Combining Tags and Reviews to Improve Social Book Search Performance

Messaoud Chaa[1,2], Omar Nouali[1], and Patrice Bellot[3]

[1] Centre de Recherche sur l'Information Scientifique et Technique, Alger, Algeria
[2] Université de Bejaia, 06000 Bejaia, Algeria
[3] Aix-Marseille Université, CNRS, LSIS UMR 7296, 13397, Marseille, France
mchaa@cerist.dz, onouali@cerist.dz, patrice.bellot@lsis.org

Abstract. The emergence of Web 2.0 and social networks have provided important amounts of information that led researchers from different fields to exploit it. Social information retrieval is one of the areas that aim to use this social information to improve the information retrieval performance. This information can be textual, like tags or reviews, or non textual like ratings, number of likes, number of shares, etc. In this paper, we focus on the integration of social textual information in the research model. As it seems logical that integrating tags in the retrieval model should not be in the same way taken to integrate reviews, we will analyze the different influences of using tags and reviews on both the settings of retrieval parameters and the retrieval effectiveness. After several experiments, on the CLEF social book search collection, we concluded that combining the results obtained from two separate indexes and two models with specific parameters for tags and reviews gives good results compared to when using a single index and a single model.

Keywords: Social Information Retrieval, Social Book Search, Document Length Normalization, User-Generated Content, Tag Based Model, Review Based Model.

1 Introduction

The rise of social media has changed the role of users in the Web. From simple users who just consume information to users that can produce it. Many kinds of social information are generated like tags, reviews, rating, users' relation or users' interests. This social information has been exploited in several application domains such as marketing [11], commerce [20], etc. The field of information retrieval is no exception to this. A few years ago, a new area of research namely social information retrieval (SIR) emerged and gained popularity. The objective of SIR systems is to identify and integrate that social information (tags, reviews, ratings...) in a search process in order to improve the IR performance [3, 17].

On the other hand, social digital libraries such as LibraryThing[4] and Goodreads[5] are social cataloging websites for books that enable users to store and share book

[4] www.librarything.com
[5] www.goodreads.com

catalogues. Users can rate, review, tag, and discuss their books. They can also create groups with other members, create discussions, and ask for book recommendations. While suggestions and recommendations can be sent to users in response to their request. This generates a large volume of data for books and represent a great challenge for information retrieval tasks in general and book search in particular. In order to exploit this wealth of information, the CLEF track social book search (SBS) has come to use the social information to search and develop techniques to support users in book search tasks.

2 Social Book Search

Social book search (SBS)[6] was a CLEF lab (2011-2016) that particularly investigated book search in social media. It was interested in the use of user-generated content from social media to support users of LibraryThing forum in finding documents(books)[7] that interest them and that are relevant to their request [13,14]. The organizers of INEX SBS have used Amazon[8] and Librarything(LT) to provide a document collection which consists of 2.8 million books. This collection contains both textual and non-textual social information about books. Textual social information consists of users' reviews and tags that are respectively extracted from Amazon and LT. As to non-textual information like rating, number of time a book is catalogued, etc, they are extracted from LT whereas ratings, number of reviews, of total votes and helpful votes are extracted from Amazon. In order to evaluate systems in SBS, a set of topics with relevance assessments are provided. These topics are based on discussion threads from LT forums, where users express their needs and ask suggestions and recommendations about books to other forum users. These topics contain many fields : group, title, narrative and examples (see Fig. 1). The group field designates the discussion group in which a user posts their thread, the title is the short representation of the topic which often contains a brief summary about the user's need. Narrative is a long representation of the topic in which the user utilizes natural language to explain their needs in details. As to the field examples, it consists of some similar books add by some LT users in order to indicate the kind of books they want.

3 Related Works

Since its first edition in 2011, participants of SBS have proposed approaches and submitted their results (runs) to get them evaluated. The majority of these approaches use textual information (tags and reviews) to estimate the initial score of books then non-textual information like (rating, popularity, number of tags, number of reviews, profile of user, etc) is used to re-rank the initial ranking and improve the initial text-based search results.

