INEX 2010 Book Track
Search Task Definitions and Submission Guidelines

Gabriella Kazai and Marijn Koolen Version 1.0

1 Introduction

The goal of the Book track is to evaluate techniques that support users in searching, navigating and reading collections of digitized books. The track investigates four tasks: Social Search for Best Books (SB), Prove It! (PI), Structure Extraction (SE) and Active Reading (ART). This document details only the two search tasks: SB and PI.

2 Participation

To take part you will need to register to the book track at
https://inex.mmci.uni-saarland.de/people/register.jsp.

Once you registered, you will be issued with a username and password. These will give you access to the data (book corpus, topics, relevance judgements) and services on http://www.booksearch.org.uk/. If you encounter any problems, please email Gabriella Kazai at v-gabkaz@microsoft.com.

Participants may take part in any of the tasks. The minimum participation requirement is to contribute results or study participants to at least one of the four tasks (SB, PI, SE or ART).

3 What's new in 2011 compared to 2010?

- This year, we are introducing a new task, Social Search for Best Books (SB), which is based on a new collection of book metadata from Amazon and LibraryThing.com.
- There is a small change to the submission format for PI runs, allowing systems to indicate if a search result confirms or refutes the factual claim of a topic.

4 Social Search for Best Books (SB)

4.1 SB goals

The goal of the SB task is to evaluate the relative value of *controlled book metadata*, such as classification labels, subject headings and controlled keywords, versus *user-generated or social metadata*, such as tags, ratings and reviews, for retrieving the most relevant books for a given user request. Controlled metadata, such as the Library of Congress Classification and Subject Headings, is rigorously curated by experts in librarianship. It is used to index books to allow highly accurate retrieval from a large catalogue. However, it requires training and expertise to use effectively, both for indexing and for searching. On the other hand, social metadata, such as tags, are less rigorously defined and applied, and lack vocabulary control by design. However, since such metadata is contributed directly by the users and it may better reflect the terminology of everyday searchers. Clearly, both types of metadata have advantages and disadvantages. The task aims to investigate whether one is more suitable than the other to support different types of search requests or how they may be fruitfully combined.

4.2 SB user scenario

The scenario is that of a user turning to Amazon Books and LibraryThing to search for books they want to gather on a given topic. Both services host large collaborative book catalogues that may be used to locate books of interest.

On LibraryThing, users can catalogue the books they read, manually index them by assigning tags, and write reviews for others to read. Users can also post messages on a discussion forum asking for help in finding new, fun, interesting, or relevant books to read. The forums allow users to tap into the collective bibliographic knowledge of hundreds of thousands of book enthusiasts.

On Amazon, users can read and write book reviews and browse to similar books based on links such as 'customers who bought this book also bought… '.

4.3 SB task description

The SB task is to reply to a user's request that has been posted on the LibraryThing forums by returning a list of recommended books. The books must be selected from a corpus that consists a collection of book metadata extracted from Amazon Books and LibraryThing. The collection includes both curated and social metadata. User requests vary from asking for books on a particular genre, looking for books on a particular topic or period or books by a given author. The level of detail also varies, from a brief statement to detailed descriptions of what the user is looking for. Some requests include examples of the kinds of books that are sought by the user, asking for similar books. Other requests list examples of known books that are related to the topic but are specifically of no interest. The challenge is to develop a retrieval method that can cope with such diverse requests.

Participants may submit up to 6 runs and can use any field of the topic statement. However, at least one run should use the title field only.

4.4 SB corpus

The corpus consists of a collection of 2.8 million records from Amazon Books and LibraryThing.com. See https://inex.mmci.uni-saarland.de/data/nd-agreements.jsp for information on how to get access to this collection. Each book record is an XML file with fields like <isbn>, <title>, <author>, <publisher>, <dimensions>, <numberofpage> and <publicationdate>. Curated metadata comes in the form of a Dewey Decimal Classification in the <dewey> field, Amazon subject headings are stored in the <subject> field, and Amazon category labels can be found in the <browseNode> fields. The social metadata from Amazon and LibraryThing is stored in the <tag>, <rating>, and <review> fields.

