a relational database system for XML documents based on XRel [14]. Our

⁶ XML fragments are easily extracted from XML documents based on their markup. That is, they are subtrees in the XML trees.

experiments on the INEX test collection show that using content-and-structure conditions of both documents and queries improves the retrieval accuracies of XML search engines.

The remainder of this paper is organized as follows. In Section 2, we describe how to calculate the scores of XML fragments based on content-and-structure conditions of both documents and queries and statistics of XML documents. In Section 3, we also describe how to present XML fragments with ancestor-descendant relationships. We report our experimental results to verify our proposal in Section 4 and related work in Section 5. Finally, we conclude this paper in the last section.

2 Our Ranking Method based on Content-and-Structure Conditions

In Section 2.1 and 2.2, we describe our two types of scoring algorithms in detail. One is for CO queries and takes the content-and-structure condition of XML documents into consideration. The other is for CAS queries and considers that of both XML documents and queries. We also explain a method for integrating two scoring algorithms in Section 2.3.

2.1 Content-and-Structure Conditions of XML Documents

As we described in Section 1, conventional methods for calculating scores of XML fragments have already studied in recent years. One of the most famous methods for calculating scores of XML fragments is element-based or path-based scoring in the vector space model [15]; however we simply explain the path-based scoring here because it has proved to perform better than element scoring [7].

The path-based scorings like the *tf-ipf* scoring [9] are expanded the versatility of the *tf-idf* scoring [2], which has been proposed to quantify the importance of terms in documents. The concept of the *tf-idf* scoring is that a *tf-idf* score of a certain term in the document becomes large if the term appears in it many times and does not appear in others at the same time. The *tf-ipf* scoring behaves the same as the *tf-idf* scoring and has been used for XML search. XML fragments extracted from an original XML document are identified their XPath expressions [16], so that they are classified according to the abbreviated syntax of their XPath expressions. Assuming that the XML fragments with the same abbreviated XPath expression have the same properties, we can quantify the importance of terms in XML fragments with same properties as *tf-ipf* scores. That is to say, a *tf-ipf* score of a certain term in an XML fragment becomes large if the term appears in it many times and does not appear in others with the same abbreviated XPath expression at the same time.

In our scoring algorithm, we define a *tf-ipf* score calculated from content-and-structure conditions of XML documents. The score S_d is composed of two factors, “Term Frequency of XML fragment (tf_d)” and “Inverse Path Frequency of XML fragment (ipf_d)” as same as the *tf-idf* scoring. These factors are inspired

from one of the path-based scorings proposed in [9]. In short, if T is the set of query terms and s is an answer XML fragment in a search result, $tf_d(s, t)$ is the number of occurrences of term $t \in T$ in s and $ipf_d(s, t)$ is the natural logarithm of quotient of the number of XML fragments which have the same structure as s and the number of such answer XML fragments containing term t . We assume the independence between paths in original XML documents and combine $ipf_d(s, t)$ of individual paths. For example, given the query `//article//sec[about(., t1 t2)]`, $tf_d(s, t_i)$ and $ipf_d(s, t_i)$ ($i = 1, 2$) are defined as follows:

$$tf_d(s, t_i) = \frac{n(s, t_i)}{l(s)} \quad ipf_d(s, t_i) = 1 + \log \frac{M(s)}{m(s, t_i)} \quad (1)$$

where $n(s, t_i)$ is the number of occurrences of t_i in s , $l(s)$ is the length of s (total number of terms in s), $M(s)$ is the number of XML fragments in the original XML documents which satisfy s 's structure, and $m(s, t_i)$ is the number of such fragments containing t_i . Therefore, a score considering content-and-structure condition of XML documents S_d is defined as the following equation:

$$S_d(s) = \sum_{t \in T} tf_d(s, t) \cdot ipf_d(s, t) \quad (2)$$

In addition, we have already found two heuristics for calculating $S_d(s)$ exactly. The first heuristic is that small XML fragments are not suitable for search results in XML search engines, especially keyword search. This is because the XML fragments in a search result are supposed to be semantically consolidated granules of original XML documents. In other words, such small XML fragments are not semantically consolidated granules, so that they should not be included in search results. We have already pointed this problem in [12], and proposed a method for deleting small XML fragments from search results using quantitative linguistics [11]. Applying this approach proposed in [11] to our XML search engine easily, we defined a threshold called the ratio of period to delete such small XML fragments from search results in [10]. The ratio of period is defined as follows:

$$r(s_e) = \frac{n_p(s_e)}{N_p(s_e)} \quad (3)$$

where $N_p(s_e)$ denotes the number of XML fragments whose tag names is s_e ⁷, and $n_p(s_e)$ is the number of XML fragments that end with the symbols like `.`, `?`, or `!` if the node with tag name s_e is a leaf node, or the number of XML fragments that have more than one document-centric leaf node if the node with tag name s_e is an internal node.

In contrast, the second heuristic is that $tf_d(s, t_i)$ has a negative effect for calculating $S_d(s)$. That is to say, the *tf* and *idf* factors in the *tf-idf* scoring are well-balanced; however, the *tf* and *ipf* factors in the *tf-ipf* scoring are not well-balanced. For example, $tf_d(s, t_i)$ is more influence over $S_d(s)$ than $ipf_d(s, t_i)$ in experiments of the INEX test collections (from 2002 to 2005), so that we believe

⁷ s_e is the tag name of an XML fragment s .

that $ipf_d(s, t_i)$ has an insignificant effect on $S_d(s)$. Liu et al. also found the same fact and proposed an well-balanced tf factors suitable for $ipf_d(s, t_i)$ based on statistics extracted from original XML documents, which were calculated by using the average number of terms of the XML fragments with the same abbreviated XPath expression $l_{ave}(s)$ and a constant parameter c as follows:

$$tf_d(s, t) = \frac{o.tf(s, t)}{n.tf(s)} \quad (4)$$

$$o.tf(s, t) = 1 + \log(1 + \log(n(s, t))) \quad (5)$$

$$n.tf(s) = 1 + \frac{l_{ave}(s) \cdot l(s)}{l_{ave}(s)} \cdot c \cdot (1 + \log(l_{ave}(s))) \quad (6)$$

In this paper, we adapted their methods to original tf - ipf scoring and calculated $tf_d(s, t)$ defined in equation (4). We call this scoring method “ ntf - ipf scoring”, which is extended by using statistics of original XML documents. The constant parameter c in equation (6) was usually set to 0.2. Owing to limited space, we do not describe the details of their method (see [17]).

2.2 Content-and-Structure Conditions of Queries

Using the path-based scorings, we can calculate scores of XML fragments related to queries before users issues them to XML search engines. Such precomputing scores solely rely on original XML documents and do not consider query conditions on both content and structure. As a result, only using the path-based scorings is unable to function to calculate scores of answer XML fragments exactly.

More concretely, let us consider the XML document given in Fig. 1. This example is extracted from the INEX 2007 document collection. If a NEXI query like `//article//p[about(., "Gates")]` is issued to this example, XML fragment s_1 : `/article[1]/body[1]/p[1]`, s_2 : `/article[1]/body[1]/section[1]/p[1]`, and s_3 : `/article[1]/body[1]/section[1]/p[2]` would return as a search result. In existing approaches, the scores of XML fragments s_1 and $s_2, 3$ are different each other because their abbreviated XPath expressions are different from the standpoint of both content and logical structure of original XML documents. From the standpoint of query condition, however, they should be identified because these XML fragments are satisfied with both content and logical structure of the query. In short, we would like to give the same scores to the XML fragments satisfied with all condition of the query. Therefore, we can account for this by considering condition on content and structure in the input queries and defining scores as a function of those conditions as well as precomputed document-based scores described in Section 2.1. This idea is basically the same in traditional document search [2], and we believe that it helps to improve the retrieval accuracies of search results.

Now we define a query-based score S_q . Similarly to the document-based score $S_d(s)$, S_q is composed of two factors, “Term Frequency of Query (tf_q)” and “Inverse Answer Document Frequency of Query (iaf_q)”, so that we call this scoring

```

<?xml version="1.0" encoding="UTF-8"?>
<article>
  <name id="3747">Bill Gates</name>
  <body>
    <p>
      <emph3>William Henry Gates III</emph3> (born October 28, 1955),
      commonly known as <emph3>Bill Gates</emph3>, is the co-founder,
      chairman and chief software architect of Microsoft Corporation,
      the largest software company in the world. According to ...
    </p>
    ...
    <section>
      <title>Early life</title>
      <p>
        Gates was born in Seattle, Washington, to William H. Gates,
        Sr., a prominent lawyer, and Mary Maxwell Gates. Gates was born
        with a million dollar trust fund set up by his grandfather, ...
      </p>
      <p>
        Gates, with an estimated I.Q. of 160, excelled in elementary
        school, particularly in mathematics and the sciences ...
      </p>
      ...
    </section>
    ...
  </body>
</article>

```

Fig. 1. A Sample XML Document in the INEX 2007 Document Collection

the *tf-iaf* scoring. iaf_q is important for calculating the query-based score S_q and has only been explored once in isolation [7]. However, the cost of calculating iaf_q can be quite expensive. Therefore, we only focus on the effectiveness of XML search engines in this paper. In the same manner as a document-based score $S_d(s)$, given the query `//article` `//sec[about(., t1 t2)]`, $tf_q(t_i)$ and $iaf_q(t_i)$ are defined as follows:

$$tf_q(t_i) = w(t_i) \quad iaf_q(t_i) = 1 + \log \frac{V(p)}{v(p \ t_i)} \quad (7)$$

where $w(t_i)$ is the number of occurrences of t_i in the query, $V(p)$ is the number of XML fragments satisfying the query path p (in this case, `//article//sec`), and $v(p \ t_i)$ is the number of XML fragments satisfying the query path p containing term t_i . In order to calculate $iaf_q(t_i)$, we also assume independence between paths in the query and combine $iaf_q(t_i)$ of individual paths. Therefore, a query-

based score S_q is defined as the following equation:

$$S_q = \sum_{t \in T} tf_q(t) \quad ia_f_q(t) \quad (8)$$

2.3 Our Ranking Method

We finally define the combination of a document-based score $S_d(s)$ and a query-based score S_q . This idea is inspired from the SMART retrieval system⁸, which has been considered the term weights of both documents and queries. In order to combine them, the SMART retrieval system calculates their product in the same spirit as document scores described in [2].

In our method, we apply the same idea to our XML search engine. Scores of an XML fragment s related to a query is thus defined as follows:

$$S(s) = \sum_{t \in T} S_d(s) \quad S_q = \sum_{t \in T} tf_d(s \ t) \quad ip_f_d(s \ t) \quad tf_q(t) \quad ia_f_q(t) \quad (9)$$

3 Search Result Presentation

As we described in Section 1, it is also important for improving the retrieval accuracy of XML search engines to propose a method for presenting search results. This is because XML search should consider the overlapping parts of answer XML fragments unlike document search. Considering this fact, Clarke has proposed to control overlapping by re-ranking the descendant and ancestor of search results [13]. Compared with his approach, we propose a concept of search result presentation which is a unit of answer XML fragments and use it in our XML search engine. We believe that our search result presentation helps for users to grasp and understand the entire search results effectively compared with conventional approaches.

3.1 Search Result Presentation for XML Search

XML search engines extract XML fragments satisfied with a query from original XML documents. In other words, it remains possible that a large number of answer XML fragments are returned from XML search engines. Such answer XML fragments may be extracted from one XML fragment. For example, XML documents in the 2005 INEX document collection are scholarly articles, so that sections, subsections, paragraphs and so on are retrieved by XML search engines. Such retrieved XML fragments may overlap due to nesting structure of XML documents. This fact causes the problem to be difficult to grasp and understand the entire search results effectively.

Because of the above situation, the INEX project has demanded some kinds of search result presentations such as not *Thorough* strategy but *Focused*, *Relevant-In-Context* and *Best-In-Context* ones. While the XML search engines with the

⁸ <ftp://ftp.cs.cornell.edu/pub/smart/>.

Thorough strategy can retrieve overlapping XML fragments, ones with other strategies can retrieve non-overlapping document parts containing answer XML fragments or a single document part per an XML document. That is, they firstly extract XML fragments related with queries, and then decide answer parts from extracted ones. We think that, however, these strategies also contain the problem because the XML search engines with these strategies find non-overlapping document parts using scored XML fragments in an XML document regardless of their scores. The best way to attain the most effective XML search is to extract some XML fragments related with the queries from one XML document, to generate document parts based on a unit appropriate for users, and to rank them for presenting search results. Considering these demands, we believe that an XML search engine would be more useful if it has a user interface which can handle a basic unit for XML search and can provide answers constructed from the unit. This is because it is natural for users to show answers mapped on original XML documents, and the users avoid the need to see the document parts not related with queries. In short, our XML search engine provides thumbnail of original XML documents and indicates the answer parts of XML documents directly in its user interface; in consequence, users can intuitively grasp and understand the search results⁹.

In order to implement such user interface of our XML search engine, we propose a new concept called “Aggregation Granularity (AG)”, which is a unit of search results determined from original XML documents. In next section, we describe our new concept in detail.

3.2 Aggregation Granularity

In conventional XML search engines, answer XML fragments are showed in their user interfaces individually on the *Thorough* strategy, so that users tend to get messed up the relationship among the answer XML fragments. In the case of our XML search engine realizing the new concept, answer XML fragments are allocated on original XML documents in its user interface. Therefore, the problem described in previous section is not caused in our XML search engine.

In some case, however, we would be better off aggregating several answer XML fragments with large scores into one document part to show the search results to users, because it is easy for users to understand the content of a search result from the viewpoint for grasping the outline of original XML document even if the score of the document part, which is also XML fragment containing the answer XML fragments, is not large. For example, a query is issued by a user, conventional XML search engines return answer XML fragments whose root nodes are gray-color elements in Fig. 2. As a result, a large number of answer XML fragments are returned, so that users cannot grasp and understand the search result. In our XML search engine, however, answer XML fragments with a certain degree of scores whose root nodes are gray-color elements in

⁹ The difference between the *Focused* strategy and our proposal is to be able to highlight answer XML fragments with large scores.

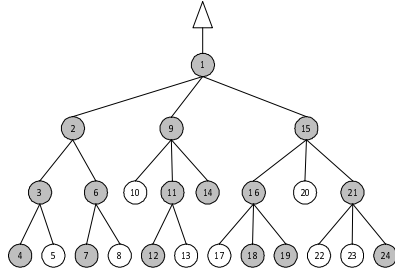


Fig. 2. Answer XML Fragments in Existing XML-IR System

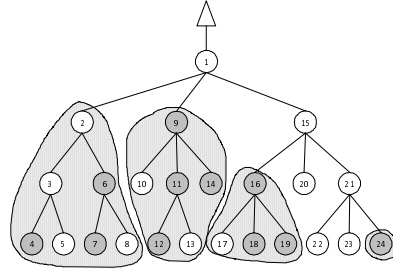


Fig. 3. Answer XML Fragments in Our XML-IR System

Fig. 3 are extracted from the search result, and then, some document parts enclosed by trajectory in Fig. 3 are constructed from them as basic units for XML search, AGs. As a result, users can grasp and understand the entire search results effectively compared with conventional XML search engines.

In this approach, the following two things become big problems. One is how to decide AGs, and the other is how to calculate the score of the document parts based on AG. To cope with the first problem, the AG can be defined if a certain standard like the threshold size, the location in original XML documents of answer XML fragments, and so on. In [18], for example, XML documents can be divided into multiple parts like physical pages, so that the AG is defined as individual pages of XML documents. Generally, XML fragments suitable for an AG tend to be located at the higher level of original XML documents, and their sizes tend to be relatively large. In short, it seems more likely that element 2, 9, and 15 in Fig. 3 would be first candidate of AG, and element 3, 6, 11, 16, and 21 would be second candidate. Alternatively, calculating scores of aggregated XML fragments varies in methodology. The easiest way to calculate their scores is the sum of the scores of answer XML fragments which constitute the document parts defined from AG. However, two problems above have a lot of things to be considered, so that now we are formulating the definition and score-calculation of AG. We would like to try every way possible to formulate and implement them in our XML search engine in the near future.

4 Experimental Evaluations

In this section, we conduct some experiments for the sake of the effectiveness of our proposals in our XML search engine. At the present time, an evaluation tool is not available, so that we show the experimental results using the 2005 INEX test collection¹⁰. This collection is composed of a document set marked up in XML, its relevance assessment, and evaluation measures. The document set contains 16,819 articles of the IEEE Computer Society's magazines and transactions published from 1995 to 2004. The size of the document set is 735MB,

¹⁰ We could not take part in INEX 2006.

an article contains 1,532 XML nodes on average, and the average depth of a node is 6.9. The relevance assessment has two graded dimensions to express relevance of XML fragments to XML queries, “exhaustivity” and “specificity”. The concept of specificity is peculiar to XML search, because it provides a measure of the size of an XML fragment as it measures the ratio of relevant to non-relevant content within the XML fragment. In order to identify relevant XML fragments to XML queries, INEX project provides two evaluation measures, recall-precision and eXtended Cumulated Gain (XCG) [19]. The recall-precision is used for evaluating the effectiveness of conventional information retrieval systems. The recall-precision in the INEX project maps the values of exhaustivity and specificity to a single scale using quantization functions [20]. On the other hand, the XCG was additionally proposed for evaluating effectiveness of XML search engines [21] because the recall-precision evaluation measure could not consider overlapping XML fragments. This problem is amply explained in [22] and can be summarized as the issue of avoiding to return both elements and their sub-elements as query results and recalculating scores on the fly when that happens. In that sense, the XCG measure accounts for both retrieval accuracy and users experience.

4.1 Evaluation of Scoring based on Document Conditions

In this evaluation, we used the recall-precision and the XCG measures to evaluate our scoring algorithm for CO queries.

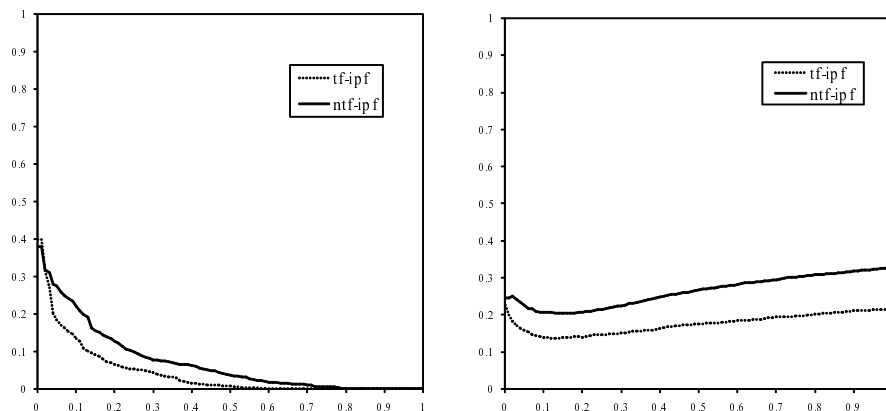


Fig. 4. Retrieval Accuracies based on Recall-Precision/nXCG

Fig. 4 shows the retrieval accuracies based on the recall-precision and the nXCG in the INEX evaluation measures. “tf-ipf” in Fig. 4 is the original *tf-ipf* scoring, “ntf-ipf” is the scoring method defined in equation (2) where $tf_d(s, t)$ is

redefined in equation (4)¹¹. Fig. 4 speaks that the *ntf-ipf* scoring could retrieve more relevant XML fragments than the *tf-ipf* one in the *Thorough* strategy. In short, we can verify the effectiveness of the *ntf-ipf* scoring for the CO queries. We also noticed that we have to formulate not only the *ipf* factor but also the *tf* factor for effective XML search, because the original *tf-ipf* scoring has never configured the *tf* factor in the *tf-idf* scoring for document search. As a result, it is important for effective XML search to use the statistics extracted from original XML documents and to formulate the scoring algorithm.

4.2 Evaluation of Scoring based on Query Conditions

In this evaluation, we also used two evaluation measures to evaluate our scoring algorithm for CAS queries.

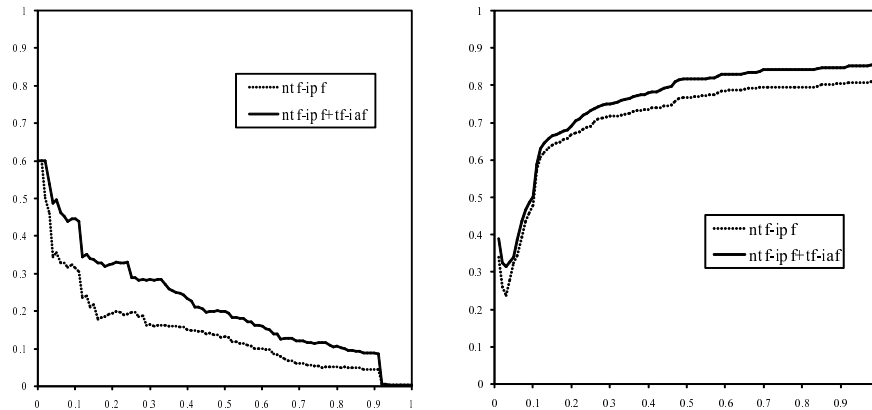


Fig. 5. Retrieval Accuracies based on Recall-Precision/nXCG

Fig. 5 shows the retrieval accuracies based on the recall-precision and the nXCG in the INEX evaluation measures. We found that our scoring algorithm (hereinafter called “*tf-ipf+tf-iaf*”) could retrieve more relevant XML fragments than one where only document-base score is considered (hereinafter called “*tf-ipf*”) in the *Thorough* strategy. In fact, we noticed that the relevant XML fragments tended to be higher on the list of each search result using *tf-ipf+tf-iaf*. This is because the XML fragments whose exhaustivity is large may be ranked lower than ones whose exhaustivity is small using only *tf-ipf* scoring. Using both *tf-ipf* and *tf-iaf* scorings, on the other hand, XML fragments whose exhaustivity and specificity are large make a point of being ranked higher in the search results. In short, introducing *tf-iaf* scoring reflects exhaustivity in the scores of

¹¹ Finally, $S_d(s)$ is weighted by the length of s . In short, the smaller the length of s is, the smaller $S_d(s)$ is.

answer XML fragments, and helps to improve the retrieval accuracies of XML search engines.

In summary, we have verified the effectiveness of the *tf-iaf* scoring for CAS queries, because it can retrieve more relevant XML fragments compared with the *tf-ipf* scoring.

5 Related Work

The application of information retrieval techniques in searching XML documents has become an area of research in recent years. Especially, the participants in the INEX project have proposed a lot of scoring proposals for XML search [5]. Over the years, it has become clear that refining the level of granularity at which document structure is taken into account in pre-computing individual term weights either in the vector space model or the probabilistic model, has increased retrieval accuracy. However, document statistics query conditions have not been explored to the extent at which we are proposing in this paper.

Fuhr et al. proposed a method for propagating scores of XML fragments leaf-to-root along the XML document tree [23]. However, although XIRQL, their proposed language, enables queries with a mix of conditions on both structure and keywords, only keywords are scored using conditions on document structure. Other scoring methods also use conditions on document structure to apply length normalization between query paths and data paths [8], to compute term weights based on element tags or paths [6, 9], or to account for overlapping elements [13]. It was reported that these methods were useful for searching XML fragments [19]; however, such methods did not use statistics of original XML documents and structural conditions of queries.

We believe that we have to utilize everything extracted from XML documents and queries for searching XML fragments accurately. In this paper, therefore, we showed that accounting for document statistics and query structure in addition to the existing methods, and combining them to improve retrieval accuracies of XML search engines. We can verify the effectiveness of our above proposals through the experimental evaluation in Section 4.

6 Conclusion

XML is emerging as the standard format for presenting data and documents on the Internet, and XML search engines are becoming necessary. Existing XML search engines can consider the content and the structure of XML documents to rank answer XML fragments to the XML queries. However, XML queries combine conditions on content and structure of both document and queries. That is, depending on the types of XML queries, we have to use the *tf-ipf* and the *tf-iaf* scorings. Based on this consideration, we proposed a method of content- and structure-based scorings in the vector space model considering both document and query conditions. Our method integrates document- and structure-based term-weighting strategies for XML search. Using our method, we found that we

could retrieve more relevant XML fragments with higher retrieval accuracy than using conventional scoring methods. We also proposed the displaying method to improve the retrieval accuracies of XML search engines. Unfortunately, we could not verify the effectiveness of this approach in this paper; however, we think that displaying search results is closely related to improving retrieval accuracies of XML search engines from the standpoint of users. This fact has already noticed in human interface research area, so that we have to implement our approach to our XML search engine as early as possible.

Acknowledgments

This work was partly supported by Grant-in-Aid for Scientific Research on Priority Areas #19024058 of the Ministry of Education, Culture, Sports, Science and Technology (MEXT), and Core Research for Evolutional Science and Technology (CREST) program “New High-performance Information Processing Technology Supporting Information-oriented Society” of the Japan Science and Technology Agency (JST).

References

1. Bray, T., Paoli, J., Sperberg-McQueen, M., Maler, E., Yergeau, F.: Extensible Markup Language (XML) 1.0 (Fourth Edition). <http://www.w3.org/TR/xml> (Sep. 2006) W3C Recommendation 16 August 2006, edited in place 29 September 2006.
2. Salton, G., Buckley, C.: Term-Weighting Approaches in Automatic Text Retrieval. *Information Processing and Management* **24**(5) (1988) 513–523
3. Trotman, A., Sigurbjörnsson, B.: Narrowed Extended XPath I (NEXI). In: *Advances in XML Information Retrieval*. Volume 3493 of *Lecture Notes in Computer Science*., Springer-Verlag (May 2005) 16–40
4. Amer-Yahia, S., Botev, C., Dorre, J., Shanmugasundaram, J.: XQuery Full-Text extensions explained. *IBM Systems Journal* **45**(2) (Dec. 2006) 335–352
5. Amer-Yahia, S., Lalmas, M.: XML Search: Languages, INEX and Scoring. *SIGMOD Record* **35**(4) (Dec. 2006) 16–23
6. Cohen, S., Mamou, J., Kanza, Y., Sagiv, Y.: XSearch: A Semantic Search Engine for XML. In: *Proceedings of 29th International Conference on Very Large Data Bases*. (Sep. 2003) 45–56
7. Amer-Yahia, S., Koudas, N., Marian, A., Srivastava, D., Toman, D.: Structure and Content Scoring for XML. In: *Proceedings of the 31st International Conference on Very Large Data Bases*. (Aug./Sep. 2005) 361–372
8. Carmel, D., Maarek, Y.S., Mandelbrod, M., Mass, Y., Soffer, A.: Searching XML Documents via XML Fragments. In: *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. (Jul./Aug. 2003) 151–158
9. Grabs, T., Schek, H.J.: PowerDB-XML: A Platform for Data-Centric and Document-Centric XML Processing. In: *Proceedings of the First International XML Database Symposium*. Volume 2824 of *Lecture Notes on Computer Science*., Springer (Sep. 2003) 100–117

10. Fujimoto, K., Shimizu, T., Terada, N., Hatano, K., Suzuki, Y., Amagasa, T., Kinutani, H., Yoshikawa, M.: An Implementation of High-Speed and High-Precision XML Information Retrieval System on Relational Databases. In: *Advances in XML Information Retrieval and Evaluation*. Volume 3977 of *Lecture Notes in Computer Science.*, Springer (June 2006) 254–267
11. Hatano, K., Kinutani, H., Amagasa, T., Mori, Y., Yoshikawa, M., Uemura, S.: Analyzing the Properties of XML Fragments Decomposed from the INEX Document Collection. In: *Advances in XML Information Retrieval*. Volume 3493 of *Lecture Notes in Computer Science.*, Springer (May 2005) 168–182
12. Hatano, K., Kinutani, H., Watanabe, M., Mori, Y., Yoshikawa, M., Uemura, S.: Keyword-based XML Fragment Retrieval: Experimental Evaluation based on INEX 2003 Relevance Assessments. In: *Proceedings of the 2nd Workshop of the Initiative for the Evaluation of XML Retrieval*. (March 2004) 81–88
13. Clarke, C.L.A.: Controlling Overlap in Content-Oriented XML Retrieval. In: *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. (Aug. 2005) 314–321
14. Yoshikawa, M., Amagasa, T., Shimura, T., Uemura, S.: XRel: A Path-based Approach to Storage and Retrieval of XML Documents using Relational Databases. *ACM Transactions on Internet Technology* **1**(1) (Aug. 2001) 110–141
15. Salton, G., Wong, A., Yang, C.S.: A Vector Space Model for Automatic Indexing. *Communication of the ACM* **18**(11) (Nov. 1975) 613–620
16. Clark, J., DeRose, S.: XML Path Language (XPath) Version 1.0. <http://www.w3.org/TR/xpath> (Nov. 1999) W3C Recommendation 16 November 1999.
17. Liu, F., Yu, C.T., Meng, W., Chowdhury, A.: Effective Keyword Search in Relational Databases. In: *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data*, ACM (June 2006) 563–574
18. Shimizu, T., Yoshikawa, M.: XML Information Retrieval Considering Physical Page Layout of Logical Elements. In: *Proceedings of the 10th International Workshop on Web and Databases*. (June 2007) 48–49
19. Kazai, G., Lalmas, M.: INEX 2005 Evaluation Metrics. In: *Advances in XML Information Retrieval and Evaluation*. Volume 3977 of *Lecture Notes on Computer Science.*, Springer-Verlag (Jun. 2006) 16–29
20. Kekäläinen, J., Järvelin, K.: Using Graded Relevance Assessments in IR Evaluation. *Journal of the American Society for Information Science and Technology* **53**(13) (Nov. 2002) 1120–1129
21. Kazai, G., Lalmas, M.: eXtended Cumulated Gain Measures for the Evaluation of Content-Oriented XML Retrieval. *ACM Transactions on Information Systems* **24**(4) (Oct. 2006) 503–542
22. Kazai, G., Lalmas, M., de Vries, A.P.: The Overlap Problem in Content-Oriented XML Retrieval Evaluation. In: *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. (Jul. 2004) 72–79
23. Fuhr, N., Großjohann, K.: XIRQL: An XML Query Language based on Information Retrieval Concepts. *ACM Transactions on Information Systems* **22**(2) (Apr. 2004) 313–356

Study on Reranking XML Retrieval Elements Based on Combining Strategy and Topics Categorization

Jingjing Liu, Hongfei Lin, Bing Han

Department of Computer Science and Engineering, Dalian University of echnology,
Dalian, P.R. China, 116024, 86-411-84706009-3928, hflin@dlut.edu.cn

ABSTRACT

XML retrieval has attracted more and more attention and many efforts on exploiting the available content and structural information have been made to improve retrieval system performance. In this paper we mainly focused on exploiting various methods for reranking the returned XML elements based on combining document and element scores and topics categorization by classifying the tags in the structural paths constraints in the structured query.

We regarded the initializing retrieval results got by lemur toolkit as our experimental baseline. And then we used following methods for reranking the returned elements. First of all, we used feedback strategy of lemur and combined document and element scores. Secondly, we classified the topics into two categories using tags in the structural paths constraints in the structured query. Further special handlings on the category of topics finding images were made. Additionally, we applied the common method for removing the overlap from the final results before evaluation by selecting the highest scored element from each element path. The experimental results in this paper have proved that our methods contribute to enhance retrieval performance.

Categories and Subject Descriptors

H.3 [Information Storage and Retrieval]: Information Search and Retrieval

General Terms

Algorithms, Experimentation

Keywords

XML retrieval, INEX, Combining, Topics Categorization

1. INTRODUCTION

The continuous growth in XML information repositories has been matched by increasing efforts in the development of XML retrieval systems [1]. The main difference between XML retrieval and traditional information retrieval is that document components so-called XML elements instead of complete documents in response to a user query are returned in order to implement a more focused retrieval strategy. This focused retrieval approach is of particular benefit for information repositories containing long documents, or documents covering a wide variety of topics (e.g. books, user manuals, legal documents), where users' effort to locate relevant content can be reduced by directing them to the most relevant parts of these documents[1]. Most of XML retrieval systems in pervious years mainly aimed at supporting content-oriented XML retrieval and less take the consideration of the structure hints. In order to improve retrieval system performance, many efforts on exploiting the available structural information in documents have been made recently.

Based on research of INEX 2006 XML documents collection structural features and topics of ad hoc track, this paper took consideration of combining document and element scores and classifying the topics by tags in the structural paths constraints expressed in their <castitle> fields synthetically to improve the quality of retrieval system. In our experiments we used the method motivated by York University [2] who experimented to combine article and paragraph score at HARD and Genomics Tracks of TREC 2004. This paper took XML element instead of paragraph. Additionally, we found some of INEX 2006 topics in ad hoc track were special because in their <narrative> fields some constraints of how to decide

whether a returned element was relevant or not were formulated in detail. For example, if one topic whose emphasis is to find images about Napoleon I, the narrative of this topic would stress that the relevant elements should be those including at least one or more images about the general. If no image is displayed, the element is not relevant. So we first classified the topics by using tags in the structural paths constraints expressed in their <castitle> fields and then processed those topics that focused on finding images with some special different handlings.

The remainder of the paper is organized as follows. In Section 2, we introduce related work about XML retrieval of INEX in previous years, and in Section 3, we describe our method in detail which based on combining document and element scores and topics categorization by classifying the tags in the structural paths constraints expressed in their <castitle> fields. Section 4 gives our experimental results and discussion. We make concluding remarks and present future work in Section 5.

2. RELATED WORK

The widespread use of XML in digital libraries, product catalogues, and scientific data repositories and across the Web prompted the development of appropriate searching and browsing methods for XML documents [3] has attracted more and more attention. Content-oriented XML retrieval has become an area of Information Retrieval (IR) research that is receiving an increasing interest and recently much effort has made to exploit the structural hints of XML documents.

2.1 INEX

Traditional information retrieval technology can be well implemented into traditional text information management. But if it was directly applied to documents, marked XML and rich in structural information, the system would lead to many new problems. A large-scale effort has been made to improve the efficiency of XML retrieval system. For example, there already exists a very active community in the IR/ XML domain which started to work on XML search engines and XML textual data. This community is mainly organized since 2002 around the INEX initiative

(INitiative for the Evaluation of XML Retrieval) which is funded by the DELOS network of excellence on Digital Libraries and it initiates an international, coordinated effort to promote evaluation procedures for content-based XML retrieval [3].

Participants who signed in INEX have an opportunity to access the corpus and evaluate their retrieval methods using uniform scoring procedures and a forum for participating organizations to compare their results. The aim of this initiative is to provide means, in the form of a large test collection and appropriate scoring methods, for the evaluation of retrieval of XML documents [3].

INEX consist of many retrieval tasks, such ad hoc track, interactive track, document mining track, and so on. The main retrieval task to be performed in INEX 2006 is the ad-hoc retrieval of XML documents. In information retrieval literature, ad-hoc retrieval is described as a simulation of how a library might be used, and it involves the searching of a static set of documents using a new set of topics. While the principle is the same, the difference for INEX is that the library consists of XML documents, the queries may contain both content and structural conditions and, in response to a query, arbitrary XML elements may be retrieved from the library [4].

2.2 Index Reduction

For effective and efficient XML retrieval indexing plays an important role [5]. Any element can, in theory, be retrieved. It has been shown, however, that not all the elements are likely to be appreciated equally as satisfactory answers to an information need [6].

Creating an index of all overlapping XML elements is costly and time-consuming. Furthermore, retrieval of the very many, very small elements can't satisfy users because there is little relevant information contained in too small elements. Reducing some unnecessary indexing units can speed up retrieval system and cut down indexing storage size. Many efforts have been made based on how to reduce the number of indexing units without harming retrieval effectiveness.

Paper [6] which described University of Amsterdam's participation in INEX 2005 ad hoc track addressed several

different index reduction schemes. Their aim was to create a more efficient retrieval system without sacrificing retrieval effectiveness. Their main finding was that even with an 80-90% reduction in the number of indexing units, no reduction was seen in retrieval effectiveness. They mainly created two categories of indexes. One was element index and the other was article index. Element index included four sub-categories of indexes. They were overlapping element index, length based index, qrel based index and section index. And article indexing also was divided into the “normal” article index and fielded index. Here, we mainly focused on addressing what elements the qrel based index indexed. In qrel based index, only some elements with certain tag-names were indexed because they were more likely than others to be regarded as relevant. Using aforementioned indexing approaches, the size of index storage was much smaller than indexing all overlap elements. Additionally, because the indexed elements belonging to the set which were more likely retrieved, retrieval system performed still well without sacrificing retrieval effectiveness.

2.3 Combining Strategy

The paths of XML elements contain twofold information: one is to make sure that each element belongs to which document and the other is to describe the specific path information about each element. York University who participated in TREC 2004 proposed a method for building two different levels of indexes and combining two level scores of returned elements retrieved from indexes mentioned previously. The basic assumption for this combination was: if an article was hit by both searches, it should be assigned more weight than others that were hit by only one search [2]. The assessment results showed that their method could get better passage retrieval performance than others and proved that the assumption was valid.

The concrete algorithm about the method was as follows. First, they built two different levels of indexes: document level and passage level. For each topic, they did both document level search and passage level search and then they combined these two searches into one. For above-mentioned algorithm, York University participants

used different merge functions to update the weights for document and paragraph by combining the results from both indexes. If the granularity was “document”, the following merge function was used:

$$W_{dnew} = \left(W_d + \frac{\sum_{x=1}^k W_{d.x}}{|p|} \right) \cdot \log_{10}(10 * |p|) \quad (1)$$

where W_{dnew} was the new weight of the document,

W_d was the weight obtained from the document level index,

$W_{d.x}$ was the weight obtained from the paragraph level

index, x ranged from 1 to k , where k equaled to the total number of paragraphs retrieved from this document in the top 1000 paragraphs from the paragraph level index. $|p|$

was the total number of paragraphs retrieved from this document [2].

If the granularity was “passage” and the paragraphs found in a document are not adjacent, the following merge function was used to assign a new weight to each of these paragraphs:

$$W_{pnew} = (W_p + h_1 * W_d) * \lg(10 * |P|) \quad (2)$$

where W_{pnew} was the new weight of the paragraph,

W_p was the weight of the paragraph obtained from the

paragraph level index, W_d was the weight of the document

containing the paragraph, which was obtained from the

document level index, $|p|$ was the total number of

paragraphs retrieved from this document, and h_1 was a coefficient, which was set to be 3 in their experiments [2].

Their experimental results got by applying the combining algorithm showed that combining the two levels research scores was better than only using anyone of both scores.

3. RERANKING METHOD

This paper mainly focused on exploiting various methods for reranking the XML retrieval to improve retrieval system performance. Two main strategies applied in our method were that combining document and element scores and topics categorization by classifying the tags in the structural paths constraints expressed in their <castitle> fields.

Based on Wikipedia element frequency shown as Table 1, we took full use of the structural and content information of XML document and created two level indexes. They were document level index and element level index. The document index was built by using the traditional IR indexing method. All XML documents were indexed and if retrieved, the returned results were independent XML documents.

Table.1 Wikipedia Element Frequency

Tag Name	Avg. Freq. In Documents	Freq. In Collection
<collectionlink>	25.80	17,14,573
<item>	8.61	5,682,358
<unknownlink>	5.98	3,947,513
cell	5.71	3,770,196
p	4.17	2,752,171
emph2	4.12	2,721,840
template	3.68	2,427,099
section	2.44	1,609,725
title	2.41	1,592,215
emph3	2.24	1,480,877

The element level index made some differences from the document index. We analyzed the Wikipedia corpus and intended to find which element was retrieved more frequently than others. We also investigated all the fields of topics in INEX 2006 ad hoc track and found those elements with tags that appeared relatively frequently in the topics set should be indexed in the element level index. By taking into account the information of Wikipedia element frequency shown as Table 1 and the hints from all the fields of topics in INEX 2006 ad hoc track, we made certain that elements whose tag names corresponding with our criterion were shown as Table 2. Because building all overlap elements is not an easy thing and the retrieval

speed could be reduced if the index storage is too large, so we merely indexed elements with 10 categories of elements whose tag-names shown as Table 2.

Table.2 Tag Set

No.	Tag Name	No.	Tag Name
1	<article>	6	<title>
2	<section>	7	<name>
3	<body>	8	<caption>
4	<p>	9	<image>
5	<table>	10	<figure>

Feedback strategy of lemur toolkit was also applied for searching from above-mentioned two levels of indexes. We used formula.2 to combine document and element score. From the results of our combining experiments, we found that most results of finding images were not high. The analysis on all fields of INEX 2006 topics told us that those topics even had other restricts on their relevant returned elements. For example, if the path in <castitle> field of one topic is //article [about (., China)]//image [about (., "Great Wall")], the author of this topic commonly wanted to get the images about the Great Wall of China. If no image was played, the returned results should be not relevant. This criteria of how to decide whether an element relevant or not declared that if the element path contains the tag image or figure, it could be relevant in all probability. So we used an easy method for determining which topic should be selected by classifying the tags in the structural paths constraints expressed in topics' <castitle> fields. If a tag was named with image or figure in the structural paths constraints expressed in the <castitle> field of one topic, this topic should be listed out. Then we made some special handlings on such topics, which are explained in detail below. Based on the new results by combining document and element score which retrieved from the document level index and the element level index, we removed the returned elements whose paths contained certain tag named with image or figure to the top one by one according to the previous order. And the remained ones were also removed backward orderly. Our new evaluation results showed most of those topics listed out had much certain increase.

Our experiments were all about focused sub-task and the

sub-task asks systems to return a ranked list of elements to the user and overlap was not permitted in the submitted run. So we must find some measures to remove overlap elements from the returned elements. To identify the appropriate element to return was not an easy problem and a common approach to remove overlap from result lists was to select the highest scored element from each of the paths [7]. Though this approach had some drawbacks, it was easy to be implemented into our system. Besides, no matter what the baseline or the final experimental results obtained by applying the new methods used the same method to remove overlap. So removing overlap didn't influence the comparisons between the baseline and the new experimental results.

4. EXPERIMENTS

4.1 INEX Test Collection

The test collection of INEX 2006 consisted of a set of XML documents, topics and relevance assessments and it uses a document collection made from English documents from Wikipedia. The collection was so far made up of the full-texts, marked-up in XML, of 659,388 articles of the Wikipedia project, covering a hierarchy of 113,483 categories, and totaling more than 60 Gigabytes (4.6 Gigabytes without images) and had a structure containing text, more than 300,000 images and some structured part corresponding to the Wikipedia templates (about 5000 different tags) [4]. The number of XML nodes an article contains on average was 161.35, where the average depth of an element was 6.72. Each participating group was asked to create a set of candidate topics, which were representative of a range of real user needs over the XML collection [4]. From the pooled set of candidate topics INEX 126 topics were selected as a final set of topics to form part of the INEX test collection. Every topic consists of the following parts: <title>, <castitle>, <description>, <narrative>, <ontopic_keywords>, <offtopic_keywords>. Within the ad-hoc XML retrieval task it defined the following four sub-tasks [8]: thorough task, focused task, relevant in context task and best in context task. Here, this paper just showed the experimental results comparisons of the focused sub-task.

4.2 Evaluation Metrics

The general aim of an IR system is to find relevant information for a given topic of request to meet users' requirement. In the case of XML retrieval there is, for each article containing relevant information, a choice from a whole hierarchy of different elements to return. Hence, within XML retrieval, we regard as relevant elements those XML elements that both [8]

- 1) contain relevant information (the element exhaustively discusses the topic), but
- 2) do not contain non-relevant information (the element is specific for the topic).

The evaluation of the retrieval effectiveness of the XML retrieval engines used by the participants would be based on the constructed INEX test collection and uniform scoring techniques. Since its launch in 2002, the issue of how to measure an XML information access system's effectiveness challenged to INEX. Then in 2005, INEX adopted a new set of metrics, the eXtended Cumulated Gain (XCG) metrics [9] to support the evaluation of XML retrieval engines, which was also used in INEX 2006[4].

4.3 Results and Discussion

This paper mainly addressed that the XML retrieval system could perform better by combining the document level score and the element level score and classifying the tags of target elements. The next paragraph showed a part of results comparison about our experiments and the details was explained below.

There were four experiments where four approaches were applied to and the description of every experiment in detail is shown in Table 3. Among them, Method B stood up the baseline method. The result got by using this method was the baseline of this paper and was also our initializing result for after-processing.

Table 3. Descriptions of all experiments

Method	Description
B	1. lemur retrieval model
BF	1. feedback 2. lemur retrieval model
BFC	1. feedback 2. lemur retrieval model

	3. combine the document and element score
BFCC	1. feedback 2. lemur retrieval model 3. combine the document and element score 4. topics categorization by classifying the tags in the structural paths constraints expressed in their <castitle> fields and make some different handlings on the topics of finding image

Table 4 showed results using different methods in focused sub-task of ad hoc track. The baseline got by using method B was 0.4327 at nxCG@5 and when applied our method BFCC, the value at nxCG@5 was 0.4703. The comparison showed the method proposed in this paper was feasible and valid to some extent.

Table 4. Comparison results of focused experiments

Method	nxCG@5	nxCG@10	nxCG@15
B	0.4327	0.383	0.3547
BF	0.4406	0.3763	0.3479
BFC	0.4632	0.4121	0.3878
BFCC	0.4703	0.4161	0.391

Additionally, we also intended to prove that classifying the tags of target element paths could help to optimize our system performance. There are 6 topics selected for further special handlings. Table 5 and 6 showed the results of topic 292 and topic 374 on the condition of un-classifying and classifying topics. Obviously, classifying topics using tags in the structural paths constraints expressed in their <castitle> fields advances retrieval effectiveness.

Besides the comparisons we obtained above-mentioned, we also found that values at nxCG@n (n=5, 10, 15...) of topic 291 shown as Table 7 declined. We took analysis on the returned elements of topic 291 and discovered that the relevance of some elements whose paths contain tag named with image or figure are too low, so they may not be what the users hope for. In the special handlings, they with less relevance were removed up instead of ones with more relevance. If that occurred, retrieval effectiveness would be cut down and values at nxCG@n (n=5, 10, 15...) reduced. So the method for classifying topics using tags in the structural paths constraints expressed in their <castitle>

fields all the same had some drawbacks and in our future work it needs more and further studies.

Table 5. Comparison results of topic 292 between un-Classifying and Classifying Topics

Metrics	un-Classify Topics	Classify Topics
nxCG@5	0.0319	0.0319
nxCG@10	0.0228	0.0962
nxCG@15	0.0152	0.1308
nxCG@25	0.0091	0.0785

Table 6. Comparison results of topic 374 between un-Classifying and Classifying Topics

Metrics	un-Classify Topics	Classify Topics
nxCG@5	0.1431	0.6
nxCG@10	0.1142	0.3119
nxCG@15	0.1007	0.2271
nxCG@25	0.0604	0.1979

Table 7. Comparison results of topic 291 between un-Classifying and Classifying Topics

Metrics	Un-Classify Topics	Classify Topics
nxCG@5	0.0019	0.0019
nxCG@10	0.0051	0.001
nxCG@15	0.0034	0.0006
nxCG@25	0.002	0.0004
nxCG@50	0.003	0.019

5. CONCLUSION AND FUTURE WORK

This paper proposed a new method for fully utilizing the strategies of selective index and topics categorization by classifying the tags in the structural paths constraints expressed in their <castitle> fields whose concrete information was explained in Section 4. Our method was motivated by statistical information of the Wikipedia corpus and all fields of INEX 2006 topics. Just indexing the elements with certain tag-names which appeared frequently in INEX 2006 document collection and topics reduced the indexing storage size without sacrificing retrieval system effectiveness. Because indexed elements belonging to the set which were more likely retrieved, our retrieval system performed still well and effectively. Additionally, the results shown as Table 5 and 6 told us that topics categorization by classifying the tags in the

structural paths constraints expressed in their <castitle> fields to some extent enhance values of most topics at nxCG@n (n=5, 10, 15...) though values of few topics had smaller reduction.

In the future, we plan to continue the experiments of different scenarios for ranking the research results to get better retrieval system performance. For example, we will take into account of the length of returned elements to remove overlap in order to identify the appropriate element granularity. Too large or too small elements should be abandoned and identifying the appropriate element granularity needs more effort and new ideas for better and more effective algorithms. Additionally, in the future research, we also intend to find a proper query expansion and feedback algorithm for improving retrieval system performance to meet users' needs. How to decide whether an element is relevant or not and build mixture models useful for ranking XML elements still remain as future work needing more attention and further research.

6. ACKNOWLEDGE

This research was supported by grant from the Natural Science Foundation of China (No.60373095 and 60673039) and the National High Tech Research and Development Plan of China (2006AA01Z151).

7. REFERENCES

[1] <http://inex.is.informatik.uni-duisburg.de/2007/>
[2] Xiangji Huang, Yanrui Huang, Miao Wen and Ming

Zhong. York University at TREC 2004: HARD and Genomics Tracks[C]. In Proceedings of the 13th Text Retrieval Conference, 2004.

[3] <http://qmir.dcs.qmw.ac.uk/INEX/>
[4] <http://inex.is.informatik.uni-duisburg.de/2006/>
[5] B. Sigurbjornsson, J. Kamps and M. de Rijke. The Effect of Structured Queries and Selective Indexing on XML Retrieval, INEX 2005.
[6] B. Sigurbjornsson, J. Kamps & M. de Rijke. The importance of length normalization for XML retrieval. Information Retrieval, 8(4): 631–654, 2005.
[7] Mihajlovic, V., Ramirez, G., Westerveld, T., Block, H., de Vries, A., and Hiemstra, D. TIJAH scratches INEX 2005 vague element selection, overlap, image search, relevance feedback, and users. In INEX 2005 Workshop Proceedings, Germany, 2005, 54, 71.
[8] Charlie Clarke, Jaap Kamps, Mounia Lalmas. INEX 2006 retrieval task and result submission specification. In INEX 2006 Workshop Pre-Proceedings, 2006, 381-388.
[9] G. Kazai, M. Lalmas, and A.P. de Vries. The Overlap Problem in Content-oriented XML Retrieval Evaluation. In Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2004.

BookSearch'07: INEX 2007 Book Search Track Overview

Gabriella Kazai¹ and Antoine Doucet²

¹ Microsoft Research Cambridge, United Kingdom

`gabkaz@microsoft.com`

² University of Caen, France

`doucet@info.unicaen.fr`

Abstract. This paper describes the new Book Search Track launched at INEX 2007.

1 Introduction

Searching for information in a collection of books is seen as one of the natural application areas of XML retrieval (and more generally, of structured document retrieval), where a clear benefit to users is to gain direct access to parts of books relevant to their information need. The ultimate goal of the track is to investigate book-specific relevance ranking strategies, UI issues and user behaviour, exploiting special features, such as back of book indexes provided by authors, and linking to associated metadata like catalogue information from libraries. However, searching over a large collection of books comes with many new challenges that need to be addressed first. For example, proper infrastructure has to be developed to allow for the scalable storage, indexing and retrieval of the content. In its first year, the track will explore these issues with the aim to provide a set of recommendations for setting up such an infrastructure. The track will also aim to run a similar task to INEX's ad-hoc track, where participants can evaluate their relevance ranking strategies.

At the time of writing this, participants were due to submit their retrieval runs. Therefore, we can only report on limited aspects of the track.

This paper is organised as follows. Section 2 gives a brief summary of the participating organisations. Section 3 details the book corpus and test topics used as the basis for the track. In Section 4, we briefly describe the retrieval tasks at BookSearch'07.

2 Participating organizations

In response to the call for participation, issued in April 2007, 27 organizations registered for the track. Of these only a handful of groups are actually active in the track. Most groups reported difficulties due to lack of sufficient resources, including space to store the dataset or scalable approach to process it, as well as lack of time or human resources.

The 27 groups along with details of their participation are summarized in Table 2.

3 Test Collection

3.1 Book corpus

The corpus is provided by Microsoft Live Book Search and the Internet Archive (for non-commercial purposes only). It consists of 42049 digitized out-of-copyright books (210Gb). The OCR content of the books is stored in djvu.xml format. Each book also has an associated metadata file (*.mrc), which contains publication (author, title, etc.) and classification information in MACHine-Readable Cataloging (MARC) record format. The basic XML structure of a book (djvu.xml) is as follows:

```
<DjVuXML>
<BODY>
  <OBJECT data="file.." ...>
    <PARAM name="PAGE" value="..">
      [...]
    <REGION>
      <PARAGRAPH>
        <LINE>
          <WORD coords="..." />
          <WORD coords="..." />
        </LINE>
      </PARAGRAPH>
    </REGION>
    [...]
  </OBJECT>
  [...]
</BODY>
</DjVuXML>
```

Essentially, an <OBJECT> corresponds to a page. A page counter is embedded in the @value attribute of the <PARAM> element with the @name="PAGE" attribute. The actual page numbers (as printed inside the book) can be found (not always) in the header/footer of a page. Note, however, that headers/footers are not explicitly recognised in the OCR: i.e. the first paragraph on a page could be a header and the last one or more paragraphs on a page could be part of a footer. Depending on the book, headers may include chapter titles and page numbers (although due to OCR error, the page number is not always present).

3.2 Topics

Topics are representations of users' information needs. Some topics were extracted from the query log of Live Search Books and others were created by

ID	Organisation	Cancelled	Corpus download	Topics created	Runs
1	University of Kaiserslautern, AG DBIS	Y	Y		
2	University of California, Berkeley		Y		
4	University of Granada	Y	Y		
5	Lexiclone Inc				
9	Queensland University of Technology		Y		
10	University of Otago				
12	University of Strathclyde	Y			
14	Center for Studies of Information Resources, School of Information Management, Wuhan University, China				
19	Indian Statistical Institute	Y	Y		
20	LAMSADE				
22	Doshisha University		Y	1	
23	Kyungpook National University		Y	1	
25	Max-Planck-Institut für Informatik		Y		
26	Dalian University of Technology		Y	5	
28	University of Helsinki		Y	2	
32	RMIT University				
33	Information Engineering Lab, ICT Centre, CSIRO				
36	University of Amsterdam		Y	3	
37	University of Waterloo	Y	Y		
40	Language Technologies Institute, School of Computer Science, Carnegie Mellon University		Y		
42	LIP6				
53	Ecoles des Mines de Saint-Etienne, France				
54	Microsoft Research, Cambridge		Y	13	
55	University of Tampere		Y	5	
61	Hong Kong University of Science and Technology				
68	University of Salford, UK				
92	Cairo Microsoft Innovation Center		Y		
	Total (27 organizations)	5	16	30	

Table 1. Participating groups at BookSearch'07

the participating organisations. For this year, topics were limited to deal with content only aspects (i.e., no structural conditions). The structure of books, however, could still be used by search engines to improve their ranking of book parts estimated relevant to the query.

Topic Format. Topics are made up of several parts, each of which describing the same information need, but for different purposes and at different levels of detail.

<title> Represents the search query that will be used by the search engines. It serves as a summary of the content of the user’s information need.

<description> A natural language definition of the information need.

<narrative> A detailed explanation of the information need and a description of what makes an element relevant or not. The narrative must be a clear and precise description of the information need in order to unambiguously determine whether or not a given text fragment in a book fulfils the need. The narrative is taken as the only true and accurate interpretation of the user’s needs. Relevance assessments will be made on compliance to the narrative alone. Precise recording of the narrative is important for scientific repeatability. To aid this, the narrative should explain not only what information is being sought, but also the context and motivation of the information need, i.e., why the information is being sought and what work-task it might help to solve. The narrative, hence, should contain the following:

<task> A description of the task for which information is sought, specifying the context, background and motivation for the information need.

<infneed> A detailed explanation of what information is sought and what is considered relevant or irrelevant.

An example topic is given in Figure 1.

3.3 Collected Topics

250 queries were extracted from the query logs of Live Search Books for which the test corpus contains relevant books. No additional information was available for these. Therefore these topics only include the topic title, while both description and narrative are missing. These queries were then used for the Book Retrieval Task.

For the Page in Context task, participants created a total of 30 topics, all of which include topic title, description and narrative. Some of these topics were created from the queries extracted from the logs. Table 2 lists the number of topics participants contributed.

Because the performance of retrieval systems varies largely for different topics, to judge whether one retrieval strategy is (in general) more effective than another, the retrieval performance must be averaged over a large and diverse set of topics. In particular, the topics need to be diverse and differ in their coverage


```

<title>Octavius Antony Cleopatra conflict ‘‘Donations of Alexandria"
‘‘battle of Actium"</title>
<description>I am looking for information on the conflict between
Octavius, Antony and Cleopatra. I am interested to learn about events
like the Donations of Alexandria and the battle of Actium.
</description>
<narrative>
<task>I am writing an essay on the relationship of Antony and Cleopatra
and currently working on a chapter that will explore the conflict
between Octavius, the brother of Antony’s wife, Octavia, and the
lovers. </task>
<infneed>Of interest is any information that details what motivated the
conflict, how it developed and evolved through events such as the
ceremony known as the Donations of Alexandria, Octavius’ propaganda
campaign in Rome against Antony, Antony’s divorce from Octavia, and the
battle of Actium in 31BC. Any information on the actions and emotions of
the lovers during this period is relevant. Any non-documentary or
non-biographical information, such as theatre plays (e.g. Shakespeare’s
play) or their critics are not relevant.</infneed>
</narrative>

```

Fig. 1. A sample topic of BookSearch’07

(e.g., broad or narrow). To be able to select and categorize candidate topics, we provided guidelines [5] and a tool to assist participants in the topic creation process. A screenshot of the tool is given in Figure 2.

This software gives participants the means to explore the corpus by interfacing with Live Search Books³. It takes advantage of the fact that all the books of the BookSearch’07 collection also belong to the index of Live Search Books, by first sending out user queries to the search engine, and then filtering the result set to only include books from the BookSearch’07 corpus. The visualization of answers again relies on the Live Search Books service.

The system provided made it easy to use the Live Search Book features over the BookSearch’07 corpus. Hence, participants were given an easy way to familiarize themselves with the collection and to determine more easily whether a candidate topic was “suitable” or not (topics with too few or too many relevant answers were to be abandoned [5]).

4 Retrieval Tasks

4.1 Book Retrieval Task

The goal of this task was to investigate the impact of book specific features on the effectiveness of book retrieval systems, where the unit of retrieval is the

³ <http://books.live.com>

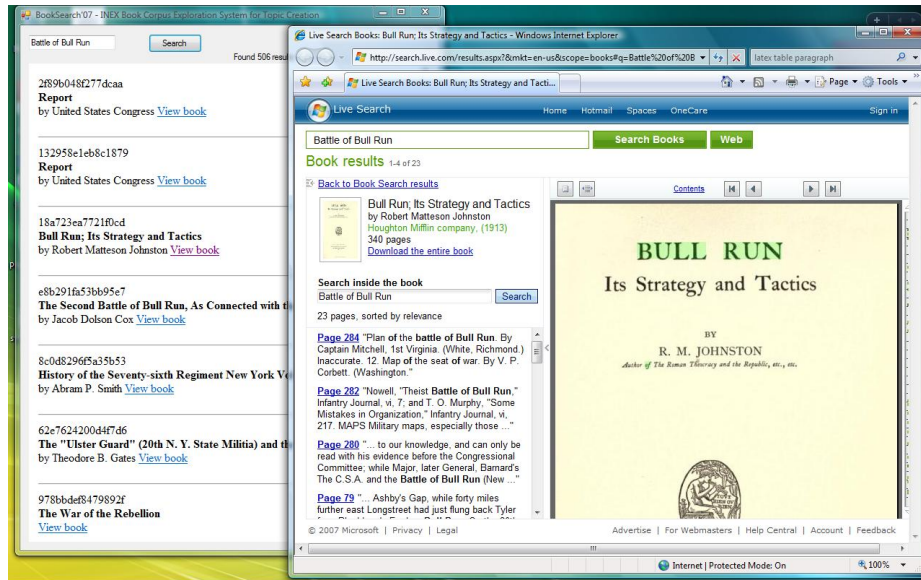


Fig. 2. Screenshot of the system to assist topic creation at BookSearch'07

(complete) book. Users are thus assumed to be searching for (whole) books relevant to their information need that they can, e.g., borrow from a Library.

Participants of this task were invited to submit pairs of runs, where one run may be the result of applying generic IR techniques to return a ranked list of books to the user in response to a query. The other run had to be generated using the same techniques (where possible) but with the use of additional book-specific features (e.g. back-of-book index, citation statistics, in or out of print, etc.) or specifically tuned methods. In both cases, a result list had to contain a maximum of 1000 books estimated relevant to the given topic, ranked in order of estimated relevance to the query.

The test queries used for this task were extracted from the query log of Live Search Books, and relevance judgements have been collected on a four point scale: Excellent, Good, Fair, and Not-relevant. The evaluation (subject to change) will be based on the measure of Normalised Discounted Cumulated Gain at various cut-off values [3, 6].

Participants could submit up to 3 pairs of runs.

4.2 Page in Context Task

Based on the assumption of an informational user request, the task of a book search system is to return the user a ranked list of books estimated relevant to the user need, and then present within each book, a ranking of relevant non-overlapping XML elements, passages or pages.

This task is based on topics created by the participants. Similarly as in the INEX ad hoc track, relevance judgements will be collected from participants in the phase following the result submissions. Evaluation measures will be selected and adopted from those used at the ad hoc track [2] (subject to change).

Participants could submit up to 10 runs. One automatic (title-only) and one manual runs were compulsory. Additional manual runs were encouraged in order to help the construction of a reliable test collection. Each run could contain for each topic a maximum of 1000 books estimated relevant to the given topic, ordered by decreasing value of relevance. For each book, a ranked list of non-overlapping XML element, passage or book page results estimated relevant had to be listed in decreasing order of relevance. A minimum of 1 result per book had to be returned. A submission could only contain one type of result, i.e. only book pages or only passages; result types cannot be mixed. Further details are available in the tasks and submission Guidelines [4].

4.3 Classification Task

In this task, systems were tested on their ability to assign the correct classification labels from the Library of Congress (LoC) classification scheme to the books of the test corpus based only on information available from the full text of the books. The distributed corpus of about 42,000 books served as the training corpus for this task: The classification labels were given in the MACHine-Readable Cataloging (MARC) files. A test corpus contained 2 sets of 1,000 books.

Participants were allowed to submit up to three runs per test set. Each run had to contain all books of the test set, and for each book include a ranked list (or set) of assigned classification labels in the form of (BookId, LoC Classifications) pairs. Classification (tagging) accuracy will be measured using standard measures such as F1 and rank-based metrics.

The Library of Congress classification headings extracted from each book's MARC record was made available by the organisers.

4.4 Taxonomy of User Intent Task

User intent is a critical component in the understanding of users' search behaviour. It defines what kinds of search tasks users engage in. In traditional information retrieval a user's intent is assumed to be informational in nature: It is driven by the user's need for information in order to complete a task at hand. Observations of Web use resulted in further two categories: navigational and transactional [1]. It is clear that these can also be applied to the book domain. However, it is possible that there are additional classes of user intent which are specific to books. It may also be the case that user tasks and user behaviour in the book domain will have specific traits and characteristics. What are the possible classes of user intent and user tasks and what properties they have is a research question that this task aims to explore.

The goal of this task was to derive a taxonomy of user intent with its associated properties and search tasks. The use of samples of users' (actual or

hypothetic) information needs demonstrating each class of intent and task was encouraged. The taxonomy could extend to include both research and design questions and possible answers regarding how a given user behaviour might be supported by a search system and its user interface. For example, a user hoping to buy a book as a present is likely to be more interested in a system function that compares retail prices, while an informational query will more likely benefit from a “find related books” feature.

Examples of questions that could be explored included: How is user intent dependent on the genre of books? Which book specific features best support which kind of intent and task? How could intent be extracted from query logs? How should one design experiments to allow for the identification of user intent from system logs? What data would enable the prediction of intent in order to aid users? What user behaviour follows from them?

Participation in this task involved the submission of a research or opinion paper detailing the proposed taxonomy. Participants could choose to report findings from the analysis of collected user log data or provide recommendations for the design of user studies to help elicit such data.

4.5 Open Task

Participants were invited to carry out and evaluate their own tasks and/or submit task proposals discussing motivation, required infrastructure and potential benefits for running the task.

Acknowledgements

We thank Steven Robertson, Nick Craswell and Natasa Milic-Frayling for their valuable comments on aspects of the organisation of the track.

Bibliography

- [1] A. Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, 2002. ISSN 0163-5840.
- [2] INEX. INitiative for the Evaluation of XML retrieval, 2007. <http://inex.is.informatik.uni-duisburg.de/>.
- [3] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, 2002. ISSN 1046-8188.
- [4] G. Kazai. INEX 2007 book search track, tasks and submission guidelines. In *This Volume*, 2007.
- [5] G. Kazai. INEX 2007 book search track, topic development guidelines. In *This Volume*, 2007.
- [6] T. Sakai. On the reliability of information retrieval metrics based on graded relevance. *Inf. Process. Manage.*, 43(2):531–548, 2007.

Logistic Regression and EVIs for XML Books and the Heterogeneous track

Ray R. Larson

School of Information
University of California, Berkeley
Berkeley, California, USA, 94720-4600
ray@ischool.berkeley.edu

Abstract. For this year's INEX UC Berkeley focused on the Book track and the Heterogeneous track, For these runs we used the TREC2 logistic regression probabilistic model with blind feedback as well as Entry Vocabulary Indexes (EVIs) for the Books Collection MARC data. For the full text records of the book track we encountered a number of interesting problems in setting up the database, and ended up using a combination of page-level indexing of the full collection. As of this writing the submission system and evaluation for the Book track are not yet ready, so there are no official results to report in this paper.

As (once again) the only group to actually submit runs for the Het track, we are guaranteed both the highest, and lowest, effectiveness scores for each task. However, because it was again deemed pointless to conduct the actual relevance assessments on the submissions of a single system, we do not know the exact values of these results.

1 Introduction

In this paper we will first discuss the algorithms and fusion operators used in our official INEX 2007 Book Track and Heterogenous (Het) track runs. Then we will look at how these algorithms and operators were used in the various submissions for these tracks, and finally we will discuss problems in implementation, and directions for future research.

2 The Retrieval Algorithms and Fusion Operators

This section describes the probabilistic retrieval algorithms used for both the Adhoc and Het track in INEX this year. These are the same algorithms that we have used in previous years for INEX, and also include the addition of a blind relevance feedback method used in combination with the TREC2 algorithm. In addition we will discuss the methods used to combine the results of searches of different XML components in the collections. The algorithms and combination methods are implemented as part of the Cheshire II XML/SGML search engine [18, 17, 15] which also supports a number of other algorithms for distributed search and operators for merging result lists from ranked or Boolean sub-queries.

2.1 TREC2 Logistic Regression Algorithm

Once again the principle algorithm used for our INEX runs is based on the *Logistic Regression* (LR) algorithm originally developed at Berkeley by Cooper, et al. [8]. The version that we used for Adhoc tasks was the Cheshire II implementation of the “TREC2” [6, 5] that provided good Thorough retrieval performance in the INEX 2005 evaluation [18]. As originally formulated, the LR model of probabilistic IR attempts to estimate the probability of relevance for each document based on a set of statistics about a document collection and a set of queries in combination with a set of weighting coefficients for those statistics. The statistics to be used and the values of the coefficients are obtained from regression analysis of a sample of a collection (or similar test collection) for some set of queries where relevance and non-relevance has been determined. More formally, given a particular query and a particular document in a collection $P(R | Q, D)$ is calculated and the documents or components are presented to the user ranked in order of decreasing values of that probability. To avoid invalid probability values, the usual calculation of $P(R | Q, D)$ uses the “log odds” of relevance given a set of S statistics derived from the query and database, such that:

$$\begin{aligned} \log O(R|C, Q) &= \log \frac{p(R|C, Q)}{1 - p(R|C, Q)} = \log \frac{p(R|C, Q)}{p(\bar{R}|C, Q)} \\ &= c_0 + c_1 * \frac{1}{\sqrt{|Q_c| + 1}} \sum_{i=1}^{|Q_c|} \frac{qt f_i}{ql + 35} \\ &+ c_2 * \frac{1}{\sqrt{|Q_c| + 1}} \sum_{i=1}^{|Q_c|} \log \frac{tf_i}{cl + 80} \\ &- c_3 * \frac{1}{\sqrt{|Q_c| + 1}} \sum_{i=1}^{|Q_c|} \log \frac{ct f_i}{N_t} \\ &+ c_4 * |Q_c| \end{aligned}$$

where C denotes a document component and Q a query, R is a relevance variable, and

$p(R|C, Q)$ is the probability that document component C is relevant to query Q ,

$p(\bar{R}|C, Q)$ the probability that document component C is not relevant to query Q , (which is $1.0 - p(R|C, Q)$)

$|Q_c|$ is the number of matching terms between a document component and a query,

$qt f_i$ is the within-query frequency of the i th matching term,

tf_i is the within-document frequency of the i th matching term,

$ct f_i$ is the occurrence frequency in a collection of the i th matching term,

ql is query length (i.e., number of terms in a query like $|Q|$ for non-feedback situations),

cl is component length (i.e., number of terms in a component), and N_t is collection length (i.e., number of terms in a test collection). c_k are the k coefficients obtained through the regression analysis.

Assuming that stopwords are removed during index creation, then ql , cl , and N_t are the query length, document length, and collection length, respectively. If the query terms are re-weighted (in feedback, for example), then qtf_i is no longer the original term frequency, but the new weight, and ql is the sum of the new weight values for the query terms. Note that, unlike the document and collection lengths, query length is the “optimized” relative frequency without first taking the log over the matching terms.

The coefficients were determined by fitting the logistic regression model specified in $\log O(R|C, Q)$ to TREC training data using a statistical software package. The coefficients, c_k , used for our official runs are the same as those described by Chen[3]. These were: $c_0 = -3.51$, $c_1 = 37.4$, $c_2 = 0.330$, $c_3 = 0.1937$ and $c_4 = 0.0929$. Further details on the TREC2 version of the Logistic Regression algorithm may be found in Cooper et al. [6].

2.2 Blind Relevance feedback

It is well known that blind (also called pseudo) relevance feedback can substantially improve retrieval effectiveness in tasks such as TREC and CLEF. (See for example the papers of the groups who participated in the Ad Hoc tasks in TREC-7 (Voorhees and Harman 1998)[22] and TREC-8 (Voorhees and Harman 1999)[23].)

Blind relevance feedback is typically performed in two stages. First, an initial search using the original queries is performed, after which a number of terms are selected from the top-ranked documents (which are presumed to be relevant). The selected terms are weighted and then merged with the initial query to formulate a new query. Finally the reweighted and expanded query is run against the same collection to produce a final ranked list of documents. It was a simple extension to adapt these document-level algorithms to document components for INEX.

The TREC2 algorithm has been combined with a blind feedback method developed by Aitao Chen for cross-language retrieval in CLEF. Chen[4] presents a technique for incorporating blind relevance feedback into the logistic regression-based document ranking framework. Several factors are important in using blind relevance feedback. These are: determining the number of top ranked documents that will be presumed relevant and from which new terms will be extracted, how to rank the selected terms and determining the number of terms that should be selected, how to assign weights to the selected terms. Many techniques have been used for deciding the number of terms to be selected, the number of top-ranked documents from which to extract terms, and ranking the terms. Harman [12] provides a survey of relevance feedback techniques that have been used.

Lacking comparable data from previous years, we adopted some rather arbitrary parameters for these options for INEX 2007. We used top 10 ranked

components from the initial search of each component type, and enhanced and reweighted the query terms using term relevance weights derived from well-known Robertson and Sparck Jones[21] relevance weights, as described by Chen and Gey[5]. The top 10 terms that occurred in the (presumed) relevant top 10 documents, that were not already in the query were added for the feedback search.

2.3 TREC3 Logistic Regression Algorithm

In addition to the TREC2 algorithm described above, we also used the TREC3 algorithm in some of our Het track runs. This algorithm has been used repeatedly in our INEX work, and described many times, but we include it below for ease of comparison. The full equation describing the ‘‘TREC3’’ LR algorithm used in these experiments is:

$$\begin{aligned}
 \log O(R | Q, C) = & \\
 & b_0 + \left(b_1 \cdot \left(\frac{1}{|Q_c|} \sum_{j=1}^{|Q_c|} \log qtf_j \right) \right) \\
 & + \left(b_2 \cdot \sqrt{|Q|} \right) \\
 & + \left(b_3 \cdot \left(\frac{1}{|Q_c|} \sum_{j=1}^{|Q_c|} \log tf_j \right) \right) \tag{1} \\
 & + \left(b_4 \cdot \sqrt{cl} \right) \\
 & + \left(b_5 \cdot \left(\frac{1}{|Q_c|} \sum_{j=1}^{|Q_c|} \log \frac{N - n_{t_j}}{n_{t_j}} \right) \right) \\
 & + (b_6 \cdot \log |Q_d|)
 \end{aligned}$$

Where:

- Q is a query containing terms T ,
- $|Q|$ is the total number of terms in Q ,
- $|Q_c|$ is the number of terms in Q that also occur in the document component,
- tf_j is the frequency of the j th term in a specific document component,
- qtf_j is the frequency of the j th term in Q ,
- n_{t_j} is the number of components (of a given type) containing the j th term,
- cl is the document component length measured in bytes.
- N is the number of components of a given type in the collection.
- b_i are the coefficients obtained through the regression analysis.

This equation, used in estimating the probability of relevance for some of the Het runs in this research, is essentially the same as that used in [7]. The b_i coefficients in the original version of this algorithm were estimated using relevance

judgements and statistics from the TREC/TIPSTER test collection. In INEX 2005 we did not use the original or “Base” version, but instead used a version where the coefficients for each of the major document components were estimated separately and combined through component fusion. This year, lacking relevance data from Wikipedia for training, we used the base version again. The coefficients for the Base version were $b_0 = -3.70$, $b_1 = 1.269$, $b_2 = -0.310$, $b_3 = 0.679$, $b_4 = -0.0674$, $b_5 = 0.223$ and $b_6 = 2.01$.

2.4 CORI Collection ranking algorithm

The resource selection task in the Heterogeneous track is basically the same as the collection selection task in distributed IR. For this task we drew on our previously experiments with distributed search and collection ranking [15, 16], where we used the above “TREC3” algorithm for collection selection and compared it with other reported distributed search results.

The collection selection task attempts to discover which distributed databases are likely to be the best places for the user to begin a search. This problem, distributed information retrieval, has been an area of active research interest for many years. Distributed IR presents three central research problems:

1. How to select appropriate databases or collections for search from a large number of distributed databases;
2. How to perform parallel or sequential distributed search over the selected databases, possibly using different query structures or search formulations, in a networked environment where not all resources are always available; and
3. How to merge results from the different search engines and collections, with differing record contents and structures (sometimes referred to as the collection fusion problem).

Each of these research problems presents a number of challenges that must be addressed to provide effective and efficient solutions to the overall problem of distributed information retrieval. Some of the best known work in this area has been Gravano, et al’s work on GLOSS [11, 10] and Callan, et al’s [2, 24, 1] application of inference networks to distributed IR. One of the best performing collection selection algorithms developed by Callan was the “CORI” algorithm. This algorithm was adapted for the Cheshire II system, and used for some of our Resource Selection runs for the Het track this year. The CORI algorithm defines a belief value for each query term using a form of tfidf ranking for each term and collection:

$$T = \frac{df}{df + 50 + 150 \cdot cw/\overline{cw}}$$

$$I = \frac{\log(\frac{|DB|+0.5}{cf})}{\log(|DB| + 1.0)}$$

$$p(r_k|db_i) = 0.4 + 0.6 \cdot T \cdot I$$

Where:

df is the number of documents containing terms r_k ,
 cf is the number of databases or collections containing r_k ,
 $|DB|$ is the number of databases or collections being ranked,
 cw is the number of terms in database or collection db_i ,
 \overline{cw} is the average cw of the collections being ranked, and
 $p(r_k|db_i)$ is the belief value in collection db_i due to observing term r_k

These belief values are summed over all of the query terms to provide the collection ranking value.

2.5 Result Combination Operators

As we have also reported previously, the Cheshire II system used in this evaluation provides a number of operators to combine the intermediate results of a search from different components or indexes. With these operators we have available an entire spectrum of combination methods ranging from strict Boolean operations to fuzzy Boolean and normalized score combinations for probabilistic and Boolean results. These operators are the means available for performing fusion operations between the results for different retrieval algorithms and the search results from different different components of a document. For Heterogeneous search we used a variant of the combination operators, where MINMAX normalization across the probability of relevance for each entry in results from each sub-collection was calculated and the final result ranking was based on these normalized scores.

In addition, for the Adhoc Thorough runs we used a merge/reweighting operator based on the ‘‘Pivot’’ method described by Mass and Mandelbrod[19] to combine the results for each type of document component considered. In our case the new probability of relevance for a component is a weighted combination of the initial estimate probability of relevance for the component and the probability of relevance for the entire article for the same query terms. Formally this is:

$$P(R | Q, C_{new}) = (X * P(R | Q, C_{comp})) + ((1 - X) * P(R | Q, C_{art})) \quad (2)$$

Where X is a pivot value between 0 and 1, and $P(R | Q, C_{new})$, $P(R | Q, C_{comp})$ and $P(R | Q, C_{art})$ are the new weight, the original component weight, and article weight for a given query. Although we found that a pivot value of 0.54 was most effective for INEX04 data and measures, we adopted the ‘‘neutral’’ pivot value of 0.5 for all of our 2007 adhoc runs, given the uncertainties of how this approach would fare with the new database.

3 Database and Indexing Issues

Because we were using the same databases for the Heterogeneous track as in 2007 we refer the reader to our INEX 2006 paper where the indexing issues

and approaches were discussed. We focus in this section on the Books database and the issues with it (as well as how the MARC data included with the Books database was converted and made searchable as XML, and how the EVIs are created).

All of the submitted runs for this year’s Book track and Heterogeneous track used the Cheshire II system for indexing and retrieval. For the Book Track The “Classification Clustering” feature of the system was used to generate the EVIs used in query expansion. The original approach for Classification Clustering was in searching was described in [13] and [14]. Although the method has experienced considerable changes in implementation, the basic approach is still the same: topic-rich elements extracted from individual records in the database (such as titles, classification codes, or subject headings) are merged based on a normalized version of a particular organizing element (usually the classification or subject headings), and each such *classification cluster* is treated as a single “document” containing the combined topic-rich elements of all the individual documents that have the same values of the organizing element. The EVI creation and search approach taken for this research is described in detail in the following section.

3.1 Book Track: MARC and Entry Vocabulary Indexes

The earliest versions of Entry Vocabulary Indexes were developed to facilitate automatic classification of MARC library catalog records, and first used in searching in [14]. Given the MARC data included with almost all of the documents for the Book track it seemed an interesting experiment to test how well EVIs “library catalog” searching would work with the books collection in addition to the full XML search approaches. It also seemed interesting to combine these two approaches.

The early work used a simple frequency-based probabilistic model in searching, but a primary feature was that the “Classification clusters”, were treated as documents and the terms associated with top-ranked clusters were combined with the original query, in a method similar to “blind feedback”, to provide an enhanced second stage of search.

Our later work with EVIs used a maximum likelihood weighting for each term (word or phrase) in each classification. This was the approach described in [9] and used for Cross-language Domain-Specific retrieval for CLEF 2005. One limitation of that approach is that the EVI can produce maximum likelihood estimates for only a single term at a time, and alternative approaches needed to be explored for combining terms (see [20] for the various approaches).

Although the method has experienced considerable changes in implementation, the basic approach for “Classification Clustering” in Cheshire II is still the same. Various topic-rich elements are extracted from individual records in the database (such as titles, classification codes, or subject headings) and are merged into single records based on a normalized version of a particular organizing element (usually the classification or subject headings, e.g., one record is created for each unique classification or subject heading). Each of these *classification clusters* is treated as a single “document” containing the combined topic-rich

Name	Description	Contents	Vector?
names	All Personal and Corporate names	//FLD[1670]00, //FLD[1678]10, //FLD[1670]11	No
pauthor	Personal Author Names	//FLD[170]00	No
title	Book Titles	//FLD130, //FLD245, //FLD240, //FLD730, //FLD740, //FLD440, //FLD490, //FLD830	No
subject	All Subject Headings	//FLD6..	No
topic	Topical Elements	//FLD6.., //FLD245, //FLD240, //FLD4.., //FLD8.., //FLD130, //FLD730, //FLD740, //FLD500, //FLD501, //FLD502 //FLD505, //FLD520, //FLD590	Yes
lclass	Library of Congress Classification	//FLD050, //FLD950	No
doctype	Material Type Code	//USMARC@MATERIAL	No
localnum	ID Number	//FLD001	No
ISBN	ISBN	//FLD020	No
publisher	Publisher	//FLD260/b	No
place	Place of Publication	//FLD260/a	No
date	Date of Publication	//FLD008	No
lang	Language of Publication	//FLD008	No

Table 1. MARC Indexes for INEX Book Track 2007

elements of all the individual documents that have the same values of the organizing element. In place of the simpler probabilistic model used in the early research, we use the same logistic regression based algorithm that is used for text retrieval. In effect, we just search the “Classification Clusters” as if they were documents using the TREC2 algorithm with blind feedback described above, then take some number of the top-ranked terms and use those to expand the query for submission to the normal document collection. Alternatively, because of the one-to-one match of books and MARC records in this collection, MARC searches or classification cluster two-stage searches can be considered a form of document search.

Two separate EVIs were built for the MARC data extracted from the Books database. The first uses the library classification code (MARC field 050) as the organizing basis and takes the searchable terms from all titles and subject headings in the MARC record (E.g., MARC fields 245, 440, 490, 830, 740, 600, 610, 620, 630, 640, 650). The second uses the topical subject fields (MARC field 650) with the same searchable fields.

The indexes used in the MARC data are shown in Table 1. Note that the tags represented in the “Contents” column of the table are from Cheshire’s MARC to XML conversion, and are represented as regular expressions (i.e., square brackets indicate a choice of a single character).

3.2 Indexing the Books XML Database

All indexing in the Cheshire II system is controlled by an XML/SGML Configuration file which describes the database to be created. This configuration file is subsequently used in search processing to control the mapping of search command index names (or Z39.50 numeric attributes representing particular types of bibliographic data) to the physical index files used and also to associated component indexes with particular components and documents.

Because the structure of the Books database was derived from the OCR of the original paper books, it is primarily focused on the page organization and layout and not on the more common structuring elements such as “chapters” or “sections”. Because this emphasis on page layout goes all the way down to the individual word and its position on the page, there is a very large amount of markup for page with content. The entire document in XML form is typically multiple megabytes in size. Given the nature of the XML/SGML parser used in the Cheshire II system, each document was taking several minutes for parsing and indexing due to the large internal representation of the parsed document taking up all available RAM space and a large portion of swap space on the available indexing machine. After indexing was run for a full 24 hours, and only 54 items had been indexes, a different approach was taken. Instead of parsing the entire document, we treated each page representation (tagged as “object” in the XML markup) as if it were a separate document. Thus the 42,000 full books were treated as a collection of ??????? page-sized documents.

As noted above the Cheshire system permits parts of the document subtree to be treated as separate documents with their own separate indexes. Thus, paragraph-level components were extracted from the page-sized documents. Because unique object (page) level indentifiers are included in each object, and these indentifiers are simple extensions of the document (book) level identifier, we were able to use the page-level identifier to determine where in a given book-level document a particular page or paragraph occurs, and generate an appropriate XPath for it.

Indexes were created to allow searching of full page (object) contents, and component indexes for the full content of each of individual paragraphs on a page. Because of the physical layout based structure used by the Books collection, paragraphs split across pages are marked up (and therefore indexed) as two paragraphs. Indexes were also created to permit searching by object id, allowing search for specific individual pages, or ranges of pages.

4 INEX 2007 Book Track and Heterogeneous Runs

4.1 Book Track Runs

Berkeley is planning to submit the maximum number of runs possible for the Book track, using various combinations of direct object-level access and access via MARC data for the documents. Within the latter we are using EVI search as well as direct MARC search and then using combining those document-level metadata searches with object-level searches using fusion operators.

4.2 Heterogeneous Runs

Three runs were submitted for the Resource Selection task, and 2 for the Content-Only task. The Resource selection runs used the TREC2, TREC3, and CORI algorithms, respectively, with no blind feedback. The two Content-Only runs used the TREC2 and TREC3 algorithms, also with no blind feedback.

Since Berkeley was the only group to submit Het track runs, it was decided not to go to the effort of evaluation with such a limited pool, so we do not have any figures on the actual or relative performance of these different techniques for the Heterogeneous track.

5 Conclusions and Future Directions

Our participation in INEX this year was very limited due to family issues which prevented us from completing the Adhoc submissions, with the later deadlines for the Book Track and Heterogeneous track we were able to do considerable work. Since heterogeneous was, in effect, cancelled, and the book track is having problems getting the submission system up, we have no evaluation or results to present this year. We hope that our approaches will prove interesting even without results.

References

1. J. Callan. Distributed information retrieval. In W. B. Croft, editor, *Advances in Information Retrieval: Recent research from the Center for Intelligent Information Retrieval*, chapter 5, pages 127–150. Kluwer, Boston, 2000.
2. J. P. Callan, Z. Lu, and W. B. Croft. Searching Distributed Collections with Inference Networks . In E. A. Fox, P. Ingwersen, and R. Fidel, editors, *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 21–28, Seattle, Washington, 1995. ACM Press.
3. A. Chen. Multilingual information retrieval using english and chinese queries. In C. Peters, M. Braschler, J. Gonzalo, and M. Kluck, editors, *Evaluation of Cross-Language Information Retrieval Systems: Second Workshop of the Cross-Language Evaluation Forum, CLEF-2001, Darmstadt, Germany, September 2001*, pages 44–58. Springer Computer Science Series LNCS 2406, 2002.
4. A. Chen. *Cross-Language Retrieval Experiments at CLEF 2002*, pages 28–48. Springer (LNCS #2785), 2003.
5. A. Chen and F. C. Gey. Multilingual information retrieval using machine translation, relevance feedback and decompounding. *Information Retrieval*, 7:149–182, 2004.
6. W. S. Cooper, A. Chen, and F. C. Gey. Full Text Retrieval based on Probabilistic Equations with Coefficients fitted by Logistic Regression. In *Text REtrieval Conference (TREC-2)*, pages 57–66, 1994.
7. W. S. Cooper, F. C. Gey, and A. Chen. Full text retrieval based on a probabilistic equation with coefficients fitted by logistic regression. In D. K. Harman, editor,

- The Second Text Retrieval Conference (TREC-2) (NIST Special Publication 500-215)*, pages 57–66, Gaithersburg, MD, 1994. National Institute of Standards and Technology.
8. W. S. Cooper, F. C. Gey, and D. P. Dabney. Probabilistic retrieval based on staged logistic regression. In *15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Copenhagen, Denmark, June 21-24*, pages 198–210, New York, 1992. ACM.
 9. F. Gey, M. Buckland, A. Chen, and R. Larson. Entry vocabulary – a technology to enhance digital search. In *Proceedings of HLT2001, First International Conference on Human Language Technology, San Diego*, pages 91–95, March 2001.
 10. L. Gravano and H. García-Molina. Generalizing GLOSS to vector-space databases and broker hierarchies. In *International Conference on Very Large Databases, VLDB*, pages 78–89, 1995.
 11. L. Gravano, H. García-Molina, and A. Tomasic. GLOSS: text-source discovery over the Internet. *ACM Transactions on Database Systems*, 24(2):229–264, 1999.
 12. D. Harman. Relevance feedback and other query modification techniques. In W. Frakes and R. Baeza-Yates, editors, *Information Retrieval: Data Structures & Algorithms*, pages 241–263. Prentice Hall, 1992.
 13. R. R. Larson. Classification clustering, probabilistic information retrieval, and the online catalog. *Library Quarterly*, 61(2):133–173, 1991.
 14. R. R. Larson. Evaluation of advanced retrieval techniques in an experimental online catalog. *Journal of the American Society for Information Science*, 43(1):34–53, 1992.
 15. R. R. Larson. A logistic regression approach to distributed IR. In *SIGIR 2002: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 11-15, 2002, Tampere, Finland*, pages 399–400. ACM, 2002.
 16. R. R. Larson. Distributed IR for digital libraries. In *Research and Advanced Technology for Digital Libraries (ECDL 2003)*, pages 487–498. Springer (LNCS #2769), 2003.
 17. R. R. Larson. A fusion approach to XML structured document retrieval. *Information Retrieval*, 8:601–629, 2005.
 18. R. R. Larson. Probabilistic retrieval, component fusion and blind feedback for XML retrieval. In *INEX 2005*, pages 225–239. Springer (Lecture Notes in Computer Science, LNCS 3977), 2006.
 19. Y. Mass and M. Mandelbrod. Component ranking and automatic query refinement for xml retrieval. In *Advances in XML Information Retrieval: Third International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX2004*, pages 73–84. Springer (LNCS #3493), 2005.
 20. V. Petras, F. Gey, and R. Larson. Domain-specific CLIR of english, german and russian using fusion and subject metadata for query expansion. In *Cross-Language Evaluation Forum: CLEF 2005*, pages 226–237. Springer (Lecture Notes in Computer Science LNCS 4022), 2006.
 21. S. E. Robertson and K. S. Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, pages 129–146, May–June 1976.
 22. E. Voorhees and D. Harman, editors. *The Seventh Text Retrieval Conference (TREC-7)*. NIST, 1998.
 23. E. Voorhees and D. Harman, editors. *The Eighth Text Retrieval Conference (TREC-8)*. NIST, 1999.

24. J. Xu and J. Callan. Effective retrieval with distributed collections. In *Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 112–120, 1998.

CMIC at INEX 2007: Book Search Track

Walid Magdy and Kareem Darwish
Cairo Microsoft Innovation Center (CMIC)
{i-wmagdy, kareemd}@microsoft.com

1. Introduction:

This paper describes the runs we performed at the Cairo Microsoft Innovation Center (CMIC) for the 2007 INEX Book Search track. We participated in the book retrieval and the page in context retrieval tasks. We explored generalized retrieval approaches and specialized book-specific approaches that made use of the indices, tables of content, and other fields in the books.

Section 2 provides an overview on the data collection and the IR engine that we used; Section 3 provides a description of the book retrieval task runs; and Section 4 describes our approach for the page in context task.

2. Data Collection and Used Search Toolkit

The collection, provided by Microsoft Live Book Search and the Internet Archive, consisted of 42,049 digitized out-of-copyright books. The actual number of books we used was 41,825, where 224 books were missed due to extraction errors or empty content books. The OCR content of the books was stored in djvu.xml format, which is described thoroughly in the track guidelines.

For all submitted runs, Indri search toolkit was used for indexing and searching the collection of books. Indri was used with stop-word removal, but with no stemming, and several runs were performed twice while enabling or disabling blind relevance feedback. Indri combines inference network model with language modeling (Metzler and Croft, 2004).

3. Book Retrieval Task

This task aimed to help users identify books of interest based on a stated information need. There were 250 queries about general subject: typically consisting of 1 word and commonly containing named entities. Two sample queries are: “Botany” and “Rigveda.” Pairs of runs were required. For each pair, one run would apply generic IR techniques and the other would use additional book-specific features such as Table Of Content (TOC) pages, index pages, and page headers. Each run was expected to return a ranked list of 1,000 books. We performed three pairs of runs.

The 3 runs using none book-specific features were as follows:

1. Each document was made up of the entire contents of each book. The books were subsequently indexed and searched using the provided queries.
2. The run was identical to the first run, except that blind relevance feedback was used, where 30 terms were extracted from the top 25 retrieved books to expand the original query.

- Each document was a single page in each book. All the documents were subsequently indexed and searched using the provided queries. Using the top 5,000 results for a given query, the score of the book was the sum scores of the individual scores within the ranked list as follows:

$$Score_{book_i} = \sum_{\forall page_j \in book_i} 10^{score_{page_j}}$$

The reason for using 10 to the power of the score is that Indri scores are log values. Given the scores of the books, a new ranked list was produced. In essence, the books with the most pages mentioning a specific topic would typically ranked first.

The runs using book-specific features were as follows:

- Each document was composed of all the headers in a book. The headers were assumed to be the first line in each page not composed entirely of digits. The documents were indexed and searched using the provided queries. The advantage of using headers is that they generally reflect the main topics in books and the titles of longer chapters are repeated more often, hence giving different weights to different titles.
- Each document was composed of the TOC and index pages in a book. We deemed a page to be a TOC or index page if any of the following conditions are met:
 - Presence of the key phrase “Table of Contents.”
 - Presence of ordinary key words such as contents, page, or index, with moderate number of lines that end with digits.
 - Absence of keywords indicating a TOC or index page, but the presence of a large number of lines that end with digits.
 - Presence of keywords such as contents, page, or index in a page that was immediately preceded by a page that was deemed as a TOC or index page.

In case we were not able to extract TOC and index pages, we used the first 3,000 characters from the OCR output and last 10 pages of a book instead, as they are likely to contain TOC and index pages or the pages with the introduction and/or preface. The rationale for using the first 3,000 characters instead of a fixed number of pages is that we found that many books typically contained many empty pages in the beginning.

- Each document was identical to documents in the second run except that we used blind relevance feedback, where 20 terms were extracted from the top 25 retrieved documents to expand the queries.

4. Book page in context retrieval task

In this task, each system was expected to return a ranked list of 1,000 books and for each book, a ranked list of relevant pages to a user’s information need. For the 30 provided queries, we performed 7 runs, 6 of which were automatic and 1 was manual. All the runs were identical to the run number “3” in the book search task in which no book-specific features were used to generate the ranked list of books. For each, we generated a ranked list of pages based on the score of each page. The differences between the 7 runs were all due to the way the queries were formulated. The formulations used the title, description, and narrative fields as follows:

1. Title only
2. Title only with blind relevance feedback
3. Title and description
4. Title and description with blind relevance feedback
5. Title, description, and narrative
6. Title, description, and narrative with blind relevance feedback
7. Manually reformulated queries that were done with consultation of Wikipedia on the topics.

For runs with blind relevance feedback, the queries were expanded with 20 terms extracted from the top 25 retrieved documents.

5. Conclusion

In our submitted runs we experimented with none book-specific as well as book-specific features for the book search and page in context tasks. We can't draw any conclusions at this time as we don't have the relevance judgments for the tasks.

6. References

Metzler, D. and Croft, W.B. Combining the Language Model and Inference Network Approaches to Retrieval. *Information Processing and Management Special Issue on Bayesian Networks and Information Retrieval*, 40(5), 735-750, (2004).

XML Document Classification using Extended VSM

Jianwu Yang¹, Fudong Zhang¹

¹Institute of Computer Sci. & Tech. , Peking University, Beijing 100871, China
{yangjianwu, zhangfudong}@icst.pku.edu.cn

Abstract. Structured link vector model (SLVM) is a representation recently proposed for modeling XML documents, which was extended from the conventional vector space model (VSM) by incorporating document structures. In this work, we describe a classification method for XML documents based on SLVM and Support Vector Machine (SVM). The experimental results on INEX 2007's data set show that it outperforms any other competitor's approach at an international competition on XML document classification.

1. Introduction

XML is the W3C recommended markup language for semi-structured data. Its structural flexibility makes it an attractive choice for representing data in application domains¹, including news items (NewsML), mathematical formulae (MathML), vector graphics (SVG), as well as some proprietary designs used by specific enterprises and institutions. Among the different possible XML-based documents, the focus of this paper is on those with elements containing *textual* descriptions.

