RSLIS at INEX 2011: Social Book Search Track

Toine Bogers[1], Kirstine Wilfred Christensen[2], and Birger Larsen[1]

[1] Royal School of Library and Information Science
Birketinget 6, 2300 Copenhagen, Denmark
{tb,blar}@iva.dk
[2] DBC, Tempovej 7, 2750 Ballerup, Denmark
kwc@dbc.dk

Abstract. In this paper, we describe our participation in the INEX 2011 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 social features. We find that the best results are obtained using all available document fields and topic representations.

Keywords: XML retrieval, social tagging, controlled metadata, book recommendation.

1 Introduction

In this paper, we describe our participation in the INEX 2011 Social Book Search track [1]. Our goals for the Social Book Search task were (1) to investigate the contribution of different types of document metadata, both social and controlled; and (2) to examine the effectiveness of using social features to re-rank the initial 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. 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 2011 Social Book Search track. This collection contains XML representations of 2.8 million books, with the book representation data crawled from both Amazon.com and LibraryThing.

A manual inspection of the collection revealed the presence of several XML fields that are unlikely to contribute to the successful retrieval of relevant books.

S. Geva, J. Kamps, and R. Schenkel (Eds.): INEX 2011, LNCS 7424, pp. 45–56, 2012.

Examples include XML fields like <image>, <listprice>, and <binding>. While it is certainly not impossible that a user would be interested only in books in a certain price range or in certain bindings, we did not expect this to be likely in this track's particular retrieval scenario of recommending books based on a topical request. We therefore manually identified 22 such fields and removed them from the book representations.

In addition, we converted the original XML schema into a simplified version. After these pre-processing steps, we were left with the following 19 content-bearing XML fields in our collection: <isbn>, <title>, <publisher>, <editorial>[1], <creator>[2], <series>, <award>, <character>, <place>, <blurber>, <epigraph>, <firstwords>, <lastwords>, <quotation>, <dewey>, <subject>, <browseNode>, <review>, and <tag>.

One of the original fields (<dewey>) contains the numeric code representing the Dewey Decimal System category that was assigned to a book. We replaced these numeric Dewey codes 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>.

2.2 Field Categories and Indexing

The 19 remaining 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. In addition, we combined all five groups of relevant fields for an index containing all fields. This resulted in the following six 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).

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

[1] Our <editorial> fields contain a concatenation of the original <source> and <content> fields for each editorial review.

[2] For our <creator> field, we disregard the different roles the creators could have in the original XML schema and simply treat all roles the same.

[3] Available at http://www.library.illinois.edu/ugl/about/dewey.html

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: <browseNode>, <dewey>, and <subject>.

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. The user reviews from the <review> fields were indexed in two different ways: (1) all user reviews belonging to a single book were combined in a single document representation for that book, and (2) each book review was indexed and retrieved separately. The former book-centric review index reviews is used in Section 3; the latter review-centric index reviews-split is used in our social re-ranking approach described in Section 4.

We used the Indri 5.0 retrieval toolkit[4] 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 2011 Social Book Search track two set of topics were released with requests for book recommendations based on textual description of the user's information need: a training set and a test set. Both topic sets were extracted from the LibraryThing forum. The training set consisted of 43 topics and also contained relevance judgments, which were crawled from the LibraryThing forum messages. Each book that was mentioned in the forum thread was deemed relevant, meaning these could possibly be incomplete or inaccurate. Despite these known limitations, we used the 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 containing 211 topics is the topic set used to rank and compare the different participants' systems at INEX. The results listed in Section 6 were obtained on this test set.

Each topic in the two sets are represented by several different fields, with some fields only occurring in the test set. In our experiments with the training and the test set, we restricted ourselves to automatic runs using the following three fields (partly based on a manual inspection of their usefulness for retrieval):

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.

[4] Available at `http://www.lemurproject.org/`

Group. The LibraryThing forum is divided into different groups covering different topics. Runs that only use the <group> field (i.e., the name of the LibraryThing group as query) are referred to as group.

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> field typically contains the richest description of the topic and runs using only this field are referred to as narrative.

All Topic Fields. In addition to runs using these three fields individually, we also performed runs with all three fields combined (all-topic-fields).

The test and training sets contained several other fields that we did not experiment with due to temporal constraints, such as <similar> and <dissimilar>. However, we list some of our ideas in Section 7.1.

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 5.0 toolkit. We preferred JM smoothing over Dirichlet smoothing, because previous work has shown that for longer, more verbose queries JM smoothing performs better than Dirichlet smoothing [2], which matches the richer topic descriptions provided in the training and test sets.

