CERIST at INEX 2015: Social Book Search Track

Messaoud CHAA[1,2], Omar NOUALI[1]

[1] Research Center on Scientific and Technical Information
05 rue des 03 frères Aissou, Ben Aknoun, Alger, 16030
mchaa@cerist.dz
onouali@cerist.dz
[2] Université Abderrahmane Mira Béjaia
Rue Targa Ouzemour, Béjaïa 6000

Abstract. In this paper, we describe our participation in the INEX 2015 Social Book Search Suggestion Track (SBS). We have exploited in our experiments only the tags assigned by users to books provided from LibraryThing (LT). We have investigated the impact of the weight of each term of the topic in the retrieval model using two methods. In the first method, we have used the TF-IQF formula to assign a weight to each term of the topic. In the second method, we have used Rocchio algorithm to expand the query and calculate the weight of the tags assigned to the example books mentioned in the book search request. Parameters of our models have been tuned using the topics of INEX 2014 and tested on INEX 2015 Social Book Search track.

Keywords: Social book Search, TF-IQF, Tag-Based, Rocchio Algorithm, Query Expansion.

1 Introduction

The emergence of Web 2.0 and social web application has completely changed the way how to publish, share, and find information on the web. This shift has led researchers in the information retrieval field to look to other techniques and tools to help users to find the most relevant information to their needs. This is what the goal of the Social Book Search Track is[1].

To reach this goal and since 2011, INEX SBS has provided a collection of 2.8 million records containing both professional metadata, from Amazon, extended with user-generated content, social metadata from LibraryThing[1] (LT). In addition, it has provided a large set of 93,976 anonymous users' profiles from LT with over 33 million cataloguing transactions.

A set of topics that were extracted from LT forum have been also made available to evaluate systems submitted by participants at the SBS task. Each of them contains many fields to describe the user needs; title, group, mediated query, narrative and a personal catalogue of the topic starter. This year the topics have been enriched by an

[1] www.librarything.com

examples field which lists all the example books mentioned in the search request. The different representations of the topic made the understanding of the users' information need and the determination of the importance of each term in the topic a very difficult task.

In this paper, we try to tackle this problem through two contributions. Firstly, we introduce the *tf-iqf* function to assign a high weight values to terms which are significant to the topic (high term frequency in the given topic) and a low weight to those appearing in many different topics. Secondly, and to better represent the topic, we add other terms by expanding the original query using Rocchio technique [2]. The example books mentioned in the search request are used as relevant feedback documents in this technique.

The organization of the rest of the paper is as follows: in section 2, we describe the data processing; in section 3, we present our approach focusing on the retrieval function used and the weighting functions of the two above methods. Reporting and describing the results of our experiments will be in section 4. Finally, we conclude in Section 5 with an outlook to future work.

2 Data processing and indexing

In this section, we describe the data processing and indexing techniques. Several studies in social information retrieval show that social tagging can improve the quality of search results by using these tags as index terms. In order to investigate the impact of social tags on SBS, we want to emphasize that in all our experiments we have used only the user profiles file[2] provided by INEX SBS track which contains over 33 million cataloguing transactions. Each transaction is represented by a row, where each row contains five columns; the user, the book, the month in which the user added that book, the rating and a set of tags assigned by this user to this book. Those columns are represented, the user profiles file, as follow:

<user_id> <book_id> <add_date> <user_rating> <user_tags>

The two columns, *book_id* and *user_tags*, are used to extract for each book all tags that are assigned to it by users. Before creating the index, the Porter stemmer [3] is used to reduce all tags into their stem. After all tags have been extracted and processing data is done, the data is indexed using the following two relational tables, implemented using the Postgres[3] database management system:

- ***BOOKS(id_book, id_tag, tf):*** contains for each book *id_book*, the tag *id_tag* used by users to tag this book and *tf* (the number of times users of LT have tagged the book *id_book* with the tag *id_tag*)
- ***TAGS (id_tag, tag, idf):*** contains the stem tag for a tag *id_tag* and *idf* (logarithm of the ratio of the number of books in the collection to the number of books tagged by the given tag).

---

[2] http://cleverdon.hum.uva.nl/sbs/profiles/sbs15.profiles.gz

[3] http://www.postgresql.org/

3 Our approach

To illustrate our approach, we first present in this section, the scoring function used to measure the similarity between query and each book in the collection. Then, we describe the two techniques used to weighting query terms and to expanding the original query.

3.1 Scoring Function

In our approach, we consider a query *Q* as a set of weighted terms issued by the topic starter to describe their needs. Each document (book) of the collection is represented by a vector where each dimension value is the number of times the document *D* is tagged by the specified tag *t*. To compute the score *S(D,Q)* of a document *D* with respect to a query *Q*, we use BM15 the simplified retrieval function of okapi-BM25[4]. The BM15 function is used because, there is no notion of length normalization and the number of tags assigned to a book cannot be considered as a length.

$$S(Q,D) = \sum_{t \in Q} \frac{(k1+1)w(t,D)}{k1+w(t,D)} . idf(t) . \frac{(k3+1)w(t,Q)}{k3+w(t,Q)} \quad (1)$$

Where $\mathrm{w(t, D)}$ and $\mathrm{w(t, Q)}$ are the weights of term *t* in the document *D* respectively in the query *Q*. *K1* and *k3* are free parameters. $\mathrm{idf(t)}$ is the inverse document frequency calculated as follow :

$$idf(t) = log \frac{|D|-df(t)+0.5}{df(t)+0.5} \quad (2)$$

Where $df(t)$is the number of documents that are tagged with t, and $|D|$ is the total number of documents in a collection.

