SBS 2016 : Combining Query Expansion Result and Books Information Score for Book Recommendation

Amal Htait, Sébastien Fournier, and Patrice Bellot

Aix Marseille Université, CNRS, ENSAM, Toulon Université, LSIS UMR 7296,13397,
Marseille, France.
Aix-Marseille Université, CNRS, CLEO OpenEdition UMS 3287, 13451,
Marseille, France.
{amal.htait, sebatien.fournier, patrice.bellot}@univ-amu.fr

Abstract. In this paper, we present our contribution in Suggestion Track at the Social Book Search Lab. This track aims to develop test collections for evaluating ranking effectiveness of book retrieval and recommender systems. In our experiments, we combine the results of Sequential Dependence Model (SDM) and the books information that includes the *price*, the *number_Of_Pages* and the *publication_Date*. We also expand topics' queries by the similar books information to improve the recommendation performance.

Keywords: Social Information Retrieval, Recommendation, Sequential Dependence Model, Expand Query.

1 Introduction

The Social Book Search (SBS) Tracks [1] were introduced by INEX in 2010 with evaluation purposes for supporting users in searching collections of books based on book metadata and associated user-generated content.

Social Book Search Lab includes the following tracks: Suggestion Track, Interactive Track and Mining Track. Our work is on Suggestion Track, which suggests a list of the most relevant books according to the request provided by the user. Since 2011, for the social books search task, the document provided is a collection of 2.8 million records containing professional metadata (Amazon[1]) extended with user-generated content and social metadata (LibraryThing[2]). In addition, a set of 113,490 anonymous users profiles is provided from LibraryThing (LT). Therefore, Information Retrieval (IR) Systems must search through editorial data, user reviews and ratings for each book, instead of searching through the whole content of the book. The topics provided each year are extracted from the LibraryThing forums and by represent real requests from real users.

[1] http://www.amazon.com/

[2] www.librarything.com

Our participation in 2011 and 2012 was based on re-ranking books using social component such as popularity and ratings [2],[3]. On 2014, we were able to achieve the second best run using InL2 model implemented in Terrier[3][4]. And for 2015 participation, we combined results of InL2 and Sequential Dependence Model (SDM). Also, we integrated tools from natural language processing (NLP) and approaches based on graph analysis to improve the recommendation performance[5].

This year's participation is through an IR system based on 3 main steps:

- We expand the topic queries using the similar books information, since the topics contain books titles mentioned by the user as similar or example books to those he seeks.
- We apply a re-ranking method using a score calculated of books information including the *price*, the $number\_Of\_Pages$ and the *publication_Date*.
- We apply these methods on Amazon book collection and on the users profiles collection.

For our participation in SBS 2016, we submitted 4 runs in which we applied the previously mentioned steps. The rest of this paper is organized as follows. The following section describes the data processing and indexing. In section 3, we have the description of our retrieval framework. In section 4, we describe the submitted runs. Finally, we present the obtained results in section 5.

