USTB at Social Book Search 2016 Suggestion Task: Active Books Set and Reranking

Shao-Hui Feng, Bo-Wen Zhang*, Zan-Xia Jin, Xu-Cheng Yin*, Jian-Lin Jin, Jian-Wei Wu, Le-Le Zhang, Hao-Jie Pan, Fan Fang, and Fang Zhou

Department of Computer Science and Technology,
University of Science and Technology Beijing (USTB), Beijing 100083, China
shaohui_feng@xs.ustb.edu.cn
bowenzhang@xs.ustb.edu.cn
xuchengyin@ustb.edu.cn

Abstract. In this paper, we describe our participation in the Social Book Search(SBS) Suggestion Task. We developed some new re-ranking models, based on what we applied in last year. We used Galago search from the index which was built with active books. The queries input to search engine were composed by key words from topics, and then we performed re-ranking models(popularity related) on Galago searching results on enriched XML index by 14 different fields. Experiments on these approaches shown that an enriched index and key query model improves the effectiveness. As our approaches in INEX2014 [1],SBS2015 [2] and combined those re-ranking models according to a specific order shown the best performance.

Keywords: XML retrieval, key query, re-ranking , active books set, popularity

1 Introduction

In this paper, we describe our participation in the Social Book Search 2016 suggestion task. Our goals for this task were (1) to investigate the contribution of key words from topics in searching; (2) to testify the active books set functioning in searching (3)to testify the effect of popularity related re-ranking approaches (4) to find an effective approach to combine the results of different re-ranking models.

The structure of this paper is as follows. We start Section 2 by describing our methodology: pre-processing on the XML formatted documents, indexing, searching by Galago, introduction of key query and active books set. In the section 3,we describe the re-ranking models and the re-ranking models experiments. In section 4, we describe the results of our enriched index, key query model, active books set and re-ranking models. Section 5 describes about the runs which we submitted,with the results of those runs presented. We discuss our results and conclude in Section 6.

2 Methodology

2.1 Data Pre-Processing

We perform a process similar to [1], such as expand and enrich the documents XML with replacing the numeric information with textual information. For instance:

$$<tag\ count="3">fiction</tag>$$

We change it to

$$<tag>fiction\ fiction\ fiction</tag>.$$

In addition, this time we expand to 14 fields when we clean the XML documents, it is different from last year. They are

$title, isbn, tags, review-content, review-summary, dewey, firstwords,$
$lastwords, chracters, places, subjects, browsenodes, abstract, addcontent.$

2.2 Indexing

Galago [1] is an open-source search engine. In order to improve the search effectiveness, we study two strategies to build the index. One indexing strategy is the normal indexing approach described as follows. Experimentally, we find that the fields (etc. the title, tag, content and summary) are more relevant and meaningful than others in the XML formatted documents. So we build our basic index by removing the other fields which were not useful. Another strategy is to enrich the basic index. Observing the book information from the Library Thing, we find out that a large proportion of books lack the content and summary fields. Therefore, documents expansion technology is expected to utilized to enrich the basic index. Firstly, we select two web sites which contain a large amount of more useful metadata of books. The books we use are the literatures written in English in douban.com [2] and all books in lookupbyisbn.com. Then we crawl the brief introduction of douban.com and the book description field of lookupbyisbn.com. Both web sites are available by ISBN. With the content from both web sites, we enrich six hundred thousand of books (see the examples of book document which is used for index in XML 1 and XML 2). The enriched index is based on the enriched information.

XML 1: Book document

```
<book>
<title>Mister Monday</title>
<summary>So good, you can't put it down!</summary>
<content>Now, I had...</content>
<tag count="9">children's literature</tag>
</book>
```

[1] http://www.galagosearch.org/
[2] http://book.douban.com/

XML 2: Enriched book document

```
<book>
<title>Mister Monday</title>
<summary>So good, you can't put it down!</summary>
<content>Now, I had...</content>
<tag count="9">children's literature</tag>
<brief introduction>the content is from the douban.com</brief introduction>
<description>the content is from the lookupbyisbn.com</description>
</book>
```

2.3 Generate Keywords from queries

We collect all the topics from 2011 to 2015. Our purpose is to get the best evaluation results after a great quantity of attempts. So that we can get the best queries for these topics, then we summed up a word list for topic queries. In Social Book Search 2016, we used the word list to filter the request field, then combine it with the title field compose the queries for Galago.

2.4 Active subset in book collections

Through the analysis of the profile, we select a subset of high frequent books which is called active book set. In this set, all the books are considered much more should be recommended to the topic author. We filter out all the books which are not in the active set. The operation is similar to filter the catalog and example books of the topic.

