Using collaborative filtering in social book search

Huurdeman, H.; Kamps, J.; Koolen, M.; van Wees, J.

Publication date
2012
Document Version
Final published version
Published in
CLEF 2012 : CLEF2012 Working Notes

Using Collaborative Filtering in Social Book Search

Hugo Huurdeman[1], Jaap Kamps[1,2], Marijn Koolen[1], and Justin van Wees[3]

[1] Archives and Information Studies, Faculty of Humanities, University of Amsterdam
[2] ISLA, Faculty of Science, University of Amsterdam
[3] ILPS, Faculty of Science, University of Amsterdam

Abstract. In this paper we describe our participation in INEX 2012 in the Social Book Search Track and the Linked Data Track. For the Social Book Search Track we compare the impact of query- and user-independent popularity measures and recommendations based on user profiles. Book suggestions are more than just topical relevance judgements and may include personal factors such as interestingness, fun and familiarity and book-related aspects such as quality and popularity. Our aim is to understand to what extent book suggestions are related to user-dependent and -independent aspects of relevance. Our findings are that evidence that is both query- and user-independent is not effective for improving a standard retrieval model using blind feedback. User-dependent evidence, on the contrary, is highly effective, leading to significant improvements. For the Linked Data Track we compare different methods of weighted result aggregation using the DBpedia ontology relations as facets and values. Facets and values are aggregated using either document counts or retrieval scores. The reason to use retrieval scores for facet ranking is that we want the top retrieved results to be summarised by the top ranked facets and values. In addition, we look at the impact of taking overlap in aggregation into account. Facet values that give access to many of the same documents have high overlap. Selecting facet values that have low overlap may avoid frustrating the user.

1 Introduction

In this paper we describe our participation in the INEX 2012 Social Book Search Track and the Link Data Track. For the Social Book Search Track we compare the impact of query- and user-independent popularity measures against recommendations based on user profiles. The web and social media have changed the way people search for books. The availability of user-reviews, ratings and tags allows users to find out more about a book than from the traditional descriptions made by professional cataloguers. This in turn may evoke more complex information needs from users, relating to issues such as how interesting, familiar or funny, educational, engaging, well-written or popular a book is. Some of these issues are user-independent, such as the popularity of a book and to some extent its quality—in the sense of the general opinion of a whole group readers—and

can be derived from data such as the number of people who reviewed, rated or tagged a book. Others are more personal, such interestingness and familiarity, and would require individual user information from user profiles or browsing and purchase history. We combine the user-dependent and -independent evidence with query-dependent evidence from a retrieval system to find out whether book suggestion can benefit from user-dependent evidence.

For the Linked Data Track (LDT), we experiment with different ways of aggregating results. A standard approach is to rank facets and values using document counts. The facets and values that summarise the most retrieval results are considered the best summarisations. We compare this approach with aggregation based on retrieval scores, which prefers facet values that summarise the highest ranked documents. Assuming most of the relevant documents will be in the top ranks, result aggregation based on retrieval scores will be focused on the most relevant documents. The document collection of the LDT is rich in structure and offers multiple ways of summarising search results. We use the DBpedia ontology relations as facets and values for summarisation.

We describe our experiments and results for the Social Book Search Track in Section 2 and for the Linked Data Track in Section 3. In Section 4, we discuss our findings and draw conclusions.

2 Social Book Search Track

The effectiveness of user-generated content on social book search may be partly due to its relation to popularity [3]. The amount of user-generated content available for individual books is heavily skewed, with popular books have many more tags, reviews and ratings than more obscure books. Much like the impact of document length on traditional ad hoc search [6], the longer descriptions of popular books have a higher probability of matching query terms and possibly better term distribution statistics as well, with the result that retrieval models favour them over shorter descriptions of less popular books. This prompts the question whether the forum suggestions are merely the most popular among the topically relevant books, or whether personal preferences of the suggestors and topic creators brings in other aspects of relevance as well. If relevance in social book search is merely a combination of topical relevance and popularity, it would seem that book suggestions are mainly user-independent.

We want to compare the effectiveness of popularity priors against recommendations based on user profiles. The goal of our experiments is to investigate whether the impact of user-dependent evidence outweighs the available evidence for popularity, which is both query- and user-independent.

2.1 User-independent Priors

From the book descriptions in the A/LT collection we can derive several indicators of popularity and quality.

