# RSLIS at INEX 2012: Social Book Search Track

Toine Bogers and Birger Larsen

Royal School of Library and Information Science
Birketinget 6, 2300 Copenhagen, Denmark
{tb,blar}@iva.dk

**Abstract.** In this paper, we describe our participation in the INEX 2012 Social Book Search track. We investigate the contribution of different types of document metadata, both social and controlled, and examine the effectiveness of re-ranking retrieval results using different social features, such as user ratings, tags, and authorship information. We find that the best results are obtained using all available document fields and topic representations. Re-ranking retrieval results works better on shorter topic representations, where there is less information for the retrieval algorithm to work with; longer topic representations do not benefit from our social re-ranking approaches.

**Keywords:** XML retrieval, social tagging, controlled metadata, book recommendation, re-ranking

## 1 Introduction

In this paper, we describe our participation in the INEX 2012 Social Book Search track[1]. Our goals for the Social Book Search task were (1) to investigate the contribution of additional controlled metadata provided for this year's task; and (2) to examine the effectiveness of using social features for re-ranking the initial content-based search results. We focus in particular on using techniques from collaborative filtering (CF) to improve our content-based search results.

The structure of this paper is as follows. We start in Section 2 by describing our methodology: pre-processing the data, which document and topic fields we used for retrieval, and our evaluation. In Section 3, we describe the results of our content-based retrieval runs, including the effect of the additional controlled metadata sources. Section 4 describes our use of social features to re-rank the content-based search results. Section 5 describes which runs we submitted to INEX, with the results of those runs presented in Section 6. We discuss our results and conclude in Section 7.

## 2 Methodology

### 2.1 Data and Preprocessing

In our experiments we used the Amazon/LibraryThing collection provided by the organizers of the INEX 2012 Social Book Search track. This collection contains XML

[1] https://inex.mmci.uni-saarland.de/tracks/books/

representations of 2.8 million books, with the book representation data crawled from both Amazon.com and LibraryThing (LT). The 2012 collection is identical to the collection provided for the 2011 track [1] in all but two ways: the collection has been expanded with additional library records from the British Library (BL) and the Library of Congress (LoC). Of the 2.8 million books in the collection, 1.15 million have a BL record and 1.25 have a LoC record. Together these two sources cover 1.82 million of the 2.8 million books in the collection.

We converted the collection's original XML schema into a simplified version to retain only those metadata fields that were most likely to contribute to the successful retrieval of relevant books[2]. After these pre-processing steps, we were left with the following 19 content-bearing XML fields in our collection: <isbn>, <title>, <publisher>, <editorial>, <creator>, <series>, <award>, <character>, <place>, <blurber>, <epigraph>, <firstwords>, <lastwords>, <quotation>, <dewey>, <subject>, <browseNode>, <review>, and <tag>.

We replaced the numeric Dewey codes in the original <dewey> fields by their proper textual descriptions using the 2003 list of Dewey category descriptions[3] to enrich the controlled metadata assigned to each book. For example, the XML element <dewey>519</dewey> was replaced by the element <dewey>Probabilities & applied mathematics</dewey>. The BL and LoC records were provided in MODS format[4], we mapped this format to the appropriate new XML fields and added them to the book representations.

### 2.2 Field categories and Indexing

The 19 selected XML fields in our collection's book representations fall into different categories. Some fields, such as <dewey> and <subject>, are examples of *controlled metadata* produced by LIS professionals, whereas other fields contains *user-generated metadata*, such as <review> and <tag>. Yet other fields contain 'regular' book metadata, such as <title> and <publisher>. Fields such as <quotation> and <firstwords> represent a book's content more directly.

To examine the influence of these different types of fields, we divided the document fields into five different categories, each corresponding to an index. To examine the contribution of the additional BL/LoC controlled metadata we created two versions of the index containing controlled metadata: one with and one without this additional controlled metadata. In addition, we combined all five groups of relevant fields for an index containing all fields. This all-fields index also comes in two variants: one with and one without the BL/LoC metadata. This resulted in a total of eight indexes:

**All fields** For our first index all-doc-fields we simply indexed all of the available XML fields (see the previous section for a complete list). The all-doc-fields-plus index contains all of the original 2011 fields as well as the BL/LoC metadata.

