Do Social Information Help Book Search?

Ludovic Bonnefoy[1], Romain Deveaud[1] and Patrice Bellot[2]

[1] LIA - University of Avignon
ludovic.bonnefoy@etd.univ-avignon.fr
romain.deveaud@univ-avignon.fr

[2] LSIS - Aix-Marseille University
patrice.bellot@lsis.org

Abstract. In this paper we describe our participation in the INEX 2012 Book Track. The collection enters its second year of age and is composed of Amazon and LibraryThing entries for real books, and their associated user reviews, ratings and tags.
Like in 2011, we tried a simple yet effective approach of reranking books using a social component that takes into account both popularity and ratings. We did experiments using tags as well.

1 Introduction

Previous editions of the INEX Book Track focused on the retrieval of real out-of-copyright books [1]. 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.

In 2011, for the books search task, the document collection changed and is now composed of the Amazon pages of real books. IR systems must now search through editorial data and 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.

Like we already did last year, we used a Language Modeling approach to retrieval. For our recommendation runs, we used the reviews and the ratings attributed to books by Amazon users. We computed a "social score" for each book, considering the amount of reviews and the ratings. This score is then used to modify the initial ranking obtained by a Markov Random Field baseline that proved to be highly effective last year. We also used tags to build a profile for both a query and the books of the collection which we compared to rank the books.

The rest of the paper is organized as follows. The following Section gives an insight into the document collection whereas Section 2 describes the our retrieval framework. Finally, we describe our runs in Section 3.

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

2 Retrieval Model

2.1 Sequential Dependence Model

We used a language modeling approach to retrieval [2]. We use Metzler and Croft's Markov Random Field (MRF) model [3] 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 (require words to be close together, but not necessarily in an exact sequence order, f_U).

Finally, documents are 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.

2.2 Modeling book likeliness

The basic idea behind this likeliness is that if a book has a lot of reviews and if its ratings are generally good, then it must be a very good book.

$$\mathcal{L}(D) = \log(\#reviews(D)) \times \frac{\sum_{r \in \mathcal{R}_D} r}{\#reviews(D)}$$

where $\mathcal{R}_D$ is the set of all ratings given by the users for the book D, and $\#reviews(D)$ is the number of reviews.

We further rerank the books by weighting the previously computed SDM with the likeliness score. The scoring function of a book D given a query Q is thus defined as follows:

$$s(Q, D) = \mathcal{L}(D) \times score_{SDM}(Q, D)$$

2.3 Modeling book thematic relatedness

We want to represent each query Q by a thematic profile and rank books according to their relatedness to it. For this first attempt at using thematic (or

topic) relatedness we choosed to rely exclusively on user tags associated with the books in the collection. We consider as a thematic profile a set of tags weighted according to their significance for Q and we call it a tag profile (TP). As a preprocessing step, a tag profile is associated to each book in the collection. Tags are weighted according to a classic tf.idf measure (where the tf is the number of users who associated the tag to the book).

The main issue is to estimate a tag profile for a query. To construct it, inspired by the pseudo relevance feedback method, we summed the profile of the x top ranked books retrieved by mean of a information retrieval model (more details in runs section). Once the query's tag profile is build, we can compare book's tag profile to it with a vector similarity measure like the cosinus.

Finally, books of the collection are ranked according to the similarity of their profile to the query's one.

3 Runs

We submitted 4 runs for the Social Search for Best Books task. We used Indri[2] for indexing and searching. We did not remove any stopword and used the standard Krovetz stemmer.

mrf-booklike This run is the implementation of the SDM model described in Section 2.1 with the likeliness score.

IOT30 and IT30 Those two runs are based on the tag profile approach presented in Section 2.3. In this approach four parameters have to be fixed : The number x of top ranked books used to build the query's tag profile, the weight given to each tag in query's profile, the information retrieval model used to retrieved books and the similarity measure to compare profiles. For both runs, x is fixed to 30, Indri's language modeling approach is used and the similarity measure is the cosinus angle between vectors.

The last parameter is the weight given to each tag of the query profile. For the IOT30 run, the t_i tag's weight is compute as the sum of its tf.idf weight in each of the top x books returned by Indri:

$$w(t_i) = \sum_{b \in Top_x} tf.idf(t_i, b)$$

where b is one the Top_x books retrieved.

However, we had the intution that all selected books can not contribute equally to the weight of a tag. So, for the IT30 run, we combine the tf.idf of a tag in a book with the relevance of this book according to the retrieval model used in order to penalize contribution of less relevant books:

$$w(t_i) = \sum_{b \in Top_x} tf.idf(t_i, b) \times score(b, Q)$$

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

where $score(b, Q)$ is the measure of relevance of the book b according to Indri.
deduce

B_IT30_30 For the last run we wanted to take advantage of both particularities of mrf-booklike run and a tag profile based one. We combine the mrf-booklike run to the IT30 run by mean of a logistic regression. We trained a model with two classes (relevant or not) and the book scores predicted by both runs as features. Training instances were the 30 top ranked books returned by each run along with their relevance judgment deduce from 2011 qrels.

4 Results

Table 1 shows 2012 official results and 2011 non official results for our 4 runs. The combination of mrf-booklike and IT30 improves the results as expected on 2012 qrels while it increase them dramatically for 2011 qrels. In 2012 our second run is mrf-booklike but it is the worse when evaluated with 2011 qrels which is surprising.

So, according to both officials results in 2012 and non official results based on 2011 qrels we can not answer to our question : *"Do Social Information Help Book Search?"*. It seems to vary a lot depending on evaluation corpus used. In order to explain those big differences we will need to make further experiments and more fine-grained analysis.

Run	nDCG@10	P@10	Recip rank	Recall@10
2012 Qrels - Officials				
Best Run 2012	0.1492	0.1198	0.3069	0.1527
B_IT30_30	0.1339	0.1260	0.3410	0.1659
mrf-booklike	0.1295	0.1250	0.3584	0.1514
IOT30	0.1141	0.1240	0.2933	0.1503
IT30	0.1082	0.1187	0.2999	0.1426
2011 Qrels - Non officials				
B_IT30_30	0.3408	0.2282	0.5398	
Best Run 2011	0.3101	0.2071	0.4811	
IT30	0.2995	0.2105	0.4626	
IOT30	0.2927	0.2081	0.4524	
mrf-booklike	0.2786	0.1890	0.4337	

Table 1. Comparison of our official results at INEX 2012 and non official results for 2011. The runs are ranked according to nDCG@10.

5 Conclusions

In this paper we presented our contributions for the INEX 2012 Book Track. We proposed a simple method for reranking books based on their likeliness and an effective way to take into account user tags. Finally a combination of both methods with a logistic regression approach gives the best results. Results does not allow us to answer on the usefulness of social information for book search despite quite good results.

References

1. Gabriella Kazai, Marijn Koolen, Antoine Doucet, and Monica Landoni. Overview of the INEX 2010 Book Track: At the Mercy of Crowdsourcing. In Shlomo Geva, Jaap Kamps, Ralf Schenkel, and Andrew Trotman, editors, *Comparative Evaluation of Focused Retrieval*, pages 98–117. Springer Berlin / Heidelberg, 2011.
2. D. Metzler and W. B. Croft. Combining the language model and inference network approaches to retrieval. *Inf. Process. Manage.*, 40:735–750, September 2004.
3. Donald Metzler and W. Bruce Croft. 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 '05, pages 472–479, New York, NY, USA, 2005. ACM.