We are in the process of crawling official library metadata for all the ISBNs in the collection. This library metadata contains official library classification information such as Library of Congress Classifications (LCC) and Subject Headings (LCSH), and will be released soon.

4.5 SB topic format

Topics consist of the following fields:

- title = the title of the topic
- group = the group name in which the topic is discussed
- narrative = the first message of the topic, posted by the topic creator
- type = the topic type, manually added by organisers. Type can be subject, author, genre, series or known-item
- genre = the genre of the topic, according to Library of Congress Classification (e.g. science:mathematics:computer_science for a topic about computer science)
- specificity = the specificity of the topic. Either broad or narrow
- similar = a list of isbns or author names that the topic creator mentions as positive examples. The topic creator wants books similar to those.
- dissimilar = a list of isbns or author names that the topic creator mentions as negative examples. The topic creator wants books dissimilar to those.

As an example topic, here is the format for topic 99309:

```
<topic id="99309">
    <title>Politics of Multiculturalism</title>
    <group>Political Philosophy</group>
    <narrative>I'm new, and would appreciate any recommended reading on the politics of
        multiculturalism. <author>Parekh</author>'s <work id="164382">Rethinking
        Multiculturalism: Cultural Diversity and Political Theory</work>(which I just finished) in the
        end left me unconvinced, though I did find much of value I thought he depended way too much
        on being able to talk out the details later. It may be that I found his writing style really
        irritating so adopted a defiant skepticism, but still... Anyway, I've read
        <author>Sen</author>, <author>Rawls</author>, <author>Habermas</author>, and
        <author>Nussbaum</author>, still don't feel like I've wrapped my little brain around the issue
        very well and would appreciate any suggestions for further anyone might offer.</narrative>
        <type>subject</type>
    <genre>politics</genre>
    <specificity>narrow</specificity>
    <similar><work id="164382"><isbn>0333608828</isbn><isbn>0674004361</isbn>
        <isbn>1403944539</isbn><isbn>0674009959</isbn></work><author>Parekh</author>
        <author>Sen</author><author>Rawls</author><author>Habermas</author>
        <author>Nussbaum</author></similar>
    <dissimilar><dissimilar>
</topic>
```

The topic and work id's are taken directly from LibraryThing. The mapping from the LibraryThing work id to the ISBNs in the collection is made using thingISBN.xml, which can be obtained from http://www.librarything.com/feeds/.

4.6 SB training topics

A set of training topics for the SB task is available at https://inex.mmci.uni-saarland.de/data/documentcollection.jsp#books.

These are not the official topics for 2011, but are meant for participants to debug and test their systems before submitting their official runs. Note that the relevance judgements for these training topics are extracted from the discussion forum conversations and are provided unfiltered (so many contain irrelevant books that were mentioned for some reason other than to provide information on relevant books). Furthermore, it is likely that these sets are incomplete, i.e., not all relevant books may be included. Thus, please use these topics with caution.

4.7 SB submission format

Submissions should conform to the TREC format of:

```
<topic-id> Q0 <ISBN> <rank> <retrieval-score> <run-id>
```

A run should contain up to 1,000 books per topic. Entries should be sorted by topic-id and then by retrieval-score. We will use ISBN numbers as book identifiers – these are provided as part of the test collection. Run IDs must be unique across all submissions sent from one organization – also please use meaningful, but short names if possible. Please note that we will use trec_eval at the evaluation stage to calculate performance scores, which ignores the rank column and orders documents by retrieval score.

4.8 SB evaluation

Systems will be evaluated using relevance judgements extracted from the LibraryThing topic threads (filtered and extended with editorial or crowdsourced judgements). The overall performance across topic types will be measured using MAP. Performance for different topic types will be individually evaluated using specific measures. For known-item topics we will use MRR and Success@10, for subject-related topics we will use MAP and P@10.

5 Prove It! (PI)

5.1 PI goals

Building on a corpus of over 50,000 digitized books, this task investigates the application of focused and semantic retrieval approaches to a collection of digitized books.