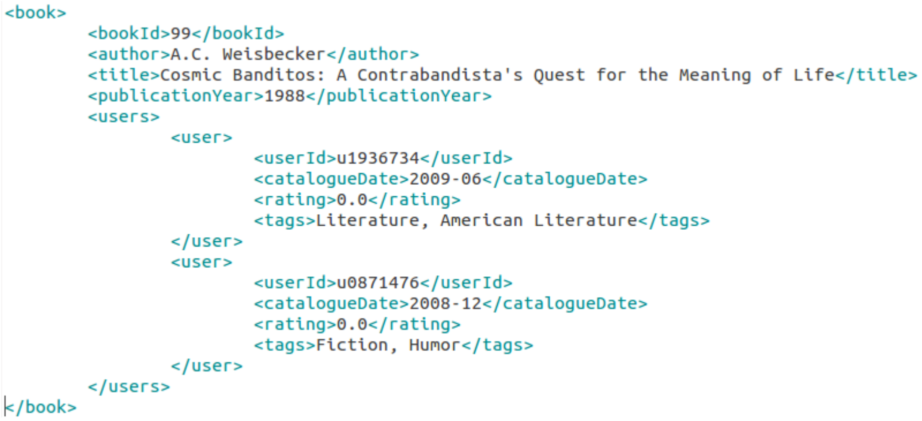
2 Data processing and indexing

We use, in addition to the Amazon book Collection, the users profiles Collection provided by SBS Lab track which contains the cataloguing transactions of 113,490 users. The cataloguing transactions of a user is a list of information concerning the books read by the user. Each transaction is represented by a row, where each row contains eight columns; user, book, author, book title, publication year, month in which the user added that book, rating and a set of tags assigned by this user to this book. From the users profiles, we create for each book an XML file with all its information. An example is illustrated in the following XML code of Figure 1.

For indexing the Amazon book collection, we take all the tags of the XML files identified by the ISBNs. And for indexing users profiles collection, we take all the tags of the created XML files identified by the LibraryThingID. Also, we use the following Indri[4] indexing parameters: *Porter Stemmer* and $Stop\ Words\ Removal$.

[3] hhtp://terrier.org

[4] http://www.lemurproject.org/indri/



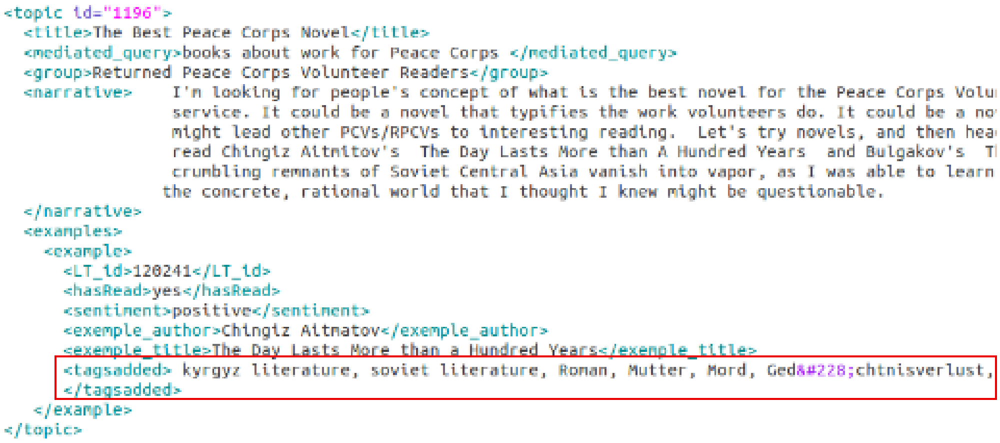
```
<book>
        <bookId>99</bookId>
        <author>A.C. Weisbecker</author>
        <title>Cosmic Banditos: A Contrabandista's Quest for the Meaning of Life</title>
        <publicationYear>1988</publicationYear>
        <users>
                <user>
                        <userId>u1936734</userId>
                        <catalogueDate>2009-06</catalogueDate>
                        <rating>0.0</rating>
                        <tags>Literature, American Literature</tags>
                </user>
                <user>
                        <userId>u0871476</userId>
                        <catalogueDate>2008-12</catalogueDate>
                        <rating>0.0</rating>
                        <tags>Fiction, Humor</tags>
                </user>
        </users>
</book>
```

Fig. 1. An example of book XML files from users profiles collection.