We look at the following popularity priors:

- *Length*: document length. Although document length is not directly related to popularity, we assume that descriptions with many tags and reviews are longer than descriptions with no or few tags and reviews.
- *Dirichlet*: without smoothing, language models favour short documents. Dirichlet smoothing introduces an implicit document length bias [7]. As smoothing parameter μ increases, document length becomes less important with respect to term frequency. In other words, documents with high term frequency will be favoured over documents with low term frequency regardless of their document lengths. With equal term frequency, a long document will still score lower than a short document, but the difference is small if μ is higher than the length of either document. The advantage of increasing μ over using a document length prior is that it only prefers longer documents when they have a higher frequency of query terms. With a global document length prior, a very long document with few occurrences of query terms still gets a big boost.
- *NumReviews*: the number of reviews. A large number of reviews means a large number of people know the book and voiced their opinion about it. Note that in constructing the A/LT collection, a maximum of 100 reviews per book were included. Books with at least 100 reviews are all considered equally popular even though the real number of reviews would differentiate between them.
- *SumTag*: the sum of all tag frequencies. The tag frequency of a tag *t*for a book b is the number of users who assigned t to b. We assume that popular books are tag by more users than more obscure books and therefore have a higher total number of tags. Of course, it is possible for a book to receive many tags from a small group of users, but we expect this to be the exception rather than the rule. Only the 50 most frequent tags of a book are included. The tag frequency is unlimited however, and therefore the total number of tags is also not capped.
- *MaxTag*: the frequency of most popular tag. This avoids the problem of conflating cases where many people assign only a few tags each to a book and cases where few people each assign many tags to a book. If the most frequent tag is assigned by n different users, then at least n users know about this book.

Next, we define two quality priors:

- *AvgRating*: average rating. The arithmetic mean over all Amazon ratings for a work.
- *BARating*: The Bayesian average rating. The Bayesian Average (BA) takes into account how many users have rated a work. As more users rates the same work, the average becomes more reliable and less sensitive to outliers. We make the BA dependent on the query, such that the BA of a book is based on books related to the query. The BA of a book b is computed as:

$$BA(b) = \frac{\hat{n} \cdot \hat{m} + \sum_{r \in R(b)} r}{n + \hat{n}} \tag{1}$$

where $R(b)$ is the set of ratings for b $\hat{m}$ is the average unweighted rating over all books in the top 1000 results and $\hat{n}$ is the average number of ratings over all the books in the top 1000.

We crawled a random set of 10,000 books from LibraryThing to obtain popularity information. Each page dedicated to a book contains information on how many members have catalogued it, how popular it is (directly determined by ranking all books by the number of members who catalogued it), how many members have reviewed it and in how many forum discussions it is mentioned (derived from Touchstone mappings).

We use this set to compare the total number of tags and the frequency of the most frequent tag against the number of members who catalogued it. The correlation between these numbers indicates how well our tag-based priors reflect popularity.

2.2 Collaborative Filtering

We want to compare the popularity based measures against methods that take the interests and preferences of the topic creator into account. Specifically, we want to look at collaborative filtering (CF) techniques to exploit the rich data available in the large network of users on LibraryThing. To build a recommender system based on CF, we had to obtain user profiles and personal catalogues of LibraryThing members. We started with a seed list of all the 1,104 users from the topic threads of the 211 topics of the 2011 SB task and crawled their personal catalogues and profiles. Links to other profiles (friends, members with interesting libraries) were extracted to continue the crawl. Because the members who participate in the forums may be different from other members, we also performed crawls based on random sets of 211, 1000 and 10,000 books. In each case, we extracted from each book page on LT the user names who have catalogued that book to generate another seed list. In total, we obtained 89,693 profiles (6% of all profiles) and 5,637,097 book ratings.

We experiment with neighbourhood-based and model-based recommendations and with rated transactions. Rated transactions indicate that a user catalogued a book and how she rated it. The k nearest neighbours ((k-NN) of a user u, denoted $N_i(u)$, are computed using the Pearson correlation of their transaction vectors. The rating r_{ui} of an unseen item i for user u is estimated as:

$$\hat{r}_{ui} = \frac{\sum_{v \in N_i(u)} w_{uv} r_{vi}}{\sum_{v \in N_i(u)} |w_{uv}|} \tag{2}$$

where users v are the nearest neighbours who have rated i. For some books, none of the nearest neighbours gave a rating, and k-NN cannot make a rating prediction. In this case, the average of the ratings of all users in our crawl for this book is used. If there is only one user who has rated the book, no reliable average can be obtained and no prediction is made.

