Cross-Document Search Engine For Book Recommendation

Chahinez Benkoussas
Aix-Marseille Université, CNRS, LSIS UMR 7296
13397, Marseille. France
chahinez.benkoussas@lsis.org
Aix-Marseille Université, CNRS, CLEO OpenEdition UMS 3287, 13451
13397, Marseille. France
chahinez.benkoussas@openedition.org

Patrice Bellot
Aix-Marseille Université, CNRS, LSIS UMR 7296
13397, Marseille. France
patrice.bellot@lsis.org
Aix-Marseille Université, CNRS, CLEO OpenEdition UMS 3287, 13451
13397, Marseille. France
patrice.bellot@openedition.org

ABSTRACT

A new combination of multiple Information Retrieval approaches are proposed for book recommendation based on complex users' queries. We used different theoretical retrieval models: probabilistic as InL2 (Divergence From Randomness model) and language models and tested their interpolated combination. We considered the application of a graph based algorithm in a new retrieval approach to related document network comprised of social links. We called Directed Graph of Documents (DGD) a network constructed with documents and social information provided from each one of them. Specifically, this work tackles the problem of book recommendation in the context of CLEF Labs precisely Social Book Search track. We established a specific strategy for queries searching after separating query set into two genres "*Analogue*" and "*Non-Analogue*" after analyzing users' needs. Series of reranking experiments demonstrate that combining retrieval models and exploiting linked documents for retrieving yield significant improvements in terms of standard ranked retrieval metrics. These results extend the applicability of link analysis algorithms to different environments.

Keywords

Document retrieval, InL2, language model, book recommendation, PageRank, graph modeling, Social Book Search.

1. INTRODUCTION

There has been much work both in the industry and academia on developing new approaches to improve the performance of retrieval and recommendation systems over the last decade. The aim is to help users to deal with information overload and provide recommendation for books, restaurants or movies. Some vendors have incorporated recommendation capabilities into their commerce services, such as Amazon.

Existing document retrieval approaches need to be improved to satisfy users' information needs. Most systems use classic information retrieval models, such as language models or probabilistic models. Language models have been applied with a high degree of success in information retrieval applications [29–31]. This was first introduced by Ponte and Croft in [27]. They proposed a method to score documents, called *query likelihood* in two steps: estimate a language model for each document and then rank documents according to the likelihood scores resulting from the estimated language model. Markov Random Field model, proposed by Metzler and Croft in [19] considers query term proximity in documents by estimating term dependencies in the context of language modeling approach. Alternatively, Divergence From Randomness model, proposed by Amati and Van Rijsbergen [2], measures the global informativeness of the term in the document collection. It is based on the idea :"*The more the term occurrences diverge from random throughout the collection, the more informative the term is*" [28]. One limit of such models is that the distance between query terms in documents is not considered.

Users' queries differ by their type of needs. In book recommendation, we identified two genres of queries : "*Analogue*" and "*Non-Analogue*" that we describe in the following sections. In this paper, the first proposed approach combines probabilistic and language models to improve the retrieval performances and show that the two models act much better

CBRecSys 2015, September 20, 2015, Vienna, Austria.

in the context of book recommendation.

In recent years, an important innovation in information retrieval is the exploitation of relationships between documents, e.g. Google's PageRank [25]. It has been successful in Web environments, where the relationships are provided by hyperlinks between documents. We present a new approach for linking documents to construct a graph structure that is used in retrieving process. In this approach, we exploit the PageRank algorithm for ranking documents with respect to users' queries. In the absence of manually-created hyperlinks, we use social information to create a Directed Graph of Documents (DGD) and argue that it can be treated in the same manner as hyperlink graphs. Our experiments will show that incorporating graph analysis algorithms in document retrieval improves the performance in term of the standard ranked retrieval metrics.

Our work focuses on search in the book recommendation domain, in the context of CLEF Labs Social Book Search track. We tested our approaches on collection contains Amazon/LibraryThing book descriptions and set of queries, called topics, extracted from the LibraryThing discussion forums.

