# Social Recommendation and External Resources for Book Search

Romain Deveaud[1], Eric SanJuan[1], and Patrice Bellot[2]

[1] LIA - University of Avignon
339, Chemin des Meinajaries, F-84000 Avignon Cedex 9
{romain.deveaud,eric.sanjuan}@univ-avignon.fr
[2] LSIS - Aix-Marseille University
Domaine Universitaire de Saint Jérôme, F-13397 Marseille Cedex 20
patrice.bellot@lsis.org

**Abstract.** In this paper we describe our participation in the INEX 2011 Book Track and present our contributions. This year a brand new collection of documents issued from Amazon was introduced. It is composed of Amazon entries for real books, and their associated user reviews, ratings and tags.

We tried a traditional approach for retrieval with two query expansion approaches involving Wikipedia as an external source of information. We also took advantage of the social data with recommendation runs that use user ratings and reviews. Our query expansion approaches did not perform well this year, but modeling the popularity and the interestingness of books based on user opinion achieved encouraging results. We also provide in this paper an insight into the combination of several external resources for contextualizing tweets, as part of the Tweet Contextualization track (former QA track).

## 1 Introduction

Previous editions of the INEX Book Track focused on the retrieval of real out-of-copyright books [3]. These books were written almost a century ago and the collection consisted of the OCR content of over 50,000 books. It was a hard track because of vocabulary and writing style mismatches between the topics and the books themselves. Information Retrieval systems had difficulties to found relevant information, and assessors had difficulties judging the documents.

This year, for the books search task, the document collection changed. It is now composed of the Amazon pages of real books. IR systems must now search through bibliographic information, user reviews and ratings for each book, instead of searching through the whole content of the book. The topics were extracted from the LibraryThing[1] forums and represent real requests from real users.

[1] http://www.librarything.com/

S. Geva, J. Kamps, and R. Schenkel (Eds.): INEX 2011, LNCS 7424, pp. 68–79, 2012.

This year we experimented with query expansion approaches and recommendation methods. Like we already did for the INEX 2010 Book track, we used a language modeling approach to retrieval. We started by using Wikipedia as an external source of information, since many books have their dedicated Wikipedia article [4]. We associate a Wikipedia article to each topic and we select the most informative words from the articles in order to expand the query. For our recommendation runs, we used the reviews and the ratings attributed to books by Amazon users. We computed a "social relevance" probability for each book, considering the amount of reviews and the ratings. This probability was then interpolated with scores obtained by Maximum Likelihood Estimates computed on whole Amazon pages, or only on reviews and titles, depending on the run.

The rest of the paper is organized as follows. The following Section gives an insight into the document collection whereas Section 3 describes the our retrieval framework. Finally, we describe our runs in Section 4 and discuss some results in Sections 5 and 6.

## 2 The Amazon Collection

The document used for this year's Book Track is composed of Amazon pages of existing books. These pages consist of editorial information such as ISBN number, title, number of pages etc... However, in this collection the most important content resides in social data. Indeed Amazon is social-oriented, and user can comment and rate products they purchased or they own. Reviews are identified by the `<review>` fields and are unique for a single user: Amazon does not allow a forum-like discussion. They can also assign tags of their creation to a product. These tags are useful for refining the search of other users in the way that they are not fixed: they reflect the trends for a specific product. In the XML documents, they can be found in the `<tag>` fields. Apart from this user classification, Amazon provides its own category labels that are contained in the `<browseNode>` fields.

**Table 1.** Some facts about the Amazon collection

| | |
|---|---|
| **Number of pages (i.e. books)** | 2,781,400 |
| **Number of reviews** | 15,785,133 |
| **Number of pages that contain at least a review** | 1,915,336 |

## 3 Retrieval Model

### 3.1 Sequential Dependence Model

Like in 2010, we used a language modeling approach to retrieval [5]. We use Metzler and Croft's Markov Random Field (MRF) model [6] to integrate multiword phrases in the query. Specifically, we use the Sequential Dependance Model

(SDM), which is a special case of the MRF. In this model three features are considered: single term features (standard unigram language model features, $f_T$), exact phrase features (words appearing in sequence, $f_O$) and unordered window features (requiring words to be close together, but not necessarily in an exact sequence order, $f_U$).