Model-based recommender systems learn a predictive model based on the transactions of a user. The Singular Value Decomposition (SVD) method reduces the domain complexity by reducing the number of dimensions in the item space to a smaller set of underlying dimensions which represent the latent topics and user preferences [4].

2.3 Experimental Setup

We used Indri [8] for indexing, removed stopwords and stemmed terms using the Krovetz stemmer. Based on the results from the 2011 Social Search for Best Books task [1] we focus on the social metadata and indexed only user-generated content—Amazon reviews and LibraryThing tags—and book identification fields: title, author, publisher, publication date, dimensions, weight, and number of pages.

The topics are taken from the LibraryThing discussion groups and contain a *title* field which contains the title of a topic thread, a *group* field which contains the discussion group name and a *narrative* field which contains the first message from the topic thread. In our experiments we only used the *title* fields of the topics as queries, which corresponds to the titles of the topic threads of the LT discussion forums. For the language model our baseline has default settings for Indri (Dirichlet smoothing with $\mu = 2500$). We submitted two runs:

xml_social : a standard LM run on the social metadata index.
xml_social.fb.10.50 : a run on the social metadata index with pseudo relevance feedback using 50 terms from the top 10 results.

For the priors, each of the scores can be turned into a prior probability by dividing it by the sum of scores of all books in the collection. For instance, the document length prior probability is calculated as $P_{Length}(d) = |d|/|D|$, where D is the set of all books in the collection and $|D| = \sum_{d \in D} |d|$. The final document score is then:

$$S_{Length}(d) = P(d|q) \cdot P_{Length}(d) \tag{3}$$

With some priors there are problems with zero scores. A book with no reviews would have a prior probability of zero, which would result in a score $S_{NumReviews} = 0$. To solve this problem, we use the simple smoothing method known as Add-One, which adds one to the number of reviews of each book. The applies to the SumTag, MaxTag, AvgRating and BARating priors. In addition to linear prior probability, we experiment with log priors to compress the score range, thereby reducing the impact of the priors on the ranking. The log SumTag prior is calculated as:

$$P_{Log(SumTag)}(d) = \frac{1 + Log(1 + SumTag(d))}{\sum_{d' \in D} 1 + Log(1 + SumTag(d'))} \tag{4}$$

To rerank the retrieval results with user-dependent evidence from the Collaborative Filtering method, we use a linear combination:

$$S_{CF}(d) = (1 - \lambda) P_{Ret}(d|q) + \lambda P_{CF}(d) \tag{5}$$

Table 1: Evaluation results for the official Social Book Search task runs. Significance levels are 0.05 (°), 0.01 (•) and 0.001 (•).

Run	MRR		nDCG@10		P@10		R@10	
p4.xml_social	0.331		0.130		0.125		0.139	
p4.xml_social.fb.10.50	**0.370**°	**11.8**%	**0.146**	**11.8**%	**0.138**°	**10.3**%	**0.142**	**2.2**%

2.4 Results

The relevance judgements for the SBS task are based on the book suggestions from the LT fourms, and are mapped to three different relevance values: *irrelevant* (rv=0) for suggestions made by the topic creator herself, *relevant* (rv=1) for suggestions by others that the topic creator did not catalogue afterwards, and *highly relevant* (rv=4) for suggestions that the topic creator catalogued after starting the topic. We refer to the latter as post-catalogued suggestions (PCSs).

We first discuss the results of the official submissions (Table 1). Differences between the two runs are tested for statistical significance using a one-tailed Bootstrap test with 100,000 resamples, at significance levels of 0.05 (°), 0.01 (•) and 0.001 (•). The standard run on the xml_social index scores 0.331 on MRR, which means the on average, the first relevant document is found at rank 3. In the 2011 SB task, for which similar topics were used but all suggestions were considered equally relevant, a run on the same index scored 0.2913 on nDCG@10, but with this year's judgements it scores only 0.130. Either the topics this year are harder, or the impact of the difference relevance values is big and the system fails to distinguish between the PCSs and the other suggestions. If we map the PCSs to relevance value $rv = 1$, the nDCG@10 score goes up from 0.130 to 0.171, and if we map all suggestions to $rv = 1$ (similar to operationalisation used for last year's task), it goes up to 0.224. This means that this year's topics are more difficult, but also that the distinction between PCSs and other suggestions has made the task more difficult.