2. RELATED WORK

This work is first related to the area of document retrieval models, more specially language models and probabilistic models. The unigram language models are most often used for ad hoc Information Retrieval work but several researchers explored the use of language modeling for capturing higher order dependencies between terms. Bouchard and Nie in [8] showed significant improvements in retrieval effectiveness with a new statistical language model for the query based on completing the query by terms in the user's domain of interest, reordering the retrieval results or expanding the query using lexical relations extracted from the user's domain of interest.

Divergence From Randomness (DFR) is one of several probabilistic models that we have used in our work. Abolhassani and Fuhr have investigated several possibilities for applying Amati's DFR model [2] for content-only search in XML documents. [1].

There has been an increasing use of techniques based on graphs constructed by implicit relationships between documents. Kurland and Lee performed structural reranking based on centrality measures in graph of documents which has been generated using relationships between documents based on language models [14]. In [16], Lin demonstrates the possibility to exploit document networks defined by automatically-generated content-similarity links for document retrieval in the absence of explicit hyperlinks. He integrates the PageRank scores with standard retrieval score and shows a significant improvement in ranked retrieval performance. His work was focused on search in the biomedical domain, in the context of PubMed search engine. Perhaps the main contrast with our work is that links were not induced by generation probabilities or linguistic items.

3. INEX SOCIAL BOOK SEARCH TRACK AND TEST COLLECTION

Social Book Search (SBS) task[1] aims to evaluate the value of professional and user's metadata for book search on the Web. The main goal is to exploit search techniques to deal with complex information needs and complex information sources that include user profiles, personal catalogs, and book descriptions.

The SBS task provides a collection of 2.8 million book description crawled by the University of Duisburg-Essen from Amazon[2] [4] and enriched with content from LibraryThing[3], which is an online service to help people catalog their books easly. Books are stored in XML files and identified by an ISBN. They contains information like: title information, Dewey Decimal Classification (DDC) code (for 61% of the books), category, Amazon product description, etc. Amazon records contain also social information generated by users like: tags, reviews, ratings (see Figure 1. For each book, Amazon suggests a set of "Similar Products" which represents a result of computed similarity based on content information and user behavior (purchases, likes, reviews, etc.) [13].

Figure 1: Example of book from the Amazon/LibraryThing collection in XML format

SBS task provides a set of queries called topics where users describe what they are looking for (books for a particular genre, books of particular authors, similar books to those that have been already read, etc.). These requests for recommendations are natural expressions of information needs for a large collection of online book records. The topics are crawled from LibraryThing discussion Forums.

The topic set consists of 680 topics in 2014. Each topic has a narrative description of the information need and other fields as illustrated in Figure 2.

[1] http://social-book-search.humanities.uva.nl/
[2] http://www.amazon.com/
[3] http://www.librarything.com/

```
<topic id="1116">
    <title>Which LISP?</title>
    <mediated_query>introduction book to Lisp</mediated_query>
    <group>Purely Programmers</group>
    <narrative>  It'll be time for me to shake things up and learn a new
        language soon. I had started on Erlang a while back and getting
        back to it might be fun. But I'm starting to lean toward Lisp--
        probably Common Lisp rather than Scheme. Anyone care to recommend
         a good first Lisp book? Would I be crazy to hope that there's
        one out there with an emphasis on using Lisp in a web development
         and/or system administration context? Not that I'm unhappy with
        PHP and Perl, but the best way for me to find the time to learn a
         new language is to use it for my work... </narrative>
</topic>
```

Figure 2: Example of topic, composed with multiple fields to describe user's need(s)

4. RETRIEVAL MODELS

This section describes the retrieval models we used for book recommendation and their combination.

4.1 InL2 of Divergence From Randomness

We used InL2, Inverse Document Frequency model with Laplace after-effect and normalization 2. This model has been used with success in different works [3,6,10,26]. InL2 is a DFR-based model (Divergence From Randomness) based on the Geometric distribution and Laplace law of succession.

