University of Amsterdam at INEX 2011: Book and Data Centric Tracks

Frans Adriaans[1,2] Jaap Kamps[1,3] and Marijn Koolen[1]

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

Abstract. In this paper we describe our participation in INEX 2011 in the Book Track and the Data Centric Track. For the Book Track we focus on the impact of different document representations of book metadata for book search, using either professional metadata, user-generated content or both. We evaluate the retrieval results against ground truths derived from the recommendations in the LibraryThing discussion groups and from relevance judgements obtained from Amazon Mechanical Turk. Our findings show that standard retrieval models perform better on user-generated metadata than on professional metadata. For the Data Centric Track we focus on the selection of a restricted set of facets and facet values that would optimally guide the user toward relevant information in the Internet Movie Database (IMDb). We explore different methods for effective result summarization by means of weighted aggregation. These weighted aggregations are used to achieve maximal coverage of search results, while at the same time penalizing overlap between sets of documents that are summarized by different facet values. We expect that weighted result aggregation combined with redundancy avoidance results in a compact summary of available relevant information.

1 Introduction

Our aims for the Book Track were to look at the relative value of user tags and reviews and traditional book metadata for ranking book search results. The Social Search for Best Books task is newly introduced this year and uses a large catalogue of book descriptions from Amazon and LibraryThing. The descriptions are a mix of traditional metadata provided by professional cataloguers and indexers and user-generated content in the form of ratings, reviews and tags.

Because both the task and collection are new, we keep our approach simple and mainly focus on a comparison of different document representations. We made separate indexes for representations containing a) only title information, b) all the professional metadata, c) the user-generated metadata, d) the metadata from Amazon, e) the data from LibraryThing and f) all metadata. With these indexes we compare standard language model retrieval systems and evaluate them using the relevance judgements from the LibraryThing discussion forums and from Amazon Mechanical Turk. We break down the results to look at performance on different topic types and genres to find out which metadata is effective for particular categories of topics.

For the Data Centric Track we focus on the selection of a restricted set of facets and facet values that would optimally guide the user toward relevant information. We aim to improve faceted search by addressing two issues: *weighted result aggregation* and *redundancy avoidance*.

The traditional approach to faceted search is to simply count the number of documents that is associated with each facet value. Those facet values that have the highest number of counts are returned to the user. In addition to implementing this simple approach, we explore the aggregation of results using weighted document counts. The underlying intuition is that facet values with the *most* documents are not necessarily the *most relevant* values [1]. That is, buying a dvd by the director who directed the *most* movies does not necessarily meet the search demands of a user. It may be more suitable to return directors who made a large number of *important* (and/or popular) movies. More sophisticated result aggregations, acknowledging the importance of an entity, may thus provide better hints for further faceted navigation than simple document counts. We therefore explore different methods for effective result summarization by means of weighted aggregation.

Another problem in faceted search concerns the avoidance of overlapping facets [2]. That is, facets whose values describe highly similar set of documents should be avoided. We therefore aim at penalizing overlap between sets of documents that are summarized by different facet values. We expect that weighted result aggregation combined with redundancy avoidance results in a compact summary of the available relevant information.

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

2 Book Track

In the INEX 2011 Book Track we participated in the Social Search for Best Books task. Our aim was to investigate the relative importance of professional and user-generated metadata. The document collection consists of 2.8 million book description, with each description combining information from Amazon and LibraryThing. The Amazon data has both traditional book metadata such as title information, subject headings and classification numbers, and user-generated metadata as well as user ratings and reviews. The data from LibraryThing consists mainly of user tags.

Professional cataloguers and indexers aim to keep metadata mostly objective. Although subject analysis to determine headings and classification codes is somewhat subjective, the process follows a formal procedure and makes use of controlled vocabularies. Readers looking for interesting or fun books to read may not only want objective metadata to determine what book to read or buy next, but also opinionated information such as reviews and ratings. Moreover, subject headings and classification codes might give a very limited view of what a book is about. LibraryThing users tag books with whatever keywords they want, including personal tags like *unread* or *living room bookcase*, but also highly specific, descriptive tags such *WWII pacific theatre* or *natives in Oklahoma*.

We want to investigate to what extent professional and user-generated metadata provide effective indexing terms for book retrieval.