The recent proliferation of XML adoption in large digital archives [1,2] calls for new document analysis techniques to support effective semi-structured document management, sometimes down to the level of the composing elements. Even though the tasks of interest are still clustering, classification and retrieval, conventional document analysis tools developed for unstructured documents [3] fail to take the full advantage of the structural properties of XML documents.

To contrast with ordinary unstructured documents, XML documents represent their syntactic structure via (1) the use of XML elements, each marked by a user-specified tag, and (2) the associated schema specified in either DTD or XML Schema format. In addition, XML documents can be cross-linked by adding IDREF attributes to their elements to indicate the linkage. Thus, techniques designed for XML document analysis normally take into account the information embedded in both the element tags as well as their associated contents for better performance. For example, the structural similarity between a pair of IDREF-free XML documents can be

¹ Hundreds of different XML applications can be found at <http://xml.coverpages.org/xmlApplications.html>.

defined as some edit distance between unordered labeled trees², i.e., to compute the minimum number of operators needed to edit the tree from one from to another. In the literature, different tree edit distances have been proposed for measuring XML document dissimilarity [4,5], which are equivalent in principle except for the edit operators allowed and whether repetitive and optional XML elements were considered. However, computing the edit distance between unordered labeled trees is NP-complete [6] and yet the distance is in general not optimal in any sense. This is undesirable for large-scale applications. An alternative is to measure the depth difference with reference to the root element for defining structural dissimilarity between a pair of XML elements [7,8]. The depth differences can then be aggregated for estimating the overall document dissimilarity. While the associated computational cost is low, the accuracy is limited. Other than trees, XML documents have also been represented as time series [9], with each occurrence of a tag corresponding to an impulse. Document similarity was then computed by comparing the corresponding Fourier coefficients of the documents. This approach does not take into account the order in which the elements appear and is adequate only when the XML documents are drastically different from each other, i.e., they have very few tags in common. In [10], WordNet -- an ontology of general concepts [11] has been used to measure the semantic similarity of the elements' names and their values. However, in many applications, domain-specific knowledge is needed instead, which is sometimes not easy to be captured.

Table 1. A comparison of related works in the literature with XML similarity considered.

References	Structural similarity	Semantic similarity	Remarks
[12]	Yes	No	Extending VSM
[7,8]	Yes	No	Tree edit distance
[9]	Yes	No	Fourier coefficients
[10]	No	Yes	Ontology-based
[13]	Yes	No	Tree-based generative language model
[14, 15]	Yes	No	Extending VSM
[16]	No	Yes	Extending query relaxation
[17]	Yes	No	Bayesian network model
[18]	Yes	No	A mixture Language model
[19]	Yes	Yes	Queries in natural language

² A labeled unordered tree is a tree structure with all its nodes labeled but the order of the children of any parent node not maintained. The use of unordered trees for representing XML documents is justified by the fact that two documents with identical contents but different orderings of their sibling elements should be considered as semantically equivalent.

Structured Link Vector Model (SLVM), which forms the basis of this paper, was originally proposed in [12] for representing XML documents. It was extended from the conventional vector space model (VSM) by incorporating document structures (represented as term-by-element matrices), referencing links (extracted based on IDREF attributes), as well as element similarity (represented as an element similarity matrix).

Table 1 shows a more complete list of related works and their comparison in terms of representation and the nature of similarity considered.

2. Structured Link Vector Model (SLVM)

2.1. Basic representation

Vector Space Model (VSM) [20] has long been used to represent unstructured documents as document feature vectors which contain term occurrence statistics. This bag of terms approach assumes that the term occurrences are *independent* of each other.

Definition 2.1 Assume that there are n distinct terms in a given set of documents D . Let doc_x denote the x^{th} document and d_x denote the **document feature vector** such that

$$d_x = [d_{x(1)}, d_{x(2)}, \dots, d_{x(n)}]^T$$

$$d_{x(i)} = TF(w_i, doc_x) IDF(w_i)$$

where $TF(w_i, doc_x)$ is the frequency of the term w_i in doc_x , $IDF(w_i) = \log(|D|/DF(w_i))$ is the inverse document frequency of w_i for discounting the importance of the frequently appearing terms, $|D|$ is the total number of the documents, and $DF(w_i)$ is the number of documents containing the term w_i .

Applying VSM directly to represent XML documents is not desirable as the document syntactic structure tagged by their XML elements will be ignored. For example, VSM considers two documents with an identical term appearing in, say, their “title” fields to be equivalent to the case with the term appearing in the “title” field of one document and in the “author” field of another. As the “author” field is semantically unrelated to the “title” field, the latter case should be considered as a piece of less supportive evidence for the documents to be similar when compared with the former case. Using merely VSM, these two cases cannot be differentiated.

Structured Link Vector Model (SLVM), proposed in [12], can be considered as an extended version of vector space model for representing XML documents. Intuitively

speaking, SLVM represents an XML document as an array of VSMS, each being specific to an XML element (specified by the <element> tag in DTD).³

Definition 2.2 SLVM represents an XML document doc_x using a **document feature matrix** $\Delta_x \in R^{n \times m}$, given as

$$\Delta_x = [\Delta_{x(1)}, \Delta_{x(2)}, \dots, \Delta_{x(m)}]$$

where m is the number of distinct XML elements, $\Delta_{x(i)} \in R^n$ is the TFIDF feature vector representing the i^{th} XML element (e_i), given as $\Delta_{x(i,j)} = TF(w_j, doc_x.e_i) \cdot IDF(w_j)$ for all $j=1$ to n , and $TF(w_j, doc_x.e_j)$ is the frequency of the term w_i in the element e_j of doc_x .

Definition 2.3 The **normalized document feature matrix** is defined as

$$\tilde{\Delta}_{x(i,j)} = \Delta_{x(i,j)} / \sum_j \Delta_{x(i,j)}$$

where the factor caused by the varying size of the element content is discounted via normalization.

Example 2.1 Figure 1 shows a simple XML document. Its corresponding document feature vector d_x , document feature matrix Δ_x , and normalized document feature matrix $\tilde{\Delta}_x$ are shown in Figure 2-4 respectively. Here, we assume all the terms share the same *IDF* value equal to one.

The form of SLVM studied in this paper is only a simplified one where only the leaf-node elements in the DTD are incorporated without considering their positions in the document DOM tree and their consecutive occurrence patterns. In addition, the interconnectivity between the documents based on IDREF is also not considered. One obvious advantage is that this simplification can make the subsequent similarity learning much more tractable. Also, this kind of unigram-like approach makes it applicable to most of the unseen XML documents as long as there are no newly encountered terms. If the consecutive occurrence patterns of the elements are to be taken into consideration, the most extreme case is to have each possible path of the DOM tree corresponds to one column in Figure 3. This however will increase the computational complexity exponentially. Also, the generalization capability will be poor (e.g., a book with three authors cannot be modeled if a maximum of two authors are assumed in the SLVM's document feature matrix).

³ In the current version of SLVM, only the elements corresponding to the leaf nodes of the XML DOM tree are modeled.

```

<article>
  <title>Ontology Enabled Web Search</name>
  <author>John</author>
  <conference>Web Intelligence</conference>
</article>

```

Fig. 1. An XML document.

$$d_x = \begin{matrix} & \text{thisDocument} \\ & \begin{bmatrix} 1 \\ 1 \\ 2 \\ 1 \\ 1 \\ 1 \end{bmatrix} \\ & \begin{matrix} \text{Ontology} \\ \text{Enabled} \\ \text{Web} \\ \text{Search} \\ \text{John} \\ \text{Intelligence} \end{matrix} \end{matrix}$$

Fig. 2. The document feature vector for the example shown in Figure 1.

$$\Delta_x = \begin{matrix} & \text{title} & \text{author} & \text{confe} \\ & & & \text{rence} \\ & \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ & \begin{matrix} \text{Ontology} \\ \text{Enabled} \\ \text{Web} \\ \text{Search} \\ \text{John} \\ \text{Intelligence} \end{matrix} \end{matrix}$$

Fig. 3. The document feature matrix for the example in Figure 1.

$$\tilde{\Delta}_x = \begin{array}{ccc|l} \text{title} & \text{author} & \text{confe} & \\ & & \text{rence} & \\ \hline 0.5 & 0 & 0 & \text{Ontology} \\ 0.5 & 0 & 0 & \text{Enabled} \\ 0.5 & 0 & \sqrt{2}/2 & \text{Web} \\ 0.5 & 0 & 0 & \text{Search} \\ 0 & 1 & 0 & \text{John} \\ 0 & 0 & \sqrt{2}/2 & \text{Intelligence} \end{array}$$

Fig. 4. The normalized document feature matrix for the example in Figure 1.

2.2. Similarity measures

Using VSM, similarity between two documents doc_x and doc_y is typically computed as the cosine value between their corresponding document feature vectors, given as

$$sim(doc_x, doc_y) = \frac{d_x d_y}{\|d_x\| \|d_y\|} = \tilde{d}_x \tilde{d}_y^T = \sum_{i=1}^k \tilde{d}_{x(i)} \tilde{d}_{y(i)} \quad (1)$$

where n is the total number of terms and $\tilde{d}_x = d_x / \|d_x\|$ denotes normalized d_x . So, the similarity measure can also be interpreted as the inner product of the normalized document feature vectors.

For SLVM, with the objective to model semantic relationships between XML elements, the corresponding document similarity can be defined with an element similarity matrix introduced.

Definition 2.4 The *SLVM-based document similarity* between two XML documents doc_x and doc_y is defined as

$$sim(doc_x, doc_y) = \sum_{i=1}^n \Delta_{x(i)}^n T M_e \Delta_{y(i)}^n \quad (2)$$

where M_e is a matrix of dimension $m \times m$ and named as the *element similarity matrix*.

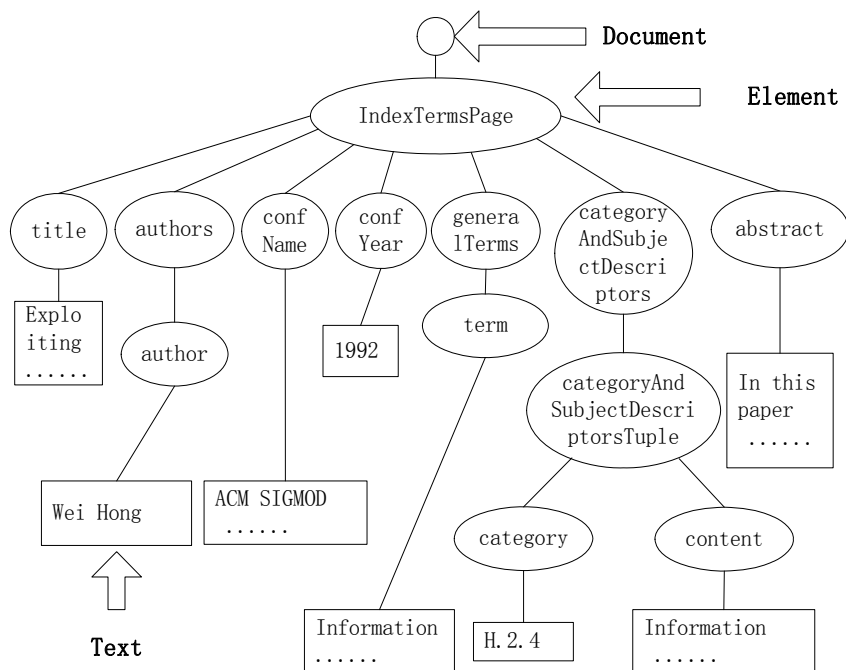


Fig. 5. The DOM tree of the ACMSIGMOD Record dataset.

The matrix M_e in Eq. (2) captures both the similarity between a pair of XML elements as well as the contribution of the pair to the overall document similarity (*i.e.*, the diagonal elements of M_e are not necessarily equal to one). An entry in M_e being small means that the two corresponding XML elements should be unrelated and same words appearing in the two elements of two different documents will not contribute much to the overall similarity of them. If M_e is diagonal, this implies that all the XML elements are not correlated at all with each other, which obviously is not the optimal choice.

The structural similarity between a pair of XML documents can thus be computed based on different tree edit distances [1,2] which differ from each others in terms of the set of allowed edit operators and their support for repetitive and optional XML elements. It has been proved in [3] that computing the edit distance for unordered labeled trees is NP-complete, and yet the distance is not optimal in any sense related to the elements' semantics. In [12], the element similarity was pre-set to be related to the path difference between two elements as well as their depth difference with reference to the root derived from the document schema. To obtain an optimal M_e for a specific type of XML data, we adopt the machine learning approach in [21].

3. SVM for XML Documents Classification

SVM was introduced by Vapnik in 1995 for solving two-class pattern recognition problems using the Structural Risk Minimization principle [22]. Given a training set containing two kinds of data (one for positive examples, the other for negative examples), which is linearly separable in vector space, this method finds the decision hyper-plane that best separated positive and negative data points in the training set. The problem searching the best decision hyper-plane can be solved using quadratic programming techniques [23]. SVM can also extend its applicability to linearly nonseparable data sets by either adopting soft margin hyper-planes, or by mapping the original data vectors into a higher dimensional space in which the data points are linearly separable [22, 23, 24].

Joachims [25] first applied SVM to text categorization, and compared its performance with other classification methods using the Reuters-21578 corpus. His results show that SVM outperformed all the other methods tested in his experiments. Subsequently, Dumais [26], Yang [27], Cooley [28], and Bekkerman [29] also explored how to solve text categorization with SVM respectively. Although based on different document collections, their experiments confirmed Joachim's conclusion that SVM is the best method for classifying text documents.

SVM success in practice is drawn by its solid mathematical foundations which convey the following two salient properties:

- **Margin maximization:** The classification boundary functions of SVM maximize the margin, which in machine learning theory, corresponds to maximizing the *generalization* performance given a set of training data.
- **Nonlinear transformation of the feature space using the kernel trick:** SVM handle a nonlinear classification efficiently using the kernel trick which implicitly transforms the input space into another high dimensional feature space.

In SVM, the problem of computing a margin maximized boundary function is specified by the following quadratic programming (QP) problem:

$$\text{minimize: } W(\alpha) = -\sum_{i=1}^n \alpha_i + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$

$$\text{subject to: } \sum_{i=1}^n y_i \alpha_i = 0; \quad \forall i: 0 \leq \alpha_i \leq C$$

where n The number of training data, α is a vector of n variables, where each component α_i corresponds to a training data (x_i, y_i) . C is the soft margin parameter controlling the influence of the outliers (or noise) in training data.

The classification function is:

$$f(x) = \text{sgn}\left\{\sum_{i=1}^n \alpha_i y_i k(x_i, x) + b\right\}$$

where b is a threshold for categorization.

The kernel $k(x_i, x_j)$ for linear boundary function is $x_i \cdot x_j$, a scalar product of two data points. The nonlinear transformation of the feature space is performed by replacing $k(x_i, x_j)$ with an advanced kernel, such as polynomial kernel $(x^T x_i + 1)^p$ or RBF kernel $\exp(-\frac{1}{2\delta^2} \|x - x_i\|^2)$.

According to definition 2.4, the kernel $k(x_i, x_j)$ for XML documents classification based on SLVM is:

$$k(x_i, x_j) = \text{sim}(doc_x, doc_y) = \sum_{i=1}^n \Delta_{x(i)}^T M_e \Delta_{y(i)}$$

4. Experiments

In the experiments, all the algorithms were implemented by us in C++, except the SVM algorithm in SVMTorch [30]. All experiments were run on a PC with a 3.0 GHz Intel CPU and 512M RAM.

4.1. Initial experiments

Test data were not available until shortly before the conclusion of the XML classification competition. As a consequence, the initial approaches addressed in this section evaluate performances on the training data. Performance evaluations on test data will be given in Section 4.2. Thus, initially we resorted to splitting the available data (the original training data set) into two sub-sets:

- **Training Set:** Part of the original training data set is selected to be used for training purposes.
- **Test Set:** The remaining is used as test data.

The number of different elements is one of key factors in efficiency, but the most of elements' occurrence times in the data set are less than 10. Thus we eliminate those elements whose occurrence times in the data set are less than 10 in the experiment, and the remaining elements are about 15% of all elements. The experiments' result show that the time cost is reduced evidently and the effect is not nearly influenced by the elimination.

As a basic format, the matrix M_e in Eq. (2) is set as diagonal, this means that all the XML elements are not correlated at all with each other, which obviously is not the optimal choice. In [12], the element similarity (the entry of the matrix M_e) was pre-set to be related to the path difference between two elements as well as their depth

difference with reference to the root derived from the document schema. But the experiment result shows it is useless in the data set. Also, the experiment result shows it is useless that the element similarity estimated using the edit distance [4,5] in the data set.

4.2 Advanced experiments

According to the experiment results, those elements whose occurrence times are more than 10 in the train data set are considered as available, and the matrix M_e in Eq. (2) is set as diagonal in the advance experiment.

Table 2. The Results for SVM Classification Based on SLVM

Group	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	Size	Recall
1	560	3	1	3	9	3	1	5	13	1	14	0	0	40	0	4	1	0	0	2	0	660	0.8485
2	0	945	27	10	34	1	0	14	1	0	5	0	0	67	1	14	11	0	0	0	0	1130	0.8363
3	2	6	546	0	1	0	0	1	0	0	5	1	0	42	0	3	2	0	0	7	1	617	0.8849
4	10	21	2	1087	47	2	7	17	3	0	24	8	0	253	0	55	9	2	0	38	3	1588	0.6845
5	5	22	3	25	7655	0	1	41	45	0	51	3	16	229	11	243	41	2	4	21	0	8418	0.9094
6	0	0	0	1	7	142	0	0	0	0	0	0	0	39	0	10	1	1	0	0	0	201	0.7065
7	0	0	0	0	2	0	421	0	0	0	0	0	0	60	0	1	0	0	0	0	0	484	0.8698
8	1	9	1	5	40	0	0	1984	52	2	10	0	0	76	2	34	8	2	0	8	0	2234	0.8881
9	6	2	2	1	59	0	0	42	1099	0	7	7	2	53	2	16	10	2	0	18	1	1329	0.8269
10	0	0	0	0	0	0	0	0	0	577	2	0	0	12	0	0	0	0	0	0	0	591	0.9763
11	1	4	4	13	46	0	0	6	6	0	6962	2	0	179	6	24	8	1	0	5	0	7267	0.958
12	0	3	0	6	16	0	1	1	2	0	4	2134	1	75	0	3	3	0	0	65	0	2314	0.9222
13	0	1	0	0	26	0	0	0	0	0	2	0	242	27	0	5	1	0	0	0	0	304	0.7961
14	12	46	30	89	290	3	35	69	46	11	191	54	18	10767	26	248	50	3	0	113	4	12105	0.8895
15	0	0	0	1	19	0	0	4	0	0	4	7	0	62	172	2	0	0	1	3	0	275	0.6255
16	4	9	3	32	320	1	4	25	20	0	72	3	4	319	0	3010	24	3	2	28	1	3884	0.775
17	5	15	0	3	115	2	0	21	11	0	47	10	8	126	0	55	780	1	3	14	3	1219	0.6399
18	0	0	0	0	14	0	0	1	2	0	0	0	0	5	0	3	0	204	0	1	0	230	0.887
19	0	0	0	0	3	0	0	0	0	0	0	0	0	4	0	1	2	0	231	0	0	241	0.9585
20	2	3	9	24	103	0	1	14	53	1	20	89	5	173	5	49	7	0	0	2088	13	2659	0.7853
21	0	1	1	2	4	0	0	2	0	0	2	0	0	7	0	2	0	0	0	6	528	555	0.9514
Micro average recall = 0.8722; Macro average recall = 0.8390																							

The classification result based on SLVM is presented in Table 2, which is based on utilizing 100% of the original training set for training purposes, and the test dataset is provided by INEX 2007.

5. Conclusion and Future Works

In this paper, we studied in detail a proposed extension of VSM called SLVM for representing XML documents so that term semantics, element similarity, as well as elements' relative importance for a given set of documents can all be taken in account. And we applied SLVM and SVM to XML documents classification. The proposed method was demonstrated to outperform any other competitor's approach at an international competition on XML document classification.

For future work, we are interested to study how the similarity matrix obtained via the machine learning approach and support multiple word sense identification which serves as an important component for automatic ontology generation.

Acknowledgment

The work reported in this paper was supported by the National Natural Science Foundation of China Grant 60642001.

References

1. Early Americas Digital Archive, URL: <http://www.mith2.umd.edu:8080/eada/intro.jsp>
2. Contemporary Culture Virtual Archives in XML, URL: <http://www.covax.org/>
3. Berry, M.: Survey of Text Mining : Clustering, Classification, and Retrieval, Springer (2003).
4. Zhang, Z.P., Li R., Cao, S.L., and Zhu, Y.Y.: Similarity Metric for XML Documents. In: Proceedings of the 2003 Workshop on Knowledge and Experience Management (FGWM 2003), Karlsruhe (2003)
5. Nierman, A. and Jagadish, H.V.: Evaluating Structural Similarity in XML Documents. In: Proceedings of the Int. Workshop on the Web and Databases (WebDB), Madison, WI, (2002)
6. Zhang, K., Statman, R. and Shasha, D.: On the editing distance between unordered labeled trees. In: Information Processing Letters, 42(3):133--139 (1992)
7. Abolhassani, M., Fuhr, N. and Malik, S.: HyREX at INEX. In: Proceedings of the 2003 INEX Workshop, Schloss Dagstuhl, (2003)
8. Azevedo, M.I.M., Amorim, L.P. and Ziviani, N.: A Universal Model for XML Information Retrieval. In: Springer-Verlag Lecture Notes in Computer Science, vol. 3493, 2005, pp 311-321 (2004)

9. Flesca, S., Manco, G., Masciari, E., Pontieri, L., and Pugliese, A.: Detecting structural similarities between xml documents. In: Proceedings of the International Workshop on the Web and Databases (WebDB), Madison, WI. (2002)
10. Schenkel, R., Theobald, A., Weikum, G.: XXL @ INEX 2003. In: Proceedings of the 2003 INEX Workshop, Schloss Dagstuhl (2003)
11. Fellbaum, C.: WordNet: An Electronic Lexical Database, MIT Press (1998)
12. Yang, J., Chen, X.: A semi-structured document model for text mining. In: Journal of Computer Science and Technology, 17(5) pp 603-610 (2002)
13. Ogilvie, P., Callan, J.: Language Models and Structured Document Retrieval. In: Proceedings of the 2002 INEX Workshop, Schloss Dagstuhl (2002)
14. Mass, Y., Mandelbrod, M., Amitay, E., Carmel, D., Maarek, Y. and Soffer, A.: JuruXML – an XML Retrieval System at INEX'02. In: Proceedings of the 2002 INEX Workshop, Schloss Dagstuhl (2002)
15. Crouch, C., Mahajan, A. and Bellamkonda, A.: Flexible XML Retrieval Based on the Extended Vector Model. In: Proceedings of the 2004 INEX Workshop, Schloss Dagstuhl (2004)
16. Liu, S. and Chu, W.: Cooperative XML (CoXML) Query Answering at INEX 03. In: Proceedings of the 2003 INEX Workshop, Schloss Dagstuhl (2003)
17. Vittaut, J., Piwowarski, B., Gallinari, P.: An algebra for Structured Queries in Bayesian Networks. In: Proceedings of the 2004 INEX Workshop, Schloss Dagstuhl (2004)
18. Sigurbjornsson, B., Kamps, J., Rijke, M.: The University of Amsterdam at INEX 2004. In: Proceedings of the 2004 INEX Workshop, Schloss Dagstuhl (2004)
19. Woodley, A. and Geva, S.: NLPX at INEX 2004. In: Proceedings of the 2004 INEX Workshop, Schloss Dagstuhl (2004)
20. Salton G, and McGill MJ, *Introduction to Modern information Retrieval*. McGraw-Hill, 1983.
21. Yang, J.W., Cheung, W. K., Chen, X.O.: Integrating Element Kernel and Term Semantics for Similarity-Based XML Document Clustering, in Proceedings of 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05) , Compiègne, France (2005)
22. Vapnic, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)
23. Cortes, C. and Vapnik, V.: Support Vector networks. *Machine Learning*, 20: 273-297, (1995)
24. Osuna, R.F., and Girosi, F.: Support vector machines: Training and applications. In A.I. Memo. MIT A.I. Lab (1996)
25. Joachims, T.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In Proceedings of the 1998 European conference on Machine Learning (ECML), pages: 137-142 (1998)
26. Dumais, S., Platt, J., Heckerman, D., and Sahami, M.: Inductive learning algorithms and representations for text categorization. In Proceedings of the 1998 ACM CIKM International Conference on Information and Knowledge Management, pages 148-155 (1998)
27. Yang, Y., Liu, X.: A re-examination of text categorization methods. In 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99), pages 42-49 (1999)
28. Cooley, R.: Classification of News Stories Using Support Vector Machines. In Proceedings of the 16th International Joint Conference on Artificial Intelligence Text Mining Workshop (1999)
29. Bekkerman, R., Ran, E.Y., Tishby, N., and Winter, Y.: On feature distributional clustering for text categorization. In Proceedings of the 24th ACM SIGIR International Conference on Research and Development in Information Retrieval, pages: 146-153 (2001)
30. Collobert, R. and Bengio, S.: SVM-Torch: support vector machines for large-scale regression problems, *Journal of Machine Learning Research*, Vol.1, pp. 143-160 (2001)

A categorization approach for wikipedia collection based on Negative Category Information and Initial Descriptions

Meenakshi Sundaram Murugesan, K.Lakshmi, Dr.Saswati Mukherjee

Department of Computer Science and Engineering,
College of Engineering, Guindy,
Anna University, Chennai, India

msundar_26@yahoo.com lakshmi_tamil@hotmail.com msaswati@annauniv.edu

Abstract. The methods that we have applied for the classification task, in this year's XML mining track, is based on profile creation using the negative category document frequency (NCD) and the average document frequency of terms in initial descriptions in wikipedia articles. NCD reduces the weight of a term according to its presence and distribution over negative categories in the training-set. We experimented with two similarity measures namely, cosine and fractional similarity.

Keywords: negative categories, profile creation, fractional similarity, initial descriptions.

1 Introduction

The two tasks in INEX 2007's XML mining track are categorization and clustering. The corpus for this track is a subset of the wikipedia corpus with 96,611 documents that belong to 21 categories.

Authors of the paper [1] have demonstrated the effectiveness of using negative category document frequency (NCD) based profile creation for non-overlapping categories in an unstructured text corpus. Since XML mining track also uses non-overlapping categories, we have applied this method combined with a method based on initial descriptions in wikipedia articles.

Creating profiles using average term frequencies (TF) and average TF*IDF have the drawback of failing to consider the distribution of terms over positive and negative categories. Authors have shown that the presence of the term over large number of negative categories is undesirable. At the same time, when such negative documents are clustered in lesser number of negative categories, the power of contribution of the term to the positive category reduces considerably.

Negative Category Document frequency (NCD), which is shown below, reduces the weight of a term according to its distribution over negative categories. We set a threshold of top 3% for profile creation.

$ncd(t) = \log(1 + ncf/ndf) \quad \text{if } t \in \text{negative document}$ $= 1 \quad \text{if } t \notin \text{negative document}$
<p>where, ncf = no of negative categories the term appears, ndf = no of negative document the term appears.</p>

We observe that each wikipedia article starts with an initial description, which clearly states what the article is about and contains terms that can be used to distinguish the categories. We created another profile, based on average document frequency and inverse document frequency (average DF*IDF). The feature weighting scheme that we have used is TF*NCD. The test documents were represented as TF*IDF. We used two kinds of similarity measures viz. cosine similarity and fractional similarity to measure the similarity between the profiles the wikipedia articles in the test-set. Fractional similarity is calculated in such a way that, the higher the terms in the testing documents that are not in the profile, the lower the similarity score of the document with that category profile.

Fractional Similarity measure [4] between profile (CD) and the document (d) is calculated by using the following Equation.

$Fraction(CD, d) = \frac{\alpha}{\gamma} \quad \text{if } \{d\} - \{CD\} \neq \phi$ $= \alpha \quad \text{if } \{d\} - \{CD\} = \phi$
<p>Where</p> $\alpha = \sum_{k=1}^p w_k * v_k \quad \text{if } term_k \in CD \text{ and } d$ $\gamma = \sum_{k=1}^p v_k \quad \text{if } term_k \notin CD \text{ and } term_k \in d$
<p>w_k - weight of $term_k$ in CD v_k - weight of $term_k$ in document d p - number of terms in the CD and document</p>

2 Evaluation

The results that we have obtained during initial INEX evaluations are given in the following table.

	Micro average recall	Macro average recall
TF-NCD profiles, fractional similarity	0.774598902805093	0.714838847158184
TF-NCD profiles, cosine similarity	0.773170479246455	0.734183147200625
TF-NCD profiles, cosine similarity with 5% weight for initial descriptions	0.78008487734189	0.757502564330129

The best of the three results we have submitted combines the similarity scores of two methods, one based on the whole article and another based on the similarity in the initial description given in a wikipedia article. In this method we gave 95% weight for the similarity of the NCD based profile with the whole wikipedia content, and 5% weight for the similarity with the initial descriptions.

3 Conclusion

Since NCD based profile creation proved to perform well over non-overlapping categories, we have experimented with this method, coupled with the initial description based profile creation. We have planned to extend this method, by exploring the wikipedia specific structures such as section titles and links in a document.

References

1. K.Lakshmi, Saswati Mukherjee, Category Based Feature Weighting for Automatic Text Categorization. Accepted for publication in 3rd Indian International Conference on Artificial Intelligence (IICAI-07). IICAI-07, December 17-19 2007, Pune, India.
2. Ludovic Denoyer, Patrick Gallinari, Anne-Marie Vercoustre, Report on the XML Mining Track at INEX 2005 and INEX 2006, Categorization and Clustering of XML Documents. In proceedings of 5th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2006.
3. G. Salton and C. Buckley.: Term-Weighting Approaches In Automatic Text Retrieval, Inf. Process. Management, 24(5), (1988), 513—523.
4. Lakshmi K. and Mukherjee S (2006). An Improved Feature Selection using Maximized Signal to Noise Ratio Technique for TC. Information Technology: New Generations, 2006. ITNG 2006. Third International Conference on 10-12 April 2006.

Document Clustering using Incremental and Pairwise Approaches

Tien Tran and Richi Nayak

Faculty of Information Technology, Queensland University of Technology,
Brisbane, Australia

{t4.tran@qut.edu.au, r.nayak@qut.edu.au}

Abstract. This paper reports on the experiments and results using a clustering approach in INEX 2007 Document Mining Challenge. We used a clustering approach that combines the concepts of incremental clustering and clustering based on pairwise distance matrix to arrive to a final clustering solution. In this paper, we used an incremental method that first groups Wikipedia documents into a number of clusters progressively. This method proceeds by comparing the documents with existing clusters which are represented by the documents that first used to form the clusters. After the grouping of the documents performed by the incremental method, a pairwise distance matrix is then computed between the documents that represented the clusters. The graph clustering method is then applied on the pairwise distance matrix to merge the clusters together in order to reduce the number of clusters according to the user-defined number. This approach enables us to perform the clustering task on a large dataset by first reducing the dimension of the dataset using the incremental method and then clustering based on a pairwise distance matrix to preserve the effectiveness of the clustering solution.

Key words: INEX 2007, structure, content, XML, clustering

1 Introduction

Most electronic data on the Web, nowadays, is presented in the format of semi-structured data. Semi-structured data on the Web follows a flexible structure resulting in heterogeneous collections in subject content including XML, XHTML, HTML etc. as well as in representation. With the continuous growth of the semi-structured data, there is an inevitable need to efficiently manage these large volume of data.

Recognizing the importance of the management of these documents, researchers have proposed tasks such as categorization and clusterings of semi-structured documents. Amongst these semi-structured documents, XML documents have a great acceptance in many industries such as in e-business and, in recent years, XML has been widely used by researchers as data input.

Challenges such as INitiative for the Evaluation of XML Retrieval(INEX) [1] has been conducted for many years for researchers to test their XML based

research approaches on a particular dataset and to evaluate the performance of individual approach with the rest. The dataset used in INEX 2007 document mining challenge is the Wikipedia dataset containing 48035 XML documents.

We used a clustering approach that utilizes the idea of clustering based on pair-wise distance matrix and incremental clustering to participate in the INEX 2007 clustering task. Clustering approaches such as graph clustering method [2] uses a pairwise distance matrix for the clustering of documents. Computing a pairwise matrix, can be very expensive in terms of memory and computation time when dealing with a large dataset such as Wikipedia collection since it has to compute the distance between each pair of documents in the whole corpus. On the other hand, incremental clustering approaches [3, 4] performs clustering by measuring the distance between input documents and existing clusters. Incremental clustering can deal with a large dataset more efficiently than the pair-wise distance matrix. However these methods suffer with the problem of poor accuracy due to dependence on input ordering. The trade-off between clustering based on pair-wise distance matrix and incremental clustering is the effectiveness of the clustering solutions generated and the scalability of the clustering process.

Thus in our approach, we proposed using a clustering approach that first performs an incremental clustering method on the dataset to reduce the dimension of the dataset. Then a pair-wise distance matrix is computed between the clusters' representations in order to merge the clusters together which is performed by graph clustering method. By combining the incremental clustering with the clustering based on pair-wise distance matrix enables our proposed clustering approach is enabled to deal with the large datasets and produces a clustering solution with higher accuracy as possible.

This paper is structured as follows. The next section gives an overview of the proposed clustering approach. Section 3 explains the pre-processing stage by discussing on how the structure and the content of the dataset are extracted and represented. Section 4 explains the clustering process in details. Section 5 evaluates the proposed clustering approach with experiments and data analysis. The paper is then concluded in section 6.

2 Overview of the Proposed Clustering Approach

Fig. 1 illustrates an overview of the proposed clustering approach used in the INEX 2007 clustering challenge. The first stage of the clustering approach begins with the pre-processing of the input dataset. It is important to first pre-process the data to remove any irrelevant information that may degrade or contribute little to the clustering process. The output of the pre-processing stage is the features and their representations. In the case of XML document, it has two important features: the structure and content. Using either the structure and/or the content information that were extracted during the pre-processing stage, it is then used as an input to an incremental clustering method to perform the first run of the clustering process. After the incremental clustering stage, the documents is usually grouped into an undefined (more than the required)

number of clusters depending on the clustering threshold set by the user. The pairwise distance matrix is then computed between the clusters' representation which is represented by the documents that first used to form the clusters using *CPSim* to measure the document structure or cosine to measure the document content. This matrix is then fed into graph clustering method [2] to get the final clustering solution. The next few sections discuss the stages in the proposed clustering approach in details.

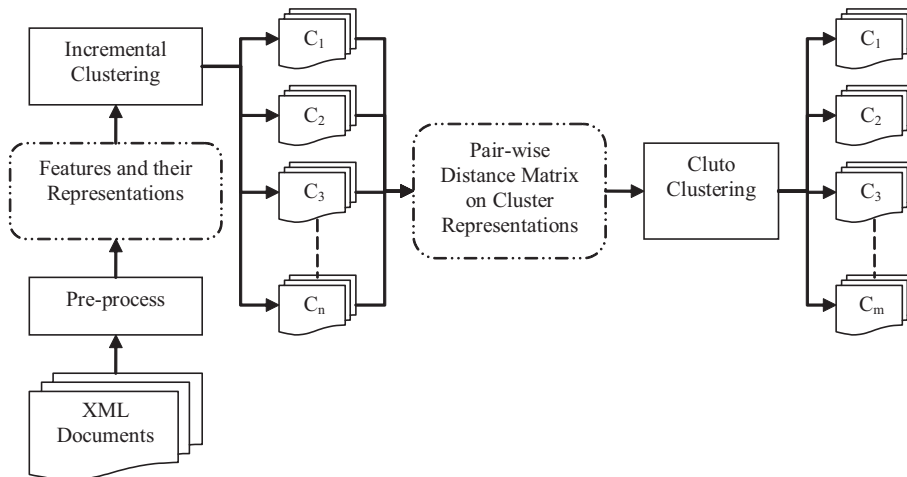


Fig. 1. Overview of the proposed clustering approach

3 Pre-processing: structure and content mining

3.1 Structure mining using hierarchical ordering of elements

The structure of a XML document is extracted and represented as a collection of paths. Each path contains the elements from root to the leaf node. Each path is considered as an individual item in the XML document structure therefore the order of the paths is ignored. Duplicated paths in a document structure are eliminated, thus, the final output result of the document structure is just a summary representation.

The structure of two XML documents is determined using a function called *CPSim* which is defined in Nayak and Tran [3]. *CPSim* is defined as:

$$CPSim(d_x, d_y) = \frac{\sum_{i=1}^{|P_x|} \max(\int_{j=1}^{|P_y|} Psim(p_i, p_j))}{\max(|P_x|, |P_y|)} \quad (1)$$

CPSim is the common path similarity coefficient between two XML documents, d_x and d_y ranges from 0 to 1 (1 is the highest). It computes the sum of the best path similar coefficient (*Psim*) of all the paths in d_x with d_y , with respect to the maximum number of paths in the two documents. *Psim* is used to measure the degree of similarity between two paths. It is defined as:

$$Psim(p_i, p_j) = \frac{\max(CNC(p_i, p_j), CNC(p_j, p_i))}{\max(|p_i|, |p_j|)} \quad (2)$$

CNC is the path similarity coefficient which is the sum of the elements occurring between two paths, p_i and p_j , in hierarchical order. *Psim* is used to measure the degree of similarity between two paths. *Psim* of paths, p_i and p_j , is the maximum similarity of the two CNC functions with respect to the maximum number of elements in both paths. Refer to Nayak and Tran [3] for more details on how the *CPSim* is measured for the structural similarity used in this clustering approach.

3.2 Content Mining using Latent Semantic Kernel.

For the content mining, given a collection of XML documents $\{d_1, d_2, \dots, d_n\}$, denoted by D , a set of distinct terms $\{t_1, t_2, \dots, t_l\}$, denoted by T , is extracted from D after the stop-word removal and stemming [5].

The content of a document is modelled as a vector $\{td_1, td_2, \dots, td_l\}$, where it contains the frequencies of the terms in the document. The text contents of the documents in a corpus, therefore, can be modelled together as a term-document matrix, $TD_{|T| \times |D|}$, where $|T|$ is the number of terms in T and $|D|$ is the number of documents in D . Each cell in matrix TD is the frequency of a term in each document.

In this approach, the semantic similarity of content is learned using a latent semantic kernel (LSK) [6] which is constructed based on latent semantic analysis (LSA) [7].

The LSA uses a term-document matrix such as $TD_{|T| \times |D|}$. SVD is applied to break this matrix into U , S and V , where U and V have orthonormal columns of right and left singular vectors, respectively, and S is diagonal matrix of singular values and are ordered in decreasing magnitude (highest value to the lowest value in diagonal). The SVD model can optimally approximate matrix TD with a smaller sample of matrices by selecting k largest singular values and setting the rest of the values to zero. Matrix U_k of $|T| \times k$ and matrix V_k of $|D| \times k$ may be redefined along with $k \times k$ singular value matrix S_k . This can approximate the matrix TD in a lower k -dimensional document space. Refer to Landauer et al. [7] for more technical details on SVD and latent semantic analysis methods. U_k is used as a kernel to learn new content information of documents in concept space.

Thus, given two content vectors, d_x and d_y , the content of the documents is measured using cosine measure which is defined as follows:

$$\frac{d_x^T P P^T d_y}{|P^T d_x| |P^T d_y|} \quad (3)$$

where matrix U_k , and P is used as a mapping function to transform two documents, d_x and d_y

In this approach, we propose to build the LSK by selecting a small subset of documents in the whole collection of input documents. Since the purpose of a clustering task is to group documents without prior taxonomy knowledge, therefore, selecting the small subset of these documents automatically can be difficult. In order to select the diverse samples of input documents, we propose to first cluster the documents based on the structural similarity according to the clustering approach as outlined in previous section. By taking this step, we assume documents describing similar information will have similar structure. Assuming that each cluster in the final clustering solution generated by the clustering approach contains similar documents, a small subset of documents from each cluster is automatically selected to construct the LSK. In cases where a cluster does not contain sufficient number of documents to be used for the development of the kernel, more documents are selected from other clusters with larger number of documents in them.

4 Clustering Approach

As mentioned throughout the paper, there are two clustering methods involved in the proposed clustering approach. The first one is an incremental clustering and the second one is a clustering method based on pairwise distance matrix.

The incremental clustering method used in this approach adopted the idea of hierarchical clustering which works as follows. It starts with no cluster; thus, the first document is used to form a new cluster. The document is used as the cluster representation for that cluster. In another word, the input documents are compared with the first document that is used to form the clusters. So when the next document comes in, it is then compared with the existing clusters using their document representation. If the similarity between the document and an existing cluster has the largest similarity value and it exceeds the clustering threshold defined by the user then the document is assigned to that cluster. However if the similarity value does not exceed the clustering threshold then the document forms a new cluster and the document is used to represent that cluster.

After the initial grouping of documents into an automatic number of clusters, an iteration process is executed. The iteration process re-runs the incremental process again but this time, it is comparing the documents with the existing clusters. Its purpose is to assign the input documents to a cluster with the maximum similarity value without using the clustering threshold. Depending on the number of clusters generated by the incremental clustering process, a merging

process may take place. The merging process is used in cases where the number of clusters generated by the incremental process is too large to compute a pairwise distance matrix for the clusters' representation documents. The merging process is used to merge all the clusters that only contain the document representation with the existing clusters that contain more than 1 document in order to reduce the dimension of the clusters. A pairwise distance matrix is computed by the clustering approach using the clusters' representation documents produced by the incremental process. For the structure, the distance between the documents and the clusters' representations is measured using *CPSim* as discussed previously and for the content, the distance between the documents and the clusters' representations is measured using cosine measure which has been discussed in the previous section. The same measures are used to compute the pairwise distance matrix.

The pairwise distance matrix is then fed into the second clustering method called graph clustering method. graph clustering method merges the clusters generated by the incremental clustering process according to the number of clusters defined by the user.

5 Experiments and Discussion

The Wikipedia collection containing 48035 documents used in the INEX 2007 document mining challenge, is used in experiments to evaluate the performance of the proposed clustering approach.

The experiments in this paper are set up to measure the two features of the XML documents for the clustering task: the structure and the content. There are two types of clusterings performed. The first clustering of the Wikipedia is based on the structure-only. The structure-only clustering uses the structure information without the content information to cluster the documents. The second clustering is based on the content-only. The content-only clustering uses the content information without the structure information to cluster the documents. The structure-only clustering uses the clustering threshold of 0.3 and the content-only clustering uses the clustering threshold of 0.9. In addition, the experiments are conducted using two different number of clusters. One is 10 clusters and the other is 21 clusters which are also required by the INEX 2007 challenge.

Tables 1 and 2 compare our clustering solution results with other participants in the INEX 2007 document mining challenge for the clustering task based on structure-only. The results for 10 clusters (table 1) do not vary much within various methods. With 21 clusters (table 2), Hagenbuchner et al. clustering solutions, with the verified version, are slightly higher than Kutty et al. and our method. Amongst the clustering solution results of all participants in the clustering task, the structure-only results are always worse in comparison to content-only or combination of both the structure and content. XML documents in Wikipedia collection conformed to the same structure definition (or are very similar in structure and name tags) which are used to describe different content

topic. Hence, it is hard to infer any unique structure representation for each individual category. Based on these results for structure-only, it can be ascertained that the structure of the testing corpus in INEX 2007 challenge does not play a significance role. In another word, no matter what approaches are used, the clustering solution based on the structure-only will possibly produce F1 values somewhere in the range of 0.2 and 0.3.

Table 1. Comparing the clustering results for structure-only on wikipedia dataset with 10 Clusters

Approaches	Micro F1	Macro F1
Hagenbuchner et al.(Not Verified)	0.251	0.257
Kutty et al.(Not Verified)	0.251	0.25
Our Approach	0.251	0.252

Table 2. Comparing the clustering Results for structure-only on wikipedia dataset with 21 Clusters

Approaches	Micro F1	Macro F1
Hagenbuchner et al.	0.264	0.269
Hagenbuchner et al.	0.258	0.252
Kutty et al.(Not Verified)	0.251	0.251
Our Approach	0.251	0.253

Figures 2 and 3 display the clustering solution results based on the content-only or content plus structure. The results in figures 2 and 3 only show one selected result (generally the best one) from each participant even though one participant may have many results submitted. From our experiments and data analysis, it has been discovered that the clustering threshold used for the incremental clustering has a great impact on the clustering solutions. Figure 4 shows the effect of the micro-f1 and macro-f1 values on the clustering solutions using different clustering thresholds. The figure shows 4 different clustering thresholds range from 0.6 to 0.9. It demonstrates that using a higher clustering threshold for the incremental clustering is better than a lower threshold. This is due to the fact that using a higher clustering threshold in the incremental clustering allows it to generate a large amount of clusters. Based on this large amount of clusters, a pairwise similarity matrix can be generated using the clusters' representation documents which can then be fed to graph clustering method for the final solution. The more data point used to compute the pairwise distance matrix the better the clustering solution is.

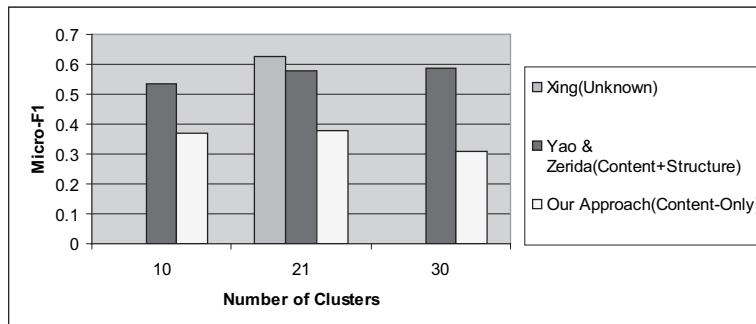


Fig. 2. Clustering micro-f1 results in INEX 2007 document mining challenge

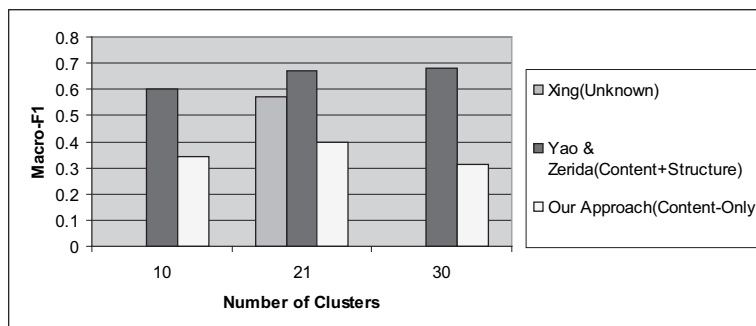


Fig. 3. Clustering macro-f1 results in INEX 2007 document mining challenge

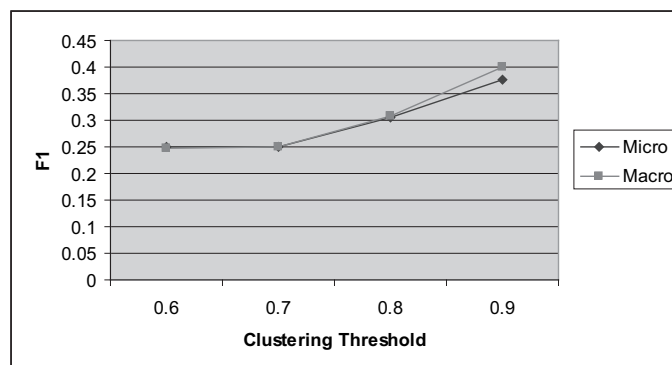


Fig. 4. The effect of the clustering solution with different clustering thresholds

6 Conclusions and Future Work

This paper has compared our proposed approach to other approaches in INEX 2007 Document Mining Challenge. Our approach is based on incremental clustering and pairwise-distance clustering using graph clustering method. From the results, it is ascertained, that based on the Wikipedia dataset, the structure does not improve significantly using different clustering approaches showing that the structure information of the Wikipedia collection plays a small role in determining the true categories of the Wikipedia dataset. Furthermore, based on the experiments and results, it has also showed that LSK does not perform well with incremental approach. However due to limited time this has not been verified significantly. In future work, we would explore in more deep on how LSK can be used in the incremental approach more effectively and how it can be used in parallel with the structure information.

References

1. : Initiative for the evaluation of xml retrieval (2007)
2. Karypis, G.: Cluto - software for clustering high-dimensional datasets — karypis lab
3. Nayak, R., Tran, T.: A progressive clustering algorithm to group the xml data by structural and semantic similarity. *IJPRAI* **21**(3) (2007) 1–21
4. Nayak, R., Xu, S.: Xcls: A fast and effective clustering algorithm for heterogenous xml documents. In: *PAKDD'2006*, Singapore (2006)
5. Porter, M.: An algorithm for suffix stripping. *Program* **14**(3) (1980) 130–137
6. Cristianini, N., Shawe-Taylor, J., Lodhi, H.: Latent semantic kernels. *Journal of Intelligent Information Systems (JJIS)* **18**(2) (2002)
7. Landauer, T.K., Foltz, P.W., Laham, D.: An introduction to latent semantic analysis. *Discourse Processes* (25) (1998) 259–284

Rare patterns to improve path-based clustering of Wikipedia articles

Jin Yao and Nadia Zerida
Caen University
GREYC Laboratory, CNRS UMR 6072
Caen, 14032, France
firstname.name@info.unicaen.fr

Abstract. In contrast of basic approach in text clustering, where documents are represented only by words and the structure is completely ignored. In this paper, we show how to improve XML clustering quality by combining content, structure and rare patterns. To represent document, we focus on completed root-based text path description, root-based text path description with lengths of two, three, and four. Finally, we combine path-based description with rare patterns. We use the word-based descriptors as a baseline. For this comparison, a constrained agglomerative clustering is used, and results show a significant improvement when rare patterns are combined with path based text description.

1 Introduction

Nowdays, XML text format represent an important element in the data exchange on the web. However, organizing XML documents according to their structural properties became a growing need. Most of available XML documents on the web, do not have an associated Document Type Descriptor (DTD). Therefore, the XML document clustering is based only on structure of the document. Several XML documents clustering methods based on their structure have been proposed. In [4], the authors view XML documents as trees, and recursively compute the overall distance between two XML trees from the root nodes to leaf nodes. In [1] the structure of an XML document is represented as a time series. By analyzing the coefficients of the corresponding Fourier transform it is possible to evaluate the degree of similarity between documents. In [8, 2], the tree is transformed into a bag of paths, bag of content or a mixture of both. On the other hand, recent results showed the important role when using rare patterns to improve XML categorization [9], these results encouraged us to think how to exploit rare patterns with path based methods in order to improve XML clustering of Wikipedia articles.

In this work, we used Wikipedia corpus [5]. The collection contains 96,611 documents that are classified into 21 categories. And each document belongs to exactly one category. The documents are formatted by XML tags that present primarily the logical structure of the document. There are not neither a general DTD for the total collection, nor any DTDs for sub sets of collection. The most

of content and tags are written in English. Other languages (French, Chinese, Arab, etc.) are used in some documents for showing proper name in the native language. The total collection is divided into two parts, one for training purpose, and another for testing.

This paper is organized as follows. Section 2 defines the different notions used in experimentation. The details of preprocessing step is given in Section 3. And the clustering method that we use is described in Section 4. In Section 5 experiments and results are discussed. Section 6 concludes.

2 Preliminaries

For an XML document, both of the content and the structure should be modeled into document representation. The content is the words sets of the document. And the structure is the XML tags sets of document. XML documents are usually represented by tree structure where the internal nodes are XML elements and the leaf nodes contain the content of document. The attributes of elements could be represented as leaf nodes or sub-element. An element could include other elements. The relationship of embed makes the hierarchy of the XML document. That could be called logical structure of the document. An example of Wikipedia XML document d_0 shown below:

```
<article>
<name id="795">Affidavit</name>
<conversionwarning>0</conversionwarning>
<body>An
<emph3>affidavit</emph3>
is a formal sworn statement of fact, written down, signed, and witnessed (as to
the veracity of the signature) by a taker of oaths, such as a
<collectionlink xlink:type="simple" xlink:href="21481.xml">
notary public
<collectionlink>for
<emph2>he has declared upon oath</emph2>
<p>One use of affidavit is before the court.</p>
<p>...</p>
<language link lang="he">????? </language link>
</body>
</article>
```

In [8], a decomposed XML document tree into a set of path-based descriptors, where the XML tags are combined with words is proposed.

Definition 1. *The path of a node n is a sequence of nodes name from root to this node, when traversing the tree from child to child. We note $p(n)$. It is also called completed path.*

Definition 2. *The length of path is the number of nodes in the path. We note $|p(n)|$.*

Definition 3. *A sub-paths of length l on a path p is a sequence of l consecutive nodes along the path p . (i.e. a sub-path does not necessarily start at the root). We note $|s|$ the length of the sub-path s .*

Definition 4. A text path (resp. sub-path) is a path (resp. sub-path) that ends with a word contained in the text associated with the last node in the path (resp. sub-path).

For instance, the Term frequency of paths and sub-paths of length 2 of document d_0 is,

$Tf(\text{article}/\text{name}/\text{affidavit})=1$, $Tf(\text{article}/\text{body}/\text{a})=3$, $Tf(\text{article}/\text{body}/\text{for})=1$,
 $Tf(\text{body}/\text{emph3}/\text{affidavit})=1$, $Tf(\text{body}/\text{emph2}/\text{oath})=1$, $Tf(\text{article}/\text{name}/)=1$,
 $Tf(\text{article}/\text{conversionwarning}/)=1$, $Tf(\text{article}/\text{body}/)=1$, $Tf(\text{body}/\text{emph3}/)=1$,
 $Tf(\text{body}/\text{p}/)=2$.

Definition 5. A root-based text path rtp of length l is a path of length l associated with a word that is contained in the descendant nodes of the last node of the path.

The Term frequency of root-based text path of length 1 and 2 is,

$Tf(\text{article}/\text{name}/\text{affidavit})=1$, $Tf(\text{article}/\text{body}/\text{a})=3$, $Tf(\text{article}/\text{affidavit})=3$,
 $Tf(\text{article}/\text{body}/\text{affidavit})=2$ and $(Tf(\text{article}/\text{body}/\text{oath})=1$.

The textual path (article/affidavit) have no the direct leaf node as the terminal node, the words in his descendant leaf nodes are associated with the path.

Instead of mix up several types of path-based descriptors, we do our experiments with a specific path-based representation.

3 Preprocessing

In order to use traditional clustering methods for flat text, the vector space model is one of common models of textual document for translating document semantic into suitable data structure [7]. The path-based descriptors is a bag of strings that could be seen as words. Then, the vector space model is used to represent the document. On the other hand, the disadvantage of the vector space model representation is that the dimension size is very huge. The size of the path-based descriptor is naturally bigger than words-based descriptors. Therefore, we limit the number of descriptors. We reduce some descriptors in two main steps. Firstly, we use standard method used for reducing words descriptors:

- Delete insignificant words from a stoplist.
- Numbers and words with length less than 3 are deleted.
- Porter stemming [6] is used.

Secondly, we create path-based descriptors with the selected words obtained in the first step. Then, we reduce path-based descriptors using the distribution of descriptors in the collection. If a descriptor presents too rare or too common, this descriptor will be deleted. A list of descriptors which occurs only once is constructed, and we delete the descriptors which occurs over 80% in the collection. For the remaining descriptors, we calculate their Tf Idf value.

The frequency of descriptor in the document Tf_t is the occurrence of the descriptor over the number of whole descriptors in the document. We normalise Tf_t

by the max occurrence of descriptors in the document. The Inverse document frequency Idf_t is calculated by,

$$Idf_t = \frac{\log|D|}{|\{t \in d\}|}$$

Where, $|D|$ is the total number of documents in the collection and $|\{t \in d\}|$ is the number of documents contains the descriptor t .

4 Agglomerative clustering

In this work, we used a constrained agglomerative method proposed by [10,11], which combines features from both partitional and agglomerative approaches. Initially, a random pair of documents is selected from the collection to act as the seeds of the two clusters. Then for each document, its similarity to these two seeds is computed and it is assigned to the cluster corresponding to its most similar seed. This clustering is then repeatedly refined so that it optimizes the desired clustering criterion function. Then, a cluster is selected to be bisected. This process continues several times until demanded cluster number is satisfied. The experimental evaluation showed that constrained agglomerative lead to better solutions than agglomerative methods alone for text clustering. For many cases, it even better then partitional methods. We used the software CLUTO developed by [10] in our experimentation.

Similarity measure The similarity of two documents is measure by calculating the cosine of the angle between the vectors:

$$\cos\theta = \frac{v_1 \cdot v_2}{\|v_1\| \|v_2\|}$$

Where, $\|v_i\|$ is the length of the vector.

5 Experimental Results on Wikipedia articles

As mentionned in Section 4, we used for all experiments the agglomerative clustering classifier to compare the influence of different document representations on the clustering performance. For each type of representation, we realised five series of clustering respectively, 5, 10, 15, 21 and 30 number of clstures. The document collection is divided into two sets. One is prepared to training and another is to test the participant's model. The test sets contains 48305 documents. Each document belongs to one category. There are 21 categories for all of documents. We used the word-based descriptors as a baseline.

Firstly, we totally ignored the organization of words in document. The document is considered as a "bag of words". The document is vectorised by separated

words. The dimension of vector space corresponds to the all words in the collection. The document vector’s position in vector space is located by the words appearing in the document. And in order to keep the non-alphabetic characters like Chinese characters, the words whose length is less than 3 are not deleted. The size of words is 130969. The size of word-based matrix is 48305×130969 .

Secondly, we computed four path-based matrixes. The sizes of descriptors are, 485614, 137034, 240585 and 360088, respectively for, completed text path descriptors, root-based path with length of 2, root-based path with length of 3 and root-based path with length of 4. We tested a set of root-based text path in our experimentation. We created the completed text path descriptor. The completed path starts at root node and ends at leaf node. The attributes of elements are ignored in all of our tests. We distributed different lengths of the root-based text path for measuring progressively the results. We created three sets of descriptors with length 2, 3 and 4 (i.e. For Wikipedia corpus, when length = 1, the root-based text path descriptor is equal to words descriptor*).

Finally, We combined a path-based matrix with filtered words descriptors that are created from the singleton path-based descriptors. We get the reduced completed text path descriptors. They are the rare patterns or the too common patterns. We got rid of the path part of the descriptor, the descriptors became words. We used these words as the candidate word-based descriptors. We passed again the feature selection to reduce descriptor size, then we created filtered words matrix. We combined the path-based matrix with filtered words matrix. And each document vector is renormalized by the maximal occurrence extracted among the descriptors of the document. The new matrix is called path-based-hapax matrix. Its descriptor size is 604581.

The clustering performance is evaluated by using measures proposed by INEX XML mining track. Then, micro and macro average purities are calculated for all experiments, and the obtained results are summerized in Table 1.

Table 1. Results for different text representations for XML Wikipedia corpus

# of Clusters	word descriptor		completed-root-path		combined-rare&path	
	Micro-purity	Macro-purity	Micro-purity	Macro-purity	Micro-purity	Macro-purity
5	44,48	44,76	30,35	35,17	41,53	46,10
10	49,94	57,86	40,36	48,44	46,06	60,66
15	53,41	63,15	42,40	47,67	46,98	58,41
21	57,94	67,28	43,56	49,08	51,53	61,04
30	58,87	67,98	44,62	52,48	54,75	68,04
# of Clusters	2-root-path descriptor		3-root-path descriptor		4-root-path descriptor	
	Micro-purity	Macro-purity	Micro-purity	Macro-purity	Micro-purity	Macro-purity
5	44,45	44,03	32,63	43,18	30,37	36,21
10	53,43	60,23	42,08	47,89	37,74	48,73
15	53,63	59,10	44,46	56,56	41,44	45,92
21	56,44	63,25	45,75	55,39	42,06	49,21
30	59,52	66,37	48,44	55,96	44,29	51,73

Figure 1 and 2 shows respectively, the micro and macro-average purity values of the different text representations. The value on X axis represents the number of clusters and the value on Y axis represents the micro and macro-average purity calculated by INEX XML mining track organizers. The micro-purity baseline value is 58,86% and the macro-purity baseline value is 67,97% (see the word based descriptor representation in Table 1 for details). All text representations methods show a significant increase in clustering performance, especially when using root-based text path descriptor with length of two, and when combining path based descriptor with rare patterns. The results obtained by combining path based descriptor with rare patterns provide the best clustering performance than the word-based descriptors, with a value of 68,04% of macro-purity.

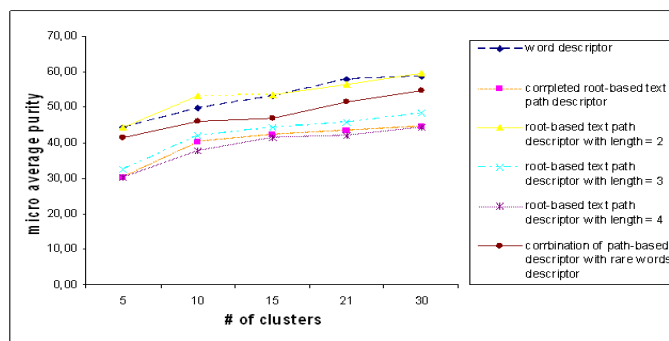


Fig. 1. The dependency of micro-average purity on the number of clusters by using different text representations

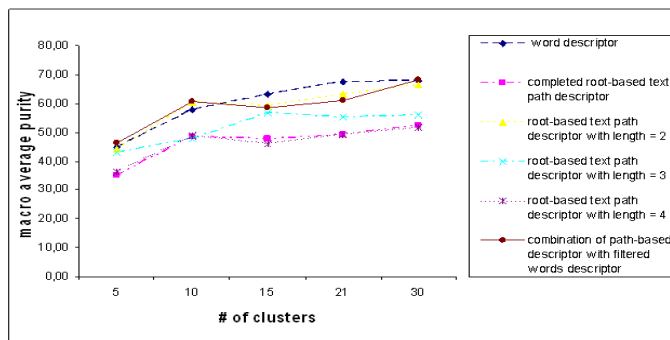


Fig. 2. The dependency of macro-average purity on the number of clusters by using different text representations

The results show that the purity values reduced when the paths length increased. The completed textual path could get the better purity values than short length path representation when number of clusters equals to 30, the possible reason of this reduction could come from adding new path names in descriptors set. The combination of path-based descriptor and filtered words descriptor wins better performance than the word-based descriptor.

The purities of clustering with path-based descriptor are worse than the purities of clustering with word-based descriptor. This result is not what we waited for. This can be interpreted by the difference of XML tags in different documents generated by the absence of a DTD. Some tags have the same function, but they have different names in different documents. Consequently, this diversity of the structures enlarges the distances between the content documents. The documents could be structured in simple form, but they could be structured with richer semantic.

6 Conclusion

In this work, we were focused on the comparison of different path based document description in terms of improving clustering quality. Our experimental results showed the effectiveness of combining content, structure and rare patterns. We strongly believe that by considering these three factors, and by adding more constraints on rare patterns will improve clustering performance, that will be the next step in the future work.

References

1. Flesca S., Manco G., Masciari E., Pontieri L. and Pugliese A. Detecting Structural Similarities between XML Documents. In Proceedings of the International Workshop on the Web and Databases(WebDB), 2002.
2. Joshi S., Agrawal N., Krishnapuram R. and Negi S. A bag of paths model for measuring structural similarity in Web documents, In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, 2003.
3. Leung H., Chung F., Chan SCF. and Luk R. XML Document Clustering Using Common XPath, In Web Information Retrieval and Integration, 2005. WIRI'05 Proceedings of the 2005 International Workshop on Challenges.
4. Long J., Schwartz D., and Soecklin S. An XML Distance Measure. In Proceedings of the International Conference on Data Mining (DMIN), 2005.
5. Denoyer, L. and Gallinari, P. The Wikipedia XML Corpus, In Advances in XML Information Retrieval and Evaluation: Fifth Workshop of the INitiative for the Evaluation of XML Retrieval (INEX'06), Dagstuhl, Germany, 2007
6. Porter, M. The Porter Stemming Algorithm, <http://www.tartarus.org/~martin/PorterStemmer/>
7. Salton G. Automatic Text Processing Addison-Wesley Publishing Company, 1988

8. Vercoustre A.M., Fegas M., Gul S. and Lechevallier, Y. A Flexible Structured-based Representation for XML Document Mining, Workshop of the INitiative for the Evaluation of XML Retrieval. (2005) 443-457
9. Zerida N., Lucas N. and Crémilleux B. Exclusion-Inclusion based Text Categorization of biomedical articles ACM Symposium on Document Engineering, Winnipeg, Canada, p.202-204, 2007.
10. Zhao Y. and Karypis G. Empirical and Theoretical Comparisons of Selected Criterion Functions for Document Clustering, Machine Learning, 55, pp. 311-331, 2004.
11. Zhao Y. and Karypis G. Hierarchical Clustering Algorithms for Document Datasets, Data Mining and Knowledge Discovery, Vol. 10, No. 2, pp. 141 - 168, 2005.

Probabilistic Methods for Structured Document Classification at INEX'07

Luis M. de Campos, Juan M. Fernández-Luna, Juan F. Huete, and Alfonso E. Romero

Departamento de Ciencias de la Computación e Inteligencia Artificial
E.T.S.I. Informática y de Telecomunicación, Universidad de Granada,
18071 – Granada, Spain
{lci, jmfluna, jhg, aeromero}@decsai.ugr.es

Abstract. This paper exposes the results of our participation in the Document Mining track of INEX'07. We have focused on the task of classification of XML documents. Our approach to deal with structured document representations uses classification methods for plain text, applied to flattened versions of the documents, where some of their structural properties have been translated to plain text. We have explored several options to convert structured documents into flat documents, in combination with two probabilistic methods for text categorization. The main conclusion of our experiments is that taking advantage of document structure to improve classification results is a difficult task.

1 Introduction

This is the first year that members of the research group “Uncertainty Treatment in Artificial Intelligence” at the University of Granada submit runs to the Document Mining track of INEX. As we had previous experience in automatic classification, particularly in learning Bayesian network classifiers [1, 3], we have limited our participation only to the task of text categorization.

The proposed methodology does not use text classification algorithms specifically designed to manage and exploit structured document representations. Instead, we use algorithms that apply to flat documents and do not take structure into consideration at all. What we want to test is whether these methods can be used, in combination with some simple techniques to transform document structure into a modified flat document representation having additional characteristics (new or transformed features, different frequencies,...), in order to improve the classification results obtained by the same methods but using purely flat document representations.

The rest of the paper is organized in the following way: in Section 2 we describe the probabilistic flat text classifiers we shall use. Section 3 gives details of the different approaches to map structured documents into flat ones. Section 4 is focused on the experimental results.

2 Methods for Flat Text Classification

In this section we are going to explain the two methods for non-structured (flat) text classification that we are going to use in combination with several methods for managing structured documents. One of them is the well-known Naive Bayes classifier, whereas the other is a new method, based on a restricted type of Bayesian network.

The classical probabilistic approach to text classification may be stated as follows: We have a class variable C taking values in the set $\{c_1, c_2, \dots, c_n\}$ and, given a document d_j to be classified, the posterior probability of each class, $p(c_i|d_j)$, is computed according to the Bayes formula:

$$p(c_i|d_j) = \frac{p(c_i)p(d_j|c_i)}{p(d_j)} \propto p(c_i)p(d_j|c_i) \quad (1)$$

and the document is assigned to the class having the greatest posterior probability, i.e.

$$c^*(d_j) = \arg \max_{c_i} \{p(c_i)p(d_j|c_i)\}$$

Then the problem is how to estimate the probabilities $p(c_i)$ and $p(d_j|c_i)$.

2.1 The Naive Bayes Classifier

The naive Bayes classifier is the simplest probabilistic classification model that, despite its strong and often unrealistic assumptions, perform frequently surprisingly well. It assumes that all the attribute variables are conditionally independent of each other given the class variable. In fact, the naive Bayes classifier can be considered as Bayesian network-based classifier, where the network structure contains only arcs from the class variable to the attribute variables, as shown in Figure 1. In the context of text classification, there exist two different models called naive Bayes, the multivariate Bernoulli naive Bayes model [4, 5, 9] and the multinomial naive Bayes model [6, 7]. In this paper we are going to use the multinomial model.

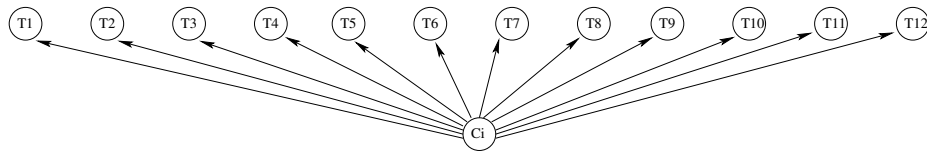


Fig. 1. The naive Bayes classifier.

In this model a document is an ordered sequence of words or terms drawn from the same vocabulary, and the naive Bayes assumption here means that

the occurrences of the terms in a document are conditionally independent given the class, and the *positions* of these terms in the document are also independent given the class. Thus, each document d_j is drawn from a multinomial distribution of words with as many independent trials as the length of d_j . Then,

$$p(d_j|c_i) = p(|d_j|) \frac{|d_j|!}{\prod_{t_k \in d_j} n_{jk}!} \prod_{t_k \in d_j} p(t_k|c_i)^{n_{jk}} \quad (2)$$

where t_k are the distinct words in d_j , n_{jk} is the number of times the word t_k appears in the document d_j and $|d_j| = \sum_{t_k \in d_j} n_{jk}$ is the number of words in d_j . As $p(|d_j|) \frac{|d_j|!}{\prod_{t_k \in d_j} n_{jk}!}$ does not depend on the class, we can omit it from the computations, so that we only need to calculate

$$p(d_j|c_i) \propto \prod_{t_k \in d_j} p(t_k|c_i)^{n_{jk}} \quad (3)$$

The estimation of the term probabilities given the class, $p(t_k|c_i)$, is usually carried out by means of the Laplace estimation:

$$p(t_k|c_i) = \frac{N_{ik} + 1}{N_i + M} \quad (4)$$

where N_{ik} is the number of times the term t_k appears in documents of class c_i , N_i is the total number of words in documents of class c_i and M is the size of the vocabulary (i.e. the number of distinct words in the documents of the training set).

The estimation of the prior probabilities of the classes, $p(c_i)$, is usually done by maximum likelihood, i.e.:

$$p(c_i) = \frac{N_{i,doc}}{N_{doc}} \quad (5)$$

where N_{doc} is the number of documents in the training set and $N_{i,doc}$ is the number of documents in the training set which are assigned to class c_i .

In our case we have used this multinomial naive Bayes model but, instead of considering only one class variable C having n values, we decompose the problem using n binary class variables C_i taking its values in the sets $\{c_i, \bar{c}_i\}$. This is a quite common transformation in text classification [10], especially for multilabel problems, where a document may be associated to several classes. In this case we build n naive Bayes classifiers, each one giving a posterior probability $p_i(c_i|d_j)$ for each document. As in the Wikipedia XML Corpus each document may be assigned to only one class, we select the class c^* such that $c^* = \arg \max_{c_i} \{p_i(c_i|d_j)\}$.

2.2 The OR Gate Bayesian Network Classifier

The second classification method for flat documents that we are going to use is based on a Bayesian network with the following topology: Each term t_k appearing

in the training documents (or a subset of these terms in the case of using some method for feature selection) is associated to a binary variable T_k taking its values in the set $\{t_k, \bar{t}_k\}$, which in turn is represented in the network by the corresponding node. There are also n binary variables C_i taking its values in the sets $\{c_i, \bar{c}_i\}$ (as in the previous binary version of the naive Bayes model) and the corresponding class nodes. The network structure is fixed, having an arc going from each term node T_k to the class node C_i if the term t_k appears in training documents which are of class c_i . In this way we have a network topology with two layers, where the term nodes are the “causes” and the class nodes are the “effects”. An example of this network topology is displayed in Figure 2.

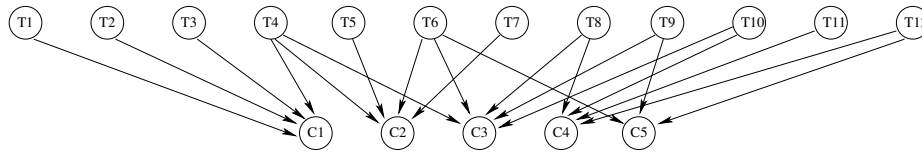


Fig. 2. The OR gate classifier.

The quantitative information associated to this network are the conditional probabilities $p(C_i|pa(C_i))$, where $Pa(C_i)$ is the set of parents of node C_i in the network (i.e. the set of terms appearing in documents of class c_i) and $pa(C_i)$ is any configuration of the parent set (any assignment of values to the variables in this set). As the number of configurations is exponential with the size of the parent set, we use a canonical model to define these probabilities, which reduce the number of required numerical values from exponential to linear size. More precisely, we use a noisy OR Gate model [8].

The conditional probabilities in a noisy OR gate are defined in the following way:

$$p(c_i|pa(C_i)) = 1 - \prod_{T_k \in R(pa(C_i))} (1 - w(T_k, C_i)), \quad p(\bar{c}_i|pa(C_i)) = 1 - p(c_i|pa(C_i)). \quad (6)$$

where $R(pa(C_i)) = \{T_k \in Pa(C_i) | t_k \in pa(B)\}$, i.e. $R(pa(C_i))$ is the subset of parents of C_i which are instantiated to its t_k value in the configuration $pa(C_i)$. $w(T_k, C_i)$ is a weight representing the probability that the occurrence of the “cause” T_k alone (T_k being instantiated to t_k and all the other parents T_h instantiated to \bar{t}_h) makes the “effect” true (i.e., forces class c_i to occur).

Once the weights $w(T_k, C_i)$ have been estimated, and given a document d_j to be classified, we instantiate in the network each of the variables T_k corresponding to the terms appearing in d_j to the value t_k , and all the other variables T_h (those associated to terms that do not appear in d_j) to the value \bar{t}_h . Then, we compute for each class node C_i the posterior probabilities $p(c_i|d_j)$. As in the case of the naive Bayes model, we assign to d_j the class having the greatest posterior probability.

The combination of network topology and numerical values represented by OR gates allows us to compute very efficiently and in an exact way the posterior probabilities:

$$p(c_i|d_j) = 1 - \prod_{T_k \in Pa(C_i)} (1 - w(T_k, C_i) \times p(t_k|d_j)) = 1 - \prod_{T_k \in Pa(C_i) \cap d_j} (1 - w(T_k, C_i)). \quad (7)$$

In order to take into account the number of times a word t_k occurs in a document d_j , n_{jk} , we replicate each node T_k n_{jk} times, so that the posterior probabilities then become

$$p(c_i|d_j) = 1 - \prod_{T_k \in Pa(C_i) \cap d_j} (1 - w(T_k, C_i))^{n_{jk}}. \quad (8)$$

The estimation of the weights in the OR gates, $w(T_k, C_i)$, can be done in several ways. The simplest one is to estimate $w(T_k, C_i)$ as $p(c_i|t_k)$, the conditional probability of class c_i given that the term t_k is present. We can do it by maximum likelihood:

$$w(T_k, C_i) = \frac{N_{ik}}{N_k} \quad (9)$$

where N_k is the number of times that the term t_k appears in the training documents.

Another way, more accurate, of estimating $w(T_k, C_i)$ is directly as $p(c_i|t_k, \bar{t}_h \forall T_h \in Pa(C_i), T_h \neq T_k)$. However, this probability cannot be reliably estimated, so that we are going to compute an approximation in the following way¹:

$$p(c_i|t_k, \bar{t}_h \forall h \neq k) \approx p(c_i|t_k) \prod_{h \neq k} \frac{p(c_i|\bar{t}_h)}{p(c_i)} \quad (10)$$

The values of $p(c_i|t_k)$ and $p(c_i|\bar{t}_h)/p(c_i)$ in eq. (10) are also estimated using maximum likelihood. Then, the weights $w(T_k, C_i)$ are in this case:

$$w(T_k, C_i) = \frac{N_{ik}}{N_k} \times \prod_{h \neq k} \frac{(N_i - N_{ih})N}{(N - N_h)N_i} \quad (11)$$

where N is the total number of words in the training documents.

3 Document representation

In this section we deal with the problem of document representation. As we have seen before, we are using flat-document classifiers for this track, so we need methods to translate structural properties to plain text document.

¹ This approximation results from assuming that $p(t_k, \bar{t}_h \forall h \neq k|c_i) \approx p(t_k|c_i) \prod_{h \neq k} p(\bar{t}_h|c_i)$.

Because these methods are independent of the classifier used, it is possible to make all possible combinations of classifiers and transformation methods, which gives us a large amount of categorization procedures.

We shall use the small XML document (the beginning of “El Quijote”) displayed in Figure 3 to illustrate the proposed transformations. We now explain the different approaches to map structural documents into flat ones.

```
<book>
<title>El ingenioso hidalgo Don Quijote de la Mancha</title>
<author>Miguel de Cervantes Saavedra</author>
<contents>
  <chapter>Uno</chapter>
  <text>En un lugar de La Mancha de cuyo nombre no quiero
    acordarme...</text>
</contents>
</book>
```

Fig. 3. “Quijote”, XML fragment used to illustrate the different transformations.

3.1 Method 1: “Only text”

This is the naive approach. It consists in removing all the structural marks from the XML file, obtaining a plain text file. Used with the previous example, we obtain the following document:

```
El ingenioso hidalgo Don Quijote de la Mancha Miguel de Cervantes
Saavedra Uno En un lugar de La Mancha de cuyo nombre no quiero
acordarme...
```

Fig. 4. “Quijote”, with “only text” approach

This method should be taken as a *baseline*, as we are losing all the structural information. We would like to improve its classification accuracy by using more advanced representations.

3.2 Method 2: “Adding”

This method adds structural features to the document, different from the textual features. That is to say, structural marks are introduced into the document as if they were “additional terms”. We can consider structural marks in an atomic way, or in the context of the other marks where they are contained (i.e. considering part of the path to the root element, until a certain depth level). Using the

previous example, the `text` mark can be considered standalone (“adding_1”, with depth = 1), `contents_text` (“adding_2”, depth = 2) or `book_contents_text` (“adding_0”, maximum depth value, the complete path to the root mark).

We show here the transformed flat document of the example document using “adding” with depth = 2. Leading underscores are used to distinguish between textual terms and terms representing structural marks:

```
_book_book_title El ingenioso hidalgo Don Quijote de la Mancha
_book_author Miguel de Cervantes Saavedra
_book_contents_contents_chapter Uno _contents_text En un lugar
de La Mancha de cuyo nombre no quiero acordarme...
```

Fig. 5. “Quijote”, with “adding_2”

3.3 Method 3: “Tagging”

This approach is the same as the one described in [2], and also named “tagging”. It considers that two appearances of a term are different if it appears inside two different structural marks. To modelize this, terms are “tagged” with a representation of the structural mark they appear in. This can be easily simulated prepending a prefix to the term, representing its container. We can also experiment at different depth levels, as we did in the method “adding”.

Data preprocessed with this method can be very sparse, and very large lexicon could be built from medium sized collections. For our example document this method, with depth = 1, obtains:

```
title_El title_ingenioso title_hidalgo title_Don title_Quijote
title_de title_la title_Mancha author_Miguel author_de
author_Cervantes author_Saavedra chapter_Uno text_En text_un
text_lugar text_de text_La text_Mancha text_de text_cuyo
text_nombre text_no text_quiero text_acordarme...
```

Fig. 6. “Quijote”, with “tagging_1”

3.4 Method 4: “No text”

This method tries to unveil the categorization power using only structural units, processed in the same way as in the “adding” method. Roughly speaking, it is equivalent to “adding” and removing textual terms. In the next example we can see the “notext_0” processing of the previous example:

```
_book_book_title _book_author _book_contents
_book_contents_chapter _book_contents_text
```

Fig. 7. “Quijote”, with “notext.0”

3.5 Method 5: “Text replication”

The previous methods deal with a structured collection, having no previous knowledge of it. That is to say, they have not taken into account the kind of mark, in order to select one action or another. This approach assigns an integer value to each mark, proportional to its informative content for categorization (the higher the value, the more informative). This value is used to *replicate* terms, multiplying their frequency in a mark by that factor. Note that you must only supply values for structural marks directly containing terms.

In the previous example, suppose we assign the following set of replication values:

```
title      1
author     0
chapter    0
text       2
```

Note that a value of 0 indicates that the terms in that mark will be removed. The resulting text is in this case:

```
El ingenioso hidalgo Don Quijote de la Mancha En En un un lugar
lugar de de La La Mancha Mancha de de cuyo cuyo nombre nombre no
no quiero quiero acordarme acordarme...
```

Fig. 8. “Quijote”, with “replication” method, using values proposed before

This method is very flexible, and it generalizes several ones, as the “only text” approach (one may select 1 for all the replication values). The method consisting of just selecting text from certain marks can be simulated here using 1 and 0 replication values if the text within a given mark is to be considered or not, respectively.

The main drawback of “replication” is that we need some experience with the collection, used to build the table of replication values before processing the files.

4 Experimentation

In order to select the best combinations of classifiers and representations, we have carried out some experiments with the training set, using cross-validation

(dividing the training set into 5 parts). The selected evaluation measures are the same used in the final evaluation procedure: macroaverage and microaverage breakeven point (for *soft categorization*) and macroaverage and microaverage F1 (for *hard categorization*).

In every case, the “notext” representation will be used as a baseline to compare results among different alternatives.

4.1 Description of the files used for replication

Note that unspecified replication values are set to 1.

Replication, id=2:

conversionwarning	0
emph2	2
emph3	2
name	2
title	2
caption	2
collectionlink	2
languagelink	0
template	0

Replication, id=3:

conversionwarning	0
emph2	3
emph3	3
name	3
title	3
caption	3
collectionlink	3
languagelink	0
template	0

Replication, id=4:

conversionwarning	0
emph2	4
emph3	4
name	4
title	4
caption	4
collectionlink	4
languagelink	0
template	0

Replication, id=5:

conversionwarning	0
emph2	5
emph3	5
name	5
title	5
caption	5
collectionlink	5
languagelink	0
template	0

Replication, id=8:

conversionwarning	0
emph2	10
emph3	10
name	20
title	20
caption	10
collectionlink	10
languagelink	0
template	0

Replication, id=11:

conversionwarning	0
emph2	30
emph3	30
name	100
title	50
caption	10
collectionlink	10
languagelink	0
template	0

We have also carried out experiments with some feature/term selection methods. For the naive Bayes model we used a simple method that removes all the terms that appear in less than a specified number of documents. For the OR gate model we used a local selection method (different terms may be selected for different class values) based on computing the mutual information measure between each term and each class variable C_i .

4.2 Numerical results

Method	Representation	Selection?	micro BEP	macro BEP	micro F1	macro F1
Naïve Bayes	Only text	no	0.76160	0.58608	0.78139	0.64324
Naïve Bayes	Only text	≥ 2 docs.	0.72269	0.67379	0.77576	0.69309
Naïve Bayes	Only text	≥ 3 docs.	0.69753	0.67467	0.76191	0.68856
Naïve Bayes	Adding_1	None	0.75829	0.56165	0.76668	0.58591
Naïve Bayes	Adding_1	≥ 3 docs.	0.68505	0.66215	0.74650	0.65390
Naïve Bayes	Adding_2	None	0.73885	0.55134	0.74413	0.54971
Naïve Bayes	Adding_2	≥ 3 docs.	0.66851	0.62747	0.71242	0.59286
Naïve Bayes	Adding_3	None	0.71756	0.53322	0.72571	0.51125
Naïve Bayes	Adding_3	≥ 3 docs.	0.64985	0.59896	0.68079	0.53859
Naïve Bayes	Tagging_1	None	0.72745	0.49530	0.72999	0.50925
Naïve Bayes	Tagging_1	≥ 3 docs.	0.65519	0.60254	0.71755	0.60594
Naïve Bayes	Replication (id=2)	None	0.76005	0.64491	0.78233	0.66635
Naïve Bayes	Replication (id=2)	≥ 2 docs.	0.71270	0.68386	0.61321	0.73780
Naïve Bayes	Replication (id=2)	≥ 3 docs.	0.70916	0.68793	0.73270	0.65697
Naïve Bayes	Replication (id=3)	None	0.75809	0.67327	0.77622	0.67101
Naïve Bayes	Replication (id=4)	None	0.75921	0.69176	0.76968	0.67013
Naïve Bayes	Replication (id=5)	None	0.75976	0.70045	0.76216	0.66412
Naïve Bayes	Replication (id=8)	None	0.74406	0.69865	0.72728	0.61602
Naïve Bayes	Replication (id=11)	None	0.72722	0.67965	0.71422	0.60451
OR Gate (ML)	Only text	None	0.37784	0.38222	0.59111	0.37818
OR Gate (ML)	Only text	MI	0.74014	0.72816	0.74003	0.68430
OR Gate (Appr.)	Only text	None	0.79160	0.76946	0.79160	0.74922
OR Gate (Appr.)	Only text	≥ 3 docs.	0.77916	0.78025	0.77916	0.73544
OR Gate (Appr.)	Only text	≥ 2 docs.	0.79253	0.78135	0.79253	0.75300
OR Gate (ML)	Adding_1	None	0.40503	0.43058	0.58777	0.39361
OR Gate (ML)	Adding_1	≥ 3 docs.	0.39141	0.41191	0.57809	0.36936
OR Gate (ML)	Adding_1	MI	0.69944	0.72460	0.69943	0.58835
OR Gate (ML)	Adding_2	None	0.40573	0.43335	0.58908	0.39841
OR Gate (ML)	Adding_2	≥ 3 docs.	0.39204	0.41490	0.57951	0.37346
OR Gate (ML)	Adding_2	MI	0.65642	0.70755	0.65642	0.52611
OR Gate (ML)	Notext_2	None	0.40507	0.42914	0.48818	0.38736
OR Gate (ML)	Tagging_1	None	0.37859	0.40726	0.57274	0.35418
OR Gate (ML)	Tagging_1	≥ 3 docs.	0.36871	0.38475	0.56030	0.32546
OR Gate (ML)	Tagging_1	MI	0.59754	0.67800	0.59754	0.39141
OR Gate (Appr.)	Tagging_1	None	0.73784	0.74066	0.73789	0.70121
OR Gate (ML)	Replication (id=2)	MI	0.74434	0.73908	0.74432	0.66995
OR Gate (Appr.)	Replication (id=2)	None	0.78042	0.76158	0.78042	0.73768
OR Gate (ML)	Replication (id=3)	MI	0.74612	0.74275	0.74608	0.67249
OR Gate (Appr.)	Replication (id=3)	None	0.78127	0.76095	0.78127	0.73756
OR Gate (ML)	Replication (id=4)	MI	0.74815	0.74623	0.74813	0.67357
OR Gate (Appr.)	Replication (id=4)	None	0.78059	0.75971	0.78059	0.73511
OR Gate (ML)	Replication (id=5)	MI	0.74918	0.74643	0.74916	0.67498
OR Gate (Appr.)	Replication (id=5)	None	0.77977	0.75833	0.77978	0.73245
OR Gate (ML)	Replication (id=8)	MI	0.75059	0.75254	0.75059	0.66702
OR Gate (Appr.)	Replication (id=11)	None	0.77270	0.74943	0.77270	0.72186
OR Gate (ML)	Replication (id=11)	MI	0.72656	0.71326	0.72656	0.64101
OR Gate (Appr.)	Replication (id=11)	None	0.73041	0.70260	0.73041	0.66733

Keys:

- OR Gate (ML): OR gate classifier using eq. (9).
- OR Gate (Appr.): OR gate classifier using eq. (10).
- $\geq i$: selected terms that appear in more than or equal to i documents.
- MI: terms selected using mutual information.

4.3 Conclusions from these results

At a first sight, the best classifier in the four measures is a flat text classifier, the better approximation for the OR Gate. However, the simpler version (maximum likelihood) over a replication data set, and using term selection works almost equal.

It is a clear fact that the “replication” approach helps the Naïve Bayes classifier. One of the main drawbacks of this classifier are the bad results obtained, generally in macro measures (due to the nature of the classifier, that benefits the classes with higher number of training examples). This fact can be solved easily using a replication approach as stated in the table of results.

On the other hand, adding and tagging methods do not seem to give good results, using these classifiers. The runs with the “notext” approach were also really disappointing and they are not listed here.

5 Submitted runs

Finally, we decided to submit the following five runs to the Document Mining track:

- (1) **Naive Bayes**, only text, no term selection. Microaverage: 0.77630. Macroaverage: 0.58536.
- (2) **Naive Bayes**, replication (id=2), no term selection. Microaverage: 0.78107. Macroaverage: 0.6373.
- (3) **Or gate**, maximum likelihood, replication (id=8), selection by MI. Microaverage: 0.75097. Macroaverage: 0.61973.
- (4) **Or gate**, maximum likelihood, replication (id=5), selection by MI. Microaverage: 0.75354. Macroaverage: 0.61298.
- (5) **Or gate**, better approximation, only text, ≥ 2 . Microaverage: 0.78998. Macroaverage: 0.76054.

Note that the order among these classifiers is the same than in the previous table, and the final evaluation measures are close to the previous presented estimators.

6 Final remarks

Our participation in the XML Document Mining track of the INEX 2007 Workshop is shown in this work. It has been the first year we apply for this track

but however, and despite the low number of participants in the Categorization approach, our participation was remarkable. The main relevant results presented here are the following:

- We have described a new approach for flat document classification, the so called “OR gate classifier”, with two different variants: ML estimation, and more accurate approximation.
- We have shown different methods of representing structured documents as plain text ones. We must also recall that some of them are new.
- According to the results, we found that we could improve categorization of structured documents using a multinomial naive Bayes classifier, which is widely known and is included in almost every text-mining package software, in combination with the replication method.

On the other hand, the present paper raises the following questions that can be stated as future lines of work:

- How are the results of our models compared to a SVM applied on the documents with only text?
- Can the Naive Bayes classifier be improved more using a more sophisticated feature selection method?
- Having in mind that the replication approach is the one that has given the best results, what are the optimum replication parameters that can be used in Wikipedia? In other words, what marks are more informative and how much?
- Is there a way to make a representation of the structure of documents that could be used to improve the results of the or gate classifier (in its better approximation)?
- Do the “adding”, “tagging” and “no text” approaches help other categorization methods, like, for instance, SVMs?

Managing structure in this problem has been shown as a difficult task. Besides, it is not really clear if the structure can make a good improvement of categorization results. So, we hope to start answering the previous questions in future editions of this track.

Acknowledgments. This work has been jointly supported by the Spanish Ministerio de Educación and Ciencia, and Junta de Andalucía, under projects TIN2005-02516 and TIC-276, respectively.

References

1. S. Acid, L.M. de Campos, J.G. Castellano. Learning Bayesian network classifiers: searching in a space of acyclic partially directed graphs. *Machine Learning* 59(3):213-235, 2005.
2. A. Bratko, B. Filipic. Exploiting structural information for semi-structured document categorization. *Information Processing and Management* 42(3):679-694, 2006.

3. L.M. de Campos, J.F. Huete. A new approach for learning belief networks using independence criteria. *International Journal of Approximate Reasoning* 24(1):11-37, 2000.
4. D. Koller, M. Sahami. Hierarchically classifying documents using very few words. In *Proceedings of the Fourteenth International Conference on Machine Learning*, 1997.
5. L.S. Larkey, W.B. Croft. Combining classifiers in text categorization. In *SIGIR-96*, 1996.
6. D. Lewis, W. Gale. A sequential algorithm for training text classifiers. In *SIGIR-94*, 1994.
7. A. McCallum, K. Nigam. A Comparison of event models for Naive Bayes text classification.
8. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan and Kaufmann, San Mateo, 1988.
9. S. E. Robertson, K. Sparck-Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27:129-146, 1976.
10. F. Sebastiani. Machine Learning in automated text categorization. *ACM Computing Surveys*, 34:1-47, 2002.

Clustering XML documents using closed frequent subtrees- A Structure-only based approach

Sangeetha Kutty, Tien Tran, Richi Nayak, and Yuefeng Li

Faculty of Information Technology,
Queensland University of Technology, Brisbane, Australia
({s.kutty,t4.tran,r.nayak, y2.li}@qut.edu.au)

Abstract. This paper presents the experimental study conducted over the INEX 2007 Document Mining Challenge Corpus employing a frequent subtree-based incremental clustering approach. In this paper, we first generate the closed frequent subtrees using only the structure of the XML document corpus. Using the closed frequent subtrees, we generate a matrix representing closed frequent subtree distribution in documents. This matrix is then used to progressively cluster the incoming XML documents against the existing clusters. In spite of the large number of documents in INEX 2007 Wikipedia dataset, using our frequent subtree-based incremental clustering approach we have demonstrated that we could effectively cluster the documents.

1. Introduction

The rapid growth of XML since its standardization has marked its acceptance in a wide array of industries ranging from education to entertainment, business to government sectors. The major reason for its success can be attributed to its flexibility and self-describing nature in using structure to store its content. With the increasing number of XML documents there arise many issues concerning the efficient data management and retrieval. XML document clustering has been perceived as an effective solution to improve information retrieval, database indexing, data integration, improved query processing[1] and so on.

Clustering task on XML documents involves grouping XML documents based on their similarity without any prior knowledge on the taxonomy[2]. Clustering has been frequently applied to group text documents based on the similarity of its content. However, clustering XML documents presents a new challenge as it contains structural information along with content. The structure of the XML documents has a hierarchical structure and it represents the relationship between the elements at various levels.

Clustering XML documents is a challenging task. Most of the existing algorithms utilize the tree-edit distance to compute the structural similarity between each pair of documents. However, this is not a useful measure as the tree edit distance could be large for very similar trees conforming to the same schema for situations in which one of the involved tree is a larger tree and the other a small tree[3]. Recent study by [4] showed that XML document clustering using tree summaries provide high accuracy for documents conforming to the DTDs. However,[4] extracts the structural summaries of the documents and computes the tree-edit distance using their structural summaries. As it involves calculating the tree-edit distance, it could be expensive for very large dataset such as INEX wikipedia test collection with 48305 documents. This lays the ground to employ a clustering algorithm which does not utilise the expensive tree-edit distance computation.

In this paper, we utilize CFSPC technique to cluster XML documents by utilizing the closed frequent subtrees as the intermediate representation of clusters. This is achieved by computing the global similarity using the closed frequent subtrees obtained by frequent mining the XML documents. Instead of computing a pair-wise similarity between XML documents as the case in traditional clustering techniques, CFSPC computes the similarity progressively between an XML document and the existing clusters. By doing so, we could alleviate the problems of computational and memory overhead inherent in pair-wise clustering techniques. CFSPC computes the similarity of XML documents in INEX wikipedia dataset efficiently.

The assumption we have made in this paper, based on the previous research[5] is that documents having a similar structure can be grouped together. For instance, the document from a publication domain will have a different structure than a document from movie domain. Using this assumption we utilize only the hierarchical structure of the documents to group the XML documents. However, we have not included the content of the document as it incurs a huge overhead.

Rest of the paper is organised as follows: Section 2 provides the overview of CFSPC method detailing about its methodology. Section 3 covers the pre-processing of XML documents for mining and Section 4 details out the mining process which includes frequent mining and clustering. In Section 5, we present the experimental results and the discussion about it and we conclude in Section 6 by presenting our future works in XML document mining.

