SOCIAL BOOK SEARCH TRACK: ISM@CLEF'16 SUGGESTION TASK

Ritesh Kumar, Guggilla Bhanodai and Rajendra Pamula

Department of Computer Science and Engineering,
Indian School of Mines Dhanbad, 826004
India
{ritesh4rmrvs,bhanodaig,rajendrapamula}@gmail.com

Abstract. This paper describes the work that we did at Indian School of Mines towards Social Book Search Track for CLEF 2016. As per requirement of CLEF-2016 we submitted six runs in its Suggestion Task. We investigated individual effect of *title*, *group*, *request*, as well as combined effect of *title, request and group* fields of the topics in our runs. For all the runs we used language modeling technique with Dirichlet smoothing. The run using combined effect of *title, request and group* field was our best. Overall, our performance is good but it needs some improvement, our scores are encouraging enough to work for better results in future.

Keywords: Book Search, Social Book Search, Language modeling, Information Retrieval, re-ranking, Normalization

1 Introduction

With growing numbers of online portals and book catalogues, our current time sees a rapid evolution in the way we acquire, share and use books. In order to enable users for searching the relevant books, Social Book Seach Track at CLEF [5] provides a relevant experimental platform to investigate techniques of searching and navigating professional metadata. These metadata are provided by publishers/booksellers and user-generated content from social media [1]. In CLEF 2016 at Social book Search Lab, they offered three different tracks: Suggestion Track, Interactive Track and Mining Track. We participated in the suggestion track where we were supposed to recommend books based on user's request and her personal catalogue data (list of books with rating and tags maintained for the user in the social cataloguing site). We were also provided with a large set of anonymised user profiles from LibraryThing forum members, consisting of almost 93,976 anonymised user profiles from LibraryThing with over 33 million cataloguing transaction. Each user request is provided in the form of topics containing different fields like *title*, *request*, *group*, *examples* and catalogue information.

Our goal is to investigate the contribution of different topic fields as well as combining effect of some fields for book recommendation. We only considered *title*, *request*, *group* fields from each topic.We did not consider topic-creator's

catalogue information nor did we consult the user profiles.
We submitted six runs (ISMD16allfieds, ISMD16titlefield, ISMD16requestfield, ISMD16titlewithoutreranking, similaritytitlefieldreranked, ISMD16groupfield) in the Suggestion Task. For all the runs, Language modelling with Dirchlet smoothing was used in Lemur's Indri search system [3].
The organization of rest of the paper is as follows. Section 2 describes about dataset. we describe our methodology: field categories and indexing, which document and topic fields we used for retrieval in section 3. Section 4 describes what approaches we have used, Section 5 reports results. In Section 6 we analyse our results. Finally, we conclude in Section 7 with directions for future work.

2 Data

The test collection provided by CLEF 2016 SBS orgainzers for Suggestion Task had a document collection and a topicset. The document collection consists of 2.8 million book description with metadata from Amazon and LibraryThing. In Amazon there is formal metadata like booktitle, author, publisher, publication year, library classification codes, Amazon categories, similar product information and user-generated content in the form of user ratings and reviews. In Amazon, there are user tags and user-provided metadata on awards, book characters, locations and blurbs. There are additional records from the British Library and the Library of Congress. The entire collection was 7.1 GB in size [2].

The topic-set contains 120 topics each describing a user's request for suggestion of books. Each topic has a set of fields like *title*, *request*, *group*, *example* and user's personal catalogue at the time of topic creation. The catalogue contains a list of book-entries with information like LibraryThing id of the book, its entry-date, rating and tags.

The organizers also supplied 94,000 anonymised user profiles from LibraryThing.

