

The Impact of Query Length and Document Length on Book Search Effectiveness

Mingfang Wu, Falk Scholer, and James A. Thom

RMIT University, Melbourne, Australia
{mingfang.wu,falk.scholer,james.thom}@rmit.edu.au

Abstract. This paper describes the RMIT group's participation in the book retrieval task of the INEX booktrack in 2008. Our results suggest that for book retrieval task, using a page-based index and ranking books based on the number of pages retrieved may be more effective than directly indexing and ranking whole books.

1 Introduction

This paper describes the participation of the RMIT group in the Initiative for the Evaluation of XML retrieval (INEX) book search track in 2008, specifically the book retrieval task. Book search is an important track at INEX – books are generally much larger documents than the scientific articles or the wikipedia pages that have been used in the main ad hoc track at INEX. The book corpus as provided by Microsoft Live Book Search and the Internet Archive contains 50 239 digitized out-of-copyright books. The contents of books are marked up in an XML format called BookML [1]. The size of this XML marked up corpus is about 420 Gb. With book retrieval, structure is likely to play a much more important role than in retrieval from collections of shorter documents.

This is the first year of RMIT's participation in the book search track at INEX, and we explore the effectiveness of book retrieval by experimenting with different parameters, namely: the length of queries and the length of documents being indexed and retrieved. We begin by describing our approach in the next section, which is followed by our results, and then the conclusion.

2 Our Approach to Book Retrieval Task

The book retrieval task investigates on a typical scenario in which a user searches for relevant books on a given topic with the intent to build a reading or reference list. With this task, we attempted to explore the following research questions.

1. What is a suitable length for a query? Does adding more query terms improve retrieval effectiveness?
2. What is the most suitable index and search granularity? Should we treat the whole book or a just a section or page of a book as a searchable document unit?

To answer these research questions, we experimented with various query construction strategies, two different units of searchable document, as well as the combinations of query length and searchable document unit. We undertook 12 runs in total, of which 10 official runs were submitted into the pool for evaluation.

2.1 Query Construction

The test collection has a total of 70 test topics that were contributed by participating groups in 2007 and 2008. Participating groups also contributed to the relevance assessment of these topics. At the time of writing, there are 25 (out of 70) topics assessed (or partially assessed), with at least one relevant book being identified. Thus our evaluation is based only on the completed assessments for these 25 topics (this is the version of the book track assessments known as v250209 from 25th February 2009).

```
<inex_topic track="book" task="book-retrieval/book-ad-hoc"
              topic_id="41" ct_no="2008-13">
<title>
    Major religions of the world
</title>
<description>
    I am interested to learn about the origin of the world's major
    religions, as well as their commonalities and differences.
</description>
<narrative>
    <task>
        Having met people from different cultural and religious
        background, I am keen to learn more about their religions
        in order to understand them better.
    </task>
    <infneed>
        A concise book that has direct comparison of the world's
        popular religions would be an ideal pick. If this book can
        not be found, multiple books about the origin and believes
        of each religion should also be acceptable.
    </infneed>
</narrative>
</inex_topic>
```

Fig. 1. A sample topic

Topics from the book track collection contain three components: title, description and narrative. The narrative component is further decomposed into two sub-components: task and information need. An example of such a topic is shown in Figure 1. As we can see, the title, description and information need components provide specificity of a search context in an increasing order: the

title mostly represents a query that a user would type into a search box, the description provides more details of what information is required, while the task and the information need depict the context of the potential use to be made of the searched books, and the criteria on which books should be counted as being relevant or irrelevant.

Generally, long queries describe information needs more specifically than those short ones [3]. It has also been reported that long and short queries perform differently [2]. We set out to investigate, given a book collection where the length of a book is much longer than other types of documents such as a newswire article or a web page from TREC test collection, whether long queries would perform better than short queries. So we explored the following four approaches to constructing queries from different components of a topic.

Title: Use all words from the topic title element.

Title+Infneed: Use all words from the topic title and the topic information need element.

TitleBoolean: Boolean operator “AND” is inserted between query words as in **Title**.

Title+InfneedManual: Use all words as in **Title**, and add some manually selected words from the information need element.