2.1 Experimental Setup

We used Indri [3] for indexing, removed stopwords and stemmed terms using the Krovetz stemmer. We made 5 separate indexes:

Full : the whole description is indexed.
Amazon : only the elements derived from the Amazon data are indexed.
LT : only the elements derived from the LibraryThing data are indexed.
Title : only the title information fields (title, author, publisher, publication date, dimensions, weight, number of pages) are indexed.
Official : only the traditional metadata fields from Amazon are indexed, including the title information (see Title index) and classification and subject heading information.
Social : only the user-generated content such as reviews, tags and ratings are indexed.

In the *Full*, *LT* and *Social* indexes, the field information from `<tag>` elements is also indexed in a separate column to be able to give more weight to terms occurring in `<tag>` elements.

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 as queries and default settings for Indri (Dirichlet smoothing with $\mu = 2500$). We submitted the following six runs:

xml_amazon : a standard LM run on the Amazon index.
xml_full : a standard LM run on the Full index.
xml_full.fb.10.50 : a run on the Full index with pseudo relevance feedback using 50 terms from the top 10 results.
xml_lt : a standard LM run on the LT index.
xml_social : a standard LM run on the Social index.
xml_social.fb.10.50 : a run on the Social index with pseudo relevance feedback using 50 terms from the top 10 results.

Additionally we created the folowing runs:

xml_amazon.fb.10.50 : a standard LM run on the Amazon index.
xml_lt.fb.10.50 : a standard LM run on the LT index.
xml_official : a standard LM run on the Official index.
xml_title : a standard LM run on the Title index.

Table 1: Evaluation results for the Book Track runs using the LT recommendation Qrels. Runs marked with * are official submissions.

Run	nDCG@10	P@10	MRR	MAP
xml_amazon.fb.10.50	0.2665	0.1730	0.4171	0.1901
*xml_amazon	0.2411	0.1536	0.3939	0.1722
*xml_full.fb.10.50	0.2853	0.1858	0.4453	0.2051
*xml_full	0.2523	0.1649	0.4062	0.1825
xml_lt.fb.10.50	0.1837	0.1237	0.2940	0.1391
*xml_lt	0.1592	0.1052	0.2695	0.1199
xml_prof	0.0720	0.0502	0.1301	0.0567
*xml_social.fb.10.50	**0.3101**	**0.2071**	**0.4811**	**0.2283**
*xml_social	0.2913	0.1910	0.4661	0.2115
xml_title	0.0617	0.0403	0.1146	0.0563

Table 2: Evaluation results for the Book Track runs using the AMT Qrels. Runs marked with * are official submissions.

Run	nDCG@10	P@10	MRR	MAP
xml_amazon.fb.10.50	0.5954	0.5583	0.7868	0.3600
*xml_amazon	**0.6055**	**0.5792**	0.7940	0.3500
*xml_full.fb.10.50	0.5929	0.5500	**0.8075**	**0.3898**
*xml_full	0.6011	0.5708	0.7798	0.3818
xml_lt.fb.10.50	0.4281	0.3792	0.7157	0.2368
*xml_lt	0.3949	0.3583	0.6495	0.2199
xml_prof	0.1625	0.1375	0.3668	0.0923
*xml_social.fb.10.50	0.5425	0.5042	0.7210	0.3261
*xml_social	0.5464	0.5167	0.7031	0.3486
xml_title	0.2003	0.1875	0.3902	0.1070

2.2 Results

The Social Search for Best Books task has two sets of relevance judgements. One based on the lists of books that were recommended on the LT discussion groups, and one based on document pools of the top 10 results of all official runs, judged by Mechanical Turk workers. For the latter set of judgements, a subset of 24 topics was selected from the larger set of 211 topics from the LT forums.

We first look at the evaluation results based on the Qrels derived from the LT discussion groups in Table 1. The runs on the Social index outperform the other runs on all measures. The indexes with no user-generated metadata–Official and Title—lead to low scoring runs. Feedback is effective on the four indexes Amazon, Full, LT and Social.

Next we look at the results based on the Mechanical Turk judgements over the subset of 24 topics in Table 2. Here we see a different pattern. With the top 10 results judged on relevance, all scores are higher than with the LT judgements. This is probably due in part to the larger number of judged documents, but perhaps also to the difference in the tasks. The Mechanical Turk workers were

Table 3: Evaluation results for the Book Track runs using the LT recommendation Qrels for the 24 topics selected for the AMT experiment. Runs marked with * are official submissions.