Documents are thus ranked according to the following scoring function:

$$\begin{aligned} score_{SDM}(Q,D) = \lambda_T \sum_{q \in Q} f_T(q,D) \\ + \lambda_O \sum_{i=1}^{|Q|-1} f_O(q_i, q_{i+1}, D) \\ + \lambda_U \sum_{i=1}^{|Q|-1} f_U(q_i, q_{i+1}, D) \end{aligned}$$

where the features weights are set according to the author's recommendation ($\lambda_T = 0.85, \lambda_O = 0.1, \lambda_U = 0.05$). $f_T$, $f_O$ and $f_U$ are the log maximum likelihood estimates of query terms in document $D$, computed over the target collection with a Dirichlet smoothing.

### 3.2 External Resources Combination

As previously done last year [2], we exploited external resources in a Pseudo-Relevance Feedback (PRF) fashion to expand the query with informative terms. Given a resource $\mathcal{R}$, we form a subset $\mathcal{R}_Q$ of informative documents considering the initial query $Q$ using pseudo-relevance feedback. To this end we first rank documents of $\mathcal{R}$ using the SDM ranking function. An entropy measure $H_{\mathcal{R}_Q}(t)$ is then computed for each term $t$ over $\mathcal{R}_Q$ in order to weigh them according to their relative informativeness:

$$H_{\mathcal{R}_Q}(t) = -\sum_{w \in t} p(w|\mathcal{R}_Q) \cdot \log p(w|\mathcal{R}_Q)$$

These external weighted terms are finally used to expand the original query. The ranking function of documents over the target collection $\mathcal{C}$ is then defined as follows:

$$score(Q,D) = score_{SDM}(Q,D) + \frac{1}{|\mathcal{S}|} \sum_{\mathcal{R}_Q \in \mathcal{S}} \sum_{t \in \mathcal{R}_Q} H_{\mathcal{R}_Q}(t) \cdot f_T(t,D)$$

where $\mathcal{S}$ is the set of external resources.

For our official experiments with the Book Track we only considered Wikipedia as an external resource, but we also conducted unofficial experiments on the Tweet Contextualization track after the workshop. In order to extract a comprehensive context from a tweet, we used a larger set $\mathcal{S}$ of resources. It is composed of four general resources: Wikipedia as an encyclopedic source, the New

York Times and GigaWord corpora as sources of news data and the category B of the ClueWeb09 collection as a web source. The English GigaWord LDC corpus consists of $4,111,240$ newswire articles collected from four distinct international sources including the New York Times. The New York Times LDC corpus contains $1,855,658$ news articles published between 1987 and 2007. The Wikipedia collection is a recent dump from July 2011 of the online encyclopedia that contains $3,214,014$ documents. We removed the spammed documents from the category B of the ClueWeb09 according a standard list of spams for this collection[2]. We followed authors recommendations [1] and set the "spamminess" threshold parameter to 70, the resulting corpus is composed of $29,038,220$ web pages. We present the results in the dedicated sections below.

### 3.3 Wikipedia Thematic Graphs

In the previous methods we expand the query with words selected from pages directly related to the query. Here, we wanted to select broader, more general words that could stretch topic coverage. The main idea is to build a thematic graph of Wikipedia articles in order to generate a set of articles that (ideally) completely covers the topic.

For this purpose we use anchor texts and their associated hyperlinks in the first Wikipedia page associated to the query. We keep the term extraction process detailed in Section 3.2 for selecting a Wikipedia page highly relevant to the query. We extract informative words from this page using the exact same method as above. But we also extract all anchor texts in this page. Given $T^{\mathcal{W}}$ the set of words extracted by entropy from the Wikipedia article $\mathcal{W}$ and $A^{\mathcal{W}}$ its set of anchor texts. We then compute the intersection between the set $T^{\mathcal{W}}$ and each anchor text $A_i^{\mathcal{W}}$. The intersection is not null if at least one contextual informative word is present in the anchor text. We then consider that the Wikipedia article that is linked with the anchor text is thematically relevant to the first retrieved Wikipedia article. Then we sum the previously computed entropies for all words from $T^{\mathcal{W}}$ occurring in the anchor text, which gives a confidence score for anchor $A_i^{\mathcal{W}}$. The computation of this score can be formalized as follows:

$$s_P(A_i^{\mathcal{W}}) = \sum_{t \in T^{\mathcal{W}} \cap A_i^{\mathcal{W}}} H_{\mathcal{W}}(t)$$

