KNOW At The Social Book Search Lab 2016 Suggestion Track

Hermann Ziak and Roman Kern

Know-Center GmbH
Inffeldgasse 13
8010 Graz, Austria
hziak, rkern@know-center.at

Abstract. Within this work represents the documentation of our approach on the Social Book Search Lab 2016 where we took part in the suggestion track. The main goal of the track was to create book recommendation for readers only based on their stated request within a forum. The forum entry contained further contextual information, like the user's catalogue of already read books and the list of example books mentioned in the user's request. The presented approach is mainly based on the metadata included in the book catalogue provided by the organizers of the task. With the help of a dedicated search index we extracted several potential book recommendations which were re-ranked by the use of an SVD based approach. Although our results did not meet our expectation we consider it as first iteration towards a competitive solution.

Keywords: SVD, recommender engine, content-based information retrieval

1 Introduction

The Social Book Search (SBS) Lab 2016 has the objective to investigate book search in a setting where the user is not inferring an actual query into a search engine. The setting can more be considered as a recommender system automatically inferring the information need of the user by its context. Therefore the organizers prepared three different tracks the 'interactive' track, the 'mining' track and the 'suggestion' track. This paper represents the approach and the results of our participation in the 'suggestion' track. Here the task was the extraction of the user's information need within a posting of the user on the 'LibraryThing'[1]. According to this initial post of the user the final goal was to suggest a ranked list of books, in that regard LibraryThing 'work IDs', out of a provided catalogue of books. The by the organizers supplied data contained a feature rich dataset of the posting it self, according metadata of the user's history and in some cases examples in terms of mentioned book titles.

[1] www.librarything.com

This catalogue contained a collection of about 2.7 million crawled records of the 'Amazon.com'[2] platform [2] combined with the information for the equivalent work available on 'LibraryThing' jointed into structured XML files.
Within the field of recommender systems there is a vast amount of sophisticated approaches to tackle such problems [1]. Basically, all those approaches fall within three different categories: content-based, collaborative, and hybrid approaches. For our first attempt to contribute to the SBS Lab we decided to use the, from our perspective least complex, content-based approach. Further, BM25 based approaches accomplished good results in recent years within this lab [3]. Therefore the book collection was indexed with all the according metadata in a Apache Lucene [3] based search engine. With the help of the mentioned search engine we implemented an approach basically just relying on the use of the tags and browse nodes provided within this dataset. Although the achieved results are below our expectations we consider it as first step towards a competitive approach.