[2] Please consult [2] for more details on this filtering and conversion process.
[3] Available at http://www.library.illinois.edu/ugl/about/dewey.html
[4] See http://www.loc.gov/standards/mods/ for more information.

**Metadata** In our metadata index, we include all metadata fields that are immutably tied to the book itself and supplied by the publisher: <title>, <publisher>, <editorial>, <creator>, <series>, <award>, <character>, and <place>.

**Content** For lack of access to the actual full-text books, we grouped together all XML fields in the content index that contain some part of the book text: blurbs, epigraphs, the first and last words, and quotations. This corresponded to indexing the fields <blurber>, <epigraph>, <firstwords>, <lastwords>, and <quotation>.

**Controlled metadata** In our controlled-metadata index, we include the three controlled metadata fields curated by library professionals harvested from Amazon: <browseNode>, <dewey>, and <subject>. The controlled-metadata-plus index contains the original metadata as well as the BL/LoC metadata.

**Tags** We split the social metadata contained in the document collection into two different types: tags and reviews. For the tags index, we used the tag field, expanding the tag count listed in the original XML. For example, the original XML element <tag count="3">fantasy</tag> would be expanded as <tag>fantasy fantasy fantasy</tag>. This ensures that the most popular tags have a bigger influence on the final query-document matching.

**Reviews** All user reviews belonging to a single book were combined in a single document representation for that book and added to our review index reviews.

We used the Indri 5.1 retrieval toolkit[5] for indexing and retrieval. We performed stopword filtering on all of our indexes using the SMART stopword list, and preliminary experiments showed that using the Krovetz stemmer resulted in the best performance. Topic representations were processed in the same manner.

### 2.3 Topics

As part of the INEX 2012 Social Book Search track three sets of topics were released with requests for book recommendations based on textual description of the user's information need: two training sets and a test set. All topic sets were extracted from the LibraryThing forum. The original training set of 43 topics created for the 2011 Social Book Search track came with unverified relevance judgments, so we only used the test set of 2011 as our training set for 2012. This second training set contains 211 topics with relevance judgments derived from the books recommended on the LibraryThing discussion threads of these 211 topics. We used this training set to optimize our retrieval algorithms in the different runs. The results we report in Sections 3 and 4 were obtained using this training set.

The test set for 2012 contains 90 additional topics which, combined with the 211 training set topics, were used to rank and compare the different participants' systems at INEX 2012. The results listed in Section 6 were obtained on this combined set of 301 topics. Each topic is represented by several different fields:

**Title** The <title> field contains the title of the forum topic and typically provide a concise description of the information need. Runs that only use the topic title are referred to as title.

[5] Available at http://www.lemurproject.org/

**Group** The LibraryThing forum is divided into different groups covering different topics.

**Narrative** The first message of each forum topic, typically posted by the topic creator, describes the information need in more detail. This often contains a description of the information need, some background information, and possibly a list of books the topic creator has already read or is not looking for. The narrative typically contains the richest description of the topic.

**All topic fields** We also performed runs with all three fields combined, referred to as all-topic-fields.

In our experiments with the training and the test set, we restricted ourselves to automatic runs using the following title and the all-topic-fields representations (based on our experiments for INEX 2011 [2]).

### 2.4 Experimental setup

In all our retrieval experiments, we used the language modeling approach with Jelinek-Mercer (JM) smoothing as implemented in the Indri 5.1 toolkit. We preferred JM smoothing over Dirichlet smoothing, because previous work has shown that for longer, more verbose queries JM smoothing outperforms Dirichlet smoothing [3], which matches the richer topic descriptions provided in the topic sets.

For the best possible performance, we optimized the $\lambda$ parameter, which controls the influence of the collection language model, with higher values giving more influence to the collection language model. We varied $\lambda$ in steps of 0.1, from 0.0 to 1.0 using the training set of topics. We also examined the value of stop word filtering and stemming and use the SMART stop word list and Krovetz stemming in these cases. This resulted in 44 different possible combinations of these three parameters. For each topic we retrieved up 1000 documents and we used NDCG@10 as our evaluation metric [4].