For the best possible performance, we optimized the λ parameter, which controls the influence of the collection language model, with higher values giving more influence to the collection language model. We varied λ in steps of 0.1, from 0.0 to 1.0 using the training set of topics. We optimized λ separately for each combination of indexes and topic sets. For each topic we retrieve up 1000 documents and we used NDCG[5] as our evaluation metric [3].

3 Content-Based Retrieval

For our first round of experiments focused on a standard content-based retrieval approach where we compared the different index and the different topic representations. We had six different indexes (all-doc-fields, metadata, content, controlled-metadata, tags, and reviews) and four different sets of topic representations (title, group, narrative, and all-topic-fields). We examined each of these pairwise combinations for a total of 24 different content-based retrieval runs. Table 1 shows the best NDCG results for each run on the training set with the optimal λ values.

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

[5] Please note that the official evaluation on the test set used NDCG@10 as an evaluation metric instead of NDCG; we were not aware of this at the time of performing our experiments.

Table 1. Results of the 24 different content-based retrieval runs on the training set using NDCG as evaluation metric. Best-performing runs for each topic representation are printed in bold. The boxed run is the best overall.

Document fields	Topic fields			
	title	narrative	group	all-topic-fields
metadata	0.2756	0.2660	0.0531	0.3373
content	0.0083	0.0091	0.0007	0.0096
controlled-metadata	0.0663	0.0481	0.0235	0.0887
tags	0.2848	0.2106	0.0691	0.3334
reviews	**0.3020**	0.2996	0.0773	0.3748
all-doc-fields	0.2644	**0.3445**	**0.0900**	**0.4436**

against the index containing all document fields (all-doc-fields). In fact, for three out of four topic sets, using all-doc-fields provides the best performance. The book-centric reviews index is close second with strong performance on all four topic sets. Finally, we observe that the content and controlled-metadata indexes result in the worst retrieval performance across all four topic sets.

When we compare the different topic sets, we see that the all-topic-fields set consistently produces the best performance, followed by the title and narrative topic sets. The group topic set generally produced the worst-performing runs.

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.

In this section, we turn our attention to the book reviews entered by Amazon's large user base. We mentioned in Section 2.1 that we indexed the user reviews from the <review> fields in two different ways: (1) all user reviews belonging to a single book were combined in a single document representation for that book (reviews), and (2) each book review was indexed and retrieved separately (reviews-split). The results of the content-based runs in the previous section showed that a book-centric approach to indexing reviews provided good performance.

Review-Centric Retrieval. However, all user reviews are not equal. Some reviewers provide more accurate, in-depth reviews than others, and in some cases reviews may be even be misleading or deceptive. This problem of spam reviews on online shopping websites such as Amazon.com is well-documented [4].

This suggests that indexing and retrieving reviews individually and then aggregating the individually retrieved reviews could be beneficial by matching the best, most topical reviews against our topics.

Our review-centric retrieval approach works as follows. First, we index all reviews separately in our reviews-split index. We then retrieve the top 1000 individual reviews for each topic (i.e., this is likely to be a mixed of different reviews for different books). This can result in several reviews covering the same book occurring in our result list, which then need to be aggregated into a single relevance score for each separate book. This problem is similar to the problem of *results fusion* in IR, where the results of *different* retrieval algorithms on the *same* collection are combined. This suggest the applicability of standard methods for results fusion as introduced by [5]. Of the six methods they investigated, we have selected the following three for aggregating the review-centric retrieval results.

- The CombMAX method takes the maximum relevance score of a document from among the different runs. In our case, this means that for each book in our results list, we take the score of the highest-retrieved individual review to be the relevance score for that book.
- The CombSUM method fuses runs by taking the sum of the relevance scores for each document separately. In our case, this means that for each book in our results list, we take the sum of the relevance scores for all reviews referring to that particular book.
- The CombMNZ method does the same as the CombSUM method, but boost the sum of relevance scores by the number of runs that actually retrieved the document. In our case, this means that for each book in our results list, we take the sum of the relevance scores for all reviews referring to that particular book, and multiply that by the number of reviews that were retrieved for that book.

Helpfulness of Reviews. One of the more popular aspects of user reviewing process on Amazon.com is that reviews can be marked as helpful or not helpful by other Amazon users. By using this information, we could ensure that the most helpful reviews have a better chance of being retrieved. We can use this information to improve the retrieval results by assigning higher weights to the most helpful reviews and thereby boosting the books associated with those reviews. The assumption behind this is that helpful reviews will be more accurate and on-topic than unhelpful reviews.