This thematic link hypothesis between Wikipedia articles relies on the fact that anchor texts are well-written and reviewed by the community. Each contributor can edit or correct an article while moderators can prevent abuses. This behavior was previously noted by [8] within the frame of experiements on the semantic relations that exist between lexical units. This study shows that using Wikipedia, an open and collaborative resource, achieves better results than the use of ontologies or hand-crafted taxonomies in some cases. These reflections hence justify our use of anchor texts to model thematic links between Wikipedia articles, or every other collaborative resource.

[2] http://plg.uwaterloo.ca/~gvcormac/clueweb09spam/

We can iterate and construct a directed graph of Wikipedia articles linked together. Children node pages (or *sub-articles*) are weighted half that of their parents in order to minimize a potential *topic drift*. We avoid loops in the graph (i.e. a child node can not be linked to one of his elder) because it brings no additional information. It also could change weights between linked articles. Informative words are then extracted from the sub-articles and incorporated to our retrieval model like another external resource.

### 3.4 Social Opinion for Book Search

The test collection used this year for the Book Track contains Amazon pages of books. These pages are composed amongst others of editorial information, like the number of pages or the blurb, user ratings and user reviews. However, contrary to the previous years, the actual content of the books is not available. Hence, the task is to rank books according to the sparse informative content and the opinion of readers expressed in the reviews, considering that the user ratings are integers between 1 and 5.

Here, we wanted to model two social popularity assumptions: a product that has a lot of reviews must be relevant (or at least popular), and a high rated product must be relevant. Then, a product having a large number of good reviews really must be relevant. However in the collection there is often a small amount of ratings for a given book. The challenge was to determine whether each user rating is significant or not. To do so, we first define $X_R^D$ a random set of "bad" ratings (1, 2 or 3 over 5 points) for book $D$. Then, we evaluate the statistical significant differences between $X_R^D$ and $X_R^D \cup X_U^D$ using Welch's t-test, where $X_U^D$ is the actual set of user rating for book $D$. Finally, we take the complement of the test $p$-value as the probability that reviewers like the book.

The underlying assumption is that significant differences occur under two different situations. First, when there is a small amount of user ratings $(X_U^i)$ but they all are very good. For example this is the case of good but little-known books. Second, when there is a very large amount of user ratings but there are average. Hence this statistical test gives us a single estimate of both likability and popularity.

We use our SDM baseline defined in section 3.1 and incorporate the above recommendation estimate:

$$score_{recomm}(Q, D) = \lambda_D \ score_{SDM}(Q, D) + (1 - \lambda_D) \ t_D$$

where the $\lambda_D$ parameter was set based on the observation over the test topics made available to participants for training purposes. Indeed we observed on these topics that the $t_D$ had no influence on the ranking of documents after the hundredth result (average estimation). Hence we fix the smoothing parameter to:

$$\lambda_D = \frac{\arg\max_D score_{SDM}(Q, D) - score_{SDM}(Q, D)_{100}}{N_{Results}}$$

In practice, this approach is re-ranking of the results of the SDM retrieval model based on the popularity and the likability of the different books.

## 4 Runs

This year we submitted 6 runs for the Social Search for Best Books task only. We used Indri[3] for indexing and searching. We did not remove any stopword and used the standard Krovetz stemmer.

**baseline-sdm.** This run is the implementation of the SDM model described in Section 3.1. We use it as a strong baseline.