## 3 Content-based Retrieval

In order to produce a competitive baseline for our experiments with re-ranking based on social features, we conducted a first round of experiments focused on optimizing a standard content-based retrieval approach for each combination of index and topic representations. We found that the best results were always produced with stop word filtering and Krovetz stemming, so all results reported in this paper share these settings. We compared the different index and the different topic representations for a total of 16 different content-based retrieval runs. Table 1 shows the best NDCG@10 results for each run on the training set.

We can see several interesting results in Table 1. First, we see that the best overall content-based run used all topic fields for the training topics, retrieved against the index containing all document fields (all-doc-fields) with an NDCG@10 score of 0.3058. Retrieving on the all-doc-fields index performs best on both topic sets (all-topic-fields and title). The reviews index is a close second with strong performance on both topic sets. When we compare the two topic sets, we see that the

**Table 1.** Results of the 16 different content-based retrieval runs on the training set using NDCG@10 as evaluation metric. Best-performing runs for each topic representation are printed in bold.

| Document fields | Topic fields | |
|---|---|---|
| | title | all-topic-fields |
| metadata | 0.0915 | 0.2015 |
| content | 0.0108 | 0.0115 |
| controlled-metadata | 0.0406 | 0.0496 |
| controlled-metadata-plus | 0.0514 | 0.0691 |
| tags | 0.0792 | 0.2056 |
| reviews | 0.1041 | 0.2832 |
| all-doc-fields | **0.1129** | **0.3058** |
| all-doc-fields-plus | 0.1120 | 0.3029 |

all-topic-fields set consistently outperforms the title topic set. These findings are all in line with our 2011 results [2].

Finally, we observe that the content and controlled-metadata indexes result in the worst retrieval performance across all four topic sets. Adding the extra BL/LoC controlled metadata has a positive effect on retrieving over only controlled metadata: the controlled-metadata-plus index outperforms the controlled-metadata on both topic sets. However, the adding this additional BL/LoC metadata to the index containing all document fields (all-doc-fields-plus) actually causes a small but surprising drop in performance. This suggests that for some topics the existing document fields better describe the documents than the information present in the BL/LoC fields.

## 4 Social Re-ranking

The inclusion of user-generated metadata in the Amazon/LibraryThing collection gives the track participants the opportunity to examine the effectiveness of using social features to re-rank or improve the initial content-based search results. One such a source of social data are the tags assigned by LibraryThing users to the books in the collection. The results in the previous section showed that even when treating these as a simple content-based representation of the collection using our tags index, we can achieve relatively good performance.

However, there are still many topics for which performance is sub-par, with many possible reasons for this performance gap. One explanation could be differences in document field sparsity, which could cause certain indexes to underperform for particular topics. The well-known vocabulary problem [5] could be another explanation, resulting in mismatches between synonymous query and document terms. Finally, content-based matches are no guarantee for high-quality recommendations, merely for on-topic recommendations.

To remedy these problems, we explore the use of social features for re-ranking the content-based search results in this section. We experiment with re-ranking based on book similarities (Section 4.1) as well as a personalized re-ranking approach (Section 4.2).

### 4.1 Book similarity re-ranking

Similar books that are equally relevant to a user's request for recommendations might appear at wildly different positions in the results list due to differences in term usage between the documents and the topic description. The goal of our re-ranking approach is to push those relevant documents that did not score well under a content-based approach to a higher position in the ranked results list. To that end we propose calculating a new retrieval score for each book that is a linear combination of (1) the original retrieval score and (2) the combined contributions of all other documents in the results list, weighted by their similarity to the book in question. This means that each of the books $j$ retrieved for a topic contributes a little bit to the final retrieval score of a specific book $i$, depending on the original retrieval score $score_{org}(j)$ of book $j$ and its similarity $sim(i,j)$ to book $i$. More similar books and books retrieved at higher ranks contribute more to book $i$'s new re-ranked score $score_{re-ranked}(i)$; others contribute less. Equation 1 shows how we calculate this score:

$$score_{re-ranked}(i) = \alpha \cdot score_{org}(i) + (1-\alpha) \cdot \sum_{j=1, i \neq j}^{n} score_{org}(j) \cdot sim(i,j) \qquad (1)$$