2 Approach

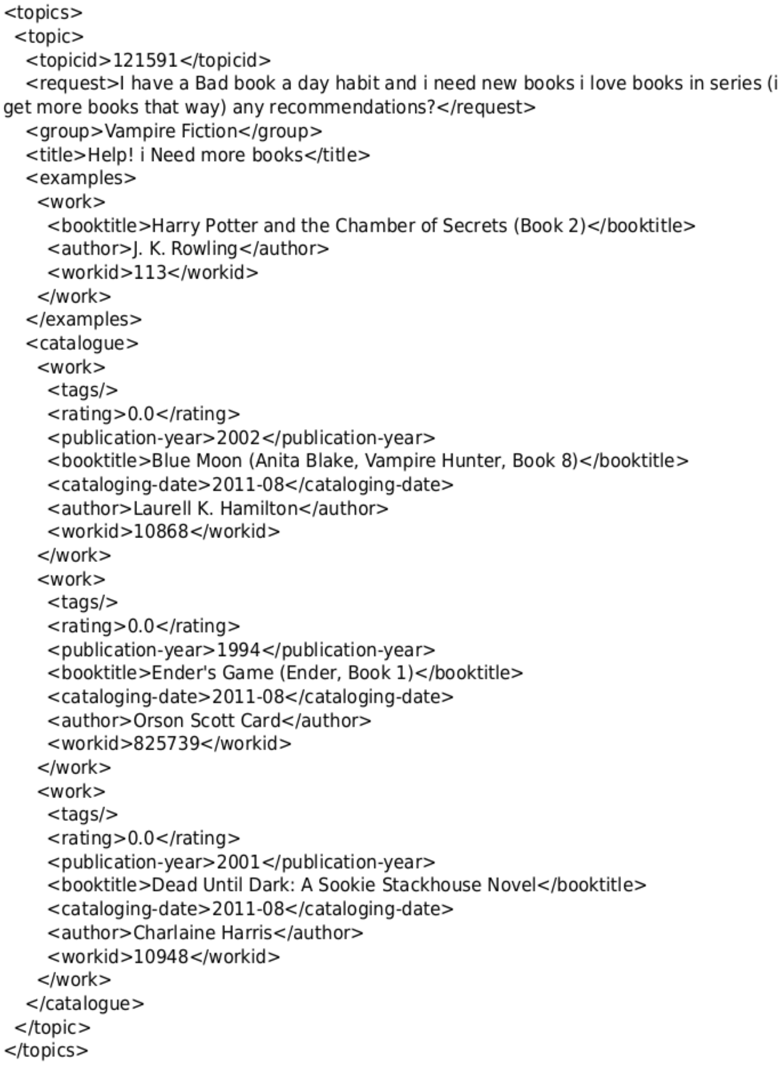
The main idea of our approach was to rely on the provided metadata, in that regard the tags (T), browse nodes (BN) and ISBNs, of the provided book catalogue. Those fields contain textual content that categorize the book. (e.g. Sci-fi, Novel, Child's book)

The provided user postings consisted of several fields: I) the name of the group where the request was initially posted (e.g. 'Sci-Fi Novels'), II) the title of the entry, III) the actual request in form of natural language, IV) potential book examples, V) the catalogue of already read books of the user. An example of such posting can be found in Figure 1.

As initial step provided ISBN numbers of the example books and the users catalogue books were sent to the search engine to get an initial dataset of T and BNs. To this set of T and BNs weights were assigned based on a heuristic. Ts or BNs that were contained within the examples of the according request were considered to be more important and therefore got higher weights assigned than ones just appearing within the catalogue of the user. The weight was further increased if the tag was contained within the provided posting, title or group. The outcome of this first steps was to separated set, one containing only the weighted tags, one containing only the weighted browse nodes. Out of this two lists two queries were formulated. With those queries we only search within the tags or the browse nodes fields. The outcome of this step were two groups of potential book candidates. To remove duplicates and already read books by the user both candidate lists were filtered by the books already mentioned within the user's catalogue or the example books. To consolidate and re-rank this two candidate lists we applied a SVD approach which are frequently used within recommender systems [4,5] Both sets were transformed into one utility matrix where the rows represented all the documents. The columns represented all T

[2] www.amazon.com

[3] https://lucene.apache.org/



```
<topics>
 <topic>
  <topicid>121591</topicid>
  <request>I have a Bad book a day habit and i need new books i love books in series (i
get more books that way) any recommendations?</request>
  <group>Vampire Fiction</group>
  <title>Help! i Need more books</title>
  <examples>
   <work>
    <booktitle>Harry Potter and the Chamber of Secrets (Book 2)</booktitle>
    <author>J. K. Rowling</author>
    <workid>113</workid>
   </work>
  </examples>
  <catalogue>
   <work>
    <tags/>
    <rating>0.0</rating>
    <publication-year>2002</publication-year>
    <booktitle>Blue Moon (Anita Blake, Vampire Hunter, Book 8)</booktitle>
    <cataloging-date>2011-08</cataloging-date>
    <author>Laurell K. Hamilton</author>
    <workid>10868</workid>
   </work>
   <work>
    <tags/>
    <rating>0.0</rating>
    <publication-year>1994</publication-year>
    <booktitle>Ender's Game (Ender, Book 1)</booktitle>
    <cataloging-date>2011-08</cataloging-date>
    <author>Orson Scott Card</author>
    <workid>825739</workid>
   </work>
   <work>
    <tags/>
    <rating>0.0</rating>
    <publication-year>2001</publication-year>
    <booktitle>Dead Until Dark: A Sookie Stackhouse Novel</booktitle>
    <cataloging-date>2011-08</cataloging-date>
    <author>Charlaine Harris</author>
    <workid>10948</workid>
   </work>
  </catalogue>
 </topic>
</topics>
```