2. The CFSPC Method : Overview

As illustrated in Fig.1. CFSPC involves two major phases Pre-processing and Mining which in turn includes *frequent subtree mining* and *clustering*.

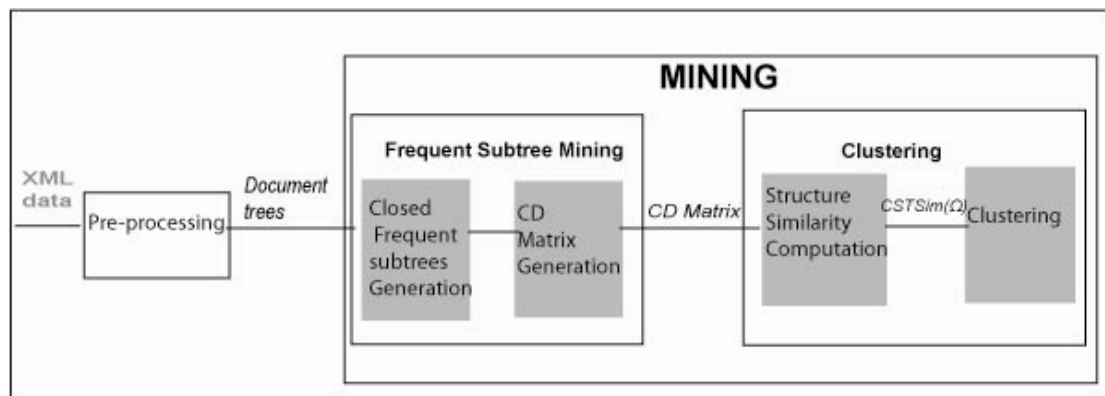


Fig. 1. The CFSPC Methodology

The pre-processing phase involves extracting the structure of a given XML document to obtain a *document tree*. Each *document tree* contains nodes which represent the tag names. As illustrated in Figure 1, there are two stages in mining namely *frequent subtree mining* and *clustering*. The frequent subtree mining stage identifies for a given support threshold the closed frequent subtrees from the *document trees*. Closed frequent subtrees are nothing but condensed representations of frequent subtrees. Using these closed frequent subtrees, a subtree-document matrix called CD matrix is generated represented by $cfs \times dt$, where cfs represents the closed frequent subtrees and dt represents the document trees in the given document tree

collection. Each cell in the CD matrix represents the presence or absence of a given closed frequent subtree against all the document trees in the given collection.

As discussed before, it is a very expensive process to cluster using the pair-wise matrix for all the INEX wikipedia documents due to its high dimension. Hence in the second phase of mining we attempt to reduce the high dimension of INEX wikipedia dataset by incrementally clustering the document trees against the existing clusters. Though incremental clustering reduces the dimensionally to a larger extent, it results in undefined number of clusters. In order to obtain the user-defined number of clusters, we utilize the pair-wise clustering algorithm CLUTO[6]. The output of the incremental clustering technique will be represented as a pair-wise similarity matrix for all the clusters generated. This matrix is provided to CLUTO[6] which in turn generates the required number of clusters.

Hence, the second phase of mining involves clustering which in turn includes two phases namely *structure similarity computation* and *clustering*. This is done by first computing the similarity between a document tree and the existing clusters using the CD matrix based on the number of common closed frequent subtree. The output of this phase in mining is the Common SubTree coefficient (Ω) between the document tree and the cluster. The second phase in mining is the clustering process in which the document tree for a given XML document is grouped into an existing cluster with which it has the maximum CSTSim or the *document tree* is assigned to a new cluster. The resulting clusters are used to compute the pair-wise similarity matrix

CFSPC is a novel algorithm utilising the frequent subtrees to cluster the XML documents. To cater for the large number of XML documents in INEX wikipedia dataset, CFSPC employs an incremental clustering method by computing the similarity using CD matrix. In the pre-processing phase, the XML document is decomposed into a tree structure with nodes representing only the tag names. Information of nodes on data types and constraints were ignored. The semantic and syntactic meanings of the tags were ignored as they did not provide any significant contribution[2, 5].

3.CFSPC Phase 1: Pre-processing

As shown in Fig. 2, the pre-processing of XML documents involves three phases namely:

1. Parsing
2. Representation
3. Duplicate branches removal

3.1. Parsing

The XML data model is a graph structure comprising of atomic and complex objects and therefore it can be modelled as a tree. Hence the XML documents in INEX wikipedia dataset is parsed and modelled as a rooted labeled ordered *document tree*. As there exists a root node in the document tree and all the nodes were labeled using the tag names, thus these document trees were *rooted* and *labeled*. Also, the left-to-right *ordering* is preserved among the child nodes of a given parent in the document tree and therefore they are ordered.

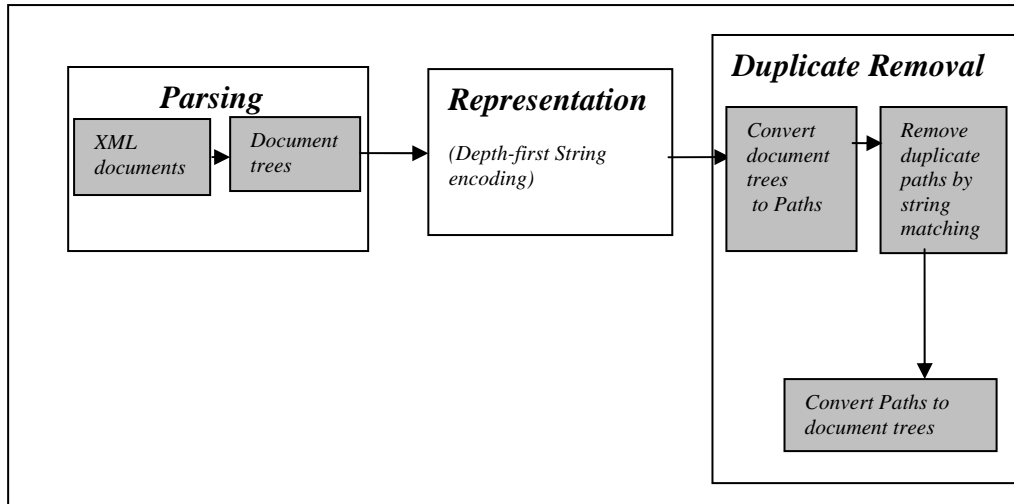


Fig. 2. The Pre-processing phase

3.2. Representation

As a next step, the document trees need to be represented in a way that is suitable for mining. A popular representation for trees is the depth-first string format and it has been chosen to represent the document trees. The *depth-first string encoding* represents the *depth-first traversal* of a document tree in a string like format where every node has a “-1” to represent backtracking and “#” to represent the end of the string encoding. For a document tree T with only one node r , the depth-first string of T is $S(T) = l_r\#$ where l is the label of the root node r . On the other hand, for multiple nodes for the document tree T , where r is the root node and the children nodes of r are r_1, \dots, r_k preserving left to right ordering. Then the depth-first string for T is $S(T) = l_r l_{r_1-1} l_{r_2-1} \dots l_{r_k-1}\#$.

3.3 Duplicate branches removal

On an analysis of the INEX Wikipedia dataset revealed that a large number of generated *document trees* contained duplicate branches. As these duplicate branches conveyed only redundant information and caused additional overhead for mining, we need to eliminate these branches. In order to remove the duplicate branches, the document tree is converted to a series of paths and then the duplicate paths are identified using string matching. The resulting duplicate paths are removed and the remaining paths are combined together to create the document trees without duplicate branches. In data mining literature, there are several works using paths; however recently researchers are focusing on using tree structures to represent XML documents. We have also chosen to use the trees to represent the XML documents as trees include the sibling information of the nodes which is not included when an XML document is represented as a series of paths.

4 CFSPC Phase 2: Mining

As mentioned before, this section includes two phases namely frequent subtree mining and clustering. We will be explaining about closed frequent subtrees and how it is extracted from the document trees. Further, how the generated closed frequent subtrees are used to cluster the document trees.

4.1 Frequent Subtree Mining

Frequent Subtree mining is applied on the XML documents from INEX Wikipedia dataset to identify closed frequent subtrees for a given user-specified support threshold. Closed frequent subtrees are condensed representations of frequent subtrees without any information loss. This phase involves generating closed frequent subtrees and utilizing them for clustering. Frequent subtree mining on XML documents can be formally defined as follows:

Problem definition for the frequent subtree mining on XML documents

For a given collection of XML documents $D = \{D_1, D_2, D_3, \dots, D_n\}$, modelled as document trees $DT = \{DT_1, DT_2, DT_3, \dots, DT_n\}$ where n represents the number of XML document or document trees. If there exists a subtree $DT' \subseteq DT_k$ preserving the parent-child relationship among the nodes as that of the document tree DT_k . $\text{Support}(DT')$ (or $\text{frequency}(DT')$) is defined as the percentage (or the number) of document trees in DT where DT' is a subtree. A subtree DT' is frequent if its support is not less than a user-defined minimum support threshold. In other words, DT' is a frequent subtree of the document trees in DT such that $(\text{frequency}(DT')/|DT|) \geq \text{min_supp}$, where min_supp is the user-given support threshold and $|DT|$ is the number of document trees in the document tree dataset DT .

Due to the large number of frequent subtrees generated at lower support thresholds, recent researchers have focused on using condensed representation without any information loss [5]. The popular condensed representation is the closed frequent subtrees which is defined as follows.

Problem definition for Closed subtree

In a given document tree dataset, $DT = \{DT_1, DT_2, DT_3, \dots, DT_n\}$, if there exists two frequent subtrees DT' and DT'' , the frequent subtree DT' is closed of DT'' iff for every $DT' \supseteq DT''$, $\text{supp}(DT') = \text{supp}(DT'')$ and there exists no superset for DT' having the same support as that of DT' . This property is called as *closure*.

In order to generate closed frequent subtrees from the pre-processed document trees, the popular CMTreeMiner[7] is utilized. In contrast to PCITMiner[8] which adopts the frequent pattern growth strategy, CMTreeMiner utilizes apriori-based approach to generate closed frequent subtrees.

Having generated the closed frequent subtrees, we need to represent in a tree-document matrix, CD matrix. CD matrix can be symbolized as $cfs \times dt$, where cfs represents the closed frequent subtrees and dt represents the document trees in the given document tree collection. Each cell in the CD matrix represents the presence or absence of a given closed frequent subtree against all the document trees in the given collection. This matrix is used to compute the similarity between the document trees for clustering.

4.2 Clustering

In order to cluster the document trees, we need to compute the structural similarity between the document trees. As discussed before, pair-wise structural similarity is an expensive operation in terms of memory usage as well as computation time for INEX wikipedia document collection. Hence, we compute the similarity of the documents in an incremental fashion. By doing so, we could group the documents into clusters and the resulting clusters are then used to compute the pair-wise similarity matrix of the

clusters. This similarity matrix of the clusters is then fed to Cluto which then groups the documents to the required number of clusters.

Clustering phase involves two sub-phases such as Structural similarity computation and the clustering of documents. We will now look into the details on how the structural similarity is computed.

4.2.1 Structural similarity Computation

Using CD matrix, we compute the structural similarity between

1. two document trees
2. a tree and a cluster

Tree-to-Tree Similarity

To begin with, there exists no cluster and hence this step is used to compute the pair-wise similarity between the first two trees to form a cluster. It is measured by first finding the common closed frequent subtrees between the two document trees.

Problem definition for tree-to-tree similarity

Let there be two document trees DT_x and DT_y , their intermediate cluster form ((ICF) d_x and d_y in the given CD matrix, is a binary vector having a length equal to the number of closed frequent subtrees in the CD matrix. For a given CD matrix, let $CFS = \{cfs_1, \dots, cfs_n\}$ be a set of closed frequent subtrees representing the rows and let $DT = \{DT_1, DT_2, DT_3, \dots, DT_n\}$ be the document trees representing the columns then the intermediate cluster form $dx = \{x_1, x_2, \dots, x_n\}$ where $x_1 \dots x_n \in \{0,1\}$ and $n = |CFS|$.

To compute the tree-to-tree similarity, using the intermediate cluster form, d_x and d_y in the CD matrix, we will firstly compute the common closed frequent subtrees between the two document trees DT_x and DT_y for a given i -th closed frequent subtree using the following Equation (1).

$$\zeta_i(d_x, d_y) = (d_x(i) \& d_y(i)=1) ? 1 : 0 \quad (1)$$

Using the intermediate cluster form, d_x and d_y in the CD matrix, we compute the possible i -th closed frequent subtrees between the two document trees DT_x and DT_y using Equation(2),

$$\alpha_i(d_x, d_y) = (d_x(i) | d_y(i)=1) ? 1 : 0 \quad (2)$$

Finally, using equation (1) and (2), we will compute the degree of similarity between the two document trees using its intermediate cluster form, d_x and d_y . The degree of similarity between the two document trees is the probability of the occurrence of a common closed frequent subtree in the possible closed frequent subtree space. In other words, it is defined as the ratio of sum of the common closed frequent subtrees over the total number of the possible closed frequent subtrees between a pair of document trees.

$$\Omega_{dx, dy} = \frac{\sum_{i=1}^j \zeta_i(d_x, d_y)}{\sum_{i=1}^j \alpha_i(d_x, d_y)} \quad \text{where } j = /CFS/ \quad (3)$$

If the tree-to-tree similarity value ($\Omega_{dx, dy}$) between the intermediate clusters, d_x and d_y of DT_x and DT_y is higher than the user-defined minimum cluster threshold (μ), then, d_x and d_y are grouped into the same cluster otherwise they are assigned to two separate clusters. If they are grouped into the same cluster then the two intermediate clusters are merged by union operation.

$$d_{clust}(i) = (d_x(i) | d_y(i)=1) ? 1 : 0 \quad (4)$$

Tree to Cluster Similarity

Once a cluster is formed, then the similarity between the incoming document tree and the existing cluster is computed using their intermediate cluster form given by d_x and d_{clust} respectively. It is computed using the Equation (3) given by

$$\zeta_i(d_x, d_{clust}) = (d_x(i) \& d_{clust}(i)=1) ? 1 : 0 \quad (5)$$

Similar to Equation (2) instead of two document tree intermediate cluster we utilize the cluster itself, we compute the possible closed frequent subtrees between the a document tree and a cluster from the CD matrix which is given by,

$$\alpha_i(d_x, d_{clust}) = (d_x(i) | d_{clust}(i)=1) ? 1 : 0 \quad (6)$$

Using equation (5) and (6), we will compute the degree of similarity between the document tree and a cluster. The degree of similarity between the document tree and a cluster is the probability of the occurrence of a common closed frequent subtree in the possible closed frequent subtree space. In other words, it is defined as the ratio of the sum of common closed frequent subtrees over the total number of possible closed frequent subtrees between a document tree and its cluster.

$$\Omega_{dx, clust} = \sum_{i=1}^j \frac{\zeta_i(d_x, d_{clust})}{\alpha_i(d_x, d_{clust})} \quad \text{where } j = /CFS/ \quad (7)$$

If the tree-to-cluster similarity value ($\Omega_{dx, clust}$) between the intermediate clusters, d_x and d_{clust} of DT_x and $clust$ is higher than the user-defined minimum cluster threshold (μ), then, d_x and d_{clust} are grouped into the $clust$ cluster otherwise d_x is assigned to a separate cluster. In situations where d_x is grouped into the $clust$ cluster then the two intermediate clusters are merged by union operation.

$$d_{clust}(i) = (d_x(i) | d_{clust}(i)=1) ? 1 : 0 \quad (8)$$

4.2.2 Clustering

The clustering of INEX Wikipedia documents includes two types of clustering namely incremental clustering and pair-wise clustering. CFSPC is a progressive or incremental clustering algorithm and hence the clusters are formed in an incremental fashion. The process starts without any cluster and when a new tree arrives, it is assigned to a new cluster. When the next tree arrives, the similarity between the current tree and the tree in the cluster is computed using the tree to tree similarity method. If the similarity value is greater than the user-defined cluster threshold (μ) then the incoming document tree is grouped into the cluster otherwise it is assigned a new cluster. If there exists new intermediate cluster form information with respect to the closed frequent subtrees in the recently clustered document tree, then the additional information is merged with the clustering information.

5. Experiment and Discussion

We implemented CFSPC using Microsoft Visual C++ 2005 and conducted experiments on the Wikipedia corpus from the INEX XML Mining Challenge 2007. As we adopt incremental clustering technique, for the given clustering threshold often we found a large number of clusters were generated. Hence, we utilise the hierarchical agglomerative clustering algorithms such as CLUTO[6] to cluster the documents to the required number of clusters. In our case, we used CLUTO to cluster the output of incremental clustering to 21 and 10 clusters.

Using the frequent mining for clustering, we had submitted 2 results one with 21 clusters and the other with 10 clusters using the cluster threshold of 0.4. The following table summarizes the results based on Micro F1 and Macro F1 measure evaluation metrics for 10 and 21 clusters with clustering threshold of 0.4.

Table 1. Submitted clustering results for INEX Wikipedia XML Mining Track 2007

Clustering Threshold	Number of Clusters	Micro F1	Macro F1
0.4	21	0.250616848783927	0.250809845156582
	10	0.250627189981547	0.249677519163818

In comparison to the other submitted results, the F1 measure of the clustering solutions obtained with CFSPC is low. In order to understand the reason for this poor performance, we analysed our experimental settings and the following experiments were conducted for varying support threshold and clustering threshold. We first conducted the experiments with varying clustering threshold to understand whether we could achieve improved performance. The experimental results for varying clustering threshold are shown in the following Table 2.

Table 2. Results from INEX Wikipedia XML Mining Track 2007 with varying clustering threshold.

Clustering Threshold	Number of Clusters	Micro F1	Macro F1
0.5	21	0.252334126901977	0.247554952427153
	10	0.250595176482766	0.249395564885681
0.3	21	0.252794799602517	0.248659811307018
	10	0.250595176482766	0.246699966671232
0.2	21	0.251485353483076	0.260469379872997
	10	0.250595176482766	0.249221108708398
0.1	21	0.250677983645585	0.257612460870218
	10	0.250595176482766	0.263320522659223

It can be seen that there is not much improvement in the Micro F1 average, however, there is an improvement for Macro F1 average for lower clustering threshold. Hence it can be attributed that there is not any significant improvement in performance for varying clustering threshold using cluster only approach. We wanted to analyse whether the number of closed frequent subtrees is an influential factor for the poor performance. Hence, we decided to use higher support threshold than the previous set of experiments. Therefore, we conducted the same experiment with 10% support threshold. We ran the experiments with varying clustering thresholds.

Also, we wanted to analyse whether the number of clusters plays a significant role. The following table summarizes the results on various numbers of clusters at 0.5 clustering threshold with 10% support threshold

Table 3. Results from INEX Wikipedia XML Mining Track 2007 for 10% Support threshold and various clustering threshold

Support Threshold	Number of Closed Frequent Subtrees	Clustering Threshold	Number of Clusters	Micro average (F1)	Macro average (F2)
10%	387	0.4	21	0.253224304	0.2688509
			10	0.251630266	0.2452841
		0.5	21	0.252561847	0.25580211
			10	0.250967809	0.24648349
		0.6	21	0.251940793	0.2480575
			10	0.250595176	0.24345386

The results from Table 3 were not satisfactory and hence we decided to compare the structure-only approaches submitted to INEX against our approach. Table 4 lists the comparison between our approach and other approaches using structure-only on INEX 2007 wikipedia dataset. It is evident from Table 4 that there is no significant difference between our approach and other approaches using only the structure of XML documents. Based on our experiments and the comparison with other approaches using structure-only, we could conclude that clustering using structural similarity between documents is not suitable for the INEX 2007 wikipedia data set.

The major reason for this situation is that the structure of the XML document plays a less important role than the content.

Table 4. Comparison of our approach against other structure-only approaches on INEX Wikipedia dataset

Approaches	Number of clusters	Micro F1	Macro F1
Hagenbuchner et.al	10	0.250595176482766	0.256914276936276
	21	0.264423972673636	0.269348265607265
Hagenbuchner	10	0.25171307318083	0.266801692533768
	21	0.257592381741021	0.252344367336805
Tien et. Al	10	0.250569664829929	0.251516558152264
	21	0.250595176482766	0.253056236274014
Our approach	10	0.250595176482766	0.263320522659223
	21	0.253224304	0.2688509

6. Conclusions and Future direction

In this paper, we presented the results of our progressive clustering algorithm for mining only the structure of XML documents in INEX 2007 Wikipedia dataset. The main aim of this study is to explore and understand the importance of structure of the XML documents over the content of XML for clustering task. In order to cluster the XML documents, we have used a frequent subtree – document matrix generated from closed frequent subtrees. Using the matrix, we have computed the similarity between XML documents and incrementally clustered them based on their similarity values. From the experimental results, it is evident that the structure plays a minor role in determining the similarity between the INEX documents.

This is the first study conducted on INEX dataset using common subtrees and hence in the future, we will aim in devising efficient similarity computation techniques to effectively cluster the XML documents. Also, as a future work, we will be focussing on including the content of XML documents to provide more meaningful cluster.

References

- [1] R. Nayak, R. Witt, and A. Tonev, "Data Mining and XML Documents," presented at International Conference on Internet Computing, 2002.
- [2] T. Tran and R. Nayak, "Evaluating the Performance of XML Document Clustering by Structure Only," in *Comparative Evaluation of XML Information Retrieval Systems*, 2007, pp. 473-484.
- [3] G. Xing, Z. Xia, and J. Guo, "Clustering XML Documents Based on Structural Similarity," in *Advances in Databases: Concepts, Systems and Applications*, 2007, pp. 905-911.
- [4] T. Dalamagas, T. Cheng, K.-J. Winkel, and T. Sellis, "A methodology for clustering XML documents by structure," *Inf. Syst.*, vol. 31, pp. 187-228, 2006.

- [5] R. Nayak, "Investigating Semantic Measures in XML Clustering," in *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*: IEEE Computer Society, 2006, pp. 1042-1045.
- [6] G. Karypis, "CLUTO - Software for Clustering High-Dimensional Datasets | Karypis Lab," 25 May 2007.
- [7] Y. Chi, S. Nijssen, R. R. Muntz, and J. N. Kok, "Frequent Subtree Mining- An Overview," in *Fundamenta Informaticae*, vol. 66: IOS Press, 2005, pp. 161-198.
- [8] S. Kutty, R. Nayak, and Y. Li, "PCITMiner- Prefix-based Closed Induced Tree Miner for finding closed induced frequent subtrees," presented at Sixth Australasian Data Mining Conference (AusDM 2007), Gold Coast, Australia, 2007.

Efficient clustering of structured documents using Graph Self-Organizing Maps

M. Hagenbuchner¹, A.C. Tsoi², A. Sperduti³, M. Kc¹

¹ University of Wollongong, Wollongong, Australia.

Email:{markus,wmc01}@uow.edu.au

² Hong Kong Baptist University, Hong Kong. Email:act@hkbu.edu.hk

³ University of Padova, Padova, Italy. Email:sperduti@math.unipd.it

Abstract. Graph Self-Organizing Maps (GSOMs) are a new concept in the processing of structured objects. These structured objects are described by graphs, e.g. acyclic directed graphs, cyclic graphs, un-directed graphs, etc. Graphs are generalizations of the more common vectors, or lists. A graph can encode relationships among structural elements of objects, or provide contextual information concerning individual data points (which may be described in vectorial form).

The GSOM itself is an extension from a number of previous attempts in extending the classic self organizing map (SOM) idea originally due to Kohonen [6]. In previous versions of such extensions, we were able to progressively study graph objects which are mainly directed acyclic graphs using what we called Self-Organizing Map for Structured Domain (SOM-SD) [1], and the Contextual Self-Organizing Map for Structured Domain (CSOM-SD) [2] where the graph objects could include cyclic graphs. However, the CSOM-SD had a nonlinear computational complexity; in most cases, this is close to quadratic. As a result, in this paper we introduce a different method, which is called Graph SOM (GSOM), in which we attempt a linear computational complexity method.

In this paper we demonstrate the efficiency and capability of the GSOM. Comparisons are made with the existing machine learning method SOM-SD [1]. SOM-SDs are capable of encoding tree-structured data and were shown to be good for tasks requiring clustering. This was demonstrated at an international competition on the clustering of XML formatted documents at which the SOM-SD produced winning performances in two consecutive years [3, 4] in the clustering category. A drawback of the SOM-SD is that it does not scale well with the size of a graph. In particular, the computational demand increases quadratically with the maximum outdegree of any node in the dataset. Moreover, the SOM-SD requires prior knowledge of the maximum outdegree, and hence, has limitations in problem domains where the maximum outdegree is not known a priori, or for which the outdegree cannot be fixed a-priori.

A recent development called GSOM is addressing these shortcomings through a modification of the underlying learning procedure. The effect is that the computational demand is reduced to a linear case and, as a side effect, allows the processing of much more general types of graphs which may feature loops, undirected links, and for which the maximum outdegree is not known a-priori. A more detailed theoretical analysis of the computational demand of the GSOM is presented in [5].

This paper applies the SOM-SD and the GSOM to a large dataset consisting of structured documents from the web. More specifically, the methods are applied

to cluster a subset of documents from Wikipedia. The documents are formatted in XML, and hence, are naturally represented as tree structures. The importance of this application is manifold:

- XML is an increasingly popular language for representing many types of electronic documents.
- An application to data mining tasks can demonstrate the advantages of the GSOM over previous machine learning methods which are capable of clustering graphs.
- The datasets considered (viz. the INEX wikipedia dataset) are a benchmark problem used at the international event INEX.

Self-Organizing Maps (SOMs) are an extension to Vector Quantization [6] in which prototype units are arranged on a n -dimensional lattice. SOMs are trained on vectorial input in an unsupervised fashion through a suitable adjustments of the best matching prototype unit and its neighbors. Training is repeated for a number of iterations where the result is a topology preserving mapping of possibly high-dimensional data onto a low dimensional, often 2-dimensional mapping space. In practise, SOMs have found a wide range of applications to problem domains requiring the clustering or projection of unlabeled high dimensional vectors.

An extension to the domain of graphs was made with the introduction of the SOM-SD. With SOM-SD it has become possible for the first time to have an unsupervised machine learning method capable of mapping graph structures onto a fixed dimensional display space. Nodes in the graph can be labeled so as to encode properties of objects which are represented by the node. One of the main advantages is that the SOM-SD is of linear computational complexity when processing graphs with a *fixed* out-degree, and hence, the SOM-SD is capable of performing tasks such as graph-matching and sub-graph matching in linear time. The SOM-SD is an extension of the standard SOM in that the network input is formed through a concatenation of the node label with the mappings of each of the node's offsprings. This implies that the SOM-SD is restricted to the processing of ordered acyclic graphs (ordered trees), and requires that the trees have a fixed (and relatively small) outdegree. The computational complexity of the SOM-SD grows quadratically with the out-degree⁴, and hence, the processing of trees with a large outdegree becomes quickly a very time consuming task. Moreover, the processing of nodes in a tree must be performed in an inverse topological order so as to ensure that the mapping of child nodes is available when processing a parent node.

The GSOM, a very recent development addresses some of the shortcomings of SOM-SD. This is achieved by concatenating the data label with the activation of the map when mapping all of a node's neighbors. Since the dimension of the map remains static, independent to the size of a training set, and to the outdegree of graphs, this implies that the GSOM's computational complexity is reduced to a linear case with respect to the outdegree of graphs. In other words, a GSOM can process graphs with a large outdegree much more efficiently than a SOM-SD. The underlying learning procedure of the GSOM is very similar to the SOM-SD,

⁴ When the out-degree is fixed then this value becomes a constant resulting in the computational complexity to remain linear. When the outdegree is variable then the computational complexity is close to quadratic.

and hence, one can expect that the clustering performances of the two methods remain very similar.

This paper gives some preliminary results⁵. Results presented here were obtained from training the SOMs for two runs each. The best result is presented in this paper. Note that under normal circumstances, a SOM would have to be run under possibly hundreds of training conditions in order to determine its peak performance. This is due to the fact that a number of training parameters need to be determined through trial and error (for any SOM training algorithm). Amongst these parameters are the dimensionality of the map, the extensions of the map, the type of neighborhood relationship between the codebook entries of the map, a learning rate, the number of training iterations, weighting measures for the data label and structural component of the inputs, and several others. A suitable choice of training parameters is essential in obtaining a well performing SOM. We were able to execute just 2 training runs to-date due to time constraints caused by software implementation problems. Hence, the quality of the results are by no means to be seen as being representative. The results of clustering the dataset into 21 clusters are as follows:

SOM-SD		GSOM	
Micro average purity	0.262457	Micro average purity	0.26885
Macro average purity	0.26159	Macro average purity	0.26635

These results are a far cry from those obtained by the following authors:

Name	Micro avg. purity	Macro avg. purity
Guangming Xing	0.62724	0.571855
Jin YAO & Nadia ZERIDA	0.51530897	0.61035

The main message we wish to convey here is that the GSOM can process datasets with a large outdegree in a more time efficient fashion. The training dataset contained 48,306 XML formatted documents. Represented as tree structures, the maximum out-degree of any graph in the training set was 1,945. This value of outdegrees would require an estimated 40 years of training time for the SOM-SD! To avoid this, and to enable the use of the SOM-SD, we pruned the graphs to a maximum outdegree of 32 by truncating nodes with a larger outdegree. This reduced the training time for the SOM-SD to a more reasonable 36 hours. In comparison, the GSOM is capable of processing the graphs without pruning in about 48 hours.

Pruning can have a negative impact on the clustering performance since the relevant information may be removed. The GSOM allows the processing of large graphs without requiring pruning, and hence, can be expected to produce performances which are at least as good as those obtained by a SOM-SD.

The SOM-SD has been proven to be good for the clustering of XML formatted documents by winning the INEX clustering competition in 2005 and 2006 respectively. Hence, we are confident that a more comprehensive set of experimental runs will reveal significantly improved results. These results are being produced and should be available for the formal proceedings in the near future.

⁵ The GSOM has been developed very recently. Time constraints and implementation issues prevented us from conducting experiments on a larger scale.

References

1. Hagenbuchner, M., Sperduti, A., Tsoi, A.: A self-organizing map for adaptive processing of structured data. *IEEE Transactions on Neural Networks* **14**(3) (May 2003) 491–505
2. Hagenbuchner, M., Sperduti, A., Tsoi, A.: Contextual self-organizing maps for structured domains. In: *Workshop on Relational Machine Learning*. (2005)
3. Hagenbuchner, M., Sperduti, A., Tsoi, A., Trentini, F., Scarselli, F., Gori, M.: Clustering xml documents using self-organizing maps for structures. In et al., N.F., ed.: *LNCS 3977, Lecture Notes in Computer Science*, Springer-Verlag Berlin Heidelberg (2006) pp. 481–496
4. KC, M., Hagenbuchner, M., Tsoi, A., Scarselli, F., Gori, M., Sperduti, S.: Xml document mining using contextual self-organizing maps for structures. In: *Lecture Notes in Computer Science*, Springer-Verlag Berlin Heidelberg (2007)
5. Hagenbuchner, M., Sperduti, A., Tsoi, A.: Self-organizing maps for cyclic and unbound graphs. In: *European symposium on Artificial Neural Networks*. (December 2007) to be submitted.
6. Kohonen, T.: *Self-Organizing Maps*. Volume 30 of *Springer Series in Information Sciences*. Springer, Berlin, Heidelberg (1995)

Multitype-Topic Models for Entity Ranking

Hitohiro Shiozaki¹ and Koji Eguchi²

¹ Graduate School of Science and Technology, Kobe University,
1-1 Rokkoudai, Nada, Kobe, 657-8501, Japan
hitohiro@cs25.scitec.kobe-u.ac.jp

² Graduate School of Engineering, Kobe University,
1-1 Rokkoudai, Nada, Kobe, 657-8501, Japan
eguchi@port.kobe-u.ac.jp

Abstract. Several topic models have been used for information retrieval. For example, cluster-based retrieval and LDA-based retrieval have been studied and have produced good results in the language modeling framework. Although, for retrieval task of structured documents that need to deal with multiple types of word tokens, we need post-processing stage (i.e. outside of the model) to distinguish word types if applying the approaches mentioned above. In this paper, we propose a multi-type topic model-based structured document model which uses Generalized-SwitchLDA model (GESwitchLDA). This model is the topic model that can deal with multiple types of word tokens. We study how to effectively apply GESwitchLDA to improve retrieval performance and show effectiveness of our method through the INEX 2007 Entity Ranking Track with Wikipedia collection which consists of words, entities, and category labels.

1 Introduction

In information retrieval (IR), several topic model-based approaches have been applied to improve the effectiveness of retrieval. For example, cluster-based retrieval, LDA-based retrieval are studied and has produced good results in the language modeling framework. Those methods were applied for unstructured documents such as newspaper articles; however, structured documents have different natures, one of which is the richer document representation using multiple types of word tokens such as words, word attributes and document labels (e.g., Wikipedia collection). To deal with such kind of documents, if using the approaches above, we need post-processing stage to distinguish such different types of word tokens. To directly handle such multiple types of word tokens, Shiozaki et al [7] proposed generalized SwitchLDA model which we call GESwitchLDA³. Using this model, we can directly deal with such kind of structured documents.

In this paper, we propose multi-type topic model based structured document model using GESwitchLDA model and investigate how to use GESwitchLDA to improve retrieval performance. We further show the effectiveness of our method for the task of

³ a.k.a W2SwitchLDA in [7].

entity ranking with Wikipedia collection which consists of words, entities, and category labels. In the *Entity Ranking Track* at the INEX 2007, each entity is represented as entity ID, text descriptions, links to other entities, and category labels. In our model, the entity ID, the text descriptions, the links to other entities, and the category labels correspond to document ID, words in the document, word attributes, and document labels, respectively. In our model, the text descriptions in the linked documents are not considered. Consideration of such link information is the future work.

2 Related Work

Statistical topic models (e.g., [4, 1, 10, 3, 8, 6]) are based on the idea that documents are mixtures of topics, where a topic is a probability distribution over words. Blei et al. [1] proposed one of the topic models called Latent Dirichlet Allocation (LDA), introducing a Dirichlet prior on multinomial distribution over topics for a document. Newman et al. [6] proposed several variations of LDA including SwitchLDA that can deal with words and entities. That motivated us to develop GESwitchLDA. To estimate the LDA model, Blei et al. used Variational Bayesian method. Instead of using the Variational Bayesian method, Griffiths et al. [3] applied the Gibbs sampling method to estimate the LDA. Teh et al. [9] applied the Collapsed Variational Bayesian method to estimate the LDA.

The cluster model, also known as the mixture of unigrams model has been well applied to IR task. In the cluster model, all documents are classified into a set of K clusters/topics. Liu and Croft [5] incorporated the cluster information into language models at smoothing stage:

$$P(w|d) = \frac{N_d}{N_d + 1} P_{ML}(w|d) + \left(\frac{1}{N_d + 1} \right) P(w|cluster) \quad (1)$$

where d is document model. Main issue of the cluster model is limitation that each document is generated from a single topic. For long documents and large collections this limitation may hurt the performance.

The statistical topic model Latent Dirichlet Allocation(LDA) [1] has also been applied to IR task. Wei and Croft [11] adopted this method to ad-hoc retrieval task by linearly combining original document model and LDA-based document model as:

$$P(w|d) = \left(\frac{N_d}{N_d + 1} P_{ML}(w|d) + \left(\frac{1}{N_d + 1} \right) P(w|coll) \right) + \left(\frac{1}{N_d + 1} \right) P_{lda}(w|d) \quad (2)$$

and significant improvements over the cluster model were reported. However, we can not directly apply this model to collections that consist with multiple types of word tokens, because LDA model does not distinguish different types of word tokens.

3 GESwitchLDA-Based Document Model

3.1 GESwitchLDA

The LDA model has brought significant improvements in ad-hoc retrieval task; however, in order to apply this model to the documents that are expressed in multiple types

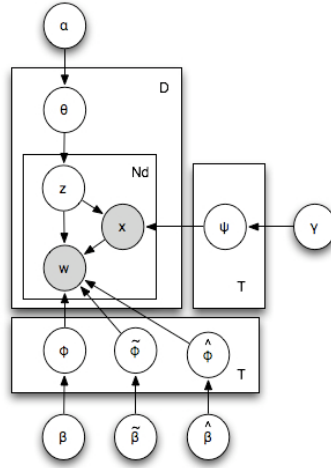


Fig. 1. GESwitchLDA when the number of word types is 3

of word tokens, we need to distinguish word tokens at post-processing stage since the LDA model does not directly distinguish word tokens. Shiozaki et al. [7] introduced a new, multi-type topic model, called *GESwitchLDA*, which can handle multiple types of word tokens (i.e. word, entity-word, and category-word). Graphical model of *GESwitchLDA* is shown in Fig.2. The variable M in Fig.2 denotes the number of types. Graphical model of *GESwitchLDA* in the case of considering three types of word tokens ($M = 3$) is shown in Fig.1. *GESwitchLDA*'s generative process ($M = 3$) is:

1. For all d documents sample $\theta_d \sim \text{Dirichlet}(\alpha)$
2. For all t topics sample $\theta_t \sim \text{Dirichlet}(\alpha)$, $\tilde{\theta}_t \sim \text{Dirichlet}(\tilde{\alpha})$, $\hat{\theta}_t \sim \text{Dirichlet}(\hat{\alpha})$ and $\gamma_t \sim \text{Dirichlet}(\gamma)$
3. For each of the N_d words w_i in document d :
 - (a) Sample a topic $z_i \sim \text{Multinomial}(\theta_d)$
 - (b) Sample a flag $x_i \sim \text{Multinomial}(\theta_{z_i})$
 - (c) If ($x_i = 0$) sample a word $w_i \sim \text{Multinomial}(\theta_{z_i})$
 - (d) If ($x_i = 1$) sample an entity-word $w_i \sim \text{Multinomial}(\tilde{\theta}_{z_i})$
 - (e) If ($x_i = 2$) sample a category-word $w_i \sim \text{Multinomial}(\hat{\theta}_{z_i})$

We used the Gibbs sampling approach as the estimation algorithm for the *GESwitchLDA*. The Gibbs sampling equations for this model are given in appendix A.3.

3.2 GESwitchLDA-based Retrieval

For IR task, the basic approach is the query likelihood model. In this model each document is ranked in order of likelihood of generating a query Q by the document model:

$$P(Q|D) = \prod_{w \in Q} P(w|D) \quad (3)$$

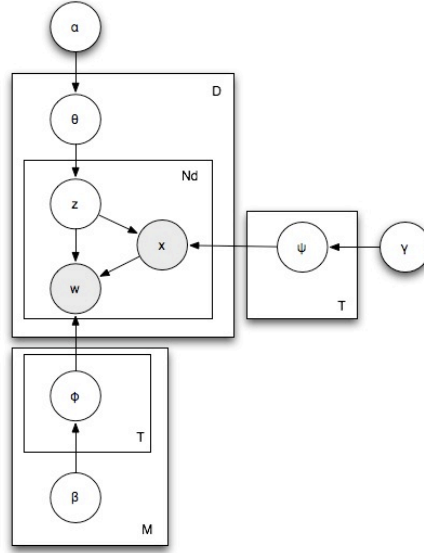


Fig. 2. GESwitchLDA

where D is a document model, Q is the query and w is a query term in Q . $P(Q|D)$ is the likelihood of generating the query terms by the document model, under the 'bag-of-words' assumption that terms are independent in the documents. We estimated $P(w|D)$ by the document model with Dirichlet smoothing [12],

$$P(w|D) = \frac{N_d}{N_d + 1} P_{ML}(w|D) + \left(1 - \frac{N_d}{N_d + 1}\right) P_{ML}(w|coll) \quad (4)$$

where $P_{ML}(w|D)$ is the maximum likelihood estimate of word w in the document D , and $P_{ML}(w|coll)$ is the maximum likelihood estimate of word w in the entire collection. α is the Dirichlet prior. In order to apply to the documents that are expressed in multiple types of word tokens, we should modify Eq.(3). Supposing the Wikipedia documents, we calculate $P(Q|D)$ as follows:

$$P(Q|D) = \prod_{w \in Q_w} P(w|D) \prod_{w_e \in Q_e} P(w_e|D) \prod_{w_\ell \in Q_\ell} P(w_\ell|D) \quad (5)$$

where Q_e is a part of Q which consists of entity-word w_e , Q_w is a part of Q which consists of category-word w , and Q_ℓ is equal to $Q \setminus (Q_e \cup Q_w)$, consisting of word w that is one of those other than w_e or w .

Similarly as when using LDA for ad-hoc retrieval, only using GESwitchLDA may be too coarse for the document representation for ad-hoc information retrieval. Therefore, we combine the original document model in the query likelihood model with the

GESwitchLDA model and construct a new GESwitchLDA-based document model. In detail, we linearly combined the models above.

$$\begin{aligned}
P(w|D) &= \left(\frac{N_d}{N_d + 1} P_{ML}(w|D) + \left(1 - \frac{N_d}{N_d + 1}\right) P_{ML}(w|coll)\right) + \\
&\quad (1 - \alpha) P_{tm}(w|D) \\
P(w_e|D) &= \left(\frac{N_{ed}}{N_{ed} + 1} P_{ML}(w_e|D) + \left(1 - \frac{N_{ed}}{N_{ed} + 1}\right) P_{ML}(w_e|coll)\right) + \\
&\quad (1 - \alpha) P_{tm}(w_e|D) \\
P(w|D) &= \left(\frac{N_d}{N_d + 1} P_{ML}(w|D) + \left(1 - \frac{N_d}{N_d + 1}\right) P_{ML}(w|coll)\right) + \\
&\quad (1 - \alpha) P_{tm}(w|D)
\end{aligned}$$

In GESwitchLDA, we can calculate the likelihood of a each type of a word in a document as following,

$$\begin{aligned}
P_{tm}(w|D) &= \sum_t^K P(w|t)P(t|D) \\
P_{tm}(w_e|D) &= \sum_t^K P(w_e|t)P(t|D) \\
P_{tm}(w|D) &= \sum_t^K P(w|t)P(t|D)
\end{aligned}$$

where t is a topic. We estimate $P(t|D)$, $P(w|t)$, $P(w_e|t)$, and $P(w|t)$ using Gibbs sampling. From Gibbs sampling we use:

$$\begin{aligned}
P(t|D) &= \frac{C_{td-i}^{TD} + \alpha}{\sum_t C_{td-i}^{TD} + T} \\
P(w|t) &= \frac{C_{wt-i}^{WT} + \alpha}{\sum_w C_{wt-i}^{WT} + W} \\
P(w_e|t) &= \frac{C_{et-i}^{ET} + \alpha}{\sum_e C_{et-i}^{ET} + E} \\
P(w|t) &= \frac{C_{t-i}^{LT} + \alpha}{\sum C_{t-i}^{LT} + L}
\end{aligned}$$

where the notation C_{pq}^{PQ} represents counts from respective count matrices, e.g. count of words in a topics, or counts of topic in a document. In this experiment, we fixed the Dirichlet prior to $\alpha = 50/T$ where T is number of topics, $\alpha = \hat{\alpha} = \alpha = 0.01$.

4 Experiments

4.1 Experimental setting

We used the 28 queries based on topic titles of the INEX 2006 Entity Ranking Track. We used the 418 stopwords included in the stop list used by *InQuery*([2]) and removed words (not entities or category labels) that occurred in less than 10 documents. We set the number of topics $T = 400$ and 800 . We carried out Gibbs sampling with a couple of different Markov chains for GESwitchLDA for each topics and averaged $P(w|t)$ and $P(t|D)$, respectively, using greedy algorithm. We set the Dirichlet prior for smoothing in the query likelihood model as $\alpha = 50$ to obtain the best results. We set $\beta = 0.5$ and $\beta = 0.6$ for $T = 400$ and $T = 800$, respectively, to obtain the best results.

4.2 Results

The best results of GESwitchLDA-based model (GES+QL), the model that uses only GESwitchLDA (GES), and query likelihood model (QL) are shown in Table 1. Comparison between GESwitchLDA-based model and the model using only GESwitchLDA with $T = 400$ or 800 are shown in Table 2.

In terms of mean average precision (MAP) value, GES+QL obtained 37%, and 77% improvements over QL, and GES, respectively. In terms of geometric mean average precision (GMAP), GES+QL obtained 117% improvement, and 326% improvement over QL, and GES, respectively.

Comparing a model with $T = 400$ and that with $T = 800$, GES with $T = 800$ obtained 72% improvement over GES with $T = 400$, in terms of MAP. GES+QL with $T = 800$ achieved slightly better improvement over GES+QL with $T = 400$. By this result, we suppose that by increasing number of topics, the characteristics of GES get close to QL.

Table 1. Best results of query likelihood model(QL), using only GESwitchLDA (GES, $T = 800$), and GESwitchLDA-based model(GES+QL, $T = 800$) (in terms of MAP)

	QL	GES($T = 800$)	GES+QL($T = 800$)
MAP	0.1799	0.1396	0.2473
GMAP	0.0469	0.0239	0.1019
R-prec	0.2028	0.1346	0.2614
Bpref	0.2389	0.2916	0.3038
MRR	0.4119	0.3053	0.4828

5 Conclusions

We proposed a new language model that combines the query likelihood model and GESwitchLDA that can deal directly with different types of word tokens. We compared

Table 2. Results of GESwitchLDA-based model(GES+QL), using only GESwitchLDA (GES) with $T = 400$ and 800

	GES($T = 400$)	GES($T = 800$)	GES+QL($T = 400$)	GES+QL($T = 800$)
MAP	0.0809	0.1396	0.2471	0.2473
GMAP	0.0177	0.0239	0.1020	0.1019
R-prec	0.0692	0.1346	0.2583	0.2614
Bpref	0.3110	0.2916	0.3110	0.3038
MRR	0.1896	0.3053	0.4653	0.4828

this model with the query likelihood model and the model that only uses GESwitchLDA and obtained significant improvement.

References

1. David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.
2. James P. Callan, W. Bruce Croft, and Stephen M. Harding. The INQUERY retrieval system. In *Proceedings of the 3rd International Conference on Database and Expert Systems Applications*, pages 78–83, Valencia, Spain, 1992.
3. Thomas L. Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101:5228–5235, 2004.
4. Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd International Conference on Research and Development in Information Retrieval*, pages 50–57, Berkeley, California, USA, 1999.
5. Xiaoyong Liu and W. Bruce Croft. Cluster-based retrieval using language models. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 186–193, New York, NY, USA, 2004. ACM.
6. David Newman, Chaitanya Chemudugunta, Padhraic Smyth, and Mark Steyvers. Statistical entity-topic models. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 680–686, New York, NY, USA, 2006. ACM Press.
7. Hitohiro Shiozaki, Koji Eguchi, and Takenao Ohkawa. Multi-entity-topic models with who-entities and where-entities. In *DMSS2007: The International Workshop on Data-Mining and Statistical Science*, 2007.
8. Mark Steyvers and Tom Griffiths. *Handbook of Latent Semantic Analysis*, chapter 21: Probabilistic Topic Models. Lawrence Erlbaum Associates, Mahwah, New Jersey, London, 2007.
9. Yee Whye Teh, David Newman, and Max Welling. A collapsed variational bayesian inference algorithm for latent dirichlet allocation. In *NIPS 2006: Neural Information Processing Systems Conference 2006*, 2006.
10. Naonori Ueda and Kazumi Saito. Parametric mixture models for multi-labeled text. In *Advances in Neural Information Processing Systems 15*, Cambridge, Massachusetts, USA, 2003. MIT Press.
11. Xing Wei and W. Bruce Croft. Lda-based document models for ad-hoc retrieval. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185, New York, NY, USA, 2006. ACM.

12. Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 334–342, New York, NY, USA, 2001. ACM.

Appendix

A. Gibbs Sampling Equations In the following equations, α and β are Dirichlet priors, and γ is a Beta prior in the case of SwitchLDA or Dirichlet prior in the case of GESwitchLDA. The notation C_{pq}^{PQ} represents counts from respective count matrices, e.g. count of words in a topics, or counts of topic in a document.

A.1 LDA

$$P(z_i = t | w_i = v, z_{-i}, w_{-i}) \propto \frac{C_{td-i}^{TD} + \alpha}{\sum_t C_{td-i}^{TD} + T} \frac{C_{wt-i}^{WT} + \beta}{\sum_w C_{wt-i}^{WT} + W}$$

A.3 GESwitchLDA when the number of word types is 3

$$\begin{aligned} P(z_i = t | w_i = v, x = 0, z_{-i}, x_{-i}, w_{-i}) &\propto \frac{C_{td-i}^{TD} + \alpha}{\sum_t C_{td-i}^{TD} + T} \frac{n_{t-i} + \gamma}{n_{t-i} + \tilde{n}_t + \hat{n}_t + 3} \frac{C_{wt-i}^{WT} + \beta}{\sum_w C_{wt-i}^{WT} + W} \\ P(z_i = t | w_i = e, x = 1, z_{-i}, x_{-i}, w_{-i}) &\propto \frac{C_{td-i}^{TD} + \alpha}{\sum_t C_{td-i}^{TD} + T} \frac{\tilde{n}_{t-i} + \gamma}{n_t + \tilde{n}_{t-i} + \hat{n}_t + 3} \frac{C_{et-i}^{ET} + \tilde{\beta}}{\sum_e C_{et-i}^{ET} + \tilde{E}} \\ P(z_i = t | w_i = o, x = 2, z_{-i}, x_{-i}, w_{-i}) &\propto \frac{C_{td-i}^{TD} + \alpha}{\sum_t C_{td-i}^{TD} + T} \frac{\hat{n}_{t-i} + \gamma}{n_t + \tilde{n}_t + \hat{n}_{t-i} + 3} \frac{C_{t-i}^{LT} + \hat{\beta}}{\sum C_{t-i}^{LT} + \hat{L}} \end{aligned}$$

An n-gram and Description-Checking based approach for Entity Ranking Track

Meenakshi Sundaram Murugesan, Dr. Saswati Mukherjee

Department of Computer Science and Engineering,
College of Engineering, Guindy,
Anna University,
Chennai, India
{msundar_26, msaswati}@yahoo.com

Abstract. Our method for this year's entity ranking track is based on two features. First one is splitting the topic into Lexical Units (LUs) and identifying the prominent n-gram(s). Second feature is to make full use of the initial description given in a wikipedia article for ranking the answers based on its similarity with the topic. In addition for the list completion task, we have explored the page for the prominent n-gram for boosting the score of a retrieved answer.

Keywords: list completion, entity ranking, n-gram checking.

1 Introduction

The Entity Ranking track in INEX 2007 consists of two tasks, namely, "Entity Ranking" and "List Completion". A collection of English Wikipedia documents (659,388 articles) is used as the corpus. Given a "category" and a "title", the task in "Entity Ranking" is to return relevant entities. In "List Completion", the task is to complete the partial list of entities, taking the "title" and a "list of example entities" as input. Here, the entities correspond to a wikipedia article.

Identifying the relevant entities can be split into two sub-tasks; first one is to form efficient queries from the topic given in the test-set, i.e., "category" and "title" for the "Entity Ranking" task, and "title" and "example entities" for the "List Completion" task. After forming queries and retrieving relevant documents using a search engine, the second task is filtering and ranking such documents based on their similarities/dissimilarities with the formed query. Query formation needs to give importance to n-gram strategy since Named Entities (NEs) usually have great impact on the relevant documents retrieved. One vital aspect to be considered is the nature and structure of the corpus used. Unstructured corpus, which consists of plain text, does not give any clue about the contents, whereas, from the semi-structured corpus, both the structure and content can be exploited. Wikipedia corpus, which is created and used by INEX, is organized in such a way that the articles contain initial descriptions followed by several sections, and references, which can be analyzed and used.

2 Proposed Approach

A wikipedia article, we observe, consists of a name followed by an initial description followed by a possible set of sections. While the name is one that can be used to succinctly describe the article, the initial description in the article gives a concise overview of what the article is about.

The method that we have adopted for the Entity Ranking Track in INEX 2007, is focused on forming effective queries using meaningful n-grams and checking the initial descriptions to find it's relevance with the given topic.

2.1 Query Processing

Methods have already been explored to form queries from the list questions given by TREC, by tagging with part-of-speech (POS) information, and identifying the focus of the question [1]. This includes using the Named-Entity information to split the question into unigrams and Named Entities [2].

In our method, our focus is to split the topic into meaningful lexical units (LUs). Existing Named Entity recognizers have the drawback of not being able to identify such entities with desired level of accuracy. To counter this problem, we wanted to make use of the key information available from the large collection of 659,388 articles available in wikipedia corpus. Each of the wikipedia articles in the corpus has a name that corresponds to its contents. For example, a bi-gram such as "Bob Dylan" has a corresponding wikipedia article with that name.

First we checked the n-grams in the given topic, against the list of names of all articles in the corpus, and split the queries into n-grams of variable size, which we call as Lexical Units (LUs). Next, to identify the prominent n-gram(s) in the topic, we tagged the topic with part-of-speech information using Monty tagger, which is based on Brill's tagger.

To form queries we consider three cases; the first case occurs when we have one or more proper nouns; the second case, when we obtain a prominent n-gram, although this n-gram may not be a proper noun; the third case happens when the topic contains no prominent n-grams. For example, the following is a topic (Q61) in the test-set.

Q61 car manufacturers of Germany
This is tagged as,
Car/NN manufacturers/NNS of/IN Germany/NNP

Here we have identified Germany as a prominent n-gram.

The second case is topics without proper nouns such as the one shown below (Q41). To find the prominent n-gram, we check the wikipedia article for each Lexical Unit (LU), for the presence of other Lexical Units (LUs) in the topic.

Q41 online book seller

In this case, all LUs are unigrams. The Wikipedia article for “book” contains the unigrams, “online” and “seller”, and hence “book” is identified as a prominent n-gram. If none of the LUs in the topic are found as prominent n-gram, then all n-grams are considered as equally important, which is the last case.

Next level of importance is given to common nouns (NNS) in the topic that are left after the prominent n-grams are identified and removed. If there is more than one common noun, we expect at least one among them to be present in the relevant article.

2.2 Secondary Terms Retrieval

For “Entity-Ranking” task, to retrieve the secondary terms, we used the given category as query and retrieved top 100 documents, from which we retrieved top 5 high TF (term-frequency) terms, which should have a minimum DF (document frequency) of 10. This along with the rest of the terms in the topic (other than prominent n-grams and common nouns) forms the set of secondary terms. For “List Completion” task, instead of using the “category” information, the terms that have appeared in all the list answers given, with high TF (top 5 terms) are taken as secondary terms.

3 Answer Retrieval and Ranking

We indexed the corpus using Lucene, the use of which as an efficient retrieval engine is demonstrated in several Question Answering systems [5]. We retrieved a maximum of 500 documents if that much hits (relevant documents) are available for the query we have formed. We observe that, each wikipedia article, starts with an initial description, which gives a concise overview of what the article is about. We retrieved such descriptions from these retrieved documents, and checked them against the LUs in the topic. While ranking, we expect the prominent n-gram to be present in this initial description - failing which that article is not considered to be relevant. The titles of such retrieved wikipedia articles are identified as expected answers.

In addition, for the “List Completion” task, if the wikipedia article for the prominent n-gram contains any of the given list answers, we check the retrieved answers against that article, and give higher rank to such answers. For example, if the page for “Friedrich Nietzsche” contains any of the listed answers, we check for the retrieved answers against this page, and if found, rank them at the top.

For answers, which have equal ranks, we use the Lucene ranking to distinguish them. Thus, the method we have applied is based on identifying and using prominent n-grams and checking them against the description given in the wikipedia article.

4 Evaluation

To demonstrate the effectiveness of our method, we have given below the top 10 answers retrieved by our system for a topic taken from the test-set (Q33).

Q33 Books written by Friedrich Nietzsche

WP2160910: The Gay Science
WP575668: Beyond Good and Evil
WP633739: Beyond Good & Evil (video game)
WP161594: Übermensch
WP202777: The Birth of Tragedy
WP897486: God is dead
WP901686: The Twilight of the Idols
WP898990: The Rebel
WP1952471: Also sprach Zarathustra (Strauss)
WP901627: Human, All Too Human

Here, whatever follows WP is the wikipedia article ID, which is followed by the name of that article, i.e., the retrieved answer.

5 Conclusion

The method that we have adopted relies on initial descriptions given in a wikipedia article. Although, it proves to be effective in most cases, there are some drawbacks in this method. The method we have adopted finds it hard to find relevant answers in few cases. For example if the expected answers are country names, our checking fails to find relevant answers in most cases. We are exploring possible ways to overcome this problem without modifying our basic approach.

References

1. Chen, J., Diekema, A., Taffett, M.D., McCracken, N., Ozgencil, N.E., Yilmazel, O., and Liddy, E.D. Question Answering: CNLP at the TREC 10 Question Answering Track. In Proceedings of the 10th Text REtrieval Conference, 2002.
2. Hui Yang, Tat-Seng Chua, Web-based list question answering. In Proceedings of the 20th international conference on Computational Linguistics, 2004.
3. N. Craswell, A.P. de Vries, I. Soboroff, Overview of the TREC 2005 Enterprise Track. In proceedings of TREC 2005.
4. Ellen M. Voorhees, Overview of the TREC 2001 Question Answering Track. In proceedings of TREC 2001.
5. Mark A. Greenwood, Mark Stevenson and Robert Gaizauskas. The University of Sheffield's TREC 2006 Q&A Experiments. In Proceedings of the 15th Text REtrieval Conference, 2006.

Structured Document Retrieval, Multimedia Retrieval, and Entity Ranking Using PF/Tijah

Theodora Tsirikika¹, Pavel Serdyukov², Henning Rode², Thijs Westerveld^{3*},
Robin Aly², Djoerd Hiemstra², and Arjen P. de Vries¹

¹ CWI, Amsterdam, The Netherlands

² University of Twente, Enschede, The Netherlands

³ Teezir Search Solutions, Ede, The Netherlands

Abstract. CWI and University of Twente used PF/Tijah, a flexible XML retrieval system, to evaluate structured document retrieval, multimedia retrieval, and entity ranking tasks in the context of INEX 2007. For the retrieval of textual and multimedia elements in the Wikipedia data, we investigated various length priors and found that biasing towards longer elements than the ones retrieved by our language modelling approach can be useful. For retrieving images in isolation, we found that their associated text is a very good source of evidence in the Wikipedia collection. For the entity ranking task we used random walks to model multi-step relevance propagation from the articles describing entities to all related entities and further.

1 Introduction

In INEX 2007, CWI and the University of Twente participated in the Ad Hoc, Multimedia, and Entity Ranking tracks. In all three tracks, we used PF/Tijah [5], a flexible system for retrieval from structured document collections, that integrates NEXI-based IR functionality and full XQuery support.

In the Ad Hoc track, we participated in all three subtasks for element retrieval, and mainly investigated the effect of various length priors within a language modelling framework. We also took part in both Multimedia tasks, where we examined the value of textual and context-based evidence without considering any of the available visual evidence. For Entity Ranking, we exploit the associations between entities; entities are ranked by constructing a query-dependent entity link graph and applying relevance propagation schemes modelled by random walks.

The remainder of this paper is organised as follows. Section 2 introduces PF/Tijah. Next, Sections 3, 4, and 5 respectively discuss our participation in each of the Ad Hoc, Multimedia, and Entity Ranking tracks. Section 6 concludes this paper by highlighting our main contributions.

* This work was carried out when the author was at CWI, Amsterdam, The Netherlands

2 The PF/Tijah System

PF/Tijah, a research project run by the University of Twente, aims at creating a flexible environment for setting up search systems. It achieves that by including out-of-the-box solutions for common retrieval tasks, such as index creation (that also supports stemming and stopword removal) and retrieval in response to structured queries (where the ranking can be generated according to any of several retrieval models). Moreover, it maintains its versatility by being open to adaptations and extensions.

PF/Tijah is part of the open source release of MonetDB/XQuery (available at <http://www.sourceforge.net/projects/monetdb/>), which is being developed in cooperation with CWI, Amsterdam and the University of München. PF/Tijah combines database and information retrieval technologies by integrating the PathFinder (PF) XQuery compiler [1] with the Tijah XML information retrieval system [11]. This provides PF/Tijah with a number of unique features that distinguish it from most other open source information retrieval systems:

- It supports retrieval of arbitrary parts of XML documents, without requiring a definition at indexing time of what constitutes a document (or document field). A query can simply ask for any XML tag-name as the unit of retrieval without the need to re-index the collection.
- It allows complex scoring and ranking of the retrieved results by directly supporting the NEXI query language.
- It embeds NEXI queries as functions in the XQuery language, leading to ad hoc result presentation by means of its query language.
- It supports text search combined with traditional database querying.

The above characteristics also make PF/Tijah particularly suited for environments like INEX, where search systems need to handle highly structured XML collections with heterogenous content. Information on PF/Tijah, including usage examples, can be found at: <http://dbappl.cs.utwente.nl/pftijah/>.

3 Ad Hoc Track

The granularity at which to return information to the user has always been an important aspect of the INEX benchmarks. The element and passage retrieval tasks aim to study ways of pointing users to the most specific relevant parts of documents. Various characteristics of the document parts or elements are of potential value in identifying the most relevant retrieval bits. Obviously the element content is a valuable indicator, but also more superficial features like the element type, the structural relation to other elements and the depth of the XML tree may play a role.

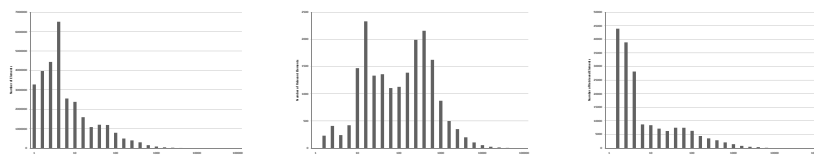
We studied the influence of a very basic feature: element size. Size priors have played an important role in information retrieval [14, 4, 8]. Kamps et al. [6] studied length normalization in the context of XML retrieval and INEX collections, and found that the size distribution of relevant elements differed

significantly from the general size distribution of elements. Emphasizing longer elements by introducing, linear, quadratic or even cubic length priors improved the retrieval results significantly on the IEEE collection.

For this paper, we performed a similar study on the Wikipedia collection. We studied the size distributions of elements in the Wikipedia collection, in the relevant elements for the INEX 2006 Focused task, and in the elements retrieved by a baseline language model run. The aim of this analysis was to experiment with different length priors on the INEX 2007 tasks.

3.1 Analysis of Element Size

We assume the distribution of element size is different for relevant and non-relevant elements. Moreover, we expect these distributions in the Wikipedia collection to be different from the IEEE collection. We studied last year's data to gain some insight in the matter. Figures 1(a) and (b) show the distribution of element sizes in the Wikipedia collection as a whole and in the relevant elements, respectively. While the collection contains many small elements, these are rarely relevant. If we would not pay attention to element length and just use a retrieval model that does not have a bias for elements of any size, we would retrieve too many small elements. Simply giving a bias towards longer elements could already improve retrieval results.



(a) XML element sizes (b) Relevant element sizes (c) Retrieved element sizes

Fig. 1. Size distribution of elements in Wikipedia collection, elements relevant to 2006 topics and elements retrieved for 2006 topics

In reality, our retrieval model, based on the language modeling approach to information retrieval, does not retrieve elements of all sizes uniformly. The model interpolates foreground and background probabilities in a standard manner and computes the foreground probability based on the relative frequency of query terms in documents. This has the effect that short elements containing query terms get a high score. Figure 1(c) shows the distribution of elements that we retrieve using this language modeling approach, when we do not compensate for document length. Clearly, we retrieve a lot of small elements.

One way of compensating for this emphasis on small elements, that nicely fits in the language modeling approach, is to incorporate document priors: a priori probabilities of relevance based on document characteristics that are independent

of a query. The probability of a document D given a query Q can be factored as the probability of drawing the query from the document ($P(Q|D)$, the documents language model) and the prior probability of the document $P(D)$ (the prior probability of the query $P(Q)$ does not influence the ranking and can be ignored):

$$P(D|Q) = \frac{P(Q|D)P(D)}{P(Q)} \propto P(Q|D)P(D) \quad (1)$$

We use the INEX 2006 results to estimate a prior based on document size. The probability of relevance given a certain size can be estimated by comparing the distributions of relevant elements to those of the collection: $P_{size}(D) = P(relevant|size(D))$. To see which elements we should emphasize given the use of our language model, we also compute a prior based on comparing relevant to retrieved elements: $P_{size}(D) = P(relevant|size(D), retrieved(D))$. Figure 2 visualises these priors.



(a) Size prior estimated from the fraction of the number of relevant and collection elements (b) Size prior estimated from the fraction of the number of relevant and retrieved elements

Fig. 2. Size priors estimated from INEX 2006 statistics for relevant, collection and retrieved elements

Judging from the comparison between relevant and collection items (Figure 2(a)), a quadratic prior as found by Kamps et al. [6] seems appropriate, but looking at what is actually retrieved by a language modeling approach (Figure 2(b)) it seems the prior should have a big peak around 1000 terms and a smaller peak around 10 terms. A mixture model seems more appropriate.

3.2 Experimental Results

In our runs for INEX 2007, our aim was to experiment with different priors based on our findings on the analysis of 2006 results. Unfortunately, at the time of run submission, we did not find the correct priors as shown in Figure 2. Instead, due to a mistake in our analysis, we found size priors that were linear and normally distributed in the log of the element size. Therefore, we submitted runs with priors that are linear in the log of the element size (`star_logLP`) and runs with a normally distributed log size prior (`star_lognormal`). We plan to

redo the experiments with priors that match the quadratic and Gaussian mixture distributions as shown in Figure 2.

Each of the prior runs is submitted for the Focused task, and, in addition, filtered for the Relevant in Context task (runIDs with `_Ric` affix); for Relevant in Context, we grouped the top 1500 results retrieved by a baseline run by article and ordered these articles based on their top scoring element. We also submitted an article-only baseline run, i.e., a run in which we only return full articles. This article run was submitted to both the Focused (`article`) and Best in Context tasks (`article_BiC`). Tables 1-3 show the results for these official submissions. Further experimentation is needed to show if the newly found quadratic and mixed priors would yield better results.

Table 1. Results for the CWI/UTwente submissions to the Ad Hoc Focused task. The table shows the rank of the run among official submissions, the run identifier and the interpolated precision at 0.01 recall.

rank	runID	iP[0.01]
57	star_logLP	0.2878
62	article	0.2686
78	star_lognormal	0.0483

Table 2. Results for the CWI/UTwente submissions to the Ad Hoc Relevant in Context task. The table shows the rank of the run among official submissions, the run identifier and Mean Average generalized Precision.

rank	runID	MAGP
18	star_logLP_RinC	0.0784
63	star_lognormal_RinC	0.0069

Table 3. Results for the CWI/UTwente submissions to the Ad Hoc Best in Context task. The table shows the rank of the run among official submissions, the run identifier and the Mean Average generalized Precision.

rank	runID	MAGP
30	articleBic	0.1338

4 Multimedia Track

CWI/Utente participated in both MMfragments and MMimages tasks of the Multimedia track. Our overall aim is to investigate the value of textual and

contextual evidence given information needs (and queries) with clear multimedia character. As a result, we only submitted text-based runs without taking into account any of the provided visual evidence. Below, we discuss our approaches and experimental results for both tasks.

4.1 MMfragments task

For MMfragments, the objective is to find relevant XML fragments (i.e., elements or passages) in the (Ad Hoc) Wikipedia XML collection given a multimedia information need. MMfragments is actually very similar to the Ad Hoc retrieval task, with the difference being that MMfragments has a multimedia character and, therefore, requires the retrieved fragments to contain at least one relevant image, together with relevant text. Furthermore, additional visual evidence, such as concepts and image similarity examples, can be provided as part of a topic. Given these similarities, MMfragments was run in conjunction with the Ad Hoc track, with MMfragments topics forming a subset of the Ad Hoc ones. In addition, MMfragments contains the same three subtasks as the Ad Hoc task. This gives us the opportunity to compare the effectiveness of MMfragments runs (i.e., runs with a clear multimedia character) against Ad Hoc runs on the same topic subset.

We only participated in the Focused MMfragments task. Given the similarities with the Ad Hoc task, we decided to (i) use only the title field of the topics, (ii) apply the same three element runs as the ones submitted for the Focused Ad Hoc task (i.e., `article`, `star_logLP` and `star_lognormal`), and (iii) realise the multimedia character by filtering our results, so that we only return fragments that contain at least one image. Not all `<image>` tags in the (Ad Hoc) Wikipedia XML collection correspond to images that are actually part of the Wikipedia image XML collection; images that are not part of this collection will not be visible to users during assessments. Therefore, we also removed all results that contained references to images that are not in the Wikipedia image XML collection. This way, we made sure all our returned fragments contain at least one *visible* image.

The results of our official submissions are presented in Table 4. Given the mistake in our earlier computation of the priors for the Ad Hoc runs, further experimentation is needed to determine whether other priors (e.g., quadratic and mixed priors) would lead to better performance. Finally, a direct comparison against our Ad Hoc runs on the MMfragments topic subset will give us more insight on the value of our filtering approach in the context of topics with clear multimedia character.

4.2 MMimages task

For MMimages, the aim is to retrieve documents (images + their metadata) from the Wikipedia image XML collection. Similarly to the Ad Hoc and MMfragments tasks, our submitted runs are based on the language modelling approach. Each image is represented either by its textual metadata in the Wikipedia image

Table 4. Results for the CWI/UTwente submissions to the MMfragments Focused task. The table shows the rank of the run among official submissions, the run identifier and the interpolated precision at 0.01 recall.

rank	runID	iP[0.01]
3	article_MM	0.2301
4	star_loglength_MM	0.1909
5	star_lognormal_MM	0.0420

XML collection, or by its textual context when that image appears as part of a document in the (Ad Hoc) Wikipedia XML collection.

To be more specific, we submitted the following three runs:

title_MMim Create a stemmed index using the metadata accompanying the images in the Wikipedia image XML collection, and perform an article run using only the topics' title field: `//article[about(.,$title)]`.

article_MMim Rank the articles in the (Ad Hoc) Wikipedia XML collection using each topic's title field and retrieve the images that these articles contain. Filter the results, so that only images that are part of the Wikipedia image XML collection are returned.

figure_MMim Rank the figures with captions in the (Ad Hoc) Wikipedia XML collection using each topic's title field (`//figure[about(.,$title)]`) and return the images of these figures (ensuring that these images are part of the Wikipedia image XML collection).

Table 5 presents the Mean Average Precision (MAP) of these runs, whereas Figure 3 compares them against all the runs submitted to the MMimages task. Our experimental results indicate that these text-based runs give a highly competitive performance on the MMimages task.

Table 5. Results for the CWI/UTwente submissions to the MMimages task. The table shows the rank of the run among official submissions, the run identifier and Mean Average Precision.

rank	runID	MAP
1	title_MMim	0.2998
3	article_MMim	0.2240
5	figure_MMim	0.1551

5 Entity Ranking by Relevance Propagation

We also participated in this year's entity ranking task. The queries here ask for a ranked list of entities, e.g. for movies, flags, or diseases. Entities are usually identified by their name and type. An entity of type movie would be identified by

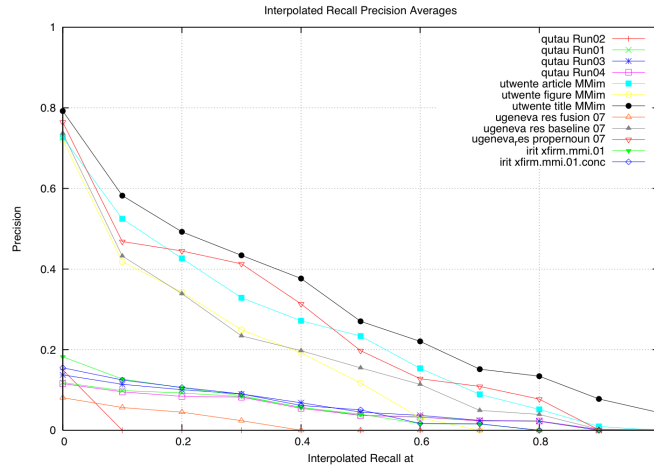


Fig. 3. MMimages results: cwi/utwente runs compared to the competition.

its title. In general, the entity ranking task differs clearly from document ranking since it requires to estimate the relevance of items that do not have text content [12, 15]. In that case, the ranking can only be done by propagating the relevance from retrieved text fragments to their contained entities. Using Wikipedia as the corpus for entity ranking experiments, the setting changes slightly. In order to use the existing mark-up of the corpus – instead of employing taggers for named entity recognition – only those entities were considered that have their own Wikipedia article. An entity is contained in an article when it is linked by that article. In consequence, the distinction of articles and entities is abandoned here. Since entities have their own article, they can also be ranked directly by their content.

The type of an entity is defined in the context of Wikipedia by the categories assigned to the entity’s article. An entity can thus have several types. Furthermore, Wikipedia categories are hierarchically organized. We can thus assume that an entity does not only belong the categories assigned to it, but also to ancestor categories. However, Wikipedia’s category hierarchy does not form a strict tree, and thus moving to far away from the original categories can lead to unexpected type assignments.

Our approach entity ranking approach can be summarized by the following processing steps:

1. initial retrieval of articles,
2. building of an entity graph,
3. relevance propagation within the graph,
4. filtering articles by the requested type.

The notion *entity graph* stands here for a query-dependent link graph, consisting of all articles (or entities) returned by the initial retrieval as vertices and the link-structure among them forming the edges. Links to other articles not returned in the initial ranking are not considered in the entity graph. The entity graph can later be used for the propagation of relevance to neighboring nodes. Starting with web retrieval [10, 7, 13], graph based ranking techniques have been used recently in several fields of IR [3, 9, 2].

5.1 Baseline: Entity Retrieval by Description Ranking

The most simple and obvious method of entity retrieval could be the ranking of their textual descriptions with some classic document retrieval method. However, due to several reasons this approach may produce unsatisfactory results. First, many entities have too short or empty descriptions, especially those that appear in novel evolving domains and just became known. Thus, many entities get the score close to zero and do not appear in the top. Second, many entities are described by showing the associations with other entities and in terms of other entities. This means that query terms have lesser chance to appear in the content of a relevant description, since some concepts mentioned in its text are not explained because explanations can be found in their own descriptions. In our experiments we rank Wikipedia articles representing entities using a language-model based retrieval method:

$$P(Q|e) = \prod_{t \in Q} P(t|e), \quad (2)$$

$$P(t|e) = (1 - \lambda_C) \frac{tf(t, e)}{|e|} + \lambda_C \frac{\sum_{e'} tf(t, e')}{\sum_{e'} |e'|} \quad (3)$$

where $tf(q, e)$ is a term frequency of q in the entity description e , $|e|$ is the description length and λ_C is a Jelinek-Mercer smoothing parameter - the probability of a term to be generated from the global language model. In all our experiments it is set to 0.8, what is standard in retrieval tasks.

5.2 Entity Retrieval Based on K-Step Random Walk

In our follow-up methods we decided that relevance propagation from initially retrieved entities to the related ones is important. We imagine and model the process in which the user, after seeing initial list of retrieved entities:

- selects one document and reads its description,
- follows links connecting entities and reads descriptions of related entities.

Since we consider this random walk as finite, we assume that at some step a user finds the relevant entity and stops the search process. So, we iteratively calculate the probability that a random surfer will end up with a certain entity after K steps of walk started at one of the initially ranked entity. In order to emphasize the importance of entities to be in proximity to the most relevant ones according to the initial ranking, we consider that both (1) the probability

to start the walk from certain entity and (2) the probability to stay at the entity node are equal to the probability of relevance of its description.

$$P_0(e) = P(Q|e) \quad (4)$$

$$P_i(e) = P(Q|e)P_{i-1}(e) + \sum_{e' \rightarrow e} (1 - P(Q|e'))P(e|e')P_{i-1}(e'), \quad (5)$$

The probabilities $P(e|e')$ are uniformly distributed among links outgoing from the same entity. Finally, we rank entities by their $P_K(e)$.

Linear Combination of Step Probabilities It is also possible to estimate entity relevance using several finite walks of different lengths at once. In the following modification of the above-described method, we rank entities considering a weighted sum of probabilities to appear in the entity node at different steps:

$$P(e) = \mu_0 P_0(e) + (1 - \mu_0) \sum_{i=1}^K \mu_i P_i(e) \quad (6)$$

In our experiments we set μ_0 to 0.5 and distribute $\mu_1 \dots \mu_K$ uniformly.

5.3 Entity Retrieval Based on Infinite Random Walk

In our second approach, we assume that the walk in search for relevant entities consists of countless number of steps. The stationary probability of ending up in a certain entity is considered to be proportional to its relevance. Since the stationary distribution of a described discrete Markov process does not depend on the initial distribution over entities, so the relevance flow becomes unfocused. The probability to appear in a certain entity node becomes dependent only on its centrality, but not on its closeness to the sources of relevance. In order to solve this issue we introduce regular jumps to entity nodes from any node of the entity graph after which the walk restarts and the user follows inter-entity links again. We consider that the probability of jumping to the specific entity equals to the probability of relevance of its description. This makes a random walker visit entities which are situated closer to the initially highly ranked ones more often during normal walk steps. The following formula is used for iterations until convergence:

$$P_i(e) = \lambda_J P(Q|e) + (1 - \lambda_J) \sum_{e' \rightarrow e} P(e|e')P_{i-1}(e') \quad (7)$$

λ_J is the probability that at any step the user decides to make a jump and not to follow outgoing links anymore. The described discrete Markov process is stochastic and irreducible, since each entity is reachable due to introduced jumps, and hence has a stationary distribution. Consequently, we rank entities by their stationary probabilities $P_\infty(e)$

5.4 Experiments

We trained our models using those 28 queries from Ad-Hoc XML Retrieval task which are suitable also for the entity ranking task. All our algorithms start from retrieval of articles from the collection using a language modeling based approach to IR for scoring documents. Further we extract entities mentioned in these articles and build entity graphs. For the initial article retrieval as well as for the graph generation the PF/Tijah retrieval system was employed. For this experiment, we generated XQueries that directly produce entity graphs in *graphml* format given a title-only query. We tuned our parameters by maximization of the MAP measure and for 100 initially retrieved articles.

The training of the following methods is discussed further:

- **Baseline**: the baseline method ranking entities by the relevance of their Wikipedia-articles (see Equations 2, 3),
- **K-Step RW**: the K-step Random Walk method using multi-step relevance propagation with K steps (see Equations 4, 5),
- **K-Step RWLin**: the K-step Random Walk method using linear combination of entity relevance probabilities at different steps up to K (see Equation 6),
- **IRW**: the Infinite Random Walk method ranking entities by probabilities to reach them in infinity during non-stop walk (see Equation 7).

For the Entity Retrieval task we had a query and the list of entity categories as input. However, according to the track guidelines and our own intuition, relevant entities could be found out of the scope of given categories. Preliminary experiments have shown that using parent categories of any level spoiled the performance of the baseline method. However, it was very important to include child categories up to 3rd level as for the Baseline method, as for our methods with tuned parameters (see Figure 4). This probably means that queries were created with an assumption that given categories should be greatest common super-types for the relevant entities. It must be mentioned that we used entities of all categories for the graph construction and relevance propagation and filtered out entities using list of allowed categories only at the stage of result output.

In all methods except the Baseline we had to tune one specific parameter. For the K-step RW and K-step RWLin methods we experimented with the number of walking steps. As we see in Figure 5 both methods reach their maximum performance after making already only 3 steps. K-step RW Lin method seems to be more robust to the parameter setup. It probably happens because it smooths the probability to appear in the certain entity after K steps with probabilities of visiting it earlier. The rapid decrease of performance for even steps for K-step RW method can be explained in the following way. A lot of relevant entities are only mentioned in the top ranked entity descriptions and do not have their own descriptions in this top, due to their low relevance probability or due to their absence in the collection. The relevance probability of these “outsider” entities entirely depends on the relevance of related entities, which are not relevant entities themselves (for example, do not match the requested entity type), but tell a lot about the ranked entity. So, all “outsider” entities have direct (backward)

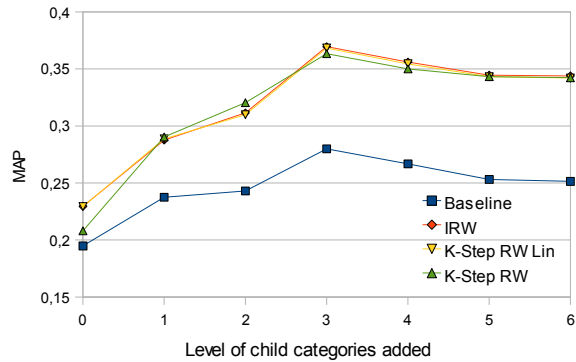


Fig. 4. MAP performance of all methods for different levels of child categories added

links only to the entities with descriptions in the top and since we always start walking only from the latter entities, the probability to appear in “outsider” entities at every even step is close to zero.

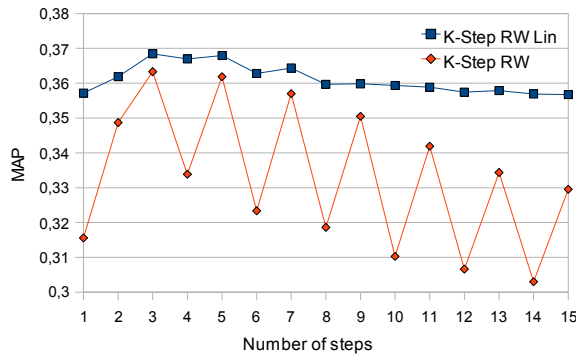


Fig. 5. MAP performance for two methods and different numbers of steps

We also experimented with the probability to restart the walk from initially ranked entities for the IRW method. According to results shown in Figure 6, values between 0.3 and 0.5 seem to be optimal. This actually means that making only 2-3 steps (before the next restart) is the best strategy what is also the case for the finite random walk methods.

To sum the things up, our experiments with the training data showed that all our three methods significantly outperform the **Baseline** method. However, the **K-Step RW** method produced a bit worse results than the other two.

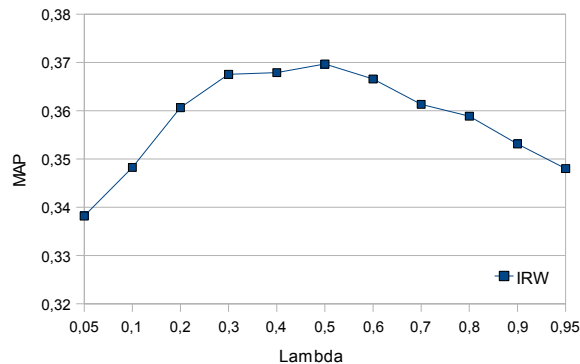


Fig. 6. MAP performance of IRW method for different values of jumping probability

6 Conclusions

This is the second year that CWI and University of Twente used PF/Tijah in INEX. The flexibility of this system is clearly demonstrated through its application in INEX tracks as diverse as ad hoc structured document retrieval, retrieval of multimedia documents and document fragments, and entity ranking.

The unigram language modelling approach we have previously applied in Ad Hoc element retrieval tasks retrieves short elements. Given that our analysis of last year's results indicates that the relevant elements tend to be longer than the ones our approach retrieves, the incorporation of length priors would be beneficial. For the Focused subtask, further experimentation is needed to determine whether the priors indicated by our recent analysis would yield better performance, whereas for the Best in Context and Relevant in Context subtasks, we need to examine in more detail our filtering strategies.

Our text only approach to Multimedia retrieval was very successful on the MMimages task. Further experimentation on the MMfragments task would reveal whether more appropriate filtering techniques or alternative priors would improve our results.

The experiments with our approaches for entity ranking demonstrated the advantage of multi-step relevance propagation from textual descriptions to related entities over the simple ranking of entity textual descriptions. The further improvement seems especially challenging because all our three methods showed quite similar effectiveness.

References

1. P. Boncz, T. Grust, M. van Keulen, S. Manegold, J. Rittinger, and J. Teubner. MonetDB/XQuery: a fast XQuery processor powered by a relational engine. In *SIGMOD '06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 479–490, New York, NY, USA, 2006. ACM Press.

2. P.-A. Chirita, J. Diederich, and W. Nejdl. Mailrank: using ranking for spam detection. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 373–380, New York, NY, USA, 2005. ACM Press.
3. N. Craswell and M. Szummer. Random walks on the click graph. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 239–246, New York, NY, USA, 2007. ACM.
4. D. Hiemstra. A linguistically motivated probabilistic model of information retrieval. In C. Nicolaou and C. Stephanidis, editors, *Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries (ECDL)*, volume 513 of *Lecture Notes in Computer Science*, pages 569–584. Springer-Verlag, 1998.
5. D. Hiemstra, H. Rode, R. van Os, and J. Flokstra. Pftijah: text search in an XML databases system. In *Proceedings of the 2nd International Workshop on Open Source Information Retrieval (OSIR)*, 2006.
6. J. Kamps, M. de Rijke, and B. Sigurbjörnsson. Length normalization in xml retrieval. In *SIGIR '04: Proceedings of the 27th annual international conference on Research and development in information retrieval*, pages 80–87. ACM Press, 2004.
7. J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.
8. W. Kraaij, T. Westerveld, and D. Hiemstra. The importance of prior probabilities for entry page search. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 27–34. ACM Press, 2002.
9. A. Kritikopoulos, M. Sideri, and I. Varlamis. Blogrank: ranking weblogs based on connectivity and similarity features. In *AAA-IDEA '06: Proceedings of the 2nd international workshop on Advanced architectures and algorithms for internet delivery and applications*, page 8, New York, NY, USA, 2006. ACM Press.
10. P. Lawrence, B. Sergey, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University, 1998.
11. J. List, V. Mihajlovic, G. Ramirez, A. de Vries, D. Hiemstra, and H. Blok. Tijah: Embracing ir methods in xml database. *Information Retrieval*, 8(4):547 – 570, December 2005.
12. P. Serdyukov, H. Rode, and D. Hiemstra. University of Twente at the TREC 2007 Enterprise Track: Modeling relevance propagation for the expert search task. In *Proceedings of the 16th Text REtrieval Conference (TREC 2007)*, 2007.
13. A. Shakeri and C. Zhai. A probabilistic relevance propagation model for hypertext retrieval. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 550–558, New York, NY, USA, 2006. ACM Press.
14. A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 21–29, New York, NY, USA, 1996. ACM.
15. H. Zaragoza, H. Rode, P. Mika, J. Atserias, M. Ciaramita, and G. Attardi. Ranking very many typed entities on wikipedia. In *CIKM '07*, Lisbon, Portugal, 2007.

Experiments on Category Expansion at INEX 2007

Janne Jämsen,¹ Turkka Näppilä,¹ and Paavo Arvola²

¹Department of Computer Sciences, Kanslerinrinne 1,
33014 University of Tampere, Finland
{janne.jamsen; turkka.nappila}@cs.uta.fi

²Department of Information Studies, Kanslerinrinne 1,
33014 University of Tampere, Finland
paavo.arvola@uta.fi

Abstract. In this study we examine the effect of the category expansion in entity ranking. In the category expansion we expanded the given category information of the topics by using a coefficient propagation method for the category hierarchy. In INEX 2007 XML Entity Ranking Track, we took part entity ranking and list completion tasks.

1. Introduction

In this paper we present the information retrieval (IR) experiments we conducted within the XML Entity Ranking (XER) Track at INEX 2007. The Track was comprised of two tasks: entity ranking (ER) and list completion (LC).

There are three components in Wikipedia XML test collection of the XER Track we discovered to be useful in entity ranking:

1. the textual content of the Wikipedia articles,
2. the category hierarchy, and
3. the link structures between Wikipedia articles.

The Wikipedia test collection consists of approximately 659,000 XML documents, which are classified using approximately 113,000 categories. In addition, there are approximately 13,900,000 interlinked document-pairs.

We start our query evaluation with a standard result list generated by a partial match-based information retrieval system. For that purpose, we use TRIX (Tampere Retrieval and Indexing for XML) with DoOrBa scoring method to process the topic titles against the collection as [1]. In the ER task, target categories are explicitly given in the topics, and in the LC task the categories are related to the sample entities (Wikipedia articles). Because of the inconsistencies in the category mark-up, the given categories ought to be interpreted as vague, and therefore a straightforward pruning of the result list with the explicitly given categories is not enough.

The essential contribution of our paper is on utilizing the category hierarchy. The categories of the Wikipedia XML test collection form a hierarchy, more specifically a directed acyclic graph (DAG). In order to get more precise answers for the queries, we use a category expansion method, which propagates descending coefficients especially to the nearby categories (i.e., parents, children, and siblings) of the given categories in the hierarchy.

The rest of the paper is organized as follows. Section 2 introduces our category expansion method and other experiments. In Section 3, the runs and preliminary results are presented. Finally, the results are discussed and the conclusions are drawn in Section 4.

2. Approach

As a baseline for our experiments, we matched topic titles against the textual contents of Wikipedia articles using TRIX. In addition, we implemented and tested two complementary methods: *category expansion* (used in both tasks) and *link expansion* (used in the ER task only). We also experimented with a method for automatically classifying Wikipedia articles. Short descriptions of these methods can be found in the following sections (2.1. and 2.2.).

2.1. Category expansion

The category expansion, as understood in this paper, stands for the act of deriving from a set of initial categories (specific to a topic) an expanded set of categories that covers the relevant entities more or less accurately. Each category in the expanded set (or in the hierarchy as a whole if also zero scores are used) can be assigned a numeric coefficient, a *matching score*, which describes its conformance to the initial categories (the greater the score, the more closely the category matches to the initial categories).¹

In a classical, well-defined is-a hierarchy (e.g., found in thesauri and in many programming and modeling languages), members of a subcategory (i.e., a specialization) are implicitly members of the corresponding supercategory, too. For example, each art museum is necessarily a museum. As a result, given that we want only museums to be included in an answer, we can prioritize the entities that have been assigned to the category *museums* and/or one of its (direct or indirect) subcategories. Provided the

¹ Note that the notion of graph-oriented expansion in the context of IR is not novel to this paper. For example, Järvelin, Kekäläinen and Niemi [4] introduce a tool for ontology-based query expansion. Also noteworthy are the various spreading activation-based techniques for keyword search and related IR tasks (see, e.g., [2, 3]) as well as many hyperlink-based IR methods.

categories have a full coverage (i.e., there are no museums besides those under the category *museums* and its subcategories), we can *restrict* ourselves to these entities.

Unfortunately, the semantics of the category hierarchy of Wikipedia follows neither of these principles in detail.² This makes it practically impossible to make a binary distinction between matching and non-matching categories. For this, we took a somewhat fuzzier approach on the category expansion in which the distance within a category hierarchy is the determining factor in approximating the extent of match between two categories. The extent may lessen both in moving upwards or downwards in the hierarchy but possibly at different rates.

In what follows we conceptualize the category hierarchy as a directed acyclic graph and adopt the conventional parent-child terminology to denote the hierarchical relationships among categories.

The starting point of the category expansion is a set of *initial categories*. For each topic in the ER task, this is the set of given categories that specify the desired type of entities in an answer. For each topic in the LC task, a set of initial categories, which may or may not be relevant, is obtained indirectly from the provided (correct) sample entities by taking each category that has at least one (explicitly assigned) member among them. Once the set of initial categories $\{c_1, c_2, \dots, c_n\}$ is established, we execute one category expansion per each included category c_i ($1 \leq i \leq n$) as follows.

Each category c_j in the hierarchy is associated with a *matching score* $M_i(c_j)$ in the range $[0,1]$, initially set to zero. The set of *current categories*, denoted by C , is initialized to $\{c_i\}$. The *current matching score* S is initialized to 1. The user-provided parameters (shared by all topics) include:

- D : *decay down*, a coefficient in the range $[0,1]$ that determines the rate the matching scores diminish during the downward expansion (see below);
- U : *decay up*, a coefficient in the range $[0,1]$ that determines the rate the matching scores diminish during the upward expansion (see below);
- T : *threshold*, a constant in the range $[0,1]$ that constraints the upward expansion.

The following steps (1 – 6) are then executed.

- (1) $S < T$, then exit.
- (2) For each category c_k in C , assign $M_i(c_k) := S$.
- (3) Assign the score $D \cdot M_i(c_m)$ for each *descendant* c_d of the nodes in C such that $M_i(c_d) = 0$ where c_m is the parent category of c_d (or the parent whose M_i

² For example, the article for the Finnish author Tove Jansson is assigned to a subcategory of the category *countries*. Obviously, this does not imply an is-a relationship.

score is maximal among the parents if c_d has several parents). In practice, assignments are made starting from the children of the categories in C and proceeding downwards, one level of depth at time until no more categories meeting the above criteria are found. This is called the *downward expansion*.

- (4) Reset C to contain the *parents* of the nodes currently in C . This is called the *upward expansion*.
- (5) Assign $S := U \cdot S$.
- (6) Return to the step (1).

The resulting scoring can be characterized as follows (c is an arbitrary category):

$$M_i(c) = \begin{cases} U^{\text{dist}(c_i, \text{LCA}(c_i, c))} \cdot D^{(c, \text{LCA}(c_i, c))}, & \text{if } U^{\text{dist}(c_i, \text{LCA}(c_i, c))} \cdot D^{(c, \text{LCA}(c_i, c))} \geq T \\ 0, & \text{otherwise} \end{cases}$$

Here $\text{dist}(x, y)$ denotes the shortest distance (measured as the number of upward transitions in the hierarchy from x to y) between the categories x and y , and $\text{LCA}(x, y)$ denotes the lowest common ancestor of the categories x and y .

After the matching scores are calculated per each initial category, the *total matching score* $M(c)$ of an arbitrary category c can be calculated using either of the formulas (1) or (2):

$$M(c) = \max_i^n M_i(c) \quad (1)$$

$$M(c) = \sum_i^n M_i(c) \quad (2)$$

Especially in the LC task, the formula (2) can be assumed to bring better out the categories that are shared by multiple (correct) sample entities (and which are therefore more likely to be relevant). In order to balance the summing effect, we also experimented with the logarithm of the formula (2) and a weighted average of the formulas (1) and (2).

Figure 1 demonstrates the category expansion in the case of two initial categories. The two expansions are depicted in the panels *a* and *b*. The final matching scores (calculated simply as sums) are depicted in the panel *c*. The category hierarchy is interpreted from the top down, ancestors shown above descendants. The decay down coefficient is 0.9 and the decay up coefficient 0.5.

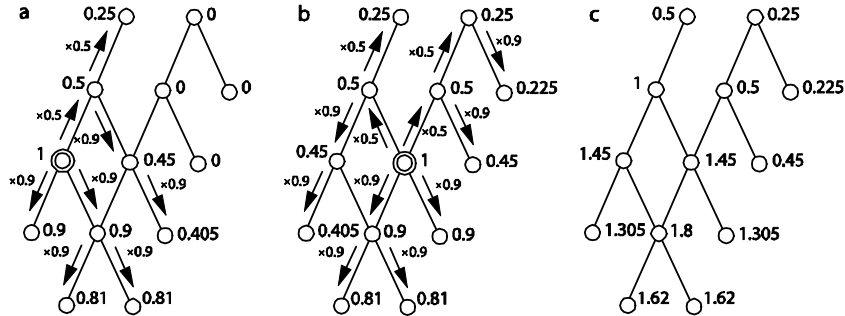


Figure 1. The category expansion with two initial categories.

