# USTB at Social Book Search 2015 Suggestion Task: Metadata Expansion and Reranking

Chun-Hua Zhao, Fang Zhou, Bo-Wen Zhang*, Xu-Cheng Yin*, Ming Hao, Zhi-Juan Zhang, and Tian-Tian Liu

Department of Computer Science and Technology,
University of Science and Technology Beijing (USTB), Beijing 100083, China
zhaochunhua1206@163.com
xuchengyin@ustb.edu.cn
bowenzhang.ustb@foxmail.com

**Abstract.** In this paper, we describe our participation in the INEX 2015 Social Book Search(SBS) Track Suggestion Task. We try out all possible groups of XML fields to find out the most effective group for relevance feedback model. We investigate the contribution of user-generated data and construct a social book search system based on several important techniques. Focus on the lack of important information of the majority books, we use document expansion by crawling book information from other two web sites which can enrich the index. And then we perform re-ranking on Galago searching results on enriched XML index by 11 different strategies and combine the results with learning to rank. Experiments on these methods show that an enriched index and query model improves the effectiveness. As our methods in INEX 2014 [1], re-ranking and Random Forests combining those re-ranking models show better performance.

**Keywords:** document expansion, social re-ranking, semantic search, learning to rank

## 1 Introduction

In this paper, we describe our participation in the INEX 2015 Social Book Search track suggestion task. Our goals for this task are (1) to investigate the contribution of textual information in the query; (2) using relevance feedback to enrich the index ; (3) modeling the query based on the important terms; and (4) using re-ranking based on different user-generated social features with Random Forest combing them.

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

# 2 Methodology

## 2.1 Data Pre-Processing

As we can referred to [2], there are several fields in the XML formatted documents shown meaningful numeric information which cannot be understood by searching engine, such as *<tag count="3">fiction</tag>* and *<dewey>519</dewey>*. According to the method from [2], we expand and enrich the XML formatted documents with replacing the numeric information with textual information.In this way, the XML element *<tag count="3">fiction</tag>* is replaced by the element *<tag>fiction fiction fiction</tag>*. And the XML element *<dewey>519</dewey>* is replaced by the element *<dewey>Probabilities & applied mathematics</dewey>*.

## 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 method describes as following. 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 rest useless fields content. Another strategy is to enrich the basic index. Observing the book information from the Library Thing, we find out that there are a large proportion of books lack of 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 Searching

To improve the query model, two strategies are concerned after analyzing the structure of query XML file: relevance feedback and filtering out capitalized words (words with the first letter in upper case). Different from the previous years, the topics of this year has an extra field <example>. This field can be considered as relevance feedback and the relevance item is the book title which is related in the field <LT_id>. We use the relevance feedback item according to the fields <hasRead> and <sentiment> under the field <example> by the following method. If the field <hasRead> is no and <sentiment> is positive, the relevance is defined to four which is the most relevant. If the field <hasRead> is yes or <sentiment> is negative, the degree of relevance is defined to zero which is the most irrelevant. We consider all the other situations of <hasRead> and <sentiment> are relevance and the relevance degree is one. The value of degree is calculated by the Equation (1). Having the above relevant information, the new query generates by adding the book title corresponding the field <LT_id> to the original galago query. A new query vector Q' is generated by the Equation (2).

$$degree = \begin{cases} 4, & \langle hasRead \rangle = no \ and \ \langle sentiment \rangle = postive. \\ 0, & \langle hasRead \rangle = yes \ and \ \langle sentiment \rangle = negtive. \\ 1, & else \end{cases} \tag{1}$$

$$Q' = \alpha \cdot Q + (1 - \alpha) \cdot \sum_{i=0}^{n} degree \cdot title \tag{2}$$

Where the $Q$ is the original query vector, the title is the relevance information vector, and the degree is the relevance degree.

Another query model improvement is about the field <narrative>. We filter out the words beginning with an upper letter which is exclude the first word of a sentence and add them to the original galago query, such as <narrative> I love alternative histories - two great ones I've enjoyed are Robert Harris's Fatherland and Kim Stanley Robinson's Years of Rice and Salt. Any other recommendations? John <narrative>. According to our method, we transform this field to <narrative> I've Robert Harris's Fatherland Kim Stanley Robinson's Years Rice Salt John <narrative>.