3 Retrieval Model

3.1 Query Expansion by example books information

To build our queries we use mainly the title of the query and the information of similar example books mentioned by the user in the topic. Also, we use the tags of these similar books extracted from the users profiles collection for query expansion. The XML code in Figure 2 illustrates an example of adding similar book tags for query expansion.



```
<topic id="1196">
  <title>The Best Peace Corps Novel</title>
  <mediated_query>books about work for Peace Corps </mediated_query>
  <group>Returned Peace Corps Volunteer Readers</group>
  <narrative>    I'm looking for people's concept of what is the best novel for the Peace Corps Volu
                 service. It could be a novel that typifies the work volunteers do. It could be a no
                 might lead other PCVs/RPCVs to interesting reading.  Let's try novels, and then hea
                 read Chingiz Aitmitov's  The Day Lasts More than A Hundred Years  and Bulgakov's  T
                 crumbling remnants of Soviet Central Asia vanish into vapor, as I was able to learn
                the concrete, rational world that I thought I knew might be questionable.
  </narrative>
  <examples>
    <example>
      <LT_id>120241</LT_id>
      <hasRead>yes</hasRead>
      <sentiment>positive</sentiment>
      <exemple_author>Chingiz Aitmatov</exemple_author>
      <exemple_title>The Day Lasts More than a Hundred Years</exemple_title>
      <tagsadded> kyrgyz literature, soviet literature, Roman, Mutter, Mord, Ged&#228;chtnisverlust,
      </tagsadded>
   </example>
</topic>
```

Fig. 2. An example of adding similar book tags for query expansion from Topics 2015.

3.2 Sequential Dependence Model

SDM relies on the idea of integrating multi word phrases by considering a combination of query terms with proximity constraints such as: single term features (standard unigram language model features, f_T), exact phrase features (words appearing in sequence, f_O) and unordered window features (require words to be close together, but not necessarily in an exact sequence order, f_U) [8]. In Table 1, more details about the term weighting functions are shown, where $tf_{e,D}$ is the number of times term e matches in document D, $cf_{e,D}$ is the number of times term e matches in the entire collection, $|D|$ is the length of document D, and $|C|$ is the size of the collection. Finally, μ is a weighting function hyperparameter that is set in our work to 2500 [4].

Table 1. Language modeling-based unigram and term weighting functions [4].

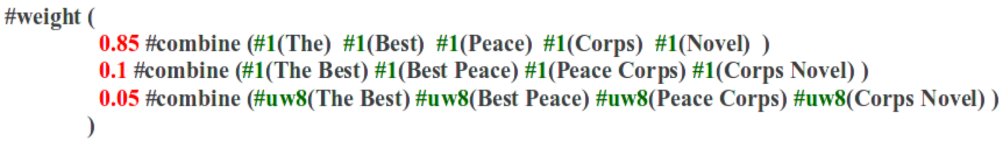
Weighting	Description
$f_T(q_i, D) = log\left[\frac{tf_{qi,D}+\mu\frac{cf_{qi}}{\lvert C\rvert}}{\lvert D\rvert+\mu}\right]$	Weight of unigram q_i in document D.
$f_O(q_i, q_{i+1}, D) = log\left[\frac{tf_{\#1(q_i,q_{i+1}),D}+\mu\frac{cf_{\#1(q_i,q_{i+1})}}{\lvert C\rvert}}{\lvert D\rvert+\mu}\right]$	Weight of exact phrase 'q_i q_{i+1}' in document D.
$f_U(q_i, q_{i+1}, D) = log\left[\frac{tf_{\#uw8(q_i,q_{i+1}),D}+\mu\frac{cf_{\#uw8(q_i,q_{i+1})}}{\lvert C\rvert}}{\lvert D\rvert+\mu}\right]$	Weight of unordered window 'q_i q_{i+1}' (span=8) in document D.

And the documents are ranked according to the below scoring equation, Equation 1:

$$\begin{aligned} SDM(Q, D) = {} & \lambda_T \textstyle\sum_{q\in Q} f_T(q, D) \\ & +\lambda_O \textstyle\sum_{i=1}^{|Q|-1} f_O(q_i, q_i + 1, D) \\ & +\lambda_U \textstyle\sum_{i=1}^{|Q|-1} f_U(q_i, q_i + 1, D) \end{aligned} \tag{1}$$

We used the Equation 1 with feature weights set to $\lambda_T = 0.85$, $\lambda_O = 0.1$ and $\lambda_U = 0.05$, like previous participation years. We applied this model to the queries using Indri 5.4 [4] Query Language [5]. An example of Indri Query Language is in Figure 3.