4.2 Sequential dependence Model of Markov Random Field

Language models are largely used in Document Retrieval search for book recommendation [6, 7]. Metzler and Croft proposed Markov Random Field (MRF) model [18, 20] that integrates multi-word phrases in the query. Specifically, we used the Sequential Dependence Model (SDM), which is a special case of MRF. In this models co-occurrence of query terms is taken into consideration. SDM builds upon this idea by considering combinations of query terms with proximity constraints which are: 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:

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

Where feature weights are set based on the authorâĂŹs recommendation ($\lambda_T = 0.85$, $\lambda_O = 0.1$, $\lambda_U = 0.05$) in [7]. f_T , f_O and f_U are the log maximum likelihood estimates of query terms in document D, computed over the target collection using a Dirichlet smoothing. We applied this model to the queries using Indri[4] Query Language[5].

4.3 Combining Search Systems

Combining the output of many search systems, in contrast to using just a single one improves the retrieval effectiveness as proved in [5] where Belkin combined the results of probabilistic with vector space models. On the basis of this approach, In our work, we combined the probabilistic model, InL2 with language model SDM. This combination takes into account both the informativeness of query terms and their dependencies in the document collection. Each retrieval model uses different weighting schemes therefore the scores should be normalized. We used the maximum and minimum scores according to Lee's formula [15].

$$normalizedScore = \frac{oldScore - minScore}{maxScore - minScore}$$

It has been shown in [6] that InL2 and SDM models have different levels of retrieval effectiveness, thus it is necessary to weight individual model scores depending on their overall performance. We used an interpolation parameter (α) that we varied to improve retrieval effectiveness.

5. GRAPH MODELING

In [17], the authors have exploited networks defined by automatically-generated content-similarity links for document retrieval. We provided document analysis to find new way to link them. In our case, we exploited a special type of similarity based on several factors. This similarity is provided by Amazon and corresponds to "Similar Products" given generally for each book. The degree of similarity depends on social information like: number of clicks or purchases and content-based information like book attributes (book description, book title, etc.). The exact formula used by Amazon to combine social and content based information to compute similarity is proprietary. The idea behind this linking method is that documents linked with such type of similarity, the probability that they are in the same context is higher than if they are not connected.

To perform data modeling into DGD, we extracted the "Similar Products" links between documents in order to construct the graph structure. Once used it to enrich results from the retrieval models, in the same spirit as pseudo-relevance-feedback. Each node in the DGD represents document (Amazon description of book), and has set of properties:

- ID: book's ISBN
- $content$: book description that include many other properties (title, product description, author(s), users' tags, content of reviews, etc.)
- $MeanRating$: average of ratings attributed to the book

[4] `http://www.lemurproject.org/indri/`
[5] `http://www.lemurproject.org/lemur/IndriQueryLanguage.php`

- PR : book's PageRank

Edges in the DGD are directed and correspond to Amazon similarity, so given nodes $\{A, B\} \in S$, if A points to B, B is suggested as Similar Product to A. In the Figure 3, we show an example of DGD, network of documents. The DGD network contains 1 645 355 nodes (89.86% of nodes are within the collection and the rest are outside) and 6 582 258 edges.

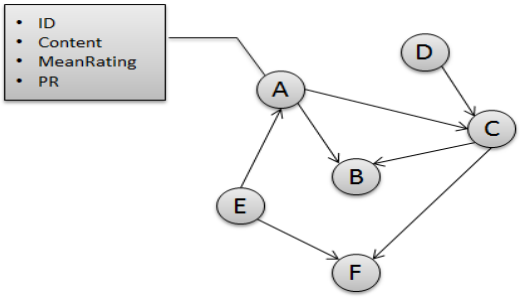

Figure 3: Example of Directed Graph of Documents

Figure 4 shows the general architecture of our document retrieval system with two-level document search. In this system, the *IR Engine* finds all relevant documents for user's query. Then, the *Graph Search* module selects resulting document returned by *Graph Analysis* module. The *Graph Structured Data* is a network constructed using *Social Information Matrix* and enriched by *Compute PageRank* module. The *Social Information Matrix* is constructed by two modules: "Ratings" and "Similar Products" Extraction from the *Data Collection* that contains description books in XML format. *Scoring Ranking* module combines scores of documents resulting from *IR Engine* and *Graph Analysis* modules and reranks them.