Before re-ranking we apply rank normalization on the retrieved results to map the score into the range $[0,1]$ [6]. The balance between the original retrieval score $score_{org}(i)$ and the contributions of the other books in the results list is controlled by the $\alpha$ parameter, which takes values in the range $[0,1]$. The actual book similarities $sim(i,j)$ can be calculated using different types of social features; we have explored five variants, which are described in more detail below.

**User ratings** As mentioned earlier, content-based matches are no guarantee for high-quality book recommendations; they merely indicate a strong term overlap between the topic description and the book descriptions. One way of dealing with this problem is to consider one of the social features in the collection that explicitly capture the quality of a book: user ratings. The reviews in the Amazon/LibraryThing collection contain the Amazon user names of the reviewers as well as their ratings on a five-star scale. We extract and use these ratings to calculate the similarities between the different books.

For each book in each of our results lists, we construct an vector of book ratings that contains all the ratings for that book from each reviewer in the Amazon/LibraryThing collection. Missing ratings—in case a reviewer did not review that particular book—receive a score of zero. We combine all item rating vectors in an **IU** ratings matrix where $I$ is the number of books retrieved in all of our results lists combined and $U$ is the number of reviewers in the collection. We normalize the **IU** ratings to compensate for individual differences in rating behavior [7].

Inspired by item-based collaborative filtering [8], we then calculate the cosine similarity between pairs of book vectors (i.e., row vectors). For re-ranking purposes we only need to calculate the book similarities for pairs of books that occur in the same results list. The resulting book similarities are then fed into our re-ranking approach (Eq. 1); we refer to this as IU-similarity.

**Amazon's "*similar products*"** The Amazon/LibraryThing collection already contains information about similar books: each book representation can contain up to ten <similarproduct> fields which contain the ISBN numbers of similar books, as seen on Amazon under the "*similar products*" section of a book Web page. We also explore the value of these book similarities in our re-ranking approaches, setting the similarity between two books $sim(i, j)$ to 1 if book $j$ is mentioned in the representation of book $i$ (and vice versa), and to 0 otherwise. We refer to this as II-similarity.

How do these "*similar products*" stack up against the ratings-based book similarities? This "*similar products*" data is likely to be a more accurate representation of book similarity based on user ratings as it is calculated over the entire set of user ratings, both with and without reviews [9]. In contrast , the ratings in our **IU** matrix only represent the ratings of a subset of reviews and not the ratings made by users with entering an actual review. However, the "*similar products*" similarities are binary even though the original similarities calculated by Amazon's algorithms were not. Moreover, the "*similar products*" data is likely to be incomplete. Amazon only shows a random selection of 10 similar books each time a book's Web page is generated. This means that the set of similar books during the original crawling of the Amazon/LibraryThing collection represents just a subset of all similarity pairs.

**Tags** Another source of information for calculating book similarities are the tags assigned to the different books. For this source of book similarities, we construct a **IT** matrix, analogous to our **IU** matrix. In the **IT** matrix, the columns represent the different tags assigned to all the books in our result lists. Each value in **IT** represents the number of times tag $t$ has been assigned to book $i$. If a tag was not assigned to a book, that cell receives the value 0. The **IT** matrix is then row-normalized. We obtain the similarity between two books by calculating the cosine similarity between their two row vectors. We refer to this as IT-similarity.

**Authors** Author-book associations represent another way of calculating book similarities: books written by the same author(s) are often similar in style and content. To explore this type of similarity, we construct a **IA** matrix where the columns represent the authors associated with all the books in our result lists. Values in **IA** are binary, with a value of 1 if author $a$ wrote book $i$, and a 0 otherwise. We obtain the similarity between two books by taking the cosine similarity between their vectors. We refer to this as IA-similarity.