## 2.4 Re-ranking and Combining

Those re-ranking methods are proposed and used by USTB at INEX2014 [1] and proposed by Toine Bogers in 2012 [3], which proved to be effective. The re-ranking method is performed by 11 different models: Tag-Rerank ($T$), Item-Rerank ($I$), Deep-Rerank ($D$), Node-Rerank ($N$), RatingBayes-Rerank ($B$), Rating-Review-Rerank ($R$), Tag-Node-Rerank ($TN$), Item-Tag-Rerank ($IT$), Deep-Tag-Rerank ($DT$), Item-Tag-Node-Rerank ($ITN$), Deep-Tag-Node-Rerank ($DTN$). We use these 11 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 like $T$,$N$. The feature matrix of $TN$ is the connection of two matrices. Equation (3) is used to calculate the $T$, $N$, $TN$ similarities of two documents.

Features like $I$, $D$ focus on the field <similar-product>, the similarities of two documents based on the feature $I$ is calculated by the Equation (4).

$$sim_{ij}(f) = \cos < \overrightarrow{f_i}, \overrightarrow{f_j} >= \frac{\overrightarrow{f_i} \cdot \overrightarrow{f_j}}{|\overrightarrow{f_i}||\overrightarrow{f_j}|} \tag{3}$$

$$sim_{ij}(I) = \begin{cases} 1, & i \text{ } is \text{ } j\text{'s } similar \text{ } product \text{ } or \\ & j \text{ } is \text{ } i\text{'s } similar \text{ } product \\ 0, & else \end{cases} \tag{4}$$

The model Deep-Rerank ($D$) concerns similar products of similar products. So the values of elements in similarity matrix is calculated by the Equation 5[4].

$$sim_{ij}(D) = \begin{cases} 1, & sim_{ij}(I) = 1 \quad or \\ & \exists \quad k \neq i, \ k \neq j, \\ & s.t. \ sim_{ik}(I) = sim_{jk}(I) = 1. \\ 0, & else \end{cases} \tag{5}$$

As we know similarity matrices $SIM(I)$ and $SIM(D)$ are sparse, so we use the multi-feature like $IT$, $DT$, $ITN$, $DTN$ to fill-in. For example, the similarity based on feature $IT$ is calculated by Equation (6). The other similarities are calculated in the same way[4].

$$sim_{ij}(IT) = \begin{cases} 1, & sim_{ij}(I) = 1. \\ sim_{ij}(T), & else \end{cases} \tag{6}$$

2) *Re-ranking.* We re-rank the top 1000 list of initial ranking for the above-mentioned features by Equation (7). For feature $R$, we use Equation (8) [5] and for $B$, we use Equation (9).

$$score'(i) = \alpha \cdot score(i) + (1 - \alpha) \cdot \sum_{j=1}^{N} sim_{ij} \cdot score(j)(j \neq i) \tag{7}$$

$$score'(i) = \alpha \cdot score(i) + (1-\alpha) \times log(|reviews(i)|) \times \frac{\sum_{r \in R_i} r}{|reviews(i)|} \times score(i) \quad (8)$$

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

$$score'(i) = \alpha \cdot score(i) + (1-\alpha) \times \frac{1 + BA(i)}{1 + BA_{max}} \times score(i) \quad (9)$$

where $BA(i)$ is the Bayesian average rating of document $i$, which can be referred to [6].

3) *Combining.* We take Ranklib [3] as toolkit and use Coordinate Ascent, Random Forest and Rank Net as training models to train the models to combine features.

## 3 Experiments

In order to choose the most effective strategies and select the optimized parameter $\alpha$, in the first round, we train our query model on SBS 2011-13 and test on SBS2014. The results are shown in Table 1.