3 Methodology

3.1 Field categories and Indexing

We are provided by Amazon/LibraryThing data collection(corpus) which consists of 2.8 million book descriptions with metadata. There are so many fields in the corpus, we took some of them for indexing which are as follow:
Metadata In our metadata index, we used these metadata field: ⟨title⟩ ⟨creator⟩, ⟨firstwords⟩, ⟨lastwords⟩.
Content In our content index, we used these metadata field:⟨content⟩ of provided corpus containing, ⟨blurbs⟩, ⟨epigraph⟩, ⟨quotation⟩.
Tags In our tags index, we used ⟨tags⟩ field for indexing.
Reviews In our reviews index, we used ⟨reviews⟩ field from corpus.

3.2 Topics

This year's Suggestion task has provided 120 topics, With help of these we built four set of queries which are:
Topic-Title: Only the⟨**title**⟩ field of each topic.
Topic-Request: It contains only the ⟨**request**⟩ field.
Topic-group: Only the ⟨**group**⟩ field.
Topic-All-Fields: It contains ⟨**title**⟩, ⟨**request**⟩, ⟨**group**⟩ field.

4 Approach

In our approach we analyzed two methods first one i.e. Content Based Retrieval and secondly re-ranking approach after rank normalization of the scores of the retrieved documents. For both retrieving approaches we used Language modeling with Dirichlet smoothing. The document collection provided was stopword-removed using **SMART** stop word list and then stemmed using Krovetz stemmer. We did not remove stopwords from provided topics. For retrieving and indexing we used Lemur 5.9 search system. We also removed punctuation marks from all the textual content of these fields and used only free text queries in all the runs.We did not consider any other information like catalogue information and user profile during retrieval. For each topic, we submitted up to 1000 book suggestions in the form of ISBNs.

4.1 Content Based Retrieval

During retrieval, we tried to see the effect of different components of a topic one by one as well as combined contribution of all the topics except ⟨**example**⟩ field. It is simply based on adhoc retrieval. We can see the result given in Table 1.

Table 1. Results of content based retrieval for different runs using nDCG@10. Best performing run for overall topic is given in bold letter

Topic List				
Documents Field	title	request	group	allfields
Metadata	0.0531	0.0478	0.0201	**0.0621**
Content	0.0510	0.0432	0.0191	0.0423
Tags	0.0507	0.0457	0.0312	0.0542
Reviews	0.0478	0.0367	0.0010	0.0592

4.2 Re-ranking

In this method we are inspired by Social Feature Re-ranking Method proposed by Toine Bogers in 2012 [6]. In order to improve the initial ranking, we perform re-ranking by two different strategies after analyzing the structure of XML: Item-Rerank (I) and RatingReview-Rerank (R), For re-ranking we have used following stages:

Similarity Calculation: The similarity of two documents based on feature I is calculated by equation (1)

$$sim_{ij}(I) = \begin{cases} 1: \ i \ is \ j's \ similar \ product \ or \ j \ is \ i's \ similar \ product \\ 0: \ otherwise \end{cases} \tag{1}$$

$$score'(i) = \alpha \cdot score(i) + (1 - \alpha) \cdot \sum_{j=1}^{N} sim_{ij} \cdot score(j)(j \neq i) \tag{2}$$

Re-Ranking: We re-rank the top 1000 list of initial ranking for the above mentioned features by Equation (2).

$$score'(i) = \alpha \cdot score(i) + (1-\alpha) \times log(|reviews(i)|) \times \frac{\sum_{r \in R_i} r}{|reviews(i)|} \times score(i) \tag{3}$$

For feature R, we use Equation (3) [7].

Before re-ranking we apply rank normalization on the retrieved results to map the score into the range [0, 1] [8]. The balance between the original retrieval score, score(i) and the contributions of the other books in the results list is controlled by the α parameter, which takes values in the range [0, 1], but in our experiment we have taken fixed value i.e. α= 0.96. Due to lack of time, we couldn't try with any other value.