5.2 PI user scenario

The scenario underlying this task is that of a user searching for specific information in a library of books that can provide evidence to *confirm or refute a given factual statement*. Users expect to be pointed directly at book pages that can help them to confirm or refute the claim of the topic. Users are assumed to view the ranked list of retrieved book pages starting from the top of the list and moving down, examining each result. No browsing is considered (only the returned book pages are viewed by users).

5.3 PI task description

Systems need to return a ranked list of up 1,000 book pages (given by their XPaths) which are estimated to contain information that can confirm or refute a factual statement expressed in the topic, and (optionally) indicating for each result whether it provides positive or negative evidence regarding the claim.

Participants may submit up to 6 runs.

5.4 PI corpus

The PI task builds on a collection of 50,239 out-of-copyright books, digitized by Microsoft. The corpus contains books of different genre, including history books, biographies, literary studies, religious texts and teachings, reference works, encyclopedias, essays, proceedings, novels, and poetry. 50,099 of the books also come with an associated MAchine-Readable Cataloging (MARC) record, which contains publication (author, title, etc.) and classification information. Each book in the corpus is identified by a 16 character bookID, which is also the name of the directory that contains the book's OCR file, e.g., A1CD363253B0F403.

The OCR text of the books has been converted from the original DjVu format to an XML format referred to as BookML, developed by Microsoft Development Center Serbia. BookML provides additional structure information, including a set of labels (as attributes) and additional marker elements for more complex structures, like table of contents. For example, the first label attribute in the XML extract below signals the start of a new chapter on page 1 (label="PT_CHAPTER"). Other semantic units include headers (SEC_HEADER), footers (SEC_FOOTER), back-of-book index (SEC_INDEX), table of contents (SEC_TOC). Marker elements provide detailed markup, e.g., indicating entry titles (TOC_TITLE) or page numbers (TOC_CH_PN) in a table of contents. The basic XML structure of a typical book in BookML is a sequence of pages containing nested structures of regions, sections, lines, and words, most of them with associated coordinate information, defining the position of a bounding rectangle.

The full corpus, totalling around 400GB, is available on USB HDDs – please contact Gabriella Kazai at v-gabkaz@microsoft.com. A reduced version (13GB compressed) is available via download from http://www.booksearch.org.uk/. The reduced version was generated by removing the word tags and propagating their content into the parent (i.e., line) elements.

5.5 PI topics

There are 83 topics for this task, each containing a factual statement, see example below.

```
<topic id=2010006 ct_nt=11>
    <fact>On November 2, 1917, British prime minister Lord Balfour, in a letter to Lord Rotschild,
    declared the British Government's intention of assisting the Jewish people to establish a national
    home for the jews in Palestine.</fact>
```

```
    <query>Balfour declaration national home</query>
    <subject>national home for jews in Palestine</subject>
    <wikiurl>http://en.wikipedia.org/wiki/Balfour_Declaration_of_1917</wikiurl>
    <narrative>
        <task> I am working on a school assignment about the establishment of the state of Israel
        as the fulfillment of the Zionist aspirations. I was told that the Balfour declaration was an
        important milestone, and I want the fact confirmed. </task>
        <infneed> All statements in books are relevant that either prove or reject the fact. I need
        confirmation of the exact date, and preferably also the text of the declaration. </infneed>
    </narrative>
</topic>
```

Topics are available from the download area at http://www.booksearch.org.uk/.

5.6 PI training data

We have page-level relevance judgements for 21 of the 83 topics, which can be used for training. The judgements can be downloaded from the download area at http://www.booksearch.org.uk/.

5.7 PI evaluation

This is a high-precision task, with graded judgements, thus the official measure will be nDCG@10.

