USTB at INEX2014: Social Book Search Track

Bo-Wen Zhang, Xu-Cheng Yin*, Xiao-Ping Cui, Jiao Qu,
Bin Geng, Fang Zhou, and Hong-Wei Hao

Department of Computer Science and Technology,
University of Science and Technology Beijing (USTB), Beijing 100083, China
zbw292@126.com
xuchengyin@ustb.edu.cn

Abstract. In this paper, we describe our participation in the INEX 2014 Social Book Search(SBS) Track Suggestion Task. We investigate the contribution of user-generated data and construct a social book search system based on several important techniques. We perform re-ranking on Galago searching results on enriched XML index by 11 different strategies and combine the results with learning to rank. We find that 1) enriched index improves the effectiveness, 2) tag is the best-performed social feature on re-ranking and 3) Random Forest shows the best performance on combining in this case.

Keywords: XML retrieval, social re-ranking, semantic search, learning to rank

1 Introduction

In this paper, we describe our participation in the INEX 2014 Social Book Search track suggestion task. Our goals for this task were (1) to investigate the contribution of textual information in searching; (2) to examine the effectiveness of re-ranking based on different user-generated social features; and (3) to find a effective method to combine results of different re-ranking models.

The structure of this paper is as follows. We start in Section 2 by describing our methodology: pre-processing and on the documents XML, indexing and searching by Galago, re-ranking, combining with Learning-to-rank. In Section 3, we describe the results of our re-ranking models, including the different combining results based on different training 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 documents XML 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 Bogers, we expand and enrich the documents XML with replacing the numeric information with textual information.

2.2 Indexing and Searching

Galago [1] is an open-source search engine. The probability of the query content produced by language models are used to rank the documents. Based on the assumption that the priori probabilities of documents are the same, documents are ranked according to $P(Q|D)$, which is calculated by Equation (1), where $f_{q_i,D}$ means the amount of times the word/phrase q_i appear in document D. With Dirichlet Smoothing, the probability estimate is calculated by Equation (2). In this way, documents are scored by Equation (3).

$$P(Q|D) = \prod_{i=1}^{n} p(q_i|D) = \prod_{i=1}^{n} \frac{f_{q_i,D}}{|D|} \tag{1}$$

$$P(Q|D) = \prod_{i=1}^{n} p(q_i|D) = \prod_{i=1}^{n} \frac{f_{q_i,D} + \mu\frac{c_{q_i}}{|C|}}{|D| + \mu} \tag{2}$$

$$\log P(Q|D) = \log \prod_{i=1}^{n} p(q_i|D) = \sum_{i=1}^{n} \log \frac{f_{q_i,D} + \mu\frac{c_{q_i}}{|C|}}{|D| + \mu} \tag{3}$$

2.3 Re-ranking and Combining

These methods are inspired by Social Feature Re-ranking Method proposed by Toine Bogers in 2012 [3]. In order to improve the initial ranking, we perform re-ranking by 11 different strategies after analyzing the structure of XML: Tag-Rerank (T), Item-Rerank (I), Deep-Rerank (D), Node-Rerank (N), RatingBayes-Rerank (B), RatingReview-Rerank (R), Tag-Node-Rerank (TN), Item-Tag-Rerank (IT), Deep-Tag-Rerank (DT), Item-Tag-Node-Rerank (ITN), Deep-Tag-Node-Rerank (DTN). These methods includes the following stages:

1)*Similarity Calculation.* Features like I, D focus on the field <tag> and <BrowseNode>. For example , the tag vector of document i and j are $\overrightarrow{t_i} = [1, 0, 0]$ and $\overrightarrow{t_j} = [0, 0, 2]$. That means 1 user tag document i with tag 1 while 2 user tag document j with tag 3. In this way, we can build a feature matrix for features like T,N. The feature matrix of TN is the connection of two matrices. Equation (4) 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 (5).

$$sim_{ij}(f) = \cos < \overrightarrow{f_i}, \overrightarrow{f_j} >= \frac{\overrightarrow{f_i} \cdot \overrightarrow{f_j}}{|\overrightarrow{f_i}||\overrightarrow{f_j}|} \tag{4}$$

