Book Recommendation based on Social Information

Chahinez Benkoussas[†] and Patrice Bellot[†]

[†]LSIS – Aix-Marseille University
chahinez.benkoussas@lsis.org
patrice.bellot@lsis.org

Abstract : *In this paper, we present our contribution in INEX 2013 Social Book Search Track. This track aim to explore social information (users reviews, ratings, etc...) for the libraryThing and Amazon collections of real books.*

In our submissions for SBSTrack, we rerank books by combining the Sequential Dependence Model (SDM) and the use of social component that takes into account both ratings and helpful votes.

Keywords : XML retrieval, controlled metadata, book recommendation, re-ranking.

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. The topics and the books of the collection have a different vocabulary and writing style. Information Retrieval systems had difficulties to found relevant information, and assessors had difficulties judging the documents.

The document collection is composed of the Amazon [1] pages of real books. IR must search through editorial data, user reviews and ratings for each book, instead of searching through the whole content of the book. The topics were extracted from LibraryThing [2] forums and represent real request from real users.

We have chosen to use 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

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

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

each book, considering the amount of reviews and the ratings. This score was then interpolated with scores obtained by a *Marcov Random Field* (MRF) baseline. We also used the "*helpfulvotes*" and "*totalvotes*" values for each rating given by users to modify the ranking obtained by the combination of social and MRF scores.

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

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 multi word phrases in the query. Specifically, we use the *Sequential Dependence Model* (SDM), which is a special case of 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:

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

where the feature weights are set according to the author's recommendation (λ_T= 0.85, λ_O= 0.1, λ_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

We modeled book likeliness basing on the following idea: more the number of reviews it has, more interesting it is reading it (it may not be a good or popular book but a book that has a high impact).

$$Likeliness(D) = \frac{\sum_{r \in R_D} r}{|Reviews_D|}$$

where $R_D$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 books according to a linear interpolation of the previously computed SDM score with the likeliness score, using a coefficient (α) to control the influence of each model. The scoring function of a book D given a query Q is thus defined as follows:

$$SDM\_Likeliness(Q, D) = \alpha \cdot (SDM(Q, D)) + (1 - \alpha) \cdot (Likeliness(D))$$

where α is a constant set according to previous results (done on 2011 and 2012 datasets), with a default value of 0,89.

2.3 Modeling usefulness of ratings' books

Into the collection of books, we have a rating for each review given by users, the rating value can or cannot be useful depending on user votes. we have chosen to weight the value of rating with the value of helpful votes according to this formula:

$$Usefulness(D) = \frac{\sum_{r \in R_D, t \in T_D, h \in H_D} r \times (\frac{t}{h})}{|Reviews_D|}$$

where R_D, T_D, H_D are respectively, the sets of all *ratings*, *totalvotes* and *helpfulvotes* given by the users for the book D, and $|Reviews\ _D|$is the number of reviews.

We further rerank books according to a linear interpolation of the previously computed SDM score with the usefulness score, using a coefficient (β) to control the influence of each model. The scoring function of a book D given a query Q is thus defined as follows:

$$SDM\_Usefulness(Q, D) = \beta \cdot (SDM(Q, D)) + (1 - \beta) \cdot (Usefulness(D))$$

where β is a constant set according to previous results (done on 2011 and 2012 datasets), with a default value of 0,93.

3 Run

We submitted 3 runs for the Social Book Search Task. We used Indri [3] for indexing and searching. We did not remove any stopword and used the standard Krovetz stemmer. Only query part of the topic has been used for the three runs.

SDM_run : This run is the implementation of the Sequential Dependence Model (SDM) described in Section 2.1.

SDM_Rating_run : This run combine the implementation of the Sequential Dependence Model and the use of social information which is the "*Ratings*" given by users. Description is given in Section 2.2

SDM_HV_run : For the last run we combine the implementation of the Sequential Dependence Model and the use of social information which are "*ratings*", "*helpful votes*" and "*total votes*" given by users. We weighted the value of "rating" with the rate of helpful votes as presented in Section 2.3.

4 Conclusion

This paper presents our contributions on the INEX 2013 Social Book Search Track. We proposed a simple method for reranking books based on their likeliness and an effective way to take into account user helpful votes. Finally we combine both methods with a linear interpolated function.

We are disappointed with the official results this year (nDCG@10 = 0.0571 for the baseline "*SDM_run*") compared to those obtained last year with the the approach that was our baseline this year (nDCG@10 = 0,1295) and we seek for explanations of a software problem. On the other side, the proposed extensions (nDCG@10 = 0.596 for "*SDM_Rating_run*" and nDCG@10 = 0.0576 for "*SDM_HV_run*") improved the results of the baseline.

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

5 Acknowledgements

This work was supported by the French program "*Investissements d'Avenir - Développement de l'Economie Numérique*" under the project Inter-Textes #O14751-408983.

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.