5.8 Submission format

Submissions for the Prove It task 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                (PI)        #REQUIRED
query               (automatic | manual) #REQUIRED
result-type         (page)      #REQUIRED
>
<!ELEMENT topic-fields EMPTY>
<!ATTLIST topic-fields
fact                (yes|no) #REQUIRED
subject             (yes|no) #REQUIRED
query               (yes|no) #REQUIRED
narrative           (yes|no) #REQUIRED
>
<!ELEMENT description (#PCDATA)>
<!ELEMENT topic (result+)>
<!ATTLIST topic topic-id CDATA #REQUIRED >
<!ELEMENT result (bookid, path, rank?, rsv?)>
<!ATTLIST result confirm (confirms | refutes | both | neither)>
<!ELEMENT bookid   (#PCDATA)>
<!ELEMENT path     (#PCDATA)>
<!ELEMENT rank     (#PCDATA)>
<!ELEMENT rsv      (#PCDATA)>
```

Each submission must contain the following:

- @participant-id: The Participant ID number of the submitting institute (available from the INEX website at: https://inex.mmci.uni-saarland.de/people/participants.jsp).
- @run-id: A run ID (which must be unique across all submissions sent from one organization – also please use meaningful, but short names if possible).
- @task: Identification of the task – please set this to "PI".
- @query: Specification whether the search query was constructed automatically ("automatic") or manually ("manual") from the topic.

- @result-type: Specification of the result-type – please set this to "page". A page is an XML element that should be given by its XPath (see Appendix A).
- topic-fields: Specification of which topic fields were used for constructing the search query (i.e., fact and/or query and/or narrative, etc.).
- description: A description of the retrieval approach applied to generate the run. Please add as much detail as you can, as this would help with the comparison and analysis of the results later on.

Furthermore, a run should contain the search results for each topic confirming to the following criteria:

- topic: Contains the ranked list of books estimated relevant to the given topic, ordered by decreasing value of relevance. Only a maximum of 1,000 book pages should be returned for each topic.
- @topic topic-id: Identifies the topic.
- result: Contains information for each book-part result in the ranking.
- **@confirm** **Whether the search result contains information that confirms or refutes the claim. The result may also do both confirm and refute some aspects of the claim at the same time, or it may simply be related to the claim without directly confirming or refuting it.**
- bookid: Each book should be identified using its bookID, which is the name of the directory that contains the XML source of the book.
- path: Book page results, identified by their XPaths, please see Appendix A.
- rank/rsv: For each result inside a book, its rank and/or RSV score can be recorded. Please note that the evaluation may rely on the rank order of the books and of the results inside books alone (values of the rank and rsv fields may be ignored).

An example submission may be as follows:

```
<bs-submission participant-id="25" run-id="BM25F-Focused-PageLevelRetrieval-With-ToC-
                BackOfBookIndex-Streams" task="book-focused" 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, indexing and retrieval only at page level, no relevance
                propagation</description>
<topic topic-id="01">
  <result confirm="confirms">
    <bookid>384D10DAEA4E34A8</bookid>
    <path>/document[1]/page[27]</path>
    <rank>1</rank>
  </result>
  <result confirm="refutes">
    <bookid>384D10DAEA4E34A8</bookid>
    <path>/ document[1]/page [122]</path>
    <rank>2</rank>
  </result>
  <result confirm="neither">
    <bookid>5AFEE130174076E3</bookid>
    <path>/ document[1]/page [5]</path>
    <rank>3</rank>
  </result>
    ...
</topic>
<topic> ... </topic>
</bs-submission>
```

6 How to submit your runs

Runs should be uploaded to http://booksearch.org.uk. Login using your INEX credentials and go to the Upload Area. Please note that no online validation of your runs is offered, so please make sure that your runs conform to the appropriate DTD and that all XPaths (see Appendix A) are valid. Please check the INEX Book Track website for submission deadline.

Appendix A: XPaths and Passages

XPath

XML element and book page paths should be given in XPath syntax[1]. 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>/document[1]/page[4]/section[2]</path>
```

This path identifies the XML element, which can be found if we start at the document root, select the first "document" element, then within that, select the fourth "page" element, within which we select the second "section" element.

Please note that XPath counts element nodes **starting with 1** and takes into account the element type. For example, if a "page" element has a title and two sections then both the title and the first section elements would be indexed with 1 (since they are different element types). Their XPaths would be given as:

/document[1]/page[1]/title[1],
/document[1]/page[1]/section[1], and
//document[1]/page[1]/section[2].

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 <b> element, and then again a newline.

When constructing passages, any whitespace that represents the **only** textual content of a text node should be ignored.

[1] Clark, J. and DeRose, S. 1999. XML Path Language (XPath) version 1.0. W3C Recommendation. http://www.w3.org/TR/xpath.