[5] http://www.lemurproject.org/lemur/IndriQueryLanguage.php



```
#weight (
        0.85 #combine (#1(The)  #1(Best)  #1(Peace)  #1(Corps)  #1(Novel)  )
        0.1 #combine (#1(The Best) #1(Best Peace) #1(Peace Corps) #1(Corps Novel) )
        0.05 #combine (#uw8(The Best) #uw8(Best Peace) #uw8(Peace Corps) #uw8(Corps Novel) )
       )
```

Fig. 3. An example of Indri Query.

3.3 Combination of Retrieval System output and books' information

We combine the results of SDM model with a sum of normalized scores, which we calculate from the book's *price*, *publication_Date* and $number\_Of\_Pages$. And since the combined values are of different weighting, we use the maximum and minimum scores according to Lees formula [3] as followed in Equation 2.

$$normalizedScore = \frac{oldScore - minScore}{maxScore - minScore} \tag{2}$$

The scores of SDM model and books information have different levels of retrieval effectiveness, thus it is necessary to weigh scores depending on their overall performance. We used an interpolation parameter (α) that varies in testing for the goal of achieving the best interpolation that provides better retrieval effectiveness, as shown in the Equation 3.

$$SDM\_bookInfo = \alpha.(SDM(Q, D)) + (1 - \alpha).(bookInfo(D)) \tag{3}$$

After several testings on 2015 SBS topics [6], α is set to 0.55 with the best result. $bookInfo(D)$ is calculated by a normalized score of the values of *price* only, since the *price* alone obtains the best result on 2015 SBS topics compared to the values of *price*, *publication_Date* and $number\_Of\_Pages$ combined. In Table 2, an example of our tests showing a modest but still an increase in the results when combining books prices to the equation with $\alpha = 0.55$.

Table 2. Results of testing applied on SBS 2015 Topics.

Method	**nDCG10**	**Recip_Rank**	**MAP**
SDM(Q, D)	0.1278	0.1231	0.0431
SDM_bookInfo_all	0.1251	0.1229	0.0407
SDM_bookInfo_price_0.4	0.1275	0.1237	0.0427
SDM_bookInfo_price_0.6	0.1267	0.1207	0.0433
SDM_bookInfo_price_0.55	0.129	0.1266	0.0428

[6] http://social-book-search.humanities.uva.nl/#/data/suggestion

4 Runs

We submit 4 runs for the SBS Suggestion Track:
Run1_ExeOrNarrativeNSW_Collection: We concatenate the title of the topic and the similar books fields (title, author and tags), then perform a retrieval using the SDM model. But since not all topics have example books, in this case we concatenate the title and the narrative fields of the topic after removing the *Stop_Words* from the narrative field. This run is applied on Amazon book collection.
Run2_ExeOrNarrativeNSW_UserProfile : This run is same as Run1 but it is applied on users profiles collection.
Run3_ExeOrNarrativeNSW_Collection_AddData : In this run, we combine the books price normalized score to the results of Run1.
Run4_ExeOrNarrativeNSW_UserProfile_AddData : Also in this run, we combine the books price normalized score to the results of Run2.

5 Results

Table 3 shows 2016 official SBS Suggestion Track results for our 4 runs. Our models presented this year show differences in results. The use of retrieval SDM model alone gave the best results between our runs. The use of users profiles file and the combination of books information with the SDM scores decreases the results.