**Fusing ratings, tags and authors** Instead of picking just one of the aforementioned sources of book similarity, we also experimented with using a combination of user ratings, tags, and authorship for calculating the book similarities. To this end we construct a combined matrix **IUTA**, which consists of the **IU**, **IT**, and **IA** matrices combined so that each book vectors contains both user ratings, tags, and authorship information. The expectation here is that the different information sources can augment each other's performance. Again, we calculate the similarity between two books by calculating the cosine similarity between their two **IUTA** row vectors. We refer to this as IUTA-similarity.

### 4.2 Personalized re-ranking

In addition to the one-size-fits-all approach to re-ranking described in Section 4.1, we also explore a personalized re-ranking approach that takes into account the past preferences of the user who originally created the LT topic requesting book recommendations. The goal is to calculate a new personalized score $score_{personalized}(u,i)$ for a LibraryThing user $u$ and a retrieved book $i$ that pushes $i$ up in the rankings if it is similar to other books read by $u$ in the past. The new personalized score is a linear combination of the original retrieval score $score_{org}(i)$ for book $i$ and the similarity between $i$ and the other books in $u$'s profile. Equation 2 shows how we calculate this personalized score:

$$score_{personalized}(u,i) = \alpha \cdot score_{org}(i) + (1-\alpha) \cdot sim_{tag}(u,i) \tag{2}$$

Again, we control the balance the original retrieval score $score_{org}(i)$ and the similarity with the user's past preferences with the $\alpha$ parameter, which takes values in the range $[0,1]$. There are different ways of calculating the similarity $sim(u,i)$ between a user's profile and a book $i$ book similarities: user ratings, tags, authors, or even term overlap between different metadata fields. Tags showed the most promising performance in preliminary experiments, so we construct a tag vector for all tags assigned by the user to books read in the past and calculated the cosine similarity $sim_{tag}(u,i)$ between that vector and the **IT** row vector corresponding to book $i$. That way, a book that shares a lot of tags with books read by a user in the past will be seen as more similar. We refer to this as pers-similarity.

### 4.3 Training set results

Table 2 shows the results of the different social re-ranking runs for the optimal $\alpha$ values. We optimized in steps of 0.01. The baseline runs for both topic representations are also included for convenience.

The results of the social re-ranking approaches are very different for the two topic representations. When using the title field for retrieval, all non-personalized re-ranking methods provide impressive boosts over the baseline. The best-performing re-ranking approach here is II-similarity, which uses Amazon's data about "*similar products*". With an NDCG@10 of 0.2429 it increase performance over the baseline by 115%. Typically, most weight is given to the original scores with $\alpha$ values ranging from 0.92 to 0.99, although the other retrieved books do seem to offer a small but valuable contribution, given the performance increases.

**Table 2.** Results of the 12 different re-ranking runs using NDCG@10 as evaluation metric. The results of the best baseline runs for each topic representation are also included for convenience. Best-performing runs for each topic representation are printed in bold.

| Runs | Topic fields | | | |
|---|---|---|---|---|
| | title | | all-topic-fields | |
| | NDCG@10 | $\alpha$ | NDCG@10 | $\alpha$ |
| Baseline | 0.1129 | - | **0.3058** | - |
| IU-similarity | 0.1631 | 0.92 | 0.3058 | 1.0 |
| II-similarity | **0.2429** | 0.94 | 0.3058 | 1.0 |
| IT-similarity | 0.1895 | 0.99 | 0.3058 | 1.0 |
| IA-similarity | 0.1535 | 0.96 | 0.3058 | 1.0 |
| IUTA-similarity | 0.1615 | 0.97 | 0.3058 | 1.0 |
| pers-similarity | 0.1293 | 0.65 | 0.3058 | 1.0 |

A possible explanation for the fact that II-similarity outperforms IU-similarity is that the latter similarities are calculated over an incomplete subset of Amazon user ratings; Amazon's "*similar products*" are likely calculated over all ratings. We can therefore also consider the results using II-similarity as an upper threshold on performance *if* we had all user ratings in the Amazon/LibraryThing collection.