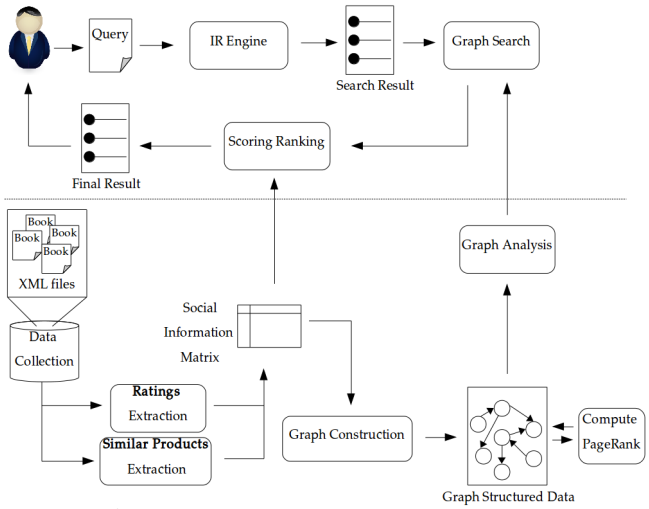

Figure 4: Architecture of document retrieval approach based on graph of documents

In this section, the collection of documents is denoted by C. In C, each document d has a unique ID. The set of queries called topics is denoted by T, the set $D_{init} \subset C$ refers to the documents returned by the initial retrieval model. $StartingNode$ identifies a document from D_{init} used as input to the graph processing algorithms in the DGD. The set of documents present in the graph is denoted by S. D_{t_i} indicates the documents retrieved for topic $t_i \in T$.

5.1 Our Approach

The DGD network contains useful information about documents that can be exploited for document retrieval. Our approach is based, first on results of a traditional retrieval engine, then on the DGD network to find new documents. The idea is to suppose that the suggestions given by Amazon can be relevant to the user queries.

Algorithm 1 takes as inputs: D_{init} returned list of documents for each topic by the retrieval techniques described in Section 3, DGD network and parameter β which is the number of the top selected $StartingNode$ from D_{init} denoted by $D_{StartingNodes}$. We fixed β to 100 (10% of the returned list for each topic). The algorithm returns a list of recommendations for each topic denoted by "D_{final}". It processes topic by topic, and extracts the list of all neighbors for each $StartingNode$. It performs mutual Shortest Paths computation between all selected $StartingNode$ in DGD. The two lists (neighbors and nodes in computed Shortest Paths) are concatenated after that all duplicated nodes are deleted. The set of documents in returned list is denoted by D_{graph}. A second concatenation is performed between initial list of documents and D_{graph} (all duplications are deleted) in new final list of retrieved documents, D_{final} reranked using different reranking schemes.

Algorithm 1 Retrieving based on DGD feedback

1: $D_{init} \leftarrow$ Retrieving Documents for each $t_i \in T$
2: **for** each $D_{t_i} \in D_{init}$ **do**
3: $\quad D_{StartingNodes} \leftarrow$ first β documents $\in D_{t_i}$
4: $\quad$ **for** each $StartingNode$ **in** $D_{StartingNodes}$ **do**
5: $\quad\quad D_{graph} \leftarrow D_{graph} + neighbors(StartingNode, DGD)$
6: $\quad\quad D_{SPnodes} \leftarrow$ **all** $D \in ShortestPath(StartingNode, D_{StartingNodes}, DGD)$
7: $\quad\quad D_{graph} \leftarrow D_{graph} + D_{SPnodes}$
8: $\quad\quad$ Delete all duplications from D_{graph}
9: $\quad D_{final} \leftarrow D_{final} + (D_{t_i} + D_{graph})$
10: Delete all duplications from D_{final}
11: Rerank D_{final}

Figure 5 shows an illustration of the document retrieval approach based on DGD feedback.