Consider the sample topic shown in Figure 1. Applying our four different approaches results in the following queries (after stopping and stemming):

Title: major religion world

TitleBoolean: major AND religion AND world

Title+Infneed: major religion world concis book direct comparison world popular religion ideal pick book found multipl book origin believ religion accept

Title+InfneedManual: major religion world direct comparison world popular religion

The average query length for the set of 25 assessed topics is 2.6 terms for the set of **Title** (or **TitleBoolean**) queries, 25.3 terms for the set of **Title+Infneed** queries, and 13.4 terms for the set of **Title+InfneedManual** queries.

2.2 Index construction and runs

We used the Zettair search engine¹ for indexing and searching in all of our submitted runs. After preprocessing into documents but before indexing, we removed off all XML tags, leaving with a corpus of about 30GB for indexing. It took about 2 hours elapsed time to create an index on a shared 4 CPU (2.80GHz) Intel Pentium running Linux. For retrieval, we applied the BM25 similarity function [4] for document weighting and ranking (with $k1 = 1.2$ and $b = 0.75$). During indexing and searching, words from a stoplist were removed and the Porter stemmer was applied. We created separate indexes based on book-level evidence and page-level evidence.

¹ <http://www.seg.rmit.edu.au/zettair/>

Ranking Based on Book-level Evidence

The book-level index treated each book as a document. We sent the four sets of queries (**Title**, **TitleBoolean**, **Title+Infneed**, and **Title+InfneedManual**) to the search engine using this index, generating four book-level runs.

RmitBookTitle
RmitBookTitleBoolean
RmitBookTitleInfneed
RmitBookTitleInfneedManual

Ranking Based on Page-level Evidence

On average, a book from the collection contains 36 455 words. Sometimes, a topic is only mentioned in passing in a book, and may not be the main theme of the book. In order to promote those books dedicated primarily to the topic, we require a topic be mentioned in most parts of a book. We therefore experimented to break a book down into pages by using the “page” tag and constructed a corpus in which each page was treated as a document.

There are a total of 16 975 283 pages (documents) in this page corpus. Both book and page collections have 1 831 505 097 indexable terms of which 23 804 740 are distinct terms. The average document length is 589 370.9 bytes per book and 1 793.6 per page, and the average number of terms in each document is 102 041 per book and 302 per page.

We used the following two methods to estimate a book’s relevance to the topic.

1. In the first page-level evidence ranking method, we first retrieve the top 3 000 pages and then rank books according to percentage of pages retrieved per book.
2. The second page-level evidence ranking method is similar to the first one but ranks books based on the maximum number of continuous pages retrieved from the book as a percentage of the total number of pages in the book.

Combing these two page-level evidence ranking methods and our four query types, we had another eight runs as follows:

RmitPageMergeTitle: Query terms are the same as in **RmitBookTitle**, books are ranked according to the method 1;
RmitConPageMergeTitle: Query terms are the same as in **RmitBookTitle**, books are ranked according to the method 2;
RmitPageMergeTitleBoolean: Queries are the same as in **RmitBookTitleBoolean**, books are ranked according to the method 1;
RmiConPageMergeTitleBoolean: Queries are the same as in **RmitBookTitleBoolean**, books are ranked according to the method 2;
RmitPageMergeTitleInfneed: Query terms are the same as in **RmitBookTitleInfneed**, books are ranked according to the method 1;

RmitConPageMergeTitleInfneed: Query terms are the same as in **RmitBookTitleInfneed**, books are ranked according to the method 2.

RmitPageMergeTitleManual: Query terms are the same as in **RmitBookTitleInfneedManual**, books are ranked according to the method 1;

RmitConPageMergeTitleManual: Query terms are the same as in **RmitBookTitleInfneedManual**, books are ranked according to the method 2.

3 Results

Table 1 shows performance of twelve runs as measured in precision at 5 (P@5), 10 (P@10) and 20 (P@20) books retrieved, average interpolated precision averages at 0.00 and 0.10 recall, and MAP (mean average precision). In what follows, we make observations about both query type and length, and document type and length, based on the measure P@5.

Table 1. Experimental result for the book search task (the additional runs in italics were not included in the pool of submitted runs).