2.2. Other experiments

In the ER task we experimented with a modification of the above-like expansion where, instead of the category hierarchy, (wiki-)links among articles (i.e., entities) were utilized. The underlying assumption is that the links contained in encyclopedia articles usually point out to other articles that are somewhat closely related to them. This is reminiscent of the modified tf-idf schemes in which the content of the neighboring hyperlinked pages is taken into account [5]. Because encyclopedia articles are usually designed to avoid extensive overlapping, this sort of strategy could be assumed to work even better for Wikipedia articles than for random web pages.

A rough outline of this type of link expansion is as follows. First, a text retrieval system (TRIX in our case) is used to select the top n matching articles together with their associated document scores. After this, each of the top articles is used as a basis for expansion whereby the initial document score, continuously multiplied by a decay coefficient, is propagated to the articles that are connected to it either by outcoming or incoming links (or both). The expansion halts once a preset threshold or depth constraint is met. After the expansions, the accumulated scores are aggregated as above (using, e.g., sum or maximum).

For example, evaluating the query “Nordic authors noted for children’s literature” using a text-based retrieval system might give a high score to the articles *Nordic countries* and *children’s literature*. An article describing a relevant author, such as Tove Jansson, even if it would not contain the words *Nordic* and *children* might contain a link to the article *Finland* which in turn contains a link to the article *Nordic countries* (or the other way around). The article *children’s literature*, for its part, might contain links that point out directly or intermediately to Tove Jansson. Ideally, after the scores gained during the expansions are summed up, articles for Tove Jansson and other relevant authors end up having significant total scores of their own.

Unfortunately, the graph structure induced by links among documents is remarkably more massive than the category hierarchy. Given the high number of final topics (over 70) and insufficient RAM memory, we were unable to test the expansion to depths greater than 1 in the available time.

Our third major experiment was to assign (i.e., to classify) Wikipedia articles automatically to the existing categories based on the textual content of the articles. The aim was to augment the “official” intellectually-made classifications by machine-made classifications. The classification algorithm tries to extract a set of potential category names from the definition of a term using a variety of language patterns. However, because of the poor results on training topics, we either heavily prioritized the official classifications or limited its usage to those topics that were otherwise short of candidate entities. Nevertheless, the issue is still far from settled and we will continue the efforts towards making the approach more workable.

3. Runs and Preliminary Results

Due to the lack of relevance assessments for the final topics, only the given 28 training topics were used for trial runs. The results were evaluated using the standard precision and recall metrics.

Figure 2 depicts the interpolated precision–recall curves (recall 0 - 100 %) of some of our experiments. These include TRIX results (*trix*), the category expansion (*category_exp* for the ER and LC tasks), TRIX results combined with the category expansion (*trix&category_exp* for the ER and LC tasks), TRIX results combined first with the link expansion and then with the category expansion (*trix&link_exp&category_exp* for the ER task), and finally TRIX results combined with the link expansion (*trix&link_exp*).

The documents scores returned by TRIX can be considered as the baseline for assessments. We can see from Figure 2 that the link expansion without any other method seems to produce the weakest result. When the link expansion is combined with the category expansion the results climb above the baseline. On the other hand, the other methods used in the ER task produced results that are under the baseline. In general, better results were gained in the LC task. What is remarkable is that in the LC task the category expansion without any other method seems to produce the best results. We discuss possible reasons for these results in Section 4.

Next, we describe the preliminary results (on the 28 training topics) for our final runs submitted to the XML Entity Ranking Track at INEX 2007.

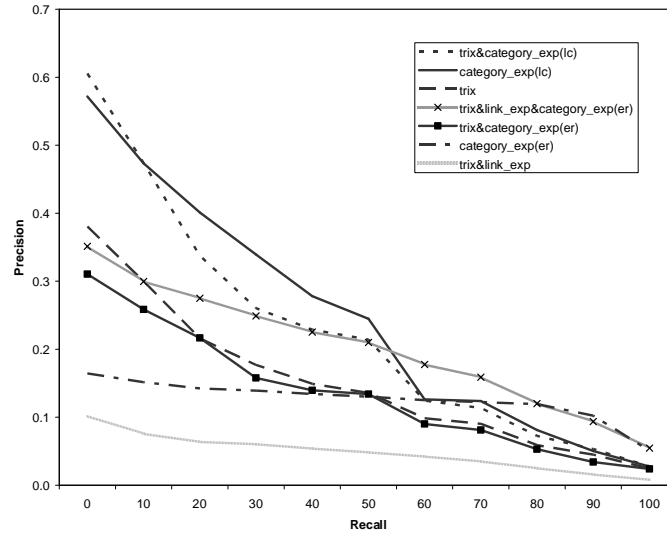


Figure 2. Experiments on different methods and their combinations

3.1. Task 1: Entity ranking

The preliminary results of our final runs for the ER task are depicted in the form of interpolated precision–recall curves in Figure 3. For the runs *utampere_er_2* and *utampere_er_3* the initial document scores returned by TRIX were propagated by the link expansion to the depth 1, whereas for the runs *utampere_er_1* and *utampere_er_4* the initial document scores were used as such. For each run, the resulting scores were combined with the matching scores from the category expansion. (In addition, automatic document classification was used in the run *utampere_er_1* for all topics.) In other respects the runs differ only slightly in parameter values. As shown, the runs *utampere_er_2* and *utampere_er_3* outperform the runs *utampere_er_1* and *utampere_er_4* on the training topics.

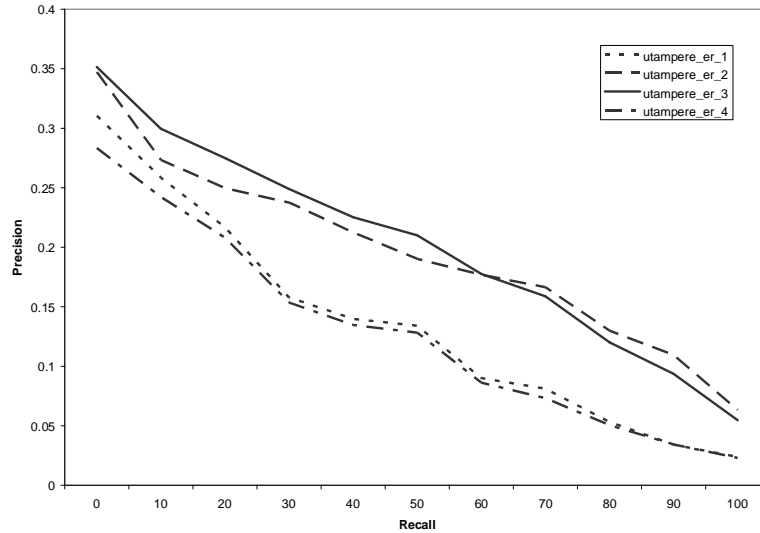


Figure 3. The preliminary results for the Entity Ranking Task

3.2. Task 2: List completion

Figure 4 reports the preliminary results of our final runs for the LC task. In the run *utampere_lc_1* we used the category expansion method alone, which seems to produce the highest mean average precision rate on the training topics. On the other hand, the highest early precision rate is produced by the run *utampere_lc_2* where the heavily prioritized matching scores from the category expansion are combined with the documents scores returned by TRIX. The runs *utampere_lc_3* and *utampere_lc_4*, which both produced lower early and mean average precision rates than the previous runs, resemble the run *utampere_lc_2* but different parameter values are used in them.

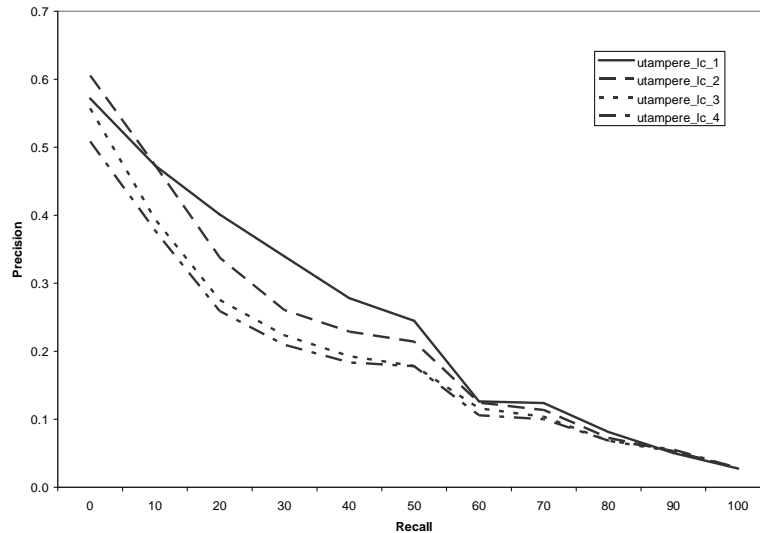


Figure 4. The preliminary results for the List Completion Task

4. Conclusions

Clearly, some of the tested methods and their combinations improve upon the baseline (TRIX) method (see Figure 2). For the LC task the category expansion alone and combined with TRIX yields a higher precision than the mere baseline. For the ER task, the baseline precision at recall 0 % is slightly better than in the case of alternative methods. However, the mean average precision for the combination of the link expansion (applied to TRIX) and the category expansion is notably higher. Other methods for the ER task seem to fall below the baseline.

The LC task produced generally better results than the ER task (see Figures 2 - 4). A likely explanation lies on the difference in the nature of these tasks. In the ER task, only one category is given per topic (there are, however, some topics with two categories) whereas in the LC task the provided sample entities (approximately 3 - 4) are usually labeled with multiple categories each. This means that the set of initial categories for the category expansion is usually more extensive and fine-grained in the LC task than in the ER task. Thus, it is more straightforward to find related entities in the LC task than in the ER task.

What was an exceptional in the LC task was the fact that the category expansion alone nearly outperformed all the other methods (see Figures 2 and 4). That is, taking the topic title into account did not improve the results as it intuitively should. This seems even more surprising as there rarely exists a single category that directly corre-

sponds to the specific information need expressed in the topic title. Although we are still waiting to see whether this holds also for the final topics or whether this is due to some sort of anomaly in the training data, this demonstrates the usefulness of prioritizing categories that are shared by multiple sample entities.

Finally, the link expansion alone seems to worsen the results while combined with the category expansion has an improving effect (see Figures 2 and 3). (This combination was not tested for the LC task). This might suggest that the positive (intended) effects of the link expansion are only able to exceed the inevitable losses in the precision if category information is taken into account as a constraining factor.

Acknowledgements

This study was funded in part by the Tampere Graduate School in Information Science and Engineering (TISE) and the Academy of Finland under grant number 115480. The travel and accommodation costs were guaranteed by the Nordic Research School in Library and Information Science (NORSLIS).

References

1. Arvola, P.: Document order based scoring for XML retrieval. In Preproceedings of INEX 2007 (6 pages).
2. Aswath, D., Ahmed, S.T., D'cunha, J., and Davulcu, H.: Boosting item keyword search with spreading activation, In Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence, (2005) 704-707.
3. Crestani, F.: Application of spreading activation techniques in information retrieval. *Artificial Intelligence Review*, 11(6), (1997) 453-482.
4. Järvelin, K., Kekäläinen, J., and Niemi, T.: ExpansionTool: Concept-based query expansion and construction. *Information Retrieval*, 4(3/4) (2001) 231-255.
5. Sugiyama, K., Hatano, K., Yoshikawa, M., and Uemura, S.: Refinement of tf-idf schemes for web pages using their hyperlinked neighboring pages, In Proceedings of the Fourteenth ACM Conference on Hypertext and Hypermedia, (2003) 198-207.

Using Wikipedia Categories and Links in Entity Ranking

Anne-Marie Vercoustre¹, Jovan Pehcevski¹, and James A. Thom²

¹ INRIA Rocquencourt, France

{anne-marie.vercoustre,jovan.pehcevski}@inria.fr

² RMIT University, Melbourne, Australia

james.thom@rmit.edu.au

Abstract. This paper describes the participation of the INRIA group in the INEX 2007 XML entity ranking and ad hoc tracks. We developed a system for ranking Wikipedia entities in answer to a query. Our approach utilises the known categories, the link structure of Wikipedia, as well as the link co-occurrences with the examples (when provided) to improve the effectiveness of entity ranking. Our experiments on the training data set demonstrate that the use of categories and the link structure of Wikipedia, together with entity examples, can significantly improve entity retrieval effectiveness. We also use our system for the ad hoc tasks by inferring target categories from the title of the query. The results were worse than when using a full-text search engine, which confirms our hypothesis that ad hoc retrieval and entity retrieval are two different tasks.

1 Introduction

Entity ranking has recently emerged as a research field that aims at retrieving entities as answers to a query [5, 8, 10, 11]. Here, unlike in the related field of entity extraction, the goal is not to tag the names of the entities in documents but rather to get back a list of the relevant entity names. It is a generalisation of the expert search task explored by the TREC Enterprise track [9], except that instead of ranking people who are experts in the given topic, other types of entities such as organizations, countries, or locations can also be retrieved and ranked.

The Initiative for the Evaluation of XML retrieval (INEX) is running a new track on entity ranking in 2007, using Wikipedia as its document collection [3]. There are two tasks in the INEX 2007 XML entity ranking (XER) track: *entity ranking*, which aims at retrieving entities of a given category that satisfy a topic described in natural language text; and *list completion*, where given a topic text and a small number of entity examples, the aim is to complete this partial list of answers. Two data sets were used by the participants of the INEX 2007 XER track: a *training* data set, comprising 28 XER topics which were adapted from the INEX 2006 ad hoc topics and proposed by our INRIA participating group; and a *testing* data set, comprising 73 XER topics most of which were proposed

```
<inex_topic>
<title>
European countries where I can pay with Euros
</title>
<description>
I want a list of European countries where I can pay with Euros.
</description>
<narrative>
Each answer should be the article about a specific European country
that uses the Euro as currency.
</narrative>
<entities>
  <entity ID="10581">France</entity>
  <entity ID="11867">Germany</entity>
  <entity ID="26667">Spain</entity>
</entities>
<categories>
<category ID="185">european countries</category>
</categories>
</inex_topic>
```

Fig. 1. Example INEX 2007 XML entity ranking topic

and assessed by the track participants. The main purpose of having two data sets is to allow participants to tune the parameters of their entity ranking systems on the training data set, and then use the optimal parameter values on the testing data set.

An example of an INEX 2007 XER topic is shown in Figure 1. Here, the **title** field contains the plain content only query, the **description** provides a natural language description of the information need, and the **narrative** provides a detailed explanation of what makes an entity answer relevant. In addition to these fields, the **entities** field provides a few of the expected entity answers for the topic (task 2), while the **categories** field provides the target category of the expected entity answers (task 1).

In this new track, the expected entities correspond to Wikipedia articles that are likely to be referred to by links in other articles. As an example, the query “European countries where I can pay with Euros” [3] should return a list of entities (or pages) representing relevant countries, and not a list of entities representing non-relevant (country or other) names found in pages about the Euro and similar currencies.

In this paper, we describe our approach to ranking entities from the Wikipedia XML document collection. Our approach is based on the following principles:

1. A good entity page is a page that answers the query (or a query extended with names of target categories or entity examples).

2. A good entity page is a page associated with a category close to the target category (task 1) or to the categories of the entity examples (task 2).
3. A good entity page is referred to by a page answering the query; this is an adaptation of the HITS [6] algorithm to the problem of entity ranking.
4. A good entity page is referred to by contexts with many occurrences of the entity examples (task 2). A broad context could be the full page that contains the entity examples, while smaller and more narrow contexts could be elements such as paragraphs, lists, or tables.

After a short presentation of the INEX Wikipedia XML collection used for entity ranking, we provide a detailed description of our entity ranking approach and the runs we submitted for evaluation to the INEX 2007 XER track. We also report on our run submissions to the INEX 2007 ad hoc track.

2 INEX Wikipedia XML collection

Wikipedia is a well known web-based, multilingual, free content encyclopedia written collaboratively by contributors from around the world. As it is fast growing and evolving it is not possible to use the actual online Wikipedia for experiments, and so we need a stable collection to do evaluation experiments that can be compared over time. Denoyer and Gallinari [4] have developed an XML-based corpus based on a snapshot of the Wikipedia, which has been used by various INEX tracks in 2006 and 2007. It differs from the real Wikipedia in some respects (size, document format, category tables), but it is a very realistic approximation.

2.1 Entities in Wikipedia

The entities have a name (the name of the corresponding page) and a unique ID in the collection. When mentioning such an entity in a new Wikipedia article, authors are encouraged to link every occurrence of the entity name to the page describing this entity. This is an important feature as it allows to easily locate potential entities, which is a major issue in entity extraction from plain text.

However in this collection, not all potential entities have been associated with corresponding pages. For example, if we look for Picasso's artworks, only three paintings ("Les Demoiselles d'Avignon", "Guernica", and "Le garçon à la pipe") get associated pages. If the query was "paintings by Picasso", we would not expect to get more than three entity pages for Picasso's paintings, while for the online Wikipedia there are about thirty entities, yet not that many compared to the actual number of his listed paintings.

The INEX XER topics have been carefully designed to make sure there is a sufficient number of answer entities. For example, in the Euro page (see Fig. 2), all the underlined hypertext links can be seen as occurrences of entities that are each linked to their corresponding pages. In this figure, there are 18 entity references of which 15 are country names; specifically, these countries are all "European Union member states", which brings us to the notion of category in Wikipedia.

“The **euro** ... is the official currency of the Eurozone (also known as the Euro Area), which consists of the European states of Austria, Belgium, Finland, France, Germany, Greece, Ireland, Italy, Luxembourg, the Netherlands, Portugal, Slovenia and Spain, and will extend to include Cyprus and Malta from 1 January 2008.”

Fig. 2. Extract from the Euro Wikipedia page

2.2 Categories in Wikipedia

Wikipedia also offers categories that authors can associate with Wikipedia pages. There are 113,483 categories in the INEX Wikipedia XML collection, which are organised in a graph of categories. Each page can be associated with many categories (2.28 as an average).

Wikipedia categories have unique names (e.g. “France”, “European Countries”, “Countries”). New categories can also be created by authors, although they have to follow Wikipedia recommendations in both creating new categories and associating them with pages. For example, the Spain page is associated with the following categories: “Spain”, “European Union member states”, “Spanish-speaking countries”, “Constitutional monarchies” (and some other Wikipedia administrative categories).

When searching for entities it is natural to take advantage of the Wikipedia categories since they would give a hint on whether the retrieved entities are of the expected type. For example, when looking for entities “authors”, pages associated with the category “Novelist” may be more relevant than pages associated with the category “Book”.

3 Our entity ranking approach

Our approach to identifying and ranking entities combines: (1) the full-text similarity of the answer entity page with the query; (2) the similarity of the page’s categories with the target categories (task 1) or the categories attached to the entity examples (task 2); and (3) the contexts around entity examples (task 2) found in the top ranked pages returned by a search engine for the query.

We have built a system based on the above ideas, and a framework to tune and evaluate a set of different entity ranking algorithms.

3.1 Architecture

The system involves several modules and functions that are used for processing a query, submitting it to the search engine, applying our entity ranking algorithms, and finally returning a ranked list of entities. We use Zettair³ as our choice for a full-text search engine. Zettair is a full-text information retrieval (IR) system

³ <http://www.seg.rmit.edu.au/zettair/>

developed by RMIT University, which returns pages ranked by their similarity score to the query. In a recent comparison of open source search engines, Zettair was found to be “one of the most complete engines” [7]. We used the Okapi BM25 similarity measure that has proved to work well on the INEX 2006 Wikipedia test collection [1].

Our system involves the following modules and functions:

- the topic module takes an INEX topic as input (as the topic example shown in Fig. 1) and generates the corresponding Zettair query and the list of target categories and entity examples (as an option, the names of target categories or example entities may be added to the query);
- the search module sends the query to Zettair and returns a list of ranked Wikipedia pages (typically 1500);
- the link extraction module extracts the links from a selected number of highly ranked pages,⁴ together with the information concerning the paths of the links (using an XPath notation);
- the category similarity module calculates a weight for a page based on the similarity of the page categories with target categories or those of the entity examples (see 3.2);
- the linkrank module calculates a weight for a page based (among other things) on the number of links to this page (see 3.4); and
- the full-text IR module calculates a weight for a page based on its initial Zettair score (see 3.4).

The global score for a page is calculated as a linear combination of three normalised scores coming out of the last three modules (see 3.4).

The architecture provides a general framework for evaluating entity ranking which allows for some modules to be replaced by more advanced modules, or by providing a more efficient implementation of a module. It also uses an evaluation module to assist in tuning the system by varying the parameters and to globally evaluate our entity ranking approach.

The current system was not designed for online entity ranking in Wikipedia. First, because we are not dealing with the online Wikipedia, and second because of performance issues. The major cost in running our system is in extracting the links from the selected number of pages retrieved by the search engine. Although we only extract links once by topic and store them in a database for reuse in later runs, for an online system it would be more efficient to extract and store all the links at indexing time.

3.2 Using Wikipedia categories

To make use of the Wikipedia categories in entity ranking, we define similarity functions between:

⁴ We discarded external links and some internal collection links that do not refer to existing pages in the INEX Wikipedia collection.

- the categories of answer entities and the target categories (task 1), or
- the categories of answer entities and a set of categories attached to the entity examples (task2).

Similarity measures between concepts of the same ontology, such as tree-based similarities [2], cannot be applied directly to Wikipedia categories, mostly because the notion of sub-categories in Wikipedia is not a subsumption relationship. Another reason is that categories in Wikipedia do not form a hierarchy (or a set of hierarchies) but a graph with potential cycles [10, 12].

Task 1 We first define a similarity function that computes the ratio of common categories between the set of categories $\text{cat}(t)$, associated to an answer entity page t , and the set $\text{cat}(C)$ which is the union of the provided target categories C :

$$S_C(t) = \frac{|\text{cat}(t) \cap \text{cat}(C)|}{|\text{cat}(C)|} \quad (1)$$

The target categories will be generally very broad, so it is to be expected that the answer entities would not be directly attached to these broad categories. Accordingly, we experimented with several extensions of the set of categories, both for the target categories and the categories attached to answer entities.

We first experimented with extensions based on using sub-categories and parent categories in the graph of Wikipedia categories. However, on the training data set, we found that these category extensions overall do not result in an improved performance [10], and so they were not used in our INEX 2007 runs.

Another approach is to use lexical similarity between categories. For example, “european countries” is lexically similar to “countries” since they both contain the word “countries” in their names. We use an information retrieval approach to retrieve similar categories, by indexing with Zettair all the categories, using their names as corresponding documents. By sending both the title of the topic T and the category names C as a query to Zettair, we then retrieve all the categories that are lexically similar to C . We keep the top M ranked categories and add them to C to form the set $\text{TCcat}(C)$. On the training data set, we found that the value $M=5$ is the optimal parameter value used to retrieve the likely relevant categories for this task [10]. We then use the same similarity function as before, where $\text{cat}(C)$ is replaced with $\text{TCcat}(C)$.

We also experimented with two alternative approaches: by sending the category names C as a query to Zettair (denoted as $\text{Ccat}(C)$); and by sending the title of the topic T as a query to Zettair (denoted as $\text{Tcat}(C)$). On the training data set we found that these two approaches were less effective than the $\text{TCcat}(C)$ approach [10]. However, we used the $\text{Tcat}(C)$ category set in the ad-hoc runs where the target category is not provided.

Task 2 Here, the categories attached to entity examples are likely to correspond to very specific categories, just like those attached to the answer entities. We define a similarity function that computes the ratio of common categories between

the set of categories attached to an answer entity page $\text{cat}(t)$ and the set of the union of the categories attached to entity examples $\text{cat}(E)$:

$$S_C(t) = \frac{|\text{cat}(t) \cap \text{cat}(E)|}{|\text{cat}(E)|} \quad (2)$$

3.3 Exploiting locality of links

For task 2, exploiting locality of links around entity examples can significantly improve the effectiveness of entity ranking [8]. The idea is that entity references (links) that are located in close proximity to the entity examples, especially in list-like elements, are likely to refer to more relevant entities than those referred to by links in other parts of the page. Here, the very notion of *list* involves grouping together objects of the same (or similar) nature. We are therefore looking for links that co-occur with links to entity examples in such list-like elements.

Consider the example of the Euro page shown in Fig. 2, where France, Germany and Spain are the three entity examples (as shown in Fig. 1). We see that the 15 countries that are members of the Eurozone are all listed in the same paragraph with the three entity examples. In fact, there are other contexts in this page where those 15 countries also co-occur together. By contrast, although there are a few references to the United Kingdom in the Euro page, it does not occur in the same context as the three examples (except for the page itself).

We have identified in the Wikipedia collections three types of elements that correspond to the notion of lists: paragraphs (tag `p`); lists (tags `normallist`, `numberlist`, and `definitionlist`); and tables (tag `table`). We use an algorithm for identifying the (static) element contexts on the basis of the leftmost occurrence of any of the pre-defined tags in the absolute XPath of entity examples. The resulting list of element contexts is sorted in a descending order according to the number of distinct entity examples contained by the element. If two elements contain the same number of distinct entity examples, the one that has a longer XPath length is ranked higher. Finally, starting from the highest ranked element, we filter all the elements in the list that either contain or are contained by that element. We end up with a final list of (one or more) non-overlapping elements that represent the statically defined contexts for the page.⁵

Consider Table 1, where the links to entity examples are identified by their absolute XPath notations. The three static contexts that will be identified by the above algorithm are the elements `p[1]`, `normallist[1]` and `p[3]`. The first two element contexts contain the three (distinct) examples, while the last one contains only one entity example.

The drawback of this approach is that it requires a predefined list of static elements that is completely dependent on the collection. The advantage is that

⁵ In the case when there are no occurrences of the pre-defined tags in the XPath of an entity example, the document element (`article[1]`) is chosen to represent the element context.

Table 1. List of links referring to entity examples (France, Germany, and Spain), extracted from the page 9272.html, for the INEX 2007 XER topic shown in Fig. 1.

Page		Links	
ID	Name	XPath	ID Name
9472	Euro	/article[1]/body[1]/p[1]/collectionlink[7]	10581 France
9472	Euro	/article[1]/body[1]/p[1]/collectionlink[8]	11867 Germany
9472	Euro	/article[1]/body[1]/p[1]/collectionlink[15]	26667 Spain
9472	Euro	/article[1]/body[1]/p[3]/p[5]/collectionlink[6]	11867 Germany
9472	Euro	/article[1]/body[1]/normallist[1]/item[4]/collectionlink[1]	10581 France
9472	Euro	/article[1]/body[1]/normallist[1]/item[5]/collectionlink[2]	11867 Germany
9472	Euro	/article[1]/body[1]/normallist[1]/item[7]/collectionlink[1]	26667 Spain
9472	Euro	/article[1]/body[1]/normallist[1]/item[8]/collectionlink[1]	26667 Spain

the contexts are fast to identify. We have also experimented with an alternative algorithm that dynamically identifies the link contexts by utilising the underlying XML document structure. On the training data set, we found that this algorithm does not significantly improve the entity ranking performance compared to the algorithm that uses the static contexts [8].

3.4 Score Functions and parameters

The core of our entity ranking approach is based on combining different scoring functions for an answer entity page, which we now describe in more detail.

LinkRank score The linkrank function calculates a score for a page, based on the number of links to this page, from the first N pages returned by the search engine in response to the query. The number N has been kept to a relatively small value mainly for performance issues, since Wikipedia pages contain many links that would need to be extracted. We carried out some experiments with different values of N and found that $N=20$ was a good compromise between performance and discovering more potentially good entities.

The linkrank function can be implemented in a variety of ways. We have implemented a linkrank function that, for an answer entity page t , takes into account the Zettair score of the referring page $z(p)$, the number of distinct entity examples in the referring page $\#ent(p)$, and the locality of links around the entity examples:

$$S_L(t) = \sum_{r=1}^N \left(z(p_r) \cdot g(\#ent(p_r)) \cdot \sum_{l_t \in L(p_r, t)} f(l_t, c_r | c_r \in C(p_r)) \right) \quad (3)$$

where $g(x) = x + 0.5$ (we use 0.5 to allow for cases where there are no entity examples in the referring page); l_t is a link that belongs to the set of links

$L(p_r, t)$ that point from the page p_r to the answer entity t ; c_r belongs to the set of contexts $C(p_r)$ around entity examples found for the page p_r ; and $f(l_t, c_r)$ represents the weight associated to the link l_t that belongs to the context c_r .

The weighting function $f(l_r, c_r)$ is represented as follows:

$$f(l_r, c_r) = \begin{cases} 1 & \text{if } c_r = p_r \text{ (the context is the full page)} \\ 1 + \#ent(c_r) & \text{if } c_r = e_r \text{ (the context is an XML element)} \end{cases}$$

A simple way of defining the context of a link is to use its full embedding page [11]. In this work we use smaller contexts using predefined types of elements such as paragraphs, lists and tables (as described in sub-section 3.3).

Category similarity score As described in sub-section 3.2, the category score $S_C(t)$ for the two tasks is calculated as follows:

task 1

$$S_C(t) = \frac{|\text{cat}(t) \cap \text{cat}(C)|}{|\text{cat}(C)|} \quad (4)$$

For task 1, we consider variations on the category score $S_C(t)$ based on lexical similarities of category names (see sub-section 3.2), by replacing $\text{cat}(C)$ with $\text{TCcat}(C)$.

task 2

$$S_C(t) = \frac{|\text{cat}(t) \cap \text{cat}(E)|}{|\text{cat}(E)|} \quad (5)$$

On the training data set, we found that extending the set of categories attached to both entity examples and answer entities did not increase the entity ranking performance [10], and so for task 2 we do not use any category extensions.

Z score The Z score assigns the initial Zettair score to an answer entity page. If the answer page does not appear among the initial ranked list of pages returned by Zettair, then its Z score is zero:

$$S_Z(t) = \begin{cases} z(t) & \text{if page } t \text{ was returned by Zettair} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Global score The global score $S(t)$ for an answer entity page is calculated as a linear combination of three normalised scores, the linkrank score $S_L(t)$, the category similarity score $S_C(t)$, and the Z score $S_Z(t)$:

$$S(t) = \alpha S_L(t) + \beta S_C(t) + (1 - \alpha - \beta) S_Z(t) \quad (7)$$

where α and β are two parameters that can be tuned differently depending on the entity retrieval task.

We consider some special cases that allow us to evaluate the effectiveness of each module in our system: $\alpha = 1, \beta = 0$, which uses only the linkrank score; $\alpha = 0, \beta = 1$, which uses only the category score; and $\alpha = 0, \beta = 0$, which uses only the Z score.⁶ More combinations for the two parameters are explored in the training phase of our system. The optimal combination is then used on the testing data set.

4 Experimental results

In this section, we present results that investigate the effectiveness of our entity ranking approach when applied to both the INEX 2007 XER and ad hoc tracks.

We first tune the system parameters using the training collection, and then we apply the optimal values on the test collection. We submitted three runs for task 1 and three runs for task 2. For this track, we aim at investigating the impact of using various category and linkrank similarity techniques on the entity ranking performance. We also compare the performances of our entity ranking runs to that achieved by a full-text retrieval run. For the ad hoc track, we submitted three entity ranking runs that correspond to the three individual modules of our system and compare it with the full text Zettair run submitted by RMIT. For this track, we aim at investigating the impact of using our entity ranking approach on the ad hoc retrieval performance.

4.1 XER training data set (28 topics)

The XER training data set was developed by our participating group. It is based on a selection of topics from the INEX 2006 ad hoc track. We chose 27 topics that we considered were of an “entity ranking” nature, where for each page that had been assessed as containing relevant information, we reassessed whether or not it was an entity answer, and whether it *loosely* belonged to a category of entity we had *loosely* identified as being the target of the topic. If there were entity examples mentioned in the original topic these were used as entity examples in the entity topic. Otherwise, a selected number (typically 2 or 3) of entity examples were chosen somewhat arbitrarily from the relevance assessments. We also added the Euro topic example (shown in Fig. 1) from the original INEX description of the XER track [3], resulting in total of 28 entity ranking topics.

⁶ This is not the same as the plain Zettair score, as apart from answer entities corresponding to the highest N pages returned by Zettair, the remaining entity answers are all generated by extracting links from these pages, which may or may not correspond to the initial 1500 pages retrieved by Zettair.

Table 2. Performance scores for Zettair and our three XER submitted runs on the training data set (28 topics), obtained for task 1 with different evaluation measures. For each measure, the best performing score is shown in bold.

Run	cat-sim	α	β	P[r]		R-prec	MAP
				5	10		
Zettair		-	-	0.229	0.232	0.208	0.172
run 1	cat(C)-cat(t)	0.0	1.0	0.229	0.250	0.215	0.196
run 2	TCcat(C)-cat(t)	0.0	1.0	0.307	0.318	0.263	0.242
run 3	TCcat(C)-cat(t)	0.1	0.8	0.379	0.361	0.338	0.287

We use mean average precision (MAP) as our primary method of evaluation, but also report results using several alternative measures that are typically used to evaluate the retrieval performance: mean of P[5] and P[10] (mean precision at top 5 or 10 entities returned), and mean R-precision (R-precision for a topic is the $P[R]$, where R is the number of entities that have been judged relevant for the topic). For task 1 all the relevant entities in the relevance assessments are used to generate the scores, while for task 2 we remove the entity examples both from the list of returned answers and from the relevance assessments, as the task is to find entities other than the provided examples.

Task 1 Table 2 shows the performance scores on the training data set for task 1, obtained for Zettair and our three submitted XER runs. Runs 1 and 2 use only the category module ($\alpha = 0.0$, $\beta = 1.0$) while run 3 uses a combination of linkrank, category, and Z scores ($\alpha = 0.1$, $\beta = 0.8$). Runs 2 and 3 use lexical similarity for extending the target categories.

We observe that the three entity ranking runs outperform the plain Zettair run, which suggests that using full-text retrieval alone is not an effective entity ranking strategy. The differences in performance between each of the three runs and Zettair are statistically significant ($p < 0.05$) only for the two entity ranking runs that use lexical similarity between categories (runs 2 and run 3 in Table 2).

When comparing the performances of the runs that use only the category module, we observe that run 2 that uses lexical similarity between category names (TCcat(C)) is more effective than the run that uses the target categories only (cat(C)). With MAP, the difference in performance between the two runs is statistically significant ($p < 0.05$). We also observe that the third run, which uses combined scores coming out from the three modules, performs the best among the three. To find the optimal values for the two combining parameters for this run, we calculated MAP over the 28 topics in the training data set as we varied α from 0 to 1 in steps of 0.1. For each value of α , we also varied β from 0 to $(1 - \alpha)$ in steps of 0.1. We found that the highest MAP score (0.287) is achieved for $\alpha = 0.1$ and $\beta = 0.8$ [10]. This is a 19% relative performance improvement over the best score achieved by using only the category module ($\alpha 0.0$ - $\beta 1.0$). This performance improvement is statistically significant ($p < 0.05$).

Table 3. Performance scores for Zettair and our three XER submitted runs on the training data set (28 topics), obtained for task 2 with different evaluation measures. For each measure, the best performing score is shown in bold.

Run	cat-sim	α	β	P[r]		R-prec	MAP
				5	10		
Zettair	–	–	–	0.229	0.232	0.208	0.172
run 1	cat(E)-cat(t)	1.0	0.0	0.214	0.225	0.229	0.190
run 2	cat(E)-cat(t)	0.0	1.0	0.371	0.325	0.319	0.318
run 3	cat(E)-cat(t)	0.2	0.6	0.500	0.404	0.397	0.377

Task2 Table 3 shows the performance scores on the training data set for task 2, obtained for Zettair and our three submitted XER runs. As with task 1, we again observe that the three entity ranking runs outperform the plain Zettair run. With the first two runs, we want to compare two entity ranking approaches: the first that uses scores coming out from the linkrank module (run 1), and the second that uses scores coming out from the category module (run 2). We observe that using categories is substantially more effective than using the linkrank scores. With MAP, the difference in performance between the two runs is statistically significant ($p < 0.05$).

Run 3 combines the scores coming out from the three modules. To find the optimal values for the two combining parameters for this run, we again varied the values for parameters α and β and we found that the highest MAP score (0.377) was achieved for $\alpha = 0.2$ and $\beta = 0.6$ [8]. This is a 19% relative performance improvement over the best score achieved by using only the category module. This performance improvement is statistically significant ($p < 0.05$).

XER testing data set (73 topics)

Runs description Table 4 lists the six XER and four ad hoc runs that we submitted for evaluation in the INEX 2007 XER and ad hoc tracks, respectively. With the exception of the plain Zettair run, all the runs were created by using our entity ranking system. However, as seen in the table the runs use various parameters whose values are mainly dependent on the task. Specifically, runs differ depending on whether (or which) Zettair category index is used, which of the two types of link contexts is used, whether categories or example entities are used from the topic, and which combination of values is assigned to the α and β parameters.

For example, the run “run 3”, which was submitted for evaluation in task 1 of the INEX 2007 XER track, can be interpreted as follows. The Wikipedia full-text Zettair index is used to extract the top 20 ranked Wikipedia pages, using the title from the INEX topic as a query. After extracting all links to potential answer entities from these 20 pages, the Zettair index of category names is used

Table 4. List of six XER and four ad hoc runs submitted for evaluation in the INEX 2007 XER and ad hoc tracks, respectively. “Cat-sim” stands for category similarity, “Ctx” for context, “Cat” for categories, “Ent” for entities, “T” for title, “TC” for title and categories, “C” for category names, “CE” for category and entity names, “FC” for full page context, and “EC” for element context.

Run ID	cat-sim	α	β	Category index			Topic		
				Query	Type	M	Ctx	Cat	Ent
Zettair		-	-	-	-	-	-	-	-
XER task 1									
run 1	cat(C)-cat(t)	0.0	1.0	-	-	-	FC	Yes	No
run 2	TCcat(C)-cat(t)	0.0	1.0	TC	C	5	FC	Yes	No
run 3	TCcat(C)-cat(t)	0.1	0.8	TC	C	5	FC	Yes	No
XER task 2									
run 1	cat(E)-cat(t)	1.0	0.0	-	-	-	EC	No	Yes
run 2	cat(E)-cat(t)	0.0	1.0	-	-	-	EC	No	Yes
run 3	cat(E)-cat(t)	0.2	0.6	-	-	-	EC	No	Yes
Ad hoc retrieval task									
run 1	Tcat(C)-cat(t)	0.0	0.0	T	CE	10	FC	No	No
run 2	Tcat(C)-cat(t)	1.0	0.0	T	CE	10	FC	No	No
run 3	Tcat(C)-cat(t)	0.0	1.0	T	CE	10	FC	No	No

to extract the top five ranked categories, using both the title and the category names (TC) from the INEX topic as a query. This set of five categories is used as an input set of target categories by the category module. The full page context (FC) is used to calculate the scores in the linkrank module. The final scores for answer entities are calculated by combining the scores coming out of the three modules ($\alpha = 0.1$, $\beta = 0.8$).

Results Results for XER task 1 and task 2 on the testing data set will be reported when they become available. The results obtained for our runs will also be compared with the results obtained for runs submitted by other track participants.

4.2 Ad hoc data set (99 topics)

There are no target categories and example entities provided for the ad hoc task. However, we wanted to apply our algorithm to test 1) whether some indication of the page categories would improve the retrieval performance, and 2) whether extracting new entities from the pages returned by Zettair would be beneficial for ad hoc retrieval.

We submitted four runs for the INEX 2007 ad hoc track: Zettair, representing a full-text retrieval run, and three entity ranking runs. As shown in Table 4, run 1 uses only the Z module for ranking the answer entities, run 2 uses only the linkrank module, while run3 uses only the category module. For each INEX 2007

Table 5. Performance scores for Zettair and our three XER submitted runs on the ad hoc data set (99 topics), obtained with different evaluation measures. For each measure, the best performing score is shown in bold.

Run	α	β	P[r]		R-prec	MAP	Foc	RiC	BiC
			5	10			iP[0.01R]	MAgP	MAgP
Zettair	-	-	0.513	0.469	0.326	0.292	0.379	0.088	0.195
run 1	0.0	0.0	0.513	0.469	0.303	0.247	0.379	0.075	0.165
run 2	1.0	0.0	0.339	0.289	0.170	0.121	0.235	0.031	0.070
run 3	0.0	1.0	0.406	0.368	0.208	0.157	0.287	0.050	0.115

ad hoc topic, we create the set of target categories by sending the title T of the query to the Zettair index of categories that has been created by using the names of the categories and the names of all their attached entities as corresponding documents.

Table 5 shows the performance scores on INEX 2007 the ad hoc data set, obtained for Zettair and our three submitted entity ranking runs. Two retrieval scenarios are distinguished in the table: a *document retrieval* scenario (the first four result columns in Table 5), where we compare how well the runs retrieve relevant documents; and a *focused retrieval* scenario (the last three result columns in Table 5), where we compare how well the runs retrieve relevant information within documents.

For the document retrieval scenario, we observe that Zettair outperforms the other three XER runs. The differences in performance between Zettair and any of these three runs are statistically significant ($p < 0.05$). Among the three XER runs, the run that only uses the Z scores performs significantly better than the other two, followed by the run that only uses the category scores which in turn performs significantly better than the worst performing run that only uses the linkrank scores.

The same trend among the four runs is observed across the three sub-tasks of the focused retrieval scenario, where again Zettair is able to better identify and retrieve the relevant information compared to the other three XER runs.

The obvious conclusion of our ad hoc experiments is that Zettair, which is especially designed for ad hoc retrieval, performs better than our entity ranking system specifically designed for entity retrieval.

5 Conclusion and future work

We have presented our entity ranking system for the INEX Wikipedia XML document collection which is based on exploiting the interesting structural and semantic properties of the collection. On the training data, we have shown that our system outperforms the full text search engine in the task of ranking entities.

On the other hand, using our entity ranking system for ad-hoc retrieval did not result in any improvement over the full-text search engine. This confirms

our hypothesis that that tasks of ad hoc retrieval and entity ranking are very different. Once the official results for the INEX 2007 XML entity ranking track are available, we will make further analysis and compare the effectiveness of our entity ranking system to those achieved by other participating systems.

Acknowledgements

Part of this work was completed while James Thom was visiting INRIA in 2007.

References

1. D. Awang Iskandar, J. Pehcevski, J. A. Thom, and S. M. M. Tahaghoghi. Social media retrieval using image features and structured text. In *Comparative Evaluation of XML Information Retrieval Systems: Fifth Workshop of the INitiative for the Evaluation of XML Retrieval, INEX 2006*, volume 4518 of *Lecture Notes in Computer Science*, pages 358–372, 2007.
2. E. Blanchard, P. Kuntz, M. Harzallah, and H. Briand. A tree-based similarity for evaluating concept proximities in an ontology. In *Proceedings of 10th conference of the International Federation of Classification Societies*, pages 3–11, Ljubljana, Slovenia, 2006.
3. A. P. de Vries, J. A. Thom, A.-M. Vercoustre, N. Craswell, and M. Lalmas. INEX 2007 Entity ranking track guidelines. In *INEX 2007 Workshop Pre-Proceedings, 2007* (to appear).
4. L. Denoyer and P. Gallinari. The Wikipedia XML corpus. *SIGIR Forum*, 40(1):64–69, 2006.
5. S. Fissaha Adafre, M. de Rijke, and E. T. K. Sang. Entity retrieval. In *Proceedings of International Conference on Recent Advances in Natural Language Processing (RANLP - 2007), September 27-29, Borovets, Bulgaria, 2007*.
6. J. M. Kleinberg. Authoritative sources in hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.
7. C. Middleton and R. Baeza-Yates. A comparison of open source search engines. Technical report, Universitat Pompeu Fabra, Barcelona, Spain, 2007. <http://wrg.upf.edu/WRG/dctos/Middleton-Baeza.pdf>.
8. J. Pehcevski, A.-M. Vercoustre, and J. A. Thom. Exploiting locality of Wikipedia links in entity ranking. Submitted for publication, 2007.
9. I. Soboroff, A. P. de Vries, and N. Craswell. Overview of the TREC 2006 Enterprise track. In *Proceedings of the Fifteenth Text REtrieval Conference (TREC 2006)*, pages 32–51, 2006.
10. J. A. Thom, J. Pehcevski, and A.-M. Vercoustre. Use of Wikipedia categories in entity ranking. In *Proceedings of the 12th Australasian Document Computing Symposium*, Melbourne, Australia, 2007 (to appear).
11. A.-M. Vercoustre, J. A. Thom, and J. Pehcevski. Entity ranking in Wikipedia. In *Proceedings of the 23rd Annual ACM Symposium on Applied Computing (SAC08)*, Fortaleza, Brazil, 2008 (to appear).
12. J. Yu, J. A. Thom, and A. Tam. Ontology evaluation using Wikipedia categories for browsing. In *Proceedings of Sixteenth ACM Conference on Information and Knowledge Management (CIKM '07)*, Lisboa, Portugal, 2007.

Integrating Document Features for Entity Ranking

Jianhan Zhu, Dawei Song, Stefan Ruger

Knowledge Media Institute and Centre for Research in Computing
The Open University, United Kingdom
{j.zhu, d.song, s.rueger} @open.ac.uk

Abstract. The Multimedia and Information Systems group at the Knowledge Media Institute of the Open University participated in the entity ranking and entity list completion tasks of the Entity Ranking Track in INEX 2007. In both the entity ranking and entity list completion tasks, we have considered document relevance to query, categorization of documents, category name relevance to query, and hierarchical relations between categories. In addition, based on our success in TREC2006 and 2007 expert search approach, we have applied our co-occurrence based entity association model to the two tasks based on the assumption that the entity often co-occur with query terms in documents containing the entity.

Keywords: co-occurrence, relevance, entity retrieval, entity ranking

1 Introduction

In this year's Entity Ranking Track, there are two related tasks, i.e., entity ranking and entity list completion, on the Wikipedia dataset. A special feature of the Wikipedia dataset is that each document corresponds to an entity. Given a query topic, the aim of entity ranking is to find a list of entities that are relevant to the query topic. A category as part of the query topic specifies the type of entities that should be returned. Some entities have been labeled with certain categories in the dataset. Since entity labeling has been done collaboratively and voluntarily mostly by end users, there is no guarantee that all entities are labeled, entities are correctly labeled. Therefore, the category specification can only be used as a guideline. There are four types of entities that are potentially relevant to a query topic in terms of their categorization. First, the entities are labeled with the specified category. Second, the entities are labeled with categories related to the specified category. Third, the entities are not labeled with neither the specified categories nor any category related to the specified category. Fourth, the entities are not labeled.

The Entity Ranking Track is related to the Expert Search task in the TREC (Text REtrieval Conference) 2005, 2006, and 2007 Enterprise Search tracks [1][2][3]. Given a query topic, the aim of expert search is to find a ranked list of experts from a complete list of candidates in an organization or domain. We have successfully used a two-stage model in expert search in TREC2006 and 2007 Expert Search tasks. The two-stage model consists of a document relevance model where a number of

documents relevant to the query topic are discovered, and a co-occurrence model where experts' relevance to the query topic are measured by their co-occurrence with query terms in a text window in these relevant documents. The two-stage model is also compatible with how the users search for experts on the web, i.e., they find relevant documents on a topic through a search engine, and then read these documents in order to find out experts in these documents.

Entity ranking is more general than expert search since in entity ranking, entities of any types can be retrieved for a topic. The nature of Wikipedia dataset makes the entity ranking track different from expert search task, since in entity ranking each document corresponds to an entity while in expert search expert names are mentioned in documents and named entity recognition tools need to be employed in identifying these occurrences of expert names.

Entity list completion can be seen as a special case of entity ranking task. In entity list completion, a few number of entities relevant to a query topic are given. This list can be used as a clue for finding other relevant entities. There are mainly two ways for using this list. First, use these entities and their corresponding documents as relevance feedback information. Second, based on the observation that these entities may often co-occur with other entities that are also relevant to the query topic, we can use a co-occurrence model to measure the relevance between new entities and entities in the list.

We think that entity ranking is sensitive to multiple document features that need to be taken into account in entity ranking. The document features we consider include: 1. Document content based relevance to the query topic, 2. Specified category in the query topic, 3. Sub-categories and parents of the specified category, and 4. The content based relevance of category names of each document to the query topic. In TREC 2006 and 2007, we have considered multiple levels of associations between experts and a query topic by using a multiple-window based approach [4][5]. Similarly, we have applied the multiple-window based approach to entity ranking. Entities are mentioned in other documents. The contexts of these occurrences of entities often include query terms in the query topic. We combine the relevance measures based on multiple document features in entity ranking. In entity list completion, we have considered the co-occurrence of entities in the list and new entities.

The rest of the paper is organized as follows. In Section 2, we introduce our work on entity ranking. We extend our entity ranking approach for entity list completion in Section 3. We report our experimental results on Wikipedia dataset in Section 4. We conclude and discuss future work in Section 5.

2 ENTITY RANKING

For each document, we will use its content based relevance to the query topic as the baseline model. We enhance the baseline model by taking into account its categories' relations with the specified category, its categories' content based relevance to the query topic, and the entity's co-occurrences with the query terms in other documents.

2.1 Content based Relevance

If an entity is relevant to a topic, the content of the document representing the entity is likely to contain keywords in the query topic. We can use standard relevance models for judging the relevance of the document content to the topic. Probabilistic models such as BM25, Boolean models, and language models can be applied.

2.2 Entity's Categories

An entity's category information can help entity retrieval in mainly three aspects.

First, since a preferred category is specified as part of a query topic, if there is a match between an entity's categories and the preferred category, the relevance of the entity to the query topic will be largely boosted.

Second, since the categorization of entities is not done completely and the preferred category specification may not cover all relevant entities, we need to find categories which are relevant to the preferred category. In our approach, we experimented with finding the sub-categories and parents of the preferred category, if there is a match between these categories and an entity's categories, the relevance of the entity to the query topic will be boosted.

In the hierarchy of categories for the Wikipedia dataset, the links between categories sometimes do not always represent a "containment" relationship between two categories, i.e., the child may not be a sub-class of the parent sometimes. In order to avoid the "concept drift" in the hierarchy, categories related to the preferred category are only limited to its parents and children in our work, although we will investigate the effect of incorporating more distantly linked categories as the next step.

Third, if an entity is relevant to a query topic, the contents of the entity's categories can often contain keywords in the query topic. Therefore, we join the categories' names of an entity as a separately metadata field about the entity, and calculate the relevance between this metadata field and the query topic.

We can envisage that the availability of categorization associated with the Wikipedia dataset will significantly assist entity retrieval. The assumption can be tested based on an anatomy of our entity retrieval system studying the effect of multiple document features in entity retrieval.

2.3 Entity's Co-Occurrences with Query Terms

So far, entity ranking task is similar to a document ranking problem, i.e., judging the relevance between a number of documents and a query topic and produced a ranked list of documents. However, due to the nature that each document corresponds to an entity, we can introduce a separate component to the entity ranking task which is based on the context information of each entity in other documents which mention the entity.

This entity context based component is very similar to the expert search task in TREC in the sense that many expert search approaches have taken into account the contextual information of experts in documents for expert search. Similarly, each entity occurs in a number of documents, and the contexts of these occurrences can give us clue about how relevant of the entity is to the query topic.

In TREC2006 and 2007, we have successfully employed a novel two-stage multiple window based approach for expert search. We have applied the two-stage model to the entity ranking task as follows. The two stages are a document search stage where documents relevant to the query topic are retrieved, and in the second stage, an entity's relevance to the query topic is judged based on the co-occurrences of the entity and query topic terms in a text window in these relevant documents.

Since entity's association with a query topic can be of multiple levels, from phrase, sentence, paragraph, up to document levels, we propose a novel multiple window based approach to capture all these levels of associations. We assume that smaller text windows lead to more accurate associations and larger windows may introduce noise thus leading to less accurate associations. Therefore, we take a weighted sum of the relevance between an entity and a topic based on a number of text windows, where smaller windows are given higher weights and larger windows are given lower weights.

3 ENTITY LIST COMPLETION

Entity list completion can be seen as a special case of entity ranking where a few given relevant entities are given as relevance feedback information. We have incorporated the given relevant entities in our two-stage approach. We assume that entities that are relevant to the query topic tend to co-occur often with the given entities in documents. Again, we adopted the novel multiple-window based approach for integrating association of multiple levels between an entity and any of the given entity.

4 EXPERIMENTAL RESULTS ON WIKIPEDIA DATASET

We have pre-processed the dataset by removing HTML tags. We trained our system on the TREC2006 expert search test collection, and applied our approach to the Wikipedia dataset. We are still carrying out collaborative evaluation of both entity ranking and list completion tasks. The detailed experimental results will be reported in the final version of this paper

5 CONCLUSIONS

We have participated in both entity ranking and list completion tasks in INEX2007. Based on the assumption that entity ranking is sensitive to multiple document features, we propose a novel approach for integrating multiple document features for effective entity ranking. In our approach, we have considered the content of the document describing an entity, matching between the entity's categories and the preferred category, the effect of hierarchical relations between categories, and the content of categories. In addition, we apply our winning approach in TREC 2006 expert search task, i.e., a multiple-window-based two stage model, for integrating multiple levels of associations between an entity and a query topic. We treat entity list completion as a special case for entity ranking by using the given list of relevant entities as relevance feedback information for incorporation into our multiple-window-based two stage model.

ACKNOWLEDGEMENTS

The work reported in this paper is funded in part by an IBM 2007 UIMA innovation award and the JISC (Joint Information Systems Committee) funded DYNIX (Metadata-based DYNAmic Query Interface for Cross(X)-searching content resources) project.

References

- [1] Bailey, P., Craswell, N., de Vries, A.P., and Soboroff, I.(2007) Overview of the TREC 2007 Enterprise Track (DRAFT). In Proc. of The Sixteenth Text REtrieval Conference (TREC 2007), Gaithersburg, Maryland USA.
- [2] Craswell, N., de Vries, A.P., Soboroff, I. (2005) Overview of the TREC-2005 Enterprise Track. In Proc. of The Fourteenth Text REtrieval Conference (TREC 2005).
- [3] Soboroff, I., de Vries, A.P. and Craswell, N. (2007) Overview of the TREC 2006 Enterprise Track. In Proc. of The Fifteenth Text REtrieval Conference (TREC 2006), Gaithersburg, Maryland USA.
- [4] Zhu, J., Song, D., R ger, S., Eisenstadt, M. and Motta, E. (2007) The Open University at TREC 2006 Enterprise Track Expert Search Task. In Proc. of The Fifteenth Text REtrieval Conference (TREC 2006).
- [5] Zhu, J., Song, D., R ger, S., Eisenstadt, M. and Motta, E. (2007) The Open University at TREC 2006 Enterprise Track Expert Search Task. In Proc. of The Sixteenth Text REtrieval Conference (TREC 2007) Notebook.

L3S Research Center at the INEX Entity Ranking Track

Gianluca Demartini, Claudiu S. Firan, Tereza Iofciu

L3S Research Center
Leibniz Universität Hannover
Appelstrasse 9a D-30167 Hannover, Germany
{demartini, firan, iofciu}@l3s.de

Abstract. Entity ranking on Web scale data scale is still an open challenge. Wikipedia-based ontologies can be used to improve the quality of the entity ranking produced by a system. In this paper we propose algorithms based on Query Relaxation using categories information to rank entities in Wikipedia. Our approach focuses on constructing the queries using not only the keywords from the topic, but also information about relevant categories leveraging on a highly accurate ontology. The evaluation is performed using the XML Wikipedia collection and the INEX 2007 Entity ranking topics. The results show that our approach performs effectively, especially for queries where the relevant entities are not consistently categorized in the Wikipedia articles.

1 Introduction

Entity search is becoming an important step over the classical document search as it is done today on the Web. The goal is to find entities relevant to a query more than just finding documents (or parts of documents) which contain relevant information. Ranking entities according to their relevance to a given query is crucial in scenarios where the amount of information is too big to be managed by the final user. That is, a correct ranking can help the systems in presenting the user only with entities of interest, and avoiding the user to analyse the entire set of retrieved entities.

In this paper we present our approach on how to rank entities in Wikipedia and we evaluate it on the Wikipedia XML corpus provided within the INEX 2007 initiative and we investigate how the extended category information influences the results.

The rest of the paper is organized as follow. In section 2 we describe the general architecture of the developed system. In section 3 we present an ontology, based on Wikipedia and WordNet, that we use to improve the effectiveness of the entity ranking algorithms. In section 4 we present the structure of the generated inverted index for the XML Wikipedia collection. In section 5 we formalise the ranking algorithms we propose. In section 6 we present the evaluation results and the comparison among the proposed algorithms. In section 7 we present and compare the previous approaches in entity search and ranking. Finally, in section 8 and 9 we describe the future improvements and we conclude the paper.

2 Architecture

In this section we describe the architecture of the Entity Ranking System we used to create the runs submitted to INEX 2007.

The architecture design is presented in figure 1. The first step is the creation of the inverted index for the XML Wikipedia document collection. Starting from the raw structured XML documents, we created a Lucene index with one Lucene document for each Wikipedia document (see more details in section 4).

After the creation of the index, the system can process the INEX Entity Rank 2007 topics. Two different approaches are adopted (see details in section 5): the INEX topic is first processed in order to create a disjunctive Lucene query using the Title and Description information of the Topic; a possible extension is done using the Category field of the Topic together with information from the YAGO[5] ontology (see figure 3) in addition to the Lucene query obtained after this first step.

After the generation of the Lucene query, the index can be queried and a ranked list of retrieved entities can be generated as output of the system.

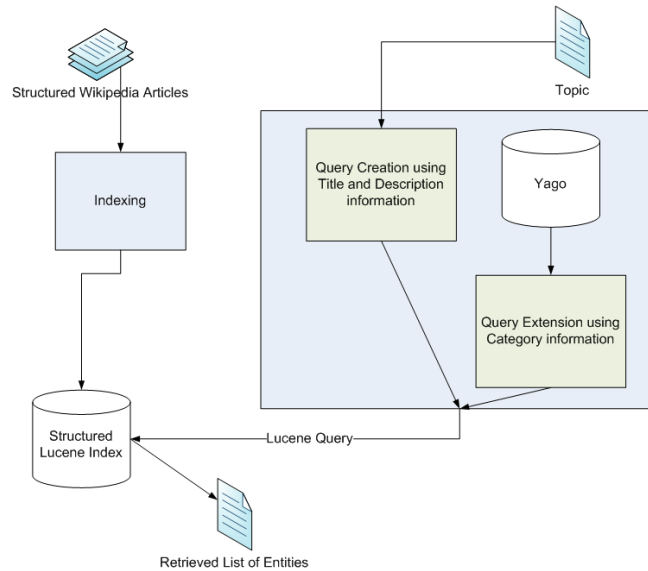


Fig. 1. Architecture of the Entity Ranking System

3 Using YAGO

YAGO¹ is a large and extensible ontology that builds on entities and relations from Wikipedia. Facts in YAGO have been automatically extracted from Wikipedia and unified with WordNet², using rule-based and heuristic methods. It contains more than 1 million entities and 5 million facts and achieves an accuracy of about 95%. All objects (e.g. cities, people, even URLs) are represented as entities in the YAGO model. The ontology is constructed in such a way as to be able to express entities, facts (the triple of an entity, a relation and an entity is called a fact), and even relations between facts and properties of relations.

The creation of YAGO focuses on integrating entities from Wikipedia with semantics from WordNet. Each Wikipedia page title is a candidate to become an entity in YAGO, and the Wikipedia categories of that page become its containing classes. Wikipedia categories are organized in a directed acyclic graph, which yields a hierarchy of categories. This hierarchy, however, reflects merely the thematic structure of the Wikipedia pages. Thus, WordNet is used to establish the hierarchy of classes, as WordNet offers an ontologically well-defined taxonomy of synsets. Each synset of WordNet becomes a class of YAGO and the *subClassOf* hierarchy of classes is taken from the hyponymy relation from WordNet. This gives for each Wikipedia page a set of conceptual categories arranged in a taxonomic hierarchy. More data is gathered exploiting WordNet synsets as synonyms and exploiting Wikipedia redirects as alternative names for the entities.

For the purpose of this article we used the MySQL export of YAGO and combined it with the INEX Wikipedia crawl. This allows us to make use of the *subClassOf* relation in YAGO, providing us with semantic concepts describing Wikipedia entities. E.g. knowing from Wikipedia that 'Married... with Children' is in the category 'Sitcoms', we reason using YAGO's WordNet knowledge that it is of the type 'Situation Comedy', same as 'BBC Television Sitcoms', 'Latino Sitcoms', 'Sitcoms in Canada', and 8 more. We also find that not all of the subcategories in Wikipedia are of the same type as the parent category, and can thus filter some out. E.g. the Wikipedia category 'Sitcoms' which is of WordNet type 'Situation Comedy' contains the subcategory 'Sitcom Characters' which is of WordNet type 'Fictional Character'.

4 Index Structure

Given the XML document collection, we created an entity-driven inverted index in order to enable the search and ranking of entities. We have chosen to use a Lucene index³ because of the possibility of generating a "structured" inverted index with fields which are searchable in parallel.

¹ Available for download at <http://www.mpii.mpg.de/~suchanek/yago>

² <http://wordnet.princeton.edu/>

³ <http://lucene.apache.org>

We followed the structure of the XML Wikipedia documents in the creation of the index. For each document, we store its content divided in the following fields:

- *id*, the unique identifier of the article;
- *title*, the title of the article;
- *text*, the entire textual content of the article;
- *emph*, the parts of the article which are emphasized;
- *categories*, the categories listed at the bottom of the article;
- *wikiLinks*, the links which point to other Wikipedia articles, with anchor text and target page;
- *webLinks*, the links which point to external Web pages, with anchor text and target page;
- *figures*, the text related to the figures in the articles;
- *sections*, the content of the article splitted into sections;

Moreover, we store a stemmed and stopped version of the text and title fields that we used during search.

5 Algorithms

We have implement two approaches for entity ranking. Both approaches are based on the extended vector space model, which has more enriched semantic information than traditional TF-IDF model. For both approaches we keep two main vectors, one for the textual information and one for the context, such as category information.

5.1 Naive approach

In the baseline approach we consider only the information given in the topic at query time. We construct a boolean query with both textual information and contextual information. For the textual part of the query we consider the keywords from the title and the description of the topic⁴ which we run against the title and text in the Wikipedia pages. In the contextual part of the query we consider the category information from the topic which we run against the category vector, we do not make this part of the query mandatory as the category information available in Wikipedia is not always true or present. The rank of Wikipedia retrieved entities is higher for the ones where there is a category match.

⁴ The narrative part containing too many non-specific keywords that might over-relax the query is not included

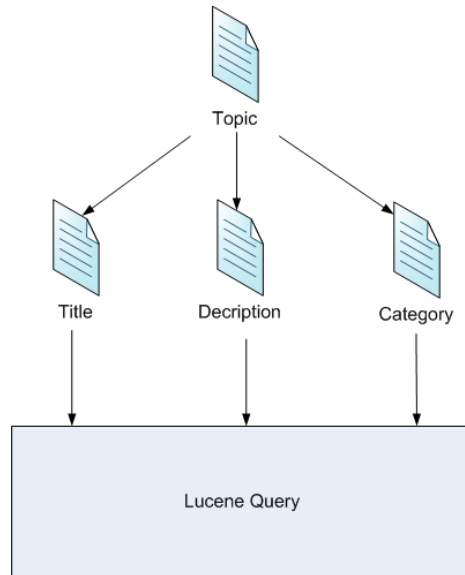


Fig. 2. Query Construction using Topic Information

5.2 Categories based search

While the category contained in the topic should contain most or all of the retrievable entities, this is in many topics not the case. Wikipedia is constructed manually by different contributors, so that the category assignments are not always consistent. Many categories are very similar and in some of these cases the difference is very subtle so that very similar entities are sometimes placed in different categories by different contributors (e.g., hybrid powered automobiles are either in the 'hybrid vehicles' or the 'hybrid cars' category, inconsistently, and very seldom they are in both).

In the previous approach the given category in the topic was used to make the query more likely to retrieve entities from within that category. The method described here constructs an additional list of categories closely linked to the given one in the topic description. This extended list of categories is then used instead of the one category in query construction. We use two types of category expansion, 'children' and 'siblings'.

Children. Wikipedia itself has a hierarchical structure of categories. For each category we are presented with a list of subcategories. These subcategories are the initial list of candidates for 'children'. We cannot include all the Wikipedia subcategories in our 'children' list as some of them are not a real subcategory, they are not of the same type. We can have as subcategories for a country categories about presidents, movie stars, or other important persons for that country. This means that although we have as a starting category a country we end up having people as subcategories, which is not what we want in the entity retrieval context. The solution to this is selecting only those subcategories having the same class as the initial category. As described in Section 3, YAGO contains

also class information about categories. We will make use of this *subClassOf* information to identify suitable categories of the same kind. Thus, a Wikipedia subcategory is included in the 'children' list only if the intersection between its ancestor classes and the ancestor classes in YAGO (excluding top classes like *entity*) of the initial category is not empty. The final list of 'children' will therefore contain only subcategories of the same type as the given category in the topic.

Siblings. Also using YAGO we can retrieve categories of the same type as one starting category, not restricting just to the Wikipedia subcategories. We first determine the type of the starting category using the *subClassOf* relation in YAGO. Knowing this type we construct a list of all categories of the same type and add them to the 'siblings' set. 'Siblings' are thus all categories of the exact same type as the initial category.

Figure 3 depicts the inclusion of 'children' and 'siblings' in the query creation process. Constructing the query is done similar to the 'Naive approach' setting. The difference relies in the category matching part. In the 'Naive approach' we had only one category (given with the topic) while here we have the additional two lists of 'children' and 'siblings'. These two related categories creation method allow us to analyze how good the given category in the topic is suitable for entity retrieval.

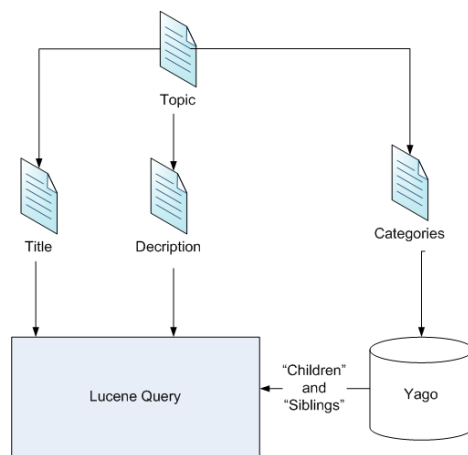


Fig. 3. Query Construction using YAGO Category Information

6 Results

We performed evaluation experiments of our system using the 28 training topics, used for the INEX 2007 Entity Ranking Track, which were derived from the

INEX 2006 topics and assessments. We note that this dataset and, in particular, the relevance assessments, were not done with the idea of a user searching for entities, but they are a selection of INEX 2006 ad hoc topics thus influencing the effectiveness metrics values. This is also shown by the high values of the bpref metrics which are computed only using the assessed entities.

We performed the experiments on the evaluation dataset and we observed that a comparison of the two proposed algorithms shows that using information about the category structure we obtain 10% improvement over the baseline in the Mean Average Precision (MAP) value.

The results (presented in table 1) show that for some specific queries (e.g. “types of bridges”, topic 23) the system is performing reasonably good (i.e. MAP 0.56 with the naive approach). In general, the system based on the naive approach obtains a bpref value of 0.14 while the categories based search system obtains a bpref value of 0.20.

Algorithms	Topic	MAP	$P@15$	bpref
baseline	23	0.5550	0.7333	0.6748
extended	23	0.5122	0.6667	0.8076
baseline	all	0.1008	0.1238	0.1398
extended	all	0.1107	0.1310	0.2014

Table 1. Results of evaluation

Another observation we can do is that the YAGO ontology is up-to-date and does not match some of the categories present in the XML Wikipedia dataset from 2005 used in the experiments, and thus the evaluation assessments might not consider relevant information which is present today in YAGO. In the case of some topics, the results computed by the categories based search are only as good as the ones computed by the naive approach.

7 Related Work

There are only a few systems that deal with entity search and ranking. ESTER, presented in [1], is a system which combines full-text and ontology search and supports prefix search and joins. They have applied to the English Wikipedia and as ontology they have used YAGO[5]. From the ontology they have used the *is a* and *subclass of* relations. ESTER focuses on efficiency, the recall is high while the precision is reasonable and it is higher when using only the Wikipedia data, without the additional information from the ontology.

Another framework focusing on effectiveness and efficiency, presented in [2], focuses on finding different types of entities (e.g. phone number, email..) on the Web. While their main accent is on scaling on Web-size dataset, we can better manage an heterogeneous set of entity types.

A related field is the one of Expert Search (ES) where the aim is to find people (i.e. a specific type of entity) who are experts on the given topic. The topic of

ES is a relatively new one but already several systems have been proposed. The systems proposed in the past use several information and features like Social Network information; co-occurrences of terms and changes in the competencies of the people; rule-based models and FOAF⁵ data; and using post on Web Forums [6]. One of the first approaches is the Enterprise PeopleFinder [4] also known as P@noptic Expert [3]. This system first builds a candidate profile attaching all documents related to her/him, giving different weights to the documents based on their type (e.g. an homepage is more important than other web pages), in one big document which represents the candidate. The problem of this systems is that it can only consider the terms of the documents as topics of expertise and that the candidate name matching (i.e. the name appears into the document or not) and the relationship between candidate and documents is only binary (i.e. a document is related to the candidate or it is not).

8 Future Work

There are a couple more approaches that we would like to investigate in the future. A first approach deals with the disambiguating of the query topic. This can be done either by extracting adjectives and nouns from the topic's title, description and narrative. A more complex approach would be, for a given topic, to extract the existing Wikipedia entities that are specified in the topic and see if these entities or entities linking to them belong to the topic's categories.

As in the Wikipedia pages one can often find lists of other entities, another approach to automatically enrich the category information would be assume that if the majority of entities in a list belong to a category, then the rest of entities in the list could be in the same category, or a new category could be created from the list's name.

9 Conclusions

In this paper we proposed two algorithms to rank entities in Wikipedia. Our approach uses a structured inverted index to represent the entities which are present in Wikipedia and uses the YAGO ontology in order to rewrite the user's query for improving the effectiveness of the results. The evaluation experiments shows that, especially for certain types of queries, our approach works well. The categories based search obtains a 10% improvements of the effectiveness (computed as MAP) over the naive approach. We will try to improve even more the effectiveness of our approach using disambiguation techniques and using the entity link structure information.

Acknowledgments. This work was partially supported by the Nepomuk and Pharos projects funded by the European Commission under the 6th Framework Programme (IST Contract No. 027705 and No. 045035).

⁵ <http://www.foaf-project.org/>

References

1. Holger Bast, Alexandru Chitea, Fabian Suchanek, and Ingmar Weber. Ester: efficient search on text, entities, and relations. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 671–678, New York, NY, USA, 2007. ACM.
2. Tao Cheng, Xifeng Yan, and Kevin Chen-Chuan Chang. Entityrank: Searching entities directly and holistically. In *VLDB*, pages 387–398, 2007.
3. N. Craswell, D. Hawking, A. Vercoustre, and P. Wilkins. P@noptic Expert: Searching for Experts not just for Documents. *Ausweb, 2001*.
4. A. McLean, A.M. Vercoustre, and M. Wu. Enterprise PeopleFinder: Combining Evidence from Web Pages and Corporate Data. *Proceedings of Australian Document Computing Symposium, 2003*.
5. F.M. Suchanek, G. Kasneci, and G. Weikum. Yago: a core of semantic knowledge. *Proceedings of the 16th international conference on World Wide Web*, pages 697–706, 2007.
6. J. Zhang, M.S. Ackerman, and L. Adamic. Expertise networks in online communities: structure and algorithms. *Proceedings of the 16th international conference on World Wide Web*, pages 221–230, 2007.

Entity Ranking using XML Retrieval Techniques

M. S. Ali, Mariano P. Consens, Shahan Khatchadourian

University of Toronto, Canada

{sali, consens, shahan}@cs.toronto.edu

Abstract—Entities are identifiable objects; as an example, “Toyota Insight” is an entity of the nonentity “hybrid cars”. The system output of an entity is the article-level element. We apply XML retrieval techniques using substructure indexes to address the INEX Entity Track challenge of entity identification and ranking. Results are filtered using topic and sample entity category neighbourhoods in the category graph. We use Apache Lucene as our search engine. Results will be presented once track assessments are completed.

Retrieval of document parts using Bayesian Networks and entropy as a degree of (dis)organization

Carlos Estombelo-Montesco, Douglas Chiodi, Taciana Kudo, Adolfo Seca Neto, Fernando Pigearde de Almeida Prado, Alessandra Alaniz Macedo

Department of Physics and Mathematics, FFCLRP,
University of Sao Paulo, Ribeirao Preto, SP, Brazil, 14040-901
estombelo@gmail.com douglaschiodi@gmail.com taciana.novo.kudo@gmail.com
adolfo.usp@gmail.com pigearde@ffclrp.usp.br ale.alaniz@usp.br

Abstract. Content-based retrieval systems usually rely on classical models to represent flat documents as a bag of words. Although some models that take advantage of document structure for Information Retrieval (IR) have been proposed, the majority does not deal with organization and structure of documents. When we consider the document structure, we are able to retrieve relevant documents as well as their most pertinent parts. Therefore, the retrieval of document parts may reduce the cognitive overhead of reading the whole document. The most important aspects of these document representations are the description languages that allow document structuring, such as the Extensible Markup Language (XML). Our proposal here is to explore and extend a generic system based on Bayesian networks in which probabilistic inferences are used for performing IR tasks. The novelty is we consider the Tsallis entropy as a measurement of (dis)organization of source data for information retrieval. Our system is being implemented and improved to obtain a more effective information retrieval system.

1 Introduction

Content-based retrieval systems usually rely on classical models to represent flat documents as a bag of words. Although some models that take advantage of document structure for information retrieval have been proposed, the majority does not tackle with the organization and structure of documents. When we take into account document structure (not only flat text of documents), we are able to retrieve relevant documents as well as their most pertinent parts. Therefore, the retrieval of document parts may reduce the cognitive overhead of reading the whole document.

Information retrieval (IR) is an active research area with many challenges [1] that the technological advances in textual and document representations and diversified approaches should be considered.

The most important of these document representations are the description languages that allow document structuring. One of these languages is the Extensible Markup Language (XML) [2]. XML allows a rich description of documents with the incorporation of metadata, annotations and multimedia information in a logical schema: XML Schema. This logical structure provides hierarchical levels of granular-

ity and consequently more precision can be achieved by means of focused retrieval. But it is still a challenge to create retrieval mechanisms for this type of document. Researchers indicated in [3] that a document structure should be treated together with its textual content. In addition, XML retrieval tasks should retrieve document components [4] or document parts, called *doxels* (document elements) [3], instead of retrieving all documents (based on content only). Further information about these tasks and a framework can be found at [5]. We can briefly mention here that in [6] was presented the pioneer proposal relating to document structure and IR, using document sections to improve the performance of IR tasks. Another approach was proposed by [7], where the theory of evidence was the core of the work.

The proposal of this work is to exploit and extend a generic system based on Bayesian networks initially proposed by [3], which aims to perform different IR tasks on collections of structured documents. Our work is still preserving a Bayesian networks approach, where probabilistic inferences are used to carry IR tasks out. We assume the same simplifications made by the authors, related to the hierarchical structure of the collection. Our differential here is the type of entropy used. We are considering Tsallis entropy [12] as a measurement of (dis)organization of the source data for information retrieval.

This paper is organized as follows: Section 2 briefly presents the Shakespeare test collection exploited by our proposal and implementation details. Section 3 details a different way to exploit the Tsallis entropy to calculate probabilistic distributions. Section 4 concludes our study.

2 Materials and Methods

Nowadays, collections of structured document are encoded into a structured representation such as XML, RDF (Resource Description Framework) or RSS (Really Simple Syndication), and they are becoming available on the Internet. Consequently, the research community has proposed retrieval methods for structured representations, where this extension is not trivial.

The INEX (Initiative for the XML Retrieval) is part of a large-scale effort to promote the evaluation of content-oriented XML retrieval by providing a large test collection of XML documents and uniform scoring procedures. In this initiative, participating organizations contribute to the construction of a large test collection of XML documents with their ad-hoc relevance. A detailed description of the INEX document collection from 2002 to 2006 can be found at [4].

In particular, since the INEX initiative started, the relevance measure for each element has been modified along years. Modifications from continuous evaluations were based on the impact of the measurements on the new methods proposed by the IR research community. These evaluations have tried to avoid the redundancy and the misinterpretations of relevance measurements. For example, INEX 2003 adopted two dimensions for relevance, exhaustivity and specificity; however, it was very tedious and costly to obtain the relevance assessments [3]. Consequently, the assessment process was simplified, the exhaustivity dimension was dropped, and since INEX 2006 relevance is defined entirely along the specificity dimension.

The specificity dimension is automatically measured on a continuous scale [4], by calculating the ratio of the relevant content of an XML element. For example, if an assessor (when assessing a topic) completely highlights an XML element, it has a specificity value of 1 for that topic. On the other hand, completely non-highlighted elements have a specificity value of 0. For all other elements, the specificity value is defined as a ratio (in characters) of the highlighted text (considered as relevant part) to the XML element size.

2.1 Collection

The collection used here is the Shakespeare test collection consisting of 37 Shakespeare plays, 43 queries and their relevance assessments. Each document set includes its DTD files. The set of documents contains 1,133.297 words distributed in 37 files, and each tag has a unique object ID. This collection is from FOCUS project of the Duisburg University, Information system group. Soon, we will evaluate our system with the INEX 2007 collection.

2.2 Implementation

Here we briefly describe the computational resources implemented. In the storage stage, we built a database whose tables and relations had a generic structure. Thus, we can store any XML collection in this database based only in the DTD tags of the collection documents.

The other implementation tasks were: (a) loading and storing XML documents; (b) computing an Okapi value for each DOXEL; (c) computing the conditional table and probabilities for scoring DOXELS.

The purpose of the load and storage task, depicted in Figure 1, is to load XML files from a predefined directory, and store them into the database for further analysis of each DOXEL. As this procedure is general, we can also store query and judgment documents. As an XML document is represented as a tree, the procedure is recursive.

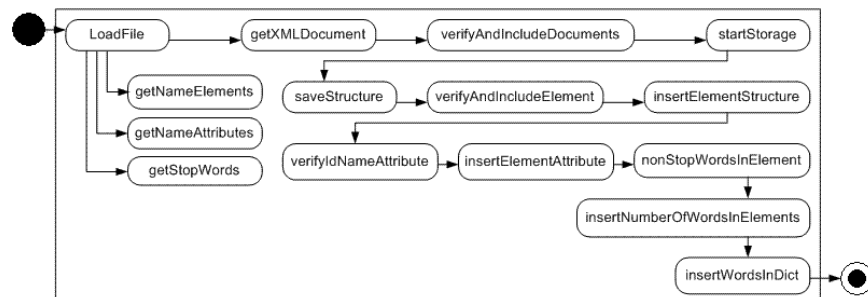


Figure 1 : Loading and storing the XML documents

We have created an array for registering each XML document into the database, therefore if we execute again the storage procedure, only new documents will be loaded and stored.

Before storing each document, a set of steps is carried out over the documents, in order to clean and validate the content of the document. These steps are: i) the definition of an identifier for each DOXEL; (ii) the crawling and storing of each attribute related to a DOXEL; (iii) the elimination of special characters from DOXEL content; (iv) the exclusion of stopwords from DOXEL content; (v) the counting words of each DOXEL; and (vi) the storage into the dictionary table of each new word found in DOXELS.

After executing task (a), an Okapi value for each DOXEL from every query document can be calculated, as depicted in Figure 2.

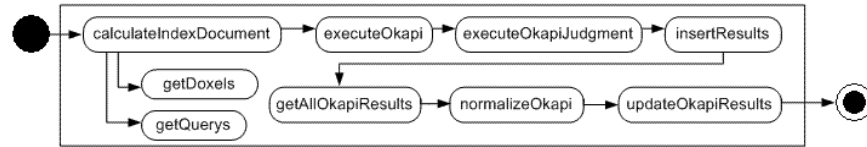


Figure 2 : Computing an Okapi value for each DOXEL

To accomplish this task, the following steps are carried out: i) acquisition of the id key from each DOXEL, ii) acquisition of the id key from each query, iii) for each id key for a DOXEL, calculation of the Okapi value related to the id key from a query, and iv) normalization of the Okapi value as a probability. The computation of the okapi value is supported by the equation proposed in [3] adapted for DOXELS. The okapi value is used for calculating the global score depicted in Figure 3.

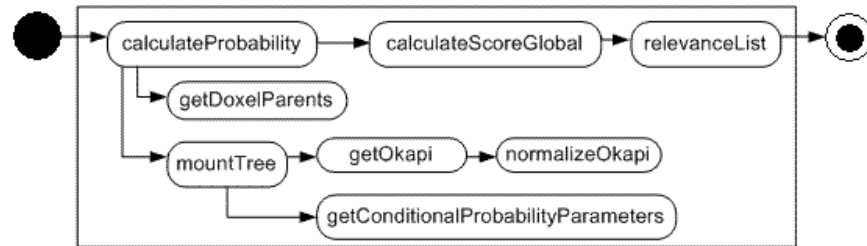


Figure 3 : Compute local and global score

3 Score and retrieval of document parts: a probabilistic approach

The Bayesian network approach [3;8;9] in this work contemplates four subsections: local score estimative of the content, global score by doxel context (based on the document structure), training, and measurement of entropy. All documents (collection, query and judgments) are validated using the same pre-processing function

(depicted in Figure 1) and stored into the database. After that, the training process can be carried out considering three parameters: document collection, queries and judgments.

At this stage a conditional probability is computed by the training stage as well as the probability of relevance for the final inference process.

When a new query is formulated, before computing the scores of the relevant doxels, we need to compute the Okapi value. This Okapi value can be used in conjunction with the conditional probability for inference, and after that we compute the score of each DOXEL. Finally we get a list of DOXEL scores for an input query.

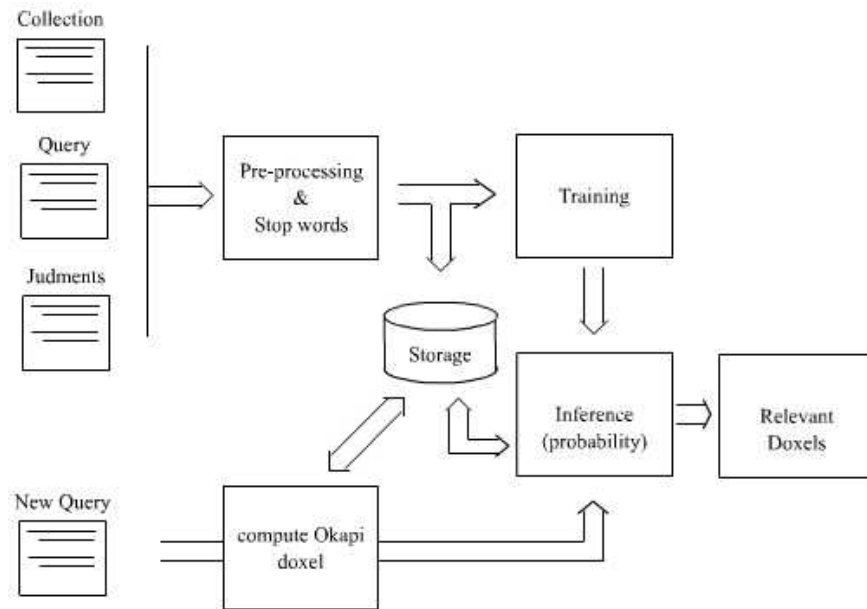


Figure 4 : Bayesian approach and process

A set of parameters depends on the estimative of the content given the local score (where the relations in the Bayesian Network are not taken into account). Another set of parameters is related to each DOXEL content, where relations and dependencies between structural parts of the document play an important role (for example parents, siblings, etc).

3.1 Local score estimative of the content

The estimative of the local scores are computed by the baseline model. It allows to measure the content (flat text) of each DOXEL. After computing the local score, it must be mapped (transformed) into a probability to be used in the global score computation stage. This local score is based on an adaptation of the Okapi model [3].

which is originally adapted for obtaining a reasonable performance on structured collections in general.

The local score used based on Okapi variant is [3]:

$$\text{Okapi}(q, X) = \sum_{j=1}^{\text{length}(q)} \omega_{j,X} \frac{(k_1 + 1)tf_{X,j}}{K_X + tf_{X,j}} \times \frac{(k_3 + 1)qtf_j}{k_3 + qtf_j} \quad (1)$$

Where:

- 1) k_1 and k_3 are constants
- 2) $\text{length}(q)$ is number of terms in query q .
- 3) $\omega_{j,X} = \log\left(\frac{N - n_j + 0.5}{n_j + 0.5}\right)$, where N is the number of doxels in the collection, and n_j the number of doxels containing term j . Amongst the different options for adapting these collection statistics to Structured Information Retrieval, we have chosen to compute these two values as defined with respect to the set of all doxels (“element frequency”) and the classical document set (“document frequency”).
- 4) $K_X = k_1\left((1 - b) + b \frac{dl}{avdl}\right)$, where b is a constant, dl is the doxel length, $avdl$ is the average length over all doxels (“corpus”) or over all the sibling doxels (“parent”).

Our experience with the computation of this local score has shown the need for more than one Okapi variant. The combination of these variants could improve the score of the set of doxels in the final retrieval stage.

3.2 Global score based on doxel context

As already proposed by Callan (1992) [10], Bayesian networks can be an effective IR model [10]. When used on structured representations, it may prove benefits for the retrieval performance. An important characteristic is that the model learns the parameters directly from the data. Therefore, different collections can be used for the same purpose.

The literature shows that a Bayesian network provides a complete description of a domain [8]. The description of a domain is represented by a joint distribution based on conditional probabilities. In this case, for the random variable x_i and using the product rule, we have:

$$P(x_1, \dots, x_n) = P(x_n | x_{n-1}, \dots, x_1) P(x_{n-1}, \dots, x_1) \quad (2)$$

Next, we can repeat the process by reducing each joint probability to one conditional probability and one (minor) conjunction, reducing it to a big product:

$$\begin{aligned} P(x_1, \dots, x_n) &= P(x_n | x_{n-1}, \dots, x_1) P(x_{n-1} | x_{n-2}, \dots, x_1) \dots P(x_2 | x_1) P(x_1) \\ &= \prod_{i=1}^n P(x_i | x_{i-1}, \dots, x_1) \end{aligned} \quad (3)$$

This identity is true for any set of random variables and is called the chain rule. From here, if we consider that assuming the hypothesis of conditional independence it can be easy to build the network topology and we can consider that the joint distribution is equivalent to the general assertion, for any x_i variable:

$$P(x_i | x_{i-1}, \dots, x_1) = P(x_i | Parents(x_i)) \quad (4)$$

where $Parents(x_i) \subseteq \{x_{i-1}, \dots, x_1\}$. Then we can write the Bayesian network as:

$$P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i | Parents(x_i)) \quad (5)$$

Therefore, any input of the joint probability is computed from information stored in the network. A generic input into the joint probability is a joint probability of specific attributions for each variable, such as, $P(X_1 = x_1, \dots, X_n = x_n)$. In our case, we use a simplification of this notation, $P(x_1, \dots, x_n)$ as an abbreviation (see equation (5)).

Equation (4) shows that the Bayesian network is a correct domain representation only if each node is conditionally independent from its predecessors (given his parents). Intuitively, the parents of node x_i should contain all nodes in x_1, \dots, x_{i-1} that have direct influence on x_i . A difficulty arises on the computation of the probabilities. The computation of these parameters is based on the training of the model. The training includes the documents collection, query documents and judgments.

3.3 Training

The training stage is based on the document collection, queries and judgments. The main goal of this stage is to measure and calculate the parameters for the Bayesian network conditional probabilities.

The logical structure of documents is composed by DOXELs, and each one must have a table of conditional probabilities. If we created one table for each DOXEL, the amount of tables would be enormous, and their processing would be very time-consuming. Therefore, a simplified number of conditional probabilities table was con-

sidered as in [3]. This reduced the volume of information need for storing these probabilities. It was made by grouping the elements by their categories.

After simplifications, the number of parameters decreases in quantity. This allows us to train the Bayesian network based on cross-entropy, but the difference is that we use the Tsallis entropy. Tsallis entropy models the instability of the disorganization degree in the information. We have implemented this approach over the Shakespeare collection, and we intend to apply the method over the INEX 2007 collection for further evaluation.

3.4 Measurement of entropy as a complex system

Information theory (IT) has its origin on the probability theory introduced by Claude Shannon [11]. Shannon's goal was to discover laws to regulate the capacity of systems for transmission, storage and processing of information. Furthermore, he intended to define quantitative measurements for each process. One of the most important contributions was to consider communication as a mathematical problem based on Statistics. Then, he proposed a way to measure information in a new probabilistic event based on the traditional expression of Boltzman entropy (1896) associating to this entropy an information measure.

In this model, information quantity transmitted in one message is a function of the predictability of the message. Therefore, independently of the message, the information quantity is related to the possibility that the message happens. If that probability is low, message content is high; otherwise, if it is predictable, then information content is low or has little information.

In order to measure information quantity, Shannon created the entropy concept, which is somewhat different of the classical concept in Physics (Thermodynamics), and he defined information quantity based on uncertainty, or the difficulty to predict that message.

Therefore, the notion of entropy is related to the degree of the disorganization that exists at the information source. With a higher disorder, higher is the potential information of that source. A source that answers with only one and same message to any question, does not transmit information, because there is no reduction of uncertainty.

In general, concepts of entropy enable us to compare properties of the system in numeric terms, examining how their probability distribution is.

In the information theory research, there have been formulated some proposals to generalize entropy. One of the most recent generalization is the Tsallis entropy [12]. In this entropy, Boltzmann classic entropy is generalized and this is convenient when it is applied for the characterization of different natural systems. Tsallis entropy extends the domain of the application of classical procedures. Its expression is [12]:

$$S_q = k \frac{1 - \sum_{i=1}^w p_i^q}{q-1} \quad (6)$$

where k is a positive constant that defines the unit of measurement of the entropy. The variable q belongs to the real numbers, which characterize a particular statistics. W is the total number of micro states, and p is the set of probabilities associated to states.

This entropy has advantages because it considers the instability of the system. In our application, it is natural to think of this instability as the growth of pages and doxels. In general, collections have an unstable growth evolution and this evolution can be characterized by the inherent disorder. We believe the Tsallis entropy can give us better results than the previous approach [3], because conceptually Tsallis entropy can handle the (dis)organization of instable structures, such as our collection of documents.

4 Discussion

Information retrieval is a non-trivial task, especially when we deal with structured documents. Here we have adopted the approach described in [3] and our intention is to extend and improve it.

An important topic in Bayesian networks is the concept of conditional independence. It is an important hypothesis that reduces computation costs as well as maintains a (conditional) independence between elements that have the same parent.

Preliminary tests of the computation of local scores have shown that this computation is like a previous selection of document parts, assigning weights (probabilities) to be used in final inference. It is important to develop new methods for computing local scores to help finding the most relevant set of doxels for a query. In other words, the local score works like a filter for calculating the global score.

Another important characteristic is that the base approach uses its own data to calculate the parameters needed by inference. Therefore, the information retrieved could get more feasible scores.

An important aspect of a collection, especially with respect to INEX collections, is the scale used for relevance assessments. In INEX 2007, the relevance scale was entirely along the specificity dimension [0,1] [4], differently from INEX 2003 [3] where the scale was two-dimensional. Therefore, modifications are being made to use the INEX 2007 scales.

Finally, we are under development of the methods here mentioned, and we will run extensive tests to compare our proposal with previous models.

5 Acknowledgments

The authors would like to thank the financial support granted by the FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo) Brazilian funding agency.

6 References

- [1] Lalmas M, Ruger S, Tsirikia T, Yavlinsky A. Progress in information retrieval. *Advances in Information Retrieval* 2006;3936:1-11.
- [2] W3C. Extensible Markup Language (XML). World Wide Web consortium 2007 Available from: URL: <http://www.w3.org/XML/>
- [3] Piwowarski B, Gallinari P. A Bayesian framework for XML information retrieval: Searching and learning with the INEX collection. *Information Retrieval* 2005 Dec;8(4):655-81.
- [4] Lalmas M, Tombros A. Evaluating XML Retrieval Effectiveness at INEX. 2007 Jun 1. Report No.: Vol 41 No 1.
- [5] Piwowarski B, Gallinari P. A machine learning model for information retrieval with structured documents. *Machine Learning and Data Mining in Pattern Recognition, Proceedings* 2003;2734:425-38.
- [6] Wilkinson R. Effective retrieval of structured documents. Dublin, Ireland 1994 p. 311-7.
- [7] Lalmas M. Uniform representation of the content and structure for structured document retrieval. London, England: Queen Mary & Westfield College, University of London; 2000.
- [8] Ribeiro-Neto B, Silva I, Muntz R. Bayesian Network Models for Information Retrieval. In: Crestani F, Pasi G, editors. *Soft computing in Information Retrieval*. Physica-Verlag; 2000. p. 259-91.
- [9] Amati G, Crestani F. Probabilistic Learning by Uncertainty Sampling with Non-Binary Relevance. In: Crestani F, Pasi G, editors. *Soft Computing in information retrieval: techniques and applications*. Physica-Verlag; 2000. p. 292-313.
- [10] Callan JP, Croft WB, Harding SM. The INQUERY Retrieval System. Spain 1992 p. 78-83.
- [11] Shannon CE. A Mathematical Theory of Communication. *Bell System Technical Journal* 1948;27(3):379-423.
- [12] Tsallis C. Possible Generalization of Boltzmann-Gibbs Statistics. *Journal of Statistical Physics* 1988 Jul;52(1-2):479-87.

How Task affects Information Search (Preliminary)

Elaine G. Toms, Tayze MacKenzie, Chris Jordan,
Heather O'Brien, Luanne Freund, Sandra Toze,
Emilie Dawe, Alexandra MacNutt
Centre for Management Informatics
Dalhousie University
Halifax, Nova Scotia, Canada

Abstract

The purpose of this research is to examine how search differs according to selected task variables. Three levels of task type and two levels of task structure were explored. This mixed within- and between-subjects designed study had 96 participants complete three of 12 search tasks in a laboratory setting using a specialized search system based on Lucene. Using a combination metrics (user perception collected by questionnaires, transaction log data, and characteristics of relevant documents), we assessed the effect of type and structure on search process and outcomes.

Introduction

People bring to a search system individual differences (i.e., reading ability, prior knowledge and experience, motivation) that influence their use of a system. But, typically they are operating within a rich context that can be described by multiple variables (Toms et al, 2004), and the challenge to date has been which of these many variables affect the search process and outcomes. Critical to search success is task - how can an outcome be measured if the results do not support the intended need? In this research, we examine the effect of task as defined by task type and task structure.

In general, tasks range from merely fact finding, (e.g., how do I get to Frankfurt?), to complex decision making (e.g., obtaining information about a competitor to devise a marketing plan, finding evidence to make a medical diagnosis). Task, it seems is another concept (not unlike relevance) that has multiple definitions, and is used particularly within research in multiple ways. Gill and Hicks (2006) suggest that a task is a set of assigned: a) goals to be achieved, b) instructions to be performed, or c) a mix of the two. Bystrom and Hansen (2005) characterize task by the fact that each has a beginning and an end, has requirements (may be conditional or unconditional), and has both a goal/result and reason/purpose. Gwizdka and Spence (2006) define task as "a sequence of actions performed by the searcher in the process of looking for information to satisfy current information need." Whatever the definition, we choose to accept, a task encompasses an information need(s) and stops when the desired information is found (or when the person stops). The commonality among these definitions, thus, is that tasks are seen as having a progression from beginning to end with a defined intent.