The feedback run improves upon the standard run for all four measures, with significant improvements for MRR and P@10. Adding terms from the top 10 documents leads to a better description of the information need. However, the improvement in nDCG@10, which emphasises the suggestions that the topic creator selects to add to her catalogue, is not significant. For our experiments with popularity and quality priors and recommendations we use the feedback run *p4.xml_social.fb.10.50* as the baseline, which is the highest scoring run of all official submissions on nDCG@10.

The results are shown in Table 2. We start with the quality priors. The ratings have little impact on performance. All variants are able to improve MRR, but on the other measures the improvements are smaller and not significant. The only exception is the plain Bayesian average prior, which is more effective than the others. This suggests that ratings are mainly useful for improving very early precision. The improvement of the BA Rating prior on nDCG@10 suggests that topic creators take ratings into account when selecting books. However, most improvements are not significant. Perhaps ratings do not reflect quality well, or

Table 2: Evaluation results for the Social Book Search task runs. Significance levels are 0.05 ($^\circ$), 0.01 ($^\bullet$) and 0.001 ($^\bullet$).

Run	MRR		nDCG@10		P@10		R@10	
Baseline	0.362		0.144		0.122		0.149	
Quality priors								
AvgRating	0.377	4.3%	0.143	-0.6%	0.122	0.0%	0.153	2.6%
Log(AvgRating)	0.373	3.2%	0.143	-0.2%	0.125	2.5%	0.152	1.9%
BA Rating	0.379	4.8%	0.151	5.1%	0.126	3.4%	0.158	5.6%
Log(BA Rating)	0.374	3.6%	0.142	-1.3%	0.124	1.7%	0.151	0.9%
Popularity priors								
MaxTag	0.290$^\bullet$	-19.7%	0.082$^\bullet$	-43.3%	0.085$^\bullet$	-29.9%	0.093$^\bullet$	-38.0%
Log(MaxTag)	0.357	-1.2%	0.137	-4.4%	0.116	-5.2%	0.143	-4.4%
SumTags	0.274$^\bullet$	-24.3%	0.080$^\bullet$	-44.1%	0.087$^\bullet$	-28.2%	0.089$^\bullet$	-40.6%
Log(SumTags)	0.371	2.6%	0.145	0.7%	0.119	-2.6%	0.149	-0.1%
NumReviews	0.370	2.4%	0.129	-10.5%	0.110	-9.4%	0.129	-13.9%
Log(NumReviews)	0.403°	11.4%	0.161	12.1%	0.130	6.8%	0.158	5.9%
Length priors								
Length	0.357	-1.2%	0.126	-12.4%	0.112	-8.5%	0.137	-8.4%
Log(Length)	0.379	4.8%	0.149	4.0%	0.121	-0.9%	0.149	-0.3%
Dirichlet $\mu = 5000$	0.357	-1.2%	0.150	4.2%	0.128	5.1%	0.160	7.2%
Dirichlet $\mu = 10000$	0.371	2.7%	0.153	6.7%	0.128	5.1%	0.160	7.2%
Dirichlet $\mu = 15000$	0.355	-1.9%	0.151	5.4%	0.124	1.7%	0.150	0.4%
Recommendation								
k-NN (N=50, λ=0.0001855)	**0.411**$^\bullet$	**13.6**%	**0.181**$^\bullet$	**26.0**%	**0.154**$^\bullet$	**26.5**%	**0.199**$^\bullet$	**32.9**%
SVD (K=100, λ=0.000185)	0.403$^\bullet$	11.3%	0.172$^\bullet$	19.8%	0.149$^\bullet$	22.2%	0.187$^\bullet$	24.9%

quality is not effective as user-independent evidence. In the latter case, it might mean that quality is perceived differently by different users.