[6] www.social-book-search.humanities.uva.nl
[7] We use the terms "document" and "book" interchangeably in this paper.
[8] www.amazon.com

<topic id="41306"> <request>Spanish Civil War : For Whom The Bell Tolls :: French Revolution: ?? I don't think there's a right answer, I'm just looking for suggestions for a good fictional book set during the French Revolution. (My interest was piqued after reading the Jim Shepard short story "Sans Farine" in The Best American Short Stories 2007.) <request>
<group>Book talk</group>
<title>Fill in this historical fiction analogy</title>
<examples>
<example>
<booktitle>For Whom the Bell Tolls</booktitle>
<author>Ernest Hemingway</author>
<workid>10084</workid>
</example>
<example>
<booktitle>The Best American Short Stories 2007</booktitle>
<author>Stephen King</author>
<workid>3539369</workid>
</example>
</examples>

Fig. 1. XML file representing an example topic in Social Book Search.

The authors in [2] used the language modeling with Jelinek-Mercer (JM) smoothing to build the initial content based results. They also experimented different re-ranking approaches using different information sources, such as user ratings, tags, authorship, and Amazon's similar products. The results show that the re-ranking approaches are often successful.

In [7,22], the probability of the query content produced by the language model is used to rank the documents based on textual information. Ratings, number of reviews, popularity and high frequent books are then used to re-rank the initial ranking. Finally, random forest was used to learn the different combinations of scores and the results are better than the initial ranking.

In [8] the BM25F model was used to optimize the weight of the four book fields (title, summary, tags, reviews). Popularity, reputation, ratings and similarity between users were used to improve the results however, their integration did not give any improvement. [10] employ the textual model BM25 and enhance it by using social signals such as rating. Finally, they applied a random forest learning to improve the results by including non-textual modalities like price and number of pages according to the user's preferences. This approach improves the results and shows good performances.

The authors in [1] used two textual retrieval model Divergence from randomness model(DFR) and Sequential Dependence Model(SDM) before combining the results with the ratings of books. The best results were obtained when using DFR model with textual information only. In [4, 5], the authors use only tags as textual information about books. They investigate the representation of

the query by transforming the long verbose queries to a reduced queries before applying the BM15 retrieval model. The result of this approach shows also good performance despite not using Amazon reviews at all.

To sum up, the majority of these works used textual information as a baseline before using non-textual information for reranking. As for indexing, they used users' tags and reviews together to represent books. However, some of them [2,8, 12,14] studied the influence of using tags only, reviews only or tags with reviews as books representation on the retrieval performance. To our knowledge, there is no study that has adapted a specific model for each document representation. Hence, when combining tags and reviews, the authors in [2, 8, 12, 14] used the same retrieval model to compute scores and rank documents for each query.

In this paper, we use both textual and non textual information of books as used in the studies mentioned above. However, instead of using one index for all textual fields and one function to compute scores of documents, we build a separate index for each field. We build two models, Tag Based Model (TBM) and Review Based Model (RBM) which respectively use the index of tags and the index of reviews. Then, we analyse the different influences of using tags and reviews on the settings of retrieval parameters as well as on the retrieval effectiveness. Then, we combine the results of the two models to form the textual score. Finally, we combine textual and non textual score to form the final result.

4 Experimental Setup

In our experiments we have used the collection provided by the organizers of SBS. For documents, we have considered two kinds of representations : LT users' tags and Amazon users' reviews. Table 1 summarizes the statistics of the collection. The numbers of tokens and single terms are calculated after stopword removal and Porter stemming. Table 1 shows that the number of books that have been tagged is greater than the number of books that have been reviewed however, the number of tokens (all occurrences) in reviews are greater than those of tags. This because, for reviews the users use natural language to give their opinions and speak freely about books. However for tags the users assign a few keywords or terms to describe books.