We estimate the helpfulness of a review by dividing the number of votes for helpfulness by the total number of votes for that review. For example, a review that 3 out of 5 people voted as being helpful would have a helpfulness score of 0.6. For each retrieved review i we then obtain a new relevance score $score_{weighted}(i)$ by multiplying that review's original relevance score $score_{org}(i)$ with its helpfulness score as follows:

$$score_{weighted}(i) = score_{org}(i) \times \frac{helpful\ vote\ count}{total\ vote\ count} \qquad (1)$$

This will results in the most helpful reviews having a bigger influence on the final rankings and the less helpful reviews having a smaller influence. We combine this weighting method with the three fusion methods CombMAX, CombSUM, and CombMNZ to arrive at a weighted fusion approach.

Book Ratings. In addition, users can also assign individual ratings from zero to five stars to the book they are reviewing, suggesting an additional method of taking into account the quality of the books to be retrieved. We used these ratings to influence the relevance scores of the retrieved books. For each retrieved review i we obtain a new relevance score $score_{weighted}(i)$ by multiplying that review's original relevance score $score_{org}(i)$ with its normalized rating r as follows:

$$score_{weighted}(i) = score_{org}(i) \times \frac{r}{5} \tag{2}$$

This will results in the positive reviews having a bigger influence on the final rankings and the negative reviews having a smaller influence. An open question here is whether positive reviews are indeed a better source of book recommendations than negative reviews. We combine this weighting method with the three fusion methods CombMAX, CombSUM, and CombMNZ to arrive at a weighted fusion approach.

Table 2 shows the results of the different social ranking runs for the optimal λ values. The results of the runs using the book-centric reviews index are also included for convenience.

Table 2. Results of the 9 different social ranking runs with the reviews-split index on the training set using NDCG as evaluation metric. The results of the runs using the book-centric reviews index are also included for convenience. Best-performing runs for each topic representation are printed in bold. The boxed run is the best overall using the reviews-split index.

Runs	Topic fields			
	title	narrative	group	all-topic-fields
CombMAX	0.3117	**0.3222**	0.0892	0.3457
CombSUM	**0.3377**	0.3185	**0.0982**	$\boxed{\mathbf{0.3640}}$
CombMNZ	0.3350	0.3193	**0.0982**	0.3462
CombMAX - Helpfulness	0.2603	0.2842	0.0722	0.3124
CombSUM - Helpfulness	0.2993	0.2957	0.0703	0.3204
CombMNZ - Helpfulness	0.3083	0.2983	0.0756	0.3203
CombMAX - Ratings	0.2882	0.2907	0.0804	0.3306
CombSUM - Ratings	0.3199	0.3091	0.0891	0.3332
CombMNZ - Ratings	0.3230	0.3080	0.0901	0.3320
reviews	0.3020	0.2996	0.0773	0.3748

What do the results of the social ranking approaches tell us? The best overall social ranking approach is the unweighted CombSUM method using all available topic fields, with a NDCG score of 0.3640. Looking at the unweighted fusion methods, we see that our results confirm the work of, among others [5] and [6], as the CombSUM and CombMNZ fusion methods tend to perform better than CombMAX. For the weighted fusion approaches where the weights are derived from information about review helpfulness and book ratings we see the same patterns for these three methods: CombSUM and CombMNZ outperform CombMAX.

Overall, however, the unweighted fusion methods outperform the two weighted fusion methods. This is *not* in line with previous research [7,8], where the optimal combination of weighted runs tends to outperform the unweighted variants. This could be due to the fact that the relevance assessments for the training set can be incomplete or inaccurate. Another possibility is that our weighting methods using helpfulness and ratings are not optimal. It may be that reviews that are helpful for users are not necessarily helpful for a retrieval algorithm. Analogously, increasing the influence of positive reviews over negative reviews may not be the ideal approach either. We do observe however that using weights based on book ratings seem to have a slight edge over weights derived from review helpfulness.

Finally, if we compare the book-centric and review-centric approaches, we see a mixed picture: while the best result using the reviews-split index is not as good as the best result using the reviews index, this is only true for one of the four topic sets. For the other topic sets where the retrieval algorithm has less text to work with the review-centric approach actually comes out on top.

5 Submitted Runs

We selected four automatic runs for submission to INEX[6] based on the results of our content-based and social retrieval runs. Two of these submitted runs were content-based runs, the other two were social ranking-based runs.

Run 1. title.all-doc-fields This run used the titles of the test topics[7] and ran this against the index containing all available document fields, because this index provided the best content-based results.

Run 2. all-topic-fields.all-doc-fields This run used all three topic fields combined and ran this against the index containing all available document fields. We submitted this run because this combination provided the best overall results on the training set.