Run	nDCG@10	P@10	MRR	MAP
xml_amazon.fb.10.50	0.2103	0.1625	0.3791	0.1445
xml_amazon	0.1941	0.1583	0.3583	0.1310
xml_full.fb.10.50	0.2155	0.1708	0.3962	0.1471
xml_full	0.1998	0.1625	0.3550	0.1258
xml_lt.fb.10.50	0.1190	0.0833	0.3119	0.0783
xml_lt	0.1149	0.0708	0.3046	0.0694
xml_prof	0.0649	0.0500	0.1408	0.0373
xml_social.fb.10.50	**0.3112**	**0.2333**	**0.5396**	**0.1998**
xml_social	0.2875	0.2083	0.5010	0.1824
xml_title	0.0264	0.0167	0.0632	0.0321

asked to judged the topical relevance of books—is the book on the same topic as the request from the LT forum—whereas the LT forum members were asked by the requester to recommend books from a possibly long list of topically relevant books. Another interesting observation is that feedback is not effective for the AMT evaluation, whereas it was effective for the LT evaluation.

Perhaps another reason is that the two evaluations use different topic sets. To investigate the impact of the topic set, we filtered the LT judgements on the 24 topics selected for AMT, such that the LT and AMT judgements are more directly comparable. The results are shown in Table 3. The pattern is similar to that of the LT judgements over the 211 topics, indicating that the impact of the topic set is small. The runs on the Social index outperform the others, with the Amazon and Full runs scoring better than the LT runs, which in turn perform better than the Official and Title runs. Feedback is again effective for all reported measures. In other words, the observed difference between the LT and AMT evaluations is not caused by difference in topics but probably caused by the difference in the tasks.

2.3 Analysis

The topics of the SB Track are labelled with topic type and genre. There are 8 different type labels: *subject* (134 topics), *author* (32), *genre* (17), *series* (10), *known-item* (7), *edition* (7), *work* (3) and *language* (2).

We break down the evaluation results over topic types and take a closer look at the *subject*, *author* and *genre* types.

The other types have either very small numbers of topics (*work* and *language*), or are hard to evaluate with the current relevance judgements. For instance, the *edition* topics ask for a recommended edition of a particular work. In the relevance judgements the multiple editions of a work are all mapped to a single work ID in LibraryThing. Some books have many more editions than others, which would create in imbalance in the relevance judgements for most topics.

Table 4: Evaluation results using the LT recommendation Qrels across different topic genres and types. Runs marked with * are official submissions.

	nDCG@10				
Run	**Fiction**	**Non-fiction**	**Subject**	**Author**	**Genre**
xml_amazon.fb.10.50	0.2739	0.2608	0.2203	0.4193	0.0888
*xml_amazon	0.2444	0.2386	0.1988	0.3630	0.0679
*xml_full.fb.10.50	0.2978	0.2765	0.2374	0.4215	0.1163
*xml_full	0.2565	0.2491	0.2093	0.3700	0.0795
xml_lt.fb.10.50	0.1901	0.1888	0.1597	0.2439	0.0850
*xml_lt	0.1535	0.1708	0.1411	0.2093	0.0762
xml_prof	0.0858	0.0597	0.0426	0.1634	0.0225
*xml_social.fb.10.50	**0.3469**	**0.2896**	**0.2644**	**0.4645**	0.1466
*xml_social	0.3157	0.2783	0.2575	0.4006	**0.1556**
xml_title	0.0552	0.0631	0.0375	0.1009	0.0000

The genre labels can be grouped into fiction, with genre label *Literature* (89 topics) and non-fiction, with genre labels such as *history* (60 topics), *biography* (24), *military* (16), *religion* (16), *technology* (14) and *science* (11).

The evaluation results are shown in Table 4. For most runs there is no big difference in performance between *fiction* and *non-fiction* topics, with slightly better performance on the *fiction* topics. For the two runs on the Social index the difference is bigger. Perhaps this is due to a larger amount of social metadata for fiction books. The standard run on the LT index (xml_lt) performs better on the non-fiction topics, suggesting the tags for non-fiction are more useful than for fiction books.

Among the topic types we see the same pattern across all measures and all runs. The *author* topic are easier than the *subject* topics, which are again easier than the *genre* topics. We think this is a direct reflection of the clarity and specificity of the information needs and queries. For author related topics, the name of the author is a very clear and specific retrieval cue. Subject are somewhat broader and less clearly defined, making it harder to retrieve exactly the right set of books. For genre-related topics it is even more difficult. Genres are broad and even less clearly defined. For many genres there are literally (tens of) thousands of books and library catalogues rarely go so far in classifying and indexing specific genres. This is also reflected by the very low scores of the Official and Title index runs for *genre* topics.