Of the three types of similarity calculated directly on the Amazon/LibraryThing collection—IU-similarity, IT-similarity, and IA-similarity—re-ranking using tag overlap seem to provide the best performance with a score of 0.1895. Surprisingly, the combination of the three sources, IUTA-similarity, does not perform better than the individual sources. This is *not* in line with previous research [10].

However, when using all available topic fields for retrieval (all-topic-fields), social re-ranking does not help at all with all optimal $alpha$ values being equal to 1.0 (which retains only the original retrieval scores. Apparently, using longer query representations makes it that much easier for the retrieval algorithm to find matching book representations so that there is no room for other types of similarities to improve upon this. This suggests that social re-ranking methods have more merit in situations where user tend to use short queries, e.g., like in Web search engines.

Personalized re-ranking does not appear to work as well as non-personalized re-ranking. The most likely explanation for this is that LibraryThing topic creators typically ask for targeted recommendations on books they do not know anything about yet and do not have in their catalog yet. However, re-ranking the results lists towards a user's past books biases the results list to a ranking that is in fact *more* like books they already know about as opposed to new and relevant books.

## 5 Submitted runs

We selected six automatic runs for submission to INEX[6] based on the results of our content-based and social re-ranking runs. Two of these submitted runs were

[6] Our participant ID was 54.

content-based runs, the other four were social re-ranking-based runs. Since the re-ranking approaches did not benefit using all topic fields, we submitted three re-ranking runs based on the title and all-doc-fields baseline and one re-ranking run based on the all-topic-fields and all-doc-fields run.

**Run 1** (title.all-doc-fields) This run used the titles of the test topics and ran this against the index containing all available document fields.

**Run 2** (all-topic-fields.all-doc-fields) This run used all topic fields combined and ran this against the index containing all available document fields.

**Run 3** (all-topic-fields.pers-similarity.$\alpha$=0.99) This run applies the personalized re-ranking approach (pers-similarity) to run 2 with $\alpha$ set to 0.99; the value producing the highest NDCG scores yet not equal to 1.0.

**Run 4** (title.pers-similarity.$\alpha$=0.65) This run applies the personalized re-ranking approach (pers-similarity) to run 1 with $\alpha$ set to 0.65, which provided the best results for run 1 on the training set.

**Run 5** (title.II-similarity.$\alpha$=0.94) This run applies the re-ranking approach based on Amazon's "*similar products*" information (II-similarity) to run 1 with $\alpha$ set to 0.94, which provided the best results for run 1 on the training set.

**Run 6** (title.IUTA-similarity.$\alpha$=0.97) This run applies the re-ranking approach based on the combination of the three information sources (IUTA-similarity) to run 1 with $\alpha$ set to 0.97, which provided the best results for run 1 on the training set.

## 6 Results

The runs submitted to the INEX 2012 Social Book Search track were evaluated using graded relevance judgments. Books suggested by members other than the topic creator are considered relevant suggestions and received the relevance value 1. Books that are added by the topic creator to his/her LibraryThing catalog after creating the topic are considered the best suggestions and receive the relevance value 4. All runs were evaluated using NDCG@10, P@10, MRR, with NDCG@10 as the main metric. Table 3 shows the official evaluation results.

**Table 3.** Results of the six submitted runs on the test set, evaluated using all 301 topics with relevance judgments extracted from the LibraryThing forum topics. The best run scores are printed in bold.

| Run # | Run description | NDCG@10 | P@10 | MRR |
|---|---|---|---|---|
| 1 | title.all-doc-fields | 0.0678 | 0.0583 | 0.1341 |
| 2 | all-topic-fields.all-doc-fields | **0.1492** | **0.1198** | **0.3069** |
| 3 | all-topic-fields.pers-similarity.$\alpha$=0.99 | 0.1488 | **0.1198** | 0.3066 |
| 4 | title.pers-similarity.$\alpha$=0.65 | 0.0875 | 0.0719 | 0.1762 |
| 5 | title.II-similarity.$\alpha$=0.94 | 0.1173 | 0.1073 | 0.2558 |
| 6 | title.IUTA-similarity.$\alpha$=0.97 | 0.0958 | 0.0823 | 0.2392 |

We see that, unsurprisingly, the best-performing run on all 301 topics was run 2 with an NCDG@10 of 0.1492. Run 2 used all available topic fields and document fields. Again we see that re-ranking does not improve over the baseline when using all available topic fields. When using the title representation, we see the same performance improvements as on the training set. Run 5, for example, improves over the title baseline by 73.0%.