Run 3. title.reviews-split.CombSUM This run used the titles of the test topics and ran this against the review-centric reviews-split index, using the unweighted CombSUM fusion method.

[6] Our participant ID was 54.

[7] While our experiments showed that using only the title topic set did not provide the best results, submitting at least one run using only the title topic set was required by the track organizers.

Run 4. all-topic-fields.reviews-split.CombSUM This run used all three topic fields combined and ran this against the review-centric reviews-split index, using the unweighted CombSUM fusion method.

6 Results

The runs submitted to the INEX Social Book Search track were examined using three different types of evaluations [1]. In all three evaluations the results were calculated using NDCG@10, P@10, MRR and MAP, with NDCG@10 being the main metric.

LibraryThing Judgments for All 211 Topics. The first evaluation was using the 211 test set topics where the relevance judgments derived from the books recommended on the LibraryThing discussion threads of the 211 topics. Table 3 shows the results of this evaluation.

Table 3. Results of the four submitted runs on the test set, evaluated using all 211 topics with relevance judgments extracted from the LibraryThing forum topics. The best run scores are printed in bold.

Runs	**NDCG@10**	**P@10**	**MRR**	**MAP**
title.all-doc-fields	0.1129	0.0801	0.1982	0.0868
all-topic-fields.all-doc-fields	0.2843	0.1910	0.4567	**0.2035**
title.reviews-split.CombSUM	0.2643	0.1858	0.4195	0.1661
all-topic-fields.reviews-split.CombSUM	**0.2991**	**0.1991**	**0.4731**	0.1945

We see that, surprisingly, the best-performing runs on all 211 topics was run 4 with an NCDG@10 of 0.2991. Run 4 used all available topic fields and the unweighted CombSUM fusion method on the review-centric reviews-split index. Run 2, with all available document and topic fields was a close second.

Amazon Mechanical Turk Judgments for 24 Topics. For the first type of evaluation the book recommendations came from LibraryThing users who actually read the book(s) they recommend. The second type of evaluation conducted by the track participants enlisted Amazon Mechanical Turk (AMT) workers for judging the relevance of the book recommendations for 24 of the 211 test topics. These 24 topics were divided so that they covered 12 fiction and 12 non-fiction book requests. The judgments were based on pools of the top 10 results of all official runs submitted to the track, evaluated using all 211 topics. Table 4 shows the results of this second type of evaluation.

We see that consistent with the results on the training set the best-performing run on the 24 selected topics was run 2 with an NCDG@10 of 0.5415. Run 2 used all available topic and document fields. Runs 3 and 4 were a close second and third.

Table 4. Results of the four submitted runs on the test set, evaluated using 24 selected topics with relevance judgments from Amazon Mechanical Turk. The best run scores are printed in bold.

Runs	NDCG@10	P@10	MRR	MAP
title.all-doc-fields	0.4508	0.4333	0.6600	0.2517
all-topic-fields.all-doc-fields	**0.5415**	0.4625	**0.8535**	**0.3223**
title.reviews-split.CombSUM	0.5207	**0.4708**	0.7779	0.2515
all-topic-fields.reviews-split.CombSUM	0.5009	0.4292	0.8049	0.2331

If we split the topics by fiction and non-fiction book requests, an interesting pattern emerges for our top two performing runs. Run 2, with all available document and topic fields, achieved an NDCG@10 score of 0.5770 on non-fiction topics, but only a score of 0.5060 on fiction topics. In contrast, our second-best performing run on the 24 AMT topics (title.reviews-split.CombSUM) performed better on fiction topics with an NDCG@10 of 0.5465 compared to a score of 0.4949 on the non-fiction topics. This suggests that the different approaches have different strengths. Topics that request recommendations for fiction books might benefit more from using the available reviews than non-fiction books, because the content and themes of such books are more difficult to capture using the different curated and user-generated types of metadata. Reviews seem to contribute more to effective retrieval here, whereas the content of non-fiction books is more easily described using the available document fields.

LibraryThing Judgments for the 24 AMT Topics. The third type of evaluation used the same 24 AMT topics from the second evaluation, but with the original LibraryThing relevance judgments. Table 5 shows the results of this third type of evaluation.

Table 5. Results of the four submitted runs on the test set, evaluated using 24 selected Amazon Mechanical Turk topics with relevance judgments extracted from the LibraryThing forum topics. The best run scores are printed in bold.