3 Re-ranking Models

Some of the re-ranking approaches were proposed and used by USTB at INEX2014 [1] and proposed by Toine Bogers in 2012 [4], which proved to be effective. This year our re-rank approach can be roughly divided into two categories 1) *Tag-Rerank* (T), *similar product re-rank* (S), *Read by one re-rank* (R) 2) *popularity re-rank* (P), *example recommended re-rank* (E), *Reader number re-rank*(R), *Browsnodes re-ranking*(B). The first category was based on computing the similarity of two books that from the original result from Galago. The second category was based on attributes (*number of readers, popularity*) of the books or the example books of the topics.

We use these models to re-rank by the following stages:

1)***Similarity Calculation***. Models like T, N focus on the field <tag> and <BrowseNode>. We can build a feature matrix for features for each field. Equation (1) is used to calculate the T, N, TN similarities of two documents.

Features like I focus on the field <similar-product>, the similarities of two documents based on the feature I is calculated by the Equation (2).

$$sim_{ij}(f) = \cos < \overrightarrow{f_i}, \overrightarrow{f_j} >= \frac{\overrightarrow{f_i} \cdot \overrightarrow{f_j}}{|\overrightarrow{f_i}||\overrightarrow{f_j}|} \tag{1}$$

$$sim_{ij}(I) = \begin{cases} 1, & i \text{ is } j\text{'s similar product or} \\ & j \text{ is } i\text{'s similar product} \\ 0.5, & i \text{ is } j\text{'s similar product's similar product} \\ & \text{or } j \text{ is } i\text{'s similar product's similar product} \\ 0, & else \end{cases} \quad (2)$$

2)***Re-ranking***

First category rerank We re-rank the top 1000 list of initial ranking for the above-mentioned features by Equation (3). For feature R, we use Equation (4) [6] and for B, we use Equation (5).

$$score'(i) = \alpha \cdot score(i) + (1-\alpha) \cdot \sum_{j=1}^{N} sim_{ij} \cdot score(j)(j \neq i) \quad (3)$$

$$score'(i) = \alpha \cdot score(i) + (1-\alpha) \times log(|reviews(i)|) \times \frac{\sum_{r \in R_i} r}{|reviews(i)|} \times score(i) \quad (4)$$

where R_i is the set of all ratings given by users for the document i, and $|reviews(i)|$ is the number of reviews.

$$score'(i) = \alpha \cdot score(i) + (1-\alpha) \times \frac{1 + BA(i)}{1 + BA_{max}} \times score(i) \quad (5)$$

where $BA(i)$ is the Bayesian average rating of document i, which can be referred to[7].

In addition, Tag re-rank and read by one re-rank (we hold the opinion that if two books are read by same reader, they are related) are very similar to similar product re-rank.

Second category reranking

Popularity re-rank approach
We got a data set from Library Thing like this:

$$Work\_id_i, \quad popularity_i$$

When we input initial search result to this model, we will change the works score by equation (6)

$$score'_i = \alpha * score_i + (1-\alpha) * score_i * (1 - popularity/30,0000) \quad (6)$$

Just as what we see, popularity is the number from the set, score is from Galago search result, score is the final output score from this re-rank modes.

Example recommended reranking approaches

Through summarizing the information from the profile, we got a example recommended list for some topics like this format:

$$Topic\_id_i \ recommended_1, recommended_2, recommended_3, etc.$$

When using the re-ranking approach, if a work of the result is contained in the topics list. We would change it rate by the equation (7)

$$score'_i = score_i + \alpha * score_i \tag{7}$$

if the work in the result was not contained in the *topic_id*'s list, $\alpha = 0$.

Reader number re-ranking approach

We extracted information from the profile, then we got a set which showing the workss reader number.Like this:

$$word\_id_i, \quad reader\ number_i$$

Through this approach, we optimize the search result by the below equation(8)

$$score'_i = \alpha * score_i + (1 - \alpha) * score_i * (reader\_num_i/1,000) \tag{8}$$

3) ***Combining***. We applied these approaches on the key query search result according to a specific order, then got the final result.

3.1 Re-Ranking Models Experiments

In order to choose the most effective feature and select the optimized parameter, in the first round, we trained our re-ranking models on SBS2015. The results were shown in Table 1.

Table 1. Training on SBS 2015 and Best α

Rerank model	SBS15	Bestα
keywords	0.1291	-
keywords+ active	0.1518	-
keywords +active +Similar product	0.1528	0.991
keywords +active +Example recommend	0.1581	0.384
keywords +active +Popularity	0.1538	0.09
keywords +active + Reader number	0.1709	0.706
keywords+ active + Read By One	0.1548	0.997
keywords +all ReRank +filter catalog and example	0.1972	-

As shown in Table 1, the best performance is obtained from $Initial+keywords+allReRank+filtercatalogandexampl$,and $active, readernumber$ make greater contributions to the improvements.

4 Submitted Runs

We selected six automatic runs for submission to SBS2016 based on our Key Query and re-ranking Models. They are:

run1. This run was made by a searching-re-ranking process where the initial retrieval result was based on the selection of query keywords and a small index of active books, the re-ranking results based on a combination of several strategies (number of people who read the book from profile, similar-product from amazon.com, popularity from LT forum, etc.).

run2. This run was made by a searching-reranking process where the initial retrieval result was based on the selection of query keywords and a small index of active books, the re-ranking results based on number of people who read the book from profile.

run3. This run was made by a searching-reranking process where the initial retrieval result was based on the selection of query keywords and a small index of active books, the re-ranking results based on the books in a same user's profile.