3.2 Query terms weighting

The topics of INEX SBS track which are derived from the LT forum contain many fields namely title, group, mediated query and narrative. In our approach we investigate all terms of all this fields however we give a weight to each term of the topic by using tf-iqf formula which is similar to tf-idf for documents [5]. Therefore, each topic will be represented by a weighted vector, where the values of this victor are the weights of terms calculated as follow:

$$w(t) = tf(t,q) . iqf(t) \quad (3)$$

Where $w(t)$ is the weight of term t, $tf(t,q)$ is the frequency of term t in the topic q, and the $iqf(t)$ is the inverse query frequency calculated as follow:

$$iqf(t) = log \frac{|Q|-qf(t)+0.5}{qf(t)+0.5} \quad (4)$$

Where $qf(t)$is the number of topics that contain t, and |Q| is the total number of topics in a collection (the 680 topics from INEX 2014 are used).

3.3 Query expansion

This year the topics of INEX SBS have been enriched by an examples field which lists all the example books mentioned in the search request with Information on if the user has read the book or not and his/her sentiment about this book (positive, negative or neutral). In order to exploit this field, we adopted the query Expansion method which is used to improve the search results by automatically adding terms to the user's original query. Rocchio relevance feedback is one of the most popular methods used for this task. Here are the steps to be followed for its application:

- For each book example in the topic, rank all the tags assigned to this book according to the *tf-idf* function;
- Select the top-k tags for each book;
- Apply the function below to construct the new query

$$\vec{Q}_{new} = \alpha.\vec{Q}_{orig} + \beta\frac{1}{P}\sum_{d\in Pos}\vec{d} + \gamma\frac{1}{NT}\sum_{d\in Neut}\vec{d} - \delta\frac{1}{N}\sum_{d\in Neg}\vec{d} \quad (5)$$

Where $\vec{Q}_{orig}$ and $\vec{Q}_{new}$ are the original and the new query vector respectively, $\vec{d}$ denotes the weighted tag vector of the example book d. P, NT and N are respectively the number of positive, neutral and negative books. The parameter α is used to measure the importance of the terms of the original query, whereas β, γ and δ are used to weight the tags of example books on the final query. The latter parameters take into account the sentiment of the topic starter about this example book. It is worth mentioning that the information on whether the topic starter has read the example book has not been taken into account in this technique.

4 Experiments & Results

In order to test and validate our approach, we ran several experiments with different representation of the query. We use the topics and the relevance judgments of INEX SBS 2014[4] to training our approach and optimizing the parameters of the different function used.

4.1 Training & optimizing from SBS 2014

In order to study the impact of term weighting on retrieval performance; we have opted for two ways of doing. In the first one, the weight of terms consisted of their frequency of appearance in the topic fields. In the second one, the weight of terms is calculated by the *tf-iqf* described in section 3.2. We then calculated the score of each

[4] http://social-book-search.humanities.uva.nl/data/judgements/inex14sbs_V2.qrels

book in the index using the retrieval function. We mention here that all terms of the topic fields are used to represent the query.
We optimized the parameters of the BM15 using the 2014 topics (k3 is set to 1000 and k1 have been optimized to 5). Table1 summarizes the results of the two weighting methods on the 680 topics and relevance judgments of SBS 2014.

Table 1. Impact of the query terms weighting method.

Weighting method	nDCG@10	MRR	MAP	R@1000
frequency of the term	0.065	0.137	0.047	0.459
tf-iqf of the term	**0.101**	**0.198**	**0.075**	**0.520**

To investigate the query expansion technique, and since the example field was not present in the topics of SBS 2014, it was necessary to perform our approach by using the 208 topics of 2015 only. The evaluation will be based on the relevance judgments of SBS 2014. In the beginning and in order to assess the impact of the example books on the retrieval performance, we selected for each of them the top-10 tags ranked by tf-idf. The values of the example book vector are set to 1 to gives all tags the same importance. The equation (5) was used to compute and weight the final query vector. For this, we fixed $\alpha = 0$ (no topic terms), $\beta = 1$, $\delta = 0.5$ while varying γ from 0 to 1 in steps of 0.2. The best parameter found was $\gamma = 0.8$. We combined then the original topic terms (top-10, top-20 and top-30 ranked by tf-iqf) with the top-10 tags of the example books. The best parameters found above ($\beta = 1, \gamma = 0.8, \delta = 0.5$) were used with varied α from 0 to 1 in steps of 0.2. The best results have been found when we have used the top-20 terms of the topic with the top-10 tags of the example books and $\alpha = 0.4$. The results of the different topic representations are shown in table 2.