**Table 1.** Training on SBS 2011-13 and testing on SBS 2014

| Method | NDCG@10 (Training Set) | NDCG@10 (Testing Set) |
|---|---|---|
| initial | 0.1625 | 0.1386 |
| example | 0.1689 | 0.1407 |
| abstract | 0.1701 | 0.1422 |
| upper-narrative | 0.1700 | 0.1425 |
| rerank-RF | 0.1712 | 0.1429 |
| example-abstract | 0.1705 | 0.1427 |
| example-abstract-upper_narrative | 0.1721 | 0.1434 |
| abstract-upper_narrative | 0.1715 | 0.1429 |
| example-upper_narrative | 0.1713 | 0.1426 |
| example-abstract-rerank-RF | 0.1724 | 0.1438 |
| example-abstract-upper_narrative-rerank-RF | 0.1828 | 0.1542 |
| abstract-upper_narrative-rerank-RF | 0.1785 | 0.1476 |
| example-upper_narrative-rerank-RF | 0.1779 | 0.1468 |

[3] http://people.cs.umass.edu/~vdang/ranklib.html

## 4 Submitted Runs

Among all the methods, we select the best five automatic runs to submission which are based on our query and Re-ranking Models. The first one of these submitted runs is the result of <example> field using as relevance feedback information. The second one is the expansion query search in the enriched index result. The third one is based on the second result with the relevance feedback method. The fourth one is applied all Re-ranking strategies and combining them by Random Forest method result. The fifth one is combing the Re-ranking, Random Forest method and query model based on the field <example>

**Run 1** (example) This run takes Galago as toolkit and applies query model using the field <example> as relevance feedback information to search.

**Run 2** (Upper_narrative-abstract) This run applies query model based on the field <narrative> to search and uses the expansive index.

**Run 3** (UpperNar-abs-ex) This run applies query model based on the field <narrative> and <example> to search and uses the expansive index.

**Run 4** (Rerank-RF) This run applied all Re-ranking strategies and combining them by Random Forest method.

**Run 5** (Rerank-RF-example) This run applied all Re-ranking strategies and combining them by Random Forest method with query model based on the field <example>.

## 5 Results

The runs submitted to the INEX 2015 Social Book Search track are 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 is built to help the labeling [4]. All runs are evaluated using NDCG@10, MRR, MAP, R@1000 with NDCG@10 as the main metric. Table 2 shows the official evaluation results.

**Table 2.** Results of the five submitted runs on Social Book Search 2015, evaluate using all 208 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 |
|---|---|---|---|---|---|
| 5 | Rerank-RF-example | **0.106** | **0.232** | **0.068** | **0.365** |
| 4 | Rerank-RF | 0.088 | 0.189 | 0.056 | 0.359 |
| 3 | UpperNar-abs-ex | 0.079 | 0.197 | 0.052 | 0.312 |
| 2 | Upper_narrative-abstract | 0.061 | 0.155 | 0.042 | 0.309 |
| 1 | example | 0.042 | 0.120 | 0.022 | 0.029 |

[4] https://inex.mmci.uni-saarland.de/tracks/books/INEX14_SBS_results.jsp#mapping

It is obvious that among all the methods, the best-performing run on all 208 topics is run 5 with an NCDG@10 of 0.106. Run 5 uses relevance feedback based on the field ¡example¿ to expand the query and all re-ranking models with combining them by Random Forest. Again we see that re-ranking models and relevance-feedback do improve over the initial results by searching engine. Run 5, improves over the initial ranking by about 10%.

## 6 Discussion & Conclusion

On both training and the testing set the best results are from combining all re-ranking results in Random Forest and the relevance-feedback method. This shows a good use of social information can improve the results of Social Book Search. We fail to make use of the profile of topic creators to improve the results. It is worth discussing whether the information is useful or not.

## 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. T. Bogers and B. Larsen. Rslis at inex 2013: Social book search track. In INEX'13 Workshop Pre-proceedings. Springer, 2013.
3. T. Bogers and B. Larsen. Rslis at inex 2012: Social book search track. In INEX'12 Workshop Pre-proceedings, pages 97-108. Springer, 2012.
4. 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.
5. R. D. Ludovic Bonnefoy and P. Bellot. Do social information help book search? In INEX'12 Workshop Pre-proceedings, pages 109-113. Springer, 2012.
6. Marijn Koolen and J. Kamps. Comparing topic representations for social book search. In INEX'13 Workshop Pre-proceedings. Springer, 2013.
7. M. Koolen, G. Kazai, J. Kamps, A. Doucet, and M. Landoni. Overview of the inex 2012 books and social search track. In Focused Retrieval of Content and Structure, pages 1-29. Springer, 2012.
8. Kazai, G., Koolen, M., Kamps, J., Doucet, A., Landoni, M.: Overview of the INEX 2011 Book and Social Search Track. In: INEX 2011 Workshop pre-proceedings. INEX Working Notes Series (2011) 1136