Run ID	P@5	P@10	P@20	MAP	incl_prn 0.00	incl_prn 0.10
RmitBookTitle	0.128	0.104	0.094	0.075	0.247	0.220
RmitBookTitleBoolean	0.128	0.104	0.049	0.075	0.247	0.220
RmitBookTitleInfneed	0.136	0.100	0.086	0.067	0.331	0.200
RmitBookTitleInfneedManual	0.112	0.108	0.088	0.068	0.276	0.187
RmitPageMergeTitle	0.144	0.116	0.084	0.074	0.302	0.260
RmitPageMergeTitleBoolean	0.144	0.116	0.084	0.074	0.302	0.260
<i>RmitPageMergeTitleInfneed</i>	0.168	0.108	0.090	0.079	0.358	0.291
RmitPageMergeTitleInfneedManual	0.216	0.132	0.098	0.106	0.367	0.346
RmitConPageMergeTitle	0.104	0.072	0.064	0.050	0.241	0.202
RmitConPageMergeTitleBoolean	0.104	0.072	0.064	0.050	0.241	0.202
<i>RmitConPageMergeTitleInfneed</i>	0.104	0.072	0.046	0.039	0.224	0.130
RmitConPageMergeTitleInfneedManual	0.128	0.084	0.058	0.054	0.279	0.213

Query Type and Length

We observe the following trends regarding queries.

- The three runs with Boolean queries (RmitBookTitleBoolean, RmitPageMergeTitleBoolean and RmitConPageMergeTitleBoolean) have almost the same performance as their corresponding runs without Boolean operators (RmitBookTitle, RmitPageMergeTitle and RmitConPageMergeTitle). This might be because the topic title is typically short (average of 2.6 terms), indeed 7 out of the 25 topics have only one query term.

- When a whole book is treated as a document, including the information need in the queries (RmitBookTitleInfneed) has a small improvement of 6.3% over just using the topic title as the queries (RmitBookTitle). However, manually adding terms from the information need (RmitBookTitleInfneedManual) is worse than the plain title queries.
- When a page of a book is treated as a document, the two runs with manually added terms from the information need (RmitPageMergeTitleInfneedManual and RmitConPageMergetitleInfneedManual) performed better than their corresponding runs with the title only queries (RmitPageMergeTitle and RmitConPageMergeTitle) (by 50% and 23.0% for ranking method 1 and 2 respectively), and the corresponding runs with queries including the information need as well as the title (RmitPageMergeTitleInfneed and RmitConPageMergeTitle) (by 28.6% and 23.0% for ranking method 1 and 2 respectively).
- The average length of the queries that added all the terms from the information need to the title (Title+Infneed) is almost double the length of the queries where the added terms were manually selected from the information need (Title+InfneedManual). This might be an explanation for why the runs with the Title+Infneed queries worked better for documents of book length, while the Title+InfneedManual queries worked better for documents of page length.

Document Type and Length

We observe the following trends regarding document ranking.

- The first page-level evidence ranking method (run IDs starting with RmitPage) has the best performance regardless of query type. The four runs from this method improved over counterpart book runs by 12.5%, 12.5%, 23.5% and 92.8% respectively. In particular, the run where terms from the information need were manually added to title (RmitPageMergeTitleInfneedManual), improved the performance over the base run (RmitBookTitle) by 68.8%. Incidentally, this run also performed the best amongst all submitted runs across all participating groups in terms of the measures MAP, P@5, P@20 and incl_prn0.10.
- The second page ranking method (run IDs starting with RmitConPage) gives worse performance for almost every query type, except the run RmitConPageMergeTitleInfneedManual, which is better than its corresponding book run RmitBookTitleInfneedManual by 14.3%.

4 Concluding Remarks

This paper has reported the results of the RMIT group's participation in the INEX book search track in 2008. We explored the impact of variation of query length and document size on the effectiveness of the book retrieval task. Based on

the current relevance assessments, the evaluation shows that treating a page of a book as a searchable unit and ranking books based on the percentage of pages retrieved performs better than indexing and retrieving whole book as a search unit. The search performance can be further improved by adding additional query words that describe information need.

This work provides a baseline for further experiments in structured information retrieval, in particular developing new approaches to book retrieval and in exploring other tasks such page-in-context.

References

1. G. Kazai, A. Doucet and M. Landoni. Overview of the INEX 2008 book track. In *INEX 2008 Workshop Proceedings*. 2008
2. G. Kumaran and J. Allan. A case for shorter queries, and helping users create them. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 220–227, 2006.
3. N. Phan, P. Bailey and R. Wilkinson. Understand the relationship of information need specificity to search query length. In *Proceedings of SIGIR 2007*, pages 709–710, 2007.
4. S. Robertson, S. Walker, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In *Proceedings of TREC-3*, pages 109–126. Available online at trec.nist.gov/pubs/, 1994.