Tasks are often described using a variety of characteristics, from type to complexity. Campbell (1988) depicts tasks as simple and complex. Within the scope of complex tasks, the doer may need to make a decision or judgment, or solve a problem in situations with varying degrees of information and uncertainty. "Fuzzy" tasks are those for which the outcome and the path for executing the task are both ill-defined, but that may be from the perception of the user. Sometimes the characteristics relate specifically to the task, and sometimes to the abilities or knowledge of the user. Bell and Ruthven (2004), for example, state that the complexity of an information task may be affected by the searchers ability to articulate the information goal and interpret the relevancy of the results, and the difficulty of searching for the information. As such, investigations of task must take into consideration a range of factors, such as characteristics of the doers, the doers' perception of their tasks, the nature of the product or task goal, the constraints around the task (e.g. time), the accessibility of information that will enable the successful completion of the task, and the usability, and interactivity of the medium for locating that information (Jarvelin, & Ingwersen, 2004; Li, 2004). This is a complex set of variables that is difficult to test and isolate in experimental settings.

Gwizdka and Spence (2006) examined subjective and objective measures (e.g., web page length, web page complexity) of task complexity in a study that involved nine search tasks of varying degrees of difficulty. They found that searchers' evaluations of task complexity were related to the number of unique web pages visited, the time spent on each page, the straightforwardness of finding information to satisfy the task requirements, and the degree of deviation from that optimal path. Kim (2006) who looked at factual, interpretive, and exploratory tasks also noted that the level of task complexity was correlated with the search interaction. However, with the exception of Ghani and Deshpande (1994), there has been little research in the area of the users' experience while performing complex tasks.

Research Design

Specifically, we sought to investigate:

1. Are there performance differences by task type? By task structure?
2. Does users' perception of the task differ by task type? By task structure?
3. Are different types of pages more likely to be pertinent to different task types/structures?
4. Are there patterns of search behaviour that are related to different task types/structures?

Methods

Overview

The study took place in a university lab setting enabling the efficient collection of data from 5-10 participants in a single session (for a total of 96 participants from multiple sessions). To conduct the research, we created a search system using open source software, and with a specialized interface.

System - wikiSearch

WikiSearch is run on Lucene 2.2, an open source search engine using the vector space model. We indexed the Wikipedia XML documents using the Lucene standard analyzer, using its default stemming and stop word filtering. The resulting 'documents' are composed of two fields, one holding the title and the other handling the rest of the contents. A second index was built by paragraph but has not been used because of the inconsistencies in the usage of the paragraph tag for the first paragraph and the abstract paragraph within the XML document collection. Our plan was to use this index to rank paragraphs within an article based on the user query, and will be implemented in our next version.

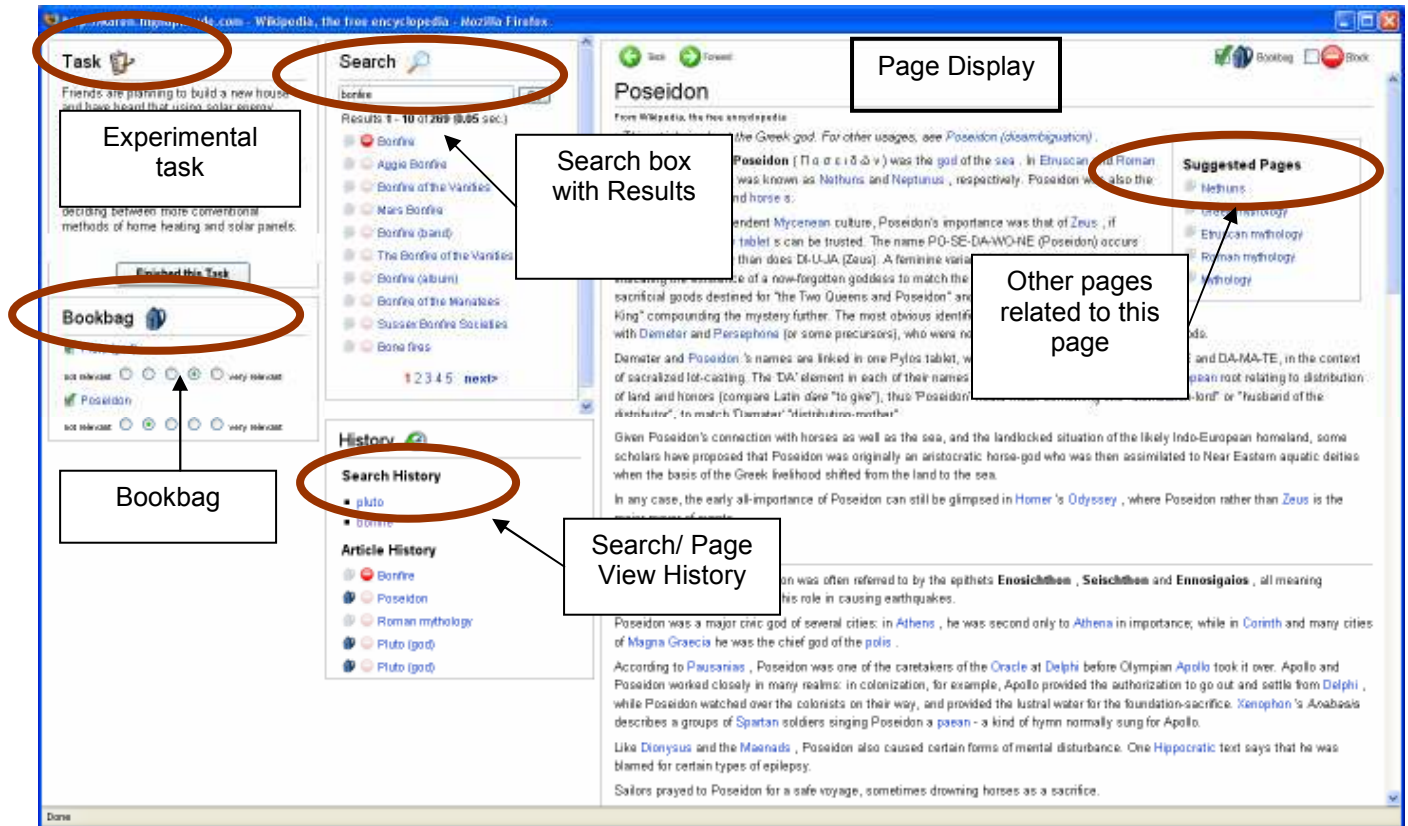
WikiSearch contains a customized interface with special features written using a combination of server-side PHP, and client-side Javascript. First, to eliminate the labyrinth effect of layering multiple pages that also results in constant backtracking, wikiSearch contains a single interface that is divided into three logical frames:

The **Page Display** contains a scrollable wiki page that can be selected from the results (or the history). Each page contains two types of links: ordinary hypertext links that connect among the wiki pages, and link that acts as a search to the wiki. Thus, when links are discovered that might serve as a search term, that capability is supplied. Within the Page Display, a further list of Suggested Pages is provided. This set was created by entering the entire first paragraph of that page as a search string. This set can serve two purposes: providing more specific pages about the topic, or by providing distractions.

The **Search** section contains the omnipresent searchbox. But, to conserve space, the results section contains only titles, while a mouseover provides a snippet that contains..... Below the search results is a History section that contains a reminder of both past searches and past articles that were viewed.

The **Task** section contains the contents of the experimental task to serve as a constant reminder. Below the task is a Bookbag that is used to collect pages that are useful to the task. This idea is similar to the shopping cart used in the online shopping environment. In addition, each page is rated by the participant before the task is considered finished. Pages can be removed from the bookbag from within it, but are added to the Bookbag from the Page Display, the Search Results, and the History sections.

In addition, from the Page Display, the Search Results, or the History sections, a participant can drop a page in the Garbage can so that it never need to be viewed again for this search task.



Variables

Initially we considered the original task pool developed by INEX. But those tasks were too simple for human searchers, e.g., a single keyword search would likely net the most relevant page for most of the standard INEX topics. Initially, we assessed the original INEX topics according to a multitude of attributes (e.g., domain, level of specificity or abstraction, named objects, etc.) before concluding that the set was not useful in the study of interactive IR. We modified and/or developed the 12 tasks used primarily by the interactive track so that no search could be answered in a single page, and the task required searchers to actively make a decision about what information was truly relevant to complete a task. In the process of doing so, we also discovered that tasks have semantic content that requires interpretation, but tasks also have syntactic content - structure - that physically represents the task. As a result of this analysis, we proposed to test two task variables, one based on the semantics, and one on the syntactics of the tasks.

Task Type:

This characteristic of task contained three levels:

- a) Fact finding, where the objective is to find "specific accurate or correct information or physical things that can be grouped into classes or categories for easy reference".
- b) Information gathering, where the objective is to collect miscellaneous information about a topic
- c) Decision making, where the objective is to select a course of action from among multiple alternatives.

Task Structure

The tasks are also split into two categories, depending on the "structure" of the search task:

- a) Parallel: where the search uses multiple concepts that exist on the same level in a conceptual hierarchy; this is a breadth search (and in a traditional Boolean likely was a series of OR relationships).

b) Hierarchical: where the search uses a single concept for which multiple attributes or characteristics are sought; this is a depth search, that is, a single topic explored more widely.

Metrics

Each variable was assessed using the following metrics:

a) User perception:

Pre: Prior knowledge/experience (used primarily as a post experimental control),

Post: Satisfaction, Mental Effort, Time, Complexity, Uncertainty, Knowledge,

Expectations/Predictability, etc.

b) Task: Time to task completion, Number of [modified] queries, Size of [modified] query, Number of pages viewed, Average rank in results, Number of items in garbage can, Number of second-order relevant pages (not coming from results list), Ratio of user assessment/external assessment by page and aggregated by task; Completeness (or likely completeness of task, given pages in the Bookbag), etc.

In addition, we related some of these metrics to characteristics of the document, e.g., size of the page, Number of paragraphs in the page.

Instruments

We administered pre-task, post-task, and post-session questionnaires which were mostly of a Likert scale style. The pre-task questions concern prior knowledge and familiarity with the topic of the search. The post-task questionnaire included the usual questions that address satisfaction with and confidence in the information they found to respond to the task. In addition, selected items pertained directly to the task based on prior research on task (e.g., Bystrom)

The Post-session Questionnaire evaluated the WikiSearch system features. Specifically, users rated the interface features (e.g., Bookbag, Suggested Pages) as well as wikiSearch's usability. Usability was assessed using the System Usability Scale (SUS), a ten item scale developed by Digital Equipment Corporation (DEC) which is widely accepted and used in the human computer interaction community (see for example, Everett, Byrne & Greene (2006)).

Participants

Participants were adults primarily from the university community, and from mixed disciplines. They were recruited via listservs and recruitment posters placed around the campus. This was a convenient sample who were paid \$10 each as an honorarium.

Procedure

Participants interacted with wikiSearch via an enhanced version of WiiRE (Web Interactive Information Retrieval Experimentation) (Toms, Freund & Li, 2004). WiiRE was written in PHP and lead the participant through the experimental process using a series of webpages. Responses to questionnaires and the contents of a customized logfile were stored in a MySQL database.

Data collection took place in a laboratory setting that ran 5 to 7 people at a time. Participants were presented with the following steps: 1) Introduction, 2) Consent Form, 3) Demographics and Use Questionnaire, 4) Tutorial and practice time using the wikiSearch system, 5) Pre-Task Questionnaire, 6) Assigned Task, 7) Post-Task Questionnaire, 8) Steps 5 to 7 were repeated for the other two tasks, 9) Post-Session Questionnaire, 10) SUS Questionnaire, 11) Thank-you for participating page.

After completing the demographic information, each participant performed three randomly assigned search tasks. For each task, participants were introduced to search task and the pre-task questions. Upon completing each task, participants completed a post-task questionnaire.

Data Analysis

The data is being analyzed using primarily SPSS analysis of variance to assess:

- 1) the effect of the task type: a) Decision-making, b) problem-solving, or c) fact-finding
- 2) the effect of task structure: a) Parallel or b)Linear
- 3) interaction effect of task and structure

These differences are being assessed by each of the metrics lists in the Metric section.

Conclusions

At the time of writing, the data was still in analysis, but will be ready for the workshop. The intent of this work is to examine how people search when given different types of search tasks, and when those tasks have a particular structure. It is also examining whether a relationship exists between different task types and the sorts of pages are more likely to be pertinent by type.

References

- Bell, D. J., & Ruthven, I. (2004). Searcher's Assessments of Task Complexity for Web Searching. In S. McDonald & J. Tait (Eds.), *Lecture Notes in Computer Science* (Vol. 2997, pp. 57-71). Berlin Heidelberg: Springer-Verlag.
- Bystrom, K., & Hansen, P. (2005). Conceptual framework for tasks in information studies. *Journal of the American Society for Information Science and Technology*, 56(10), 1050-1061.
- Campbell, D. J. (1988). Task complexity: a review and analysis. *Academy of Management Review*, 13(1), 40-52.
- Ghani, J. A., & Deshpande, S. P. (1994). Task characteristics and the experience of optimal flow in human-computer interaction. *The Journal of Psychology*, 128(4), 381-391.
- Gill, T. G., & Hicks, R. C. (2006). Task Complexity and Informing Science: A Synthesis. *Informing Science*, 9 [electronic version].
- Gwizdka, J. & Spence, I. (2006). What Can Searching Behavior Tell Us About the Difficulty of Information Tasks? A Study of Web Navigation. In *Proceedings of ASIS&T*, Austin, Texas.
- Jarvelin, K., & Ingwersen, P. (2004). Information seeking research needs extension towards tasks and technology. *Information Research*, 10(1), <http://informationr.net/ir/10-11/paper212.html>.
- Kim, J. (2006, April 22-27). Task Difficulty as a Predictor and Indicator of Web Searching Interaction. In *Proceedings of CHI*, Montreal, Quebec.
- Li, Y. (2004, November 12-17). Task Type and A Faceted Classification of Tasks. Poster presented at the *Annual Meeting of the American Society of Information Science and Technology*, Providence, RI.
- Toms, EG, Bartlett, J, Freund, L, Dufour, C, & Szigeti, S. (2004). Identifying the significant contextual factors of search. In: *SIGIR 2004 Workshop on Information Retrieval in Context Proceedings*. Sheffield (UK), 29 July 2004.
- Toms, E.G., Kopak, R. Freund, L. & Bartlett, J. (2003). The effect of task domain on search. In *Proceedings of CASCON 2003*, Markham, Ontario, Canada, 6-9 October 2003, 1-9.
- Toms, E.G., O'Brien, H., Kopak, R. & Freund, L. (2005). Searching for relevance in the relevance of search. In *Proceedings of COLIS5*, Glasgow, Scotland, June 2005.

A Comparison of Interactive and Ad-hoc Relevance Assessments

Birger Larsen¹, Saadia Malik², and Anastasios Tombros³

¹Royal School of Library and Information Science, Denmark

²University Duisburg-Essen, Germany

³Queen Mary University of London, UK

{blar@db.dk, malik@is.informatik.uni-duisburg.de, tassos@dcs.qmul.ac.uk}

Abstract. In this paper we report an initial comparison of relevance assessments made as part of the INEX 2006 Interactive Track (itrack'06) to those made for the topic assessment phase of the INEX 2007 ad-hoc track. We focus on investigating the effect of the different assessment conditions on the perceived relevance of document elements.

1 Introduction

In this paper, we report on a comparison of relevance assessments made as part of the INEX 2006 interactive track [4] (itrack'06) and those made as part of the topic assessment phase for the INEX 2007 ad-hoc track. Our analysis is based on eight topics that were assessed as part of both tracks.

The conditions under which the eight topics were assessed were significantly different, with searchers in itrack'06 assessing the usefulness of elements in addressing information seeking tasks, while topic assessors for the ad-hoc track focused on providing comprehensive assessments for each retrieved document. These different conditions provide the main motivation for carrying out this research. More specifically, we primarily interested in investigating:

- The extent to which the different conditions affect the relevance of document elements, as perceived by itrack'06 searchers and ad-hoc topic assessors.
- The overlap of the assessed information, i.e. to what extent the information that searchers and assessors perceived as being useful in their respective tasks was similar.

In addition, the eight topics used in the study are classified into different task types [4,8], providing thus the opportunity to also study the effect of different topic types in the two above issues. Further, in itrack'06 two versions of an XML IR system were used (more details in section 2.1 and in [4]), allowing us to also study the effect of system type perception of document element relevance.

In the remaining of this paper, we first describe some methodological issues in section 2, we then present some initial results and analysis in section 3, and we conclude and outline our further plans for analysis in section 4.

2 Methodology

In this section we describe the methodology of our study. First in sections 2.1 and 2.2 we briefly summarise the frameworks under which relevance assessments were made for itrack'06 and the INEX 2007 ad-hoc track, respectively, and in section 2.3 we discuss the methodology by which the assessments in the two tracks were compared.

2.1 Interactive track 2006

In the INEX 2006 interactive track (itrack'06) searchers from various participating institutions were asked to find information for addressing information seeking tasks by using two interactive retrieval systems: one based on a Passage retrieval backend¹ and one on an Element retrieval backend². Both versions had similar search interfaces but differed in the returned retrieval entities: The passage retrieval backend returns non-overlapping passages derived by splitting the documents linearly. The element retrieval system returns elements of varying granularity based on the hierarchical document structure. For a full description of the systems used in itrack'06 the reader can refer to [4].

Twelve search tasks of three different types [8] (*Decision making*, *Fact finding* and *Information gathering*), further split into two structural kinds (*Hierarchical* and *Parallel*), were used in the track [4]. The tasks were split into different categories allowing the searchers a choice between at least two tasks in each category, and at the same time ensuring that each searcher will perform at least one of each type and structure.

An important aspect of the study was to collect the searcher's assessments of the relevance of the information presented by the system. We chose to use a relevance scale based on work by Peheceviski et al. [5]. Searchers were asked to select an assessment score for *each viewed piece of information* that reflected the usefulness of the seen information in solving the task. Five different scores were available, expressing two aspects, or dimensions, in relation to solving the task: How much *relevant information* does the part of the document contain, and how much *context is needed* to understand the element? This was combined into five scores as follows:

- **Not relevant (NR)**. The element does not contain any information that is useful in solving the task
- **Relevant, but too broad (TB)**. The element contains relevant information, but also a substantial amount of other information
- **Relevant, but too narrow (TN)**. The element contains relevant information, but needs more context to be understood
- **Partial answer (PA)**. The element has enough context to be understandable, but contains only partially relevant information

¹ The Passage retrieval backend was based on CSIRO's Panoptic™/Funnelback™ platform. See <http://www.csiro.au/csiro/content/standard/ppsf6f...html> for more information.

² The Element retrieval backend was based on Max Planck Institute for Informatics' TopX platform. See [7] for more information.

- **Relevant answer (R).** The element contains highly relevant information, and is just right in size to be understandable.

In the interactive track, the intention is that each viewed element should be assessed with regard to its relevance to the topic by the searcher. This was, however, not enforced by the system as it may be regarded as intrusive by the searchers [3]. Note that in contrast to the assessments made for the ad-hoc track, there is no requirement for searchers to view each retrieved element as independent from other components viewed. Experiences from user studies clearly show that users learn from what they see during a search session. To impose a requirement for searchers to discard this knowledge would create an artificial situation and will restrain the searchers from interacting with the retrieved elements in a natural way.

Overall, 88 interactive track searchers made 2170 relevance assessments for the eight tasks analysed in this paper. Table 1 in Section 3 gives a detailed account of this data.

2.2 INEX 2007 ad-hoc assessments

The purpose of the INEX 2007 ad-hoc track is to create a test collection consisting of a corpus of documents, a set of questions directed at the documents (called topics) and a set of relevance assessments specifying which documents (or the elements that are part thereof) that are relevant to each topic. The elements to be assessed were identified by pooling the output of multiple retrieval systems following the method first proposed in [6]; the pool of retrieved elements for each topic was then assessed by the topic author.

In INEX 2007 the assessment process focussed on the notion of specificity, that is, the extent to which the element focuses on the information need expressed in the topic. A highlighting approach was taken, where the assessor first skims the document and then highlights any parts that contain only relevant information. From this, the specificity of any element with highlighted content can be calculated automatically. This may be done by computing the ratio of relevant content ($rsize$) to all content ($size$), measured in the number of characters.

All twelve topics that were used in *itrack'06* were also submitted as topics for the ad-hoc track. Up to the point of writing this paper, full assessments for eight of these topics were available – we use these as the basis of our result presentation and analysis in section 3.

2.3 Mapping ad-hoc and interactive track assessments

Whereas the interactive track assessments are given in terms of one of the five categories in section 2.1, the ad-hoc assessments are of a continuous nature. This necessitates a mapping between them. As mentioned above, there was a difference in the scope of the two types of assessments: where the ad-hoc track aimed at getting comprehensive assessments for each retrieved document, the interactive track searchers were free to assess as much or as little information as they saw fit. In

addition, no attempt was made to control learning effects across a search session in the interactive track, while ad-hoc assessors were explicitly asked to assess each element on its own merit.

In the interactive track, non-relevant elements could be specified explicitly (by selecting the NR assessment), as well as implicitly (by searchers viewing an element but not giving any assessment). As such, there is a good correspondence with the ad-hoc track, where only relevant information was highlighted and the rest ignored.

The notion of relevant information (R) in the interactive track would correspond in the ad-hoc assessments to elements that are either fully highlighted or have a large ratio of highlighted content, for example elements with more than 75% relevant content might be considered as being relevant. Following the same line of argument, the interactive track notion of Too Broad (TB) would correspond to elements that in the ad-hoc assessments have a relatively small amount of highlighted content, for example, elements with less than 25% relevant content might be considered as being Too Broad.

It is, however, more difficult to identify a direct parallel to the notion of Too Narrow (TN) in the ad-hoc assessment data. It might be argued though that it is unlikely that small elements would have been relevant to the itrack'06 topics. Pragmatically, such small elements can be filtered out by excluding elements smaller than a given absolute size, e.g., 125 characters³. A similar reasoning based on absolute size could be applied as a supplemental criterion to the notion of Relevant (R): elements that contain, e.g., 500 characters of highlighted content could be deemed Relevant, regardless of the ratio of highlighted content.

The notion of Partial Answer (PA) is also difficult to translate to the ad-hoc assessments, because only relevant information was highlighted in the assessment process.

3 Results and analysis

In the interactive track 88 searchers were recruited by 8 research groups, and overall they completed 334 search sessions⁴. Table 1 presents some basic statistics for the assessments provided as part of itrack'06. For the eight topics analysed in the present paper, 2170 elements were assessed. As different searchers would often assess the same elements for the same topic, the number of unique assessed elements was 1039 (an average of 2.1 assessments per element). For 177 of these uniquely assessed elements, two or more different assessments (e.g. R, TB and TB) were given by searchers. These present a particular challenge in our study, because we need to arrive at a single assessment for each element in order to compare it to the ad-hoc assessments.

³ Based on that a typical sentence length in English text is around 125 characters (<http://hearle.nahoo.net/Academic/Maths/Sentence.html>).

⁴ Due to system problems, logs of some search sessions are missing.

Table 1. Basic statistics on the relevance assessments provided by the INEX 2006 interactive track searchers (including elements that were viewed, but not assessed).

Total number of assessments (including elements assessed more than once)	2170
Unique elements assessed	1039
Unique elements with two or more different assessments	177

In Table 2, we provide details about how these different assessments are distributed among the 1039 uniquely assessed elements. Both rows and columns list the relevance categories and the table shows how many elements have been assessed under both categories by any number of different searchers. There are for instance 57 elements that have been assessed both as Relevant and as Too Broad.

The distribution of values in Table 2 is fairly uniform, with the maximum value being the 10% of the elements marked as NA and R. This largest value corresponds to searchers viewing, but not assessing (NA), elements that other searchers had assessed as relevant. Overall, elements that were not assessed by some searchers but were assessed by other searchers (i.e. the NA row) correspond to the largest percentage in Table 2. Elements assessed as non-relevant (NR) are noteworthy as they correspond to cases where searchers have explicitly indicated that the elements are particularly ill-fitted to the topic. Elements assessed as non-relevant overlap with relevant of any category in 3-5% of the cases. In the heuristics applied to derive a single assessment for the 177 elements, special weight is given to those that were explicitly assessed as non-relevant.

Table 2. Details of how different assessments are distributed among document elements in raw counts (left) and percentages over the 1039 unique assessed elements (right).

	R	NA	NR	PA	TB	TN		R	NA	NR	PA	TB	TN
R	-	103	52	68	57	36	R	-	9.9%	5.0%	6.5%	5.5%	3.5%
NA	103	-	77	75	59	34	NA	9.9%	-	7.4%	7.2%	5.7%	3.3%
NR	52	77	-	47	32	19	NR	5.0%	7.4%	-	4.5%	3.1%	1.8%
PA	68	75	47	-	35	20	PA	6.5%	7.2%	4.5%	-	3.4%	1.9%
TB	57	59	32	35	-	18	TB	5.5%	5.7%	3.1%	3.4%	-	1.7%
TN	36	34	19	20	18	-	TN	3.5%	3.3%	1.8%	1.9%	1.7%	-

We applied the following heuristics to arrive at a single category of relevance for each of the 177 elements that were assessed differently by different searchers:

1. For elements that were viewed, but not-assessed, the explicit assessments are given priority.
2. If there was a majority vote, the majority category was chosen regardless of the difference.
3. If there was a tie with an element assessed as non-relevant, NR was chosen.
4. In remaining ties, any elements assessed as Relevant were categorised as relevant.
5. Any outstanding ties (i.e., between PA, TB and TN in any combination) were left as ties (indicated as -tie- below).

Table 3 shows the resulting distribution of the interactive track assessments in total and over the eight topics. Less than 25% were Partially Relevant, Narrow or Broad

including only 10 ties. The rest are roughly divided into three equally sized groups of Relevant, non-relevant and non-assessed elements, each of around 25%. We plan to further analyse the per-task data in time for the workshop. In particular, we plan to investigate whether there are differences between topics that correspond to different task types.

Table 3. Distribution of interactive track assessments over topics after application of heuristics on elements with two or more different assessments.

Topic	T1	T3	T4	T5	T7	T8	T9	T12	Total
R	15	52	11	26	21	37	67	50	279
NA	21	31	27	23	16	55	60	35	268
NR	13	31	16	60	20	71	42	10	263
PR	4	16	9	14	11	25	15	11	105
TB	5	16	4	7	6	5	18	3	64
-tie-	1	5		1	1	1	1		10
TN	9	7	3	5	11	5	4	6	50
Total	68	158	70	136	86	199	207	115	1039

In order to compare the interactive assessments to those of the ad-hoc track, we applied the mapping heuristics discussed in Section 2.3 to the ad-hoc assessments. We regard any element with 75% or more highlighted content as relevant (R), and any with less than 25% as Too Broad. Table 4 shows the distribution of inferred Relevant and Too Broad assessments, as well as 801 elements assessed in the interactive track but not assessed in the ad-hoc track (the NA column). In addition, the 39 assessments that fall outside the range defined by the inferred R and TB categories are shown distributed over 5 intermediate bins according to the rsize to size ratio. Excluding 23 elements that were viewed but not assessed in the interactive track (NA, second row) leaves an overlap of only 215 elements between the two tracks.

The data from Table 4 suggest that there is little agreement in what kind of information interactive and ad-hoc assessors deem as useful for the same information-seeking tasks, since there is relatively small overlap in the common elements assessed. A further observation from the data is that, with regards to the commonly assessed elements, there is a certain degree of agreement on relevant and not relevant information, as demonstrated by the level of agreement in the R and NR⁵ rows. For instance, of the 129 elements assessed as relevant in the interactive track, 75 were relevant in the ad-hoc assessments and 12 more had between 50% - 75% relevant content as measured by the rsize to size ratio. In addition, looking at marginal cases such as TB and TN in the interactive assessments, we notice that relatively few of these are Relevant in the ad-hoc data.

⁵ Especially so given that non-assessed (NA) elements in the ad-hoc track are an explicit indication of non relevance.

Table 4. Distribution of inferred relevance categories (Relevant and Too Broad) of ad-hoc assessments as well as non-assessed ad-hoc elements over interactive track assessments.

Ad-hoc data: Inferred relevance categories & non-assessed elements

		$\frac{rsize}{size}$	0.3-0.4	0.4-0.5	0.5-0.6	0.6-0.7	0.7-0.8	R	TB	Total	NA	Grand total
Interactive track data	R		7	2	3	5	4	75	33	129	150	279
	NA							13	10	23	245	268
	NR		2	3			1	6	13	25	238	263
	PR		1	1	2	1	2	11	5	23	82	105
	TB				2		1	9	12	24	40	64
	-tie-							1	1	2	8	10
	TN		1		1			4	6	12	38	50
	Total		11	6	8	6	8	119	80	238	801	1039

The rather small overlap between the two sets of assessments indicates that each set contains significant numbers of elements not assessed in the other set. Because of the procedure of only highlighting relevant information in the ad-hoc track, we wanted to investigate how many of the 801 elements exclusively retrieved by interactive track searchers were originally in the ad-hoc pools. Table 5 shows that 510 of the interactive track elements were actually not included in the ad-hoc track pools. In slightly more than half of these cases, the interactive track searchers found these elements either non-relevant or not worth assessing. However, in 117 cases (23%) they did find the elements fully relevant and in another 114 cases (22%) relevant to some degree.

Table 5. Distribution of non-assessed elements from the ad-hoc track over interactive track assessments, including and excluding elements in the ad-hoc pools.

	NA	NA, not in ad-hoc pool
R	150	117
NA	245	150
NR	238	129
PR	82	49
TB	40	25
-tie-	8	7
TN	38	33
Total	801	510

4 Concluding remarks and future work

We reported an initial comparison of relevance assessments made as part of the INEX 2006 Interactive Track (itrack'06) to those made for the topic assessment phase of the INEX 2007 ad-hoc track. The data that we presented suggest that there are significant differences in what information was assessed under the two different conditions, but it also suggests a certain level of agreement in what constitutes relevant and non-relevant information.

The analysis that we present in this paper is only preliminary. At the workshop we plan to present more extensive results, and to report on the two issues that we left unaddressed in this paper, namely the effect of task type and itrack'06 retrieval system (passage vs. element) on the perceived relevance of document elements. For further work, we also plan to further investigate the effect that specific differences in the assessment conditions might have had in the relevance assessments.

References

1. Denoyer, L., Gallinari, P. (2006): The Wikipedia XML corpus. *SIGIR Forum*, 40(1):64-69.
2. Fuhr, N., Klas, C.P., Schaefer, A., Mutschke, P. (2002): Daffodil: An integrated desktop for supporting high-level search activities in federated digital libraries. In *Proceedings of the 6th European Conference on Research and Advanced Technology for Digital Libraries (ECDL)*, p. 597-612.
3. Larsen, B., Tombros, A. and Malik, S. (2005): Obtrusiveness and relevance assessment in interactive XML IR experiments. In: Trotman, A., Lalmas, M. and Fuhr, N. eds. *Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology, held at the University of Glasgow*. Dunedin (New Zealand): Department of Computer Science, University of Otago, p. 39-42.
4. Malik, S., Tombros, A., Larsen, B. (2006). The interactive track at INEX 2006. In: Fuhr, N., Lalmas, M., Trotman, A. eds. *Proceedings of the 5th International Workshop of the Initiative for the Evaluation of XML Retrieval*, p. 387-399.
5. Pehcevski, J., Thom, J. A. and Vercoustre, A.M. (2005): Users and assessors in the context of INEX: Are relevance dimensions relevant? In: Trotman, A., Lalmas, M. and Fuhr, N. eds. *Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology, held at the University of Glasgow*. Dunedin (New Zealand): Department of Computer Science, University of Otago, p. 47-62.
6. Spärck Jones, K., van Rijsbergen, C. J. (1975): Report on the need for and provision of an 'ideal' information retrieval test collection. *British Library Research and Development Report 5266*, University Computer Laboratory, Cambridge.
7. Theobald, M., Schenkel, R., Weikum, G. (2005): An efficient and versatile query engine for TopX search. In *Proceedings of the 31st International Conference on Very Large Data Bases (VLDB)*, p. 625-636.
8. Toms, E.G., Freund, L., Kopak, R., Bartlett, J.C. (2003): The effect of task domain on search. In *Proceedings of the 2003 Conference of the Centre for Advanced Studies on Collaborative Research*, p. 303-312.

Overview of INEX 2007 Link the Wiki Track

Wei Che (Darren) Huang

Faculty of IT
Queensland University of
Technology
Brisbane, Australia
w2.huang@student.qut.edu.au

Yue Xu

Faculty of IT
Queensland University of
Technology
Brisbane, Australia
yue.xu@qut.edu.au

Shlomo Geva

Faculty of IT
Queensland University of
Technology
Brisbane, Australia
s.geva@qut.edu.au

Abstract

Going beyond traditional document level retrieval, the Link-the-Wiki track (LTW) focuses on producing a standard procedure and metrics for the evaluation of link discovery at different element levels. This means that each anchor text can be linked to either a specific XML element within (i.e. passage) or to a Best Entry Point (i.e. BEP). Therefore, the tasks offered by the LTW track presents considerable research challenges in wiki link discovery and its evaluation. Since the manual assessment and evaluation generally utilized in previous work is exhaustive, automated evaluation without involving manual assessment was used in the LTW 2007. Although evaluation results may be inaccurate because of the biased and incomplete result sets, the advantage of using automated evaluation may well compensate for this. Automated evaluation facilitates both speedy turnaround of the LTW task for minimal assessment effort, and it supports the use of a very large number of topics. The paper provides a description of the LTW task, the evaluation procedure which adopts the existing Wikipedia link collection for Qrels, and an evaluation tool. This paper also provides an overview of the Link-the-Wiki task preliminary results.

1. Introduction

Geva and Trotman (2006) introduced the Link-the-Wiki task that aims at providing an evaluation forum for participants to propose and discuss algorithms for performing automated link discovery in XML documents and evaluating the performance of such algorithms. The test collection includes documents, judgments, and metrics for evaluating different systems and comparing various approaches to automated discovery of hypertext links.

The Wikipedia is a free online document repository written collaboratively by wiki contributors around the world. Composed of millions of articles in numerous languages it offers many attractive features as a corpus for information retrieval tasks. The INEX Wikipedia collection has been converted from its original wiki-markup text into XML [3]. That collection is composed of a set of XML files where each file corresponds to an online article in Wikipedia. A semantic annotation of the Wikipedia was also undertaken by others (e.g. [4]). Search as well as retrieval could benefit from rich semantic information in the XML Wikipedia collection, where it exists.

The semi-structured format provided by the XML-based collection offers a useful property for the evaluation of various semi-structured retrieval techniques. Specifically, the linkage within a document is an especially interesting aspect of the Wikipedia and offers opportunities for investigating article categorization as well as the user interaction (e.g. browsing and searching) with a hyperlinked corpus. In consequence, the Wikipedia collection has been used for a variety of purposes such as

XML information retrieval, machine learning, clustering, structure mapping, and categorization.

The user scenario for the Link-the-Wiki task is that of an end user who creates a new article in the Wikipedia. The wiki system then automatically nominates a number of prospective anchor texts, and multiple link destinations at the element level for each. The wiki system also offers prospective updates to related links in other (e.g. older) wiki articles, which may point to passages or elements within this newly created article. Therefore, links on each article can always be up-to-date with the latest information existing within the wiki system (or even linking outside the individual wiki system). An existing link suggestion tool, developed by Jenkins (2007), suggests a number of anchors that have not been linked within a given article and can potentially be linked to other pages in the Wikipedia [2]. From a list of suggested links on this tool, the user can accept or reject proposed links.

At INEX 2007, the LTW task is still focused on document-to-document links. 90 topics were “orphaned” from existing links and distributed to participants for link discovery. The result set (i.e. incoming and outgoing links for each topic) was generated automatically by parsing the entire collection for existing incoming and outgoing links for the topics. The detailed procedure of assessment and evaluation is described in section 5, including the result set generation, the concept of automatic evaluation and the evaluation tool. In general, the procedure can be divided into the following steps. Firstly, a number of orphan documents nominated by participants are used as example link-less documents. Then Participants are required to generate incoming and outgoing links for these selected topics and submit results (i.e. runs) to the track. Finally, performance is measured using standard IR metrics.

24 groups registered for the Link-the-Wiki track. However at the finishing line only 13 runs were submitted by 4 groups. They are University of Amsterdam with 5 runs, the University of Waterloo with 1 run, the University of Otago with 5 runs and the Queensland University of Technology with 2 runs.

An overview of Wikipedia research was presented by Voss, which consists of different aspects of wiki studies [6]. This includes the visualization of wiki editing, relations of readers and authors, citation of wiki articles, the (hyperlinked) structure of Wikipedia and the statistic of Wikipedia. Recently, more research with regard to Wikipedia has been undertaken in particular for identifying the relevance of wiki articles. Bellomi and Bonato utilize network analysis algorithms such as HITS and PageRank to find out the potential relevance of wiki pages (content relevant entries) in order to explore the high level (hyperlinked) structure of Wikipedia and gain some insights about its content regarding to cultural biases [7].

Ollivier and Senellart have conducted a set of experiments for examining the performance of approaches on finding related pages within Wikipedia collection [8]. There are totally 5 methods included in the evaluation, including Green-based methods, *Green* and *SymGreen*, and three classical approaches, *PageRankOfLinks*, *Cosine* with tf-idf weight and *Co-citations*. The concept of these methods is to find out the most related neighborhood of a given node. They can be derived to achieve the task of finding the related pages. Another interesting topic in finding related pages is to explore potential links in a wiki page by utilizing an automatic approach. Adafre and de Rijke propose a method of discovering missing links in Wikipedia pages via clustering of topically related pages by LTRank and identification of link candidates by matching the anchor texts [9]. Kumar et al. also apply the concept of co-citation in

the web graph for the similarity measure [10]. Beside co-citation, bibliographic coupling and SimRank can be used to determine the similarity of objects (e.g. web pages), which are based on the citation patterns of documents and the similarity of structural context respectively [11][12]. Moreover, the Companion algorithm derived from HITS (**H**yperlink-**I**nduced **T**opic **S**election) is proposed for finding related pages by exploiting links and their order on a page [13][14]. This conducts a strategy of using a page's URL, instead of query terms, to search a set of related Web pages.

According to the given set of queries, retrieval systems search the set of documents and returned a ranked list ordered to represent the relevance to each query. This pooling technique taking a set of to-be-judged documents provides the certain quality of the first N search results returned by each system for evaluation. In order to prevent from the judgment of the entire document set, depth- N pooling has been shown that it could be an effective way to evaluate the relative performance of retrieval systems in the case of TREC settings [15][16]. The main idea is that only the top n documents will be retrieved for assessment and the rest of the documents in the corpus are assumed as non-relevant to eliminate the unnecessary human effort. However, it may cause the biased and incomplete evaluation. The incomplete relevance data in Information Retrieval evaluation has been paid the attention of IR researchers. Large-scale test collections, such as the TREC, CLEF and NTCIR collections, created via the pooling technique can be refer as incomplete by some degree since only a subset of the document collection has been judged for relevance for each given topic [17][18][19].

2. Topics

Participants were given a set of orphan Wikipedia documents. Each participant contributed several topics and there were 90 topics in total in the LTW task in 2007. These 90 files were orphaned by removing all `<collectionlink>`, `<wikipedialink>`, and `<unknownlink>` mark-up. A `links.xml` file contains all the links removed from the original topic files, which can be used for automated evaluation (see below). Duplicated and decrepit links were discarded. The proposed topics with related file names are showed in appendix B-1.

```
<Links>
<File Id="XML\part-0\305.xml">
  <collectionlink xmlns:xlink=http://www.w3.org/1999/xlink
xlink:type="simple" xlink:href="31140.xml">
    The Divine Comedy
  </collectionlink>
  ...
  <unknownlink src="29 August">29 August</unknownlink>
  ...
</File>
</Links>
```

Figure 1 Example links.xml content

3. Submission

The official task specification can be summarised as follows:

- Up to 5 submissions per participant are allowed.
- Each run can contain up to 90 topics. Missing topics are regarded as having a score of zero for the purpose of calculating system rank when using all topics. Of course, the system can only evaluate submitted topics.
- Up to 250 incoming links and up to 250 outgoing links can be specified per topic. Surplus links are discarded when computing the performance.
- Runs that violate these requirements are disqualified.

Once the runs were uploaded onto the INEX submission area, these runs were validated against the submission DTD. An evaluation tool was provided for offline validation of the runs using the embedded XML schema (See Appendix A-1). An example submission was provided for reference purposes (See Appendix A-2). The DTD is shown below.

```
<!ELEMENT inex-submission (details+, description, collections, topic+)>
<!ATTLIST inex-submission participant-id CDATA #REQUIRED
                        run-id CDATA #REQUIRED
                        task (LinkTheWiki) #REQUIRED>
<!ELEMENT details (machine, time)>
<!ELEMENT machine (cpu, speed, cores, hyperthreads, memory)>
<!ELEMENT cpu (#PCDATA)>
<!ELEMENT speed (#PCDATA)>
<!ELEMENT cores (#PCDATA)>
<!ELEMENT hyperthreads (#PCDATA)>
<!ELEMENT memory (#PCDATA)>
<!ELEMENT time (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT collections (collection)>
<!ELEMENT collection (#PCDATA)>
<!ELEMENT topic (outgoing, incoming)>
<!ATTLIST topic file CDATA #REQUIRED
                name CDATA #REQUIRED>
<!ELEMENT outgoing (link*)>
<!ELEMENT incoming (link*)>
<!ELEMENT link (anchor,linkto)>
<!ELEMENT anchor (file, start, end)>
<!ELEMENT linkto (file, bep)>
<!ELEMENT file (#PCDATA)>
<!ELEMENT start (#PCDATA)>
<!ELEMENT end (#PCDATA)>
<!ELEMENT bep (#PCDATA)>
```

Figure 2 Submission DTD

4. Retrieval Tasks

The XML Wikipedia collection is composed of 660,000 documents in English and is around 5GB in size. Many articles in the Wikipedia collection are already extensively hyperlinked. The task is two fold:

1. Recommend anchor text in response to the given topic, and the corresponding destination documents within the Wikipedia collection.
2. Recommend incoming links from other Wikipedia documents.

For 2007 we operated a retrieval task at the document level, which means that only document-to-document links were evaluated. Up to 250 outgoing links and 250 incoming links were allowed for each topic. Since the orphaned documents were nominated in the existing Wikipedia collection, there are still links existing in other documents to the orphaned ones (i.e. incoming links were not removed). Moreover, the nominated topics were left in the collection and could conceivably be linked to each other. To simulate the genuine case in which these documents are truly orphans, some constraints, which prohibit the use of such residual linking information, have been specified in the Link-the-Wiki result submission specification [5].

5. Evaluation

5.1 Result Set Generation

One of the aims of the Link-the-Wiki track in 2007 was to explore the automated evaluation procedure (i.e. without manual assessment). In order to achieve this goal, the existing wikipedia links collection and an evaluation tool were developed. The existing links on each topic were extracted by removing `<collectionlink>`, `<unknownlink>` and `<wikipedialink>` from the topics content (see figure 3). Only collection links were recorded for subsequent evaluation. “What links here” provided by Wikipedia can be utilized to identify the incoming links (see figure 4). In practice, these incoming and outgoing links can be derived directly from the collection.

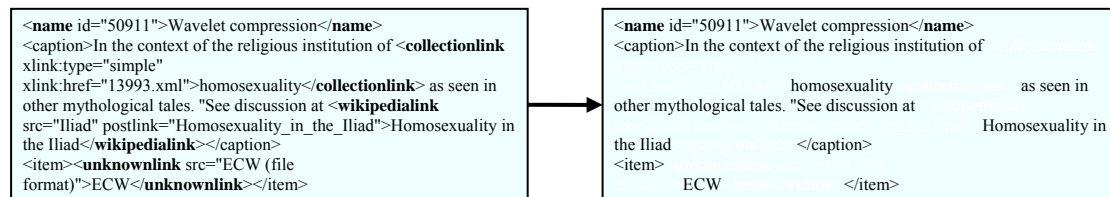


Figure 3 The elimination of internal Wikipedia links

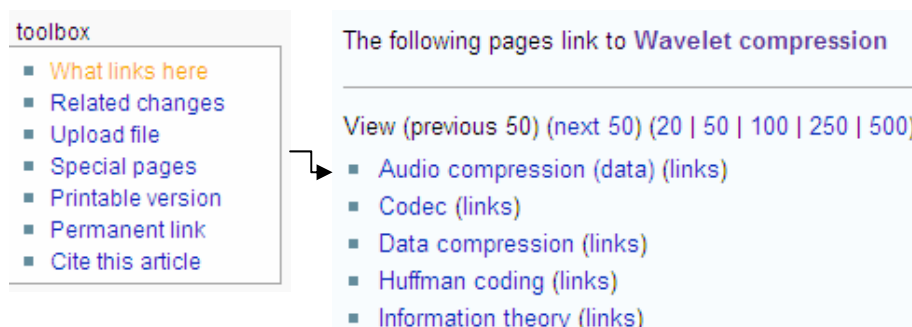


Figure 4 Wikipedia *What links here* function

Some existing Wikipedia links may not be useful as they were generated automatically and not by page authors (e.g. year links). Furthermore, some returned links may be suitable, but do not exist in the Wikipedia. Besides, some existing (older) wiki pages may not be linked to newly created pages. In consequence, evaluation results may be inaccurate. On one hand, results may appear optimistic because some links are easier to discover (e.g. the automated ones, like year). On the other hand, results may appear pessimistic because some useful returned links are not recorded in

the existing Wikipedia pages. It is not yet clear which way the results will go, but at any rate one has to bear this in mind when comparing systems in 2007.

5.2 Evaluation Procedure

An evaluation tool, named *ltwEval*, with the official result set was developed for LTW 2007 (see figure 5). The performance measures include Mean Average Precision (MAP), precision at the point of the number of relevant documents (R-Prec), and precision at varying numbers of documents retrieved (e.g. P@5, P@10, P@20, P@30 and P@50). Plots for incoming, outgoing and a combined score are also computed for comparison. By combined score we refer to the harmonic mean of the various values obtained for incoming and outgoing links. The *ltwEval* program was developed in Java for platform independence, but is GUI driven and provides more extensive functionality than traditional evaluation software. This should assist participants by making result exploration and analysis easier.

Performance measures can be calculated by using all 90 topics or only submitted/selected topics. It is convenient for users to only measure specified topics. The measures include Mean Average Precision (MAP), R-Precision, P@5, P@10, P@20, P@30 and P@50. From the results table, the user can choose the colour and line width (i.e. thick or thin) for each type of a run (i.e. incoming, outgoing and combined). The interpolated Precision-Recall plots can identify the performance of each run by supporting selection of colour and line thickness for individual runs.

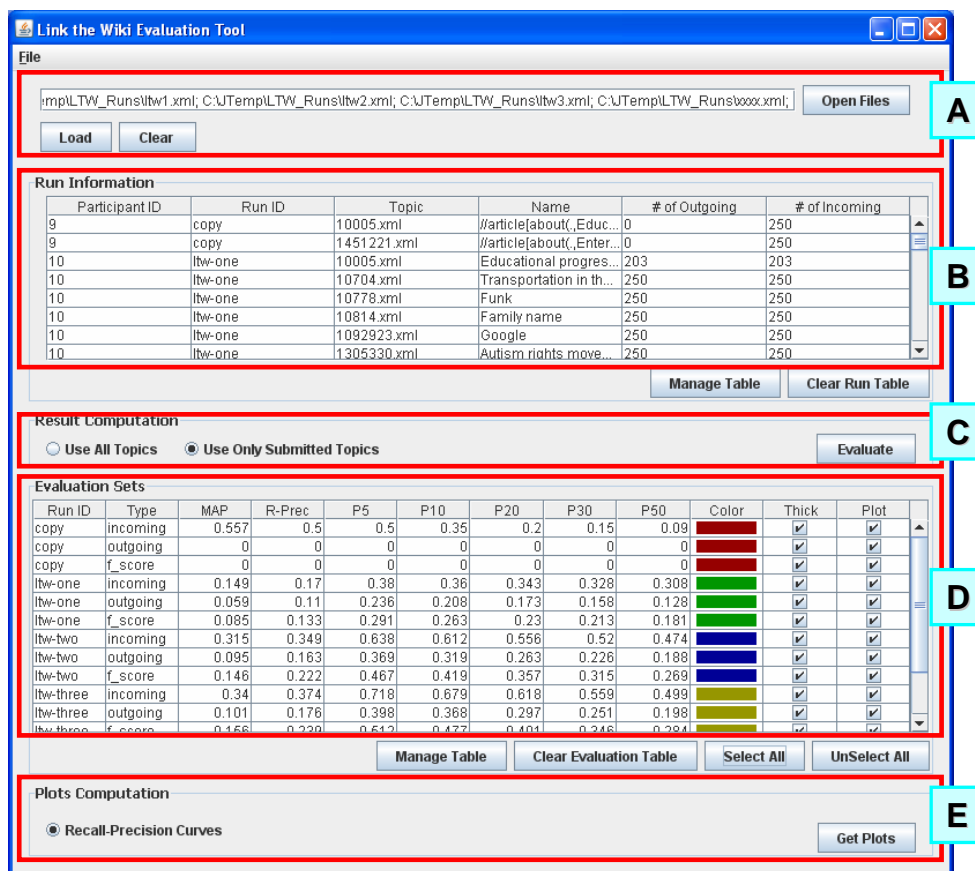


Figure 5 The Evaluation Tool

6. Approaches to Link the Wiki

In this section we briefly describe the approaches that were taken by the participants and the preliminary results of evaluation. The University of Amsterdam had submitted 5 runs. They assumed that Wikipedia pages link to each other when articles are similar or related in content. For each of the 90 topics (orphans), the system queries the index of the entire collection, but excluding the topics. This was tested by using the full topic as query (excluding stop words), and important terms derived from a language model. The top 100 files (anchors) were selected for each topic. They experimented with line matching from the orphans to the anchor files. For the outgoing links, the system matched each line of a topic with the lines of the anchors until a matching line has been found. For the incoming links, the system iterated over all lines of each anchor for each line of the topic. However, the number of matches was restricted to 250 for both types of links, which hurt performance for incoming links because of duplicated article-to-article links. The generated runs were based on the names of the pages, exact lines, and longest common substrings (LCSS) expanded with WordNet synonyms. The results show that the run based on restricting the line matching to the names of pages performed best.

The University of Otago had submitted 5 runs. The system identified terms within the document that were over represented and from the top few generated queries of different lengths. A BM25 ranking search engine was used to identify potentially relevant documents. Links from the source document to the potentially relevant documents (and back) were constructed (at a granularity of whole document). The best performing run used the 4 most over represented search terms to retrieve 200 documents, and the next 4 to retrieve 50 more.

The University of Waterloo contributed 1 run in the LTW track. For incoming links, the system found the first 250 documents in the order of file numbers that contain the topic titles and then made article-to-article links from them. For outgoing links, the system computed the probabilities that each term is an anchor to a destination file, and then found those terms with more than 60% probability in topic files, and linked them to the corresponding destination files.

The Queensland University of Technology contributed two runs. Incoming links were identified by using the GPX search engine to search for elements that were about the topic name element. Results were ordered by *article* score with the more likely relevant links returned earlier in the list. Outgoing links were identified by running a window over the topic text (having discarded all XML markup) and looking for matching page names in the collection. The window size varied from 8 words down to 1 word, and included stop words. Longer page names were ranked higher than shorter page names, motivated by the trivial observation that the system was less likely to hit on a longer page name by accident. A naïve approach perhaps, but quite effective as it turns out.

7. Preliminary Results

The overall measures for all submission runs are shown in Appendix B-2, which include the incoming and outgoing scores.

Table 3 MAP of Outgoing and Incoming Links

MAP Outgoing Links			MAP Incoming Links		
1	QUT02	0.484	1	QUT02	0.318
2	QUT01	0.483	2	QUT01	0.314
3	Waterloo_LTW_01	0.465	3	Amsterdam_LTW_01	0.147
4	Otago_ltw-four	0.339	4	Otago_ltw-four	0.102
5	Otago_ltw-five	0.319	5	Otago_ltw-five	0.101
6	Otago_ltw-three	0.318	6	Waterloo_LTW_01	0.093
7	Otago_ltw-two	0.284	7	Otago_ltw-three	0.092
8	Amsterdam_LTW_01	0.226	8	Otago_ltw-two	0.081
9	Otago_ltw-one	0.123	9	Amsterdam_LTW04	0.080
10	Amsterdam_LTW03	0.110	10	Amsterdam_LTW02	0.080
11	Amsterdam_LTW02	0.108	11	Amsterdam_LTW03	0.073
12	Amsterdam_LTW04	0.093	12	Amsterdam_LTW07	0.067
13	Amsterdam_LTW07	0.004	13	Otago_ltw-one	0.048

Table 4 R-Precision of Outgoing and Incoming Links

Outgoing Links R-Prec			Incoming Links R-Prec		
1	QUT01	0.415	1	Waterloo_LTW_01	0.512
2	QUT02	0.411	2	QUT02	0.505
3	Otago_ltw-four	0.183	3	QUT01	0.503
4	Otago_ltw-five	0.183	4	Otago_ltw-four	0.379
5	Amsterdam_LTW_01	0.182	5	Otago_ltw-three	0.363
6	Otago_ltw-three	0.173	6	Otago_ltw-five	0.356
7	Otago_ltw-two	0.156	7	Otago_ltw-two	0.331
8	Amsterdam_LTW02	0.154	8	Amsterdam_LTW_01	0.258
9	Amsterdam_LTW04	0.149	9	Amsterdam_LTW02	0.165
10	Amsterdam_LTW03	0.141	10	Otago_ltw-one	0.153
11	Amsterdam_LTW07	0.127	11	Amsterdam_LTW03	0.144
12	Waterloo_LTW_01	0.103	12	Amsterdam_LTW04	0.142
13	Otago_ltw-one	0.098	13	Amsterdam_LTW07	0.020

7. Conclusion and Outlook

This is the first year of the Link-the-Wiki track. A new concept of assessment procedure has been brought to the participants with the aim of reducing the manual assessment effort. An evaluation tool has also been developed for participants to explore their runs. Submission results are briefly analysed and the findings are concisely described. Although we still operate the evaluation on the document level retrieval, it has opened a door for participants to share their suggestions and opinion for the track and discuss issues for the next year of the Link-the-Wiki task. Since the task will be on the element level retrieval, new assessment procedure and related tools will be researched and proposed. In 2008, participants will be required to study how to nominate anchor texts for each topic based on the context and also search link destinations (e.g. BEP and Passage) for each anchor. Incoming links to BEPs within the nominated article are also required. Rather than having one link per anchor several links will be allowed for the same anchor.

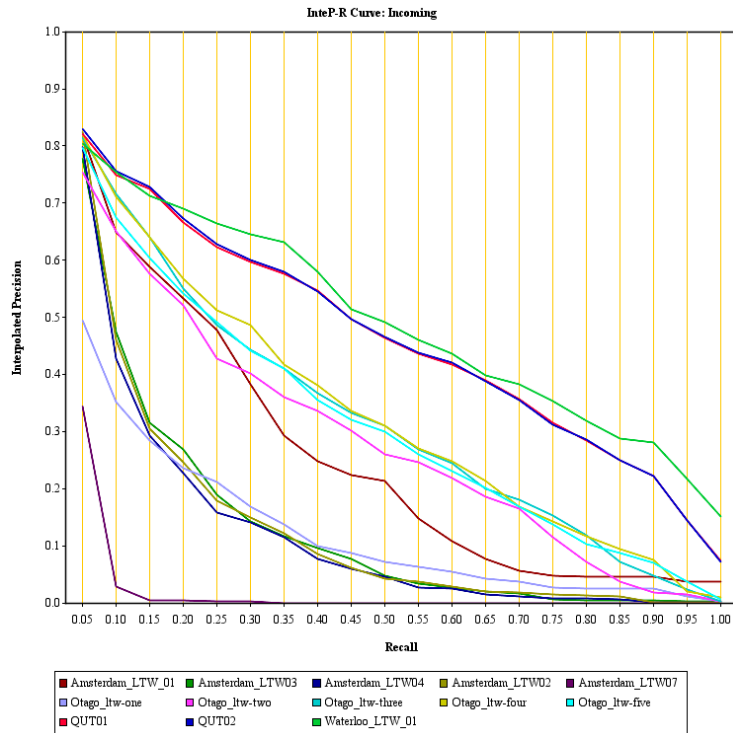


Figure 6. Interpolated Precision-Recall plots for Incoming links

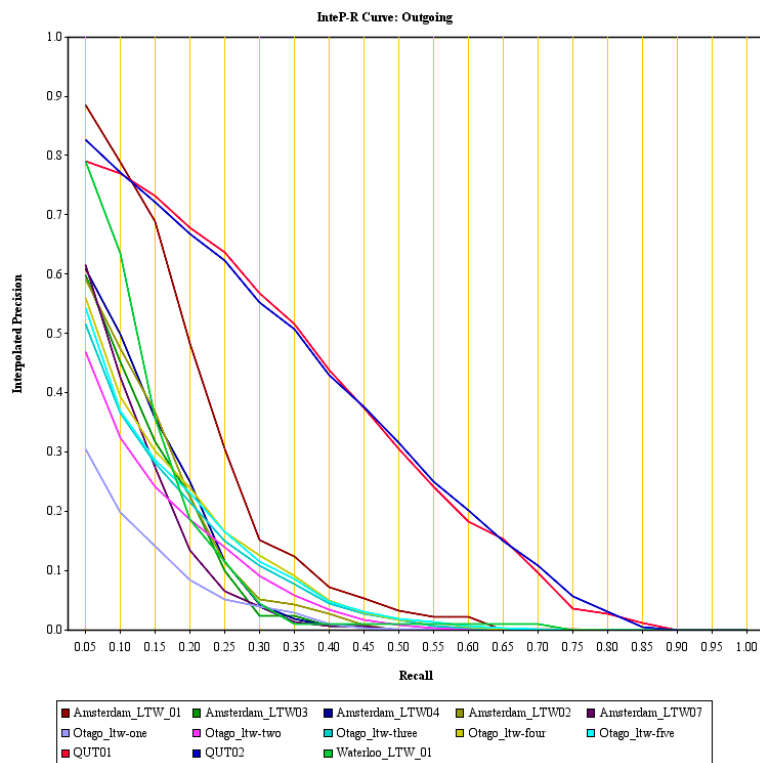


Figure 7. Interpolated Precision Recall plots for Outgoing links

8. References

- [1] Trotman, A. and Geva, S. Passage Retrieval and other XML-Retrieval Tasks, *In Proceedings of the SIGIR 2006 Workshop on XML Element Retrieval Methodology*, Seattle, Washington, USA, 10 August 2006, 48-50.

- [2] Jenkins, N. *Can We Link It*, 2007, http://en.wikipedia.org/wiki/User:Nickj/Can_We_Link_It
- [3] Denoyer, L. and Gallinari, P. The Wikipedia XML Corpus, *SIGIR Forum*, vol. 40, no. 1, June 2006, 64-69.
- [4] Schenkel, R., Suchanek, F. M. and Kasneci, G. YAWN: A Semantically Annotated Wikipedia XML Corpus, *In 12. GI-Fachtagung für Datenbanksysteme in Business, Technologie und Web (BTW 2007)*, Aachen, Germany, 2007, 277-291.
- [5] Geva, S. and Trotman, A., 2007, INEX 2007 Link the Wiki Task and Result Submission Specification, <http://inex.is.informatik.uni-duisburg.de/2007/inex07/protected/downloads/INEX%202007%20Link%20the%20Wiki%20Task%20Specification%20V1p0.pdf>
- [6] Voss, J. Measuring Wikipedia, *In Proceedings of the 10th International Conference of the International Society for Scientometrics and Informetrics (ISSI 2005)*, Stockholm, Sweden, 24-28 July 2005.
- [7] Bellomi, F. and Bonato, R. Network Analysis for Wikipedia, *In Proceedings of the 1st International Wikipedia Conference (Wikimania'05)*, Frankfurt am Main, Germany, 4-8 August 2005.
- [8] Ollivier Y. and Senellart P. Finding Related Pages Using Green Measures: An Illustration with Wikipedia, *In Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI'07)*, Vancouver, Canada, 22-26 July 2007.
- [9] Adafre, S. F. and de Rijke, M. Discovering missing links in Wikipedia, *In Proceedings of the SIGIR 2005 Workshop on Link Discovery: Issues, Approaches and Applications*, Chicago, IL, USA, 21-24 August 2005.
- [10] Kumar, R., Raghavan, P., Rajagopalan, S. and Tomkins, A. Trawling the Web for emerging cyber-communities. *Computer Networks*, 31(11-16), 1999, 1481-1493.
- [11] Jeh, G. and Widom, J. SimRank: a measure of structural-context similarity, *In Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'02)*, Edmonton, Canada, 23-26 July 2002, 538-543.
- [12] Kessler, M. M. Bibliographic coupling between scientific papers. *American Documentation*, 14(10-25), 1963.
- [13] Dean, J. and Henzinger, M. R. Finding related pages in the World Wide Web. *Computer Networks*, 1999, 31(11-16):1467-1479.
- [14] Kleinberg, J. Authoritative sources in a hyperlinked environment, *In Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, San Francisco, CA, USA, 25-27 January 1998, 668-677.
- [15] Harman, D. Overview of the third text REtrieval conference (TREC-3), *In Overview of the 6th Text REtrieval Conference*, 2-4 November 1994, Gaithersburg, Maryland USA, 1-19.
- [16] Zobel, J. How reliable are the results of large-scale retrieval experiments?, *In Proceedings of the 21th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, August 1998, Melbourne, Australia, 307-314.
- [17] Carterette, B., Allan, J. and Sitaraman, R. Minimal Test Collections for Retrieval Evaluation, *In Proceedings of the 15th Annual International ACM SIGIR Conference*, 6-11 August 2006, Seattle, Washington USA, 268-275.
- [18] Cormack, G. V., Palmer, C. R. and Clarke, C. L. A. Efficient Construction of Large Test Collections, *In Proceedings of the 21st ACM SIGIR Conference*, 24-28 August 1998, Melbourne Australia, 282-289.
- [19] Yilmaz, E. and Aslam, J. A. Estimating Average Precision with Incomplete and Imperfect Judgments, *In Proceedings of the 15th ACM International Conference on Information and Knowledge Management (CIKM'06)*, 5-11 November 2006, Arlington, Virginia USA, 102-111.

Appendix A

A-1 Submission XML Schema

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="inex-submission" type="inexSubmission"/>
  <xsd:complexType name="inexSubmission">
    <xsd:sequence>
      <xsd:element name="description"/>
      <xsd:element ref="collections"/>
      <xsd:element ref="topic" minOccurs="1" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="participant-id" type="xsd:string" use="required"/>
    <xsd:attribute name="run-id" type="xsd:string" use="required"/>
    <xsd:attribute name="task" type="xsd:string" use="required"/>
  </xsd:complexType>
  <xsd:element name="collections" type="collectionType"/>
  <xsd:complexType name="collectionType">
    <xsd:sequence>
      <xsd:element name="collection" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="topic" type="topicType"/>
  <xsd:complexType name="topicType">
    <xsd:sequence>
      <xsd:element ref="outgoing"/>
      <xsd:element ref="incoming"/>
    </xsd:sequence>
    <xsd:attribute name="file" type="xsd:string" use="required"/>
    <xsd:attribute name="name" type="xsd:string" use="required"/>
  </xsd:complexType>
  <xsd:element name="outgoing" type="linkingType"/>
  <xsd:element name="incoming" type="linkingType"/>
  <xsd:complexType name="linkingType">
    <xsd:sequence>
      <xsd:element ref="link" minOccurs="1" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="link" type="linkType"/>
  <xsd:complexType name="linkType">
    <xsd:sequence>
      <xsd:element ref="anchor"/>
      <xsd:element ref="linkto"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="anchor" type="anchorType"/>
  <xsd:complexType name="anchorType">
    <xsd:sequence>
      <xsd:element name="file"/>
      <xsd:element name="start"/>
      <xsd:element name="end"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="linkto" type="linktoType"/>
  <xsd:complexType name="linktoType">
    <xsd:sequence>
      <xsd:element name="file"/>
      <xsd:element name="bep"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

A-2 Example Submission

```
<inex-submission participant-id="12" run-id="LTW_01" task="LinkTheWiki">
<details>
  <machine>
    <cpu>Intel(R) Pentium (R) D</cpu>
    <speed>3.00GHz</speed>
    <cores>2</cores>
    <hyperthreads>None</hyperthreads>
    <memory>2GB</memory>
  </machine>
  <time>166295 seconds</time>
</details>
<description>Using text chunking etc.</description>
<collections>
  <collection>wikipedia</collection>
</collections>
<topic file="13876.xml" name="Albert Einstein">
  <outgoing>
    <link>
      <anchor>
        <file>13876.xml</file>
        <start>/article[1]/body[1]/p[3]/text()[2].10</start>
        <end>/article[1]/body[1]/p[3]/text()[2].35</end>
      </anchor>
      <linkto>
        <file>123456.xml</file>
        <bep>/article[1]/sec[3]/p[8]<bep>
      </linkto>
    </link>
    ...
  </outgoing>
  <incoming>
    <link>
      <anchor>
        <file>654321.xml</file>
        <start>/article[1]/body[1]/p[3]/text()[2].10</start>
        <end>/article[1]/body[1]/p[3]/text()[2].35</end>
      </anchor>
      <linkto>
        <file>13876.xml</file>
        <bep>/article[1]/sec[3]/p[8]<bep>
      </linkto>
    </link>
    ...
  </incoming>
</topic>
</inex-submission>
```

Appendix B

B-1 Official Result Set

Topics	# of Outgoing	# of Incoming	Topics	# of Outgoing	# of Incoming
Donald Bradman (87021.xml)	72	144	Dalai Lama (8133.xml)	71	237
Unified Modeling Language (32169.xml)	62	91	Within You Without You (1451526.xml)	13	11
Sukhoi Su-33 (552810.xml)	23	15	Software engineering (27010.xml)	107	404
Funk (10778.xml)	126	755	Philately (23681.xml)	41	108
Star Trek (26717.xml)	143	1649	Marie Curie (20408.xml)	75	127
Cartilage (166945.xml)	41	166	Stockholm syndrome (90910.xml)	49	36
Organic food (177593.xml)	73	50	Pink Floyd (24370.xml)	175	718
Pope Clement V (24102.xml)	69	56	Wavelet compression (50911.xml)	21	13
David (8551.xml)	124	513	Computer science (5323.xml)	241	1606
Aranyaka (321947.xml)	10	6	Pizza (24768.xml)	189	262
Greater Tokyo Area (354951.xml)	32	28	Joshua (16121.xml)	57	136
Xorn (322085.xml)	42	17	Skin cancer (64993.xml)	18	54
Kennewick Man (92818.xml)	47	10	Prince (artist) (57317.xml)	252	475
Frank Klepacki (752559.xml)	13	1	Family name (10814.xml)	165	474
University of London (60919.xml)	193	564	Search engine (27804.xml)	64	254
Latent semantic analysis (689427.xml)	16	10	Charleston, South Carolina (61024.xml)	200	947
Use case (300006.xml)	12	16	Elf (9896.xml)	235	378
Gout (55584.xml)	95	118	Akira Kurosawa (872.xml)	95	186
Thomas Edison (29778.xml)	132	358	Database (8377.xml)	99	186
Baylor University basketball scandal (493525.xml)	44	3	Radical feminism (25998.xml)	29	49
Search engine optimization (187946.xml)	49	45	Educational progressivism (10005.xml)	6	15
Civil Constitution of the Clergy (410450.xml)	40	34	Software development process (27565.xml)	49	33
Nokia (21242.xml)	48	196	Alastair Reynolds (69168.xml)	29	40
Achilles (305.xml)	124	219	Kazi Nazrul Islam (539155.xml)	31	20
Sunscreen (294419.xml)	38	46	Muammar al-Qaddafi (53029.xml)	159	149
Experiential education (447089.xml)	16	17	Neo-Byzantine architecture (1453013.xml)	36	5

Yitzhak Rabin (43983.xml)	77	145	Waseda University (376791.xml)	67	85
Triple J's Impossible Music Festival (2542756.xml)	103	1	Text Retrieval Conference (1897206.xml)	9	2
World Wide Web Consortium (33149.xml)	23	181	Autism rights movement (1305330.xml)	86	27
Excel Saga (265496.xml)	74	73	Ballpoint pen (4519.xml)	53	55
Link popularity (210641.xml)	20	6	Digital library (8794.xml)	13	43
Coca-Cola (6690.xml)	171	506	Sloe gin (392900.xml)	13	7
Entertainment robot (1451221.xml)	17	3	Koala (17143.xml)	70	104
Indira Gandhi (15179.xml)	100	199	Billie Holiday (50420.xml)	53	196
Leukemia (18539.xml)	64	403	Softball (80763.xml)	50	368
Miss Universe (150340.xml)	159	182	Information retrieval (15271.xml)	40	45
Neuilly-sur-Seine (234647.xml)	18	80	Cheminformatics (575697.xml)	13	17
Jihad (16203.xml)	56	254	Requirement (544592.xml)	9	27
Google (1092923.xml)	192	541	Susan Haack (321979.xml)	27	10
Joseph Stalin (15641.xml)	373	1324	Math rock (221484.xml)	72	49
Seasonal energy efficiency ratio (2189642.xml)	8	0	Transportation in the Faroe Islands (10704.xml)	18	0
Sony (26989.xml)	136	965	Anthropology (569.xml)	129	808
Doctor of Philosophy (8775.xml)	64	2110	Red Bull (61123.xml)	75	74
Taiwanese aborigines (53787.xml)	68	86	Lithography (18426.xml)	32	281
Hyperlink (49547.xml)	60	118	Isaac Newton (14627.xml)	207	611

B-2 Statistics of Submission Results

B-2-1 Incoming Measures on each Run sorted by MAP

Run ID	MAP	R-Prec	P5	P10	P20	P30	P50
QUT02	0.48412	0.50527	0.73556	0.68222	0.63278	0.58296	0.52756
QUT01	0.48283	0.50347	0.72889	0.67889	0.63167	0.58148	0.52911
Waterloo_LTW_01	0.46543	0.51183	0.66222	0.65333	0.60333	0.56963	0.51644
Otago_ltw-four	0.33908	0.37918	0.75111	0.68444	0.61278	0.55481	0.48400
Otago_ltw-five	0.31910	0.35566	0.72889	0.66333	0.60278	0.54037	0.46711
Otago_ltw-three	0.31784	0.36332	0.71778	0.67889	0.61778	0.55926	0.49933
Otago_ltw-two	0.28426	0.33115	0.63778	0.61222	0.55556	0.51963	0.47400
Amsterdam_LTW_01	0.22644	0.25834	0.70222	0.66222	0.57667	0.50519	0.39200
Otago_ltw-one	0.12265	0.15207	0.38000	0.36000	0.34278	0.32741	0.30644
Amsterdam_LTW03	0.10958	0.14368	0.62222	0.51333	0.36444	0.27704	0.18267
Amsterdam_LTW02	0.10846	0.16478	0.66000	0.51667	0.32667	0.24111	0.15711
Amsterdam_LTW04	0.09272	0.14180	0.64000	0.48889	0.33167	0.24407	0.15911
Amsterdam_LTW07	0.00398	0.01956	0.23778	0.16667	0.08833	0.05963	0.03578

B-2-2 Outgoing Measures on each Run sorted by MAP

Run ID	MAP	R-Prec	P5	P10	P20	P30	P50
QUT02	0.31774	0.41111	0.67111	0.62778	0.57722	0.52407	0.43689
QUT01	0.31366	0.41496	0.61333	0.62333	0.57944	0.52519	0.43956
Amsterdam_LTW_01	0.14679	0.18165	0.76667	0.68333	0.48556	0.35037	0.21178
Otago_ltw-four	0.10191	0.18334	0.44444	0.37889	0.30389	0.25593	0.20089
Otago_ltw-five	0.10052	0.18288	0.44000	0.37000	0.29944	0.25333	0.20067
Waterloo_LTW_01	0.09245	0.10282	0.61333	0.49000	0.32167	0.23148	0.15089
Otago_ltw-three	0.09228	0.17324	0.39778	0.36778	0.29667	0.25074	0.19800
Otago_ltw-two	0.08142	0.15586	0.36889	0.31889	0.26333	0.22630	0.18778
Amsterdam_LTW04	0.08056	0.14943	0.49778	0.42778	0.35167	0.28704	0.18822
Amsterdam_LTW02	0.08037	0.15378	0.46667	0.43444	0.35167	0.28852	0.19578
Amsterdam_LTW03	0.07326	0.14103	0.47778	0.42111	0.34722	0.27667	0.17889
Amsterdam_LTW07	0.06709	0.12734	0.50000	0.42556	0.32056	0.24667	0.15000
Otago_ltw-one	0.04763	0.09781	0.23556	0.20778	0.17278	0.15778	0.12800

Wikipedia *Ad hoc* Passage Retrieval and Wikipedia Document Linking

Dylan Jenkinson and Andrew Trotman

Department of Computer Science
University of Otago
Dunedin
New Zealand
{djenkins, andrew}@cs.otago.ac.nz

Abstract. *Ad hoc* passage retrieval within the Wikipedia is examined in the context of INEX 2007. An analysis of the INEX 2006 assessments suggests that fixed sized window of about 300 terms is consistently seen and that this might be a good retrieval strategy. In runs submitted to INEX, potentially relevant documents were identified using BM25 (trained on INEX 2006 data). For each potentially relevant document the location of every search term was identified and the center (mean) located. A fixed sized window was then centered on this location. A method of removing outliers was examined in which all term occurring outside one standard deviation of the center were considered outliers and the center recomputed without them. Both techniques were examined with and without stemming. The best technique in focused retrieval and relevant-in-context retrieval used outlier removal and stemming. The best run for best-in-context used outlier reduction without stemming.

For Wikipedia linking we identified terms within the document that were over represented and from the top few generated queries of different lengths. A BM25 ranking search engine was used to identify potentially relevant documents. Links from the source document to the potentially relevant documents (and back) were constructed (at a granularity of whole document). The best performing run used the 4 most over represented search terms to retrieve 200 documents, and the next 4 to retrieve 50 more.

1. Introduction

The University of Otago participated in new tasks introduced to INEX in 2007. In the passage retrieval task three runs were submitted to each of the focused, relevant-in-context and best-in-contest tasks (and one run held-back). In the Link-the-Wiki track five runs were submitted. In all cases performance was adequate (middle of the pack or better).

An analysis of the 2006 INEX assessments (topics version:2006-004, assessments version:v5) shows that documents typically contain only one relevant passage, and that that passage is 301 characters in length. This leads to a potential retrieval strategy of first identifying potentially relevant documents, then from those

identifying the one potentially relevant passage (of a fixed length). In essence this has reduced the passage retrieval problem to that of placing a fixed sized window on the text.

The approach we took was to identify each and every occurrence of each search term within the document. From there the mean position was computed and the window centered there. Outliers could potentially effect the placement of the window so an outlier reduction strategy was employed. All occurrences lying outside one standard deviation of the mean were eliminated and the mean recomputed. This new mean was used to place the window.

Porter stemming [6] was tested in combination with and without outlier reduction. Of interest to XML-IR is that our approach does not use document structure to identify relevant content. Kamps & Koolen [4] suggest relevant passages typically start (and end) on tag boundaries, however we leave exploitation of this to future work.

Our best passage retrieval run when compared to element retrieval runs of other participants ranked 29th of 79 in the focused task, 33rd of 66 runs in the relevant-in-context task, and 40th of 71 in the best-in-context task. That run used outlier reduction and only in the case of best-in-context was stemming not useful.

In the Link-the-Wiki task we again ignored document structure and used a naive method. A score for each term in the orphaned document was computed as the ratio of length normalized document frequency to the expected frequency computed from collection statistics. Terms were ranked then queries of varying length (from 1 to 5 terms) were constructed from the top ranked terms in the list.

No attempt was made to identify anchor text or best entry points into target documents – instead linking from document to document was examined. We found that in this kind of linking query lengths of 4 terms performed best.

2. *Ad hoc* Passage Retrieval

The INEX evaluation forum currently investigates subdocument (focused) information retrieval in structured documents, specifically XML documents. Focused retrieval has recently been defined as including element retrieval, passage retrieval and question answering [11]. In previous years INEX examined only element retrieval but in 2007 this was extended to include passage retrieval and book page retrieval. Common to all these paradigms is the requirement to return (to the user) only those parts of a document that are relevant, and not the whole document.

These focused searching paradigms are essentially identical and can be compared on an equal basis (using the same queries and metrics). If an XML element is specified using the start and end word number within a document (instead of XPath) then an XML element can be considered a passage. The same principle is true of a book page if word numbers are used instead of page numbers. A question answer within the text can also be considered a passage if it, too, is consecutive in the text.

Our interest in passage retrieval is motivated by a desire to reduce the quantity of irrelevant text in an answer presented to a user, that is, to increase focused precision. We believe that element granularity is too coarse and that users will necessarily be

presented with irrelevant text along with their answers because any element large enough to fully contain a relevant answer is also likely to be sufficiently large that it contains some irrelevant text. Exactly this was examined by Kamps & Koolen [4] who report that, indeed, the smallest element that fully contains a relevant passage of text often contains some non-relevant text. The one way to increase precision is to remove the irrelevant text from the element, one obvious way to do this is to shift to a finer granularity than element, perhaps paragraph, sentence, word, or simply passage.

2.1. INEX 2007 Tasks

There were three distinct retrieval tasks specified at INEX 2007: focused retrieval; relevant-in-context retrieval; and best-in-context retrieval. In focused retrieval the search engine must generate a ranked non-overlapping list of relevant items. This task might be used to extract relevant elements from news articles for multi-document summarization (information aggregation).

The relevant-in-context task is user-centered, the aim is to build a search engine that presents, to a user, a relevant document with the relevant parts of that document highlighted. For evaluation purposes documents are first ranked on topical relevance then within the document the relevant parts of the document are listed.

Assuming a user can only start reading a document from a single point within a document, a search engine should, perhaps, identify that point. This is the aim of the best-in-context task, to rank documents on topical relevance and then for each document to identify the point from which a user should start reading in order to satisfy their information need.

For all three tasks both element retrieval and passage retrieval are applicable. For both it is necessary to identify relevant documents and relevant text within those documents. For element retrieval it is further necessary to identify the correct granularity of element to return to the user (for example, paragraph, sub-section, section, or document). For passage retrieval it is necessary to identify the start and end of the relevant text. It is not yet known which task is hardest, or whether structure helps in identification of relevant text within a document. It is known that the precision of a passage retrieval system must, at worst, be at least equal to that of an element retrieval system.

2.2. Passage Retrieval

Passages might be specified in several different ways: an XML element, a start and end word position, or any granularity in-between (sentences, words, and so on). The length of a passage can be either fixed or variable. Within a document separate passages might either overlap or be disjoint.

If element retrieval and passage retrieval are to be compared on an equal basis it must be possible to specify an XML element as a passage. This necessitates a task definition that allows variable sized passages. Interactive XML-IR experiments show that users do not want overlapping results [10], necessitating a definition of disjoint passages. The INEX passage retrieval tasks, therefore, specify variable length non-

overlapping passages that start and end on word boundaries. We additionally chose to ignore document structure as we are also interested in whether document structure helps with the identification of relevant material or not.

2.3. Window Size

Previous experiments suggest that fixed sized windows of between 200 and 300 words is effective [2]. To determine the optimal size for the Wikipedia collection an analysis of the INEX 2006 results was performed.

In 2006 INEX participants assessed documents using a yellow-highlighting method that identified all relevant passages within a document. For each passage the start and end location are given in XPath and the length is given in characters. Best entry points are also specified.

Kamps & Koolen [4] performed a thorough analysis of the assessments and report a plethora of statistics. We reproduce some of those analyses, but present results in a different way.

Figure 1 presents the number of relevant documents in the assessment set that contain the given number of passages. The vast majority of relevant documents (70.63%) contain only one relevant document. This suggests that any passage retrieval algorithm that chooses to identify only one relevant passage per document will be correct the majority of the time. Because it is reasonable to expect only one relevant passage per document the tasks can be simplified to identifying *the relevant passage* in a document, not the relevant *passages* within a document. 17.27% contain 2 passages and 12.10% contain 3 or more passages.

Figure 2 presents the mean passage length (in words) of a passage as the number of passages within a document increases. It was reasonable to expect that as the number of passages increased that the mean length of the passage would decrease as there is a natural limit on the sum of the lengths (the document length). Instead it can be seen that the average length is about constant. In a multiple-assessor experiment on the same document collection Trotman *et al.* [12] asked assessors whether they preferred to identify fixed-sized passages or variable sized passages and found that half preferred fixed sized passages of about a paragraph in length. This is consistent with the observation that passages are all about the same length – when a single passage is seen the mean is 283 words, but if more than one passage is sent then it varies between 73 and 153 words. Given this is the case then it is reasonable to expect that the length of a document is related to the number of passages it contains – this is shown to be the case in Figure 3 where it can be seen that document length increases with number of passages.

The mean relevant content per document is 301 words. In Figure 4 the length of all relevant passages in all documents is presented – very few passages are long (over 1000 words) or short (under 10 words).

Given the mean length of relevant content in a document is about 300 words, and that only one passage is expected per document, it is reasonable to develop a passage retrieval algorithm that identifies one passage of 300 words. There does, however, remain the problem of identifying where, within a document, that passage should be placed.

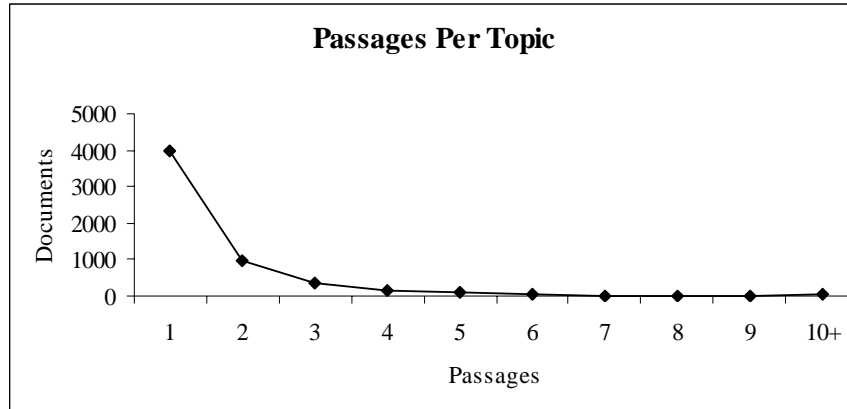


Figure 1: Number of documents containing the given number of passages.

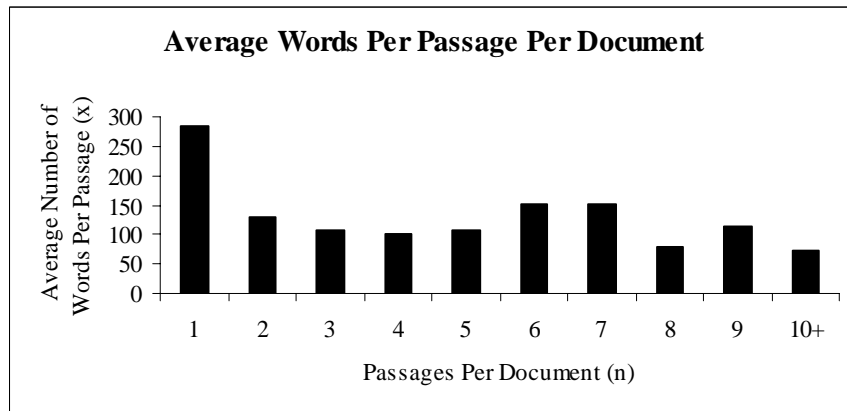


Figure 2: Passage length varies with number of passages per document.

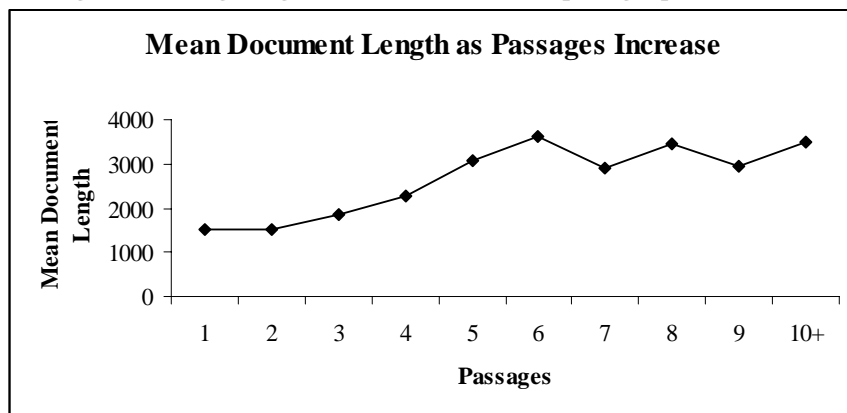


Figure 3: mean document length as the number of passages increases.

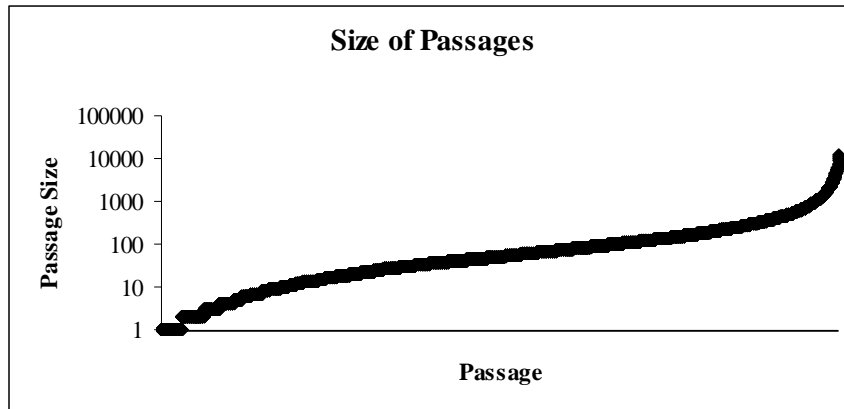


Figure 4: Log of passage size for all relevant passages.

2.4. Window Location

A heat map of the document can be built by noting the location of all search terms within the document. Areas where search terms do not occur (cold areas) are unlikely to be relevant to the user's query; conversely areas where there are many occurrences of the search terms (hot areas) are likely to be relevant.

Our hypothesis is that centering the one fixed-sized window over the middle of the dense areas will be an effective retrieval strategy. This method ignores the structure of the document, which we believe makes the comparison to element-retrieval systems of particular interest. Our method is as follows:

For each document identified as potentially relevant the XML structure is removed and the location of all occurrences of all search terms is identified. The mean of these locations is considered to be the center of relevance and so the window is centered on this point. If the window extended outside the document (before the beginning for example) then the window is truncated at the document boundary.

Problematically, in a well structured document it is reasonable to assume search terms will occur in the abstract and conclusions, but for the relevant text to occur elsewhere, in the body of the document for example. Several early or late term occurrences might shift the window towards the outliers which will in turn reduce precision. A method is needed to identify and remove outliers before the window is placed. We hypothesize that removing outliers will increase precision.

Two window placement methods were implemented: *meanselection* and *stddevselection*. With *meanselection* the center point (mean) of all occurrences of all search terms was used. With *stddevselection* the mean search term position was found and the standard-deviation computed. Then all occurrences outside one standard deviation from the mean were discarded. A new mean was then computed from the pruned list, and this was used as the passage midpoint.

2.5. Stemming

The identification of search terms within the document is essential to the performance of the window placement technique. It is reasonable to expect authors to use different morphological variants and synonyms of search terms within their documents. The inclusion of these in the algorithms is, therefore, important. We experimented with Porter's stemming algorithm [6].

2.6. Potentially Relevant Documents

The identification of relevant documents in *ad hoc* retrieval has been studied extensively by others. Several effective methods have been presented including language models [13], pivoted cosine normalization [9], and BM25 [7]. We chose BM25.

BM25 is parametric and requires scores for $k1$, $k3$ and b . We used genetic algorithms [1] and trained on the INEX 2006 data to obtain good scores, the details are immaterial, however it resulted in the values 0.487, 25873, and 0.288 for $k1$, $k3$ and b respectively.

Stemming was not used during training and was not used to identify potentially relevant documents

2.7. Best Entry Points

Kamps *et al.* [5] show a correlation between the best entry point and the start of the first relevant passage. They report 67.6% of best entry points in a single-passage document lying at the start of the passage (17.16% before and 15.24% after). For a document with two passages these numbers are substantially different. The chance that the best entry point coincides with the start of the first passage in the document is reduced to 35.33%, whilst the chance that the best entry point is before the first passage is increased to 45.21%. The chance of the best entry point coming after the first passage is about 19.46%. Figure 5 presents our analysis. It shows, for all documents with a single relevant passage, the distance (in characters) from the start of that passage to the best entry point. The vast majority of all passages start at or very close to the best entry point. This suggests a best entry point identification strategy of "just choose the start of the first relevant passage".

3. Ad Hoc Experiments

3.1. Ad Hoc Runs

We conducted two experiments: the first was the effect of stemming, the second was the effect of removing outliers. This gave 4 possible combinations (runs) for each task as outlined in Table 1, however we were only permitted to submit 3 official runs per task and so the last run was scored informally. We expect the performance with standard-deviation and stemming to be most effective as this run will be better at identifying occurrences of search terms, while also better at removing outliers.

The same runs were submitted to each of the *ad hoc* tasks (focused, relevant-in-context, and best-in-context) and the runs differ only in name.

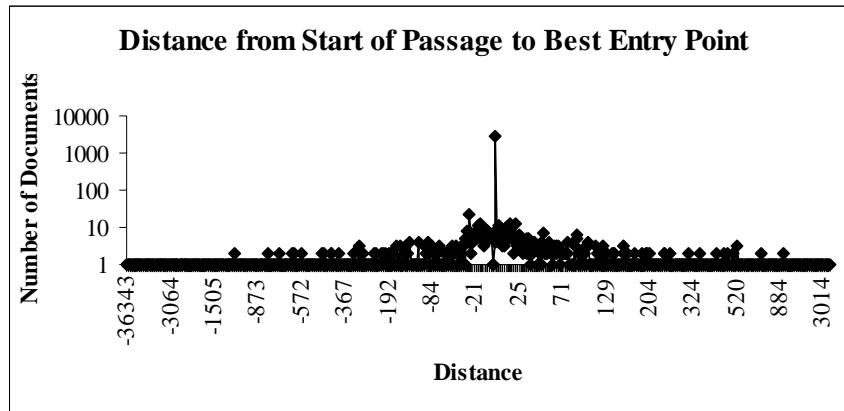


Figure 5: Distance (in characters) of the best entry points from the start of the first passage. Negative are before the first passage.

Table 1. Runs submitted to the INEX 2007 *ad hoc* track.

Run	Focused	Relevant-in-context	Best-in-context
1	DocsNostem-PassagesStem-StdDevYes-Focused	DocsNostem-PassagesStem-StdDevYes	DocsNostem-PassagesStem-StdDevYes-BEP
2	DocsNostem-PassagesStem-StdDevNo-Focused	DocsNostem-PassagesStem-StdDevNo	DocsNostem-PassagesStem-StdDevNo-BEP
3	DocsNostem-PassagesNoStem-StdDevNo-Focused	DocsNostem-PassagesNoStem-StdDevNo	DocsNostem-PassagesNoStem-StdDevNo-BEP
4	DocsNostem-PassagesNoStem-StdDevYes-Focused	DocsNostem-PassagesNoStem-StdDevYes	DocsNostem-PassagesNoStem-StdDevYes-BEP

3.2. Ad hoc Results

Table 2 presents the scores and relative rank of the focused runs. The best run used stemming and the *stddevselection* method. Both stemming runs performed better than the no-stemming runs. This suggests that stemming has a greater effect than standard-deviation pruning at focused retrieval. The relative rank of all runs is similar (about 29th); the differences are small.

Of particular note is that of the 79 runs submitted to the task our runs that did not use document structure placed well (37%).

Table 2. Focused task results computes at 0.01 recall. ⁺values computed locally.

Run	iMAP	iMAP ⁺	Rank
DocsNostem-PassagesStem-StdDevYes-Focused	0.3617	0.3639	29
DocsNostem-PassagesStem-StdDevNo-Focused	0.3562	0.3582	31
DocsNostem-PassagesNoStem-StdDevYes-Focused	-	0.3517	-
DocsNostem-PassagesNoStem-StdDevNo-Focused	0.3498	0.3515	33

The performance of the runs submitted to the relevant-in-context task is shown in Table 3. Here there is no material difference in the score of the runs. As with focused retrieval, stemming and standard-deviation selection appears most effective. Of 66 runs submitted to the task our top run that ignores structure performed in the middle of the pack (33rd)

Table 3. Relevant-in-context results. ⁺values computed locally.

Run	MAgP	MAgP ⁺	Rank
DocsNostem-PassagesStem-StdDevYes	0.0653	0.0659	33
DocsNostem-PassagesStem-StdDevNo	0.0653	0.0657	34
DocsNostem-PassagesNoStem-StdDevNo	0.0651	0.0655	36
DocsNostem-PassagesNoStem-StdDevYes	-	0.0646	-

The performance with respect to the best-in-context task is shown in Table 4. Here outlier reduction was effective but stemming was not. The relative system performance of our best submitted run was 40 of 71. The un-submitted run placed between ranks 39 and 40.

Table 4. Best-in-context results. ⁺values computed locally.

Run	MAgP	MAgP ⁺	Rank
DocsNostem-PassagesNoStem-StdDevYes-BEP	-	0.1101	-
DocsNostem-PassagesStem-StdDevYes-BEP	0.1082	0.1084	40
DocsNostem-PassagesStem-StdDevNo-BEP	0.1076	0.1066	41
DocsNostem-PassagesNoStem-StdDevNo-BEP	0.1073	0.1062	42

3.3. Discussion

We chose to ignore document structure and submitted run that, instead, simply used term locations to place a fixed sized window on the text. From the relative system performance it is reasonable to conclude that selecting a single fixed sized passage of text produces reasonable results.

The stemming experiment shows that stemming is important for choosing the location of the window. When searching a very large document collection it is reasonable to ignore stemming because any relevant document will satisfy the user's information need. This is not the case when looking within a single document where missing some occurrences of morphological variants of search terms has an effect on window placement and system performance.

The use of the *stddevselection* method for selecting the centre point of a passage produced better results than the *meanselection* method. That is, there are, indeed, outliers in the document that effect window placement.

4. Link-the-Wiki

In 2007 INEX introduced a new track, Link-the-Wiki. The aim is to automatically identify hypertext links for a new documents when added to a collection [3]. The task contains two parts, the identification of out-going links to other documents in the collection and the identification of in-going links from other documents to the new document. In keeping with the focused retrieval theme, links are from passages of text (anchor text) to best entry points in a target document. In 2007, as the task is new, a reduced version of the track was run in which the task is simply document to document linking (both incoming and outgoing) [3]. Participants were also asked to supply information about the specifications of the computer used to generate the results, and the time taken to perform the generation. We used Intel Pentium 4, 1.66GHz, single core, no hyper-threading, and only 512MB memory. Our execution times were all less than 4 minutes and are presented in Table 5.