[1] http://www.galagosearch.org/

$$sim_{ij}(I) = \begin{cases} 1, & i \text{ is } j\text{'s similar product or} \\ & j \text{ is } i\text{'s similar product} \\ 0, & else \end{cases} \tag{5}$$

Considering the asymmetry, the method D concerns similar products of similar products. So the values of elements in similarity matrix is calculated by the Equation 6[1].

$$sim_{ij}(D) = \begin{cases} 1, & sim_{ij}(I) = 1 \;\; or \\ & \exists \;\; k \neq i, \; k \neq j, \\ & s.t. \; sim_{ik}(I) = sim_{jk}(I) = 1. \\ 0, & else \end{cases} \tag{6}$$

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 (7). The other similarities are calculated in the same way[1].

$$sim_{ij}(IT) = \begin{cases} 1, & sim_{ij}(I) = 1. \\ sim_{ij}(T), & else \end{cases} \tag{7}$$

2) Re-ranking. We re-rank the top 1000 list of initial ranking for the abovementioned features by Equation (8). For feature R, we use Equation (9) [7] and for B, we use Equation (10).

$$score'(i) = \alpha \cdot score(i) + (1 - \alpha) \cdot \sum_{j=1}^{N} sim_{ij} \cdot score(j)(j \neq i) \tag{8}$$

$$score'(i) = \alpha \cdot score(i) + (1 - \alpha) \times log(|reviews(i)|) \times \frac{\sum_{r \in R_i} r}{|reviews(i)|} \times score(i) \tag{9}$$

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

$$score'(i) = \alpha \cdot score(i) + (1 - \alpha) \times \frac{1 + BA(i)}{1 + BA_{max}} \times score(i) \tag{10}$$

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

3) Combining. We take Ranklib [2] as toolkit and use Coordinate Ascent, Random Forest and Rank Net as training models to train the models to combine features.

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

3 Experiments on Re-ranking Models

In order to choose the most effective feature and select the optimized parameter α, in the first round, we train our re-ranking model on SBS2011-2012 and test on SBS2013. The results are shown in Table 1.

Table 1. Training on SBS 2011-2012 and testing on SBS2013

Method	NDCG@10 (Training Set)	Best α	NDCG@10 (Testing Set)	α
Initial	0.1635	-	0.1383	-
Tag	0.1724	0.93	0.1456	0.93
Item	0.1701	0.94	0.1422	0.94
Deep	0.1700	0.96	0.1425	0.96
Node	0.1689	0.99	0.1407	0.99
RatingBayes	0.1645	0.97	0.1404	0.97
RatingReview	0.1712	0.98	0.1429	0.98
Tag-Node	0.1699	0.97	0.1418	0.97
Item-Tag	0.1697	0.96	0.1414	0.96
Deep-Tag	0.1696	0.95	0.1415	0.95
Item-Tag-Node	0.1698	0.98	0.1418	0.98
Deep-Tag-Node	0.1694	0.95	0.1410	0.95

As we can see from the table, the feature T shows the best performance with an improvement of 5.4%. All features shows improvements of different degree. So we use all features to combine the results. The results of three chosen training models are shown in Table 2. As can be seen from the table, Random Forest is the most effective model in this case.

Table 2. Results of learning-to-rank

Method	NDCG@10 (Testing Set)
Coordinate Ascent	0.1658
Random Forest	0.1614
Rank Net	0.1545

Then we train our model on SBS2011-2013. The results are shown in Table 3.

4 Submitted Runs

We selected six automatic runs for submission to INEX based on our Re-ranking Models and Similar Query Re-ranking method. One of these submitted runs was

Table 3. Training on SBS 2011-2013

Method	NDCG@10 (Training Set)	Best α
Initial	0.1486	-
Tag	0.1536	0.92
Item	0.1511	0.93
Deep	0.1510	0.96
Node	0.1490	0.98
RatingBayes	0.1489	0.96
RatingReview	0.1522	0.98
Tag-Node	0.1508	0.97
Item-Tag	0.1505	0.95
Deep-Tag	0.1505	0.93
Item-Tag-Node	0.1508	0.96
Deep-Tag-Node	0.1503	0.95

the initial ranking result. Another one was the tag re-ranking result. Another three were the different combining results based on three different learning-to-rank training model. The other one was Similar Query Re-ranking method, which is described in Run 6. Since the Random Forest was well-performed among all, the results of Random Forest was used to re-rank in Similar Query Re-ranking method. All runs used enriched new documents XML.