Table 1. Statistics on tags and reviews of SBS collection

Number of books	2,781,400
Number of books that have been reviewed at least once	1,915,336
Number of books that have been tagged at least once	2,306,368
Number of tokens in reviews	1,161,240,462
Number of single terms in reviews	1,135,910
Number of tokens in tags	246,552,598
Number of single terms in tags	194,487

As to queries, we have used all the six-year topics (1646 topics) provided by SBS from 2011 to 2016. Because each query is composed of title and narrative, we have considered two types of queries: short and long queries. The short query is constructed from the title of the topic while the long query is the narrative and title+narrative.
The prime target of our study is to investigate the impact of using the different query representations as well as the two representations of documents on the retrieval performance. To achieve this, we built three indexes, the first contains tags assigned by users to books in LT, the second contains reviews extracted from Amazon while the last one merges tags and reviews in the same index. The Terrier IR platform [16] was used to index the collection by applying basic stopword filtering and Porter stemming algorithm. The BM25 model [18] was used for querying. Using the BM25 model, the relevance score of a book d for query Q is given by:

$$S(d,Q) = \sum_{t \in Q} \frac{(k_1+1)tf_{td}}{tf_{td} + k_1(1-b+b.\frac{|D|}{avgdl})}.idf(t).\frac{(k_3+1)tf_{tq}}{k_3+tf_{tq}} \quad (1)$$

Where $\mathrm{tf_{td}}$ and $\mathrm{tf_{tq}}$ are respectively the frequency of term t in document d and in query Q. the three free parameters of the function are : $\mathrm{k_1}$ and $\mathrm{k_3}$ that respectively controls term frequency scaling of the document and the query, the parameter b controls the document length normalization. $idf(t)$ is the inverse document frequency of term t, given 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 where the term t appears, and $|D|$ is the number of documents in the collection.

4.1 Length Normalization vs Document and Query Representation

Document length normalization is a technique that attempts to adjust the term frequency or the relevance score in order to normalize the effect of document length on the document ranking. Several works [6,9,15] show that this technique has an important impact on the performance of the model.

In order to determine the sensitivity of the model performance as to the length normalization, using the different cases of document representations as well as query representations, we have set the BM25 standard parameters for $k1$ and $k3$ ($k1 = 2, k3 = 1000$) and varied the length normalization parameter b from 0 to 0.75 (in steps of 0.05) then we evaluated the results in terms of NDCG@10 on both indexes (tags and reviews). The results obtained for all queries together are shown in Figure 2.
From Figure 2, we can see that in TBM, the performance of the model is very sensitive to the normalization of the length of the document. NDCG@10 drops from 0.1415 (b=0) to 0.0375 (b=0.75) in the case of title (short queries). It

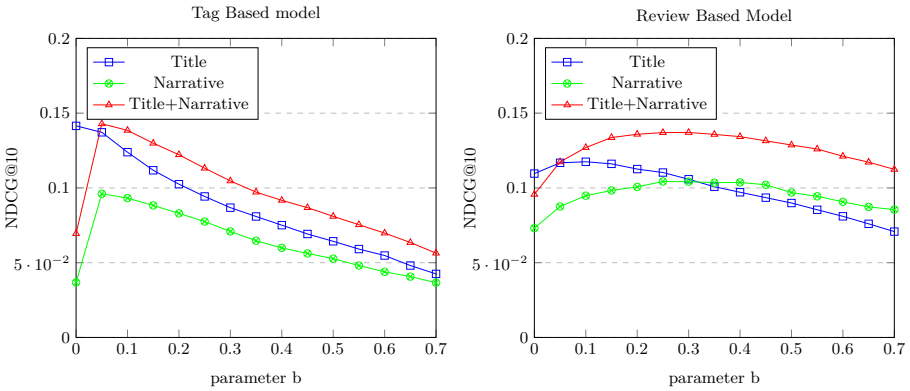


Fig. 2. Sensitivity of ndcg@10 for length normalization for Tag based model and Review based model.

increases from 0.0368 (b=0) to 0.0961 (b=0.05) then drops to 0.0321 (b=0.75) when using narrative (long queries).The same sensitivity of the model was shown When combining title and narrative to represent the query, NDCG@10 increases from 0.0695 (b=0) to 0.1429 (b=0.1) and then drops to 0.0490 (b=0.75).