**baseline-tags-browsenode.** This is an attempt to produce an improved baseline that uses the Amazon classification as well as user tags. We search all single query terms in the specific XML fields (`<tag>` and `<browseNode>`). This part is then combined with the SDM model, which is weighted four times more than the "tag searching" part. We set these weights empirically after observations on the test topics. The Indri syntax for the query `schumann biography` would typically be:

```
#weight (
  0.2 #combine ( #1(schumann).tag #1(biography).tag
                 #1(schumann).browseNode #1(biography).browseNode )
  0.8 #weight ( 0.85 #combine( Schumann Biography )
                0.1  #combine( #1(schumann biography) )
                0.05 #combine( #uw8(schumann biography) ) )
       )
```

**sdm-wiki.** This run is the implementation of the external resources combination model described in Section 3.2, only applied to a single resource: Wikipedia. The Wikipedia API was queried on August, 2011. For each topic we extract the 20 top informative words based on their entropy measure from the top ranked article given by the Wikipedia API. We then reformulate the initial query by adding these words with their entropy as weights. The motivation to do this was that there are many books that have their dedicated Wikipedia article [4]. If we could select the proper article and extract informative words about a book topic or a book series, it could help retrieval.

**sdm-wiki-anchors.** This run is the implementation of the Wikipedia thematic graph approach described in Section 3.3. For each topic, we queried the Wikipedia API to retrieve the first ranked article. We then computed all thematic links between this first Wikipedia article (we call it $reference$) and all the others that are linked to it. We then extract the 20 top informative words from these linked articles in order to enrich the query with several thematically linked sources. In these experiments we only consider the top 5 linked articles with best $s_P(A_i^{\mathcal{W}})$ confidence score.

[3] `http://www.lemurproject.org`

**sdm-reviews-combine.** This run uses the social information contained in the user reviews, it is the implementation of the approach described in Section 3.4. First, a **baseline-sdm** is performed. We then extract the number of reviews and their ratings for each document previously retrieved. A probability that the book is popular is then computed with a Welch's t-test. This interestingness and popularity score is finally interpolated to the SDM score.

**Recommendation.** This run is similar to the previous one except that we compute a query likelihood estimate only on the `<title>` and on the `<content>` fields, instead of considering the whole document like the SDM does. Scores for the title and the reviews, and the popularity of the books are interpolated the same way as above. The sum of these three scores gives a recommendation score for each book based only on its title and on user opinions, without tanking into account any other editorial information.

## 5 Book Search Results and Discussion

The evaluation results shown below are based on the official INEX 2011 Social Books topic set, consisting of 211 topics from the LibraryThing discussion groups. There are two separate sets of relevance judgements. The first set is derived from the suggestions from members of the discussion groups, and is considered as the principal mean of evaluation for this task. The second set is based on judgements from Amazon Mechanical Turk for 24 out of the 211 topics. We present the results for the first set of relevance judgements in Table 2.

We observe that our **recommendation** approach performs the best amongst our other runs, while our two query expansion approaches with Wikipedia both fail. Our **baseline-sdm** run do not use any additional information except the user query (which is in fact the title of the corresponding LibraryThing thread), hence this is a good mean of comparison for other runs using social information for example. Despite that using an external encyclopedic resource like Wikipedia do not work for improving the initial query formulation, we see that a traditional pseudo-relevance feedback (PRF) approach achieved the best results overall this year. Indeed the approach of the University of Amsterdam (p4) was to expand the query with 50 terms extracted from the top 10 results, either performing over a full index or over an index that only include social tags (such as reviews, tags and ratings). The latter performed the best with their PRF approach, and it is coherent with the results of our **recommendation** run. Indeed in this run we only consider the content of the user reviews, which correspond to a limited version of the social index mentioned above. It also suggests that the baseline model is quite effective and selects relevant feedback documents, which is confirmed by the results computed with the Amazon Mechanical Turk judgements shown in Table 3.

In this table we see that the baselines perform very well compared to the others, and it confirms that a language modeling base system performs very well on this test collection. It is very good at retrieving relevant documents in the first

**Table 2.** Official results of the Best Books for Social Search task of the INEX 2011 Book track, using judgements derived from the LibraryThing discussion groups. Our runs are identified by the *p*62 prefix and are in boldface.