Table 2. Query expansion performances

Topic representation	nDCG@10	MRR	MAP	R@1000
Top-20 terms of the topic only ($\beta, \gamma, \delta = 0$)	0.094	0.200	0.067	0.486
Top-10 tags of example book only ($\alpha = 0$)	0.133	**0.275**	0.094	0.478
Top-10 tags + top-20 terms	**0.141**	0.272	**0.103**	**0.549**

In the last stage and in order to avoid returning books that already exist in the catalogue of the topic starter, we removed all these books from the ranked list. The table below shows that the results have been improved when using this technique.

Table 3. Evaluation results of removing the catalogued books

Run	nDCG@10	MRR	MAP	R@1000
Before removing catalogued books	0.141	0.272	0.103	0.549
After removing catalogued books	**0.160**	**0.321**	**0.116**	**0.554**

4.2 Submitted Runs

The 04 runs submitted at INEX SBS 2015 are the best ones in each step of the experiments. Table 3 summarizes the submitted runs whereas table5 shows their results compared to the best run submitted to INEX 2015 SBS track.

Table 4. Description of the submitted runs

Run	**Topic representation**
CERIST_TOPICS_EXP_NO	Top-20terms+Top-10tags+Remove catalogued books
CERIST_TOPICS_EXP	Top-20 topic terms +Top-10 tags of example books
CERIST_TOPICS	Top-20 terms of the topics ranked by *tf-iqf*
CERIST_EXAMPLES	Top-10 tags of the example books ranked by *tf-idf*

Table 5. The official evaluation measure by INEX 2015 of our runs compared to the best run.

Run	**Rank**	**nDCG@10**	**MRR**	**MAP**	**R@1000**
Best run (MIIB Run6)	01	0.184	0.394	0.105	0.374
CERIST_TOPICS_EXP_NO	02	0.137	0.285	0.093	0.562
CERIST_TOPICS_EXP	04	0.113	0.228	0.080	0.558
CERIST_TOPICS	12	0.093	0.204	0.066	0.497
CERIST_EXAMPLES	15	0.090	0.189	0.060	0.448

4.3 Analysis

After all the performed experiments we note that, from Table1, using the *tf-iqf* function to weight the topic terms improves the results more than using the frequency of the term. In term of nDCG@10 measure, the result increases from 0.065 to 0.101. From table 2, we notice that using the query expansion technique to add other terms to the original query can also improve the results. This technique increases nDCG@10 from 0.094 to 0.113 when we using the tags of example books only, and from 0.113 to 0.137 when we combining both the tags of the example books and the original query terms.

It is important also mentioning that the use of the topics of INEX 2014 as a training and the topics of INEX 2015 as a testing sets, which are almost the same , can overfitting the parameters of the model learned. To avoid this overfitting, it would have been better if we had used the n-fold cross-validation technique.

5 Conclusion

In this paper, we described our participation to the INEX 2015 Social Book Search track. Our proposed approach investigates the query terms weighting techniques to select the most significant terms of the topic. Two methods were performed, the tf-iqf function to weight the terms of the topic and rocchio technique to expand and reweight the query terms. Both methods have given interesting results, especially, the

query expansion method. It is true that we used the user profiles file in our experiments but it was limited only to the tags assigned by users to books. In future works, it will be interesting to use other information from this file like rating, personnel catalogues of each user and similarity between them to experiment with collaborative filtering and recommender system to improve the results.

6 References

1. Bellot, P., Bogers, T., Geva, S., Hall, M., Huurdeman, H., Kamps, J., Kazai, G., Koolen, M., Moriceau, V., Mothe, J., Preminger, M., SanJuan, E., Schenkel, R., Skov, M., Tannier, X. and Walsh, D. (2014). Overview of INEX 2014. In *Information Access Evaluation. Multilinguality, Multimodality, and Interaction* (pp. 212-228). Springer International Publishing.
2. Rocchio, J.: Relevance Feedback in Information Retrieval. Prentice Hall, Englewood, Cliffs, New Jersey (1971).
3. Porter, M. F. (1980). An algorithm for suffix stripping. *Program: Electronic Library and Information Systems*, *40*(3), 211-218.
4. Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M., and Gatford, M. (1995). Okapi at TREC-3. NIST *SPECIAL PUBLICATION SP*, 109-109.
5. Salton, G., Wong, A., and Yang, C. S. (1975). A vector space model for automatic indexing. Commun. ACM, 18(11):613–620.

No need to log in or register to track the status of your order. Order books without registering at Book Depository.Â Please enter manually:","bd_js_keep_typing_to_refine_search_results":"Keep typing to refine the search results","bd_js_top_categories":"Top Categories","bd_price_save":"Save {0}","bd_js_name_only_letters":"Sorry, full name can only contain letters","bd_js_show_more":"show more","bd_js_enter_valid_email_address":"Please enter a valid email address"