In the case of RBM, the model performance is not very sensitive to the normalization of the document length compared to TBM. The values of NDCG@10 are very close so there is no big difference between them especially when varying b from 0 to 0.35. NDCG@10 is 0.1096 (b=0) and 0.1003 (b=0.35) in the case of short queries. The same measure increases from 0.0731 (b=0) to 0.1042 (b=0.25) and then decreases to 0.0803 (b=0.75) in the case of long queries. In the same way, the NDCG@10 increases from 0.0957 (b=0) to 0.1370 (b=0.25) and then decreases to 0.1062 (b=0.75) when combining title and narrative.

The same figure clearly shows that in short queries the best performance is obtained when b is very small, b=0 for the tags (no length normalization is required) and b=0.1 for reviews. However, when using long queries we find that the best performance is obtained when b=0.1 for tags and b=0.25 for reviews (length normalization is required). We can conclude that the setting parameters of the model are not the same in the TBM model or in the RBM model. Each of them has its specific parameters.

Because these evaluations are obtained for all queries together and to make our results more meaningful, we have decided to learn the parameter b by selecting the set of topics of each year as the testing set and the remaining sets of topics (of other years) as the training set. The evaluation in term of NDCG@10 of the results, for Single index(b=0.3), TBM model (b=0.05) and RBM Model(b=0.25), obtained after training and testing is shown in Table 2.

As shown in Table 2, TBM gave the best results, compared to RBM and Single index, except for 2011 when RBM gave the best results. By the way, we were surprised that, in most cases, using tags only (TBM) or reviews only

Table 2. Results in term of NDCG@10 of the three index , using Title+Narrative as a representation of the topic; best results are shown in bold.

Year	Single Index	TBM	RBM
2011	0.2810	0.2459	**0.3007**
2012	0.1387	**0.2012**	0.1479
2013	0.1143	**0.1193**	0.1191
2014	0.1169	**0.1291**	0.1175
2015	0.0682	**0.1222**	0.0794
2016	0.0803	**0.0951**	0.0906
All	0.1307	**0.1429**	0.1370

(RBM) to index documents gave better results compared to when using tags and reviews together. The only exceptional case was in 2011 where single index (NDCG=0.2810) was better than TBM (NDCG=0.2459).

4.2 Query Expansion Using Example Books

In addition to title and narrative, the topics of SBS contain an example field which includes a list of books that the user has mentioned in their topic to show the kind of books he/she requests. In our approach, we consider this list of books as relevant books; therefore we use the query expansion technique to extract the highly relevant terms of these books and expand the original query. The Rocchio function [19] used to expand the query is as follows:

$$\overrightarrow{Q}_{new} = \overrightarrow{Q}_{T+N} + \frac{\beta}{|EXP|} \sum_{d \in EXP} \overrightarrow{d} \tag{3}$$

Where $\overrightarrow{Q}_{new}$ is the expansion query, $\overrightarrow{Q}_{T+N}$ is the original query represented by Title+Narrative. $\overrightarrow{d}$ denotes the weighted term vector of the example book d using the default term weighting model Bo1(Bose-Einstein 1). EXP is the set of example books and $|EXP|$ is the number of example books mentioned in the topic. The function was used with their default parameter settings β =0.4, and the number of terms selected from each example book was set to 10.

Table 3 shows the results after the query expansion for the Single index, TBM model as well as for the RBM model. The same table shows that the query expansion technique has improved the results for the three indexes. We also notice that TBM gave the best results for all cases except for 2011 when RBM gave the best NDCG@10 (0.3418) and for 2013 when Single index has an NDCG@10=0.1496.

4.3 Combining the Scores of the Two Models

Once the parameters of TBM and RBM are optimized, a combination of the two scores obtained is necessary to obtain the final textual score of each book with

Table 3. NDCG@10 results obtained after applying a query expansion technique.