3 Data Centric Track

For the Data Centric Track we participated in the Ad Hoc Task and the Faceted Search Task. Our particular focus was on the Faceted Seach Task where we aim to discover for each query a restricted set of facets and facet values that best describe relevant information in the results list. Our general approach is to use weighted result aggregations to achieve maximal coverage of relevant documents in IMDb. At the same time we aim to penalize overlap between sets of documents that are summarized by different facet values. We expect that weighted result aggregation combined with redundancy avoidance results in a compact summary of the available relevant information. Below we describe our setup and provide details about the different runs that we submitted to INEX. At the time of writing no results are available for the Faceted Search Task. We are therefore unable to provide an analysis of the performance of our different Faceted Search runs at this point in time.

3.1 Experimental Setup

In all ad hoc runs (Ad Hoc and Faceted Search) we use Indri [3] with Krovetz stemming and default smoothing (Dirichlet with $\mu = 2500$) to create an index. All XML leaf elements in the IMDb collection are indexed as fields. The XML document structure was not used for indexing. Documents were retrieved using title fields only. The maximum number of retrieved documents was set to 1000 (Ad Hoc Task) and 2000 (Faceted Search Task). We submitted one run for the Ad Hoc Search Task and three runs for the Faceted Search Task.

Ad Hoc Task Since the Ad Hoc Search Task was not the focus of our participation, only one run (*UAms2011adhoc*) was generated for the Ad Hoc topic set, using the settings described above.

Faceted Search Task Three runs were generated for the Faceted Search Task (*UAms2011indri-c-cnt*, *UAms2011indri-cNO-scr2*, *UAms2011lucene-cNO-lth*). In each run, a hierarchy of recommended facet values is constructed for each topic. Every path through the hierarchy represents an accumulated set of conditions on the retrieved documents. The search results become more refined at every step, and the refinement ultimately narrows down a set of potentially interesting documents. Below we describe our approach to faceted search in more detail.

3.2 Step 1: Ad hoc run on IMDb collection

Two ad hoc result files were used: the *2011-dc-lucene.trec* file provided by the INEX organization, and an ad hoc run that was created on the fly using Indri. The maximum number of results was set to 2000.

3.3 Step 2: Facet selection

The candidate set consists of all numerical and categorical fields in the IMDb collection. (Free-text fields were not allowed as candidate facets by the organization.) The goal was to select useful facets (and values) from the set of candidate facets.

Result aggregation We explored two different methods of weighted result aggregation. The first method is aggregation using document lengths rather than document counts. Since popular movies in IMDb have larger entries (which we measure by file size), we reasoned that summing over document lengths may help in getting popular facet values (associated with popular movies) at the top of the ranked set of facet values. The second method is aggregation using retrieval scores. That is, we sum the retrieval scores of each document taken from the ad hoc run. The idea is that higher-ranked documents in the results file display those facet values that are most likely to be of interest to the user, given the user's query. The difference between the two weighted aggregation methods is that document length is a static ('global') measure of document importance, whereas retrieval scores are dynamic ('local'), resulting in different degrees of importance for different topics. We compare both methods to traditional non-weighted aggregation of search results. The result aggregations form the basis of facet selection, which is described below.

Coverage The idea behind our approach to facet selection is the simple intuition that facets which provide compact summaries of the available data would allow for fast navigation through the collection. This intuition was implemented as *facet coverage*: the number of documents that are summarized by a facet's top n values[1]. Two types of coverage were implemented. The first version, *coverage*, simply sums up the (weighted) document counts that are associated with the facet's top n values. A potential pitfall of this approach, however, is that this method favors redundancy. That is, the sets of documents that are associated with different facet values may have a high degree of overlap. For example, the keywords 'murder' and 'homicide' may point to almost identical sets of documents. Since we want to give the user compact overviews of different, non-overlapping sets of documents that may be of interest to the searcher, we implemented a second version: *coverageNO* ('coverage, no overlap'). Rather than summing up document counts, *coverageNO* counts the number of unique documents that are summarized by the facet's top n values. This way redundancy in facet values is penalized.