Runs	NDCG@10	P@10	MRR	MAP
title.all-doc-fields	0.0907	0.0680	0.1941	0.0607
all-topic-fields.all-doc-fields	**0.2977**	**0.1940**	**0.5225**	**0.2113**
title.reviews-split.CombSUM	0.2134	0.1720	0.3654	0.1261
all-topic-fields.reviews-split.CombSUM	0.2601	**0.1940**	0.4758	0.1515

We see that, again consistent with the results on the training set, the best-performing run on the 24 selected topics with LibraryThing judgments was run 2 with an NCDG@10 of 0.2977. Run 2 used all available topic and document fields. Run 4 was a close second and third.

We also see that for the same 24 topics, evaluation scores are much lower than for the second type of evaluation. This is probably due to the varying number of documents judged relevant for the two sets of relevance judgments. The AMT judgments were produced by pooling the first ten of each officially submitted run, thus ensuring that each of the result would be judged. For the LibraryThing judgments, 67 of the 211 topics have fewer than five judgments, which negatively influences the calculation of NDCG@10, P@10 and MAP scores.

7 Discussion and Conclusions

Both in the the training set and the test set good results were achieved by combining all topic and document fields. This shows support for the principle of polyrepresentation [9] which states that combining cognitively and structurally different representations of the information needs and documents will increase the likelihood of finding relevant documents. However, using only the split reviews as index gave in four cases in the test set even better results, which speaks against the principle of polyrepresentation.

We also examined the usefulness of user-generated metadata for book retrieval. Using tags and reviews in separate indexes showed good promise, demonstrating the value of user-generated metadata for book retrieval. In contrast, the effort that is put into curating controlled metadata was not reflected its retrieval performance. A possible explanation could be that user-generated data is much richer, describing the same book from different angles, whereas controlled metadata only reflects the angle of the library professional who assigned them.

We also experimented with a review-centric approach, where all reviews were indexed separately and fused together at a later stage. This approach yielded good results, both on the training and the test set. We attempted to boost the performance of this approach even further by using review helpfulness and book ratings as weights, but this only decreased performance. At first glance, this is surprising since a helpful review can be expected to be well-written and well-informed. The quality of a book as captured by the rating could also be expected to have an influence on the review usefulness for retrieval, as could have been expected. Our current weighting scheme was not able to adequately capture these features though.

However, experimental evidence suggests that using a review-centric approach is a more promising approach to requests for fiction books than using all available document and topic fields is. More research is needed to confirm this however. Our overall recommendation would therefore be to always use all available document fields and topic representations for book retrieval.

7.1 Future Work

Future work would include exploring additional social re-ranking methods. As we are dealing with a book recommendation task, it would be a logical next step to explore techniques from the field of recommender systems, such as collaborative filtering (CF) algorithms. One example could be to use book ratings to calculate the neighborhood of most similar items for each retrieved book and use this to re-rank the result list. The lists of (dis)similar items in the topic representations could also be used for this.

References

1. Kazai, G., Koolen, M., Kamps, J., Doucet, A., Landoni, M.: Overview of the INEX 2011 Book and Social Search Track. In: Geva, S., Kamps, J., Schenkel, R. (eds.) INEX 2011. LNCS, vol. 7424, pp. 1–29. Springer, Heidelberg (2012)
2. Zhai, C., Lafferty, J.: A Study of Smoothing Methods for Language Models Applied to Information Retrieval. ACM Transactions on Information Systems 22(2), 179–214 (2004)
3. Järvelin, K., Kekäläinen, J.: Cumulated Gain-based Evaluation of IR Techniques. ACM Transactions on Information Systems 20(4), 422–446 (2002)
4. Jindal, N., Liu, B.: Review Spam Detection. In: WWW 2007: Proceedings of the 16th International Conference on World Wide Web, pp. 1189–1190. ACM, New York (2007)
5. Fox, E.A., Shaw, J.A.: Combination of Multiple Searches. In: TREC-2 Working Notes, pp. 243–252 (1994)
6. Lee, J.H.: Analyses of Multiple Evidence Combination. SIGIR Forum 31(SI), 267–276 (1997)
7. Renda, M.E., Straccia, U.: Web Metasearch: Rank vs. Score-based Rank Aggregation Methods. In: SAC 2003: Proceedings of the 2003 ACM Symposium on Applied Computing, pp. 841–846. ACM, New York (2003)
8. Bogers, T., Van den Bosch, A.: Fusing Recommendations for Social Bookmarking Websites. International Journal of Electronic Commerce 15(3), 33–75 (2011)
9. Ingwersen, P.: Cognitive Perspectives of Information Retrieval Interaction: Elements of a Cognitive IR Theory. Journal of Documentation 52(1), 3–50 (1996)