Fig. 1. Example of a user's suggestion request

and BN within both candidate lists. As value within the resulting matrix 1 was set if T or BN was actually represented within the actual document or 0 if not. Finally the resulting matrix was decomposed. Further, we created a vector out of all the tags corresponding to the columns of the matrix was created. As values for this tags the initial applied weights were set that were generated in the third processing step according to Figure 2. If the tag was not existing in the documents of the examples or catalogue the values was set to zero. To get the final candidate ranking we applied this vector to the decomposed matrix. The final outcome of the whole process was the ranked list of documents which we could map to ISBN numbers. In a last step this ISBN numbers were translated into the LibraryThing work IDs. This detour over the ISBN numbers was only necessary since we had to report work IDs in the task and the Amazon dataset only contained the ISBN numbers. The whole process is visualized in Figure 2.

3 Results

Table 1 shows the results of one of our conducted pre-tests. In this particular case, we only submitted a small catalogue of books to the system to get similar books recommended.

Table 1. One of the pre-test conducted to initially validate the approach. Within the catalogue row the book titles that were used as input are stated. The result row contains the titles returned by the system.

	Book Tiles
catalogue	Data Mining: Practical Machine Learning Tools and Techniques ... Statistics, Data Analysis, and Decision Modeling Software Architecture in Practice
result	Introduction to Algorithms Software Engineering: A Practitioner's Approach Artificial Intelligence: A Modern Approach Artificial Intelligence (Handbook Of Perception And Cognition) Machine Learning (Mcgraw-Hill International Edit) Prolog Programming for Artificial Intelligence An Introduction to Support Vector Machines and Other Kernel-based Learning Methods

Table 2 shows the results of our approach on the testing data provided by the lab organizers. The difference between the two runs submitted to the lab only lie in the different weights applied for containing the tags within the examples, group name, tiles, text or the users book catalogue.