Run4. This run was made by a searching-reranking process where the initial retrieval result was based on the selection of query keywords and a small index of active books, the re-ranking results based on similar products provided by amazon.

run5. This run was made by a searching-reranking process where the initial retrieval result was based on the selection of query keywords and the full index filtered by active books, the re-ranking results based on the books in a same user's profile.

run6. This run was made by a searching-reranking process where the initial retrieval result was based on the selection of query keywords and the full index filtered by active books, the re-ranking results based on a combination of several strategies (number of people who read the book from profile, similar-product from amazon.com, popularity from LT forum, etc.).

5 Result

The runs submitted to the Social Book Search 2016 were evaluated using graded relevance judgments. The relevance value were labeled manually according to the behaviors of topic creators, for example, if creator adds book to catalogue after it's suggested, the book is treated as highly relevant. A decision tree was built to help the labeling 3. All runs were evaluated using NDCG@10, MRR, MAP, R@1000 with NDCG@10 as the main metric. Table 2 shows the official evaluation results. Results of the six submitted runs on Social Book Search 2016, evaluated using all 120 topics with relevance value calculated from the decision

tree. The best run scores are printed in bold, we got the first place in the competition. In addition, all the results we submitted are in the top six.

So we got the first place in the Social Book Search suggestion task 2016.

Table 2. Results of the five submitted runs on Social Book Search 2016, evaluate using all 120 topics with relevance value calculated from the decision tree. The best run scores are printed in bold

.

Run #	Run Description	NDCG@10	MRR	MAP	R@1000
1	run1.keyQuery_active_combineRerank	**0.2157**	**0.5247**	**0.1253**	**0.3474**
2	run2.keyQuery_active_userNumRerank	0.2047	0.4700	0.1177	0.3474
3	run3.keyQuery_active_readByOneReRank	0.1989	0.4923	0.1157	0.3474
4	run4.keyQuery_active_similarRerank	0.1935	0.4685	0.1106	0.3474
5	run5.keyQuery_readByOne	0.2009	0.4767	0.1128	0.3146
6	run6.keyQuery_AllRerank	0.2030	0.4868	0.1144	0.3146
7	Initial+stopwords	0.1265	-	-	-
8	Initial+keyQuery	0.1567	-	-	-
9	Initial+keyQuery+active	0.1943	-	-	-

It necessary to state that run 7, run 8 and run 9 are our additional experiments, obvious we can find that key query and active books set have a great increase in results .Key query improve *Initial+stopwords* run by about 25 percent, then active books set improves the *Initial+keyQuery* run by about 24 percent.We see that the best-performing run on all 120 topics was run1 with an NDCG@10 of 0.2157. Run 1 used Key Query, small index built by active books and all re-ranking models combine. Also we see that re-ranking model does improve over the initial results by Galago searching engine.

All the runs from 1 to 6 were filtered by the topics catalog books set and example books set.

6 Discussion & Conclusion

All of the re-ranking approaches can improve the evaluation results, the best results are from combined all re-ranking approaches. Of course, the use of key queries and active book set make the greatest contributions to the effectiveness of our systems.

This year, we used much information from profile, and we got a better performance, but we failed to make use of the random forest to combine the re-ranking to improve the result. We keep the opinion that machine learning can get a better results. So, it is worth discussing how to combining the re-ranking results with machine learning algorithms.

References

1. Bo-Wen Zhang, Xu-Cheng Yin, Xiao-Ping Cui, Bin Geng, Jiao Qu, Fang Zhou, Li Song and Hong-Wei Hao. USTB at INEX2014: Social Book Search Track. In INEX'13 Workshop Pre-proceedings. Springer, 2013.
2. Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum, Toulouse, France, September 8-11, 2015. 2015
3. T. Bogers and B. Larsen. Rslis at inex 2013: Social book search track. In INEX'13 Workshop Pre-proceedings. Springer, 2013.
4. T. Bogers and B. Larsen. Rslis at inex 2012: Social book search track. In INEX'12 Workshop Pre-proceedings, pages 97-108. Springer, 2012.
5. Bo-Wen Zhang, Xu-Cheng Yin, Xiao-Ping Cui, Bin Geng, Jiao Qu, Fang Zhou, Li Song and Hong-Wei Hao. Social Book Search Reranking with Generalized Content-Based Filtering. CIKM'14.
6. R. D. Ludovic Bonnefoy and P. Bellot. Do social information help book search? In INEX'12 Workshop Pre-proceedings, pages 109-113. Springer, 2012.
7. Marijn Koolen and J. Kamps. Comparing topic representations for social book search. In INEX'13 Workshop Pre-proceedings. Springer, 2013.