We should mention that we tested our methods with the topics of SBS 2015, which had a field named *mediated_query* containing the key words of the user's request. Since this field is not in the topics of SBS 2016, we used the field *narrative* and that caused a massive amount of noise in the query. This can also explain the bad results of using users profiles collection, since it's difficult to find similarity between the query with noise and the limited information in the users profiles collection.

Table 3. Official results at SBS 2016. The runs are ranked according to *nDCG*@10.

Run	**nDCG10**	**Recip_Rank**	**MAP**	**R1000**
Best_Run_2016	0.2157	0.5247	0.1253	0.3474
Run1_ExeOrNarrNSW_Collection	0.0450	0.1166	0.0251	0.2050
Run2_ExeOrNarrNSW_UserProfile	0.0239	0.1018	0.0144	0.1742
Run3_ExeOrNarrNSW_Collection_AddData	0.0177	0.0533	0.0101	0.2050
Run4_ExeOrNarrNSW_UserProfile_AddData	0.0152	0.0566	0.0079	0.1742

6 Conclusion

In this paper, we present our contribution for the Suggestion Track of Social Book Search Lab. In the 4 submit runs, we use SDM retrieval model and we extend the query by the similar books information (title, author and tags). We apply the retrieval on Amazon book collection, and on users profiles collection. We combine the results of the retrieval system (SDM) with the normalized score of the books prices. The best result is achieved by using SDM retrieval model with the extended query on Amazon book Collection. We should note that the topics of SBS 2015 had a field named *mediated_query*, which contained the key words of the user's request (from field *narrative*). The $mediated_query$ field is used in our testing on SBS 2015 topics and helped to increase the results. But since this field is not in the topics of SBS 2016, we had to use the *narrative* field which contains many useless information that effect negatively the information research. Thus, to increase the results for future participation, we must work on extracting only the key words from the narrative field to be used in the query, and eliminate any noise information.

Acknowledgments. This work was supported by the French program Investissements dAvenir Equipex "A digital library for open humanities" of OpenEdition.org.

References

1. Gabriella Kazai, Marijn Koolen, Jaap Kamps, Antoine Doucet, and Monica Landoni. Overview of the inex 2010 book track: Scaling up the evaluation using crowdsourcing. In Shlomo Geva, Jaap Kamps, Ralf Schenkel, and Andrew Trotman, editors, INEX, volume 6932 of Lecture Notes in Computer Science, pages 98117. Springer, 2010.
2. Deveaud, R., SanJuan, E., & Bellot, P.. Social recommendation and external resources for book search. Working Notes for CLEF 2011 Conference, 7424 LNCS, 6879. 2011.
3. Ludovic Bonnefoy, Romain Deveaud, and Patrice Bellot. *Do social information help book search?* In Pamela Forner, Jussi Karlgren, and Christa Womser-Hacker, editors, CLEF (Online Working Notes/Labs/Workshop), 2012.
4. Benkoussas, C., Hamdan, H., Albitar, S., Ollagnier, A., & Bellot, P. . Collaborative Filtering for Book Recommendation. Working Notes for CLEF 2014 Conference, 501507. (2014).
5. Benkoussas, C., Ollagnier, A., & Bellot, P.. Book Recommendation Using Information Retrieval Methods and Graph Analysis. Working Notes for CLEF 2015 Conference. 2015.
6. Donald Metzler and W. Bruce Croft. A markov random field model for term dependencies. In Ricardo A. Baeza-Yates, Nivio Ziviani, Gary Marchionini, Alistair Moffat, and John Tait, editors, SIGIR, pages 472479. ACM, 2005.
7. Joon Ho Lee. Combining multiple evidence from different properties of weighting schemes. In Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 95, pages 180188, New York, NY, USA, 1995. ACM.

8. Benkoussas, C., Ollagnier, A., & Bellot, P.. Book Recommendation Using Information Retrieval Methods and Graph Analysis, CLEF 2015 Conference and Labs of the Evaluation Forum, pp. 8 p., Toulouse (France), sep 2015.