| **Run** | nDCG@10 | P@10 | MRR | MAP |
|---|---|---|---|---|
| p4-inex2011SB.xml_social.fb.10.50 | 0.3101 | 0.2071 | 0.4811 | 0.2283 |
| p54-run4.all-topic-fields.reviews-split.combSUM | 0.2991 | 0.1991 | 0.4731 | 0.1945 |
| p4-inex2011SB.xml_social | 0.2913 | 0.1910 | 0.4661 | 0.2115 |
| p4-inex2011SB.xml_full.fb.10.50 | 0.2853 | 0.1858 | 0.4453 | 0.2051 |
| p54-run2.all-topic-fields.all-doc-fields | 0.2843 | 0.1910 | 0.4567 | 0.2035 |
| **p62.recommendation** | 0.2710 | 0.1900 | 0.4250 | 0.1770 |
| p54-run3.title.reviews-split.combSUM | 0.2643 | 0.1858 | 0.4195 | 0.1661 |
| **p62.sdm-reviews-combine** | 0.2618 | 0.1749 | 0.4361 | 0.1755 |
| **p62.baseline-sdm** | 0.2536 | 0.1697 | 0.3962 | 0.1815 |
| **p62.baseline-tags-browsenode** | 0.2534 | 0.1687 | 0.3877 | 0.1884 |
| p4-inex2011SB.xml_full | 0.2523 | 0.1649 | 0.4062 | 0.1825 |
| ***wiki-web-nyt-gw*** | 0.2502 | 0.1673 | 0.4001 | 0.1857 |
| p4-inex2011SB.xml_amazon | 0.2411 | 0.1536 | 0.3939 | 0.1722 |
| **p62.sdm-wiki** | 0.1953 | 0.1332 | 0.3017 | 0.1404 |
| **p62.sdm-wiki-anchors** | 0.1724 | 0.1199 | 0.2720 | 0.1253 |
| p4-inex2011SB.xml_lt | 0.1592 | 0.1052 | 0.2695 | 0.1199 |
| p18.UPF_QE_group_BTT02 | 0.1531 | 0.0995 | 0.2478 | 0.1223 |
| p18.UPF_QE_genregroup_BTT02 | 0.1327 | 0.0934 | 0.2283 | 0.1001 |
| p18.UPF_QEGr_BTT02_RM | 0.1291 | 0.0872 | 0.2183 | 0.0973 |
| p18.UPF_base_BTT02 | 0.1281 | 0.0863 | 0.2135 | 0.1018 |
| p18.UPF_QE_genre_BTT02 | 0.1214 | 0.0844 | 0.2089 | 0.0910 |
| p18.UPF_base_BT02 | 0.1202 | 0.0796 | 0.2039 | 0.1048 |
| p54-run1.title.all-doc-fields | 0.1129 | 0.0801 | 0.1982 | 0.0868 |

**Table 3.** Top runs of the Best Books for Social Search task of the INEX 2011 Book track, using judgements obtained by crowdsourcing (Amazon Mechanical Turk). Our runs are identified by the *p*62 prefix and are in boldface.

| **Run** | nDCG@10 | P@10 | MRR | MAP |
|---|---|---|---|---|
| **p62.baseline-sdm** | 0.6092 | 0.5875 | 0.7794 | 0.3896 |
| p4-inex2011SB.xml_amazon | 0.6055 | 0.5792 | 0.7940 | 0.3500 |
| **p62.baseline-tags-browsenode** | 0.6012 | 0.5708 | 0.7779 | 0.3996 |
| p4-inex2011SB.xml_full | 0.6011 | 0.5708 | 0.7798 | 0.3818 |
| p4-inex2011SB.xml_full.fb.10.50 | 0.5929 | 0.5500 | 0.8075 | 0.3898 |
| **p62.sdm-reviews-combine** | 0.5654 | 0.5208 | 0.7584 | 0.2781 |
| p4-inex2011SB.xml_social | 0.5464 | 0.5167 | 0.7031 | 0.3486 |
| p4-inex2011SB.xml_social.fb.10.50 | 0.5425 | 0.5042 | 0.7210 | 0.3261 |
| p54-run2.all-topic-fields.all-doc-fields | 0.5415 | 0.4625 | 0.8535 | 0.3223 |

ranks which is an essential quality for a system that performs PRF. Hence a query expansion approach can be very effective on this dataset, but feedback documents must come from the target collection and not from an external resource. It is however important to note that these judgements are coming from people that often are not experts or that do not have the experience of good readers.

Their assessments may then come from the suggestions of well-known search engines or directly from Amazon. This behavior could possibly explain the high performances of the baselines for the AMT judgements set.