Run 1(newXml.feedback)This run took Galago as toolkit and used pseudo feedback to search.

Run 2(newXml.rerank_T)This run applied Re-ranking Model based on the field <tag>.

Run 3(newXml.rerank_all.L2R_Coordinate)This run applied all Re-ranking strategies and combining them by Coordinate Ascent method.

Run 4(newXml.rerank_all.L2R_RandomForest)This run applied all Re-ranking strategies and combining them by Random Forest method.

Run 5(newXml.rerank_all.L2R_RankNet)This run applied all Re-ranking s-trategies and combining them by Rank Net method.

Run 6(SimQuery.rerank_all.L2R_RandomForest)This run firstly applied all Re-ranking strategies and combining them by Random Forest method. As we know, sometimes users search topics similar to topics which used to appear. We bravely make a assumption that for two similar queries i,j, if document A is useful to query i, it's useful for query j too. A weight is multiplied to the normalization score of document D if 1) document A appears in the result list of query i; 2) query i and previous query j are similar to each other according to the calculation above; and 3) document D is useful for previous query j. The weight w is calculated by Equation $w = sim(q_i, q_j) * \frac{score(q_j,D)}{score(q_i,D))}$, where $score(q_j, D)$ and $score(q_j, D)$ are the normalization scores of document D with query i and j.

5 Results

The runs submitted to the INEX 2014 Social Book Search track were evaluated using graded relevance judgments. The relevance value were labeled manually according to the behaviours 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 4 shows the official evaluation results.

Table 4. Results of the six submitted runs on Social Book Search 2014, evaluated using all 680 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
6	SimQuery.rerank_all.L2R_RandomForest	**0.303**	**0.464**	**0.232**	**0.390**
4	newXml.rerank_all.L2R_RandomForest	0.142	0.258	0.102	0.390
5	newXml.rerank_all.L2R_RankNet	0.138	0.256	0.101	0.390
3	newXml.rerank_all.L2R_Coordinate	0.133	0.246	0.098	0.390
2	newXml.rerank_T	0.131	0.246	0.096	0.390
1	newXml.feedback	0.128	0.246	0.095	0.390

We see that, apart from Similar Query method, the best-performing run on all 680 topics was run 4 with an NCDG@10 of 0.142. Run 4 used all re-ranking models and combined them with Random Forest. Again we see that re-ranking model does improve over the initial results by searching engine. Run 4, improves over the initial ranking by about 10%. Run 6, from Similar Query method, have a best-performance just because there are many similar query topics in SBS 2014 with previous years.

6 Discussion & Conclusion

On both training and the testing set the best results are from combining all re-ranking results in Random Forest. This shows a good use of social information can improve the results of Social Book Search. The high evaluation value of Similar Query method indicates the amount of similar topics not the effectiveness of the model. We fail to make use of the catalog 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. Social Book Search Reranking with Generalized Content-Based Filtering. Submitted to CIKM'14.

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

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. 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
5. 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.
6. Marijn Koolen and J. Kamps. Comparing topic representations for social book search. In INEX'13 Workshop Pre-proceedings. Springer, 2013.
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.

Overview of the INEX 2012 Social Book Search Track - Hal. INEX2014: Tweet Contextualization Using Association Rules between Tweets are short messages that do not exceed 140 characters.Â To cite this version: Liana Ermakova, Josiane Mothe. IRIT at INEX 2014 : Tweet Contextualization Track. CLEF 2014 - Conference on Multilingual and Multimodal Information Access Evaluation, Sep 2014, Sheffield, United Kingdom. pp.0, 2014. HAL Id: hal-01156597 https://hal.archives-ouvertes.fr/hal-01156597 Submitted on 27 May 2015.