4.1. Themes

Almost all words or phrase in a document could be linked to another document (if for no other reason than to define the term). The task, therefore, is not the identification of links, but the identification of salient links. The approach we took was the identification of themes (terms) that are over-represented within the document, and the identification of documents about those themes. Our approach is based on that of Shatkay & Wilbur [8].

An over represented term is a term that occurs more frequently with in the source document than expected, that is, the document is more about that term that would be expected if the term was used *ordinarily*. The actual frequency (*af*) of a term within the document is computed as the term frequency (*tf*) over the document length (*dl*).

$$af = \frac{tf}{dl}$$

The expected frequency (*ef*) of the term is computed on the prior assumption that the term does occur within the document. Given the collection frequency (*cf*) and the document frequency (*df*), and the average length of a document (*ml*), this is expressed as

$$ef = \frac{cf}{df \times ml}$$

The amount by which the term is over represented (*repval*) in the document is the ratio of the actual frequency to the expected frequency.

$$repval = \frac{af}{ef}$$

Terms that occur in a document but not the collection are assigned negative scores.

4.2. Link-the-Wiki Runs

We generated document to document linking runs using a relevance ranking search engine that used BM25 ($k1=0.421$, $k3=242.61$, $b=0.498$). Incoming links and outgoing links were strictly reciprocal, that is, the list of incoming links was generated from the outgoing list by reversing the direction of each link (and maintaining the relative rank order).

The runs were generated thus:

First the source (orphan) document was parsed and a list of all unique terms and *repval* scores was generated. Stop words were removed from the list.

Five runs were generated from the term list. In the first the single most over-represented term was used to generate a query for which we searched the collection returning the top 50 documents. The second term was then used to identify the next 50 documents, and so on until 250 documents had been identified.

In the second run the top two terms were used and 100 documents identified. 100 more for the third and fourth term, and 50 for the sixth and seventh term. In the third run triplets of terms were used to identify 150 documents each. In the fourth run quads of terms were used, and in the final run sets of 5 terms were used to identify all 250 documents. The details are outlined in Table 5.

In our experiment the total length of the result set was held constant (at 250) and the number of documents retrieved per search terms was held constant (at 50). The aim of our experiment was to identify whether or not there was a query-length effect in identifying related documents.

Table 5. Runs submitted to the Link-the-Wiki track.

Run	Query length	Results per query	Time
ltw-one	1	50/50/50/50/50	134s
ltw-two	2	100/100/50	170s
ltw-three	3	150/100	161s
ltw-four	4	200/50	225s
ltw-five	5	250	124s

4.3. Results

The performance of the runs measured using mean average precision (MAP) is presented in Table 6. The relative rank order of our runs for both incoming and outgoing links was the same. The best run we submitted performed 4th of 13 submitted runs.

Figure 6 graphs outgoing precision (and Figure 7 incoming precision) at early points in the results list. Comparing the two, the technique we used is far better at identifying incoming links than outgoing links. When compared to runs from other participants, our best incoming precision at 5 and 10 documents placed first.

Table 6: Link-the-Wiki results.

Run	Outgoing		Incoming	
	MAP	Rank	MAP	Rank
ltw-four	0.102	4	0.339	4
tw-five	0.101	5	0.319	5
ltw-three	0.092	7	0.318	6
ltw-two	0.081	8	0.284	7
ltw-one	0.048	13	0.123	9

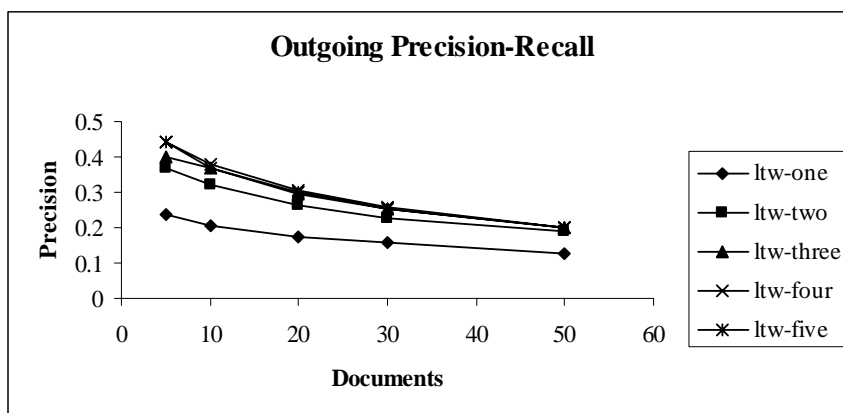


Figure 6: Precision – Recall of outgoing links.

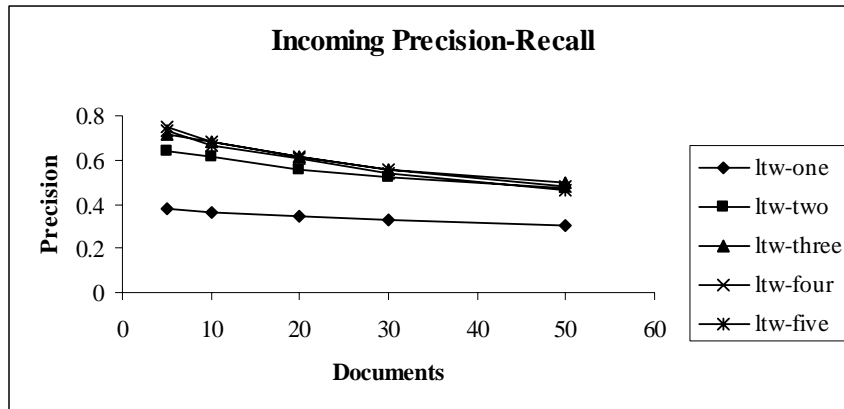


Figure 7: Precision – Recall of incoming links.

4.4. Discussion

We experimented with queries of different length and discovered that queries of 4 terms work better than either longer or shorter queries. When adding search terms to a query there comes a point at which the query becomes general resulting in the retrieval any an increasing number of irrelevant documents. This point appears to be 4 terms.

Of particular interest to us is the difference in performance of incoming and outgoing links. We constructed outgoing links from a document using a simple technique to identify terms that were over represented. Incoming links were simply the same list inverted in direction. The technique appears capable of identifying the salient concepts within the document (such that it might be beneficial to link to), but not extracting from a document concepts that require further details (such that it might be beneficial to link from).

Our results suggests a future strategy in which the technique we used is applied to all documents to identify incoming links, and flipping those to get outgoing links for a document. This is, however, likely to be computationally expensive.

5. Conclusions

Passage retrieval and link discovery in the Wikipedia was examined in the context of INEX 2007. For both tasks naïve methods that ignored document structure were studied. We found that for passage retrieval both stemming and outlier reduction were effective. In link discovery we found that queries containing 4 search terms was effective.

In future work we intend to extend our naive methods and to include document structures. Others have already shown that relevant passages typically start and end on tag boundaries, none the less we chose to ignore structure. Methods of using

structure in passage length identification will be examined for passage retrieval and use for BEP identification will be used for link identification.

We intent to examine the granularity of structural markup necessary before good ranking performance can be expected. Even though we chose to ignore structure the performance of our runs was reasonable when compared to those of others. This raises the question of the value of the structural markup within a document when used for relevance ranking.

The Link-the-Wiki runs we submitted also performed adequately. Queries of various length were constructed from concept terms. The concept terms were extracted from the orphaned document by taking terms overly represented in the document. The best query length we found was 4 terms.

The technique was better at identifying incoming links than outgoing links – that is, the technique identifies the concepts of the document and not concepts that require further expansion. Future work will examine fast and efficient ways to identify outgoing links.

6. Acknowledgements

Funded in part by a University of Otago Research Grant.

7. References

- [1] Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor: University of Michigan Press.
- [2] Huang, W., Trotman, A., & O'Keefe, R. A. (2006). Element retrieval using a passage retrieval approach. *Australian Journal of Intelligent Information Processing Systems (AJIIPS)*, 9(2):80-83.
- [3] Huang, W. C., Trotman, A., & Geva, S. (2007). Collaborative knowledge management: Evaluation of automated link discovery in the Wikipedia. In *Proceedings of the SIGIR 2007 Workshop on Focused Retrieval*, 9-16.
- [4] Kamps, J., & Koolen, M. (2007). On the relation between relevant passages and XML document structure. In *Proceedings of the SIGIR 2007 Workshop on Focused Retrieval*, 28-32.
- [5] Kamps, J., Koolen, M., & Lalmas, M. (2007). Where to start reading a textual XML document? In *Proceedings of the 30th ACM SIGIR Conference on Information Retrieval*.
- [6] Porter, M. (1980). An algorithm for suffix stripping. *Program*, 14(3):130-137.
- [7] Robertson, S. E., Walker, S., Beaulieu, M. M., Gatford, M., & Payne, A. (1995). Okapi at TREC-4. In *Proceedings of the 4th Text REtrieval Conference (TREC-4)*, 73-96.
- [8] Shatkay, H., & Wilbur, W. J. (2000). Finding themes in medline documents probabilistic similarity search. In *Proceedings of the Advances in Digital Libraries*, 183-192.

- [9] Singhal, A., Buckley, C., & Mitra, M. (1996). Pivoted document length normalization. In *Proceedings of the 19th ACM SIGIR Conference on Information Retrieval*, 21-29.
- [10] Tombros, A., Larsen, B., & Malik, S. (2004). The interactive track at INEX 2004. In *Proceedings of the INEX 2004 Workshop*, 410-423.
- [11] Trotman, A., Geva, S., & Kamps, J. (2007). *Proceedings of the sigir 2007 workshop on focused retrieval*.
- [12] Trotman, A., Pharo, N., & Jenkinson, D. (2007). Can we at least agree on something? In *Proceedings of the SIGIR 2007 Workshop on Focused Retrieval*, 49-56.
- [13] Zhai, C., & Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. *Transactions on Information Systems*, 22(2):179-214.

University of Waterloo at INEX2007: Ad Hoc and Link-the-Wiki Tracks

Kelly Y. Itakura and Charles L. A. Clarke

University of Waterloo, Waterloo, ON N2L3G1, Canada,
{yitakura, claclark}@cs.uwaterloo.ca

Abstract. In this paper, we describe University of Waterloo’s approaches to ad hoc and Link-the-Wiki tracks. In ad hoc track, we submitted runs for the focused and the best-in-context tasks. We again show that Okapi BM25 works well for XML retrieval. We also analyze why our element-based best entry point result is better than our passage-based counterpart. Finally, we present our baseline algorithm for embedding incoming and outgoing links in Link-the-Wiki track.

1 Introduction

In 2007, University of Waterloo participated in ad hoc and Link-the-Wiki tracks. In ad hoc track, we implemented passage retrieval and element retrieval to turn these results into submissions for the focused and the best-in-context tasks. For the focused task, we only submitted an element retrieval result that used the same algorithm as Waterloo’s focused submission in INEX2004. In the best-in-context task, we submitted element results based on both element and passage retrieval. In Link-the-Wiki track, since it is the first year of administration, we decided to submit runs using relatively simple techniques that might be suitable as a baseline for future work.

This paper is organized as follows. In Sect. 2, we describe our approaches to ad hoc track, and in Sect. 3, we describe our approaches to Link-the-Wiki track. We conclude this paper with directions for future work in Sect. 4.

2 Ad hoc Track

In ad hoc track, we used two retrieval schemes, element retrieval and passage retrieval to return XML elements for the focused task and best entry points for the best-in-context task.

Both element and passage retrieval work in essentially the same manner. We converted each topic into a disjunctive of query terms, removing negative query terms. We located positions of all query terms and XML tags using Wumpus [1]. We then used a version of Okapi BM25 [5] to score passages and elements. The score of an element/passage P is defined as follows.

$$s(P) \equiv \sum_{t \in Q} W_t \frac{f_{P,t}(k_1 + 1)}{f_{P,t} + k_1(1 - b + b \frac{pl_P}{avgdl})} , \quad (1)$$

where Q is a set of query terms, $f_{P,t}$ is the sum of term frequencies in a passage P , pl_P is a passage length of P , and $avgdl$ is an average document length in Wikipedia collection. We tuned parameters k and b using INEX2006 ad hoc track focused and best-in-context tasks and the accompanying nxCG and BEPD metrics respectively. The actual parameters used for element retrieval for focused task is $k = 1.2$ and $b = 0.9$, for best-in-context task, $k = 0.8$ and $b = 0.7$. For passage retrieval in best-in-context task, we chose $k = 1.4$ and $b = 0.7$. This is interesting because when we worked on INEX2004/2005 IEEE collection [2] [3], we speculated that a large k is necessary for Okapi-based passage retrieval to work. However, it seems that this is not the case for Wikipedia corpus.

In element retrieval, we scored all of the following most common elements in corpus.

```
<p>, <section>, <normallist>, <article>, <body>, <td>, <numberlist>,
<tr>, <table>, <definitionlist>, <th>, <blockquote>, <div>, <li>,
<u>.
```

In passage retrieval, we scored all possible passages. For both algorithms, we ignored elements/passages of size less than 25 word-long.

2.1 Focused Task

In the focused task, we returned the top 1500 elements obtained from element retrieval after removal of nestings. In INEX2007 official metrics, we ranked 5th among different organizations, which indicates that both our approach and our scoring scheme, Okapi BM25, work well.

2.2 Best-in-Context Task

For the first submission, we used element retrieval to obtain the top 1500 elements with distinct files. For the second submission, we used passage retrieval to choose the best scoring passage for each file. We then chose the top 1500 among these. We returned the XML tags listed above nearest to these 1500 passages that is closer to the beginning of the article.

The official INEX2007 results show that our element-based approach ranked 2nd among different organizations. However, it is to our surprise that our passage-based approach did not work as well as our element-based approach. Our initial assumption was that since elements are passages, the highest scoring passage would give a better best entry point than the highest scoring element. After looking at the official assessments set which we will treat as a gold standard for the purpose of our analysis, we speculate two causes for our under-achieving passage-based result. First, highest scoring passages do not tend to appear at the beginning of an article, whereas as shown in [4], the gold standard tend to appear at the beginning of an article. However, this does not explain why highest scoring elements give a better result. By examining the assessment set, we speculate that the performance of our passage-based approach is largely explained by the gap in relevant information content between the highest scoring

passage and the best entry point derived from it. This is because XML elements returned as the best entry point from the top passages almost always have much lower score than the highest scoring element, which indicate that there is a lot of irrelevant material between the start of the highest scoring passage and the best entry point associated with it. Therefore, we think that the best entry point must be very close to the highest scoring passage. This leads to a preference towards either highest scoring passages or highest scoring elements over elements starting before the highest scoring passages. Raw passage results, however, do not seem to appear frequently in the relevant assessment. Moreover, we could see our passage-based element best entry point to be *context BEPs* as in [4], and since there are many relevant passages in the gold standard, we think that the preference is more towards the highest-level element that contain all relevant passages, termed *container BEPs* [4]. The exact same phenomena also apply to results of our training set on INEX2006.

For future work, instead of returning the nearest significant XML elements that start before the highest scoring passages do, we plan to return the nearest significant XML elements that start *after* the highest scoring passages. We hope that in this way, there would be no irrelevant material between the proposed best entry point and the highest scoring passages. Additionally, we hope that by tuning Okapi parameters well the resulting best entry points would be closer to the beginning of articles. Another way to avoid an information gap is to set passages to start at element boundaries, which is a generalization of element-based best entry point that we performed well.

3 Link the Wiki Track

This year, we decided to submit a result set made from a simple algorithm to act as a baseline. Before working on outgoing or incoming links, we removed all topic files from corpus. When creating a list of anchor-destination pairs for each corpus file, we also ignored pairs that have a topic file as the destination.

3.1 Outgoing Links

To create outgoing links from topic files, we first created for each file in the corpus, a list of outgoing links specified by an anchor term a and the destination file d . We then selected the most frequent target d for each anchor a over all titles and then computed the following ratios γ .

$$\gamma = \frac{\# \text{ of pages that has a link from anchor } a \text{ to a file } d}{\# \text{ of pages in which } a \text{ appears at least once}}$$

We set all destinations to the entire articles. We only picked those terms whose γ value is above certain threshold, in this case, 0.6.

For example, an anchor term, *bacteria*, appears most often with the destination file 3752.xml for 1197 times. There are 1981 number of files that contain the term *bacteria*. The value of γ for *bacteria* is then $1197/1981 = 0.604$ which is over

0.6. Similarly, there is another anchor term, *proteobacteria* with the most frequent destination file 24863.xml for 159 times. There are 161 number of files that contain the term *proteobacteria*, and the value of $\gamma = 159/161 = 0.988$ is also above the threshold of 0.6. Therefore, we add both *bacteria* and *proteobacteria* to our list of anchors.

Next, we found the first positions of each anchor in every topic file using Wumpus [1], then linked the anchors to the corresponding destinations. If an anchor *a* is a substring of another anchor *b*, we chose the longer anchor to make a link from.

For example, suppose a position 1234 in topic files contain a term *proteobacteria*. Then we make a link to a file 24863.xml, not to a file 3752.xml.

To see how we perform for various probability thresholds, we plotted a precision/recall graph for thresholds varying from 10% to 90%. We computed precision by how many outgoing links we embedded in topics appear in the original topic files in corpus. We computed recall as how many outgoing links in the original topic file in corpus appear in our embedded topic files. Figure 1 shows that precision increases as the threshold increases, and the precision is generally good. The recall decreases as thresholds increases as expected, however, the overall recall is fairly low. Therefore, it suggests that we need additional ways to identify outgoing links while still keeping the high accuracy.

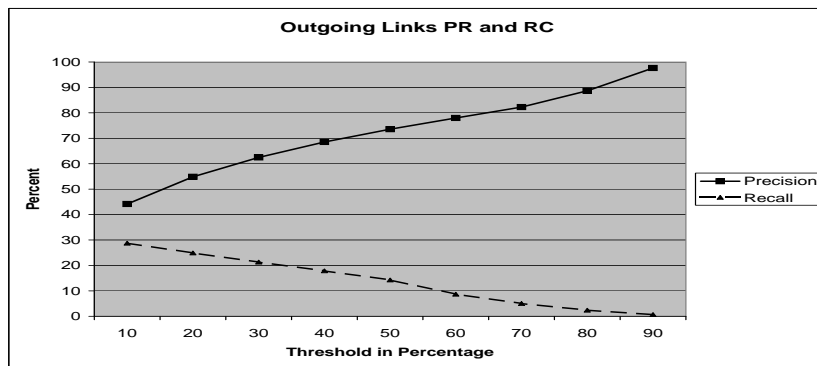


Fig. 1. Precision and Recall Plot at Various Thresholds

Official results for outgoing links show that we achieve quite high precisions at early levels. This is because with 60% threshold, we did not return many outgoing links, and so recall is low as in Figure 1. We discovered that we did not return a ranked list of anchors as specified in the use case, but instead returned

all anchors in the order of appearance. Therefore, in this paper, we decided to use a similar methodology to return a ranked list of outgoing links.

Instead of making a list of anchor terms by ignoring anchors with γ values below a certain threshold, we decided to make a list of anchor terms with the values of γ . We then found in topic files all occurrences of anchor terms in the list, and returned the anchors with the top 250 γ values. Figure 2 and Tab. 3.1 show that ranking by the γ values greatly increase the scores in official metrics and achieve the highest scores among all organizations participated for outgoing links. Concavity of Fig. 2 may be due to files that have less than 250 results.

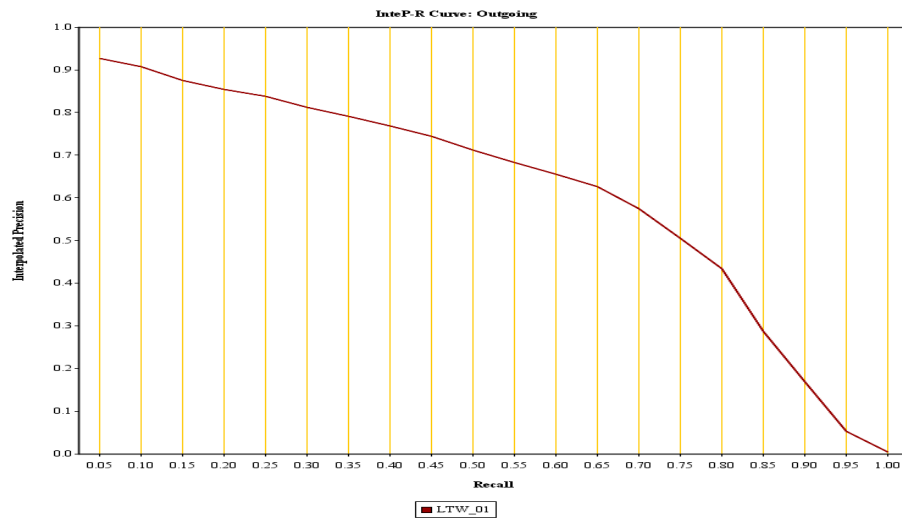


Fig. 2. Interpolated Precision and Recall for Ranked Outgoing Links

3.2 Incoming Links

We decided to work at an article level for incoming links. That is, both a source and the destination are articles. For each topic title, we chose the first 250 pages using Wumpus [1] that have the topic title without an intra-corporis link from

	MAP	R-Prec	P5	P10	P20	P30	P50
Official Unranked Outgoing	0.092	0.103	0.613	0.490	0.322	0.231	0.151
Unofficial Ranked Outgoing	0.607	0.628	0.849	0.816	0.75	0.698	0.614
Official Best of All Org.	0.318	0.415	0.767	0.683	0.579	n/a	0.440

Table 1. Ranked v.s. Unranked Using Official Metrics

the title. We then returned a result set that consists of the first 250 pages as the source and the topic title as the destination.

The official result in Fig. 3 shows that although our precision decreases as the rank increases, our performance relative to other submissions increases. We expect that if we did not simply choose the first 250 pages to return, our precision would increase overall. Figure 4 is the final result for our ranked outgoing and incoming links combined using the official evaluation software. Table 3.2 shows scores of different official metrics for our incoming and combined submissions.

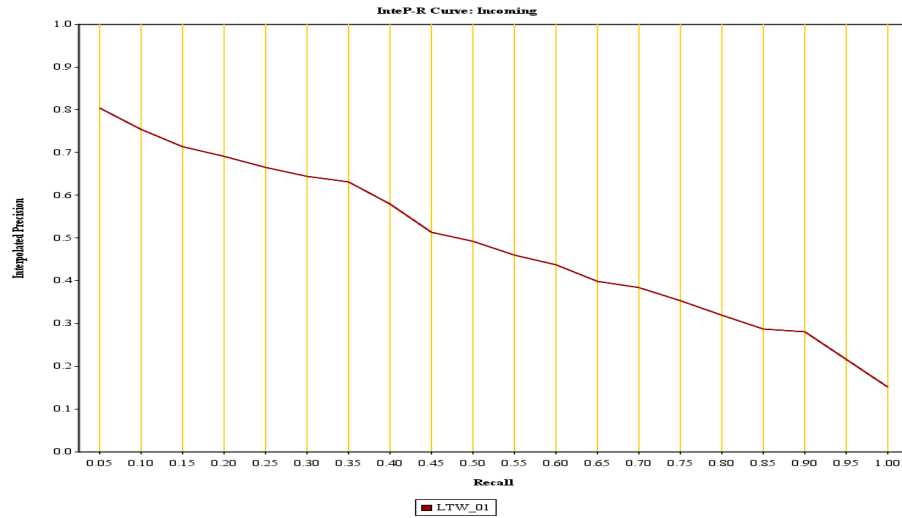


Fig. 3. Interpolated Precision and Recall for Incoming Links

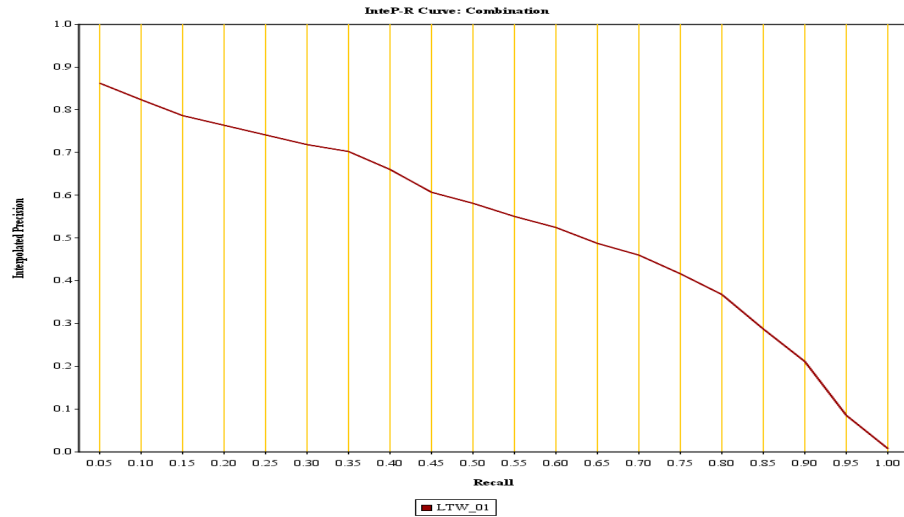


Fig. 4. Combined Interpolated Precision and Recall for Incoming and Outgoing Links

4 Conclusions and Future Work

We implemented a simple element retrieval technique and a more sophisticated passage retrieval technique to return result sets for ad hoc focused and best-in-context tasks. We showed that our implementation of focused task along with Okapi BM25 scoring scheme works well for both IEEE collection [2] and Wikipedia collection. We speculate that the reason the passage-based best entry point retrieval did not work well is because the best entry point should start with relevant passages. Another reason is that the most relevant passage tend not to be at the beginning of an article, whereas the best entry point tend to be [4]. Therefore, we think that our passage-based retrieval may improve by returning the first element in the highest scoring passage.

We implemented a baseline algorithm for embedding incoming and outgoing links for Link-the-Wiki track. We showed that our selection of outgoing links has a high accuracy, but a raw recall. However, our ranked outgoing links performs very well against the official metrics. Our result for incoming links show that the

	MAP	R-Prec	P5	P10	P20	P30	P50
Official Incoming	0.465	0.512	0.662	0.653	0.603	0.57	0.516
Official Combined w/ Unranked Outgoing	0.154	0.171	0.637	0.56	0.42	0.329	0.234
Unofficial Combined w/ Ranked Outgoing	0.527	0.564	0.744	0.725	0.669	0.627	0.561

Table 2. Incoming and Combined Results Using Official Metrics

simple algorithm generally perform well overall, but need to increase precision more at an early stage.

References

1. S. Büttcher. the Wumpus Search Engine. Accessible at <http://www.wumpus-search.org>, 2007.
2. C. L. A. Clarke. Controlling Overlap in Content-oriented XML retrieval. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 314–321, New York, NY, USA, 2005. ACM.
3. K. Y. Itakura and C. L. A. Clarke. From Passages into Elements in XML Retrieval. In *SIGIR 2007 Workshop on Focused Retrieval*, 2007.
4. J. Kamps, M. Koolen, and M. Lalmas. Where to start reading a textual xml document? In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 723–724, New York, NY, USA, 2007. ACM.
5. S. Robertson, S. Walker, and M. Beaulieu. Okapi at trec-7: Automatic ad hoc, filtering, vlc and interactive track. *7th Text REtrieval Conference*, 1998.

The University of Amsterdam at INEX 2007

Khairun Nisa Fachry¹, Jaap Kamps^{1,2}, Marijn Koolen¹, and Junte Zhang¹

¹ Archives and Information Studies, Faculty of Humanities, University of Amsterdam

² ISLA, Faculty of Science, University of Amsterdam

Abstract. In this paper, we document our efforts at INEX 2007 where we participated in the Ad Hoc Track, the Link the Wiki Track, and the Interactive Track that continued from INEX 2006. Our main aims at INEX 2007 were the following. For the Ad Hoc Track, we investigated the effectiveness of incorporating link evidence into the model, and of a CAS filtering method exploiting the structural hints in the INEX topics. For the Link the Wiki Track, we investigated the relative effectiveness of link detection based on the Wikipedia article's name only, and on the matching arbitrary text segments of different pages. For the Interactive Track, we took part in the interactive experiment comparing an element retrieval system with a passage retrieval system. The main results are the following. For the Ad Hoc Track, we see that link priors improve most of our runs for the Relevant in Context and Best in Context Tasks, and that CAS pool filtering is effective for the Relevant in Context and Best in Context Tasks. For the Link the Wiki Track, the results show that name matching works best, and can still be expanded and fine-tuned to achieve better performance. For the Interactive Track, our test-persons showed a weak preference for the element retrieval system over the passage retrieval system.

1 Introduction

In this paper, we describe our participation in the INEX 2007 Ad Hoc and Link the Wiki tracks, and the INEX 2006 Interactive Track. For the Ad Hoc track, our aims were: a) to investigate the effectiveness of incorporating link evidence into the model, to rerank retrieval results and b) to compare several CAS filtering methods that exploit the structural hints in the INEX topics. Link structure has been used effectively in Web retrieval [9] for known-item finding tasks. Although the number of incoming links is not effective for general ad hoc topics on Web collections [5], Wikipedia links are of a different nature than Web links, and might be more effective for informational topics.

For the Link the Wiki Track, we investigated the relative effectiveness of link detection based on the Wikipedia article's name only, and on the matching arbitrary text segments of different pages. Information Retrieval methods have been employed to automatically construct hypertext on the Web [2], as well for specifically discovering missing links in Wikipedia [4]. The track is aimed at detecting missing links between a set of topics, and the remainder of the collection, specifically detecting links between an origin node and a destination

Table 1. Relevant passage statistics

Description	Statistics	
	2006	2007
# topics	114	99
# articles with relevance	5,648	6,042
# relevant passages	9,083	10,818
mean length relevant passage	1,090	944
median length relevant passage	297	272

node. To detect whether two nodes are implicitly connected, it is necessary to search the Wikipedia pages for some text segments that both nodes share.

For the Interactive Track, we took part in the interactive experiment comparing an element retrieval system with a passage retrieval system. Trotman and Geva [16] argued that, since INEX relevance assessments are not bound to XML element boundaries, retrieval systems should also not be bound to XML element boundaries. Their implicit assumption is that a system returning passages is at least as effective and useful as a system returning XML elements. Since the document structure may have additional use beyond retrieval effectiveness, think for example of browsing through a result article using a table of contents, the INEX 2006 Interactive Track set up concerted experiment compare an element retrieval system to a passage retrieval system [11]. The element retrieval system returns element of varying granularity based on the hierarchical document structure and passage retrieval returns non-overlapping passages derived by splitting the document linearly. The INEX 2006 Interactive Track run well into INEX 2007, so we report our findings here.

The rest of the paper is organized as follows. First, Section 2 describes our retrieval approach. Then, in Section 3, we report the results for the Ad Hoc Track: the Focused Task in Section 3.1; the Relevant in Context Task in Section 3.2; and the Best in Context Task in Section 3.3. Followed by Section 4 detailing our approach and results for the INEX 2007 Link the Wiki Track. In Section 5 we discuss our INEX 2006 Interactive Track experiments. Finally, in Section 6, we discuss our findings and draw some conclusions.

2 Experimental Setup

2.1 Collection, Topics, and Relevance Judgments

The document collection is based on the English Wikipedia [17]. The collection has been converted from the wiki-syntax to an XML format [3]. The XML collection has more than 650,000 documents and over 50,000,000 elements using 1,241 different tag names. However, of these, 779 tags occur only once, and only 120 of them occur more than 10 times in the entire collection. On average, documents have almost 80 elements, with an average depth of 4.82.

There have been 130 topics selected for the INEX 2007 Ad Hoc track, which are numbered 414-543. Table 1 shows some statistics on this years assessments.

We have included the numbers from last years assessments for comparison. The number of relevant articles and passages is slightly higher than last year, while the number of assessed topics is lower. Last year, 114 topics were assessed, with 49.54 relevant articles and 79.68 relevant passages per topic. This year, 99 topics were assessed, with 60.48 relevant articles and 108.39 relevant passages per topic. The average number of relevant passages per relevant articles is 1.61 for the 2006 topics and 1.79 for the 2007 topics. On the other hand, the size of the relevant passages this year has decreased compared to last year. Both average (948) and median (272) size (in character length) are lower than last year (1,090 and 297 respectively).

2.2 Indexing

Our indexing approach is based on our earlier work [8, 13, 14, 15].

- *Element index*: Our main index contains all retrievable elements, where we index all textual content of the element including the textual content of their descendants. This results in the “traditional” overlapping element index in the same way as we have done in the previous years [14].
- *Contain index*: We built an index based on frequently retrieved elements. Studying the distribution of retrieved elements, we found that the <article>, <body>, <section>, <p>, <normallist>, <item>, <row> and <caption> elements are the most frequently retrieved elements. Other frequently retrieved elements are <collectionlink>, <outsidelink> and <unknownlink> elements. However, since these links contain only a few terms at most, and say more about the relevance of another page, we didn’t add them to the index.
- *Article index*: We also build an index containing all full-text articles (i.e., all wikipages) as is standard in IR.

For all indexes, stop-words were removed, but no morphological normalization such as stemming was applied. Queries are processed similar to the documents, we use either the CO query or the CAS query, and remove query operators (if present) from the CO query and the about-functions in the CAS query.

2.3 Retrieval Model

Our retrieval system is based on the Lucene engine with a number of home-grown extensions [7, 10].

For the Ad Hoc Track, we use a language model where the score for a element e given a query q is calculated as:

$$P(e|q) = P(e) \cdot P(q|e) \tag{1}$$

where $P(q|e)$ can be viewed as a query generation process—what is the chance that the query is derived from this element—and $P(e)$ an element prior that provides an elegant way to incorporate link evidence and other query independent evidence [6, 9].

We estimate $P(q|e)$ using Jelinek-Mercer smoothing against the whole collection, i.e., for a collection D , element e and query q :

$$P(q|e) = \prod_{t \in q} ((1 - \lambda) \cdot P(t|D) + \lambda \cdot P(t|e)), \quad (2)$$

where $P(t|e) = \frac{\text{freq}(t,e)}{|e|}$ and $P(t|D) = \frac{\text{freq}(t,D)}{\sum_{e' \in D} |e'|}$.

Finally, we assign a prior probability to an element e relative to its length in the following manner:

$$P(e) = \frac{|e|^\beta}{\sum_e |e|^\beta}, \quad (3)$$

where $|e|$ is the size of an element e . The β parameter introduces a length bias which is proportional to the element length with $\beta = 1$ (the default setting). For a more thorough description of our retrieval approach we refer to [15]. For comprehensive experiments on the earlier INEX data, see [12].

For our Link the Wiki Track runs, we use a vector-space retrieval model. Our vector space model is the default similarity measure in Lucene [10], i.e., for a collection D , document d and query q :

$$\text{sim}(q, d) = \sum_{t \in q} \frac{tf_{t,q} \cdot idf_t}{norm_q} \cdot \frac{tf_{t,d} \cdot idf_t}{norm_d} \cdot coord_{q,d} \cdot weight_t, \quad (4)$$

where $tf_{t,x} = \sqrt{\text{freq}(t, X)}$; $idf_t = 1 + \log \frac{|D|}{\text{freq}(t, D)}$; $norm_q = \sqrt{\sum_{t \in q} tf_{t,q} \cdot idf_t^2}$; $norm_d = \sqrt{|d|}$; and $coord_{q,d} = \frac{|q \cap d|}{|q|}$.

2.4 Link Evidence as Document Priors

One of our aims for the Ad Hoc Track this year was to investigate the effectiveness of using link evidence as an indicator of relevance. We have chosen to use the link evidence priors to rerank the retrieved elements, instead of incorporating it directly into the retrieval model.

In the official runs, we have only looked at the number of incoming links (indegree) per article. Incoming links can only be considered at the article level, hence we apply all the priors at the article level, i.e., all the retrieved elements from the same article are multiplied with the same prior score. We experimented with *global* indegree, i.e., the number of incoming links from the entire collection, and *local* indegree, i.e., the number of incoming links from within the subset of articles retrieved for one topic. Although we tried global and local indegree scores separately as priors, we limit our discussion to a weighted combination of the two degrees, as this gave the best results when we tested on the 2006 topics. We compute the link degree prior $P_{\text{LocGlob}}(d)$ for an article d as:

$$P_{\text{LocGlob}}(d) \propto 1 + \frac{\text{Local}_{\text{In}}(d)}{1 + \text{Global}_{\text{In}}(d)}$$

Since the local indegree of an article is at most equal to the global indegree (when all the articles pointing to it are in the subset of retrieved articles), $P_{\text{LocGlob}}(d)$ is a number between 1 and 2. This is a much more conservative prior than using the indegree, local or global, directly. We will, for convenience, refer to the link evidence as prior, even though we do not actually transform it into a probability distribution. Note that we can turn any prior into a probability distribution by multiplying it with a constant factor $\frac{1}{\sum_{d \in D} \text{prior}(d)}$, leading to the same ranking.

3 Ad Hoc Retrieval Results

This year, there was no official Thorough task. The remaining tasks were the same as last year: Focused, Relevant in Context and Best in Context. For the Focused Task, no overlapping elements may be returned. For the Relevant in Context Task, all retrieved elements must be grouped per article, and for the Best in Context Task only one element or article offset may be returned indicating the best point to start reading. However, since both our indexes contain overlapping elements, the initials runs might contain overlapping results.

To get CAS runs, we use a filter over the CO runs, using the pool of target elements of all topics. If a tag X is a target element for a given topic, we treat it as target element for all topics. We pool the target element tags of all topics, resulting in the following tags (by decreasing frequency): `<article>`, `<section>`, `<figure>`, `<p>`, `<image>`, `<title>`, and `<body>`. Then, we filter out all other elements from the results list of each topic. In other words, a retrieved element is only retained in the list if it is a target element for at least one of the topics.

We used the following runs Thorough runs as base runs for the various tasks.

- `inex07_contain_beta1_thorough_cl` a standard *contain* index run, with $\beta = 1$ and $\lambda = 0.15$.
- `inex07_contain_beta1_thorough_clp_10000_cl` like the previous run, but reranked over all 10,000 results using the conservative link prior.
- `inex07_contain_beta1_thorough_cl_pool_filter` a CAS version of the standard run, where only the pool of target elements are retained.
- `inex07_contain_beta1_thorough_clp_10000_cl_pool_filter` a CAS version of the conservatively reranked run.
- `inex07_element_beta1_thorough_clp_10000_cl` a standard *element* index run, reranked using the conservative link prior.
- `inex07_element_beta1_thorough_clp_10000_cl_pool_filter` the CAS version of the previous run.

3.1 Focused Task

To ensure the Focused run has no overlap, it is post-processed by a straightforward list-based removal strategy. We traverse the list top-down, and simply remove any element that is an ancestor or descendant of an element seen earlier in the list. For example, if the first result from an article is the article itself, we will not include any further element from this article.

Table 2. Results for the Ad Hoc Track Focused Task (runs in emphatic are no official submissions)

Run	iP[0.00]	iP[0.01]	iP[0.05]	iP[0.10]	MAiP
<i>element_beta1_focused</i>	0.4662	0.4126	0.3837	0.3621	0.2621
<i>element_beta1_focused_cas_pool_filter</i>	0.4409	0.4029	0.3676	0.3476	0.2544
<i>element_beta1_focused_clp_10000_cl</i>	0.4780	0.3938	0.3236	0.2974	0.1326
<i>element_beta1_focused_clp_10000_cl_cas_pool_filter</i>	0.4261	0.3723	0.3108	0.2771	0.1210
<i>contain_beta1_focused_cl</i>	0.4505	0.3837	0.3201	0.2959	0.1324
<i>contain_beta1_focused_cl_cas_pool_filter</i>	0.4230	0.3779	0.3181	0.2885	0.1302
<i>contain_beta1_focused_clp_10000_cl</i>	0.4493	0.3865	0.3224	0.2957	0.1352
<i>contain_beta1_focused_clp_10000_cl_cas_pool_filter</i>	0.4225	0.3787	0.3201	0.2872	0.1325

Table 2 shows the results for the Focused Task. The *element* run scores higher than the *contain* run on all measures, which might be explained by the many smaller elements in the *element* index. The <collectionlink> element is by far the most frequently retrieved element throughout the result list. Since these elements contain only a few words, they add little to recall, but all relevant <collectionlink> elements are completely relevant, thus leading to high precision scores.

The CAS filter has a negative effect on the scores, for both the *element* and *contain* runs. The pool of target elements is very small. The only elements that are mentioned as target elements in this years CAS topics are <article>, <body>, <section>, <p>, <figure>, <image> and <title>. Clearly, some relevant elements are removed by the filter. Also on the link prior runs, the CAS filter has a negative effect.

The link evidence helps in boosting relevant elements to the top ranks for the *element* run, leading to an improvement of early precision (iP[0.00]), but further down the list, precision drops rapidly. For the *contain* run, link evidence has a very small positive effect for iP[0.01], iP[0.05] and MAiP. The link prior has a clustering effect, pushing elements with a low retrieval score but with a high link indegree above elements with a higher retrieval score but a lower link indegree. The top ranked elements are often from articles with a lot of relevance, thus lower scoring elements from the same article have a high probability of containing relevance as well, leading to an improvement in early precision. But for articles with little relevance, this clustering effect might have a negative effect, since the high scoring elements of such articles contain most of the relevance and pushing up low scoring elements from those articles hurts precision.

3.2 Relevant in Context Task

For the Relevant in Context task, we use the Focused runs and cluster all elements belonging to the same article together, and order the article clusters by the highest scoring element. Table 3 shows the results for the Relevant in Context Task. Again, the standard *element* run scores better than the standard *contain* run. If we look at the different cut-offs, we see that the difference between the two runs becomes smaller. However, the *element* run also has a higher

Table 3. Results for the Ad Hoc Track Relevant in Context Task (runs in emphatic are no official submissions)

Run	gP[5]	gP[10]	gP[25]	gP[50]	MAgP
<i>element_beta1_ric_hse</i>	0.2009	0.1775	0.1282	0.0951	0.0905
<i>element_beta1_ric_hse_cas_pool_filter</i>	0.2227	0.1784	0.1366	0.1052	0.1003
<i>element_beta1_clp_10000_cl_ric_hse</i>	0.1808	0.1508	0.1104	0.0811	0.0831
<i>element_beta1_clp_10000_cl_cas_pool_filter_ric_hse</i>	0.1704	0.1373	0.1000	0.0766	0.0761
<i>contain_beta1_cl_ric_hse</i>	0.1696	0.1440	0.1036	0.0822	0.0805
<i>contain_beta1_cl_cas_pool_filter_ric_hse</i>	0.1665	0.1370	0.1059	0.0801	0.0805
<i>contain_beta1_clp_10000_cl_ric_hse</i>	0.1732	0.1487	0.1086	0.0831	0.0860
<i>contain_beta1_clp_10000_cl_cas_pool_filter_ric_hse</i>	0.1683	0.1459	0.1069	0.0820	0.0846

Table 4. Results for the Ad Hoc Track Best in Context Task (runs in emphatic are no official submissions)

Run	gP[5]	gP[10]	gP[25]	gP[50]	MAgP
<i>element_beta1_bic_hse</i>	0.2727	0.2623	0.2016	0.1601	0.1598
<i>element_beta1_cas_pool_filter_bic_hse</i>	0.3124	0.2749	0.2093	0.1647	0.1623
<i>element_beta1_clp_10000_cl_bic_hse</i>	0.3029	0.2690	0.2111	0.1645	0.1561
<i>element_beta1_clp_10000_cl_cas_pool_filter_bic_hse</i>	0.3192	0.2662	0.2026	0.1606	0.1456
<i>contain_beta1_cl_bic_hse</i>	0.2643	0.2552	0.1913	0.1537	0.1553
<i>contain_beta1_cl_cas_pool_filter_bic_hse</i>	0.3289	0.2807	0.2129	0.1647	0.1618
<i>contain_beta1_clp_10000_cl_bic_hse</i>	0.2816	0.2694	0.2123	0.1667	0.1684
<i>contain_beta1_clp_10000_cl_cas_pool_filter_bic_hse</i>	0.3311	0.2906	0.2266	0.1775	0.1736

MAgP score. This might be the effect of the length prior. Without the length prior, the *element* run would consist of many really small elements, which would give low recall. By adding a length prior, much larger elements, like `<article>`, `<body>` and `<section>` receive a higher score and give higher recall. However, some `<collectionlink>` elements still receive a high score, indicating that they contain many of the query terms, and can add to recall without losing precision.

For the CAS filter and link prior, we see the following. The CAS filter is effective for the standard *element* run, but not for the *contain* run. For the *element* run, the link prior has a negative effect, while on the *contain* run, it has a positive effect. The CAS filter is also not effective for the link prior runs.

3.3 Best in Context Task

The aim of the Best in Context task is to return a single result per article, which gives best access to the relevant elements. Table 4 shows the results for the Best in Context Task. Of the two base runs, the *element* run scores better on all measures. This is not surprising when looking at the results for the previously described tasks. The *element* scores consistently better in both the Focused and Relevant in Context tasks, although here the differences are smaller.

For the CAS filter and link prior, we see the following. The pool filter is especially effective for early precision. Where the link prior is effective for the first 50 ranks on both runs, it improves MAgP for the *contain* run, but hurts

MAGP for the *element* run. The combination of the pool filter and the link prior is less effective than the filter or link prior separately for the *element* run. For the *contain* run, the combination is more effective than the separate methods, and even outperforms the *element* runs.

4 Link the Wiki Track

In this section, we discuss our participation in the Link The Wiki (LTW) track. LTW is aimed at detecting missing links between a set of topics, and the remainder of the collection, specifically detecting links between an origin node and a destination node. Existing links in origin nodes were removed from the 90 topics, in this case whole Wikipedia articles, and the task was to detect these links again and find the correct destination node. This year we submitted five official runs to the LTW Track, and one post-submission run. We describe our approach, our results based on the official qrels, and an analysis of the errors.

4.1 Approach

Information Retrieval methods have been employed to automatically construct hypertext on the Web [1, 2], as well for specifically discovering missing links in Wikipedia [4]. To detect whether two nodes are implicitly connected, it is necessary to search the Wikipedia pages for some text segments that both nodes share. Usually it is only one specific and extract string [1]. Our approach is mostly based on this assumption, where we defined one text segment as a single line, and a string that both nodes share is a relevant substring. A substring of a string $T = t_1 \dots t_n$ is a string $\hat{T} = t_{i+1} \dots t_{m+i}$, where $0 \leq i$ and $m + i \leq n$. Only relevant substrings of at least 3 characters length are considered in our approach.

We adopt a *breadth m-depth n* technique for automatic text structuring for identifying candidate anchors and text node, i.e. a fixed number of documents accepted in response to a query and fixed number of iterative searches by looking at the similarity. This similarity can be evaluated in two dimensions: global similarity between an origin node and destination node where the whole document is used, and local similarity where only text segments are compared pairwise. The local similarity is used as a precision filter. To evaluate the global similarity between an orphan page and a target page, we used Lucene's Vector Space Model on an article index (see Section 2).

Global Similarity We focus on the global similarity by collecting a set of similar or related pages using the set of topics. We search in the collection by retrieving the top 100 similar documents by using the whole document as a query against the index of the Wikipedia collection without the topic files, but filtering with the English Snowball stopwords list for efficiency reasons. We also retrieved the top 100 similar documents for a topic by using top N terms derived from a language model as query.

Local Similarity We search on the local level with text segments. Normalized (lower case, removal of punctuation trailing spaces) lines are matched with string processing. At the same time we parse the XML and keep track of the absolute path for each text node and calculate the starting and end position of the identified anchor text. For all our official runs, we blindly select the first instance of a matching line, and continue with the next line so an anchor text can only have one link.

The INEX LTW Track focuses on structural links, which have an anchor and refers to the Best Entry Point of another page. Our Best Entry Points are paths to the closest located elements that contain substrings which match with the specified anchor text, thus the deepest node. Anchors are identified with the element path and the offset. The LTW task consists of identifying outgoing and incoming links between the 90 topics and existing Wikipedia pages. We have not focused on local links within the topics.

Incoming Links This type of link consists of a specified XPath expression (anchor) from destination nodes in the target pages to the Best Entry Point (origin node) of one of the related 90 topics. Incoming links are detected by top-down processing the relevant related pages, and for each page iteratively do (partial) line-matching with all lines of that file with the lines of the topic.

Outgoing Links A link from an anchor in the topic file to the Best Entry Point of existing related pages. We iterate over all lines of the topic file, and (partially) match the lines top-down with candidate target files until a link has been detected for that line.

In the current Wikipedia, links only point directly to entire articles, thus the beginning or name of the page. The run LTW01 is based on this observation. In this run, we extract for each topic the title enclosed with the <NAME> tag with a regular expression and match that title with (substrings of) lines in the target files to identify incoming links. To retrieve outgoing links, we extract the names of the 100 target pages and iteratively match those titles with each line (substring) of the topic file until a link has been detected or if none has been found in the file. For run LTW01 the 100 related target files are retrieved for each topic by using that full topic as query.

The runs LTW02, LTW03, and LTW04 are based on identifying the local similarity between text segments with exact line matching, effectively only accepting a local similarity of 100% to improve precision. The purpose of these runs was to test the effect of the global similarity between documents on link detection using the full topic as query by building a Vector Space Model and the top N most relevant terms derived from a language model. The top 100 target files was selected for each of the 90 topics. For run LTW03 we used the full topic (excluding Snowball stopwords) as query. The top 10 terms is selected as query for run LTW03 and the top 25 for run LTW04.

The run LTW07 was completely experimental, where we explored the use of the Longest Common Substring (*LCSS*) and WordNet as anchor text expansion. The *LCSS* between string *S* and string *T* is the longest substring that occurs

Table 5. Results Link The Wiki: Number of Links and Time

Run	\bar{x} Incoming	\bar{x} Outgoing	Time (s)
LTW01	86.1	43.8	169,225
LTW02	273.6	90.0	340,473
LTW03	243.1	83.9	154,732
LTW04	280.1	88.9	179,445
LTW07	312.6	176.9	55,216
<i>LTW03'</i>	231.6	94.0	106,449

both in S and T denoted by $LCSS(S, T)$. The lengths and starting positions of the longest common substrings of S and T can be found with the help of a generalised suffix tree. We have built such a tree for each pair of lines. The longest common suffix ($LCSuff$) is computed as

$$LCSuff(S_{1\dots i}, T_{1\dots j}) = \begin{cases} LCSuff(S_{1\dots i-1}, T_{1\dots j-1}) + 1 & \text{if } S[i] = T[j] \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

The longest common substrings of S and T must be the maximal of these longest common suffixes of possible prefixes.

$$LCSS(S, T) = \max_{1 \leq i \leq m, 1 \leq j \leq n} LCSuff(S_{1..i}, T_{1..j}) \quad (6)$$

We also expect that anchor texts do not always exactly match with the (sub)string destination node as links can be associative. To deal with this problem, we used a Perl module that looks up synonyms for a candidate anchor text in the lexical database WordNet, thus switching to a semantically equivalent substring that is to be matched with potential destination nodes. Stopwords were filtered to avoid these being matched as the longest common substring and thus as an anchor text.

4.2 Results

For the evaluation, only article-to-article links are considered in the scores. The threshold for the number of incoming and outgoing links was each set to 250 for each topic, however, for LTW02, LTW03, LTW04 and LTW07 that threshold was unintentionally set outside the line matching iteration of a target file. Table 5 shows the mean of incoming and outgoing links. The time needed to generate the runs was also recorded. For all runs there were more incoming links than outgoing links. LTW07 was generated with the least time, but also had most number of links.

We show the scores for the runs in Table 6: (a) incoming links, (b) outgoing links, and (c) a combined score. The run LTW01 performed best overall, and LTW07 performed poorly. There is little difference between LTW02, LTW03, and LTW04. We have one post-submission LTW03', which is the same as LTW03 but corrects the approach for incoming links set to reduce duplicated article-to-article links, and hence improves the result. However, the results show that restricting the partial line-matching to the names of Wikipedia pages performs best as expected.

Table 6. Results for the Link The Wiki Track

(a) Incoming links							
Run	MAP	R-Prec	P5	P10	P20	P30	P50
LTW01	0.2264	0.2583	0.7022	0.6622	0.5767	0.5051	0.3920
LTW02	0.1085	0.1648	0.6600	0.5167	0.3267	0.2411	0.1571
LTW03	0.1096	0.1437	0.6222	0.5133	0.3644	0.2770	0.1827
LTW04	0.0927	0.1418	0.6400	0.4889	0.3317	0.2441	0.1591
LTW07	0.0039	0.0196	0.2378	0.1667	0.0883	0.0596	0.0358
<i>LTW03'</i>	0.1282	0.1755	0.6867	0.5978	0.4667	0.3767	0.2591

(b) Outgoing Links							
Run	MAP	R-Prec	P5	P10	P20	P30	P50
LTW01	0.1377	0.1739	0.7844	0.6844	0.4844	0.3437	0.2073
LTW02	0.0803	0.1538	0.4667	0.4344	0.3517	0.2885	0.1958
LTW03	0.0733	0.1410	0.4778	0.4211	0.3472	0.2767	0.1789
LTW04	0.0806	0.1494	0.4978	0.4278	0.3517	0.2870	0.1882
LTW07	0.0671	0.1273	0.5000	0.4256	0.3206	0.2467	0.1500
<i>LTW03'</i>	0.0744	0.1467	0.4911	0.4122	0.3489	0.2867	0.1873

(c) Combined with F-Score							
Run	MAP	R-Prec	P5	P10	P20	P30	P50
LTW01	0.1712	0.2079	0.7411	0.6731	0.5265	0.4091	0.2712
LTW02	0.0924	0.1591	0.5467	0.4720	0.3387	0.2627	0.1743
LTW03	0.0878	0.1423	0.5405	0.4626	0.3556	0.2769	0.1808
LTW04	0.0862	0.1455	0.5600	0.4563	0.3414	0.2638	0.1724
LTW07	0.0075	0.0339	0.3223	0.2395	0.1385	0.0960	0.0578
<i>LTW03'</i>	0.0941	0.1598	0.5727	0.4879	0.3993	0.3256	0.2175

4.3 Link the Wiki Track Findings

Our incoming links performed poorly. This year’s evaluation is based on article-to-article links. We over-generated incoming links, while at the same time setting the threshold of incoming links at 250. Moreover, since we generated links as Best Entry Points into the target pages, we created too many duplicated article-to-article links, which hurt our performance. The exact line-matching (LTW02, LTW03, LTW04) does not perform well. The post-submission run improved the incoming links, but the results are still not satisfactory.

Our assumption that pages that link to each other are related or similar in content may not necessarily hold, thus reducing the pool of relevant pages that can be linked. The granularity of text segments as lines could work well, however, more context may be required to properly detect the local similarity between two nodes. LTW07 was technically most complicated, and performed worst. The reason was that the local similarity matching was not discriminative enough, a candidate link was too easily accepted, and thus both incoming and outgoing links were over-generated.

In summary, the results show that of our different approaches to detect links, name matching works best, and that this run should be expanded and fine-tuned to achieve better performance.

Table 7. Post-task questionnaire

- Q1 How would you rate this experience?
(1=frustrating, 3=neutral, 5=pleasing)
- Q2 How would you rate the amount of time available to do this task?
(1=much more needed, 3=just right, 5=a lot more than necessary)
- Q3 How certain are you that you completed the task correctly?
(For Q3 until Q6, 1=not at all, 3=somewhat, 5=extremely)
- Q4 How easy was it to do the task?
- Q5 How satisfied are you with the information you found?
- Q6 To what extent did you find the presentation format (interface) useful?

5 Interactive Experiments

In this section, we discuss out interactive experiments of the INEX 2006 Interactive Track (which has run well into INEX 2007). For details about the track and set-up we refer to [11]. For the interactive track, we conducted an experiment where we took part in the concerted effort of Task A, in which we compare element and passage retrieval systems. We reported the result of the track based on the users responses on their searching experience and comparative evaluation on the element and passage retrieval systems. The element and passage retrieval systems evaluated are developed in a java-based retrieval system built within the Daffodil framework by the track organizers. The element retrieval system returns element of varying granularity based on the hierarchical document structure and passage retrieval returns non-overlapping passages derived by splitting the document linearly.

We participated in task A with nine test persons in which seven of them completed the experiment. Two persons failed to continue the experiment due to systems down time. Each test person worked with four simulated tasks in the Wikipedia collection. Two tasks were based on the element retrieval and the other two tasks were based on the passage retrieval. The track organizer provided a multi-faceted set of 12 tasks in which the test person can choose from. The 12 tasks consist of three task types (decision making, fact finding and information gathering) which further slit into two structural kinds (hierarchical and parallel). The experiment was conducted in accordance with the track guideline.

5.1 Post Experiment Questionnaire

For each task, each test person filled in questionnaires before and after each tasks, and before and after the experiment, resulting in 70 completed questionnaires. Table 7 shows the post task questionnaire. Table 8 shows the responses for the post-task questionnaire. First, we look at the result for all tasks. We found that the test persons were positive regarding both systems. Next, we look at responses for the element and passage system, without considering the task types and structures. We found that the element system is rated higher in terms of the amount of time used (Q2), certainty of completing the task (Q3), easiness of task (Q4), and satisfaction (Q5). As for the experience rate (Q1) and the usefulness of presentation (Q6), the passage retrieval system is rated higher. The fact that

Table 8. Post-task responses on searching experience: mean scores and standard deviations (in brackets)

	Q1	Q2	Q3	Q4	Q5	Q6
All tasks	3.11 (1.45)	3.63 (1.28)	3.30 (1.32)	3.30 (0.99)	3.33 (1.21)	3.48 (0.70)
Element	2.93 (1.44)	3.64 (1.22)	3.43 (1.22)	3.36 (1.01)	3.36 (1.22)	3.43 (0.76)
Passage	3.31 (1.49)	3.62 (1.39)	3.15 (1.46)	3.23 (1.01)	3.31 (1.25)	3.54 (0.66)

Table 9. Post-experiment responses on ease of use and learn: mean scores and standard deviations (in brackets)

	Ease of learning	Ease of use
System 1: Element	4.29 (0.49)	4.14 (0.38)
System 2: Passage	3.86 (0.90)	3.86 (0.69)

element retrieval system is rated less pleasing than the passage retrieval while it is regarded as a more effective system (Q3, Q5) is rather surprising.

5.2 Post Experiment Questionnaire

After each completed task, the test persons filled in a post-experiment questionnaire. Table 9 shows the responses to questions on ease of using, and easy of learning. The answer categories used a 5-point scale with 1=not at all, 3=some-what, and 5=extremely. With respects to ease of learning and ease of use of the systems, we found out that the test persons gave higher scores to element system than to passage system.

We can see that there is a tendency to favor the element retrieval system. This also shown by the answers of the post experiment questionnaire where the users were more positive for the element retrieval system. Furthermore, we also asked the test persons opinion about what they like and dislike about the search systems. In both systems all of the test persons appreciated the table of content. The table of content was detailed enough and gave a good overview of the document. They also think that detailed information on the result list, links to other document, term and paragraph highlighting, and document back and forward functions helped them during searching tasks. Almost all of the test persons complain about the performance of the system. They also claim that the result list sometimes gave to many irrelevant documents. In comparison between the two systems, the element system seemed to give a more complete table of content compare to the passage system, resulting a better overview to see the relations between sections. Furthermore, the result list in the passage system seemed to give a poorer result in the result list and in some cases it missed the relevant document.

5.3 Interactive Track Findings

From the result of the experiment, we mainly focus on the comparison of element and passage retrieval systems. Although the users appreciated both systems positively, there is a tendency that the users prefer the element retrieval system

to the passage retrieval system. From the user tasks questionnaires we discovered that the element retrieval is considered more effective than the passage retrieval system. Furthermore, from the post experiment questionnaires we found that element retrieval system seems to provide a clearer overview of the document. However, it is too early to conclude that element retrieval is better than passage retrieval on this experiment. Because our finding is based on a small user test that only involved seven test persons. Furthermore, the system performance was slow and we think that this might influence our result. Over the whole experiment, perhaps the most striking result is that none of the users find any striking difference between element and passage system. Several users did not even notice the differences at all. In addition, table of content was found the most useful feature of the system. The table of content for both element and passage retrieval were rated positively by the users. They argue that the content of table gave them a good overview of the document. The least appreciated feature of the system was related terms. From the comment we found out that the related terms did not help the users because they are too long and often off-topics.

6 Discussion and Conclusions

In this paper, we documented our efforts at INEX 2007 where we participated in the Ad hoc Track, the Link the Wiki Track, and the Interactive Track that continued from INEX 2006.

For the Ad Hoc Track, we investigated the effectiveness of incorporating link evidence into the model, and of a CAS filtering method exploiting the structural hints in the INEX topics. We found that link priors improve most of our runs for the Relevant in Context and Best in Context Tasks, and that CAS pool filtering is effective for the Relevant in Context and Best in Context Tasks.

For the Link the Wiki Track, we investigated the relative effectiveness of link detection based on the Wikipedia article's name only, and on the matching arbitrary text segments of different pages. Our results show that name matching works best, and can still be expanded and fine-tuned to achieve better performance. It is too early to conclude that more sophisticated approaches are ineffective, since the current evaluation was restricted to article-to-article links.

For the Interactive Track, we took part in the interactive experiment comparing an element retrieval system with a passage retrieval system. Our test-persons showed a weak preference for the element retrieval system over the passage retrieval system. Of course, our small study does not warrant a general conclusion on the usefulness of passage-based approaches in XML retrieval. The technique may still be immature, or the system's response may be improved.

Acknowledgments Jaap Kamps was supported by the Netherlands Organization for Scientific Research (NWO, grants # 612.066.302, 612.066.513, 639.072.601, and 640.001.501), and by the E.U.'s 6th FP for RTD (project MultiMATCH contract IST-033104). Marijn Koolen was supported by NWO under grant # 640.001.501. Khairun Nisa Fachry and Junte Zhang were supported by NWO under grant # 639.072.601.

Bibliography

- [1] M. Agosti, F. Crestani, and M. Melucci. On the use of information retrieval techniques for the automatic construction of hypertext. *Information Processing and Management*, 33:133–144, 1997.
- [2] J. Allan. Building hypertext using information retrieval. *Information Processing and Management*, 33:145–159, 1997.
- [3] L. Denoyer and P. Gallinari. The Wikipedia XML Corpus. *SIGIR Forum*, 40: 64–69, 2006.
- [4] S. Fissaha Adafre and M. de Rijke. Discovering missing links in wikipedia. In *LinkKDD '05: Proceedings of the 3rd international workshop on Link discovery*, pages 90–97. ACM Press, New York NY, USA, 2005.
- [5] D. Hawking and N. Craswell. Very large scale retrieval and web search. In *TREC: Experiment and Evaluation in Information Retrieval*, chapter 9, pages 199–231. MIT Press, 2005.
- [6] D. Hiemstra. *Using Language Models for Information Retrieval*. PhD thesis, Center for Telematics and Information Technology, University of Twente, 2001.
- [7] ILPS. The ILPS extension of the Lucene search engine, 2007. <http://ilps.science.uva.nl/Resources/>.
- [8] J. Kamps, M. Koolen, and B. Sigurbjörnsson. Filtering and clustering XML retrieval results. In *Comparative Evaluation of XML Information Retrieval Systems: Fifth Workshop of the INitiative for the Evaluation of XML Retrieval (INEX 2006)*, volume 4518 of *Lecture Notes in Computer Science*, pages 121–136. Springer Verlag, Heidelberg, 2007.
- [9] W. Kraaij, T. Westerveld, and D. Hiemstra. The importance of prior probabilities for entry page search. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 27–34. ACM Press, New York NY, USA, 2002.
- [10] Lucene. The Lucene search engine, 2007. <http://lucene.apache.org/>.
- [11] S. Malik, A. Tombros, and B. Larsen. The interactive track at INEX 2006. In *Comparative Evaluation of XML Information Retrieval Systems: Fifth Workshop of the INitiative for the Evaluation of XML Retrieval (INEX 2006)*, volume 4518 of *Lecture Notes in Computer Science*, pages 387–399. Springer Verlag, Heidelberg, 2007.
- [12] B. Sigurbjörnsson. *Focused Information Access using XML Element Retrieval*. SIKS dissertation series 2006-28, University of Amsterdam, 2006.
- [13] B. Sigurbjörnsson and J. Kamps. The effect of structured queries and selective indexing on XML retrieval. In *Advances in XML Information Retrieval and Evaluation: INEX 2005*, volume 3977 of *LNCS*, pages 104–118, 2006.
- [14] B. Sigurbjörnsson, J. Kamps, and M. de Rijke. An Element-Based Approach to XML Retrieval. In *INEX 2003 Workshop Proceedings*, pages 19–26, 2004.
- [15] B. Sigurbjörnsson, J. Kamps, and M. de Rijke. Mixture models, overlap, and structural hints in XML element retrieval. In *Advances in XML Information Retrieval: INEX 2004*, volume 3493 of *LNCS 3493*, pages 196–210, 2005.
- [16] A. Trotman and S. Geva. Passage retrieval and other XML-retrieval tasks. In *Proceedings of the SIGIR 2006 Workshop on XML Element Retrieval Methodology*, pages 43–50. University of Otago, Dunedin New Zealand, 2006.
- [17] Wikipedia. The free encyclopedia, 2006. <http://en.wikipedia.org/>.

GPX@INEX2007: Ad-hoc Queries and Automated Link Discovery in the Wikipedia

Shlomo Geva
Faculty of IT
Queensland University of Technology
Brisbane, Australia
s.geva@qut.edu.au

***Abstract** The INEX 2007 evaluation was based on the Wikipedia collection in XML format. In this paper we describe some modifications to the GPX search engine and the approach taken in the Ad-hoc and the Link-the-Wiki tracks. The GPX retrieval strategy is based on the construction of a collection sub-tree, consisting of all nodes that contain one or more of the search terms. Nodes containing search terms are assigned a score using the GPX ranking scheme which incorporates an extended TF-IDF variant. In earlier version of GPX scores were recursively propagated from text containing nodes, through ancestors, all the way to the document root of the XML tree. In this paper we describe a simplification whereby the score of each node is computed directly, doing away with the score propagation mechanism. Preliminary results indicate improved performance. The GPX search engine was used in the Link-the-Wiki track to identify prospective incoming links to new Wikipedia pages. We also describe a simple and efficient approach to the identification of prospective outgoing links in new Wikipedia pages. We present preliminary evaluation results.*

1. The GPX Search Engine

For the sake of completeness we provide a very brief description of GPX. The reader is referred to earlier papers on GPX in INEX previous proceedings for a more complete description. The search engine is based on XPath inverted lists. For each term in the collection we maintain an inverted list of XPath specifications. This includes the file name, the absolute XPath identifying a specific XML element, and the term position within the element. The actual data structure is designed for efficient storage and retrieval of the inverted list which are considerably less concise by comparison with basic text retrieval inverted lists. We briefly describe the data structure, then we describe the node scoring calculation, and finally we present the results.

2. GPX Inverted List Representation

The GPX search engine is using a relational database implementation (Apache Derby) to implement an inverted list data structure. It is a compromise solution provides the convenience of a DBMS at the cost of somewhat reduced performance which may otherwise be possible.

Consider the XPath:

`/article[1]/bdy[1]/sec[5]/p[3]`

This could be represented by two expressions, a Tag-set and an Index-set:

Tag-set: **article/bdy/sec/p**

Index-Set: **1/1/5/3**

The original XPath can be reconstructed from the tag-set and the index-set. It turns out that there are over 48,000 unique tag-sets, and about 500,000 unique index-sets in the collection. We assign to each tag set and each index-set a hash code and create auxiliary database tables mapping the hash-codes to the corresponding tag-set and index-set entries. These hash tables are small enough to be held in memory and so decoding is efficient.

The GPX database tables are then:

```
Term-Context = { Term-ID, File-ID, XPath-Tag-ID, XPath-IDX-ID, Position }
Terms =       { Term, Term-ID }
Files =       { File-Name, File-ID }
TagSet =      { XPath-Tag-ID, Tag-Set }
IndexSet =    { XPath-IDX-ID, Index-Set }
XPathSize =   { XPath-ID, Node-Size }
```

Given a search term the database can be efficiently accessed to obtain an inverted list containing the context of all instances where the term is used (identified by File Name, full XPath, and term position). Having retrieved a set of inverted lists, one for each term in the query, the lists are merged so as to keep count of query terms in each node and also keeping the term positions. Stop words are actually indexed, but too frequent terms are ignored by applying a run-time stop-word frequency threshold of 300,000. We also used plural/singular expansion of query terms. We have found that - on average - the use of a Porter stemmer is not adding to system performance and so it was not used.

Having collected all the nodes that contain at least one query term the system proceeds to compute node scores. Calculation of node relevance score from its content is based on a variation of TF-IDF. We used the inverse collection frequency of terms rather than the inverse document frequency (TF-ICF). The score is then moderated by a step function of the number of unique terms contained within the node. The more unique terms the higher the score. The score is further moderated by the proximity within which the terms are found. Additionally, the scores of all article nodes that contained query terms in the <name> node were further increased. All this can be calculated with the information in the inverted lists.

3. Calculation of Text Nodes Score

GPX 2007 deviates significantly from earlier with respect to the way that ancestor node scores are calculated. For clarity we shall refer to GPX-2007 to denote the current system and GPX to denote the older system. In the earlier version GPX computed node scores on the basis of direct text content (having a text node in the DOM model) and then the scores were propagated upwards in the XML tree. GPX accumulated all children node scores for a parent and reduced the score by a decay factor (typically about 0.7) to account for reduced specificity as one moved upwards in the XML tree. In GPX 2007 the scores are computed directly from the node text content, direct, or indirect. That means that any node is scored by the text it contains regardless of whether it has a direct text node in the DOM representation - all the text in the node and its descendents is used.

Naturally, nodes closer to the root could receive a higher score on account of more query terms in descendent nodes. A common variation to TF-IDF is to normalise the score by taking into account the document size. The motivation there is to account for the increased probability of finding query terms in larger documents and hence biasing the selection towards larger documents. The motivation here is similar with a slight twist. Node normalisation in the XML score calculation is motivated by the need to compensate for the reduced specificity of larger nodes. We are aiming for focused retrieval and look for nodes of "just the right size" (whatever that may be.) Node normalisation introduces a penalty in a parent node that contains large amounts of irrelevant text in descendent nodes and which do not contribute towards an increased score. However, when two nodes have a similar size but contain different amount of relevant text then the more relevant node will score higher.

But there is another twist here. We also know that nodes that are too small are unlikely to satisfy a user information need (except perhaps in factoid type QA). At least with the Wikipedia we know that the most common element selected by assessors is a paragraph (or passage). Very small passages are not common in the qrels of past experiments. Therefore, we do not want to normalise the scores of too small nodes thereby unduly increasing their score relative to otherwise similarly scoring nodes which are somewhat larger. Node scores are normalised by dividing the raw score by the node size (measured as the number terms), but all nodes with size of below 75 terms are normalised by 75. This heuristic is convenient in the XML case because when breaking ties in node selection (focused

retrieval) we prefer the ancestor to the descendant when the scores are equal. This means that we prefer parent nodes as long as the parent is larger than the descendant and below 75 terms in size. For example, this means that a very deep XML branch with no breadth will be collapsed to an ancestor of up to size 75 terms (if such exists). So in summary, node size normalisation is biasing the selection towards passages of 75 terms, both from above and from below. We experimented with other values for node size from 50 to 150 with little difference in results. More careful sensitivity analysis is still pending.

Since GPX 2007 we now computes node scores over much larger text segments it is necessary to take account of term proximity. The intuition is that we should award higher scores to nodes in which search terms are found in closer proximity to each other. In earlier versions of GPX this was not critical since node scores were computed at text nodes and these were typically paragraphs, titles, captions, and other such relatively small nodes. A proximity function was defined and incorporated into the score calculation. So finally we have the following score calculation:

Equation 1: Calculation of S, node size for normalisation

$$S = \begin{cases} \text{NodeSize} & > 75 \\ 75 & \leq 75 \end{cases}$$

The value of S, the node size for the purpose of normalization, is thus equal to 75 for nodes smaller than 75 terms, but taken as the actual node size for nodes with more terms.

(1)

Equation 2: Calculation of P, node terms proximity score

$$\text{Pr} = 10 \sum_{i=1}^n \exp\left(-\left(\frac{p_i - p_{i+1} + 1}{5}\right)^2\right) \quad (2)$$

Here terms are processed in the order in which they appear in the text node. P_i is the position of term i in the text node. This is a Gaussian function with a maximum value of 10 and decaying exponentially with increased term distance between successive terms. The function is depicted in Figure 1. Note that in practice, a table lookup is more efficient than the numerical calculation.

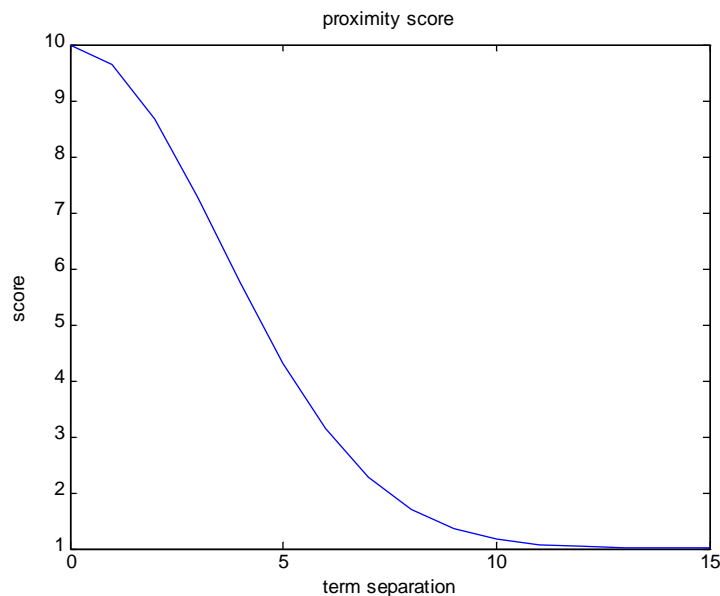


Figure 1. Proximity score as a function of term separation

Equation 3: Calculation of element relevance score from its content

$$L = \frac{\mathbf{Pr}}{S} K^{n-1} \sum_{i=1}^n \frac{t_i}{f_i} \quad (3)$$

Here n is the count of unique query terms contained within the element, and K is a small integer (we used $K=5$). The term K^{n-1} is a step function which scales up the score of elements having multiple distinct query terms. This heuristic of rewarding the appearance of multiple distinct terms can conversely be viewed as taking more strongly into account the absence of query terms in a document. Here it is done by rewarding elements that do contain more distinct query terms. The system is not sensitive to the value of K and a value of $k=5$ is adequate. The summation is performed over all n terms that are found within the element where t_i is the frequency of the i^{th} query term in the element and f_i is the frequency of the i^{th} query term in the collection.

Finally, nodes that contain query terms that are preceded by a minus sign (undesirable) are eliminated.

At this point we have computed the score of all (overlapping) nodes in each article that contains query terms. The score of the <article> node itself is then added to all nodes in the article. This lifts the scores of all nodes that appear in a high scoring article. The intuition is that an article with many scoring nodes is more likely to be relevant and so all its scoring elements are ranked higher on account of more scoring nodes appearing in the same article. Without this modification, two similar nodes, one being an isolated instance of a relevant node in an article, and the other being one of many relevant nodes in an article, would receive a similar score.

Although more analysis of the results is required, preliminary results suggest an improved performance in GPX. The runs labelled RIC_04 and BIC_04 were produced with the 2006 GPX version (score propagation) while BIC_07 and RIC_07 were run with the GPX_07 version with direct score calculation. The GPX_07 version seems to perform better than the earlier GPX version over almost all reported measures. It does not require any magic numbers (decay constants) and is therefore more appealing.

4. Ad-Hoc retrieval tasks

The Ad-Hoc track at INEX 2007 consisted of 3 tasks – Focused, Relevant in Context, and Best in Context. These tasks are described elsewhere in this proceedings collection. We briefly describe the approach taken to each of the tasks in our best performing run.

4.1 Focused Retrieval

Focused Retrieval starts with the thorough results recall base. Within each article the highest scoring elements on a path are selected by keeping only elements that have a higher score than any of their descendants or ancestors. The submission consists of the remaining overlap free focused elements, sorted by descending score.

4.2 Relevant in Context (RIC)

The objective of the task was to balance article retrieval and element retrieval. Whole articles are first ranked in descending order of relevance and within each article a set of non-overlapping most focused elements are grouped. We have used the focused results, which were overlap free already, but grouped the elements within articles and sorted the articles by article score.

4.3 Best in Context (BIC)

We tested a trivial approach here – we simply kept the highest scoring element in each document appearing in the focused recall base.

5. Link the Wiki

The Link the Wiki task is described in detail elsewhere in this proceedings collection. The objective of this task was to identify a set of incoming links and a set of outgoing links for new Wikipedia pages. In practice, the topics were existing Wikipedia pages that were stripped of existing links. The links were only at the article-to-article level. We adopted rather simple approaches.

5.1 Incoming links

Incoming links were identified by using the GPX search engine to search for elements that were about the topic name element. For each topic the *name* element was used to construct a standard NEXI query:

```
//article[about(.,name)]
```

We have used the SCAS task setting whereby the results were interpreted strictly. In this case it only means that articles nodes were returned. This was sufficient since only article-to-article links were needed. Results were ordered by article score with the more likely relevant articles returned earlier in the list. The process took an average of 8.5 seconds per topic.

5.2 Outgoing links

We have adopted a very simple approach to this task. All existing page names in the Wikipedia were loaded into an in-memory hash table (with collision resolution). With 660,000 articles this is not an onerous task. The identification of potential links was based on a systematic search for anchor text that matches existing page names. In the first stage we have extracted the text of the topic (eliminating all markup information.) Prospective anchors for outgoing links were identified by running a window over the topic text and looking for matching page names in the collection. The window size varied from 8 words down to 1 word, and included stop words. Longer anchors were ranked higher than shorter ones, motivated by the trivial observation that the system was less likely to hit on a longer page name by accident. A naïve approach perhaps, but quite effective as it turns out. The process is purely computational and does not incur any I/O operations. The process took an average of 0.6 seconds per topic.

5. Results

The GPX system performed well and produced particularly good results in the Relevant in Context and Best in Context tasks of the Ad-hoc track, and in the Link-the-Wiki track.

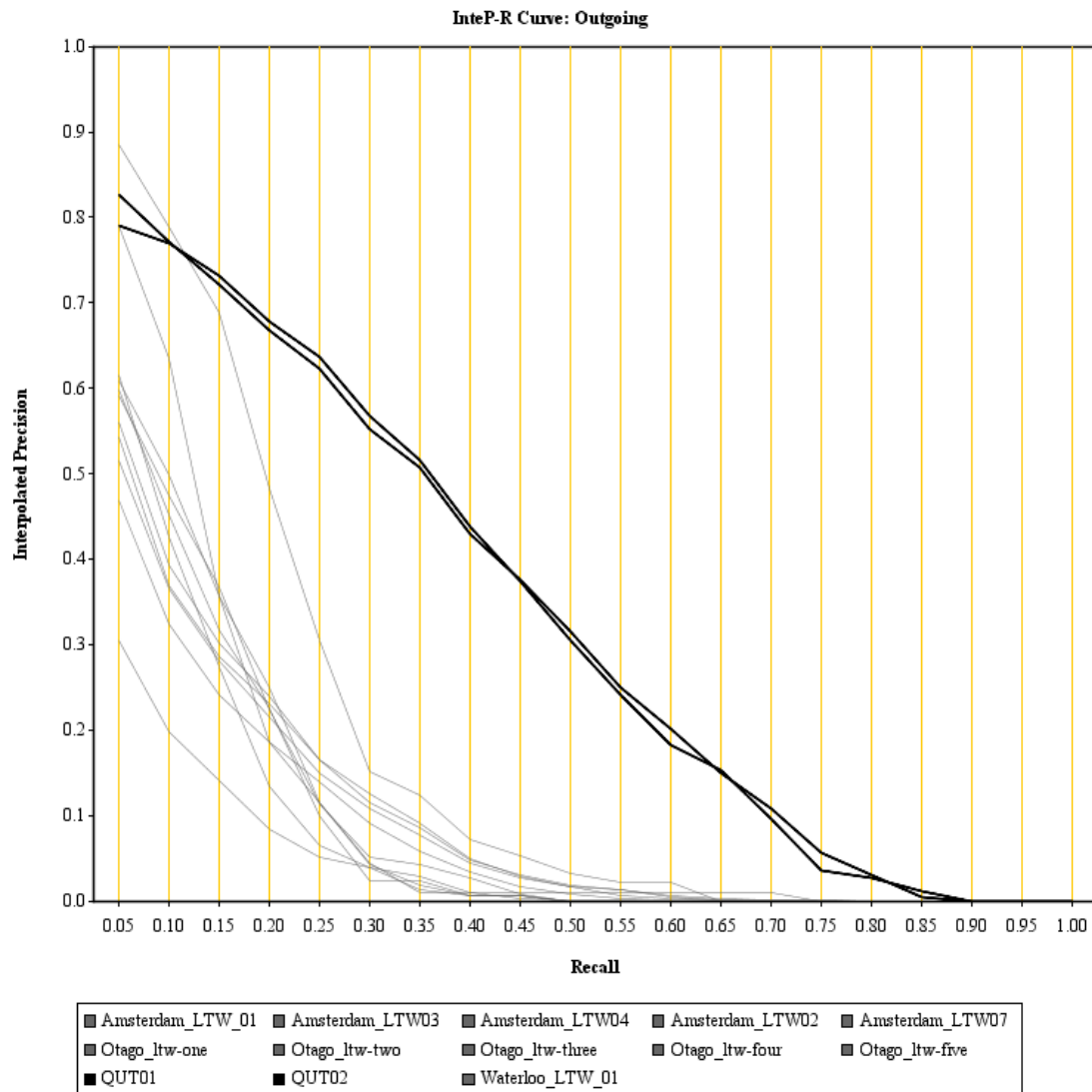
5.1 Ad Hoc retrieval

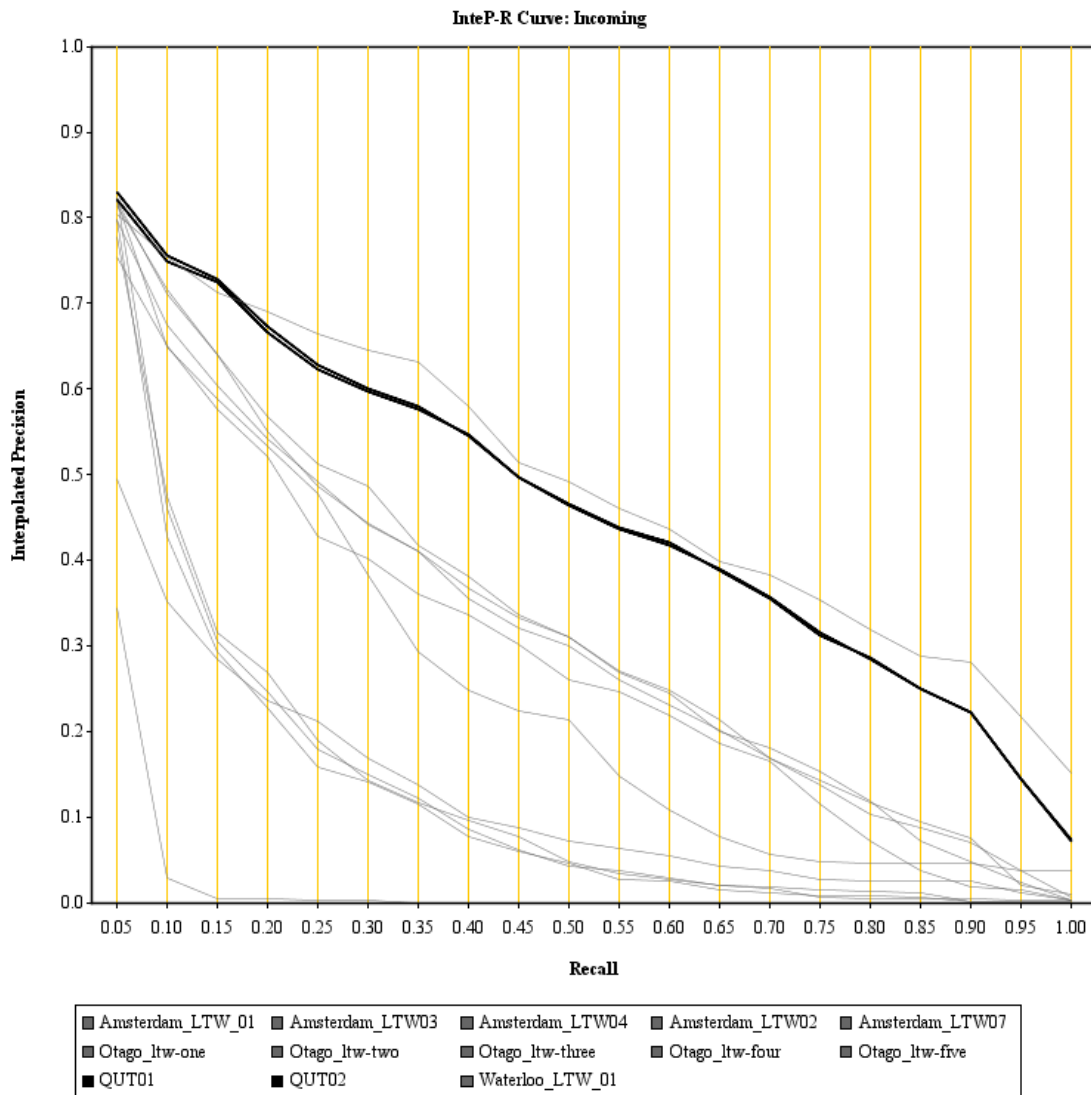
Task	Best MAgP	Best GPX MagP	Run rank	System rank
Relevant in Context	0.1013	0.0975	6/66	2/17
Best in Context	0.1951	0.1823	4/71	3/19
Focused	0.4259	0.3842	13/79	7/25

Relatively good results were achieved in terms of precision at early recall levels on most of the tasks. Complete results sets are available on the INEX web site:

<http://inex.is.informatik.uni-duisburg.de/2007/adhoc-protected/Evaluation.html>

5.2 Link-the-Wiki





In the Link-the-Wiki task apparently good results were achieved but the significance of this had not yet been established given the nature of the evaluation (no manual assessment was involved). There is also no baseline for comparison since this is the first time that the task was run.

Complete results sets are available on the INEX web site:

<http://inex.is.informatik.uni-duisburg.de/2007/lw-protected/results.html>

and also in the paper describing the Link the Wiki task in these proceedings.

6. References

1. S. Geva, GPX - Gardens Point XML IR at INEX 2006. In: Comparative Evaluation of XML information Retrieval Systems 5th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2006, Dagstuhl Castle, Germany, December 17-20, 2006, Springer, Lecture Notes in Computer Science LNCS, ISBN 978-3-540-73887-9, pp 137-150, 2007
2. S. Robertson, "Understanding Inverse Document Frequency: On theoretical arguments for IDF", Journal of Documentation 60 no. 5, pp 503-520, 2004.

Report on the INEX 2007 Multimedia Track

Theodora Tsirikika¹ and Thijs Westerveld^{2*}

¹ CWI, Amsterdam, The Netherlands

² Teezir Search Solutions, Ede, The Netherlands

Abstract. The INEX Multimedia track focuses on using the structure of XML documents to extract, relate, and combine the relevance of different multimedia fragments. This paper presents a brief overview of the track for INEX 2007, including the track’s test collection, tasks, and goals. We also report the approaches of the participating groups and their main results.

1 Introduction

Structured document retrieval from XML documents allows for the retrieval of XML document fragments, i.e., XML elements or passages, that contain relevant information. The main INEX Ad Hoc task focuses on text-based XML retrieval. Although text is dominantly present in most XML document collections, other types of media can also be found in those collections. Existing research on multimedia information retrieval has already shown that it is far from trivial to determine the combined relevance of a document that contains several multimedia objects.

The objective of the INEX Multimedia track is to exploit the XML structure that provides a logical level at which multimedia objects are connected, in order to improve the retrieval performance of an XML-driven multimedia information retrieval system. To this end, it provides an evaluation platform for the retrieval of multimedia documents and document fragments. In addition, it creates a discussion forum where the participating groups can exchange their ideas on different aspects of the multimedia XML retrieval task.

This paper reports on the INEX 2007 Multimedia track and is organised as follows. First, we introduce the main parts of the test collection: documents, tasks, topics, and assessments (Sections 2–5). Section 6 presents the approaches employed by the different participants and Section 7 summarises their main results. Section 8 concludes the paper and provides an outlook on next year’s track.

2 Wikipedia collections and additional resources

In INEX 2007, the Multimedia track employed the following two Wikipedia-based collections (the same as in 2006):

* Part of this work was carried out when the author was at CWI, Amsterdam, The Netherlands

Wikipedia XML collection: This is a structured collection of 659,388 Wiki-text pages from the English part of Wikipedia, the free content encyclopedia (<http://en.wikipedia.org>), that have been converted to XML [2]. This collection has been created for the Ad Hoc track. Given, though, its multimedia nature (as indicated by its statistics listed in Table 1), it is also being used as the target collection for a multimedia task that aims at finding relevant XML fragments given a multimedia information need (see Section 3).

Table 1. Wikipedia XML collection statistics

Total number of XML documents	659,388
Total number of images	344,642
Number of unique images	246,730
Average number of images per document	0.52
Average depth of XML structure	6.72
Average number of XML nodes per document	161.35

Wikipedia image XML collection: This is a collection consisting of the images in the Wikipedia XML collection, together with their metadata. These metadata, usually containing a brief caption or description of the image, the Wikipedia user who uploaded the image, and the copyright information, have been formatted in XML. Figure 1 shows an example of such a document consisting of an image and its associated metadata. Some images from the Wikipedia XML collection have been removed due to copyright issues or parsing problems with their metadata, leaving us with a collection of 171,900 images with metadata. This collection is used as the target collection for a multimedia/image retrieval task that aims at finding images (with metadata) given a multimedia information need (see Section 3).

Although the above two Wikipedia-based collections are the main search collections, additional sources of information are also provided to help participants in the retrieval tasks. These resources are:

Image classification scores: For each image, the classification scores for the 101 different MediaMill concepts are provided by UvA [5]. The UvA classifier is trained on manually annotated TRECVID video data and the concepts are selected for the broadcast news domain.

Image features: For each image, the set of the 120D feature vectors that has been used to derive the above image classification scores is available [3]. Participants can use these feature vectors to custom-build a CBIR system, without having to pre-process the image collection.

These resources were also provided in 2006, together with an online CBIR system that is no longer available. The above resources are beneficial to researchers who wish to exploit visual evidence without performing image analysis.



Fig. 1. Example Wikipedia image+metadata document from the Wikipedia image XML collection.

3 Retrieval Tasks

The aim of the retrieval tasks in the Multimedia track is to retrieve relevant (multimedia) information, based on an information need with a (structured) multimedia character. To this end, a structured document retrieval approach should be able to combine the relevance of different media types into a single ranking that is presented to the user.

For INEX 2007, we define the same two tasks as last year:

MMfragments task: Find relevant XML fragments in the **Wikipedia XML collection** given a multimedia information need. These XML fragments can correspond not only to XML elements (as it was in INEX 2006), but also to passages. This is similar to the direction taken by the INEX Ad Hoc track. In addition, since MMfragments is in essence comparable to the ad hoc retrieval of XML fragments, this year it ran along the Ad Hoc tasks. As a result, the three subtasks of the Ad Hoc track (see [1] for detailed descriptions) are also defined as subtasks of the MMfragments task:

1. **FOCUSED TASK** asks systems to return a ranked list of elements or passages to the user.
2. **RELEVANT IN CONTEXT TASK** asks systems to return relevant elements or passages clustered per article to the user.
3. **BEST IN CONTEXT TASK** asks systems to return articles with one best entry point to the user.

The difference is that MMfragments topics ask for multimedia fragments (i.e., fragments containing at least one image) and may also contain visual hints (see Section 4).

MMimages task: Find relevant images in the **Wikipedia image XML collection** given a multimedia information need. Given an information need, a retrieval system should return a ranked list of documents(=image+metadata) from this collection. Here, the type of the target element is defined, so basically this is closer to an image retrieval (or a document retrieval) task, rather than XML element or passage retrieval. Still, the structure of (supporting) documents, together with the visual content and context of the images, could be exploited to get to the relevant images (+their metadata).

All track resources (see Section 2) can be used for both tasks, but the track encourages participating groups to also submit a baseline run that uses no sources of information except for the target collection. This way, we hope to learn how the various sources of information contribute to the retrieval results. Furthermore, we also encourage each group to submit a run that is based on only the `<mmtitle>` field of the topic description (see Section 4). All other submissions may use any combination of the `<title>`, `<castitle>`, `<mmtitle>` and `<description>` fields (see Section 4). The fields used need to be reported.

4 Topics

The topics used in the INEX Multimedia track are descriptions of (structured) multimedia information needs that may contain not only textual, but also structural and multimedia hints. The structural hints specify the desirable elements to return to the user and where to look for relevant information, whereas the multimedia hints allow the user to indicate that results should have images similar to a given example image or be of a given concept. These hints are expressed in the NEXI query language [7].

The original NEXI specification determines how structural hints can be expressed, but does not make any provision for the expression of multimedia hints. These have been introduced as NEXI extensions during the INEX 2005 and 2006 Multimedia tracks [8, 9]:

- To indicate that results should have images similar to a given example image, an *about* clause with the keyword *src:* is used. For example, to find images of cityscapes similar to the image at <http://www.bushland.de/hksky2.jpg>, one could type:

```
//image[about(.,cityscape) and  
about(.,src:http://www.bushland.de/hksky2.jpg)]
```

In 2006, only example images from within the Wikipedia image XML collection were allowed, but this year it was required that the example images came from outside the Wikipedia collections.

- To indicate that the results should be of a given concept, an *about* clause with the keyword *concept*: is used. For example, to search for cityscapes, one could decide to use the concept “building”:

```
//image[about(.,cityscape) and about(.,concept:building)]
```

This feature is directly related to the concept classifications that are provided as an additional source of information (see Section 2). Therefore, terms following the keyword *concept*: are obviously restricted to the 101 concepts for which classification results are provided.

It is important to realise that all structural, textual and visual filters in the query should be interpreted loosely. It is up to the retrieval systems to decide how to use, combine or even ignore this information. The relevance of a document, element or passage does not directly depend on these hints, but is determined by manual assessments.

4.1 Topic format

The INEX Multimedia track topics are similar to the Content Only + Structure (CO+S) topics of the INEX Ad Hoc track. In INEX, “Content” refers to the textual or semantic content of a document part, and “Content-Only” to topics or queries that use no structural hints. The Ad Hoc CO+S topics include structural hints, whereas the Multimedia CO+S topics may also include visual hints.