To confirm this assessment, we tried to combine the four heterogenous resources mentioned in Section 3.2 and we reported the results on Table 2 under the unofficial run identified by ***wiki-web-nyt-gw***. Although the combination of multiple external resources does much better than using Wikipedia alone, it still does not beat our baseline. Hence can safely affirm that reformulating the query using a wide range of external sources of knowledge does not work when the target collection is mainly composed of recommendation or opinion-oriented text.

The other part of our contribution lies in the social opinion that we took into account in our ranking function. Indeed we are the only group that submitted runs that model the popularity and the likability of books based on user reviews and ratings. Royal School of Library and Information Science's group (p54) tried in their early experiments to define an helpfulness score for each review, aiming to give more weight to a review found truthful, and also tried to weigh books reviews according to their associated ratings. However these experiments showed that it didn't performed well compared to an approach where they sum the relevance score of all the reviews for a given book. The two runs we submitted that make use of social information (**recommendation** and **sdm-reviews-combine**) can both be viewed as a re-ranking of the baseline, and both of them improve its performance. The recommendation run only uses reviews content and the title of the book for the retrieval of books while the sdm-reviews-combine run uses the whole content of the Amazon/LibraryThing pages. The fact that the recommendation run performs best than the sdm-reviews-combine is coherent with the approach of Royal School of Library and Information Science described above. Additional information seems to be considered as noise while the real informative content is situated inside the reviews, but this may also be a smoothing issue. Indeed the size of the reviews are much larger than any other component in the documents ($\approx$ 156 words per review, while tags are only composed of 1 or 2 words), and defining specific smoothing parameter values for each field based on the average length of their length could perform better.

## 6 Contextualizing Tweets by Combining General Resources

Considering that the use of an external resource did not bring anything to social book search, we wanted to evaluate our resource combination approach on another track. This approach intuitively matches well against the Tweet Contextualization one (former QA track). Indeed its purpose is to extract relevant passages from Wikipedia in order to generate a readable summary (500 words maximum) giving insights into a topic of current interest. These topics are represented by tweets, which are in fact titles of New York Times articles. We use the exact same approach previously described in Section 3.2. Tweets are enriched with additional information coming from the various external resources, and sentences are extracted from the target Wikipedia collection to form a contextual excerpt. The organizers provided a full baseline for participants that could not implement their own index of the Wikipedia collection. It is composed of a full state-of-the-art XML-element retrieval system which was already available for the previous edition of the INEX QA track [7]. We tried every combination of one, two, three and four resources, but we only report the approaches that use a single resource and the full combination of the four. Some official results are reported in Table 4 as well as those of our unofficial runs.

**Table 4.** Official results of the INEX 2011 Tweet Contextualization track. Our runs are unofficial and are in boldface, runs in italic are the official baselines.

| Run | Unigram | Bigram | With 2-gap |
|---|---|---|---|
| ID12R_IRIT_default.run | 0.8271 | 0.9012 | 0.9028 |
| ID126R_Run1.run | 0.7982 | 0.9031 | 0.9037 |
| ID128R_Run2.run | 0.8034 | 0.9091 | 0.9094 |
| ID138R_Run1.run | 0.8089 | 0.9150 | 0.9147 |
| ID129R_Run2.run | 0.8497 | 0.9252 | 0.9253 |
| **wiki-web-nyt-gw** | 0.8267 | 0.9273 | 0.9289 |
| *Baseline_sum.run* | 0.8363 | 0.9350 | 0.9362 |
| **gigaword** | 0.8409 | 0.9371 | 0.9383 |
| ID18R_Run1.run | 0.8642 | 0.9368 | 0.9386 |
| **nyt** | 0.8631 | 0.9437 | 0.9443 |
| ID46R_JU_CSE_run1.run | 0.8807 | 0.9453 | 0.9448 |
| **web** | 0.8522 | 0.9454 | 0.9466 |
| **wiki** | 0.8515 | 0.9454 | 0.9471 |
| *Baseline_mwt.run* | 0.9064 | 0.9777 | 0.9875 |