6. EXPERIMENTS AND RESULTS

In this section, we describe the experimental setup we used for our experiments. Furthermore, we present the different reranking schemes used in previously defined approaches. We discuss the results we achieved by using the InL2 retrieval model, its combination to the SDM model, and retrieval system proposed in our approach that uses the DGD network.

6.1 Experiments setup

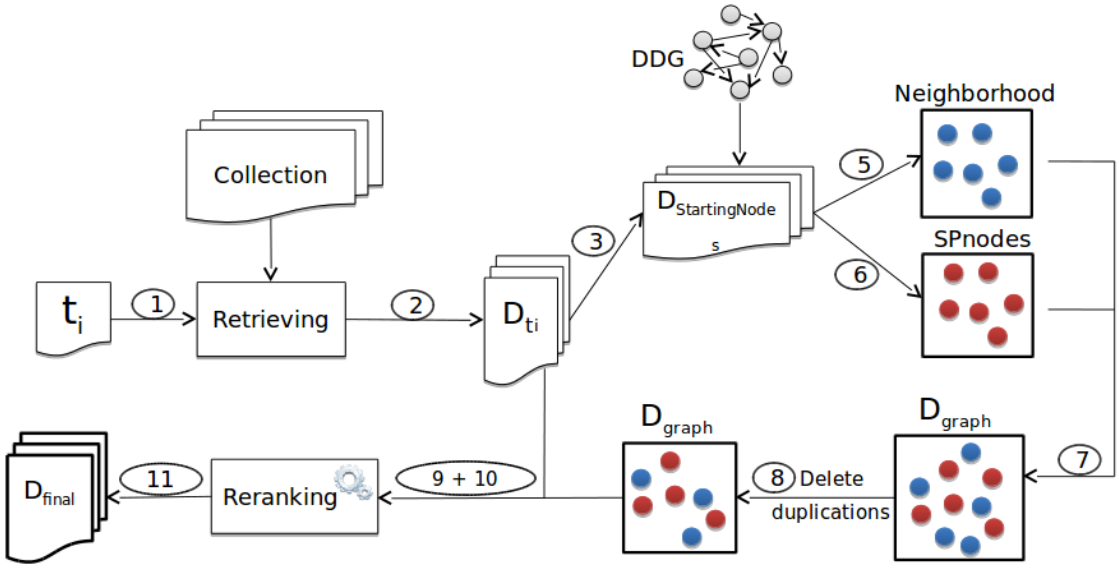

Figure 5: Book retrieval approach based on DGD feedback. Numbers on the arrows refer to the instructions in the Algorithm 1

For our experiments, we used different tools that implement retrieval models and handle the graph processing. First, we used *Terrier* (TERabyte RetrIEveR)[6] *Information Retrieval* framework developed at the University of Glasgow [21–23]. Terrier is a modular platform for rapid development of large-scale IR applications. It provides indexing and retrieval functionalities. It is based on DFR framework and we used it to deploy InL2 model described in section 4.1. Further information about Terrier can be found at http://ir.dcs.gla.ac.uk/terrier.

A preprocessing step was performed to convert INEX SBS corpus into the Trec Collection Format[7], by considering that the content of all tags in each XML file is important for indexing; therefore the whole XML file was transformed on one document identified by its ISBN. Thus, we just need two tags instead of all tags in XML, the ISBN and the whole content (named text).

Secondly, *Indri*[8], *Lemur Toolkit for Language Modeling and Information Retrieval* was used to carry out a language model (SDM) described in section 4.2. Indri is a framework that provides state-of-the-art text search methods and a rich structured query language for big collections (up to 50 million documents). It is a part of the Lemur project and developed by researchers from UMass and Carnegie Mellon University. We used Porter stemmer and performed Bayesian smoothing with Dirichlet priors (Dirichlet prior $\mu = 1500$).

In section 5.1, we have described our approach based on DGD which includes graph processing. We used NetworkX[9] tool of Python to perform shortest path computing, neighborhood extraction and PageRank calculation.

To evaluate the results of retrieval systems, several measurements have been used for SBS task: Discounted Cumulative Gain (nDCG), the most popular measure in IR [11], Mean Average Precision (MAP) which calculates the mean of average precisions over a set of queries, and other measures: Recip Rank and Precision at the rank 10 (P@10).