Year	Single Index		TBM		RBM	
	Before expansion	After expansion	Before expansion	After expansion	Before expansion	After expansion
2011	0.281	0.3310	0.2459	0.2725	0.3007	**0.3418**
2012	0.1387	0.1667	0.2012	**0.2341**	0.1479	0.1754
2013	0.1143	**0.1496**	0.1193	0.1452	0.1191	0.1478
2014	0.1169	0.1436	0.1291	**0.1525**	0.1175	0.1423
2015	0.0682	0.0887	0.1222	**0.1409**	0.0794	0.0907
2016	0.0803	0.1088	0.0951	**0.1367**	0.0906	0.1205
All	0.1307	0.1619	0.1429	**0.1686**	0.1370	0.1639

respect to the query. The linear combination function is as follows:

$$S(d,Q) = \alpha.S_{TBM}(d,Q) + (1-\alpha).S_{RBM}(d,Q) \tag{4}$$

Where $S(d,Q)$ is the final score of document d with respect to query Q. $S_{TBM}(d,Q)$ and $S_{RBM}(d,Q)$ are the scores of document d with respect to query Q respectively obtained from TBM and RBM models. The query Q is represented by Title+Narrative and expanded using similar books as explained in the previous section. α [0 1] is a free parameter that controls the weight of the two models.This parameter was tuned using six-fold cross-validation in the same way it was performed to tuned the parameter b as indicated in section 4.1. Thus, a single set of topics of one year is used for testing the model and the remaining five-year topics are used as a training set. After the process is repeated six times (once for each year), the results have been summed up in Table 4. This table shows the results in term of NDCG@10 obtained from the combination and compares them to the results obtained from the previous experiments. The best results of the combination are obtained when $\alpha \in$ is set to 0.4.

Table 4. NDCG@10 obtained by the combination after tuning the parameter α using six-fold cross-validation compared to a single index, RBM model and TBM Model. Asterisks indicate statistically significant differences compared with single index (Student's t-test, $P < 0.05$)

Year	Single Index	TBM	RBM	Combination
2011	0.3310	0.2725	0.3418	**0.3595**
2012	0.1667	0.2341	0.1754	**0.2425***
2013	0.1496	0.1452	0.1478	**0.1888***
2014	0.1436	0.1525	0.1423	**0.1886***
2015	0.0887	0.1409	0.0907	**0.1526***
2016	0.1088	0.1367	0.1205	**0.1793***
All	0.1619	0.1686	0.1639	**0.2091***

From the results, we note that for all years the combination of the two scores gave good results, compared to the results of each index. For all queries, the NDCG@10 increased from 0.1619 (single model), 0.1686 (TBM model) and 0.1639 (RBM model) to 0.2091 when combining the two scores so there is an improvement of 29.15%, 24.02% and 27.54% compared to the three models. This results shows that the technique of using two separate indexes and combining the results is an effective technique to get best results.

4.4 Non textual Information to Re-rank documents

The non textual information of documents in social media like number of likes, number of rating, number of times the document was catalogued or rated represents an important information and can be used to re-rank the documents to improve the results. Several Re-ranking approaches were proposed by [22] at INEX2014 and [2] in 2012 , which proved to be effective. In this paper we combine the textual score obtained above with the the number of times the book was rated to re-rank the documents. The combination of scores is calculated by the flowing function:

$$S(d,Q) = \lambda.S_{Textual}(d,Q) + (1-\lambda).S_{Non-textual}(d) \tag{5}$$

Where $S(d,Q)$ is the final score of document d with respect to query Q. $S_{Textual}(d,Q)$ is the textual score of document d obtained by combining the scores of TBM and RBM as explained in the previous section. λ [0 1] is a free parameter that controls the weight of the two scores. $S_{Non-textual}(d)$ is the normalized non-textual score of document d calculated as follow:

$$S_{Non-textual} = \frac{Nb_rated - Min_nb_rated}{Max_nb_rated - Min_nb_rated} \tag{6}$$

Where nb_rated is the number of times the document is rated, Min_nb_rated and Max_nb_rated is the minimum and the maximum of the number of times that all books of the collection have. Table 5 shows the results obtained when using the non-textual information for re-ranking. The best results obtained when $\lambda \in$ was set to 0.9. From this Table we show that when using the Non-Textual information to re-rank documents gave good results of all years of topics, except in 2011 when the NDCG@10(0.3595) obtained by using the textual information only is better than the NDCG@10(0.3551) after re-ranking the documents.