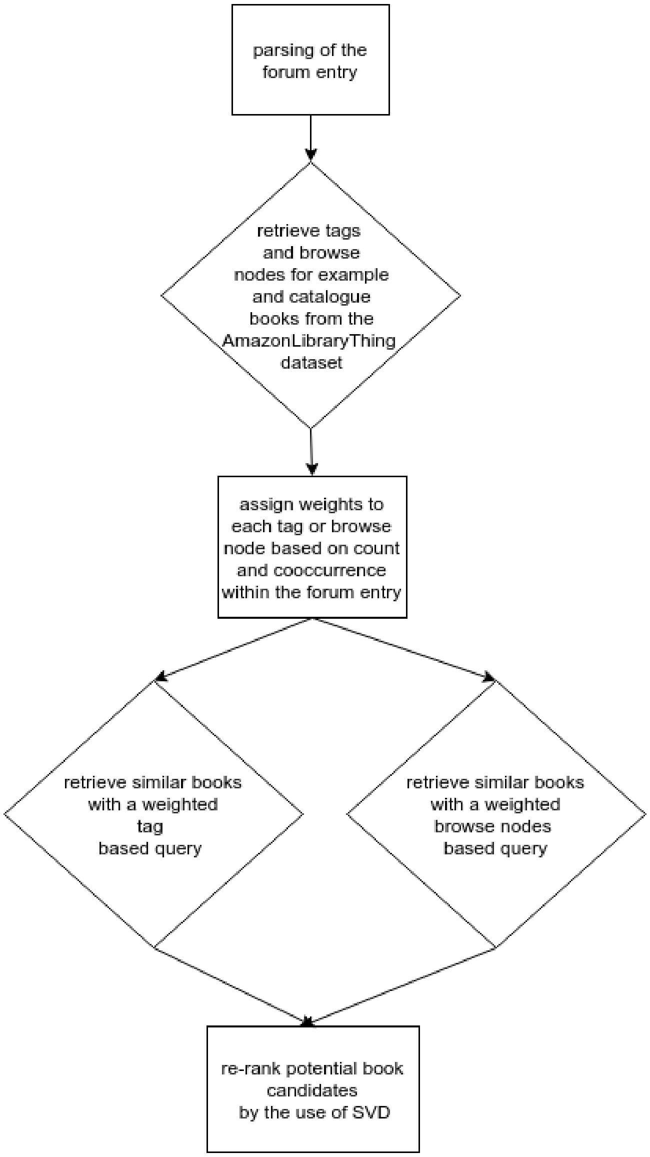

parsing of the forum entry
retrieve tags and browse nodes for example and catalogue books from the AmazonLibraryThing dataset
assign weights to each tag or browse node based on count and cooccurrence within the forum entry
retrieve similar books with a weighted tag based query
retrieve similar books with a weighted browse nodes based query
re-rank potential book candidates by the use of SVD

Fig. 2. Overview of the processing pipeline. At first the forum entry is parsed. In the next step the books of the examples and catalogue are retrieved from the Amazon dataset. The free text is analyzed for tags and browse nodes co-occurrences. For each tags or browse nodes a corresponding weight is assigned representing the count within all books and within the free text. Afterwards two queries are send to the index; one containing the tags as keywords applied on the tag field within the index one with the same procedure for the browse nodes. The resulting two lists are combined within the utility matrix that combines and re-ranks both lists by the use of SVD.

Table 2. Official results of our submissions on the testing set provided by the lab organizers. Reported measures are the normalized discounted cumulative gain at 10, mean reciprocal rank, mean average precision, and recall at 1000.

run	**nDCG@10**	**MRR**	**MAP**	**R@1000**
submission 2	0.0058	0.0227	0.0010	0.0013
submission 1	0.0018	0.0084	0.0004	0.0004

4 Discussion

Although our approach seemed to work fine in our initial pre-test the system did not perform well in the official results provided by the organizer. Although the system performed below our expectation we did not expect excellent results since we did not optimize at all on the testing set.

5 Conclusion and Future Work

The presented content-based system is considered a first initial step towards a competitive system. The next logical steps will be the optimization towards the training dataset. As first approach to improve the results we consider to evaluate different setting upon the weights. We also consider to use general learning to rank approaches.

Acknowledgments

The presented work was developed within the EEXCESS project funded by the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement number 600601. The Know-Center is funded within the Austrian COMET Program - Competence Centers for Excellent Technologies - under the auspices of the Austrian Federal Ministry of Transport, Innovation and Technology, the Austrian Federal Ministry of Economy, Family and Youth and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency FFG.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. Knowledge and Data Engineering, IEEE Transactions on 17(6), 734–749 (2005)
2. Beckers, T., Fuhr, N., Pharo, N., Nordlie, R., Fachry, K.N.: Overview and results of the inex 2009 interactive track. In: Research and Advanced Technology for Digital Libraries, pp. 409–412. Springer (2010)
3. Koolen, M., Bogers, T., Gäde, M., Hall, M., Huurdeman, H., Kamps, J., Skov, M., Toms, E., Walsh, D.: Overview of the clef 2015 social book search lab. In: Experimental IR Meets Multilinguality, Multimodality, and Interaction, pp. 545–564. Springer (2015)

4. Miller, B.N., Konstan, J.A., Riedl, J.: Pocketlens: Toward a personal recommender system. ACM Transactions on Information Systems (TOIS) 22(3), 437–476 (2004)
5. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Incremental singular value decomposition algorithms for highly scalable recommender systems. In: Fifth International Conference on Computer and Information Science. pp. 27–28. Citeseer (2002)