5 Results

The scores obtained by our six runs are given in Table 2. The official evaluation measure provided by CLEF'16 is nDCG@10 [4]. The performance of our runs are in decreasing order. Our best performance is by ISMD16allfieds where we use *title,request and group* field. We also show the best score in the task demonstrated by run-id **run1.keyQuery_active_combineRerank**(*), for the sake of comparison.

Table 2. Results - The official evaluation Measure by CLEF 2016

RUN ID	Rank	MRR	**nDCG@10**	MAP	R@1000
ISMD16allfieds	24	0.1722	**0.0765**	0.0342	0.2157
ISMD16titlefield	28	0.1197	0.0639	0.0333	0.1933
ISMD16requestfield	29	0.1454	0.0613	0.0287	0.1870
ISMD16titlewithoutreranking	33	0.1114	0.0542	0.0386	0.2556
similaritytitlefieldreranked	35	0.0966	0.0445	0.0307	0.1933
ISMD16groupfield	43	0.0527	0.0104	0.0069	0.0564
*best**	1	0.5247	0.2157	0.1253	0.3474

6 Analysis

Although our performance is not up to the mark, there are few take-home lessons. In our run_id:**ISMD16allfieds**, **ISMD16titlefield**, **ISMD16requestfield** and **ISMD16groupfield**, we have reranked the retrieved score based on reviews(R) by taking α=0.96.
In our top score i.e. **ISMD16allfieds**, we have taken combination of all the fields *title, request, group* except *example* field from topic, In ISMD16titlefield, we have taken only*title* field, In ISMD16requestfield we have taken only field of the topic, For ISMD16groupfield we have taken only *group* field. For run_id: **ISMD16titlewithoutreranking** we simply used as content based retrieval. For run_id: **similaritytitlefieldreranked** we have used similarity as well as reranking by taking $\alpha = 0.96$.

7 Conclusion

This year we participated in the Suggestion Task of Social Book Search. We tried to see the individual effect as well as combined effect of different topic-fields on book recommendation. We considered only a handful of fields like *request*, *title*, *group* etc from the topics. While there can be no denial of the fact that our overall performance is average, initial results are suggestive as to what should be done next. We need to consult other fields like book catalogue of the topic creators, ratings of the books in the catalogue during retrieval. We also need to take into account profiles of other users. It is also imperative to see the learning to rank for different fields, and taking the α parameter range between [0,1], this time we have taken fixed vale of α= 0.96. We will also use other fields in user catalogues and user profiles. We shall be exploring some of these tasks in the coming days.

References

1. Marijn Koolen, Gabriella Kazai, Jaap Kamps, Michael Preminger, Antoine Doucet and Monica Landoni, *Overview of the INEX 2012 Social Book Search Track.*

INEX'12 Workshop Pre-proceedings, Shlomo Geva, Jaap Kamps, Ralf Schenkel (editors), September 17-20, 2012, Rome , Italy.
2. INEX, Initiative for the Evaluation of XML Retrieval. https://inex.mmci.uni-saarland.de/data/documentcollection.jsp
3. INDRI: Language modeling meets inference networks, Available at http://www.lemurproject.org/indri/
4. Jarvelin, K., Kekalainen, J.: Cumulated Gain-based Evaluation of IR Techniques. ACM Transactions on Information Systems 20(4) (2002) 422-446.
5. CLEF, Conference and labs of the Evaluation Forum. http://clef2016.clef-initiative.eu/index.php
6. T. Bogers and B. Larsen. Rslis at inex 2012: Social book search track. In INEX'12 Workshop Pre-proceedings, pages 97-108. Springer, 2012.
7. R. D. Ludovic Bonnefoy and P. Bellot. Do social information help book search? In INEX'12 Workshop Pre-proceedings, pages 109-113. Springer, 2012.
8. Renda, M.E., Straccia, U.: Web Metasearch: Rank vs. Score-based Rank Aggregation Methods. In: SAC 03: Proceedings of the 2003 ACM Symposium on Applied Computing, New York, NY, USA, ACM (2003) 841846