Finally, we compare the performance of our best results, obtained by our approach, to the best official and non official runs. The official runs are those that have been submitted by participants to SBS during the six last years. The non official runs are the results obtained recently in the works of [21,23]. Table 6 represents the comparative results. The results show that the NDCG@10 value of our approach is better than the best official runs of the four years (2011, 2012, 2013 and 2014) but lower than the best runs of the two years (2015 and 2016). This table shows also that our approach gave good results in the three first years compared to the non official runs.

Table 5. NDCG@10 obtained by the combination of Textual and Non-textual scores

	Textual score only	Textual and Non-Textual scores
2011	**0.3595**	0.3551
2012	0.2425	**0.2469**
2013	0.1888	**0.1907**
2014	0.1886	**0.1943**
2015	0.1526	**0.1585**
2016	0.1793	**0.1935**
All	0.2091	**0.2133**

Table 6. Comparison between the results of our approach and the best runs submitted to the different years of SBS; best results are shown in bold.

	Our approach	Best non official runs	Best official runs
2011	**0.3551**	0.3423	0.3101
2012	**0.2469**	0.2325	0.1456
2013	**0.1907**	0.1856	0.1361
2014	0.1943	**0.1960**	0.1420
2015	0.1585	**0.2040**	0.1870
2016	0.1935	**0.2157**	**0.2157**

5 Analysis

From the results, we have noticed that TBM model requires a smaller value of b for optimal performance whether for short or long queries compared to RBM model. This led us to ask the following question :

Why document length normalization is required when using reviews as a document representation and not required when using tags as document representation?

As an answer to the question, this may be due to the fact that the users' reviews are a natural language text in which users can repeat freely the same term many times in the same review. Hence, the frequency value of any given term present in the query will be increased thereby increasing the relevance scores of long documents that contain long reviews. That is why document length normalization is required to penalize very long documents. Contrary to the reviews, the users' tags are keywords that users assign them to documents and the same user can not assign the same term many times to the same document. Hence, for long documents when the frequency value of any query term is great this means that this document was tagged by several users using the same term then it is relevant to the topic and it is unreasonable to penalize them. Therefore, the document length normalization is not required for tag representation.

6 Conclusion

In this paper, we have studied the exploitation of user tags and reviews in social book search. Three indexes have been created, the first for tags, the second for reviews and the third that merges the two. After several experimentations, we concluded that the Tag Based Model does not require a document length normalization especially for short queries. The best results of this model were obtained when b has very low values, This may be due to the fact that the number of tags assigned to books by users cannot be regarded as a length of text documents. However, the Review Based Model requires the document length normalization, this may be because the user reviews are long and be seen as a classical textual document. We have noticed that using two indexes for tags and reviews separately and combining the results of the models gives good and better results compared to when using a single index and a single model. We have also demonstrated that the proposed combination has given satisfactory results, especially when using non-textual information to re-rank documents, and outperforms the best runs submitted to SBS in the four first years.