The evaluation metrics considers the absolute normalized log-difference between the result passages and the textual assessments. The main metric is **With 2-gap** and evaluates the frequency differences between "*pairs of consecutive lemmas, allowing the insertion between them of a maximum of two lemmas*". We see that despite the fact that the passage extraction method we use was the baseline provided by the organizers, using a single resource to reformulate the initial query (or tweet) does not beat the *Baseline_sum.run*. However we see that the GigaWord and the NYT corpora are the ones that harm retrieval the less, mainly because of their coverage of the news topics. Surprisingly, the use of Wikipedia as a single source of expansion (i.e. pseudo-relevance feedback) achieves the worst results of our unofficial runs. We did not have the time to further investigate, but this may be a first coverage indication of Wikipedia for the given topics. It also suggests that constructing a coherent summary based exclusively on Wikipedia for a given news topic is not an easy task. Despite the negative effect of single resources, we observe that the combination of the four resources performs better than the baseline. The improvement is statistically significant (t-test with

$p$-value $< 0.05$). The combination thus contextualizes effectively the information need from 3 different points of view corresponding to the 3 types of resources, namely: encyclopedic, news and web. This contextualization acts in the form of contextual features extracted from the different sources and used to reformulate the initial query (or tweet). These results are very promising and encouraging, and we aim at experimenting other means of contextualization with several kind of external data.

## 7 Conclusions

In this paper we presented our contributions for the INEX 2011 Book track. One main observation from this year's Book track was that the baselines based on a language modeling approach to retrieval were very hard to beat. This also helped the approaches that used pseudo-relevance feedback to perform well. We proposed a query expansion method that exploit four different resources as external sources of expansion terms. This method considers the most informative words of the best ranked articles in order to reformulate the query. It did not perform well overall and did not manage to beat our baseline. We also tried to build a limited thematic graph of Wikipedia articles in order to extract more expansion terms, but this approach was even less effective. This collection is mainly composed of user reviews that contain opinion-oriented text more than factual information, and using external information seems not to work here. However we tried to extend our method to the Tweet Contextualization track and saw that combining the four resources is effective and beats the baseline, while every single resource harms passage retrieval.

We also submitted two runs to the Book track that took advantage of the social information available in the Amazon collection. They exploit the number of reviews and the user ratings to compute popularity and likability scores that we interpolate with query likelihood probabilities. These approaches showed to be effective but still need some improvements, especially with the estimation of a "good" review. We aim to model the quality of a reviewer for the upcoming year, thus weighting the different reviews of a given book according to several criteria.

## References

1. Cormack, G., Smucker, M., Clarke, C.: Efficient and effective spam filtering and re-ranking for large web datasets. Information Retrieval (2011)
2. Deveaud, R., Boudin, F., Bellot, P.: LIA at INEX 2010 Book Track. In: Geva, S., Kamps, J., Schenkel, R., Trotman, A. (eds.) INEX 2010. LNCS, vol. 6932, pp. 118–127. Springer, Heidelberg (2011)
3. Kazai, G., Koolen, M., Kamps, J., Doucet, A., Landoni, M.: Overview of the INEX 2010 Book Track: Scaling Up the Evaluation Using Crowdsourcing. In: Geva, S., Kamps, J., Schenkel, R., Trotman, A. (eds.) INEX 2010. LNCS, vol. 6932, pp. 98–117. Springer, Heidelberg (2011)

4. Koolen, M., Kazai, G., Craswell, N.: Wikipedia pages as entry points for book search. In: Proceedings of the Second ACM International Conference on Web Search and Data Mining, WSDM 2009, pp. 44–53. ACM, New York (2009)
5. Metzler, D., Croft, W.B.: Combining the language model and inference network approaches to retrieval. Inf. Process. Manage. 40, 735–750 (2004)
6. Metzler, D., Croft, W.B.: A markov random field model for term dependencies. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2005, pp. 472–479. ACM, New York (2005)
7. SanJuan, E., Bellot, P., Moriceau, V., Tannier, X.: Overview of the INEX 2010 Question Answering Track (QA@INEX). In: Geva, S., Kamps, J., Schenkel, R., Trotman, A. (eds.) INEX 2010. LNCS, vol. 6932, pp. 269–281. Springer, Heidelberg (2011)
8. Strube, M., Ponzetto, S.P.: Wikirelate! computing semantic relatedness using wikipedia. In: Proceedings of the 21st National Conference on Artificial Intelligence, vol. 2, pp. 1419–1424 (2006)