Coverage-based facet selection is applied recursively. Starting with the complete set of ad hoc results (corresponding to the root node of the facet hierarchy), the facet with the highest coverage is chosen. The set of results is then narrowed down to the set of documents that are covered by this facet. With this new set, a second facet is chosen with the highest coverage within the new set. This selection process continues until a specified number of facets has been selected.[2] We apply facet selection to movie candidate facets and person candidate facets independently, since these facets describe different types of documents (i.e., you cannot drill-down into person files after you've narrowed down the results using a movie facet). An example of a ranked set of movie facets for the query 'Vietnam' is given in Table 5.

[1] In our runs, we explored $n = 5$ and $n = 10$.

[2] We set the maximum number of selected facets to 5.

Table 5: Facets ranked by coverage (based on document counts).

Rank	**Coverage**	**Facet**	**Top-5 values**
1	945	genre	Drama (306) Documentary (207) War (199) Action (157) Comedy (76)
2	850	keyword	vietnam (286) vietnam-war (220) independent-film (162) vietnam-veteran (110) 1960s (72)
3	477	language	English (400) Vietnamese (42) French (16) Spanish (10) German (9)
4	437	country	USA (345) UK (30) Canada (27) France (19) Vietnam (16)
5	397	color	Color (291) Color - (Technicolor) (45) Black and White (40) Color - (Eastmancolor) (11) Color - (Metrocolor) (10)

3.4 Step 3: Path construction

The selected set of facets with corresponding top n ranked values form the basis of the facet hierarchy. Paths in the hierarchy are generated by selecting a value for the first facet, then a value for the second facet, etc. The paths respect the rankings of the values along the path. That is, paths through high-ranked facet values are listed at the top of the hierarchy, followed by paths through lower-ranked facet values. In order to restrict the number of paths in the hierarchy (not all logically possible paths are considered relevant) we return only paths that we think are useful recommendations for the user, using a formal criterium. In our current implementation, only paths are included which lead to a set of documents of a specified size.[3] Paths that lead to fewer documents (e.g., < 10 documents) are ignored because they are too specific. Conversely, paths that lead to a larger number of documents (e.g., > 20 documents) are considered too general, and the system will attempt to branch into a deeper, more specific level.

We generate trees for ‘movies’ and ‘persons’ independently and join them in the order of the highest number of paths. (For most queries there were more movie paths than person paths.) An example of our approach to constructing paths through facet values is shown below. We display the tree corresponding to the ‘Vietnam’ query, using the facets from Table 5:

```
<topic tid=”2011205”>
<fv f=”/movie/overview/genres/genre” v=”Drama”>
  <fv f=”/movie/overview/keywords/keyword” v=”vietnam”>
    <fv f=”/movie/additional_details/languages/language” v=”Vietnamese”>
      <fv f=”/movie/additional_details/countries/country” v=”USA”>
        <fv f=”/movie/additional_details/colors/color” v=”Color”/>
          </fv>
        </fv>
      </fv>
  <fv f=”/movie/overview/keywords/keyword” v=”vietnam-war”>
    <fv f=”/movie/additional_details/languages/language” v=”English”>
      <fv f=”/movie/additional_details/countries/country” v=”USA”>
        <fv f=”/movie/additional_details/colors/color” v=”Color - (Technicolor)”/>
          </fv>
        </fv>
    <fv f=”/movie/additional_details/languages/language” v=”Vietnamese”>
      <fv f=”/movie/additional_details/countries/country” v=”USA”/>
        </fv>
      </fv>
  <fv f=”/movie/overview/keywords/keyword” v=”vietnam-veteran”>
    <fv f=”/movie/additional_details/languages/language” v=”English”>
      <fv f=”/movie/additional_details/countries/country” v=”USA”>
        <fv f=”/movie/additional_details/colors/color” v=”Color - (Technicolor)”/>
```

[3] As an inclusion criterium, we keep all paths that lead to a set of 10-20 documents.

```
          </fv>
        </fv>
      </fv>
    </fv>
<fv f=”/movie/overview/genres/genre” v=”Documentary”>
  <fv f=”/movie/overview/keywords/keyword” v=”vietnam”>
    <fv f=”/movie/additional_details/languages/language” v=”English”>
      <fv f=”/movie/additional_details/countries/country” v=”USA”>
        <fv f=”/movie/additional_details/colors/color” v=”Black and White”/>
          </fv>
        </fv>
      </fv>
  <fv f=”/movie/overview/keywords/keyword” v=”vietnam-war”>
    <fv f=”/movie/additional_details/languages/language” v=”English”>
      <fv f=”/movie/additional_details/countries/country” v=”USA”>
        <fv f=”/movie/additional_details/colors/color” v=”Black and White”/>
          </fv>
        </fv>
      </fv>
  <fv f=”/movie/overview/keywords/keyword” v=”vietnam-veteran”>
    <fv f=”/movie/additional_details/languages/language” v=”English”>
      <fv f=”/movie/additional_details/countries/country” v=”USA”>
        <fv f=”/movie/additional_details/colors/color” v=”Color”/>
          </fv>
        </fv>
      </fv>
  <fv f=”/movie/overview/keywords/keyword” v=”1960s”>
    <fv f=”/movie/additional_details/languages/language” v=”English”>
      <fv f=”/movie/additional_details/countries/country” v=”USA”>
        <fv f=”/movie/additional_details/colors/color” v=”Color”/>
          </fv>
        </fv>
...
```