Next we discuss the popularity priors. The tag-based priors lead to significant drops in performance when used directly. Curbing their impact by taking the log of the MaxTag or SumTag scores is still not effective. Only the Log(SumTag) prior leads to small but insignificant improvements on MRR and nDCG@10. The number of reviews is more effective. The plain NumReviews prior only improves MRR but hurts performance on the other measures. The compressed score range of the Log(NumReviews) prior is more effective. Performance on all measures improves, with more than 11% improvements for MRR and nDCG@10. The larger improvement for nDCG@10 than for P@10 indicates the reviews are particularly useful for promoting suggestions that the topic creator decides to catalogue. Only the improvement on MRR is significant. This can mean that the number of reviews is a better indicator of popularity than SumTags and MaxTag, or that the topic creator tends to select books for which multiple reviews are available.

The Length prior is only effective when logged, and only improves performance on MRR and nDCG@10, but not significantly. The implicit length prior of the Dirichlet smoothing parameter μ is more stable, and improves performance on all measures for $\mu = 10,000$. With higher values for μ, performance starts to drop. Completely ignoring document length and only considering term frequency and document frequency is not good for performance. Even though promoting

longer document is effective, it is still important to connect term frequency to the amount of text in a document.

Although some popularity and quality ratings can improve performance, any improvements on the official measure nDCG@10 are not significant. Evidence that is both user- and query-dependent seems not effective for social book search.

Finally, we turn to the impact of combining retrieval with recommendation. For the k-NN method we experimented with different neighbourhood sizes (25, 50, and 100 neighbours) and λ values. Typically, the best performance with k-NN is achieved with $20 \leq k \leq 50$ ([2]). We show only the best performing combination, where $k = 50$ and $\lambda = 0.0001855$. For the SVD method, best performance was achieved with 100 dimensions (K=100) and $\lambda = 0.000185$). The recommendations from both SVD and k-NN lead to significant improvements on all measures. User-dependent evidence is highly effective for social book search. The k-NN method performs better than the more complex SVD method.

In sum, evidence based on personal preferences of the user seems much more effective than user-independent evidence based on popularity and quality. The low impact of the quality priors might indicate that quality in book search is more user-dependent. The effectiveness of the number of reviews may be an indicator that popularity can be effective, but also that forum members looking for books only catalogue books for which reviews are available. This is in line with our previous findings that workers on Mechanical Turk, when judging the relevance of books for the same LT forum topics, find it hard to judge books for which no reviews are available Koolen et al. [3]. With the presence of user reviews, the nature of relevance judgements has become more complex and goes beyond mere topical relevance.

3 Linked Data Track

For the Faceted Search Task of the Linked Data Track, systems are required to create a list of both facets and facet values for the explorative search queries contained in the topics of this task. The derived facets should describe relevant information for each of the queries featured in the task, preferably resulting in compact summaries of the available data. Our aim is to experiment with different ways of aggregating results, using either document counts or retrieval scores, and either ignoring or penalising document overlap in the ranking of facet values. The idea behind using retrieval scores for aggregation is that we want to focus on the top ranked results, as the retrieval model ranks documents by relevance, with the most relevant documents in the top of the list. Facet values that summarise many of the top documents give the user easy access to the most relevant documents.

Of course, the point of aggregation is to summarise long lists of results effectively and efficiently, so focussing on facet values that summarise only the top few documents defies the purpose of result aggregation. Good facet value selection requires a careful balance between high coverage and giving access to the most relevant documents.

Experimental Setup

We use Indri [8] with Krovetz stemming and default smoothing (Dirichlet with $\mu = 2500$) for indexing. Up to 2000 documents were retrieved using title fields only. We submitted one run for the Ad Hoc Search Task. For the Faceted Search task we were not able to finish any runs in time for the submission deadline.

The Ad Hoc run is used as the basis for carrying out the Faceted Search Task. We explored possibilities to extract different facets and facet values from the data available in the Wikipedia-LOD collection of the Linked Data Track. The candidate facets consist of the DBpedia relations and properties for each Wikipedia article included in the collection. For our exploration, we are also using additional ontological data available from DBpedia itself.

Facet selection

As a basic approach to performing the selection of facets, we used the concept of '*facet coverage*' [1]. This refers to the number of documents that are summarized by a facets top n values. The aim is to provide compact summaries of the available data using the selected facets, so these facets ideally should cover a high number of documents.