The 2007 Multimedia CO+S topics consist of the following parts:

- <title>** The topic **<title>** simulates a user who does not know (or does not want to use) the actual structure of the XML documents in a query and who does not have (or want to use) example images or other visual hints. The query expressed in the topic **<title>** is, therefore, a Content Only (CO) query. This profile is likely to fit most users searching XML digital libraries and also corresponds to the standard web search type of keyword search.
- <castitle>** A NEXI expression with structural hints.
- <mmtitle>** A NEXI expression with structural and visual hints.
- <description>** A brief, matter of fact, description of the information need. Like a natural language description one might give to a librarian.
- <narrative>** A clear and precise description of the information need. The narrative unambiguously determines whether or not a given document or document part fulfils the given need. It is the only true and accurate interpretation of a user’s needs. Precise recording of the narrative is important for scientific repeatability - there must exist, somewhere, a definitive description of what is and is not relevant to the user. To aid this, the **<narrative>** should explain not only what information is being sought, but also the context and motivation of the information need, i.e., why the information is being sought and what work-task it might help to solve.

In previous years, both structural and visual/multimedia hints were expressed in the **<castitle>** field. This year, the **<castitle>** contains only structural hints, while the **<mmtitle>** is an extension of the **<castitle>** that also

incorporates the additional visual hints (if any). The introduction of a separate `<mmtitle>` is particularly useful, since it makes it easier for systems to compare runs using structural hints to those using structural+visual hints, without having to modify the query expression. In addition, Multimedia CO+S topics can now also be used in Ad Hoc tasks, since they contain fields (all, except `<mmtitle>`) that can be directly processed by an Ad Hoc system.

The fact that the MMfragments task is similar to ad hoc retrieval, not only led to the decision to run the MMfragments tasks along the Ad Hoc ones, but also to include the MMfragments topics as a subset of the Ad Hoc ones. This means that submissions for the INEX 2007 Ad Hoc track also considered the subset of topics used for the MMfragments task. This allows us to compare ad hoc XML retrieval systems submissions on the MMfragments topic subset (i.e., submissions that retrieve XML document parts by using any of the available fields except `<mmtitle>`) to multimedia XML retrieval submissions on the same topic subset (i.e., to submissions that can use any of the topic fields, together with the knowledge that a multimedia XML fragment is required as a retrieval result).

MMimages, on the other hand, runs as a separate task with a separate set of topics. Given that MMimages requires retrieval at the document level, rather than elements or passages, the queries in the `<castitle>` and `<mmtitle>` fields are restricted to: `//article[X]`, where X is a predicate using one or more *about* functions with textual and/or multimedia hints.

4.2 Topic development

The topics in the Multimedia track are developed by the participants. Each participating group has to create 2 multimedia topics for the MMfragments task and 4 topics for MMimages. Topic creators first create a 1-2 sentence description of the information they are seeking. Then, in an exploration phase, they obtain an estimate of the amount of relevant information in the collection. For this, they can use any retrieval system, including their own system or the TopX system [6] provided through the INEX organisation. The topic creator then assesses the top 25 results and abandons the search if fewer than two or more than 20 relevant fragments are found. If between 2 and 20 fragments are found to be relevant, the topic creator should have a good idea of what query terms should be used, and the `<title>` is formulated. Using this title a new search is performed and the top 100 elements are assessed. Having judged these 100 documents, topic creators should have a clear idea of what makes a fragment relevant or not. Based on that, they could then first write the narrative and then the other parts of the topic. After each created topic, participants are asked to fill a questionnaire that gathers information about the users familiarity with the topic, the expected number of relevant fragments in the collection, the expected size of relevant fragments and the realism of the topic. The submitted topics are analysed by the INEX Multimedia organisers who check for duplicates and inconsistencies before distributing the full set of topics among the participants.

Table 2 shows the distribution over tasks as well as some statistics on the topics. The MMfragments topics correspond to Ad Hoc topics 525-543. Their average number of terms in `<title>` (3.21) is slightly lower than the average number of terms in the remaining 80 Ad Hoc topics (3.92). This is to be expected, since users who submit multimedia topics express their requirements not only by textual, but also by visual hints. Table 2 indicates that not all topics contain visual/multimedia hints; this corresponds well with realistic scenarios, since users who express multimedia information needs do not necessarily want to employ visual hints.

Table 2. Statistics for the INEX 2007 MM topics

	MMfragments	MMimages	All
Number of topics	19	20	39
Average number of terms in <code><title></code>	3.21	2.35	2.77
Number of topics with <code><mmtitle></code>	6	10	16
Number of topics with <code>src:</code>	2	7	9
Number of topics with <code>concept:</code>	4	6	10
Number of topics with both <code>src:</code> and <code>concept:</code>	0	3	3

5 Assessments

Since XML retrieval requires assessments at a sub-document level, a simple binary judgement at the document level is not sufficient. Still, for ease of assessment, retrieved fragments are grouped by document. Since the INEX 2007 MMfragments task was run in parallel with the Ad Hoc track, the assessments for this task were arranged by the Ad Hoc track organization as follows. Once all participants have submitted their runs, the top N fragments for each topic are pooled and grouped by document. The documents are alphabetised so that the assessors do not know how many runs retrieved fragments from a certain document or at what rank(s) the fragments were found. Assessors then look at the documents in the pool and highlight the relevant parts of each document. The assessment system stores the relevance or non-relevance of the underlying XML elements and passages.

We did not give any additional instructions to the assessors of multimedia topics, but assumed that topic creators who indicated that their topics have a clear multimedia character would only judge elements relevant if they contain at least one image. For the final proceedings we plan to analyse the assessments and to have some statistics on the actual amount of multimedia in the recall base.

The MMimages task is a document retrieval task. A document, i.e., an image with its metadata, is either relevant or not. For this task, we adopted TREC style document pooling of the documents and binary assessments at the document

(i.e., image with metadata) level. In 2006, the pool depth was set to 500 for the MMimages task, with post-hoc analysis showing that pooling up to 200 or 300 would have given the same system ordering [9]. This led to the decision to pool this year’s submissions up to rank 300, resulting in pools of between 348 and 1865 images per topic, with both mean and median around 1000 (roughly the same size as 2006).

6 Participants

Only four participants submitted runs for the INEX 2007 Multimedia track: CWI together with the University of Twente (CWI/UTwente), IRIT (IRIT), Queensland University of Technology in Australia (QUTAU) and University of Geneva (UGeneva). For the MMfragments task, three of the participants (CWI/UTwente, IRIT and QUTAU) submitted a total of 12 runs, whereas for the MMimages task, all four participants submitted a total of 13 runs.

Table 3 gives an overview of the topic fields used by the submitted runs. For MMfragments, six submissions used the topics’ `<title>` field, and six submissions used the `<castitle>` field; the `mmtitle` field was not used by any participant. For MMimages, seven submissions used the topics’ `<title>` field, and six submissions used the `<mmtitle>` field; no submissions used the `<castitle>` field which is to be expected since this is a document retrieval task.

Table 3. Topic fields used by the submitted runs

topic field	<i>#MMfragments</i> runs using it	<i>#MMimages</i> runs using it
title	6	7
castitle	6	0
mmtitle	0	6
description	0	0
narrative	0	0

Table 4 gives an overview of the resources used by the submitted runs. Not all groups detailed the resources they used, but judging from the descriptions it appears most submissions only used the target Wikipedia collection of the task at hand. It seems the Wikipedia images collection and the UvA features and classification scores have not been used in the MMfragments task this year. In the MMimages task, the visual resources provided are used by IRIT and UGeneva, whereas some runs also used the main Wikipedia XML collection.

Below we briefly discuss the approaches taken by the groups that participated in the Multimedia track at INEX 2007.

CWI/UTwente CWI/UTwente participated in both MMfragments and MMimages tasks of the INEX 2007 Multimedia track. For MMfragments, they limited their system to return only fragments that contain at least one image that

Table 4. Resources used by the submitted runs

resource	<i>#MMfragments</i> runs using it	<i>#MMimages</i> runs using it
wikipedia	12	4
wikipedia_IMG	0	8
UvAfeatures	0	1
UvAconcepts	0	2

was part of the Wikipedia images XML collection. They did not use any further multimedia processing and experimented with traditional text based approaches based on the language modelling approach and different length priors. For MMimages, they represented each image either by its textual metadata in the Wikipedia image XML collection, or by its textual context when that image appears as part of a document in the (Ad Hoc) Wikipedia XML collection. Retrieval was then based on purely text-based approaches.

IRIT IRIT participated in both the MMfragments and MMimages tasks of the INEX 2007 Multimedia track, with methods based on the context (text and structure) of images to retrieve multimedia elements. For MMimages topics, the "MMI" method proposed last year that uses 3 sources of evidence (descendants nodes, brother nodes and ascendant nodes) is compared to a new method "MMIConc" that uses in addition images classification scores. For the MMfragments task, the "MMF" method based on the "XFIRM Content and Structured" method and "MMI" method were evaluated. In future work, IRIT plan to extend images context by using links.

QUTAU TO BE COMPLETED.

UGeneva For their first participation at INEX MM, they submitted three runs to the MMimages task: (1) a baseline run based only on text-based retrieval, (2) an improvement of (1) with additional proper noun detection, and (3) a multi modal fusion approach using a hierarchical SVM approach.

For the simple text-based baseline run (1), the ready-to-use Matlab library TMG [10] is applied to the MMimages collection. It creates a term-document matrix filled with term frequencies of the textual input. The retrieval is done based on the Vector Space Model (VSM). In (2) the simple baseline run is improved by adding to the approach a proper noun detection based on Google result counts. This proved to be an easy and inexpensive way to reliably detect proper nouns. The multi modal fusion run (3) used all available features: textual and visual (color and texture histogram) low level features, plus the visual concepts provided by University of Amsterdam. The approach was set up hierarchically. First a VSM-based retrieval on the extended term-document matrix was executed. Then the result list was classified into N classes with the k-NN algorithm of the TMG library. The documents of the cluster containing the most relevant

documents were taken as input for a hierarchical Support Vector Machine (SVM) classification, which processes first each modality alone, before fusing all result lists in a final step.

Université de Saint-Etienne/JustSystems These two groups did not submit any official runs for the track, but they did help with assessments for the MMimages task, and plan to use the track’s data for future studies.

7 Results

This section presents the results for the submitted runs in each of the tasks.

7.1 MMfragments

Three participating groups (CWI/UTwente, IRIT and QUTAU) submitted a total of 12 MMfragment runs (5 Focused, 2 Relevant in Context and 5 Best in Context runs). These runs have been evaluated using the standard measures as used in the Ad Hoc track [4]: interpolated Precision (iP) and Mean Average interpolated Precision (MAiP) for the Focused task and non-interpolated generalized precision at early ranks gP[r] and non-interpolated mean average generalized precision. MAgP). Tables 5-7 show the results.

Table 5. MMfragment Results for Focused task.

MAiP	iP[0.00]	iP[0.01]	iP[0.05]	iP[0.10]	Group	Run
0.0719	0.1812	0.1719	0.1566	0.1511	qutau	CO_Focused
0.0737	0.2831	0.2370	0.1630	0.1434	qutau	COS_Focused
0.1045	0.3205	0.2301	0.2256	0.1799	utwente	article_MM
0.1017	0.1912	0.1909	0.1898	0.1827	utwente	star_loglength_MM
0.0033	0.2038	0.0420	0.0000	0.0000	utwente	star_lognormal_MM

Table 6. MMfragment Results for Relevant in Context task.

MAgP	gP[5]	gP[10]	gP[25]	gP[50]	Group	Run
0.0533	0.0741	0.0842	0.0658	0.0590	qutau	CO_RelevantInContext
0.0635	0.1345	0.1219	0.0957	0.0750	qutau	COS_RelevantInContext

Since the MMfragments topics were mixed with the Ad Hoc topics we received many more submissions that were not aimed at doing well on answering information needs with a multimedia character. We evaluated these runs on the subset of 19 multimedia topics and compared the results to the runs that were

Table 7. MMfragment Results for Best in Context task.

MAgP	gP[5]	gP[10]	gP[25]	gP[50]	Group	Run
0.0506	0.1133	0.1319	0.1267	0.0943	irit	iritmmf06V1
0.0541	0.1423	0.1394	0.0784	0.0437	irit	iritmmf06V2_BIC
0.0458	0.1164	0.1316	0.1114	0.0876	irit	iritmmf06V3_BIC
0.1783	0.3210	0.3039	0.2558	0.2099	qutau	CO_BestInContext
0.1533	0.3671	0.3084	0.2334	0.1761	qutau	COS_BestInContext

submitted specifically for the MMfragments task. For none of the tasks the best performing submission was a multimedia submission (more details in the final version of this paper). That shows that for this task standard text retrieval techniques are competitive. Since we do not have full insight in the details of all submissions, we can however not conclude that specific treatment of multimedia topics is useless. It may still be the case that a combination of techniques from the top performing Ad Hoc and Multimedia submissions would give better results on these topics than either alone.

7.2 MMimages

The four participating groups (CWI/UTwente, IRIT, QUTAU, and UGeneva) submitted a total of 13 MMimages runs. Figure 2 shows the interpolated recall precision graphs of these runs and Table 8 shows their mean average precision scores. Similarly to last year, the top performing runs do not use any image analysis or visual processing; they are purely text-based.

Table 8. Mean average precision (MAP) for submitted MMimages runs

group	run	MAP
utwente	title_MMim	0.2998
ugeneva	res_propernoun_07	0.2375
utwente	article_MMim	0.2240
ugeneva	res_baseline_07	0.1792
utwente	figure_MMim	0.1551
qutau	Run03	0.0482
irit	xfirm.mmi.01	0.0448
qutau	Run01	0.0447
irit	xfirm.mmi.01.conc	0.0445
qutau	Run04	0.0411
ugeneva	res_fusion_07	0.0165
qutau	Run02	0.0011

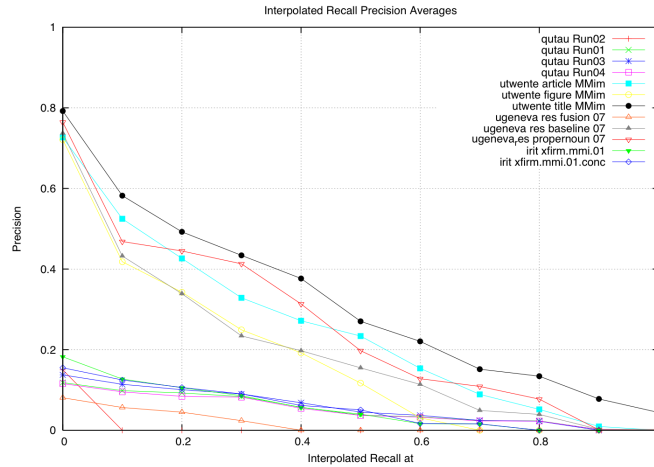


Fig. 2. MMimages: Interpolated Recall Precision Averages

8 Conclusions and Outlook

The INEX 2007 Multimedia track provides a nice collection of related resources (Wikipedia-based collections, together with a set of resources that are either starting points for or results of visual processing) to be used in the track's two retrieval tasks: MMfragments and MMimages. The main research questions these tasks aimed at addressing are the following: Do textual and structural hints need to be interpreted differently for the MMfragments compared to the Ad Hoc tasks? How do visual hints in the query help image and XML document fragment retrieval?

Since the number of participants in the multimedia track was disappointing with only four groups submitting runs, it is hard to draw general conclusions from the results. What we could see so far is that the top runs in both tasks did not make use of any of the provided visual resources. More detailed analyses of the results and the participants' system descriptions is needed to see if groups managed to improve over a textual baseline using visual indicators of relevance. Also, a topic by topic analysis could shine some light. Perhaps these techniques did contribute for only a limited number of topics and hurt for others.

For next year's multimedia track, we hope to draw more participants, from inside as well as outside INEX. The set of related collections and resources, makes this track an interesting playing ground, both for groups with a background in databases or information retrieval, and for groups with a deeper understanding of computer vision or image analysis.

References

1. C. L. A. Clarke, J. Kamps, and M. Lalmas. INEX 2007 retrieval task and result submission specification. Unpublished document distributed to INEX 2007 participants.
2. L. Denoyer and P. Gallinari. The Wikipedia XML Corpus. *SIGIR Forum*, 2006.
3. J. C. v. Gemert, J.-M. Geusebroek, C. J. Veenman, C. G. M. Snoek, and A. W. M. Smeulders. Robust scene categorization by learning image statistics in context. In *CVPRW '06: Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, page 105, Washington, DC, USA, 2006. IEEE Computer Society.
4. J. Pehcevski, J. Kamps, G. Kazai, M. Lalmas, P. Ogilvie, B. Piwowarski, and S. Robertson. INEX 2007 evaluation measures (draft). 2007.
5. C. G. M. Snoek, M. Worring, J. C. van Gemert, J.-M. Geusebroek, and A. W. M. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 421–430, New York, NY, USA, 2006. ACM Press.
6. M. Theobald, R. Schenkel, and G. Weikum. An efficient and versatile query engine for topx search. In *VLDB '05: Proceedings of the 31st international conference on Very large data bases*, pages 625–636. VLDB Endowment, 2005.
7. A. Trotman and B. Sigurbjörnsson. Narrowed extended xpath I (NEXI). In N. Fuhr, M. Lalmas, S. Malik, and Z. Szlavik, editors, *Advances in XML Information Retrieval: Third International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2004, Dagstuhl Castle, Germany, December 6-8, 2004, Revised Selected Papers*, volume 3493. Springer-Verlag GmbH, may 2005. <http://www.springeronline.com/3-540-26166-4>.
8. R. van Zwol, G. Kazai, and M. Lalmas. Inex 2005 multimedia track. In *Advances in XML Information Retrieval*, Lecture Notes in Computer Science. Springer, 2006.
9. T. Westerveld and R. van Zwol. The inex 2006 multimedia track. In N. Fuhr, M. Lalmas, and A. Trotman, editors, *Advances in XML Information Retrieval: Fifth International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2006*, Lecture Notes in Computer Science/Lecture Notes in Artificial Intelligence (LNCS/LNAI). Springer-Verlag, 2007.
10. D. Zeimpekis and E. Gallopoulos. TMG : A MATLAB toolbox for generating term-document matrices from text collections. In *Grouping Multidimensional Data (Recent Advances in Clustering)*, pages 187–210. Springer Berlin Heidelberg, 2006.

MM-XFIRM at INEX Multimedia track 2007

Mouna Torjmen, Karen Pinel-Sauvagnat, and Mohand Boughanem

SIG-RFI, IRIT, Toulouse, France

Abstract. This paper describes experiments carried out with the *MM-XFIRM* system in the *INEX Multimedia 2007* framework. The *MM-XFIRM* system is an adaptation of the XML textual information retrieval system *XFIRM* to support multimedia elements and answer to MMFragment et MMImages queries. Proposed approaches use the structural and textual context of multimedia elements to retrieve them.

Key words: xml, multimedia, image, fragment, context, structure

1 Introduction

Despite the main objective of the *INEX Multimedia Track*, which is "to exploit the XML structure that provides a logical level at which multimedia objects are connected, to improve the retrieval performance of an XML-driven multimedia information retrieval system" [10] [11], few works really reach this objective. Some of them use a textual XML retrieval system, without any specification for multimedia elements while others are based only on the images content. Some works however try to combine the two approaches. The two main questions behind the multimedia track are: can the document structure really improves the detection of relevant multimedia elements? And if so, how should it be used?

In this article, we present the *MultiMedia-XFIRM* system, based on the structured information retrieval system : *XFIRM* [7]. We use the multimedia elements context and especially the document structure to retrieve images and answer to multimedia fragments queries and multimedia images queries.

The rest of the paper is organised as follows. We first present the *XFIRM* system in section 2. Section 3 summarises our approaches for both tasks: MMImages and MMFragments. We evaluated two methods for the MMImages task. We processed in a first step images queries using only the main collection *Wikipedia Image XML Collection*, and in a second step, we tested the influence of adding an additional source of information like the *Image Classification Scores*. For the MMFragments task, we proposed a method for each sub-task: Focused, Relevant in Context and Best in Context. Section 4 shows our results for the two tasks. Finally, some conclusions and future works are given in section 5.

2 XFIRM model

The model is based on a relevance propagation method. During query processing, relevance scores are computed at leaf nodes level and then at inner nodes level

thanks to a propagation of leaf nodes scores through the document tree. An ordered list of subtrees is then returned to the user.

2.1 CO method

Let $q = t_1, \dots, t_n$ be a content-only query. Relevance values are computed thanks to a similarity function $RSV(q, ln)$.

$$RSV(q, ln) = \sum_{i=1}^n w_i^q * w_i^{ln}, \text{ where } w_i^q = tf_i^q \text{ and } w_i^{ln} = tf_i^{ln} * idf_i * ief_i \quad (1)$$

Where w_i^q and w_i^{ln} are the weights of term i in query q and leaf node ln respectively. tf_i^q and tf_i^{ln} are the frequency of i in q and ln respectively, $idf_i = \log(|D|/(|di| + 1)) + 1$, with $|D|$ the total number of documents in the collection, and $|di|$ the number of documents containing i , and ief_i is the inverse element frequency of term i , i.e. $\log(|N|/|nf_i| + 1) + 1$, where $|nf_i|$ is the number of leaf nodes containing i and $|N|$ is the total number of leaf nodes in the collection.

Each node in the document tree is then assigned a relevance score which is function of the relevance scores of the leaf nodes it contains and of the relevance value of the whole document.

$$r_n = \rho * |L_n^r|. \sum_{ln_k \in L_n} \alpha^{dist(n, ln_k)-1} * RSV(q, ln_k) + (1 - \rho) * r_{root} \quad (2)$$

$dist(n, ln_k)$ is the distance between node n and leaf node ln_k in the document tree, i.e. the number of arcs that are necessary to join n and ln_k , and $\alpha \in]0..1]$ allows to adapt the importance of the $dist$ parameter. $|L_n^r|$ is the number of leaf nodes being descendant of n and having a non-zero relevance value (according to equation 1). $\rho \in]0..1]$, inspired from work presented in [4], allows the introduction of document relevance in inner nodes relevance evaluation, and r_{root} is the relevance score of the *root* element, i.e. the relevance score of the whole document, evaluated with equation 2 with $\rho = 1$.

2.2 COS method

The evaluation of a CO+S query is carried out with the following steps:

1. INEX (NEXI) queries are translated into XFIRM queries
2. XFIRM queries are decomposed into sub-queries SQ and elementary sub-queries ESQ , which are of the form: $ESQ = tg[q]$, where tg is a tag name, i.e. a structure constraint, and $q = t_1, \dots, t_n$ is a content constraint composed of simple keywords terms.
3. Relevance values are then evaluated between leaf nodes and the content conditions of elementary sub-queries
4. Relevance values are propagated in the document tree to answer to the structure conditions of elementary sub-queries

5. Sub-queries are processed thanks to the results of elementary sub-queries
6. Original queries are evaluated thanks to upwards and downwards propagation of the relevance weights

Step 3 is processed thanks to formula 1. In step 4, the relevance value r_n of a node n to an elementary subquery $ESQ = tg[q]$ is computed according the following formula:

$$r_n = \begin{cases} \sum_{ln_k \in L_n} \alpha^{dist(n,ln_k)-1} * RSV(q,ln_k) & \text{if } n \in construct(tg) \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where the $construct(tg)$ function allows the creation of set composed of nodes having tg as tag name, and $RSV(q,ln_k)$ is evaluated during step 2 with formula 1. The $construct(tg)$ function uses a *Dictionary Index*, which provides for a given tag tg the tags that are considered as equivalent. This index is built manually.

More details about *CO* and *COS* methods can be found in [8].

3 Multimedia approaches

For the *MMImages* task, two methods are proposed: the *MMI* method, and the *MMIConc* method. The first one was already tested last year and the aim this year is to compare it with other methods. The second one tries to use images classification scores to improve results.

For the *MMFragment* task, we proposed a method for each sub-task: *MMF* method for Focused sub-task, *MMFBC* for Best in Context sub-task and *MMFRC* for Relevant in Context sub-task.

3.1 MMI method

In the *Wikipedia Images XML Collection*, each document contains exactly one image with metadata (often a short description and information about author and copyright).

Our method uses the text surrounding images and the document structure to judge images relevance. A first step is to search relevant nodes according to the *CO* method. Then, we only use documents having a score > 0 and we reduce our retrieval domain to both relevant nodes and images nodes belonging to relevant documents. For each image, we use the closest nodes to judge its relevance. The used nodes are: the descendant nodes, the ancestor nodes and the brother nodes (see Fig1).

An image score corresponding to each of the preceding sources of evidence is computed in function of:

- W_d^{im} is the image score computed using descendant nodes,
- W_b^{im} is the image score computed using brother nodes,
- W_a^{im} is the image score computed using ancestor nodes,

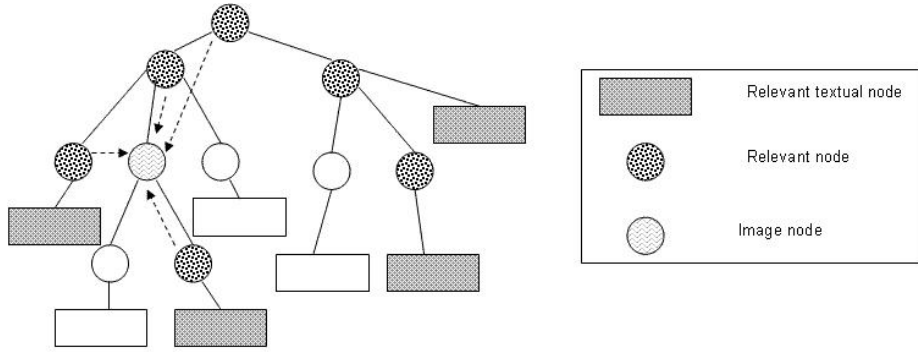


Fig. 1. Use of ancestor, brother and descendant nodes to evaluate images relevance

The total image score is then expressed as follows:

$$W_{im} = p_1 \cdot W_d^{im} + p_2 \cdot W_b^{im} + p_3 \cdot W_a^{im} \quad (4)$$

where p_1, p_2 and p_3 are parameters used to emphasize some weights types and $p_1 + p_2 + p_3 = 1$.

With this method, all the images of the relevant documents are evaluated and will have a score > 0 . Indeed, they will inherit at least of the root node score W_a^{im} . We summarize the evaluation of each score in the following paragraphs.

To evaluate the score of an image using its descendant nodes, we use the score of each relevant descendant node obtained by the *CO* method (W_{rdi}), the number of relevant descendant nodes according to the *XFIRM* model ($|d|$) and the number of non-relevant descendant nodes ($|\bar{d}|$).

$$W_d^{im} = f(W_{rdi}, |d|, |\bar{d}|) \quad (5)$$

If the number of relevant descendant nodes is greater than the number of non-relevant descendant nodes then they will have more importance in the score evaluation. Using this intuition, we apply the following formula in our experiments.

$$W_d^{im} = \left(\frac{|d| + 1}{|\bar{d}| + 1} \right) * \sum_{i=1}^{|d|} W_{rdi} \quad (6)$$

To evaluate the score of an image using its brother nodes, we use the score of each relevant brother node obtained by the *CO* method (W_{rbi}), the distance between the image node and each brother node ($dist(im, b_i)$): the larger the distance of the brother node from the image node is, the less it contributes to the image relevance. Finally, we use the number of relevant brother nodes $|b|$ and the number of non-relevant brother nodes $|\bar{b}|$

$$W_b^{im} = f(W_{rbi}, dist(im, b_i), |b|, |\bar{b}|) \quad (7)$$

The formula used in experiments presented here is :

$$W_b^{im} = \left(\frac{|b| + 1}{|\bar{b}| + 1}\right) * \left(\sum_{i=1}^{|b|} \frac{W_{rbi}}{dist(im, b_i)}\right) \quad (8)$$

To evaluate the score of an image using its ancestor nodes, we add the scores of relevant ancestor nodes obtained with the *CO* method (W_{rai}). The *CO* method uses the distance between the relevant node and its ancestors to evaluate the ancestors scores of an element: the larger the distance of a node from its ancestor is, the less it contributes to the relevance of its ancestor. Our method also uses the distance $dist(im, a_i)$ between the image node and its ancestors: the larger the distance from an ancestor node is, the less it contributes to the relevance of the image node. We used the following formula:

$$W_a^{im} = \sum_{i=1}^{|a|} \frac{\log(W_{rai} + 1)}{dist(im, a_i) + 1} \quad (9)$$

where $|a|$ is the number of relevant ancestor nodes according to the *XFIRM* model.

3.2 MMIConc method

In addition to the main collection *Wikipedia Images XML Collection*, a number of additional sources of information is provided. We used in this paper *Image Classification Scores* [9].

Some queries contain a needed concept (see Fig2). To process them, we use the *MMIConc* method which is composed of two steps. The first step is to identify relevant images (e.g documents) in the whole collection. In this phase, we use either the *CO*, *COS*, *MMI* or *MMF* methods on the *Wikipedia Image XML Collection*.

The second step is to refine results obtained in the first phase. For this purpose, we rank each result image according to the *Image Classification scores*. To do this, we added the image concept score to the original image score.

```
<mmtitle>#article[about(.,famous buildings of Paris) and about(.,concept:building)]</mmtitle>
```

Fig. 2. Example of query containing a concept need

3.3 MMF method

In this method, we adapted the COS method to process queries: we decomposed the query into sub-queries (see Fig3). For each sub-query, if its structure constraint is different from "image", we applied the COS method and if the sub-query element is "image", we applied the MMI method. Then, we propagated scores of sub-queries to the target element using the COS method.

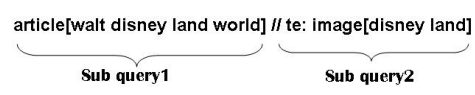


Fig. 3. MMFragment query decomposition in sub-queries

More details about *MMF* method can be found in [3].

3.4 MMFRC method

Here, we use the *MMF* method results that are already sorted from highest to lowest score. We traverse the results list and we group results by file name. For each file name in the result set, we construct a new result using the filename, *article[1]* as *path* and the highest element score in the document set result. We removed other results having the same file name, and we thus have one result for each document.

More precisely, we evaluate the context document using the element having the highest score in *MMF* method results.

3.5 MMFBC method

Using sorted *MMF* method results, we get the result having the highest score for each document, and we delete the other results for this document. We obtain for each document a result with the best element.

4 Runs and results

4.1 MMImages task

Table 1 shows *MMImages* track results using the different methods. The *MMI* (*Run6*) and *MMIConc* (*Run1* and *Run1'*) methods are compared to

Table 1. MMImages task results

Runs	Method	α	ρ	p_1	p_2	p_3	MAP	BPREF
Run1	MMIConc (with COS)	0.1	-	-	-	-	0.1376	0.1591
Run1'	MMIConc (with COS)	0.1	-	-	-	-	0.0445	0.0739
Run2	COS	0.1	-	-	-	-	0.1316	0.1447
Run2'	COS	0.1	-	-	-	-	0.0448	0.0730
Run3	COS	0.9	-	-	-	-	0.1585	0.1814
Run4	COS	0.6	-	-	-	-	0.1503	0.1630
Run5	CO	0.1	0.9	-	-	-	0.0089	0.0116
Run6	MMI	0.1	0.9	0.33	0.33	0.33	0.1270	0.1636
Run7	MMF	0.1	0.9	0.33	0.33	0.33	0.1211	0.1641

results obtained with *XFIRM* approaches (*Run2*, *Run2'*, *Run3*, *Run4*, *Run5*) and with the *MMF* method (*Run7*). For the *MMIConc* method, the *COS* method is used in the first step.

For *COS* method (*Run1*, *Run1'*, *Run2*, *Run2'*, *Run3*, *Run4*), *article* is used as the target element (e.g te: article[...]).

For *MMF* method (*Run7*), *image* is used as the target element (e.g te: image[...]).

Grayed boxes are results of our official runs (*Run1'* and *Run2'*). Bad results can be explained as follows: some returned documents were not part of the official INEX-MM collection and some other contained a few duplicates (within a topic the same document was returned more than once). *Run1* and *Run2* are the corrected runs corresponding to *Run1'* and *Run2'*.

The objective of these runs is to compare results with and without using concepts classification scores. With *MAP* measure, results are not really different (*MAP* increases of only 6%), but with *BPREF* metric, improvements increase up to 15%.

Comparing the 4 methods (*CO*: *Run5*, *COS*: *Run2*, *MMI*: *Run6*, *MMF*: *Run7*) by fixing $\alpha=0.1$ (and $\rho=0.9$), we obtained best results for *MAP* measure using *COS* method and for *BPREF* measure using *MMF* method. These results are discussed in the following paragraphs using the *MAP* measure as it is the *INEX* official metric.

In the *CO* method, we evaluate a score for each document element, and we use the highest element score as a document score in results. Thus, we don't use all document information to evaluate relevance, and we consequently obtain bad results (*MAP*=0.089).

In the *MMI* method, we search all images of relevant document and we evaluate for each one a score using the nearest nodes scores. *MAP* obtained is 0.1270.

Using "article" as target element in *COS* method, we evaluate a score for each leaf node, then, we propagate scores until the root element (e.g *article*). Here, the relevance is judged using all the contextual information of document. Results

are improved ($MAP=0.1316$).

The *MMF* method has the same principle as the *COS* method, but when query element is "image", we process it using the *MMI* method. In our case, all target elements are "image", so all queries are processed with the *MMI* method. The only difference of our *MMI* method runs is the scores propagation. *MMF* method uses the propagation formula of *COS* method, by using the parameter ρ , varying the participation of the root element (document context) in each inner node score. *MAP* obtained is 0.1211.

α is a parameter of the *XFIRM* system, used to quantify the importance of the distance between the nodes in the propagation formula (formula 3). As the best *MAP* is obtained for *COS* method, we varied α in *COS* runs (*Run2*, *Run3*, *Run4*). Best results are obtained with $\alpha=0.9$. We note that the more α increases, the more *MAP* increases: relevance weights should then be not too down weighted during propagation.

To conclude, we showed the interest of concepts classification scores to evaluate relevance of multimedia elements.

Moreover, by comparing the 4 methods (*CO: Run5*, *COS: Run2*, *MMI: Run6*, *MMF: Run7*), we note that methods using document context in nodes scores give best results (*COS and MMF methods*). Thus, in *Multimedia Images track*, the whole document allow to better evaluate image relevance than document elements.

4.2 MMFragments task

For Focused and Relevant In Context tasks, our runs are invalid because they contain *overlap*. We plan to correct and evaluate them soon.

For Best In Context task, we used *MMF* method. Table 2 shows our results.

Table 2. MMFragment Best In Context task results

Runs	p_1	p_2	p_3	MAgP	gP[5]	gP[10]	gP[25]	gP[50]
Run1	0.33	0.33	0.33	0.0506	0.1134	0.1319	0.1267	0.0944
Run2	0.5	0.5	0	0.0542	0.1423	0.1395	0.0785	0.0438
Run3	0	0.5	0.5	0.0459	0.1165	0.1317	0.1114	0.0876

For all runs, we used $\alpha=0.6$ and $\rho=0.9$.

p_1 is the parameter used for descendant elements.

p_2 is the parameter used for brother elements.

p_3 is the parameter used for ascendant elements.

Best *MAgP* is obtained using *Run2*, where only descendant and brother nodes are used to evaluate images relevance. These two sources of evidence give the best specification of images, and consequently, the best entry of the document when target element is "image".

In *Run1*, we added ascendant nodes to evaluate images relevance. This source of evidence degrades results (*MAGP* declines to 0.0506). We don't use descendant nodes in *Run3*. No images specification is used. *MAGP* obtained is 0.0459.

To conclude, using descendant and brother nodes to evaluate images relevance gives the best results in Best in Context task (*Run2*). Generally, they contain specific information on images. In the other hand, using the whole document degrades results (*Run1*).

5 Conclusion and future works

We presented the *MM-XFIRM* system that is composed of two parts: the first is designed to textual information retrieval and the second to images retrieval. This second part belongs to context based multimedia retrieval. At the time being, the context is composed of textual and structural information. In future work, we plan to extend the image context by integrating other elements. More precisely, we will study the use of links and content images features.

References

1. C. Buckley and E. M. Voorhees. Retrieval evaluation with incomplete information. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, Sheffield, United Kingdom*, pages 25–32. ACM, 2004.
2. N. Fuhr, M. Lalmas, S. Malik, and G. Kazai. INEX 2005 workshop proceedings, 2005.
3. L. Hlaoua, M. Torjmen, K. Pinel-Sauvagnat, and M. Boughanem. XFIRM at INEX 2006. Ad-hoc, Relevance Feedback and MultiMedia tracks. In N. Fuhr, M. Lalmas, and A. Trotman, editors, *International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX), Dagstuhl, Allemagne, 18/12/06-20/12/06*, volume LNCS 4518, pages 373–386, <http://www.springerlink.com>, mars 2007. Springer.
4. Y. Mass and M. Mandelbrod. Experimenting various user models for XML retrieval. In [2], 2005.
5. K. Nahrstedt, M. Turk, Y. Rui, W. Klas, and K. Mayer-Patel, editors. *Proceedings of the 14th ACM International Conference on Multimedia, October 23-27, 2006, Santa Barbara, CA, USA*. ACM, 2006.
6. J. Pehcevski, J. Kamps, G. Kazai, M. Lalmas, P. Ogilvie, B. Piwowarski, and S. Robertson. Inex 2007 evaluation measures. In *Pre-Proceedings of the 6th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2007, Dagstuhl Castle, Germany, December 17-19, 2007*, 2007.
7. K. Sauvagnat. *Modele flexible pour la recherche d'information dans des corpus de documents semi-structures*. PhD thesis, Toulouse : University Paul Sabatier, 2005.
8. K. Sauvagnat, L. Hlaoua, and M. Boughanem. Xfirm at inex 2005: ad-hoc and relevance feedback track. In *INEX 2005 Workshop proceedings*, pages 88–103, 2005.

9. C. Snoek, M. Worring, J. van Gemert, J.-M. Geusebroek, and A. W. M. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. In [5], pages 421–430, 2006.
10. T. Westerveld and R. van Zwol. Benchmarking multimedia search in structured collections. In *MIR '06: Proceedings of the 8th ACM international workshop on Multimedia information retrieval, Santa Barbara, California, USA*, pages 313–320, New York, NY, USA, 2006. ACM.
11. T. Westerveld and R. van Zwol. Multimedia retrieval at inex 2006. *SIGIR Forum*, 41(1):58–63, 2007.

An XML fragment retrieval method with image and text using textual information retrieval techniques (Extended Abstract)

Yu Suzuki¹, Masahiro Mitsukawa¹, Kenji Hatano²,
Toshiyuki Shimizu³, Jun Miyazaki⁴, and Hiroko Kinutani⁵

¹ Ritsumeikan University, 1-1-1 Noji-Higashi, Kusatsu, Shiga 5258577, Japan

² Doshisha University, 1-3 Tatara Miyakodani, Kyotanabe, 6100394, Japan

³ Kyoto University, Yoshida, Sakyo, Kyoto 6068501, Japan

⁴ Nara Institute of Science and Technology 8916-5 Takayama, Ikoma, Nara 6300192, Japan

⁵ University of Tokyo, 4-6-1 Komaba, Meguro, Tokyo 1538505, Japan

Abstract. In this paper, we propose a retrieval method using textual information retrieval techniques, such as vector space model, for XML documents which consist of image and text. Recently, many image retrieval systems are proposed. However, these systems are mainly based on pattern recognition techniques. Therefore, the features of images are also based on these recognition techniques, such as color histogram, and shape of the object in images. Generally, these systems do not consider weight of features, which means how important these features are, which are generally used in textual information retrieval systems. In this paper, we propose a method considering weight, such as TFIDF, to identify the importance degree of features. Using our proposed method, the system can retrieve intuitively similar retrieval target images to user's query images.

1 Introduction

In this paper, we propose a method for retrieving images using textual information retrieval method, such as tfidf-based method.

Recently, there are many documents which consist of not only textual information but also images, movies and music information, especially on the Internet. Therefore, many researchers have developed many information retrieval systems which deal with data which includes textual information, images, and other media. Then, these retrieval systems are widely used for many purposes. In this paper, we focus on these multimedia document retrieval system, especially multimedia XML document retrieval system.

When we develop XML document retrieval system, we should develop image retrieval system for retrieving images which is placed on retrieval target XML documents. Most image retrieval systems are based on the recognition techniques of images. This means that, the system try to recognize objects, scene, color, texture from images for constructing Content-based Image Retrieval (CBIR) systems.

On the other hand, when we develop textual information retrieval system, we do not deal with simple n-gram or basic boolean-like method. Instead of these naive methods, we deal with vector space model[1], probabilistic model, or several modern information

retrieval model to discover appropriate retrieval result documents. One of the reason about this retrieval model selection is that users do not need exactly same terms as query terms. Users should retrieve intuitively relevant documents to the user's queries.

We believe that if we merge textual information retrieval method with image information retrieval method, we can retrieve intuitively relevant images to the user's query images. We think that the aims of the pattern recognition techniques based retrieval methods are similar to simple n-gram and basic boolean-like methods of textual information retrieval systems.

We develop multimedia XML document retrieval systems which consist of textual XML document retrieval system and the image retrieval system described in this paper.

2 Overview of our Multimedia XML retrieval system

In this section, we describe an overview of our proposed system. Our system can divide into three parts.

1. **The system retrieves XML fragments using textual XML retrieval system.**
Using textual XML fragment retrieval method described in paper [2], we retrieve XML fragments.
2. **The system retrieves images using image XML retrieval system.**
Using image retrieval method described in section 3, we retrieve retrieval target images.
3. **The system merge two result lists.**
Using two retrieval results from textual XML fragments retrieval system and image retrieval system, the system generate one retrieval result by calculating the integrated retrieval status values of each XML fragment.

In this paper, we describe the second part, such as a method for retrieving images.

3 Image retrieval method using textual information retrieval techniques

In this section, we describe a method for retrieving images using textual information retrieval techniques.

In this method, we process the following four methods.

1. **The system extracts features from retrieval target images.**
From retrieval target image retrieval system, the system extracts feature values. We use CIELab color space as feature values.
2. **The system extracts features from user's query image.**
From user's query image, the system also extract feature values using same method as the method for retrieval target images.
3. **The system calculates tfidf-based weights from image features.**
General image retrieval simply deals with image features in straight. In our method, we calculate weights of feature values for emphasizing intuitively characteristic features of images.

4. **The system calculates retrieval status values using Earth Mover's Distance.**
We deals with Earth Mover's Distance [3] to calculate the retrieval status values between the retrieval target image and the user's query image.

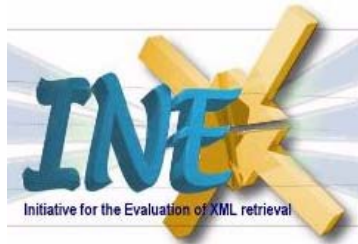
4 Conclusion

In this paper, we proposed a multimedia XML fragment retrieval system using image retrieval method with textual information retrieval techniques. In our proposed method, we deal with textual information retrieval method for retrieving images.

Our implemented system can only process queries which specify exact images, then our system cannot process concept clauses. Therefore, we should consider how to calculate retrieval status values using concept clauses.

References

1. Salton, G., McGill, M.: Introduction to Modern Information Retrieval. McGraw-Hill Book Company (1983)
2. Hatano, K., Shimizu, T., Miyazaki, J., Suzuki, Y., Kinutani, H., Yoshikawa, M.: Ranking and displaying search results based on content-and-structure conditions of xml documents and queries. In: Informal Proceedings of INitiative for the Evaluation of XML Retrieval. (2007)
3. Rubner, Y., Tomasi, C., Guibas, L.J.: The earth mover's distance as a metric for image retrieval. International Journal of Computer Vision **40**(2) (2004) 99 – 121



INEX 2007 Guidelines for Topic Development

Andrew Trotman, Birger Larsen, *et al*[†]

1 Aims

The aim of the INEX initiative is to provide the means, in the form of a large test collection and appropriate measures, for the evaluation of content-oriented XML element retrieval. Within the INEX initiative it is the task of the participating organizations to provide the topics and relevance assessments that will contribute to the test collection. Each participating organization, therefore, plays a vital role in this collaborative effort.

2 Introduction

Test collections, as traditionally used in information retrieval (IR), consist of three parts: a set of documents, a set of information needs called topics, and a set of relevance assessments listing (for each topic) the set of relevant documents.

A test collection for XML retrieval differs from traditional IR test collections in many respects. Although it still consists of the same three parts, the nature of these parts is fundamentally different. In IR test collections, documents are considered units of unstructured text, queries are generally treated as collections of terms and / or phrases, and relevance assessments provide judgments whether a document as a whole is relevant to a query or not. XML documents, on the other hand, organize their content into smaller, nested structural elements. Each of these elements in the document's hierarchy, along with the document itself (the root), is a retrievable unit. In addition, with the use of XML query languages, users of an XML retrieval system can express their information need as a combination of content and structural conditions: they can restrict their search to specific structural elements within the collection. Consequently the relevance assessments for an XML collection must also consider the structural nature of a document and provide assessments at different levels of the document hierarchy.

This guide deals only with topics. Each group participating in INEX will have to submit **6 CO+S topics by 11th May 2007**. This guide provides detailed guidelines for creating these topics.

3 Topic Creation Criteria

Creating a set of topics for a test collection requires a balance between competing interests. The performance of retrieval systems varies largely for different topics. This variation is usually greater than the performance variation of different retrieval methods on the same topic. Thus, to judge whether one retrieval strategy is (in general) more effective than another, the retrieval performance must be averaged over a large and diverse set of topics. In addition, to be a useful diagnostic tool, the average performance of the retrieval systems on the topics can be neither too good nor too bad as little can be learned about retrieval strategies if systems retrieve no, or only relevant, documents.

When creating topics, a number of factors should be taken into consideration. Topics should:

- be authored by an expert in (or someone familiar with) the subject areas covered by the collection,
- reflect real needs of operational systems,
- represent the type of service an operational system might provide,
- be diverse,
- differ in their coverage, e.g. broad or narrow topic queries,
- be assessed by the topic author.

[†] Based on prior guidelines additionally authored by Börkur Sigurbjörnsson, Shlomo Geva, Mounia Lalmas, and Saadia Malik

4 Topic Format

In previous years, different topic types have been used for the two main *ad hoc* retrieval tasks at INEX (i.e., a distinction was made between Content Only (CO) and Content And Structure (CAS) topics). In addition, different parts of the topics were designed also to be used in other tracks (e.g., the topic description was tuned to the needs of the Natural language Processing (NLP) track). These topic types were all merged into one type for INEX 2006: **Content Only + Structure (CO+S) Topics**. Likewise, in the 2007 topics all the information needed by the different *ad hoc* tasks and tracks are expressed in the individual topic parts, and only one topic type is needed. The 2007 CO+S topics consist of the following parts, which are explained in detail below:

<title>	in which Content Only (CO) queries are given
<castitle>	in which Content And Structure (CAS) queries are given
<description>	a one or two sentence natural language definition of the information need
<narrative>	in which the definitive definition of relevance and irrelevance are given

4.1 General considerations

A clear and precise description of the information need is required in order to unambiguously determine whether or not a given element fulfills the given need. In a test collection this description is known as the **narrative**. It is the only true and accurate interpretation of a user's needs. Precise recording of the narrative is important for scientific repeatability – there must exist, somewhere, a definitive description of what is and is not relevant to the user. To aid this, the <narrative> should explain not only what information is being sought, but also the context and motivation of the information need, i.e., *why* the information is being sought and what work-task it might help to solve.

Many different queries could be drawn from the <narrative>, and some are better than others. For example, some might contain phrases; some might contain ambiguous words; while some might even contain domain specific terms or structural constraints. Regardless of the query, the search engine results are not necessarily relevant. Even though a result might contain search terms from the query, it might not match the explanation given in the <narrative>. Equally, some relevant documents might not be found, but they remain relevant because they are described as so by the <narrative>.

The different CO+S topic parts relate to different scenarios that lead to different types of queries.

The topic <title> simulates a user who does not know (or does not want to use) the actual structure of the XML documents in a query. The query expressed in the topic <title> is therefore a Content Only (CO) query. This profile is likely to fit most users searching XML digital libraries.

Upon discovering their <title> query returned many irrelevant hits, a user might decide to add structural hints (to rewrite as a CAS query). This is similar to a user adding + and – to a web query when too many irrelevant pages are found. At INEX, these added structural constraints (+S) are specified using the formal syntax called NEXI [1] (see the INEX website for the specification) - and recorded in the topic <castitle>.

Example

Suppose a user wants to find pictures of the Apple II computer. They enter the CO query:

```
Apple II figure
```

but discover that most results are figures of products for the Apple II. They decide to add structural hints:

```
//figure[about(.., Apple II)]
```

restricting the results to `figure` elements only, known to contain the captions of figures.

4.1.1 The Ad Hoc Task

The CO+S task is investigating relevance ranking algorithms for *ad hoc* element retrieval. This year the task is continuing to investigate the usefulness of structural hints.

4.1.2 The MMfragments Task

The MMfragments task studies relevance ranking algorithms for Multimedia element retrieval. The task investigates the usefulness of structural hints as well as the contributions of the various media representations in the topic. If your information need has a clear multimedia character, i.e., if you expect relevant fragments to contain images (for example when you want to find pictures of the Apple II computer), the topic may be suitable for the MMfragments task. You can indicate this using a checkbox on the submission form. Optionally, you can add specific multimedia hints in the NEXI query, as explained in Appendix 2.

4.2 Topic parts

Topics are made up of several parts, these parts explain the *same information need*, but for different purposes. An example of a full topic combining all these is given in the Appendix 1.

<narrative> A detailed explanation of the information need and the description of what makes an element relevant or not. The <narrative> should explain not only what information is being sought, but also the context and motivation of the information need, i.e., *why* the information is being sought and what work-task it might help to solve. Assessments will be made on compliance to the narrative alone; it is therefore important that this description is clear and precise.

<title> A short explanation of the information need. It serves as a summary of the content of the user's information need. The exact format of the topic title is discussed in more detail below.

<castitle> A short explanation of the information need, specifying any structural requirements. The exact format of the castitle is discussed in more detail below. The castitle is optional but the majority of topics should include one.

<description> A brief description of the information need written in natural language.

Note that the <description> must be interchangeable with the <title> and <castitle>. Any ambiguity or disagreement is resolved by reference to the <narrative>, the only accurate definition of the information need. The topic <title>, <castitle> and <description> are discussed in detail below.

4.2.1 Topic <title>

To ensure topics are syntactically correct, a parser has been implemented in Flex and Bison (the GNU tools compatible with LEX and YACC) and is available for download or online use (see <http://metis.otago.ac.nz/abin/nexi.cgi>)

The topic title is a short representation of the information need. Each term is either a word or a phrase. Phrases are encapsulated in double quotes. Furthermore the terms can have either the prefix + or -, where + is used to emphasize an important concept, and - is used to denote an unwanted concept.

Example

A user wants to retrieve information about computer science degrees that are not master degrees:

```
"computer science" +degree -master
```

the + and - signs are used as hints to the search engine and do not have strict semantics. As an example the following text might be judged relevant to the information need, even though it contains the word master.

```
The university offers a program leading to a PhD degree in computer science.
Applicants must have a master degree...
```

Example

A user wants to retrieve information about information retrieval from semi-structured documents:

```
"information retrieval" +semi-structured documents
```

As in the previous example the following text might be judged relevant, even though it neither contains the word semi-structured, nor the phrase “information retrieval”.

The main goal of INEX is to promote the evaluation of content-oriented XML retrieval by providing a large test collection of XML documents, uniform scoring procedures, and a forum for organizations to compare their results...

Although the semantics of phrases and the + / – tokens is not strict, they may be of use to the search engine.

4.2.2 Topic <castitle>

As structural constraints are not an inherent part of all information needs the <castitle> is optional. However, we aim at having topics for INEX 2007 where the **majority of topics do include a castitle**. This is needed in order to facilitate the evaluation of structural hints, which is a central concern at INEX.

Only a high level description is included here, for a more formal specification of the topic description language (NEXI) see the INEX web-site or in the proceedings of INEX 2004 [1].

To make sure that topics are syntactically correct, parsers have been implemented in Flex and Bison (the GNU tools compatible with LEX and YACC) and are available for download. An online version of the parser is also available: <http://metis.otago.ac.nz/abin/nexi.cgi>

Castitles are XPath (<http://www.w3c.org/TR/xpath>) expressions of the form:

A[B]

or

A[B]C[D]

where A and C are navigational XPath expressions using only the descendant axis. B and D are predicates using *about* functions for text (explained below); the arithmetic operators <, <=, >, and >= for numbers; and the connectives *and* and *or*. The *about* function has (nearly) the same syntax as the XPath function *contains*. Usage is restricted to the form:

```
about(.path, query)
```

where *path* is empty or contains only tag-names and descendant axis; and *query* is an IR query having the same syntax as the CO titles (i.e. query terms). The *about* function denotes that the content of the element located by the path is about the information need expressed in the query. As with the title, the castitle is only a hint to the search engine and does not have definite semantics.

Example

A user wants to know about Tolkien’s languages and assumes an article on Tolkien will have a section discussing these languages:

```
//article[about(., Tolkien)]//section[about(., language)]
```

But the user might be happy with retrieving whole articles. In the formalism expressed above,

```
A = //article
B = about(., Tolkien)
C = //section
D = about(., language)
```

A CAS query contains two kinds of structural hints: where to look (support elements; in this case `//article` and `//article//section`), and which elements to return (target elements; in this case `//article//section`). In prior INEX workshops the target element hint has been interpreted either strictly or loosely (vaguely). Where to look has always been interpreted loosely. This created considerable debate over how to interpret where to look. There is the database view: *all* structural constraints must

be followed strictly (by exact match). Then there is the information retrieval view: an element is relevant if it satisfies the information need, irrespective of the structural constraints.

The main purpose of the INEX initiative is to build a test collection for the evaluation of content oriented XML retrieval. The most valuable part of the collection is the human made relevance assessments. Thus, each structured query **must** have at least one *about* function in the rightmost predicate.

4.2.3 Topic <description>

The <description> should be precise and concise, but it must contain the same terms and the same structural requirements that appear in the <title> and the <castitle>, albeit expressed in natural language.

Example

A user wants to retrieve information about computer science degrees that are not master degrees and has chosen the title query:

```
"computer science" +degrees -master
```

for the <title>. From this they might choose a <description> of either:

```
retrieve information about degrees in computing science, but not masters degrees
```

or

```
I want descriptions of computer science degrees that are not master degrees
```

as they are equivalent, but:

```
get information about computing degrees, but not about master or PhD computing degrees
```

cannot be chosen as it expresses a different information need - there is an additional requirement that information about PhD degrees is not sought.

It is important to compare results that are based on natural language queries (the <description>) with queries that are based on the more formal languages (the <title>, and <castitle>). The description must, therefore, be as informative as the <title> and <castitle>.

5 Procedure for Topic Development

Each participating group will have to submit **6 CO+S** topics by the **11^h May 2007**. Submission is done by filling in the Candidate Topic Submission Form on the INEX web site:
<http://inex.is.informatik.uni-duisburg.de/2007/> under Tasks/Tracks → Adhoc → Topics.

The topic creation process is divided into several steps. When developing a topic, use a print out of the online Candidate Topic Form to record all information about the topic you are creating.

Step 1: Initial Topic Statement

Create a one or two sentence description of the information you are seeking. This should be a simple description of the information need without regard to retrieval system capabilities or document collection peculiarities. This should be recorded in the Initial Topic Statement field. Record also the context and motivation of the information need, i.e. *why* the information is being sought. Add to this a description of the *work-task*, that is, with what task it is to help (e.g. writing an essay on a given topic).

Step 2: Exploration Phase

In this step the initial topic statement is used to explore the collection. Obtain an estimate of the number of relevant elements then evaluate whether this topic can be judged consistently. You may use any retrieval engine for this task, including your own or the TopX system (<http://infao5501.ag5.mpi-sb.mpg.de:8080/topx/>), provided through the INEX website.

Step 2a: Assess Top 25 Results

Judge the top 25 retrieval results. To assess the relevance of a retrieved element use the following working definition: *mark it relevant if it would be useful if you were writing a report on the subject of the topic, or if it contributes toward satisfying your information need.* Each result should be judged on its own merits. That is, information is still relevant even if it is the thirtieth time you have seen the same information. It is important that your judgment of relevance is consistent throughout this task. Using the Candidate Topic Submission Form record the number of found relevant elements and the path representing each relevant element. Then if there are:

- fewer than 2 or more than 20 relevant within the top 25, abandon the topic and use a new one,
- more than 2 and fewer than 20 relevant within the top 25, perform a feedback search (see below).

Step 2b: Feedback Search

After assessing the top 25 elements, you should have an idea of which terms (if any) could be added to the query to make the query as expressive as possible for the kind of elements you wish to retrieve.

Use the expanded query, to retrieve a new list of candidates. Judge the top 100 results (some are already judged), and record the number of relevant results in Candidate Topic Form. Record the expanded query in the title field of the Candidate Topic Submission Form.

Step 3: Write the <narrative>

Having judged the top 100 results you should have a clear idea of what makes a component relevant or not. It is important to record this in minute detail as the <narrative> of the topic. The <narrative> is the definitive instruction used to determine relevance during the assessment phase (after runs have been submitted). Record not only what information is being sought, but also what makes it relevant or irrelevant. Also record the context and motivation of the information need. Include the work-task, which is: the form the information will take after having been found (e.g. written report), or record a use-case which is: the reason the user needs XML-IR to solve their problem. Make sure your description is exhaustive as there will be several months between topic development and topic assessment.

Step 4 CO+S: Optionally write the <castitle>

Optionally re-write the title by adding structural constraints and target elements. Record this as the <castitle> on the Candidate Topic Submission Form. Also record why you think the structural hints might help in the <narrative>. Please note that we aim at having castitles in most topics.

Step 5: Write the <description>

Write the <description>, the natural language interpretation of the query. Ensure the information need as expressed in the <title>, and <castitle> is also expressed in the <description>.

Step 6: Refining Topic Statements

Finalize the topic <title>, <castitle>, <description>, and <narrative>. It is important that these parts all express the same information need; it should be possible to use each part of a topic in a stand-alone fashion (e.g. title for retrieval, description for NLP, etc.). In case of dispute, the <narrative> is the definitive definition of the information need – all assessments are made relative to the <narrative> and the <narrative> alone.

Step 7: Topic Submission

Once you are finished, fill out and submit the on-line Candidate Topic Submission Form on the INEX website <http://inex.is.informatik.uni-duisburg.de/2007/> under Tasks/Tracks → Adhoc → Topics. After submitting a topic you will be asked to fill out an online questionnaire (this should take no longer than 5-10 minutes). It is important that this is done as part of the topic submission as the questions relate to the individual topic just submitted and the submission process. This is part of an effort to collect more context for the INEX topics, thereby increasing the reusability of the test collection. Initial results demonstrating the applicability of this can be found in [2].

Please make sure you submit all candidate topics no later than the **11th May 2007**.

6 Topic Selection

From the received candidate topics, the INEX organizers will decide which topics to include in the final set. This is done to ensure inclusion of a broad set of topics. The data obtained from the collection exploration phase *is* used as part of the topic selection process. The final set of topics will be distributed for use in retrieval and evaluation.

7 Acknowledgments

The topic format proposed in this document is based on the outcome of working groups set up during previous INEX workshops along with the online discussions they created. We are very grateful for this contribution. This document is a modified version of the topic development guides from previous INEX workshops additionally authored by Börkur Sigurbjörnsson, Shlomo Geva, Mounia Lalmas, and Saadia Malik.

References

- [1] Trotman, A., & Sigurbjörnsson, B. (2004). Narrowed Extended XPath I (NEXI). In *Proceedings of the INEX 2004 Workshop*, (pp. 16-40).
- [2] Kamps, J. and Larsen, B. (2006). Understanding Differences between Search Requests in XML Element Retrieval. In *Proceedings of the SIGIR 2006 Workshop on XML Element Retrieval Methodology*, p. 13-19. [<http://www.cs.otago.ac.nz/sigirmw/>]

Appendix 1: Example CO+S Topic

```
<inex_topic query_type="CO+S">
<title>Tolkien languages "lord of the rings"</title>
<castitle>//article[about(., Tolkien) or about(., "lord of the rings")]//sec[about(.,
Tolkien languages)]</castitle>
<description>Find information about Tolkien languages from the Lord of the
Rings.</description>
<narrative>The "Lord of the Rings" movie trilogy fascinate me. I have learned from
other fans that the languages spoken by e.g., elves and dwarfs in the screen version
are not just the usual effects. Apparently, these languages were invented by Tolkien
himself and are central to his work with the original books.
```

For my own personal interest, I would like to learn more background about Tolkien's artificial languages, and how they have affected the world portrayed in the Lord of the Rings universe. Later I may want to add a section on the influence languages to my Lord of the Rings fan web page. As Tolkien's languages seem to be a rather specialized topic, I expect to find relevant information as elements in larger documents that deal with Tolkien or Lord of the Rings, e.g., as sections in documents about Tolkien or the Lord of the Rings (although I would be pleasantly surprised to see whole documents on the topic of Tolkien's languages).

```
To be relevant an element should discuss Tolkien's artificial languages and their
influence on the Lord of the Rings books or movies. Information on the languages alone
without explicit discussion of their impact on the books or movies is not relevant;
nor is general information on Tolkien or the Lord of the Rings.</narrative>
</inex_topic>
```

Appendix 2: MMfragments extensions to NEXI

In the MM track two special types of about clauses are allowed, both specifying visual hints or constraints. These visual about clauses can never be the only about clause. Systems that do not handle visual information must also be able to process these topics (the topics will be mixed with Ad Hoc topics). Therefore a traditional, textual about clause is always required.

The first type is used for visual similarity. If a user wants to indicate that results should have images similar to a given example image, this can be indicated in an about clause with an URL. For example to find pictures of the Apple II computer similar to the one at <http://lrs.ed.uiuc.edu/students/scooper/AppleII.jpg>, one could type

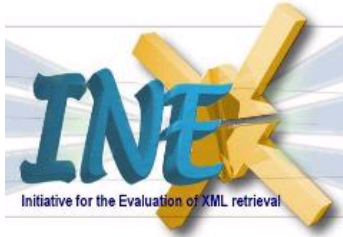
```
//figure[about(., Apple II) and  
about(.,http://lrs.ed.uiuc.edu/students/scooper/AppleII.jpg)]
```

Please make sure the image you use as an example is not part of the INEX wikipedia collection, since we do not want to give credit for finding the example image itself. Also, try to use images from the .edu and .gov domains as they are expected to be more stable. Although, we will keep copies of the images, we cannot guarantee they will always be available, the URL in the NEXI query is the primary source for the image.

The second type of visual hints is directly related to the image classifications that are provided as an additional source of information (for details, see the multimedia track pages at the INEX 2007 website: <http://inex.is.informatik.uni-duisburg.de/2007/> under Tracks → Multimedia). If a user thinks the results should be of a given concept, this can be indicated with an about clause with the keyword `concept:`. For example, to search for cityscapes one could decide to use the concept building:

```
//image[about(.,cityscape) and about(.,concept:building)]
```

Terms following the keyword `concept:` are restricted to the 101 concepts for which classification results are provided (cf. the multimedia track pages at the INEX 2007 website).



INEX 2007 Retrieval Task and Result Submission Specification

Charles L. A. Clarke, Jaap Kamps, Mounia Lalmas

Sunday, June 3, 2007

What's New in 2007? The INEX 2007 Adhoc track sees a continuation of three retrieval tasks: the FOCUSED TASK, the RELEVANT IN CONTEXT TASK, and the BEST IN CONTEXT TASK. The main change at INEX 2007 is the liberalization to arbitrary passages. That is, a retrieval result can be either an XML element, or an arbitrary passage (a sequence of textual content either from within an element, or spanning a range of elements).

1 Retrieval Task

The retrieval task to be performed by the participating groups of INEX 2007 is defined as the adhoc retrieval of XML elements or passages. In information retrieval (IR) literature, adhoc retrieval is described as a simulation of how a library might be used, and it involves the searching of a static set of documents using a new set of topics. While the principle is the same, the difference for INEX is that the library consists of XML documents, the queries may contain both content and structural conditions and, in response to a query, arbitrary XML elements may be retrieved from the library. Moreover, at INEX 2007, we also allow the submission of arbitrary passages.

The general aim of an IR system is to find *relevant information* for a given topic of request. In the case of XML retrieval there is, for each article containing relevant information, a choice from a whole hierarchy of different elements or passages to return. Hence, within XML retrieval, we regard as *relevant results* those results that both

- contain relevant information (the result exhaustively discusses the topic), but
- contain as little non-relevant information as possible (the result is specific for the topic).

For example, if an XML element contains another element but they have the same amount of relevant text, the shorter element is strictly more specific and a preferred result. The same holds for different passages covering the same amount of relevant text.

Within the adhoc XML retrieval task we define the following three sub-tasks:

1. FOCUSED TASK asks systems to return a ranked list of elements or passages to the user.
2. RELEVANT IN CONTEXT TASK asks systems to return relevant elements or passages clustered per article to the user.
3. BEST IN CONTEXT TASK asks systems to return articles with one best entry point to the user.

1.1 FOCUSED TASK

1.1.1 Motivation for the Task

The scenario underlying the FOCUSED TASK is to return to the user a ranked-list of element or passages for her topic of request. The INEX 2007 FOCUSED TASK asks systems to find the most focused results that satisfy a information need, without returning “overlapping” elements. That is, for a given topic, no retrieval result in the result set may contain text already contained in another result. Or, in terms of the XML tree, no element in the result set should be a child or descendant of another element. The task makes a number of assumptions:

Display the results are presented as a ranked-list of results to the user.

Users view the result list top-down, one-by-one. Users do not want overlapping results in the result-list, and if equally relevant prefer shorter results over longer ones.

What we hope to learn from this task is: How important is the XML document-structure? How effective are XML element retrieval approaches relative to pure passage retrieval? How do structural constraints in the query help retrieval?

1.1.2 Results to Return

The aim of the FOCUSED TASK is to return a ranked-list of elements or passages, where no result may be overlapping with any other result. Since ancestors elements and longer passages may also be relevant (be it to a lesser or greater extent) it is a challenge to chose the results appropriately. *Please note that submitted runs containing overlapping results will be disqualified.*

Summarizing: FOCUSED TASK returns results ranked in relevance order (where specificity is rewarded). Overlap is **not** permitted in the submitted run.

1.2 RELEVANT IN CONTEXT TASK

1.2.1 Motivation for the Task

The scenario underlying the RELEVANT IN CONTEXT TASK is to return the relevant information (captured by a set of elements or passages) within the context of the full article. As a result, an article devoted to the topic of request, will contain a lot of relevant information across many elements. The INEX 2007 RELEVANT IN CONTEXT TASK asks systems to find a set of results that corresponds well to (all) relevant information in each article. The task makes a number of assumptions:

Display results will be grouped per article, in their original document order, providing access through further navigational means.

Users consider the article as the most natural unit, and prefer an overview of relevance in their context.

What we hope to learn from this task is: How does the user-oriented RELEVANT IN CONTEXT TASK differ from FOCUSED TASK? What techniques are effective at locating relevance within articles? How do structural constraints in the query help retrieval?

1.2.2 Results to Return

The aim of the RELEVANT IN CONTEXT TASK is to first identify relevant articles (the fetching phase), and then to identify the relevant results within the fetched articles (the browsing phase). In the fetching phase, articles should be ranked according to their topical relevance. In the browsing phase, we have a set of results that cover the relevant information in the article. The `/article[1]` element itself need not be returned, but is implied by any result from a given article. Since the content of an element is fully contained in its parent element and ascendants, the set may **not** contain overlapping elements. Also passage results may **not** be overlapping. *Please note that submitted runs containing results from interleaved articles will be disqualified, as will submitted runs containing overlapping results.*

Summarizing: RELEVANT IN CONTEXT TASK returns a ranked list of articles. For each article, it returns an unranked **set** of results, covering the relevant material in the article. Overlap is not permitted.

1.3 BEST IN CONTEXT TASK

1.3.1 Motivation for the Task

The scenario underlying the BEST IN CONTEXT TASK is to find the best-entry-point for starting to read articles with relevance. As a result, even an article completely devoted to the topic of request, will only have one best starting point to read. The INEX 2007 BEST IN CONTEXT TASK asks systems to find the XML elements or passages that corresponds to these best-entry-points. The task makes a number of assumptions:

Display single result per article.

Users consider the article as the most natural unit, and prefer to be guided to the best point to start to read the most relevant content.

What we hope to learn from this task is: How does the BEST IN CONTEXT TASK differ from the RELEVANT IN CONTEXT TASK? How do best-entry points relate to the relevance of elements (FOCUSED TASK)? How do structural constraints in the query help retrieval?

1.3.2 Results to Return

The aim of the BEST IN CONTEXT TASK is to first identify relevant articles (the fetching phase), and then to identify the element corresponding to the best entry points for the fetched articles (the browsing phase). In the fetching phase, articles should be ranked according to their topical relevance. In the browsing phase, we have a single element or passage whose first content corresponds to the best entry point for starting to read the relevant information in the article. Note that there is no implied end-point: if (the start of) a paragraph is returned, it's not indicating that the reader should stop at the end of the paragraph. Similarly, although we request a complete passage with start and end-point out of practical convenience, the end-point of a passage result is ignored in the evaluation. The `/article[1]` element itself may be returned in case it is the best entry point, otherwise it will implied by any result from a given article. *Please note that submitted runs containing multiple results per article will be disqualified.*

Summarizing: BEST IN CONTEXT TASK returns a ranked list of articles. For each article, it returns a **single** result, representing the best entry point for the article with respect to the topic of request.

1.4 Passage Retrieval and Structured Queries

Within the INEX 2007 Adhoc retrieval tasks, we invite participants to experiment with two sets of different retrieval approaches:

- XML element retrieval versus passage retrieval.
- Standard keyword query (CO) retrieval versus structured query (CAS) retrieval.

1.4.1 Elements or arbitrary passages

XML element retrieval makes use of the document structure to determine the potential units of retrieval. The document structure is capturing both semantic labels (think of a link, a figure, or a figure caption) or ways of structuring the text (think of the sectioning structure). For example, in case of the XML Wikipedia collection, the sectioning structure of the document reflects the author's view on which parts of the text naturally group together and form a coherent subpart of the article. But how useful is the document structure to single out the relevant text for a particular search request? For example, a section element may be too short, and the relevant text should also contain the last paragraph of a preceding section. Or, alternatively, a paragraph element may be too long since all relevant information is contained in the first two sentences. To answer such questions, INEX 2007 also allows arbitrary passages—a result starting anywhere in the content of an element, and ending anywhere in the content of the same or another element later in document order.

At INEX 2007 there is no separate passage retrieval task, but for all three tasks arbitrary passages may be returned instead of elements. Although both types of retrieval may be used for each task, the best performing element and passage runs will also be reported. The type of results, XML elements or arbitrary passages, is recorded in submission format.

1.4.2 Structured Queries

Queries with content-only conditions (CO queries) are requests that ignore the document structure and contain only content related conditions, e.g. only specify what an element should be about without specifying what that component is. The need for this type of query for the evaluation of XML retrieval stems from the fact that users may not care about the structure of the result components or may not be familiar with

the exact structure of the XML documents. CAS queries are more expressive topic statements that contain explicit references to the XML structure, and explicitly specify the contexts of the user's interest (e.g. target elements) and/or the context of certain search concepts (e.g. containment conditions). More precisely, a CAS query contains two kinds of structural constraints: where to look (i.e. the support elements), and what to return (i.e. the target elements). The structural constraints are considered as structural hints, and similar to CO queries the elements will be assessed using the `<narrative>` part of the topics. Runs using CO queries and runs using CAS queries will be merged to create the assessment pool (this will in fact improve the pool quality).

At INEX 2007 there is no separate CAS task, but the vast majority of topics have both a keyword CO query and a structured CAS query.¹ As noted above, for all the tasks, we want to find out if, when and how the structural constraints in the query have an impact on retrieval effectiveness. Although both types of queries may be used for each task, mixing runs with both query types, also the best performing CAS query runs (restricted to topics containing a CAS query), and the best CO query runs will be reported. The use of CO/CAS query fields is recorded in submission format.

2 Result Submission

Fact sheet:

- For all three tasks, we allow up to 3 CO submissions, and up to 3 CAS submissions. That is, a participant can never submit more than 18 runs in total. For all runs, we allow element or passage results.
- All participants are invited to submit title-only runs (free choice between the CO title and the CAS title, and element or passage results) in a common submission format (details below), which allows up to 1,500 results per topic.
- There are additional requirements on the submissions the tasks:
 - FOCUSED TASK: for the same topic, results may not be overlapping.
 - RELEVANT IN CONTEXT TASK: articles may not be interleaved, and results may not be overlapping.
 - BEST IN CONTEXT TASK: only **one single** result per article is allowed.

Runs that violate these requirements in any way, will be disqualified.

2.1 INEX 2007 Topics

There is only one set of topics to be used for all adhoc retrieval tasks at INEX 2007. The format of the topics is defined in the following DTD:

```
<!ELEMENT inex_topic
  (title, castitle?, mmtitle?, description, narrative)>
<!ATTLIST inex_topic
  id      CDATA #REQUIRED
  ct_no   CDATA #REQUIRED
>
<!ELEMENT title      (#PCDATA)>
<!ELEMENT castitle   (#PCDATA)>
<!ELEMENT mmtitle    (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT narrative   (#PCDATA)>
```

¹Of course, any CO query can be directly rephrased as a CAS query `//*[about(., "CO query")]` using the tag wildcard `*` that matches any element.

The submission format will record the precise topic fields that are used in a run. Participants are allowed to use all fields, but only runs using either the `<title>`, `<castitle>`, or `<description>` fields, or a combination of these, will be regarded as truly *automatic*, since the additional fields will not be available in operational settings.

The `<title>` part of the INEX 2007 topics should be used as queries for the CO submissions. The `<mmtitle>` part of the INEX 2007 topics is dedicated for use in the INEX 2007 Multimedia Track. The `<castitle>` part of the INEX 2007 topics should be used as queries for the CAS submissions. In the number of runs allowed to be submitted, runs using more fields than the `<title>` (or `<castitle>`) will still be regarded as an CO (or CAS) submission.

Since the comparative analysis of CO and CAS queries is a main research question at INEX 2007, we encourage participants to submit runs using only the `<title>` field (CO query) or only the `<castitle>` field (CAS query). We do not outlaw the use of the other topic fields, to allow participants to conduct their own experiments involving them, and since such deviating runs may in fact improve the quality of the assessment pool.

2.2 Runs

For each of the three tasks, we allow up to 3 CO submissions, and up to 3 CAS submissions. The results of one run must be contained in one submission file (i.e. up to 18 files can be submitted in total). A submission may contain up to 1,500 retrieval results for each of the INEX topics included within that task.

There are however a number of additional task-specific requirements.

For the FOCUSED TASK, it is not allowed to retrieve elements or passages that contain text already retrieved by another result. For example, within the same article, the element `/article[1]/section[1]` is disjoint from `/article[1]/section[2]`, but overlapping with all ancestors (e.g., `/article[1]`) and all descendants (e.g., `/article[1]/section[1]/p[1]`).

For the RELEVANT IN CONTEXT TASK, articles may not be interleaved. That is, if a result from article `a` is retrieved, and then a result from a different article `b`, then it is not allowed to retrieve further results from article `a`. Additionally, it is not allowed to retrieve results than contain text already retrieved by another result (similar to the FOCUSED TASK). Note also that for this task the `/article[1]` result is implied by any result from the article, and need not be returned.

For the BEST IN CONTEXT TASK, only a single element or passage per article is allowed. To allow for a single submission format, we request a complete passage result, but the end-point will be ignored in the evaluation. The `/article[1]` element may be returned in case it is regarded as the best place to start reading, otherwise it is implied by any other result from this article.

2.3 Submission format

For relevance assessments and the evaluation of the results we require submission files to be in the format described in this section. The submission format for all tasks is defined in the following DTD:

```
<!ELEMENT inex-submission (topic-fields, description, collections, topic+)>
<!ATTLIST inex-submission
  participant-id CDATA #REQUIRED
  run-id         CDATA #REQUIRED
  task           (Focused | RelevantInContext | BestInContext ) #REQUIRED
  query         (automatic | manual) #REQUIRED
  result-type   (element | passage) #REQUIRED
>
<!ELEMENT topic-fields EMPTY>
<!ATTLIST topic-fields
  title         (yes|no) #REQUIRED
  mmtitle       (yes|no) #REQUIRED
  castitle      (yes|no) #REQUIRED
  description   (yes|no) #REQUIRED
  narrative     (yes|no) #REQUIRED
>
<!ELEMENT description (#PCDATA)>
<!ELEMENT topic (result*)>
```

```

<!ATTLIST topic topic-id CDATA #REQUIRED >
<!ELEMENT collections (collection+)>
<!ELEMENT collection (#PCDATA)>
<!ELEMENT result (in?, file, (path|passage), rank?, rsv?)>
<!ELEMENT in (#PCDATA)>
<!ELEMENT file(#PCDATA)>
<!ELEMENT path (#PCDATA)>
<!ELEMENT passage EMPTY>
<!ATTLIST passage
  start      (#PCDATA) #REQUIRED
  end        (#PCDATA) #REQUIRED
>
<!ELEMENT rank (#PCDATA)>
<!ELEMENT rsv (#PCDATA)>

```

Each submission must contain the participant ID of the submitting institute (available at the INEX website <http://inex.is.informatik.uni-duisburg.de/2007/ShowParticipants.html>), a run ID (which must be unique for the submissions sent from one organization – also please use meaningful names as much as possible), the identification of the task (e.g. Focused, etc), the identification of whether the query was constructed automatically or manually from the topic, and whether XML elements or passages have been retrieved. Furthermore, the used topic fields must be indicated in the `<topic-fields>` tag. Moreover, each submitted run must contain a description of the retrieval approach applied to generate the search results. A submission contains a number of topics, each identified by its topic ID (as provided with the topics).

For compatibility with the heterogeneous collection track, the `<collections>` tag is mandatory. There should be with `<collections>` at least one `<collection>` tag, which is by default set to "wikipedia" for the adhoc track. The `<in>` tag is optional for the adhoc track (`<in>` states from which collection each result comes from).

For each topic a maximum of 1,500 results may be included per task. A result element is described by a file name and an element path, and it may include rank and/or retrieval status value (rsv) information. For the adhoc retrieval task, `<collection>` is set to "wikipedia". Here is a sample submission file for the FOCUSED TASK:

```

<inex-submission participant-id="12" run-id="VSM_Aggr_06"
  task="Focused" query="automatic" result-type="element">
  <topic-fields title="no" castitle="yes" description="no"
    narrative="no"/>
  <description>Using VSM to compute RSV at leaf level combined with
    aggregation at retrieval time, assuming independence and using
    augmentation weight=0.6. Top-down removal of overlapping
    elements</description>
  <collections>
    <collection>wikipedia</collection>
  </collections>
  <topic topic-id="01">
    <result>
      <file>9996</file>
      <path>/article[1]/name[1]</path>
      <rsv>0.67</rsv>
    </result>
    <result>
      <file>9996</file>
      <path>/article[1]/body[1]/p[1]</path>
      <rsv>0.1</rsv>
    </result>
    [ ... ]
  </topic>
  <topic topic-id="02">
    [ ... ]
  </topic>
  [ ... ]
</inex-submission>

```

Rank and RSV The rank and rsv elements are provided for submissions based on a retrieval approach producing ranked output. The ranking of the result elements can be described in terms of:

- Rank values, which are consecutive natural numbers, starting with 1. Note that there can be more than one element per rank.
- Retrieval status values (RSVs), which are positive real numbers. Note that there may be several elements having the same RSV value.

Either of these methods may be used to describe the ranking within a submission. If both rank and rsv are given, the rank value is used for evaluation. These elements may be omitted from a submission if a retrieval approach does not produce ranked output. In case there is no complete ranking specified by the submission, the results are evaluated in arbitrary order.

File and path Since XML retrieval approaches may return arbitrary results from the documents of the INEX collection, we need a way to identify these nodes without ambiguity.

We will first explain XML element results, which are identified by means of a file name and an element (node) path specification, which must be given in XPath syntax. The file names in the Wikipedia collection uniquely define an article, so there is no need for including the directory in which the file resides (in contrast with the earlier IEEE collection). The extension .xml must be left out. Example:

```
<file>9996</file>
```

Element paths are given in XPath syntax. To be more precise, only fully specified paths are allowed, as described by the following grammar:

```
Path      ::= '/' ElementNode Path | '/' ElementNode | '/' AttributeNode
ElementNode ::= ElementName Index
AttributeNode ::= '@' AttributeName
Index      ::= '[' integer ']'
```

Example:

```
<path>/article[1]/body[1]/section[1]/p[1]</path>
```

This path identifies the element which can be found if we start at the document root, select the first "article" element, then within that, select the first "body" element, within which we select the first "section" element, and finally within that element we select the first "p" element. Important: XPath counts elements starting with 1 and takes into account the element type, e.g. if a section had a title and two paragraphs then their paths would be given as: `title[1]`, `p[1]` and `p[2]`.

A result element may then be identified unambiguously using the combination of its file name and element path. Example:

```
<result>
  <file>9996</file>
  <path>/article[1]/body[1]/section[1]/p[1]</path>
  <rsv>0.9999</rsv>
</result>
```

Passage paths are given in the same XPath syntax, but allow for an optional character-offset.

```
PassagePath ::= Path | Path '/text()' Index '.' Offset
Offset      ::= integer
```

The following example is effectively equivalent to the example element result above. Since we want to start and end at an element boundary, we can use the same path expressions as above.

```
<result>
  <file>9996</file>
  <passage start="/article[1]/body[1]/section[1]/p[1]"
    end="/article[1]/body[1]/section[1]/p[1]"/>
  <rsv>0.9999</rsv>
</result>
```

Note the the start attribute will refer to the beginning of an element (or its first content), and the end attribute will refer to the ending of an element (or its last content).

In the next example, however, the result starts at the second sentence of the paragraph and continues until a list item in list below the paragraph.

```
<result>
  <file>9996</file>
  <passage start="/article[1]/body[1]/section[1]/p[1]/text() [1].85"
    end="/article[1]/body[1]/section[1]/normallist[1]/item[2]/text() [2].106"/>
</result>
```

The offset can be placed anywhere in the text node starting from 0 (first character) to the *node-length* (last character).

2.4 Example: Identifying results

This section uses an example document to explain how results are to be identified.

XML We use a small excerpt from `12.xml`, a page on “Anarchism”:

```
<item>
<collectionlink xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"
  xlink:href="58198.xml">
Mikhail Bakunin
</collectionlink>,
<emph2>
<outsidelink xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"
  xlink:href="http://dwardmac.pitzer.edu/Anarchist_Archives/bakunin/">
God and the State
</outsidelink></emph2>,
<emph2>
The Paris Commune and the Idea of the State
</emph2>, others
</item>
```

This is in fact element `/article[1]/body[1]/section[4]/section[2]/normallist[1]/item[3]` which was relevant for INEX 2006 topic 306.

XML and whitespace XML is very flexible in its handling of whitespace, i.e., the following two documents are usually regarded as identical.

```
<a>
  <b/>
</a>
<a><b/></a>
```

However, strictly speaking the document on the left contains whitespace content (newlines, tabs, spaces) which is not present in the document on the right. That is, the element `<a>` in the document on the left contains first a newline and some spaces, then an empty `` element, and then again a newline.

The whitespace which is only inserted to make the XML file human readable has to be ignored, but white-space used to format the content of an element should be retained.

DOM tree We view the XML document using the Document Object Model (DOM), which results in a tree of root, element, attribute, text, and entity nodes like:

```
root
  |__element 'item'
    |__element 'collectionlink'
      |__attribute 'type' [...]
      |__attribute 'href' [...]
      |__text '\nMikhail Bakunin\n'
      |__text ', \n'
      |__element 'emph2'
```



```

|         |__element 'outsidelink'
|         |   |__attribute 'type' [...]
|         |   |__attribute 'href' [...]
|         |   |__text '\nGod and the State\n'
|__text ', \n'
|__element 'emph2'
|   |__text '\nThe Paris Commune and the Idea of the State\n'
|__text ', others \n'

```

In the DOM tree model, all content or text is in special text nodes.

Again, strictly speaking there would be an additional text node between the start tag of `<item>` and the start tag of `<collectionlink>` containing only newline whitespace. We completely ignore text nodes containing *only* whitespace (spaces, newlines, tabs).² As a result the `<item>` element has three text nodes as direct children. Note that all whitespace in other text nodes is preserved, including the newlines (indicated with `\n` in the tree).

Locating elements and text nodes We can use the XPath style location paths for locating elements. Since all content is in text nodes, passages start either on an element or inside a text node. For example, if we want our passage to start inside the first `<collectionlink>` we can locate the text node by `/item[1]/collectionlink[1]/text()[1]` and use any offset between 0 and 17 (the total character length including the newlines). For example, the last name Bakunin starts at offset 9, and the passage

```

<passage start="/item[1]/collectionlink[1]/text()[1].9"
end="/item[1]/collectionlink[1]/text()[1].16"/>

```

would precisely select the whole last name.

Mapping local offsets to global offsets The offsets used in the passages are all local to the text nodes. But since all textual content is in text nodes, we can use these to generate global offsets on the concatenation of all text nodes. Below we list all elements and text nodes with global start and end offsets for the example document:

```

/item[1] 0 97
/item[1]/collectionlink[1] 0 17
/item[1]/collectionlink[1]/text()[1] 0 17
/item[1]/text()[1] 17 20
/item[1]/emph2[1] 20 39
/item[1]/emph2[1]/outsidelink[1] 20 39
/item[1]/emph2[1]/outsidelink[1]/text()[1] 20 39
/item[1]/text()[2] 39 42
/item[1]/emph2[2] 42 87
/item[1]/emph2[2]/text()[1] 42 87
/item[1]/text()[3] 87 97

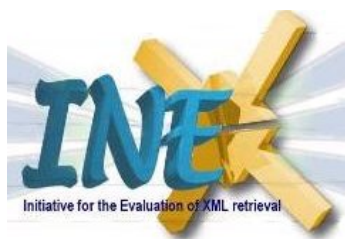
```

The length of (the content of) an element, in characters, is simply the end-offset minus the start-offset. For example, the `<item>` element contains 97 characters.

2.5 Result Submission Procedure

To submit a run, please use the following link: <http://inex.is.informatik.uni-duisburg.de/2007/> Then go to Tasks/Tracks, Adhoc, Submissions. The online submission tool will be available soon.

²Or in alternative terminology, these text nodes contain *element content whitespace*.



INEX 2007 Relevance Assessment Guide

Mounia Lalmas and Benjamin Piwowarski⁺

1. Introduction

During the retrieval runs, participating organisations evaluated the 130 INEX 2007 topics (CO+S) against the Wikipedia document collection and produced a list (or set) of document components (XML elements¹) as their retrieval results for each topic. The top 1500 components in a topic's retrieval results were then submitted to INEX. The submissions received from the different participating groups have now been pooled and redistributed to the participating groups (to the topic authors whenever possible) for relevance assessment. Note that the assessment of a given topic should not be regarded as a group task, but should be provided by one person only (e.g. by the topic author or the assigned assessor).

The aim of this guide is to outline the process of providing relevance assessments for the INEX 2007 test collection. This requires first a definition of relevance (Section 2), followed by details of how to assess (Section 3). Finally, we describe the on-line relevance assessment system that should be used to record your assessments (Section 4).

2. Relevance in INEX

Relevance in INEX is defined according to the notion of **specificity**, which describes the extent to which the document component focuses on the topic of request. This definition was adopted after a number of studies that showed that in terms of retrieval effectiveness, the same conclusions could be in most cases generated from using the specificity dimension of relevance compared to using more complex definitions. Up to INEX 2005, relevance was defined according to two dimensions, specificity and exhaustivity. The latter describes the extent to which the document component discusses the topic of request. This year (as for 2006), only the specificity dimension is used. Its measuring is based on the highlighting procedure used since INEX 2005. The main advantage of this highlighting approach is the specificity of any (partially highlighted) elements can be calculated automatically as some function of the contained relevant and irrelevant content (e.g. in the simplest case as the ratio of relevant content to all content, measured in number of words or characters).

3. How to assess

The assessment process is to be done as follows. Assessors highlight text fragments that contain only relevant information. It is important that only purely relevant information fragments get highlighted. To decide which text to highlight, you should skim-read the whole article and identify any relevant information as you go along. The on-line system can assist you in this task by highlighting keywords (that are chosen using the interface) and pool elements (elements retrieved by participating systems) within the article (see Section 5). If you highlight any part of a document, the document is considered relevant. For any such document, you should also select a so-called "best entry point" (BEP) of the document.

During the relevance assessment of a given topic, all parts of the topic specification should be consulted in the following order of priority: narrative, topic description, and topic title. The narrative should be treated **as the most authoritative description of the user's information need**, and hence it serves as the main point of reference against which relevance should be assessed. In case there is conflicting information between the narrative and other parts of a topic, the information contained in the narrative is decisive. *Note that it is not because that a term listed within the topic is not present in an element that the element is not relevant. Similarly, the presence of topic terms does not imply its relevance.* It may be that a component contains some or maybe all the terms, but is irrelevant to the

⁺Based on a prior guidelines authored by M. Lalmas, B. Piwowarski and G. Kazai

¹ The terms document component and XML element are used interchangeably.

topic of the request. Also, there may be components that contain none of the terms yet are relevant to the topic.

For the CO+S, the topic titles (may) contain structural constraints in the form of XPath expressions. These structural conditions should be ignored during your assessment. This means that you should assess the elements returned for a CO+S topic as whether they satisfy your information need (as specified by the topic) **with respect to the content criterion only**.

You should judge each text fragment on its own merit. That is, a text fragment is still relevant even if it is the twentieth you have seen with the same information. It is imperative that you maintain consistency in your judgement during assessment. Referring to the topic text from time to time will help you maintain judgement consistency.

4. Using the on-line assessment system (X-Rai)

There is an on-line relevance assessment system (XML Retrieval Assessment Interface) provided at:

<https://inex.lip6.fr/2007/adhoc>

which allows you to view the pooled result set of the topics assigned to you for assessment, to browse the Wikipedia document collection and to record your assessments. Use your INEX username and password to access this system.

The assessment tool works with opera and recent "gecko" browsers: we highly recommend you to use Opera (version 8 or up only; version 9 is recommended) available at <http://www.opera.com>. Other compatible browsers are:

- **Mozilla** (version 1.7 or up) at <http://www.mozilla.org/products/firefox/>.
- **Firefox** (version 1 and up) at <http://www.mozilla.org/products/mozilla1.x/>.

Note that **JavaScript must be enabled** for the assessment tool to work and that **the assessment tool is not compatible with Internet Explorer. Any bug report should be submitted using the project homepage (<https://developer.berlios.de/projects/x-rai/>) using the link in the "Links" menu of the interface (Figure 1).**

4.1. Home page

After logging in, you will be presented with the Home page (see Figure 1) listing the topic ID numbers of the topics assigned to you for assessment (under the title "Choose a pool"). This page can always be reached by clicking on the "**X-Rai**" link of the menu bar on any subsequent pages.




Each X-Rai page is composed of the following components:

- The menu bar, which is itself composed of four parts:
 1. The login name (e.g. "demo" in Figure 1),
 2. A list of menu items, which can be accessed by holding the mouse over the menu label (e.g. "**Links**" in Figure 1),
 3. The location within X-Rai, where each location step is a hyperlink (in Figure 1, we are at the root of the web site, so the only component of the location is "**X-Rai**", which is a link to the home page),
 4. The menu bar may also contain a number of icons (displayed on the right hand side, see Figure 2a). Click on one of these icons to display (or hide):



Information about X-Rai.

Toggle the help displayed when holding the mouse over icons or hyperlinks

- The main window.
- An optional status bar (see Figure 4), displayed only when assessing a pool, i.e. in pool, sub-collection or article view (see relevant sections below) appears at the bottom of the window and shows the number of unknown assessments you have to judge before completing assessing the document (in Figure 4, there is only one unknown assessment).
- In the status bar, three arrows (,  and ) may be used to navigate quickly between the elements to be assessed. You may also use the shortcut keys of 1 (left), 2 (up) and 3 (right). The up arrow enables you to move to a level up in the hierarchy, e.g. from an article or a collection part to its innermost enclosing part of the collection (you move in the opposite direction by

selecting a sub-collection or an article). The left arrow can be used to go to the previous element to be assessed, while the right arrow to go to the next element to be assessed.

The on-line assessment system provides three main views (Sections 4.2 to 4.4):

1. Pool view,
2. Sub-collection view, and
3. Article view

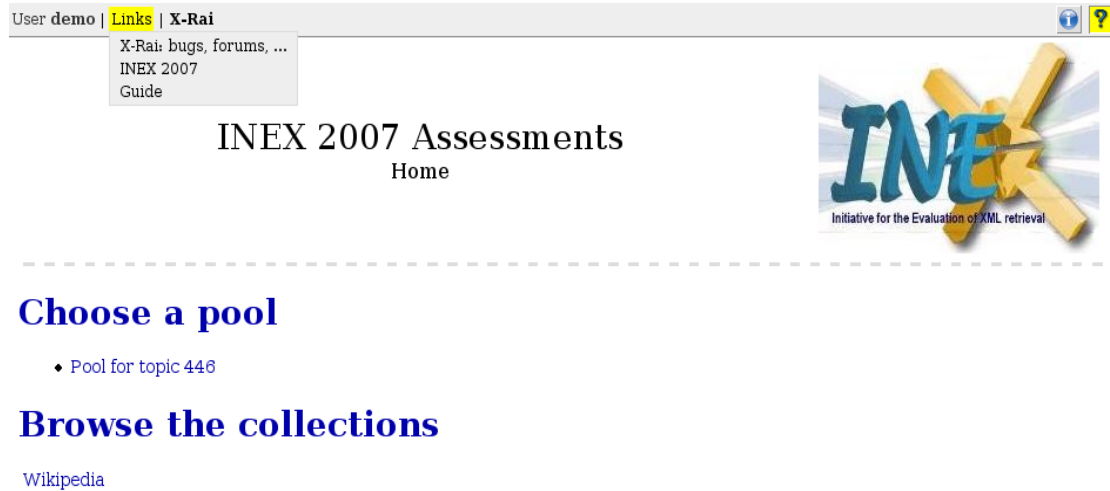


Figure 2: Home page and menu bar

In the “Links” menu

- **INEX 2007**: link to the official INEX web site.
- **X-Rai project**: link to the development web site of X-Rai where you can submit bug reports or/and feature requests.
- **Guide**: the latest version of this assessment guide.

4.2. Pool view

Clicking on a topic ID will display the Pool main page for that topic (see Figure 2a).