## 7 Discussion & Conclusions

On both the training and the test sets the best results were achieved by combining all topic and document fields. This shows continued support for the principle of polyrepresentation [11] which states that combining cognitively and structurally different representations of the information needs and documents will increase the likelihood of finding relevant documents. Adding extra controlled metadata from BL and LoC did not benefit the retrieval results however.

We also experimented with different re-ranking approaches where all the books retrieved in a run were able to contribute the final scores of each separate book by weighting those scores by their similarity to the target book. We examined the usefulness of different information sources for calculating these book similarities, such as user ratings, tags, authorship, and Amazon's "*similar products*" information. We found that all re-ranking approaches are successful when using shorter queries; longer topic representations did not benefit from re-ranking. Although all re-ranking approach improved retrieval results using the title representations as our topics, we found that Amazon's "*similar products*" information—being based on the complete set of Amazon user ratings—provides the best performance.

Personalized re-ranking did not work as well as the non-personalized methods, which is likely due its inappropriate for the recommendation task: the goal is not to find books similar to what the user has read in the past, but new books that are unlike the user's past interests.

## References


1. 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) 11–36
2. Bogers, T., Christensen, K.W., Larsen, B.: RSLIS at INEX 2011: Social Book Search Track. In Geva, S., Kamps, J., Schenkel, R., eds.: INEX 2011: Proceedings of the 10th International Workshop of the Initiative for the Evaluation of XML Retrieval. Volume 7424 of Lecture Notes in Computer Science., Berlin, Heidelberg, Springer Verlag (2012) 45–56
3. Zhai, C., Lafferty, J.: A Study of Smoothing Methods for Language Models Applied to Information Retrieval. ACM Transactions on Information Systems **22**(2) (2004) 179–214
4. Järvelin, K., Kekäläinen, J.: Cumulated Gain-based Evaluation of IR Techniques. ACM Transactions on Information Systems **20**(4) (2002) 422–446
5. Furnas, G., Landauer, T., Gomez, L., Dumais, S.T.: The Vocabulary Problem in Human-System Communication. Communications of the ACM **30**(11) (1987) 964–971

6. Renda, M.E., Straccia, U.: Web Metasearch: Rank vs. Score-based Rank Aggregation Methods. In: SAC '03: Proceedings of the 2003 ACM Symposium on Applied Computing, New York, NY, USA, ACM (2003) 841–846
7. Wang, J., de Vries, A.P., Reinders, M.J.: Unifying User-based and Item-based Collaborative Filtering Approaches by Similarity Fusion. In: SIGIR '06: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, ACM (2006) 501–508
8. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-Based Collaborative Filtering Recommendation Algorithms. In: WWW '01: Proceedings of the 10th International Conference on World Wide Web, New York, NY, USA, ACM (2001) 285–295
9. Linden, G., Smith, B., York, J.: Amazon.com Recommendations: Item-to-Item Collaborative Filtering. IEEE Internet Computing **7**(1) (2003) 76–80
10. Tso-Sutter, K.H.L., Marinho, L.B., Schmidt-Thieme, L.: Tag-aware Recommender Systems by Fusion of Collaborative Filtering Algorithms. In: SAC '08: Proceedings of the 2008 ACM symposium on Applied computing, New York, NY, USA, ACM (2008) 1995–1999
11. Ingwersen, P.: Cognitive Perspectives of Information Retrieval Interaction: Elements of a Cognitive IR Theory. Journal of Documentation **52**(1) (1996) 3–50