Using the ontology relations of DBpedia, we generated a list of all possible facets for a topic from the available DBpedia properties contained in each Wikipedia-LOD article in the collection. The list of facets includes the top 5 values for each facet, based on the number of documents a value covers, and the top 5 values based on their retrieval scores (originating from the baseline run created using Indri). To select a number of top facets out of the list of all facets for a given query, we are using different methods. One way to select the facets is based on the facet coverage. A disadvantage of this method, however, is that this does not take the overlap between facets into account. Therefore, a second method has been used, *coverageNO*, that focuses on the number of unique documents summarized by the facets top n values (see also [1]).

Based on a recursive selection method, it is possible to create a hierarchical list of facets and facet values. There are some issues with the available data from DBpedia, which influenced the facet selections that we explored in our research. First of all, there is a wide range of properties that are used for DBpedia entities, but not all of them are applied consistently. Furthermore, a substantial number of the top-ranked results from our baseline run do not have DBpedia properties, except for links to other pages, and therefore are not included in the generated facets. Finally, some of the entities have incorrect properties, possibly due to the semi-automatically generated structure of DBpedia, that is based on the user-authored data of Wikipedia. To overcome these limitations, we are also exploring ways to include additional data from DBpedia in the process of selecting facets, for example the ontological structure of DBpedia.[1]

[1] URL: http://mappings.dbpedia.org/server/ontology/classes/

4 Conclusion

In this paper we discussed our participation in the INEX 2012 Social Book Search Track and the Linked Data Track.

For the Social Book Search Track, we experimented with user-dependent and user-independent evidence in the form of document priors—length, book ratings, and numbers of tags and reviews—and user-dependent evidence in the form of recommendations from collaborative filtering approaches. We crawled a large set of user profiles and personal catalogues of LibraryThing members and experimented with neighbourhood-based and model-based recommender systems.

We found that document priors reflecting quality and popularity do not improve performance of a standard language model with blind feedback. The number of reviews of a book is the most effective prior, but does not lead to significant improvements. It is not clear whether the number of reviews is effective because it reflects popularity or because it promotes books for which the searcher can read multiple reviews and therefore make a more informed selection. Our findings suggest that evidence that is both query- and user-independent is not effective for social book search.

In contrast, user-dependent information from recommender systems is highly effective. Both k-nearest neighbour and SVD approaches lead to significant improvements. Although the k-NN method is less complex than SVD, it is the more effective of the two. Our findings suggest that user-dependent evidence is more important than user-independent information.

For the Linked Data Track, our aims are to compare the effectiveness of different result aggregation approach and of ignoring or penalising overlap in the results summarised by the chosen values of a selected facet. We are still implementing this model and the relevance judgements are not yet available, so we have no evaluation results yet.

Acknowledgments This research was supported by the Netherlands Organization for Scientific Research (NWO projects # 612.066.513, 639.072.601, and 640.005.001) and by the European Communitys Seventh Framework Program (FP7 2007/2013, Grant Agreement 270404).

Bibliography

[1] F. Andriaans, M. Koolen, and J. Kamps. The importance of document ranking and user-generated content for faceted search and book suggestions. In S. Geva, J. Kamps, and R. Schenkel, editors, *Focused Retrieval of Content and Structure: 10th International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX 2011)*, volume 7424 of *LNCS*. Springer, 2012.

[2] C. Desrosiers and G. Karypis. A comprehensive survey of neighborhood-based recommendation methods. In Ricci et al. [5], pages 107–144. ISBN 978-0-387-85819-7.

[3] M. Koolen, J. Kamps, and G. Kazai. Social Book Search: The Impact of Professional and User-Generated Content on Book Suggestions. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM 2012)*. ACM, 2012.

[4] Y. Koren and R. M. Bell. Advances in collaborative filtering. In Ricci et al. [5], pages 145–186. ISBN 978-0-387-85819-7.

[5] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors. *Recommender Systems Handbook*. Springer, 2011. ISBN 978-0-387-85819-7.

[6] A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In SIGIR *'96: Proceedings of the 19th annual international ACM* SIGIR *conference on Research and development in information retrieval*, pages 21–29, New York, NY, USA, 1996. ACM. ISBN 0-89791-792-8. doi: http://doi.acm.org/10.1145/243199.243206.

[7] M. D. Smucker and J. Allan. An investigation of dirichlet prior smoothings performance advantage. Technical report, 2005.

[8] T. Strohman, D. Metzler, H. Turtle, and W. B. Croft. Indri: a language-model based search engine for complex queries. In *Proceedings of the International Conference on Intelligent Analysis*, 2005.