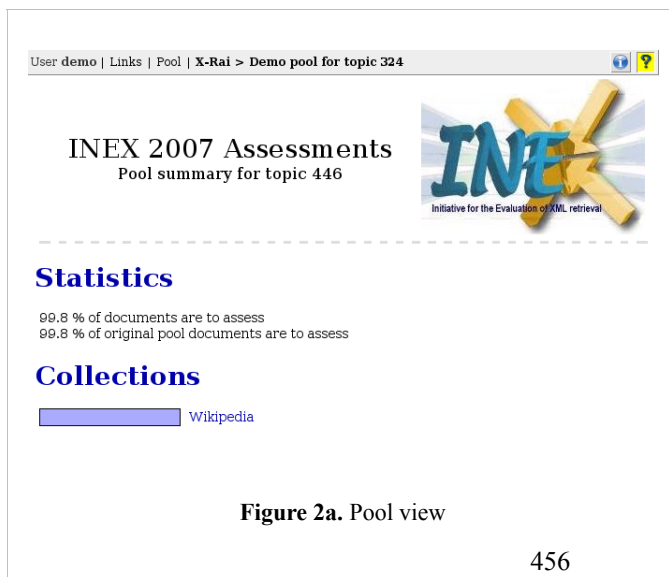


Figure 2a. Pool view

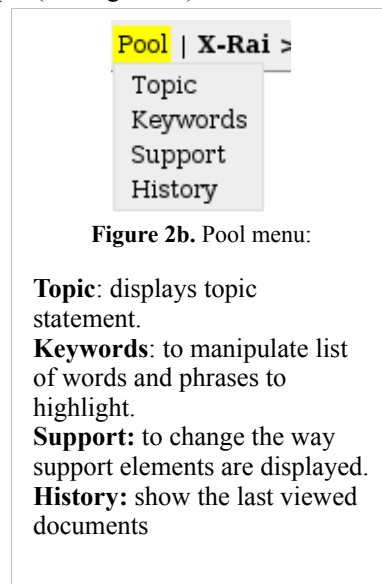


Figure 2b. Pool menu:

- Topic**: displays topic statement.
- Keywords**: to manipulate list of words and phrases to highlight.
- Support**: to change the way support elements are displayed.
- History**: show the last viewed documents

Here, a new menu item, “**Pool**”, appears on the menu bar at the top of the window. This menu item will remain whenever you are viewing a pool related page.

Within the “**Pool**” menu (Figure 2b), with the “**Topic**” submenu item you can display the topic statement in a popup window. This is useful as it allows you to refer to the topic text at any time during your assessment. An example of the topic popup window is given below:

Topic n°446 ()

title: +spanish chess players
castitle: //article[about(.. +spanish chess players)]
description: I would like to find articles about Spanish chess players
narrative: I like chess and want to collect as much information as possible about important Spanish chess players, in order to write a section of a more general report devoted to the development of chess in Spain. To this end, I want to identify the names of famous Spanish chess players (either born in Spain or becoming Spanish citizens) and also some biographical details. I am interested in both past and present chess players. To be relevant, an element should identify at least the name of a famous or important Spanish chess player, although I would prefer elements including also their biographical information and achievements. Articles devoted to chess in general or to chess players which are not from Spain are not relevant.

[Close window](#)

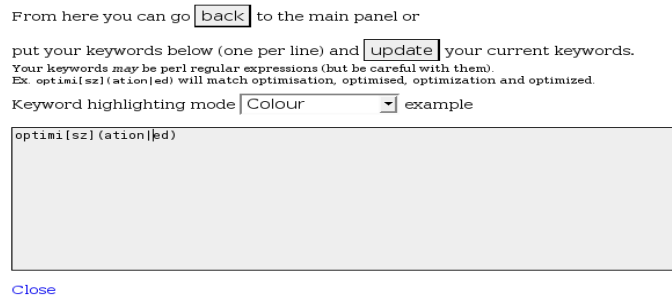
The “**Keywords**” submenu item allows you to access a feature, where you can specify a list of words or phrases to be highlighted when viewing the contents of an article during assessment. These cue words or phrases can help you in locating potentially relevant texts within an article and may aid you in speeding up your assessment (so add as many relevant cue words as you can think of!). You may edit, add to or delete from your list of keywords at any time during your assessment (remember, however, to refresh the currently assessed article to reflect the changes).

You may also specify the preferred highlighting colour for each and every keyword. After selecting the “**Keywords**” menu item, a popup window will appear showing a table of coloured cells. A border surrounding a cell signifies a colour that is already used for highlighting some keywords. Move the mouse over a coloured cell to display the list of keywords that will be highlighted in that colour. To edit the list of words or phrases for a given colour, click on the cell of your choice.

Choose a colour

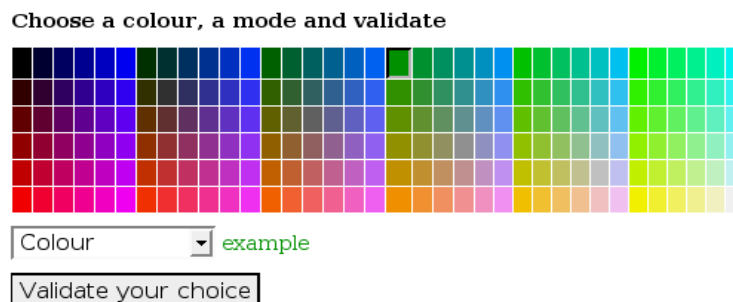
[Close](#)

You will be prompted to enter a list of words or phrases (one per line) to highlight (see figure below). You can choose three different highlighting modes using the drop-down menu: using coloured fonts, drawing a border around the phrase or using a background colour. You can use different highlighting modes with the same colour. To edit the list of words for a given mode, select the highlighting mode in the drop-down menu. You can then edit the list in the text area below. Note that the words or phrases you specify will be matched against the text in the assessed documents in their exact form, *i.e.* no stemming is performed. Your keywords *may* be perl regular expressions (but be careful with them). For example, `optimi[sz](ation|ed)` will match optimisation, optimised, optimization and optimized.



The **"Support"** item allows you to change the way support elements (i.e. elements retrieved by the participating systems) are displayed. When selecting this menu item, a pop-window (shown below) appears and allow you to change the colour (clicking on a colour) and the mode (background, font colour, or border, by selecting an item in the drop-down menu) of the highlighting. An example of the support element display is shown at the right of the mode selection.

Support element display



The **"History"** item allows you to access the list of last viewed documents, which can be useful if you want to go back to a wrongly assessed document. When selecting this menu item, a pop-up window appears and display the list of the last accessed documents, beginning by the last accessed. Icons show the status of the document:

- if the document is validated, a green mark is displayed. If the document is not validated and in the pool, a "highlighting" icon is displayed
- If the document belongs to the pool, a dashed blue box before the name of the document is displayed.
- If the document is relevant (contains highlighted passages), a plus sign is displayed; otherwise, a negative sign is displayed.

An example of the history popup window is shown below.



Under the title **"Collections"** is the list of collections to be assessed. In INEX 2007 (ad hoc task) there is only one such collection, the English Wikipedia collection.

The left or right arrows on the status bar move the focus to the previous or next collection, where there is at least one element to assess (since there is only one collection, no change will occur).

Clicking the hyperlink of “Wikipedia (English)” will take you into the sub-collection view.

4.3. Sub-collection view

The sub-collection views allow you to browse the different sub-collections within the Wikipedia collection. Sub-collections within Wikipedia are based on the alphabetical order, as depicted Figure 3. The first link of the page let you browse the Wikipedia sub-collection starting from "" to "Ali Baba...". This part will then be in turn divided into other sub-collections within the "" to "Ali Baba" range. Eventually, the last sub-collection view will contain a list of Wikipedia documents. Note that this view will show all articles within the collection, and not only those that need to be assessed.

For each possible sub-collection, there is an indication on the number of documents to be assessed in it (if this number is greater than 0), both for documents that were initially in the pool and for documents you chose to assess while browsing in the Wikipedia collection: You are free to assess more documents that there are in the pool, and it is advised to browse to documents that might contain relevant information if you can.

The left or right arrows on the status bar move the focus to the previous or next sub-collection, where there is at least one document to assess. You can also directly click on a link to a sub-collection.

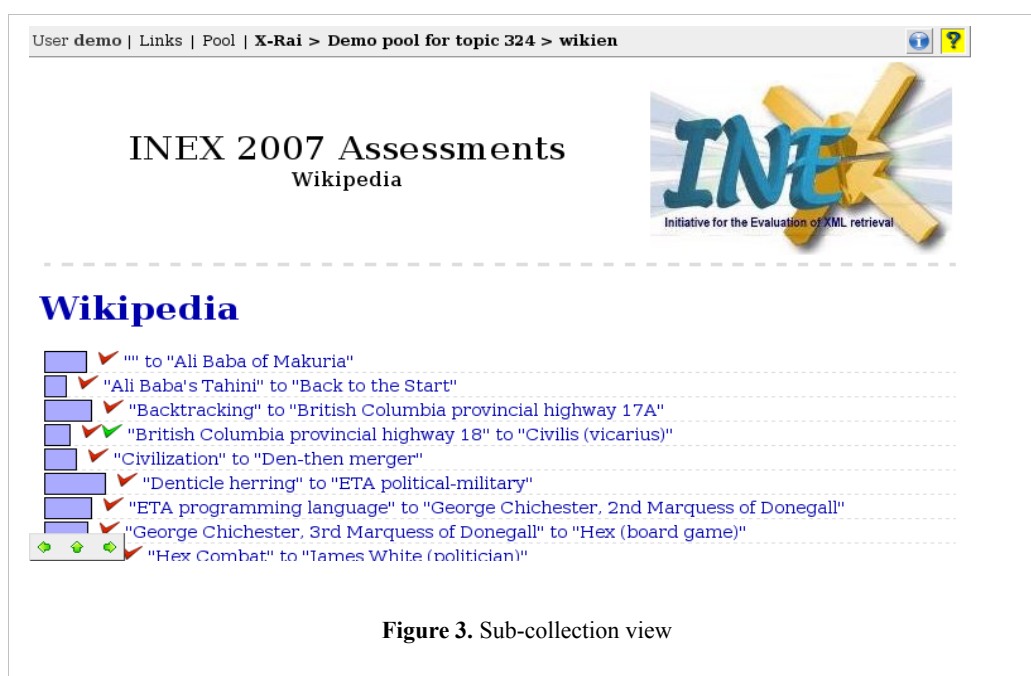


Figure 3. Sub-collection view

4.4. Article view

It is in this article view that elements can be assessed. The article view (see Figure 4) displays all the XML elements of an article together with their content. There are two types of objects within an article view: XML elements and passages. The latter are defined by the assessor while highlighting whereas the former are predefined by the XML file. **A highlighted passage in the interface has a yellow background.** Note that you should take care of not selecting colours for keyword highlighting too close to the colour X-Rai uses to mark highlighted passages.



Ali Baba

0 conversion warning(s)

Ali Baba (Arabic : علي بابا) is a fictional character described in the adventure tale of "**Ali Baba and the Forty Thieves**" which was added to the traditional collection of *The Book of One Thousand and One Nights* by its European transcriber, **Antoine Galland** , an 18th-century French orientalist who had heard it in oral form from a **Maronite** story-teller from **Aleppo** . This story has also been used as a popular pantomime plot.

Story Summary

Ali Baba, a poor woodcutter, happens to see and overhear a large band of thieves - **forty** in all - visiting their treasure store in the forest where he is cutting wood. The thieves' treasure is in a cave, the mouth of which is sealed by magic - it opens on the words "Open, Sesame", and seals itself on the words "Close, Sesame". When the thieves are gone, Ali Baba enters the cave himself, and takes some of the treasure home.

Ali Baba's rich brother, Kasim, finds out about his brother's unexpected wealth, and Ali Baba tells Kasim about the cave. Kasim goes to the cave to take more of the treasure, but forgets the magic words to get back out of the cave, and the thieves find him there, and kill him. When his brother does not come back, Ali Baba goes to the cave to look for him, and finds the body, bringing it home.



girl in Kasim's household, they are able to give Kasim a ons about his death.

Figure 4. Article view

Highlighting

During the highlight phase, you should identify only relevant (i.e. totally specific) passages by highlighting them. Passages can span over XML element boundaries. The passage limits are predefined by a pre-processing of XML files and correspond "more or less" to sentence boundaries. A consequence of this is that you should highlight the smallest passage that encloses the only relevant information if the predefined boundaries do not correspond exactly to the totally specific fragment. Another consequence is that, it is not necessary to highlight from the first character to the last one – which might be impractical in some case. For example, to highlight the previous sentence, you could start highlighting at the "o" of another and end on the "a" of case.

To highlight a passage, select it with the mouse as you would do in any word processor or text editor, and click on the square with the yellow background (or press "h").

If you make an error, you can unhighlight it by selecting the non relevant passage and clicking on the square with the white background (or press "u"). If you highlighted too much text, it is easier to unhighlight only the non relevant part.

Note that adjacent passages are merged together: You can highlight large regions of text in more than one step if this is more convenient.



Figure 5. Status bar (article view only)

The disk icon (here disabled): saving your assessments

The disk icon with the left (respectively right) arrow: save (if necessary) and goes to the previous (respectively next) document to assess.

The up arrow allows you to go up to the sub-collection view.

The eye (if applicable): shows or hides the pool elements

The target is used to set the BEP. The stroked target is used to remove the current BEP (if it is already defined for the document).


The mark reflects the status of the document: completely assessed and validated (green), completely assessed but not validated (red), and not completely assessed and not validated (grey). You can validate a document (i.e., mark it as finished) only if the mark is red.

The yellow/white square permits to (un)highlight the selected passage.


The clipboard shows the boundaries of the currently selected passage (as a couple of XPath expressions). This can be useful e.g. to submit bug reports.

Best Entry Point

Focussed structured document retrieval employs the concept of best entry point (BEP), which is intended to provide optimal starting-point from which users can browse to relevant document components. In INEX, you are requested to indicate one and only one BEP for every document that that has relevant content (that has highlighted passages). No BEP should be defined if the document is not relevant (i.e. does not contain any highlighted passage).



To set the BEP within a document (i.e. to be in the BEP mode), click on the  button (or press b) and then click on the position that you want to set as the BEP of that document. It is not possible to set the BEP at an arbitrarily position within the document. The same constraints to those used for highlighting apply for the BEP. In order to help you to know where the BEP will be located, when the mouse pointer is over a Wikipedia text and that you clicked on the "target button", the BEP symbol should appear at the position it would be set if you have clicked. Also note that there are one and one only BEP per relevant document.

Note that although you can set the BEP at any moment, we recommend that you first highlight and then set the BEP.

To remove any previously set BEP, simply click on  (or press shift+b).

Validation of assessments

When you have finished to assess an article, you have to validate it: By validating an article, you guarantee that *every* relevant passage of the displayed file has been highlighted. A BEP has also to be set before you can validate the file, unless the article does not contain any relevant information. In this latter case, setting a BEP would not make sense.

The status of a document is displayed in the status bar when viewing it (or in front of its name in the sub-collection views). The red mark  means the article is not validated, while the green mark  means that it has been validated. You can change its validation status only when viewing it, by pressing the key f or by clicking on the mark. Note that an article is automatically reset to the non validated state whenever you highlight, unhighlight or change the BEP.

Please keep in mind that an article is assessed if and only if it has been validated and saved (see below).

4.6. Saving your assessments















The assessment tool this year does not automatically save the assessments, but you NEED TO SAVE YOUR RELEVANCE ASSESSMENTS by clicking on the disk icon:

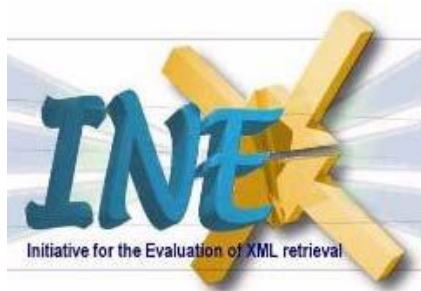


The icon is disabled (grey shade) when all assessments are saved.

Be warned that Opera does not provide a way to prevent from exiting a page without saving assessments. PLEASE ONLY USE THE X-RAI INTERFACE TO NAVIGATE INTO THE SITE as this is the only way to prevent you from leaving a page with non-saved assessment(s).

X-Rai reference card

<i>Icon</i>	<i>Shortcut</i>	<i>Action description</i>
All views within a pool		
	1	Highlight the previous (sub)collection or document to assess.
	2	Go to the container (sub-collection for an article, etc.)
	3	Highlight the next (sub)collection or document to assess
Article view		
	control+s	Save the current assessment
	p	Hide the pool elements
	p	Show the pool elements
	b	Set the BEP
	shift+b	Remove the BEP
shift + 	9	Go to the previous article to assess.
shift + 	0	Go to the next element to assess.
Article view - assessing		
	h	Highlight the currently selected passage.
	u	Unhighlight the currently selected passage
	f	Mark the article as finished
	f	Mark the article as not finished



INEX 2007

Book Search Track

Topic Development Guidelines v3.0

Gabriella Kazai

1 Introduction

The goal of the Book Search track is to investigate book-specific relevance ranking strategies for subject search tasks, UI issues and user behaviour, exploiting book-specific features, such as back of book indexes provided by authors, and associated metadata like library catalogue information. In order to provide the means for the evaluation of ranking strategies, the track aims to build a test collection.

Test collections, as traditionally used in information retrieval (IR) evaluation, comprise a set of documents, a set of information needs called topics, and a set of relevance assessments (i.e. the set of relevant documents for each topic).

For BookSearch'07, the set of documents is a collection of over 42,000 books, marked up in XML and complemented with metadata information in MACHine-Readable Cataloging (MARC) format. Unlike in traditional IR test collections, where documents are taken as atomic units of retrieval, the books are considered as structured texts, where any part of a book can be an answer to the user's query.

The topics, which are representations of users' information needs, will be created by participating organisations. For this year, topics will be limited to deal with content only aspects (i.e. no structural conditions). The structure of books, however, can still be used by search engines to improve their ranking of book parts estimated relevant to the query.

Relevance judgements will be provided at a later stage by the topic authors. Assessors will be asked to highlight relevant texts inside books as well as to provide an overall score for each relevant book.

As both topic creation and relevance judgements are results of collaborative effort, each participating organisation plays a vital role in the building of the test collection. The quality of the resulting test collection will determine its usefulness as a platform for evaluation both in the short and long term future.

This document deals only with topics: It provides guidelines on how to create topics.

2 Requirements and deadline

Each participating organisation is asked to create **at least 5 topics**, which should be submitted **by the deadline given on the INEX BookSearch'07 website at:**

<http://inex.is.informatik.uni-duisburg.de/2007/bookSearch.html>.

Topics can be created from scratch or participants are invited to choose from a set of sample queries selected from the query log of Microsoft Live Book Search to create topics around these (see Appendix A). Note that using these queries does not mean that the topic title should be fixed to contain these (and/or only these) query terms: A query can be expanded/modified, broadened or narrowed.

3 User task and topics

People may search a collection of books for a variety of reasons. They may be looking for a book to buy or could be looking for information that is contained in books. BookSearch'07 focuses on the latter task. The assumption is that searchers are seeking information on a certain subject, as they may, for example, be writing an essay on the subject. Their information need is expressed in the form of a topic.

4 Topic Creation Criteria

Since the performance of retrieval systems varies largely for different topics, to judge whether one retrieval strategy is (in general) more effective than another, the retrieval performance must be averaged over a large and diverse set of topics. In addition, to be a useful diagnostic tool, the performance of the retrieval systems on the topics can be neither too good nor too bad as little can be learnt about retrieval strategies if systems retrieve no, or only relevant, documents.

Therefore, when creating topics, a number of factors should be taken into consideration:

- Topics should cover subjects for which the topic author will be able to provide relevance judgements
- Topics should reflect real information needs
- Topics should represent the type of service an operational system might provide
- Topics should be diverse and differ in their coverage (e.g. broad or narrow).

5 Topic Format

Topics are made up of several parts, each of which explain the *same information need*, but for different purposes and at different levels of detail.

<title> Represents the search query that will be used by the search engines. It serves as a summary of the content of the user's information need.

<description> A natural language definition of the information need.

<narrative> A detailed explanation of the information need and a description of what makes an element relevant or not. The narrative must be a clear and precise description of the information need in order to unambiguously determine whether or not a given text fragment in a book fulfils the need. The narrative is taken as the only true and accurate interpretation of the user's needs. Relevance assessments will be made on compliance to the narrative alone.

Precise recording of the narrative is important for scientific repeatability. To aid this, the narrative should explain not only what information is being sought, but also the context and motivation of the information need, i.e., *why* the information is being sought and what work-task it might help to solve. The narrative, hence, should contain the following:

- **<task>** A description of the task for which information is sought, specifying the context, background and motivation for the information need
- **<infneed>** A detailed explanation of what information is sought and what is considered relevant or irrelevant

An example topic is given below.

<title>Octavius Antony Cleopatra conflict "Donations of Alexandria" "battle of Actium"**</title>**

<description>I am looking for information on the conflict between Octavius, Antony and Cleopatra. I am interested to learn about events like the Donations of Alexandria and the battle of Actium.**</description>**

<narrative>

<task>I am writing an essay on the relationship of Antony and Cleopatra and currently working on a chapter that will explore the conflict between Octavius, the brother of Antony's wife, Octavia, and the lovers. **</task>**

<infneed>Of interest is any information that details what motivated the conflict, how it developed and evolved through events such as the ceremony known as the Donations of Alexandria, Octavius' propaganda campaign in Rome against Antony, Antony's divorce from Octavia, and the battle of Actium in 31BC. Any information on the actions and emotions of the lovers during this period is relevant. Any non-documentary or non-biographical information, such as theatre plays (e.g. Shakespeare's play) or their critics are not relevant.**</infneed>**

</narrative>

6 Procedure for Topic Development

Submission is done by filling in the Candidate Topic Submission Form on the INEX web site at <http://inex.is.informatik.uni-duisburg.de/2007/> under Tasks/Tracks >> BookSearch >> Topics.

Please follow the steps below to create your topics. You can use a printout of the online Candidate Topic Form (see also Appendix B) to record all information about the topic you are creating.

Step 1: Initial Topic Statement

Create a one or two sentence description of the information you are seeking. This should be a simple description of the information need without regard to retrieval system capabilities or document collection peculiarities. This should be recorded in the Initial Topic Statement field. Record also the context and motivation of the information need, i.e. why the information is being sought. Add to this a description of the work-task, that is, with what task it is to help (e.g. writing an essay on a given topic).

Step 2: Corpus Exploration

Explore the book collection using queries you generate based on your initial topic statement in order to obtain an estimate of the number of relevant books and the number of relevant pages within these books, and to evaluate whether this topic can be judged consistently. You may use any retrieval engine for this task, including your own (on the test corpus) or the system provided through the INEX Book Search website.

You should try a number of different queries and make a record of any relevant books you find on the Candidate Topic Submission Form. For each relevant book you find, record its book id and how many pages¹ within the book contain relevant information; list up to 10 of these pages². Also write down the query that has lead you to finding the book.

To assess the relevance of a book or a page of a book use the following working definition: Mark it relevant if it would be useful if you were writing a report on the subject of the topic, or if it contributes toward satisfying your information need. Note that each result should be judged on its own merits. That is, information is still relevant even if it is the second time you have seen the same information. It is important that your judgment of relevance is consistent throughout this task.

If you find less than 3 or more than 20 relevant books, or in total more than 1000 relevant pages you should abandon the topic and start the process with a new topic.

Step 3: Topic Narrative

By now you should have a clear idea of what information is available within the book corpus and what information you consider relevant or not for your chosen topic. It is important to record this knowledge in detail as the narrative of the topic. Record not only what information is being sought, but also what makes it relevant or irrelevant. Also record the context and motivation of the information need. Include the work-task, i.e. how would the found information be used (e.g. written report). Make sure your description is exhaustive for two reasons: 1) you may not remember all of it several months later when you need to provide relevance assessments, 2) this information will be important for others to understand your information need.

Step 4: Topic Title and Description

During the exploration phase your initial topic statement has likely evolved. Record in natural language the final version of your topic statement in the topic description. Compose the topic title by recording the query words that you would use to search for information on your topic. Ensure that the information need as expressed in the title is also expressed in the description.

¹ You can use approximates. However, statements such as “most of the book is relevant” should be avoided. Please use more quantitative information instead, e.g. about 1/3rd of the 350 page book is relevant.

² Use either bookid + xpath if you are using your own system or the page number if you are using the system provided by the organizers.

Step 5: Finalization

Finalize the topic title, description and narrative. It is important that these parts all express the same information need; it should be possible to use each part of a topic in a stand-alone fashion.

Step 6: Topic Submission

To submit your topic, fill out the online Candidate Topic Submission Form on the INEX website <http://inex.is.informatik.uni-duisburg.de/2007/> under Tasks/Tracks >> BookSearch >> Topics.

After submitting a topic you will be asked to fill out an online questionnaire (this should take no longer than 5 minutes). It is important that this is done as part of the topic submission as the questions relate to the individual topic just submitted and the submission process. This is part of an effort to collect more context for the INEX topics, thereby increasing the reusability of the test collection. Initial results demonstrating the applicability of this can be found in [1].

7 Topic Selection

From the received candidate topics, the organizers will decide which topics to include in the final set. This is done to ensure inclusion of a broad set of topics. The data obtained from the collection exploration phase is used as part of the topic selection process. The final set of topics will be distributed for use in retrieval and evaluation.

Acknowledgments

Many thanks to Antoine Doucet, Stephen Robertson, Natasa Milic-Frayling, and Vishwa Vinay for their comments. This document is loosely based on the topic development guides from previous INEX workshops additionally authored by Börkur Sigurbjörnsson, Shlomo Geva, Mounia Lalmas, and Saadia Malik.

References

[1] Kamps, J. and Larsen, B. (2006). Understanding Differences between Search Requests in XML Element Retrieval. In *Proceedings of the SIGIR 2006 Workshop on XML Element Retrieval Methodology*, p. 13-19. [<http://www.cs.otago.ac.nz/sigirmw/>]

Appendix A: Sample base queries

Participants are invited to choose from the following set of queries, picked from the query logs of Microsoft Live Book Search, to create topics around them. Note that using these queries does not mean that the topic title should be fixed to contain these (and/or only these) query terms.

Alexandria	Immigrants	philosophers
Algebra	Immigration	Plant Genetics
American Colonies	Indian Myths	Prison Camps
American Presidents	Indian Summer	Psalms
Americus Vesputius	Indians	Pythagoras
Anabaptists	Israel	Quakers
Ancient History of the Earth	Jewish Theology	Queen Elizabeth
Atheism	Judaism	Qur'an
Battle of Bull Run	Juggernaut	Ralph Waldo Emerson
Buddha	Last Supper	Rationalism
Calculus	Laws of Nature	Reformation
California	Life of Shakespeare	Reminiscences
Catholic Church	Life of Ulysses S. Grant	Republicanism
Catholicism	Logic	Sabbath Hours
Charlemagne	Major religions of the world	Saint Boniface
Christianity	Meditation	Sermons
Church of England	Metaphysical Philosophy	Sidney Lanier
Church of Scotland	Metaphysics	Sikh History
Confederate States	Michelangelo	Socratic method
Confucius	Middle Ages	Spiritual Experience of St. Paul
Crusades	Military	Symphony
Descartes	Missionaries	The Algebra of Logic
Dreams	Mississippi River	The Fall of Mexico
Emancipation Proclamation	Moravian Church	The Sistine Chapel
Eucharist	Motherhood	The War in Cuba
European Settlements in America	Muhammad	The Works of Daniel Webster
Geometry	Mysticism	Time of Nero
Great Rebellion	Myths Every Child Should Know	Treaty of Washington
Hebrew culture	Napoleon	Vedanta Philosophy
Hebrew History	New Hampshire	Vedas
History of America	Niagara Falls	War for Independence
History of Mexico	North Pacific Coast	Wind and Weather
History of the Early Church	Ohio River	Wine Industry
History of the Southwest	Old Testament	Works of Aristotle
Holy Scriptures	Paganism	
	Passover	

Appendix B: Candidate Topic Form

Candidate Topic

Step 1: Initial Topic Statement

Initial Topic Statement:

* Step 2: Book Corpus Exploration Phase

Total number of relevant books found:

Total number of relevant book pages found:

Query and found relevant books. For each relevant book, record book id + XPath, or book id + page numbers. List up to 10 relevant pages or XML elements per book. You can use simple text format or XML (see Appendix C for examples).

<collectionexplorationresults>

</collectionexplorationresults>

*** Step 3: Finalised Topic**

<title>

</title>

<description>

</description>

<narrative>

<task>

</task>

<infneed>

</infneed>

</narrative>

Appendix C

Examples of recording relevant books found for different query formulations on a given topic during the collection exploration phase. Any of these formats can be used to record information for Step 2 in the Candidate Topic Form.

Example 1: Using natural language

Query: Octavius Antony Cleopatra</query>

Relevant books:

BookId: c8ft44gff765kjh6f

Relevant content: Approximately 65 pages out of a total of 120 pages are relevant

Relevant pages: 3, 6, 7, 9, 23, 27, 29, 35, 46, 47

BookId: f35tfnkolp903e4w

Relevant content: 5 pages out of a total of 43 pages are relevant

Relevant pages: 33,34, 36, 37, 38

Example 2: Using XML and BookId + page numbers

```
<queryresult>
```

```
<qurey>Octavius Antony conflict</query>
```

```
<books>
```

```
<book>
```

```
<bookid>a65hh9hv53fk0p</bookid>
```

```
<relinfo>The whole of chapter 6 is relevant: pages 120-145 out of 248 pages</relinfo>
```

```
<relpages>120, 123, 124, 129, 130, 131, 134, 135, 136, 144</relpages>
```

```
</book>
```

```
<book>....</book>....
```

```
</books>
```

```
</queryresult>
```

Example 3: Using XML and BookId + XPath

```
<queryresult>
```

```
<qurey>battle of Actium</query>
```

```
<books>
```

```
<book>
```

```
<bookid>bbv657gfr43y9ii0w</bookid>
```

```
<relinfo>Approximately 90 pages out of 342 are relevant: chapters 2, 6 and 7</relinfo>
```

```
<relpages>DjVuXML[1]/BODY[1]/OBJECT[2], DjVuXML[1]/BODY[1]/OBJECT[2],  
DjVuXML[1]/BODY[1]/OBJECT[12], DjVuXML[1]/BODY[1]/OBJECT[34], DjVuXML[1]/BODY[1]/OBJECT[44],  
DjVuXML[1]/BODY[1]/OBJECT[45], DjVuXML[1]/BODY[1]/OBJECT[46], DjVuXML[1]/BODY[1]/OBJECT[134],  
DjVuXML[1]/BODY[1]/OBJECT[135], DjVuXML[1]/BODY[1]/OBJECT[136]</relpages>
```

```
</book>
```

```
<book>....</book>....
```

```
</books>
```

```
</queryresult>
```



INEX 2007

Book Search Track

Tasks and Submission Guidelines

Gabriella Kazai

Version 5, October 23, 2007

1 Goals

The overall goal of the Book Search track is to investigate book-specific relevance ranking strategies for subject search tasks, user interface issues and user behaviour, exploiting book-specific features, such as back of book indexes provided by authors, and associated metadata like library catalogue information.

To work towards this goal, BookSearch'07 aims to explore aspects of the problem space around the indexing, retrieval and presentation of books in order to guide system developers and chart directions for future research. Individual tasks are formed around research questions that participants are invited to explore and report on. The tasks vary in their complexity, resource requirements and the type of output expected. They are described in the following sections:

- Section 2: Taxonomy of user intent task
- Section 3: Book classification (tagging) task
- Section 4: Book retrieval task
- Section 5: Book page in context retrieval task
- Section 6: Open task

Participants may participate in any of the tasks.

2 Taxonomy of user intent for book search

User intent is a critical component in the understanding of users' search behaviour. It defines what kinds of search tasks users engage in. In traditional information retrieval a user's intent is assumed to be of informational in nature: It is driven by the user's need for information in order to complete a task at hand. Observations of Web use resulted in further two categories: navigational and transactional. It is clear that these can also be applied to the book domain. However, it is possible that there are additional classes of user intent which are specific to books. It may also be the case that user tasks and user behaviour in the book domain will have specific traits and characteristics. What are the possible classes of user intent and user tasks and what properties they have is a research question that this task aims to explore.

The goal of this task is to derive a taxonomy of user intent with its associated properties and search tasks. The use of samples of users' (actual or hypothetical) information needs demonstrating each class of intent and task is encouraged. The taxonomy can extend to include both research and design questions and possible answers regarding how a given user behaviour may be supported by a search system and its user interface. For example, a user hoping to buy a book as a present is likely to be more interested in a system function that compares retail prices, while an informational query will more likely benefit from a "find related books" feature.

Examples of questions that can be explored include: How is user intent dependent on the genre of books? Which book specific features best support which kind of intent and task? How intent could be extracted from query logs? How should one design experiments to allow for the identification of user intent from system logs? What data would enable the prediction of intent in order to aid users? What user behaviour follows from them?

Participation in this task involves the submission of a research or opinion paper detailing the proposed taxonomy. Participants may report findings from the analysis of collected user log data or provide recommendations for the design of user studies to help elicit such data. Selected papers will be published as part of the INEX proceedings.

3 Book classification (tagging)

In this task, systems are tested on their ability to assign the correct classification labels from the Library of Congress (LoC) classification scheme to the books of the test corpus based only on information available from the full text of the books. The distributed corpus of about 42,000 books serves as the training corpus for this task: The classification labels are given in the MACHine-Readable Cataloging (MARC) files. A test corpus containing 2 sets of 1,000 books will be made available five days before the submission deadline.

Participants may submit up to three runs per test set. Each run should contain all books of the test set, and for each book include a ranked list (or set) of assigned classification labels in the form of (BookId, {LoC Classifications}) pairs following the DTD below. Classification (tagging) accuracy will be measured using standard measures such as F1 and ranked based metrics.

The Library of Congress classification headings extracted from each book's MARC record is available on the INEX website under the Submission page of the Book Search track (<http://inex.is.informatik.uni-duisburg.de/2007/bs-protected/submissions.html>).

```
<!ELEMENT bs-submission (description, book+)>
<!ATTLIST bs-submission
  participant-id    CDATA      #REQUIRED
  run-id           CDATA      #REQUIRED
  task             (classification) #REQUIRED
>
<!ELEMENT description (#PCDATA)>
<!ELEMENT book (bookid, category+)>
<!ELEMENT category (label, rank?, rsv?)>
<!ELEMENT bookid   (#PCDATA)>
<!ELEMENT label    (#PCDATA)>
<!ELEMENT rank     (#PCDATA)>
<!ELEMENT rsv      (#PCDATA)>
```

Each submission must contain the following:

- The Participant ID number of the submitting institute (available at the INEX website <http://inex.is.informatik.uni-duisburg.de/2007/ShowParticipants.html>)
- A run ID (which must be unique for the submissions sent from one organization – also please use meaningful names as much as possible)
- Identification of the task (i.e. “classification”)
- A description of the classification approach applied to generate the results.

Furthermore, a submission should contain:

- All books in the test set, where each book should have a ranked list (with minimum length of 1) of assigned classification categories, ordered by decreasing value of relevance. Rank and rsv values for each category may also be recorded. Please note that the evaluation will rely on the ordering alone (values of the rank and rsv fields will be ignored). Note that the ordering of the books is of no concern here.

An example submission may be as follows:

```
<bs-submission participant-id="25" run-id="SVM-lambda1" task="classification">
<description>Used a simple SVM with no parameter tuning</description>
<book>
  <bookid>384d10daea4e34a8</bookid>
  <category><label>BL1-50</label><rank>1</rank></category>
  <category><label>BR83</label><rank>2</rank></category>
</book>
```

```

<book>...</book>
...
</bs-submission>

```

4 Book retrieval

The goal of this task is to investigate the impact of book specific features on the effectiveness of book retrieval systems, where the unit of retrieval is the (complete) book. Users are thus assumed to be searching for (whole) books relevant to their information need that they can, e.g., borrow from a Library or purchase from a retailer, etc.

Participants of this task are invited to submit pairs of runs, where one run should be the result of applying generic IR techniques to return a ranked list of books to the user in response to a query. The other run should be generated using the same techniques (where possible) but with the use of additional book-specific features (e.g. back-of-book index, citation statistics, in or out of print, etc.) or specifically tuned methods. In both cases, a result list should contain a maximum of 1000 books estimated **relevant** to the given topic, ranked in order of estimated relevance to the query.

The test queries used for this task have been extracted from the query log of a commercial search engine, and relevance judgements have been collected on a four point scale: Excellent, Good, Fair, and Not-relevant. The evaluation (subject to change) will be based on the measure of Normalised Discounted Cumulated Gain at various cut-off values.

Participants may submit up to 3 pairs of runs. The DTD describing the submission format is as follows:

```

<!ELEMENT bs-submission (topic-fields, description, topic+)>
<!ATTLIST bs-submission
participant-id      CDATA      #REQUIRED
run-id             CDATA      #REQUIRED
paired-run-id      CDATA      #REQUIRED
task               (book-retrieval) #REQUIRED
query              (automatic | manual) #REQUIRED
result-type        (book)          #REQUIRED
retrieval-type     (non-specific | book-specific) #REQUIRED
>
<!ELEMENT topic-fields EMPTY>
<!ATTLIST topic-fields
title              (yes|no) #REQUIRED
description        (yes|no) #REQUIRED
narrative          (yes|no) #REQUIRED
>
<!ELEMENT description (#PCDATA)>
<!ELEMENT topic (book*)>
<!ATTLIST topic topic-id CDATA #REQUIRED >
<!ELEMENT book (bookid, rank?, rsv?)>
<!ELEMENT bookid  (#PCDATA)>
<!ELEMENT rank    (#PCDATA)>
<!ELEMENT rsv     (#PCDATA)>

```

Each submission must contain the following:

- The Participant ID number of the submitting institute (available at the INEX website <http://inex.is.informatik.uni-duisburg.de/2007/ShowParticipants.html>)
- A run ID (which must be unique for the submissions sent from one organization – also please use meaningful names as much as possible)
- The run-id identifying the run that the current submission is paired with (i.e. if the current run is the book-specific ranking then the paired run-id is the generic ranking – these two runs will be compared with each other)
- Identification of the task, i.e. “book-retrieval”
- Identification of whether the search query was constructed automatically or manually from the topic
- The result-type, which should be set to “book”

- Identification of the retrieval-type, which can be either “non-specific” when generic IR techniques have been applied or “book-specific” when the IR techniques have been in some shape or form enhanced with book-specific features
- Specification of which topic fields were used in constructing the search query (i.e. title and/or description and/or narrative)
- A description of the retrieval approach applied to generate the results.

Furthermore, a submission should contain the search results for each test topic, confirming to the following criteria:

- The ranked list of books per topic can contain a maximum of 1000 books estimated **relevant** to the topic, ordered by decreasing value of relevance. Each book should be identified using its bookID, which is the name of the directory that contains the XML source of the book along with the MARC metadata file. The rank position and RSV value can be recorded for each book in the ranking. Please note, however, that the evaluation will rely on the actual ordering of results alone (values of the rank and rsv fields will be ignored).

An example submission may be as follows:

```
<bs-submission participant-id="25" run-id="BM25F-ToC-BackOfBookIndex-Streams" paired-run-id="BM25" task="book-retrieval" query="automatic" result-type="book" retrieval-type="book-specific">
  <topic-fields title="yes" description="no" narrative="no"/>
  <description>BM25F using 2 streams extracted from the table of contents and the back-of-book index sections</description>
  <topic topic-id="01">
    <book>
      <bookid>384d10daea4e34a8</bookid>
      <rank>1</rank>
    </book>
    <book>
      <bookid>cdc234f3f2858ba5</bookid>
      <rank>2</rank>
    </book>
    <book>...</book>
  ...
</topic>
<topic> ... </topic>
</bs-submission>
```

5 Book page in context retrieval task

Based on the assumption of an informational user request, the task of a book search system is to return the user a ranked list of books estimated relevant to the user need, and then present within each book, a ranking of relevant non-overlapping XML elements, passages or pages.

This task is based on topics created by the participants for which relevance judgements will be collected from participants in the phase following the result submissions. Evaluation measures will be selected and adopted from those used at the INEX ad hoc track (subject to change).

Participants may submit up to 10 runs. One automatic (title-only) and one manual runs are compulsory. Additional manual runs are encouraged in order to help the construction of a reliable test collection. Each run can contain for each topic a maximum of 1000 books estimated **relevant** to the given topic, ordered by decreasing value of relevance. For each book, a ranked list of non-overlapping XML element, passage or book page results estimated **relevant** should be listed in decreasing order of relevance. A minimum of 1 result per book must be returned. A submission can only contain one type of result, i.e. only book pages or only passages; result types cannot be mixed. Submissions should conform to the following DTD:

```
<!ELEMENT bs-submission (topic-fields, description, topic+)>
<!ATTLIST bs-submission
  participant-id CDATA #REQUIRED
  run-id CDATA #REQUIRED
  task (book-ad-hoc) #REQUIRED
```



```

query                (automatic | manual) #REQUIRED
result-type          (element | passage | page) #REQUIRED
>
<!ELEMENT topic-fields EMPTY>
<!ATTLIST topic-fields
title                (yes|no) #REQUIRED
description           (yes|no) #REQUIRED
narrative            (yes|no) #REQUIRED
>
<!ELEMENT description (#PCDATA)>
<!ELEMENT topic (book*)>
<!ATTLIST topic topic-id CDATA #REQUIRED >
<!ELEMENT book (bookid, rank?, rsv?, result+)>
<!ELEMENT result ((path|passage), rank?, rsv?)>
<!ELEMENT bookid (#PCDATA)>
<!ELEMENT path (#PCDATA)>
<!ELEMENT passage EMPTY>
<!ATTLIST passage
start                (#PCDATA) #REQUIRED
end                  (#PCDATA) #REQUIRED
>
<!ELEMENT rank (#PCDATA)>
<!ELEMENT rsv (#PCDATA)>

```

Each submission must contain the following:

- The Participant ID number of the submitting institute (available at the INEX website <http://inex.is.informatik.uni-duisburg.de/2007/ShowParticipants.html>)
- A run ID (which must be unique for the submissions sent from one organization – also please use meaningful names as much as possible)
- Identification of the task, i.e. “book-ad-hoc”
- Identification of whether the search query was constructed automatically or manually from the topic
- The result-type, which can be either “element”, “passage” or “page”. An element is an XML element of arbitrary granularity, given by its XPath (see Appendix A), that cannot overlap with any other retrieved element. A passage is an arbitrary sized passage, given by its start and end offset, that cannot overlap with any other retrieved passage. A page is an XML element of given granularity, given by its XPath.
- Specification of which topic fields were used in constructing the search query (i.e. title and/or description and/or narrative)
- A description of the retrieval approach applied to generate the results.

Furthermore, a submission should contain the search results for each test topic confirming to the following criteria:

- The ranked list of books per topic can contain a maximum of 1000 books estimated **relevant** to the topic, ordered by decreasing value of relevance. Each book should be identified using its bookID, which is the name of the directory that contains the XML source of the book along with the MARC metadata file. The rank position and/or RSV value can be recorded for each book in the ranking.
- For each book, a ranked list of non-overlapping XML elements, passages or book pages estimated **relevant** should be returned, identified by its XPath (in the case of XML element and page type results) or start and end offset (in the case of passage type results). For each result inside a book, its rank and/or RSV score can be recorded. For information on XPath, please see Appendix A.

Please note that the evaluation will rely on the rank order of the books and of the results inside books alone (values of the rank and rsv fields will be ignored).

An example submission may be as follows:

```

<bs-submission participant-id="25" run-id="BM25F-ToC-BackOfBookIndex-Streams" task="book-ad-hoc" query="automatic" result-type="page">
  <topic-fields title="yes" description="no" narrative="no"/>
  <description>BM25F using 2 streams extracted from the table of contents and the back-of-book index sections</description>

```

```

<topic topic-id="01">
  <book>
    <bookid>384d10daea4e34a8</bookid><rank>1</rank>
    <result><path>/DjVuXML[1]/BODY[1]/OBJECT[27]</path><rank>1</rank></result>
    <result><path>/DjVuXML[1]/BODY[1]/OBJECT[122]</path><rank>2</rank></result>
    <result><path>/DjVuXML[1]/BODY[1]/OBJECT[5]</path><rank>3</rank></result>
    ...
  </book>
  <book>
    <bookid>5afee130174076e3</bookid><rank>2</rank>
    <result><path>/DjVuXML[1]/BODY[1]/OBJECT[531]</path><rank>1</rank></result>
    <result><path>/DjVuXML[1]/BODY[1]/OBJECT[14]</path><rank>2</rank></result>
    ...
  </book>
  <book>...</book>
  ...
</topic>
<topic> ... </topic>
</bs-submission>

```

6 Open task

Participants are invited to carry out and evaluate their own tasks and/or submit task proposals discussing motivation, required infrastructure and potential benefits for running the task.

7 Submission procedure

An online submission tool will be provided at: <http://inex.is.informatik.uni-duisburg.de/2007/> a week before the submission deadline. Please note that currently there are no plans to provide online validation of submission runs, so please make sure that your runs conform to the appropriate DTD and that all XPath expressions (see Appendix A) are valid.

Acknowledgments

Many thanks to Stephen Robertson, Natasa Milic-Frayling, Vishwa Vinay, Nick Craswell, Miro Lehtonen, Andrew Trotman and Antoine Doucet for helpful comments.

Appendix A: XPath and Passages

XPath

XML element and book page paths should be given in XPath syntax¹. To be more precise, only fully specified paths are allowed, as described by the following grammar:

```
Path ::= '/' ElementNode Path | '/' ElementNode | '/' AttributeNode
ElementNode ::= ElementName Index
AttributeNode ::= '@' AttributeName
Index ::= '[' integer ']'
```

Example:

```
<path>/DjVuXML[1]/BODY[1]/OBJECT[1]</path>
```

This path identifies the XML element which can be found if we start at the document root, select the first “DjVuXML” element, then within that, select the first “BODY” element, within which we select the first “OBJECT” element.

Please note that XPath counts element nodes **starting with 1** and takes into account the element type. For example, if the BODY element had a title and two paragraphs then their XPathS would be given as: /DjVuXML[1]/BODY[1]/title[1], /DjVuXML[1]/BODY[1]/p[1] and /DjVuXML[1]/BODY[1]/p[2]. I.e. both the title and the first paragraph is numbered as 1 since they are different element types.

Passage

Passage paths are given in the same XPath syntax, but allow for an optional character-offset.

```
PassagePath ::= Path | Path '/text()' Index '.' Offset
Offset ::= integer
```

The example above can be given as the passage:

```
<passage start="/DjVuXML[1]/BODY[1]/OBJECT[1]/text()[1].0"
end="/DjVuXML[1]/BODY[1]/OBJECT[1]/text()[1].876"/>
```

The offset can be placed anywhere in the text node starting from 0 (first character) to the node-length (last character).

When a passage starts and ends on an element boundary it can be written without the optional character offset. In this case, it is assumed that the passage starts on the first character (i.e. character 0) of the XML element given in the start attribute and ends on the last character of the XML element given in the end attribute:

```
<passage start="/DjVuXML[1]/BODY[1]/OBJECT[1]" end="/DjVuXML[1]/BODY[1]/OBJECT[1]"/>
```

XML and whitespace

XML is very flexible in its handling of whitespace, i.e., the following two documents are usually regarded as identical.

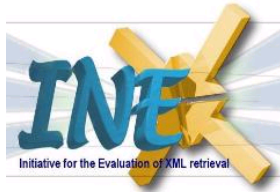
```
<a>
  <b />
</a>
```

```
<a><b/></a>
```

¹ Clark, J. and DeRose, S. 1999. XML Path Language (XPath) version 1.0. W3C Recommendation. <http://www.w3.org/TR/xpath>.

However, strictly speaking the document on the left contains whitespace content (newlines, tabs, spaces) which is not present in the document on the right. That is, the element <a> in the document on the left contains first a newline and some spaces, then an empty element, and then again a newline.

When constructing passages, any whitespace that represents the only textual content of a text node should be ignored.



INEX 2007 Entity Ranking Track Guidelines - V1

Arjen P. de Vries, James A. Thom, Anne-Marie Vercoustre, Nick Craswell and Mounia Lalmas

Monday, 06 August, 2007

This document contains the guideline information for the Entity Ranking track at INEX 2007. Some of the steps have not been finalized, and as such the document should be viewed as a draft. This version (V1) contains a new schedule, and guidelines regarding the topic creation step (i.e. the candidate entities). It also gives an overview of the tasks that will be investigated this year, and how they are likely to be measured.

1 Introduction

Search engines are ever more interested in returning *entities* instead of ‘just’ web pages. INEX has started a track called *Entity Ranking* to provide a forum where researchers may compare and evaluate techniques for engines that return lists of entities. The goal of the task is to evaluate how well systems can rank entities in response to a query; the set of entities to be ranked is assumed to be loosely defined either by a generic category or by some example entities.

The entity track concerns triples of type $\langle \text{category}, \text{query}, \text{entity} \rangle$. The category (that is *entity type*), specifies the type of ‘things’ to be retrieved. The query is a free text description that attempts to capture the information need. The entity specifies example instances of the entity type. The usual information retrieval tasks of document and element retrieval can be viewed as special instances of this more general retrieval problem, where the category membership relates to a syntactic (layout) notion of ‘text document’, or ‘XML element’. Expert finding uses the semantic notion of ‘people’ as its category, where the query would specify ‘expertise on T’ for finding expert on topic T.

For this year (at least) the goal is not to evaluate how well systems identify instances of entities within text (to some extent this is part of the goal of the *Link-the-Wiki* track¹).

2 Data

The track uses the Wikipedia XML data, where we exploit the category metadata about the pages to loosely define the entity sets. The category metadata is contained in the following files:

- `categories_name.csv` which maps category ids to category names
- `categories_hcategories.csv` which defines the category graph (which is not a strict hierarchy!)
- `categories_categories.csv` which maps article ids (that is pages that correspond to entities) to category ids

¹<http://inex.is.informatik.uni-duisburg.de/2007/linkwiki.html>

The entities in such a set are assumed to loosely correspond to those Wikipedia pages that are labeled with this category (or perhaps a sub-category of the given category). For example, consider the category ‘art museums and galleries’ (10855), an article about a particular museum such as the ‘Van Gogh Museum’ (155508) may be mapped to a sub-category such as category ‘art museums and galleries in the netherlands’ (36697).

Obviously, this is not perfect as many Wikipedia articles are assigned to categories in an inconsistent fashion. Your retrieval method should handle the situation that the category assignments to Wikipedia pages are not always consistent, and also far from complete. The human assessor will not be constrained by the category assignments made in the corpus when making his or her relevance assessments!

We expect that the data set provides a sufficiently useful collection as a starting point for the purpose of the track. The challenge for participants is to exploit the rich information from text, structure and links to perform this task.

3 Tasks

The track will investigate two tasks in 2007.

3.1 Entity Ranking

The motivation for our Entity Ranking task is to return entities that satisfy a topic described in natural language text. In other words, in the entity ranking task, the information need includes which category (entity type) is desired as answers. As such, a topic specifies the category identifier and the free-text query specification. Results consist of a list of Wikipedia pages (our assumption is that all entities have a corresponding page in Wikipedia).

For example, with ‘Art museums and galleries’ as the input category and a topic text ‘Impressionist art in the Netherlands’, we expect answers like the ‘Van Gogh museum’ and the ‘Kröller-Müller museum’. Of course, the entity type is only loosely defined by its category ‘art museums and galleries’, and correct answers may belong to other categories close to this category in the Wikipedia category graph, or may not have been categorized at all by the Wikipedia contributors.

An example of a topic for the Entity Ranking task is given below.

```
<inex_topic topic_id="9999" ct_no="0">
<title>Impressionist art in the Netherlands</title>
<description>
I want a list of art galleries and museums in the Netherlands
that have impressionist art.
</description>
<narrative>Each answer should be the article about a specific
art gallery or museum that contain impressionist or
post-impressionist art works.
</narrative>
<categories>
<category id="10855">art museums and galleries</category>
</categories>
</inex_topic>
```

The category name(s) should be the exact names used in the INEX version of the Wikipedia, but they should be loosely interpreted.

Again, systems should not assume that all the good entity answers actually belong explicitly to the target category. For example, if you are looking for explorers, not all the explorers will have been labelled with category ‘explorers’; some may have category label ‘explorers of australia’ or ‘explorers of the pacific’ (which are sub-categories of ‘explorers’). The category ‘explorers’ in the topic is only an indication of what is expected, not a strict constraint (like in the CAS title for the ad-hoc track). When the query is ‘find explorers/navigators who where looking for Australia’, we are expecting answers such as James Cook, Tasman, Lapérouse – who happen to be all classified as ‘explorers of australia’, but such consistency cannot always be expected. We cannot expect that a given category such as ‘explorers of australia’ contains all the relevant entity nor only the relevant ones (‘explorers of australia’ category contains many explorers who explored inland Australia).

3.2 List Completion

The List Completion task is a problem related to entity ranking. Instead of knowing the desired category (entity type), the topic specifies a number of correct entities (instances) together with the free-text context description. Results consist again of a list of entities (Wikipedia pages).

If we provide the system with the topic text and a number of entity examples, the task of List Completion refers to the problem of completing the partial list of answers. As an example, when ranking ‘Countries’ with topic text ‘European countries where I can pay with Euros’, and entity examples such as ‘France’, ‘Germany’, ‘Spain’, then the ‘Netherlands’ would be a correct completion, but the ‘United Kingdom’ would not.

```
<inex_topic topic_id="9999" ct_no="0">
<title>European countries where I can pay with Euros</title>
<description>
I want a list of European countries where I can pay with Euros.
</description>
<narrative>
Each answer should be the article about a specific European
country that uses the Euro as currency.
</narrative>
<entities>
<entity id="10581">France</entity>
<entity id="11867">Germany</entity>
<entity id="26667">Spain</entity>
</entities>
</inex_topic>
```

The entity names should be the exact names used in the INEX version of the Wikipedia. When creating a topic, the topic developer should specify between one and three example entities.

4 Topics

Participants are asked to create a small number (say 5) of (partial) entity lists with corresponding topic text. Candidate entities correspond to the names of articles that loosely belong to categories (for example may be subcategory) in the Wikipedia XML corpus.

The format for the candidate entities is as follows (the precise DTD will be given later):

```
<inex_topic topic_id="9999" ct_no="0">
<title>European countries where I can pay with Euros</title>
<description>
I want a list of European countries where I can pay with Euros.
</description>
<narrative>
Each answer should be the article about a specific European
country that uses the Euro as currency.
</narrative>
<entities>
<entity id="10581">France</entity>
<entity id="11867">Germany</entity>
<entity id="26667">Spain</entity>
</entities>
<categories>
<category id="61">countries</category>
</categories>
</inex_topic>
```

The result submissions for task 1 (Entity Ranking) can use the `title`, `categories`, and `category_ids` elements and the result submissions for task 2 (List Completion) can use the `title`, `entities`, and `entity` elements. More details about the tasks will be given later. Participants are however welcome to provide feedback on the description of the tasks (Section 3).

The submission site for the candidate entities will be open shortly. There will be an email announcement with the exact submission details.

5 Evaluation

Participants judge their own submitted (and accepted²) topics. Because we make the assumption that an entity corresponds to a Wikipedia page, the answer pool correspond to a list of links into the collection. We assume that the nature of the task is such that it is feasible to assess answer correctness quickly. Quite often, the title of the page should be enough for a topic author to judge the answer's relevance, and judging can be quick.

We plan to also experiment with a form of participant judging based on voting. In this case, each participant gets assigned a subset of the topics, and they vote on the correctness of the pooled answers. The answers that get a sufficient number of votes will be assumed the correct ones. Again, the answer pool corresponds to a list of links into the collection. This list will be ordered based on their frequency in the pool. Each

²The selection of the topics will be done by the organizers of the track.

participant will be asked to go as deep into the list as they can in a fixed amount of time. This voting procedure is designed for optimal assessment efficiency. We will analyse the data obtained for their accuracy, by comparing to the topic author's assessments.

Traditional evaluation measures (such as MAP and average R-precision) will be used to measure performance of systems on both tasks.

For the list completion task, there is no need to return the entity instances given as examples in the topic description.

6 Training topics

A small training set of 27 topics (based on a selection of 2006 ad hoc adapted to the entity task) developed at INRIA will be made available for participants to develop and train their systems. The training topics will be in the format required by this track and can be used for training for both tasks. Relevance assessments derived from the articles judged relevant in 2006, limited to the loosely defined categories, will also be made available.

7 Proposed (revised) Schedule

Aug 7: This document released which provides participants with detailed instructions and formatting criteria for candidate "topic entities" as well as preliminary guidelines on the ranking tasks.

Aug 27: Training collection based on a subset of 2006 topics to be released.

Sep 10: Submission deadline for candidate "topic entities".

Sep 24: Distribution of final set of "topic entities" to participants along with detailed information on the formatting requirements for the entity ranking results.

Oct 13: Submission deadline of entity ranking results.

Oct 22: Distribution of results to participants for assessments.

Nov 4: Submission deadline for assessments.

Nov 16: Distribution of assessments and evaluation scores to participants.

Nov 26: Submission of papers for the workshop pre-proceedings.

Dec 7: Workshop pre-proceedings and workshop programme online.

Dec 17-19: Workshop in Schloss Dagstuhl. (<http://www.dagstuhl.de/>).

8 Future tasks

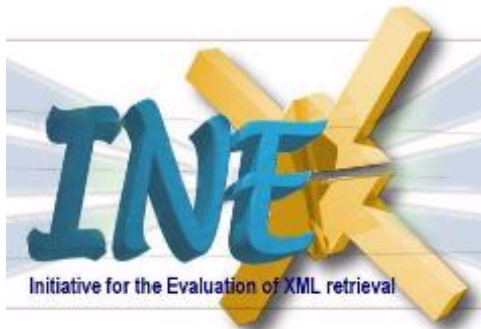
8.1 Associative Ranking

A possible future task for this track is Associative Ranking. As a pilot for 2007, participants are invited to suggest how to generate or complete a second list, that differs from a given list of entities in some attribute. For example, given 'Dutch impressionist art musea', we could request the system to provide a list of 'modern art musea in

The Netherlands'. Participants may want to consider how would users specify such requests?

Acknowledgements

Thanks to Jovan Pehcevski for his comments on a preliminary version of this document. We also want to thank Hugo Bast, Hugo Zaragosza, Wray Buntine, Ingo Frommholz, Peter Bailey, Sisay Fissaha Adafre, Erik Tjong Kim Sang, and Maarten de Rijke for their comments on very very early drafts of this document.



INEX 2007 Link the Wiki Task and Result Submission Specification

Shlomo Geva and Andrew Trotman

Sunday, August 3, 2007

Using the Wikipedia as a corpus, the Link-the-Wiki track aims to produce a reusable test collection including a standard procedure and metrics for the evaluation of (automated) link identification (from anchor-text to different element levels within a document). Given a new orphan Wikipedia document, the task is to analyse the text and recommend a set of incoming and outgoing links. Both the anchor-text source and the best-entry-point destination within the existing collection are to be automatically identified. Going beyond traditional text document analysis, and in the context of INEX, we operate at the fundamental level of XML element. This means that anchor text will be linked not only to a related document, but to a specific XML element within that document - the best entry point (BEP) from which to start reading the referenced material.

1 Link The Wiki Task in 2007

Participants will be given a set of 90 existing Wikipedia documents which have already been nominated by participants. These documents are referred to interchangeably as either topics or orphans. All Wikipedia collection links have been removed from the topics. The task is two fold:

1. Recommend, for these documents, anchor-text and destinations within the remainder of the Wikipedia (or within the same document).
2. Recommend incoming links, to these documents, from other Wikipedia documents.

1.1 Anchor Text Specification

An anchor is specified in three parts, the filename of the XML document, the start location of the anchor-text and the end location of the anchor-text. The first is specified as a unique file name and the other two are specified using the DOM.

The file name is the INEX Wikipedia collection file name, for example: **23816.xml**

The start and end location are specified using the absolute XPath expression relative to the file's root element. For instance:

```
/article[1]/body[1]/section[5]/section[2]/p[4]
```

the particular text node character position is specified using the DOM. In the following expression the last number (3) is the character position within the 6th text node.

```
/article[1]/body[1]/section[5]/section[2]/p[4]/text()[6].3
```

Term numbers start from 0, before the first character and finish at n, the length of the text node.

1.2 Destination Specification

Destinations are specified in two parts: a unique filename and a best entry point. We use the term best entry point as already defined in the INEX ad hoc track. It is the best location from which a user should start reading in order to satisfy their information need.

The identification of BEPs is already a sub-task in the *ad hoc* track at INEX and the method used there is used by Link-the-Wiki unmodified. It is an absolute XPath to a location within a document.

1.3 Specification of Links

The LTW task accepts submissions that identify links from anchor-text to best entry points. Anchor-text must be identified precisely by using the DOM as it is a passage of text and not an XML element. An anchor-text may only be linked once and may link to only one destination BEP. That is, it is invalid to link from the same location in a document to multiple destinations within the collection. It is also invalid to identify overlapping anchor-texts. Further, multiple instances of the same text within an anchor and within the same document may not be identified more than once – these are essentially repetitions of the same result and serve no purpose in evaluation.

An example submission is given in Section 3. As shown, each topic (orphan page) is identified by a filename (e.g. “1234.xml”) *and* by the document name (by which the page can be identified in the online Wikipedia, e.g. “Albert Einstein”). While the second is redundant, both are included for convenience and clarity of manual examination.

For each orphan two sets of links are identified - *outgoing* and *incoming*.

Outgoing links are composed of a set of *links* from the orphan page to existing Wikipedia pages within the INEX collection. Each *link* consists of an *anchor*, a target *file*, and a *best entry point* within that file. Collectively these identify a unique link, including the source and destination of the link.

Incoming links are composed of a set of *links* from anchor-texts within existing Wikipedia pages (in the INEX collection) to a *best entry point* in the orphan page.

See-also links, as seen in existing Wikipedia pages, are a reduction of this generalisation. By specifying the anchor-text (start and end) and best entry point as **/article[1]**, a link from an article to an article is identified. This is a deliberate decision made to accommodate low-cost entry into the Link-the-Wiki track in 2007.

2 Result Submissions

Each participating organization may submit up to 5 runs. Each run is expected to contain the results for all topics but may contain a subset of the topics. If a given topic is missing from a run then the run is considered to have returned no links for that topic (for the purpose of scoring). Up to 250 incoming links and 250 outgoing links may be specified for each orphan. Links for each document must be unique and non-overlapping. Any run violating the specification may be rejected.

2.1 LTW topics

The topics are available for download from the INEX 2007 web site.

2.2 Special Rules

The orphans are drawn from the INEX Wikipedia collection; consequently they are not truly orphaned documents. To simulate the genuine case in which these documents truly are orphans, the following rules must be adhered to:

- a. The un-orphaned version of the topics must be deleted from the collection. It is left to the participant to choose the best way to do this. The simplest (and safest) way is to physically delete and re-index the collection, but a virtual deletion may also be possible. The collection should appear as though none of the orphans exist when identifying the links for a given topic. Topics are not added to the collection once links have been identified. That is, delete all the un-orphaned documents from the collection, and use this reduced version for each and every topic. Links between the orphans are assumed not to exist.
- b. The reduced version of the collection will still contain links to the orphans. Systems **must not** use these links in any way. Links to non-orphan documents may be fully exploited. It is fully valid to use the text but it is not valid to use the XML for text segmentation. For example, if the “Albert Einstein” page is a topic file then it is OK to identify the text “Albert Einstein” in another existing

document and suggest a link. It is NOT OK to search for collection links for the “Albert Einstein” page. In other words, text can be used, but not the knowledge that it appears inside a Wikipedia link element. For clarity, the following tags linking to the orphaned documents must be ignored (but if they link elsewhere they may be used):

- <collectionlink>
- <wikipedialink>
- <unknownlink>

2.3 Run Timings

A Link-the-Wiki submission can contain 250 incoming and 250 outgoing links for each of the 90 topics, that is a total of 45,000 links per submission. This task is expected to thoroughly tax the search engine. One solution is to submit fewer than 250 links - but the consequence will be a reduction in precision. Other solutions include generating the results on multiple machines or on one expensive machine. Of particular interest is this precision / time / cost trade-off.

Along with the submission of links participants are asked to submit details of the time taken to generate the results (excluding the output to the submission file, but including the reading and parsing of the orphan) and details of the hardware used to generate the result. Time should be CPU time in seconds (to 2 decimal places). The details of the computer are difficult to specify as such details as cache size differ from machine to machine. For 2007 the details of the CPU, core numbers, hyperthreading, and main memory are requested. Please be as accurate as reasonably possible.

2.4 Submission DTD

```
<!ELEMENT inex-submission (details+, description, collections, topic+)>
<!ATTLIST inex-submission
    participant-id CDATA #REQUIRED
    run-id CDATA #REQUIRED
    task (LinkTheWiki) #REQUIRED
>
<!ELEMENT details (machine, time)>
<!ELEMENT machine (cpu, speed, cores, hyperthreads, memory)>
<!ELEMENT cpu (#PCDATA)>
<!ELEMENT speed (#PCDATA)>
<!ELEMENT cores (#PCDATA)>
<!ELEMENT hyperthreads (#PCDATA)>
<!ELEMENT memory(#PCDATA)>
<!ELEMENT time (#PCDATA)>

<!ELEMENT description (#PCDATA)>
<!ELEMENT collections (collection+)>
<!ELEMENT collection (#PCDATA)>
<!ELEMENT topic (outgoing, incoming)>
<ATTLIST topic
    file CDATA #REQUIRED
    name CDATA #REQUIRED
>
<!ELEMENT outgoing (link+)>
<!ELEMENT incoming(link+)>
<!ELEMENT link (anchor,linkto)>
<!ELEMENT anchor (file, start, end)
<!ELEMENT linkto (file, bep)
<!ELEMENT file (#PCDATA)>
<!ELEMENT start (#PCDATA)>
<!ELEMENT end (#PCDATA)>
<!ELEMENT bep (#PCDATA)>
```

2.5 Example submission

```
<inex-submission participant-id="12" run-id="LTW_01">
<details>
  <machine>
    <cpu> Intel Celeron</cpu>
    <speed>1.06GHz</speed>
    <cores>1</cores>
    <hyperthreads>1</hyperthreads>
    <memory>128MB</memory>
  </machine>
  <time>3.04 seconds</time>
</details>
<description>Using text chunking etc.</description>
<collections>
<collection>wikipedia</collection>
</collections>
<topic file="13876.xml" name="Albert Einstein">
  <outgoing>
    <link>
      <anchor>
        <file> 13876.xml </file>
        <start> /article[1]/body[1]/p[3]/text()[2].10 </start>
        <end> /article[1]/body[1]/p[3]/text()[2].35 </end>
      </anchor>
      <linkto>
        <file> 123456.xml </file>
        <bep> /article[1]/sec[3]/p[8] <bep>
      </linkto>
    </link>
    ...
  </outgoing>
  <incoming>
    <link>
      <anchor>
        <file> 654321.xml </file>
        <start> /article[1]/body[1]/p[3]/text()[2].10 </start>
        <end> /article[1]/body[1]/p[3]/text()[2].35 </end>
      </anchor>
      <linkto>
        <file> 13876.xml </file>
        <bep> /article[1]/sec[3]/p[8] <bep>
      </linkto>
    </link>
    ...
  </incoming>
</topic>
</ inex-submission>
```

INEX 2007 Multimedia Track: Guidelines for Topic Development for the *MMimages* Task

Thijs Westerveld Theodora Tsikrika et al.*

May 30, 2007

1 Introduction

Structured document retrieval allows for the retrieval of document fragments, i.e., XML elements, containing relevant information. The main INEX Ad Hoc task focuses on text-based XML element retrieval. Although text is dominantly present in most XML document collections, other types of media can also be found in those collections. Existing research on multimedia information retrieval has already shown that it is far from trivial to determine the combined relevance of a document that contains several multimedia objects. The objective of the INEX 2007 multimedia track is to exploit the XML structure that provides a logical level at which multimedia objects are connected, to improve the retrieval performance of an XML-driven multimedia information retrieval system.

For INEX 2007 multimedia (*MM*) track, we define two main tasks: *MMfragments* and *MMimages*. For the *MMfragments* task, the topics have been created during the topic development phase of the Ad Hoc task¹. Here, we are only concerned with the *MMimages* task. We present the available resources for the task, describe the task in detail, and provide the topic development guidelines for the task. Topic formats follow the Ad Hoc topic formats.

2 Track resources

The resources used for the multimedia track are based on wikipedia data. The following five resources are available. Detailed description of each of them and information on how to obtain them are provided at the INEX *MM* track website at <http://inex.is.informatik.uni-duisburg.de/2007/mmtrack.html>.

Wikipedia Ad Hoc XML collection: This is the same collection that is used for the INEX 2007 Ad Hoc track. The INEX 2007 wikipedia collection

*Based on prior guidelines for the Multimedia and Ad Hoc Tracks [6, 5, 7, 1, 3].

¹The topic development guidelines for the Ad Hoc and *MMfragments* tasks are provided in [3] available at the INEX Ad Hoc track website (<http://inex.is.informatik.uni-duisburg.de/2007/adhoc.html>) under Topics.

is a copy of the INEX 2006 collection with image identifiers added to the `<image>` tags for those images that are part of the multimedia corpus. The assumption is that a user will be able to see images from the multimedia corpus (those with *id* > 0) in-place. The identifiers refer to the images and metadata files in the wikipedia image XML collection.

Wikipedia image collection: A subset of images referred to in the Wikipedia Ad Hoc XML collection is chosen to form the Wikipedia image collection. Note that (due to possible copyright issues) not all images referred to in the Ad Hoc collection are included in the multimedia corpus. Only the multimedia corpus images are assumed to be available for the user.

Wikipedia image XML collection: This XML collection is specially prepared for the multimedia track. It consists of XML documents containing image meta-data. See Figure 1 for an example document. The corresponding image is given in Figure 2. Each document contains exactly one image with (often) a short description. This corresponds with the information that is also available on wikipedia, consider for instance: <http://en.wikipedia.org/wiki/Image:AnneFrankHouseAmsterdam.jpg>.

Image classification scores: For each image the classification scores for 101 different concepts are derived by UvA [2].

Image features: A set of 120D feature vectors, one for each image, is available that has been used to derive the image classification scores. These feature vectors can be used to build a custom CBIR-system, without having to pre-process/access the image collection [4].

All these resources can be used for the *MMimages* task, described next.

3 Task description

The task for the multimedia track is to retrieve relevant information, based on an information need with a (structured) multimedia character. A structured document retrieval approach in that case should be able to combine the relevance of different media types into a single ranking that is presented to the user. The INEX multimedia track differs from other approaches in multimedia information retrieval, like TRECVID and IMAGECLEF, in the sense that it focuses on using the structure of the document to extract, relate and combine the relevance of different multimedia fragments.

This document describes the *MMimages* task for INEX 2007 *MM* track:

MM Images task: Find relevant images given an information need. The target collection for this task is the **Wikipedia image XML collection**. Given an information need, participants are required to return a ranked list of **documents** (=image+metadata) from this collection. Here, the type of the target element is defined, so basically this is closer to image


```

<?xml version="1.0"?>
<article>
  <name id="1116948">AnneFrankHouseAmsterdam.jpg</name>
  <image xmlns:xlink="http://www.w3.org/1999/xlink"
    xlink:type="simple" xlink:actuate="onLoad"
    xlink:show="embed"
    xlink:href="../Pictures/AnneFrankHouseAmsterdam.jpg">
AnneFrankHouseAmsterdam.jpg</image>
  <text>Anne Frank House - The Achterhuis - Amsterdam.
    Photo taken by
  <wikilink type="internal" parameters="2">
    <wikiparameter number="0">
      <value>User:Rossrs</value>
    </wikiparameter>
    <wikiparameter number="1" last="1">
      <value>Rossrs</value>
    </wikiparameter>
  </wikilink>mid 2002
  <wikitemplate parameters="1">
    <wikiparameter number="0" last="1">
      <value>PD-self</value>
    </wikiparameter>
  </wikitemplate>
  <p />
  <wikilink type="internal" parameters="1">
    <wikiparameter number="0" last="1">
      <value>es:Image:AnneFrankHouseAmsterdam.jpg</value>
    </wikiparameter>
  </wikilink>
  <p />
  <wikilink type="internal" parameters="1">
    <wikiparameter number="0" last="1">
      <value>Category:Building and structure images</value>
    </wikiparameter>
  </wikilink></text>

```

Figure 1: XML document containing meta-data for image: AnneFrankHouse-Amsterdam.jpg

retrieval, rather than XML element retrieval. Still, the structure of (supporting) documents, together with the visual content and context of the images, could be exploited to get to the relevant images (+their metadata).

4 Topic development

4.1 Topic Creation criteria

Creating a set of topics for a test collection requires a balance between competing interests. The performance of retrieval systems varies largely for different topics. This variation is usually greater than the performance variation of different retrieval methods on the same topic. Thus, to judge whether one retrieval



Figure 2: Example image: AnneFrankHouseAmsterdam.jpg

strategy is (in general) more effective than another, the retrieval performance must be averaged over a large and diverse set of topics. In addition, the average performance of the retrieval systems on the topics can be neither too good nor too bad as little can be learned about retrieval strategies if systems retrieve no, or only relevant, documents.

When creating topics, a number of factors should be taken into consideration. Topics should:

- be authored by an expert in (or someone familiar with) the subject areas covered by the collection,
- reflect real needs of operational systems,
- represent the type of service an operational system might provide,
- be diverse,
- differ in their coverage, e.g. broad or narrow topic queries,
- be assessed by the topic author.

4.2 Topic Format

The INEX *MM* track topics are Content Only + Structure (CO+S) topics, like in the Ad Hoc track. While in multimedia the term content often refers to

visual content, in INEX it means textual or semantic content of a document part. The term content-only is used within INEX for topics or queries that use no structural or visual hints.

The 2007 CO+S topics consist of the following parts, which are explained in detail below:

<title> in which Content Only (CO) queries are given

<castitle> in which Content And Structure (CAS) queries are given

<description> a one or two sentence natural language definition of the information need

<narrative> in which the definitive definition of relevance and irrelevance are given

4.2.1 **<narrative>**

A clear and precise description of the information need is required in order to unambiguously determine whether or not a given document fulfills the given need. In a test collection this description is known as the narrative. It is the only true and accurate interpretation of a user's needs. Precise recording of the narrative is important for scientific repeatability - there must exist, somewhere, a definitive description of what is and is not relevant to the user. To aid this, the **<narrative>** should explain not only what information is being sought, but also the context and motivation of the information need, i.e., why the information is being sought and what work-task it might help to solve.

Many different queries could be drawn from the **<narrative>**, and some are better than others. For example, some might contain phrases; some might contain ambiguous words; while some might even contain domain specific terms, visual constraints or hints. Regardless of the query, the search engine results are not necessarily relevant. Even though a result might contain search terms from the query, it might not match the explanation given in the **<narrative>**. Equally, some relevant documents might not be found, but they remain relevant because they are described as so by the **<narrative>**.

The different CO+S topic parts relate to different scenarios that lead to different types of queries.

4.2.2 **<title>**

The topic **<title>** simulates a user who does not have (or want to use) example images or other visual constraints. The query expressed in the topic **<title>** is therefore a Content Only (CO) query. This profile is likely to fit most users searching XML digital libraries.

4.2.3 <castitle>

Upon discovering their <title> query returned many irrelevant hits, a user might decide to add visual hints (by rewriting as a CAS query). At INEX, these added visual constraints are specified using the NEXI formal syntax (see the INEX website for the specification) and recorded in the topic <castitle>.

In the *MMimages* task, the target collection is the Wikipedia image XML collection. Given an information need, participants are required to return a ranked list of **documents** (=image+metadata) from this collection, i.e. the target element is defined (a document). Therefore, the <castitle> query should be:

```
//article[X]
```

where *X* is a predicate using one or more *about* functions. Apart from the *about* functions for text (cf. [3]), two additional special types of *about* clauses are allowed, both specifying visual hints or constraints.

1. The first type is used for visual similarity. If a user wants to indicate that results should have images similar to a given example image, this can be indicated in an *about* clause with an URL. For example to find pictures of the Apple II computer similar to the one at <http://lrs.ed.uiuc.edu/students/scooper/AppleII.jpg>, one could type

```
//article[about(., Apple II) and  
about(.,src:lrs.ed.uiuc.edu/students/scooper/AppleII.jpg)]
```

Please make sure the image you use as an example is **NOT** part of the INEX wikipedia collection, since we do not want to give credit for finding the example image itself. Also, try to use images from the .edu and .gov domains as they are expected to be more stable. Although, we will keep copies of the images, we cannot guarantee they will always be available. The URL in the NEXI query will be the primary source for the image.

2. The second type of visual hints is directly related to the image classifications that are provided as an additional source of information (for details, see the multimedia track pages at the INEX 2007 website: <http://inex.is.informatik.uni-duitburg.de/2007/mmtrack.html>). If a user thinks the results should be of a given concept, this can be indicated with an *about* clause with the keyword *concept:*. For example, to search for cityscapes one could decide to use the concept building:

```
//article[about(.,cityscape) and about(.,concept:building)]
```

Terms following the keyword *concept:* are obviously restricted to the 101 concepts for which classification results are provided (cf. the INEX *MM* track website).

The three different types of about clauses (textual terms, visual examples and visual concepts) can be used in any combination. It is up to the systems how to use, combine or ignore this information; the relevance of an result item does not directly depend on these constraints, but it is decided by manual assessments based on the <narrative>.

The NEXI parser is extended for this purpose and available from the multimedia track web-site.

4.2.4 <description>

As an alternative to entering queries into search engines, a user might ask a librarian to find the information to satisfy their need. Such a user would give a verbal description to the librarian using a natural language. Just as there are many CO queries derivable from the <narrative>, there are many ways to express the need in natural language. However it is expressed, it is important that it is precise, concise, and as informative as the <title> and <castitle>, i.e. it contains the same terms and the same structural requirements that appear in the <title> and <castitle>, albeit expressed in natural language.

4.3 Procedure for Topic Development

Each participating group will have to submit **4 topics by the 4th June 2007** for the *MMimages* task. Submission is done by filling in the Candidate Topic Submission Form on the INEX web site: <http://inex.is.informatik.uni-duisburg.de/2007/> under Tracks → Multimedia → Topics. We encourage the participants to define more than 4 topics, to increase the reliability of the results, as argued in Section 4.1

The topic creation process is divided into several steps. When developing a topic, use a print out of the on-line Candidate Topic Form to record all information about the topic you are creating.

Step 1: Initial Topic Statement Create a one or two sentence description of the information you are seeking. This should be a simple description of the information need without regard to retrieval system capabilities or document collection peculiarities. This should be recorded in the Initial Topic Statement field. Record also the context and motivation of the information need, i.e. why the information is being sought. Add to this a description of the work-task, that is, with what task it is to help (e.g. writing an essay on a given topic).

Step 2: Exploration Phase In this step the initial topic statement is used to explore the collection. Obtain an estimate of the number of relevant documents then evaluate whether this topic can be judged consistently. You may use any retrieval engine for this task, including your own or the TopX system (<http://infa05501.ag5.mpi-sb.mpg.de:8080/topx/>), provided through the INEX website. Make sure you select the wikipedia image XML collection for the *MMimages* task (INEX Wikipedia[Multi-Media] in TopX).

Alternatively, you may choose to perform a visual exploration. For this, you can use the provided UvA concept classifications (available at <http://inex.is.informatik.uni-duisburg.de/2007/downloads/UvAconcepts>) to get an impression of the top 100 images for a given topic. Or you can use your own content based information retrieval system to perform a similarity search, using the images you find on the Web.

Step 2a: Assess Top 25 Results While you use one or more search engines to explore the collection, assess the relevance of a retrieved document using the following working definition: mark it relevant if it would be useful if you were writing a report on the subject of the topic, or if it contributes toward satisfying your information need. Each result should be judged on its own merits. That is, information is still relevant even if it is the thirtieth time you have seen the same information. It is important that your judgment of relevance is consistent throughout this task. Using the Candidate Topic Submission Form record the number of found relevant documents and the path representing each relevant document. We ask you to look at the top 25 results for at least one search engine. Then if you found fewer than 2 relevant documents in total, or more than 20 using a single search engine, abandon the topic and use a new one. Otherwise, perform a feedback search (see below).

Step 2b: Feedback Search After assessing the top 25 results, you should have an idea of which terms (if any) could be added to the query to make the query as expressive as possible for the kind of results you wish to retrieve. You should also have an idea of which terms could be used to disambiguate relevant from irrelevant results and if visual clues are present in the query.

Use the expanded query and a single search engine of choice (preferably the one that produced the most relevant answers), to retrieve a new list of candidates. Judge the top 100 results (some are already judged), and record the number of relevant results in Candidate Topic Form. Record the expanded query in the title field of the Candidate Topic Submission Form.

Step 3: Write the <narrative> Having judged the top 100 results you should have a clear idea of what makes a component relevant or not. It is important to record this in minute detail as the <narrative> of the topic. The <narrative> is the definitive instruction used to determine relevance during the assessment phase (after runs have been submitted). Record not only what information is being sought, but also what makes it relevant or irrelevant. Also record the context and motivation of the information need. Include the work-task, which is: the form the information will take after having been found (e.g. written report). Make sure your description is exhaustive as there will be several months between topic development and topic assessment.

Step 4 CO+S: Optionally write the <castitle> Optionally re-write the title by adding visual examples and/or visual concepts. Record this as the

<castitle> on the Candidate Topic Submission Form. The form contains a link to the online NEXI parser to check the syntax of your castitle.

Also record why you think the visual hints might help in the <narrative>. **Please note that we aim at having castitles in most topics.** Also note that since the *MMimages* task is a document retrieval task, here the target element should be an article element, thus all NEXI queries in this task should be of the form:

```
//article[X]
```

where X is a predicate using one or more about functions.

Step 5: Write the <description> Write the <description>, the natural language interpretation of the query. Ensure the information need as expressed in the <title>, and <castitle> is also expressed in the <description>. Make sure the <description> does not express any additional information needs.

Step 6: Refining Topic Statements Finalize the topic <title>, <castitle>, <description>, and <narrative>. It is important that these parts all express the same information need; it should be possible to use each part of a topic in a stand-alone fashion. In case of dispute, the <narrative> is the definitive definition of the information need - all assessments are made relative to the <narrative> and the <narrative> alone.

Step 7: Topic Submission Once you are finished, fill out and submit the online Candidate Topic Submission Form on the INEX website <http://inex.is.informatik.uni-duisburg.de/2007/> under Tracks → Multimedia → Topics. After submitting a topic you will be asked to fill out an online questionnaire (this should take no longer than 5-10 minutes). It is important that this is done as part of the topic submission as the questions relate to the individual topic just submitted and the submission process. This is part of an effort to collect more context for the INEX topics as discussed at the Dagstuhl workshop.

Please make sure you submit all candidate topics no later than the **4th June 2007**.

5 Topic Selection

From the received candidate topics, the INEX organizers will decide which topics to include in the final set. This is done to ensure inclusion of a broad set of topics. The data obtained from the collection exploration phase is used as part of the topic selection process. The final set of topics will be distributed for use in retrieval and evaluation.

References

- [1] Birger Larsen and Andrew Trotman. INEX 2006 guidelines for topic development. Unpublished document distributed to INEX 2006 participants.
- [2] C.G.M. Snoek, M. Worring, J.C. van Gemert, J.M. Geusebroek, and A.W.M. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *ACM Multimedia Conference*, 2006.
- [3] Andrew Trotman and Birger Larsen. INEX 2007 guidelines for topic development. Unpublished document distributed to INEX 2007 participants.
- [4] Jan C. van Gemert, Jan-Mark Geusebroek, Cor J. Veenman, Cees G.M. Snoek, and Arnold W.M. Smeulders. Robust scene categorization by learning image statistics in context. In *CVPR Workshop on Semantic Learning Applications in Multimedia*, New York, USA, June 2006.
- [5] R. van Zwol, G. Kazai, and M. Lalmas. INEX 2005 multimedia track. In *Advances in XML Information Retrieval*, Lecture Notes in Computer Science. Springer, 2006.
- [6] Roelof van Zwol, Mounia Lalmas, and Gabriella Kazai. INEX 2005 multimedia track – working document. Unpublished document distributed to INEX 2005 MM participants.
- [7] Thijs Westerveld and Roelof van Zwol. INEX 2006 multimedia track guidelines. Unpublished document distributed to INEX 2006 MM participants.

INEX 2007 Multimedia Track: Specification of Retrieval Tasks and Result Submissions

Theodora Tsikrika Thijs Westerveld

June 13, 2007

1 Introduction

For INEX 2007 multimedia (*MM*) track, we define two main tasks: *MMfragments* and *MMimages*. The *MMfragments* task is similar to the adhoc retrieval of **XML fragments** (i.e., elements or passages, as these are defined in the INEX 2007 Ad Hoc track), with the main difference being that *MMfragments* topics ask for multimedia fragments (i.e., fragments containing at least one image). The *MMimages* task, on the other hand, is similar to the adhoc retrieval of **XML documents**, with the requirement that these documents contain only images and their metadata. For both *MM* tasks, topics are based on information needs with a (structured) multimedia character and may contain structural and visual hints.

Given the similarities with adhoc retrieval tasks, we define the *MMfragments* subtasks to be the same with the INEX 2007 Ad Hoc subtasks (i.e., Focused, Relevant in Context, and Best in Context). In addition, the result submission format for both *MM* tasks follows a submission format highly similar to that of the INEX 2007 Ad Hoc track, in order to maintain a consistency between the two tracks.

The specification of the INEX 2007 Ad Hoc track retrieval tasks and result submission is given in document [1]. The document here complements the specifications of the Ad Hoc track, by detailing only the points that differ between the two tracks.

2 Retrieval Tasks

For INEX 2007 *MM* track, we define the following two tasks: *MMfragments* and *MMimages*.

2.1 *MMfragments* task

The objective of the *MMfragments* retrieval task is to find relevant XML fragments (i.e., elements or passages) in the **Wikipedia Ad Hoc XML collection**

given a multimedia information need. Within the *MMfragments* task, we define the same three subtasks as in the Ad Hoc track:

1. FOCUSED TASK asks systems to return a ranked list of elements or passages to the user.
2. RELEVANT IN CONTEXT TASK asks systems to return relevant elements or passages clustered per article to the user.
3. BEST IN CONTEXT TASK asks systems to return articles with one best entry point to the user.

For detailed descriptions of the three subtasks, see [1]. The difference is that *MMfragments* topics ask for multimedia fragments (i.e., fragments containing at least one image) and may also contain visual hints.

Similarly to the INEX 2007 Ad Hoc tasks, participants to the *MMfragments* task are invited to experiment with XML element retrieval versus passage retrieval. Details of these two retrieval approaches are discussed in [1].

What we hope to learn from this task (in addition to the research questions outlined in [1]) is: Do content conditions and structural hints need to be interpreted differently for the *MMfragments* compared to the Ad Hoc tasks? How do visual hints in the query help *MM* retrieval?

2.2 *MMimages* task

The objective of the *MMimages* retrieval task is to find relevant images in the **Wikipedia image XML collection** given a multimedia information need. Given an information need, participants are required to return a ranked list of **documents** (=image+metadata) from this collection. Here, the type of the target element is defined, so basically this is closer to image retrieval (or a document retrieval task), rather than XML element or passage retrieval. Still, the structure of (supporting) documents, together with the visual content and context of the images, could be exploited to get to the relevant images (+their metadata).

What we hope to learn from this task is: How do visual hints in the query help image retrieval? How does the (structural) context in which an image is used contribute to image retrieval effectiveness?

3 INEX 2007 *MM* Topics

There are two separate topic sets for the *MM* track at INEX 2007, one for each of its tasks:

- For the *MMfragments* task, the topic set is a subset of the topic set for the Ad Hoc tasks and contains topics **525-543**. The *MMfragments* topic set cannot be downloaded independently, but only as part of the Ad Hoc topic set found at the INEX Ad Hoc track web site (<http://inex.is.informatik.uni-duisburg.de/2007/adhoc.html>) under Topics.

- For the *MMimages* task, the topic set can be downloaded from the INEX *MM* track web site (<http://inex.is.informatik.uni-duisburg.de/2007/mmtrack.html>) under Topics.

The format of the topics is defined in the **INEX topics DTD** provided in [1], where details on the use of the different topic fields are also outlined.

The topic fields correspond to the following types of queries:

1. Queries with content-only conditions are requests that ignore the document structure and contain only content related conditions, e.g. only specify what a retrieval result should be about without specifying what that result is (XML document, XML element or passage). More details can be found in [1].
2. Queries with content conditions and structural hints are more expressive topic statements that contain explicit references to the XML structure. They explicitly specify the contexts of the user's interest (e.g. target elements) and/or the context of certain search concepts (e.g. containment conditions). More details can be found in [1].
3. Queries with content conditions, structural and visual hints also contain explicit references to either or both of the following: (i) example images that can be used for visual similarity, and (ii) concepts related to the image classifications (w.r.t. the 101 classes) that are provided as an additional source of information. The example images are available on the Web and the primary source of their identification is their URL; these images will also be provided for research purposes at the INEX *MM* website <http://inex.is.informatik.uni-duisburg.de/2007/mmtrack.html> under Topics.

At INEX 2007 *MM* track, all topics have both a keyword `<title>` query and a structured `<castitle>` query. Some of the topics also have an `<mmtitle>` query. For the *MMimages* task, in particular, which requires the retrieval of documents, rather than elements or passages, the `<castitle>` and `<mmtitle>` queries are restricted to: `//article[X]`, where `X` is a predicate using one or more *about* functions.

As noted above, for both *MM* tasks, we want to find out if, when, and how the visual hints in the query have an impact on retrieval effectiveness; therefore, participants to the *MM* track are encouraged to submit runs using the `<mmtitle>` field (where available). Participants are allowed to use all fields, but only runs using `<title>`, `<castitle>`, `<mmtitle>`, `<description>`, or a combination of these will be regarded as truly *automatic*, since the additional fields will not be available in operational settings. The submission format will record the precise topic fields that are used in a run.

4 Result Submission

Once the topics are distributed, participants can start working on their runs. For each run we would like to know which sources are used (Ad Hoc Collection, Image XML collection, visual features, classification data). We would encourage groups to do a baseline run that uses the `<mmtitle>` part of the query (where available) and no sources of information except for the target collection (image XML collection for *MMimages* task, Ad Hoc Collection for *MMfragments* task).

4.1 Runs

For the *MMfragments* task, we allow up to 3 submissions per participant per subtask (Focused, Relevant in Context, and Best in Context). The requirements for these three subtasks are described in [1]. Focused will be the main task to compare systems on; therefore, participants are required to submit at least one run for the focused subtask of the *MMfragments* task in the *MM* track. Since submissions for the Ad Hoc track will also consider the subset of topics used for the *MMfragments* task (Ad Hoc topics 525-543), submissions for the *MMfragments* task will also be compared to the adhoc submissions on these 19 topics. So, groups participating in both tracks should not resubmit to *MMfragments* a run already submitted to the Ad Hoc track.

For the *MMimages* task, we allow up to 6 submissions per participant. Since this is an image retrieval task (or rather a document retrieval task), only full documents should be retrieved (i.e., images + metadata). No fragments should be returned. This means the path of each of the results for this task should be `/article[1]`. In addition, duplicate files are not allowed in the results.

For both *MMfragments* and *MMimages* tasks, the results of one run must be contained in one submission file (i.e., up to 9 files can be submitted in total for the *MMfragments* task and up to 6 files for the *MMimages* task). A submission may contain up to 1,500 retrieval results for each of the topics.

4.2 Submission Format

For relevance assessments and the evaluation of the results for both tasks of the *MM* track, submission files follow a submission format highly similar to that of the Ad Hoc track. The **submission DTD** for the *MM* tasks is given below:

```
<!ELEMENT inex-submission (topic-fields, description, collections, topic+)>
<!ATTLIST inex-submission
  participant-id CDATA #REQUIRED
  run-id        CDATA #REQUIRED
  task          (MM_Focused | MM_RelevantInContext | MM_BestInContext | MMimages) #REQUIRED
  query         (automatic | manual) #REQUIRED
  result-type   (element | passage) #REQUIRED
>
<!ELEMENT topic-fields EMPTY>
<!ATTLIST topic-fields
  title         (yes|no) #REQUIRED
```

```

mmtitle          (yes|no) #REQUIRED
castitle         (yes|no) #REQUIRED
description      (yes|no) #REQUIRED
narrative       (yes|no) #REQUIRED
>
<!ELEMENT resources EMPTY>
<!--ATTLIST resources
wikipedia          (yes|no) #REQUIRED
wikipedia_IMG      (yes|no) #REQUIRED
UvAfeatures       (yes|no) #REQUIRED
UvAconcepts       (yes|no) #REQUIRED
-->
<!--ELEMENT description (#PCDATA)>
<!--ELEMENT topic (result*)>
<!--ATTLIST topic topic-id CDATA #REQUIRED -->
<!--ELEMENT collections (collection+)>
<!--ELEMENT collection (#PCDATA)>
<!--ELEMENT result (in?,file, (path | passage), rank?, rsv?)>
<!--ELEMENT in (#PCDATA)>
<!--ELEMENT file (#PCDATA)>
<!--ELEMENT path (#PCDATA)>
<!--ELEMENT passage EMPTY>
<!--ATTLIST passage
start             (#PCDATA) #REQUIRED
end_IMG          (#PCDATA) #REQUIRED
-->
<!--ELEMENT rank (#PCDATA)>
<!--ELEMENT rsv (#PCDATA)>

```

Here, we only present the elements and attributes of the above DTD that are different to the ones in the submission DTD of the Ad Hoc track (provided in [1]). The differences are the following:

- The identification of the task (MM_Focused or MM_RelevantInContext or MM_BestInContext or MMimages)
- The collections should be set to ‘wikipedia’ for the *MMfragments* task and to ‘wikipedia_IMG’ for the *MMimages* task.
- The resources used by the retrieval algorithm should be recorded (wikipedia, wikipedia_IMG, UvAfeatures, UvAconcepts).
- Once again, *MMimages* is a document retrieval task, the only submitted <path> allowed for this task is /article[1]. In practice, we will ignore the submitted <path> and only use the <file> field of the result (duplicate files in a single run are not allowed).

Here, is a sample submission file for the *MMfragments* Focused task:

```

<inex-submission participant-id="12" run-id="VSM_Aggr_06"
  task="MM_Focused" query="automatic" result-type="element">
  <topic-fields title="no" mmtitle="yes" castitle="no" description="no" narrative="no"/>
  <resources wikipedia="yes" wikipedia_IMG="yes" UvAfeatures="no" UvAconcepts="no"/>
  <description>Using VSM to compute RSV at leaf level combined with
  aggregation at retrieval time, assuming independence and using

```

```
augmentation weight=0.6.</description>
<collections>
  <collection>wikipedia</collection>
</collections>
<topic topic-id="01">
  <result>
    <file>9996</file>
    <path>/article[1]</path>
    <rsv>0.67</rsv>
  </result>
  <result>
    <file>5492</file>
    <path>/article[1]/name[1]</path>
    <rsv>0.1</rsv>
  </result>
  [ ... ]
</topic>
<topic topic-id="02">
  [ ... ]
</topic>
[ ... ]
</inex-submission>
```

4.3 Result Submission Procedure

To submit a run, please use the following link: <http://inex.is.informatik.uni-duisburg.de/2007/mmtrack.html> Then go to Submissions. The online submission tool will be available soon.

References

- [1] Charles L. A. Clarke, Jaap Kamps, and Mounia Lalmas. INEX 2007 retrieval task and result submission specification. Unpublished document available to INEX 2007 participants at the INEX 2007 Ad Hoc track website at <http://inex.is.informatik.uni-duisburg.de/2007/adhoc.html>.