References

1. Benkoussas, C., Hamdan, H., Albitar, S., Ollagnier, A., Bellot, P.: Collaborative filtering for book recommandation. In: CLEF (Working Notes). pp. 501–507 (2014)
2. Bogers, T., Larsen, B.: Rslis at inex 2012: Social book search track. In: INEX. vol. 12, pp. 97–108 (2012)
3. Bouadjenek, M.R., Hacid, H., Bouzeghoub, M.: Social networks and information retrieval, how are they converging? a survey, a taxonomy and an analysis of social information retrieval approaches and platforms. Information Systems 56, 1–18 (2016)
4. Chaa, M., Nouali, O., Bellot, P.: Verbose query reduction by learning to rank for social book search track. In: CLEF (Working Notes). pp. 1072–1078 (2016)
5. Chaa, M., Nouali, O., Bellot, P.: New technique to deal with verbose queries in social book search. In: Proceedings of the International Conference on Web Intelligence. pp. 799–806. ACM (2017)
6. Cummins, R., O'Riordan, C.: The effect of query length on normalisation in information retrieval. In: Irish Conference on Artificial Intelligence and Cognitive Science. pp. 26–32. Springer (2009)
7. Feng, S.H., Zhang, B.W., Yin, X.C., Jin, Z.X., Jin, J.L., Wu, J.W., Zhang, L.L., Pan, H.J., Fang, F., Zhou, F.: Ustb at social book search 2016 suggestion task: Active books set and re-ranking. In: CLEF (Working Notes). pp. 1089–1096 (2016)
8. Hafsi, M., Géry, M., Beigbeder, M.: Lahc at inex 2014: Social book search track. In: Working Notes for CLEF 2014 Conference (2014)
9. He, B., Ounis, I.: Term frequency normalisation tuning for bm25 and dfr models. In: ECIR. vol. 5, pp. 200–214. Springer (2005)
10. Imhof, M., Badache, I., Boughanem, M.: Multimodal social book search. In: 6th Conference on Multilingual and Multimodal Information Access Evaluation (CLEF 2015). pp. pp–1 (2015)
11. Kim, A.J., Ko, E.: Do social media marketing activities enhance customer equity? an empirical study of luxury fashion brand. Journal of Business Research 65(10), 1480–1486 (2012)

12. Koolen, M.: " user reviews in the search index? that'll never work!". In: ECIR. vol. 8416, pp. 323–334. Springer (2014)
13. Koolen, M., Bogers, T., Gäde, M., Hall, M., Hendrickx, I., Huurdeman, H., Kamps, J., Skov, M., Verberne, S., Walsh, D.: Overview of the clef 2016 social book search lab. In: International Conference of the Cross-Language Evaluation Forum for European Languages. pp. 351–370. Springer (2016)
14. Koolen, M., Kamps, J., Kazai, G.: Social book search: comparing topical relevance judgements and book suggestions for evaluation. In: Proceedings of the 21st ACM international conference on Information and knowledge management. pp. 185–194. ACM (2012)
15. Lv, Y., Zhai, C.: Lower-bounding term frequency normalization. In: Proceedings of the 20th ACM international conference on Information and knowledge management. pp. 7–16. ACM (2011)
16. Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C., Lioma, C.: Terrier: A high performance and scalable information retrieval platform. In: Proceedings of the OSIR Workshop. pp. 18–25 (2006)
17. Park, D.H., Liu, M., Zhai, C., Wang, H.: Leveraging user reviews to improve accuracy for mobile app retrieval. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 533–542. ACM (2015)
18. Robertson, S.E., Walker, S., Beaulieu, M.: Experimentation as a way of life: Okapi at trec. Information processing & management 36(1), 95–108 (2000)
19. Rocchio, J.J.: Relevance feedback in information retrieval. The Smart retrieval system-experiments in automatic document processing (1971)
20. Swamynathan, G., Wilson, C., Boe, B., Almeroth, K., Zhao, B.Y.: Do social networks improve e-commerce?: a study on social marketplaces. In: Proceedings of the first workshop on Online social networks. pp. 1–6. ACM (2008)
21. Yin, X.C., Zhang, B.W., Cui, X.P., Qu, J., Geng, B., Zhou, F., Song, L., Hao, H.W.: Isart: A generic framework for searching books with social information. PloS one 11(2), e0148479 (2016)
22. Zhang, B.W., Yin, X.C., Cui, X.P., Qu, J., Geng, B., Zhou, F., Hao, H.W.: Ustb at inex2014: Social book search track. In: CLEF (Working Notes). pp. 536–542 (2014)
23. Zhang, B.W., Yin, X.C., Zhou, F., Jin, J.L.: Building your own reading list anytime via embedding relevance, quality, timeliness and diversity. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1109–1112. ACM (2017)