3.5 The Faceted Seach runs

With the methodology described above, a total of 32 runs was generated by varying the parameters listed in Table 6. From this set the following three runs were selected for submission:

UAms2011indri-c-cnt This is our baseline run which implements the standard approach of selecting those facet values that summarize the largest number of documents.

UAms2011indri-cNO-scr2 : This run uses weighted result aggregation (using retrieval scores, in contrast to the unranked aggregation in the baseline run). In addition, this run penalizes overlap between document sets that correspond to different facet values.

Table 6: Experimental parameters (which resulted in 2x4x2x1x2x1x1 = 32 runs)

Parameter	**Values**
Ad hoc input	*lucene, indri*
Document weights	*count, length, score, score*^{2}
Selection method	*coverage, coverageNO*
Number of facets	5
Number of values	5, 10
Min. number of path results	10
Max. number of path results	20

UAms2011lucene-cNO-lth : The third run uses the Lucene run that was provided by the INEX organizers. The run uses weighted result aggregation based on document lengths (file sizes, as opposed to retrieval scores).

3.6 Results and discussion

Our run for the Ad Hoc Task ranked 1st (based on MAP scores; MAP = 0.3969). The success of our Ad Hoc run indicates that indexing the complete XML structure of IMDb is not necessary for effective document retrieval. It appears, at least for the Ad Hoc case, that it suffices to index leaf elements. Results of the Faceted Search Task are unknown at this time.

4 Conclusion

In this paper we discussed our participation in the INEX 2011 Book and Data Centric Tracks.

In the Book Track we participated in the Social Search for Best Books task and focused on comparing different document representations based on professional metadata and user-generated metadata. Our main finding is that standard language models perform better on representations of user-generated metadata than on representations of professional metadata.

In our result analysis we differentiated between topics requesting fiction and non-fiction books and between subject-related topics, author-related topics and genre-related topics. Although the patterns are similar across topic types and genres, we found that social metadata is more effective for fiction topics than for non-fiction topics, and that regardless of document representation, all systems perform better on author-related topics than on subject related topics and worst on genre-related topics. We expect this is related to the specificity and clarity of these topic types. Author-related topics are highly specific and target a clearly defined set of books. Subject-related topics are broader and less clearly defined, but can still be specific. Genre-related topics are very broad—many genres have tens of thousands of books—and are also more vague information needs that are closer to exploratory search.

In future work we will look closer at the relative value of various types of metadata and directly compare individual types of metadata such as reviews, tags and subject headings. We will also look at the different search scenarios underlying the relevance judgements and topic categories, such as subject search, recommendation and exploratory search.

In the Data Centric Track we participated in the Ad Hoc and Faceted Search Task. While our Ad Hoc approach worked fairly well (as demonstrated by the high MAP), the results of the Faceted Search Task are not yet available. Our expectation is that weighted result aggregation will improve faceted search, since it acknowledges either the global or local importance of different documents in the results list. In addition, we expect that redundancy avoidance will lead to a more compact representation of the results list.

Acknowledgments Frans Adriaans was supported by the Netherlands Organization for Scientific Research (NWO) grants # 612.066.513 and 446.010.027. Jaap Kamps was supported by NWO under grants # 612.066.513, 639.072.601, and 640.005.001. Marijn Koolen was supported by NWO under grant # 639.072.601.

Bibliography

[1] O. Ben-Yitzhak, N. Golbandi, N. Har'El, and R. Lempel. Beyond basic faceted search. In *WSDM'08*, 2008.

[2] C. Li, N. Yan, S. B. Roy, L. Lisham, and G. Das. Facetedpedia: Dynamic generation of query-dependent faceted interfaces for wikipedia. In *Proceedings of WWW 2010*, 2010.

[3] 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.