6.2 Reranking Schemes

Two approaches were proposed. The first one (see section 4.3) merges the results of two different information retrieval models which are the Language Model (SDM) and DFR model (InL2). For topic t_i, the models give 1000 documents and each retrieved document has an associated score. The linear combination method uses the following formula to calculate final score for each retrieved document d by SDM and InL2 models:

$$S_{final}(d, t_i) = \alpha * S_{InL2}(d, t_i) + (1 - \alpha) * S_{SDM}(d, t_i)$$

Where $S_{InL2}(d, t_i)$ and $S_{SDM}(d, t_i)$ are normalized scores. α is the interpolation parameter set up at 0.8 after several tests on the 2014 topics.

The second approach (described in 5.1) uses the DGD constructed from the "Similar Products" information. The document set returned by the retrieval model are fused to the documents in neighbors set and Shortest Path results. We tested many reranking methods that combine the retrieval model scores and other scores based on social information. For each document in the resulting list, we calculated the following scores:

- **PageRank**, computed using NetworkX tool. It is a well-known algorithm that exploits link structure

[6] http://terrier.org/
[7] http://lab.hypotheses.org/1129
[8] http://www.lemurproject.org/indri/
[9] https://networkx.github.io/

to score the importance of nodes in a graph. Usually, it was been used for hyperlink graphs such as the Web [24]. The values of PageRank are given by the following formula.

$$PR(A) = (1-d) + d(PR(T_1)/C(T_1) + ... + PR(T_n)/C(T_n))$$

Where document A has documents $T_1 ... T_n$ which point to it (i.e., Similar products). The parameter d is a damping factor set between 0 and 1 (0.85 in our case). $C(A)$ is defined as the number of links going out of page A.

- **Likeliness**, computed from information generated by users (reviews and ratings). It is based on the idea that more the book has a lot of reviews and good ratings, the more interesting it is (it may not be a good or popular book but a book that has a high impact).

$$Likeliness(D) = \log(\#reviews(D)) \times \frac{\sum_{r \in R_D} r}{\#reviews(D)}$$

Where $\#reviews(D)$ is the number of reviews attributed to D, R_D is the set of reviews of D.

The computed scores were normalized using this formula: $normalized_{score} = old_{score}/max_{score}$. After that, to combine the results of retrieval systems and each of normalized scores, an intuitive solution is to weight the retrieval model scores with the previously described scores (normalized PageRank and Likeliness). However, this would favor documents with high PageRank and Likeliness scores even though their content is much less related to the topics.

6.3 Results

We used two topic sets provided by INEX SBS task in 2014 (680 topics). The systems retrieve 1000 documents per topic. We assessed the narrative field of each topic and provided automatic classification of the topic set into 2 genres. *Analogue* topics (261) in which users give the already read books (generally, titles and authors) to have similar books. In the second genre "*Non-Analogue*" (356 topics), users describe their needs by defining the thematic, interested field, event, etc. without citing other books. Notify that, 63 topics are ignored because of their ambiguity.

In order to evaluate our IR methodologies described in sections 4.3, 5 we performed retrieving for each topic genre individually. The experimental results, which describe the performance of the different retrieval systems on Amazon/LibraryThing document collection, are shown in Table 1.

As illustrated in Table 1, the system that combines probabilistic model InL2 and the Language Model SDM (InL2_SDM) achieves a significant improvement for each topic set comparing to InL2 model (Baseline) but the improvement is highest for *Non-Analogue* topic set where the content of queries are more explicit than the other topic set. This improvement is mainly due to the increase of the number of relevant documents that are retrieved by both systems.

The results of run InL2_DDG_PR using the *Analogue* topic set confirm that exploiting structured documents and performing reranking with PageRank improves significantly performances but in contrast, it lowers the baseline performances when using the *Non-Analogue* topic set. This can be explained by the fact that *Analogue* topics contain examples of books (Figure 6) which require the use of graph to extract the similar connected books.

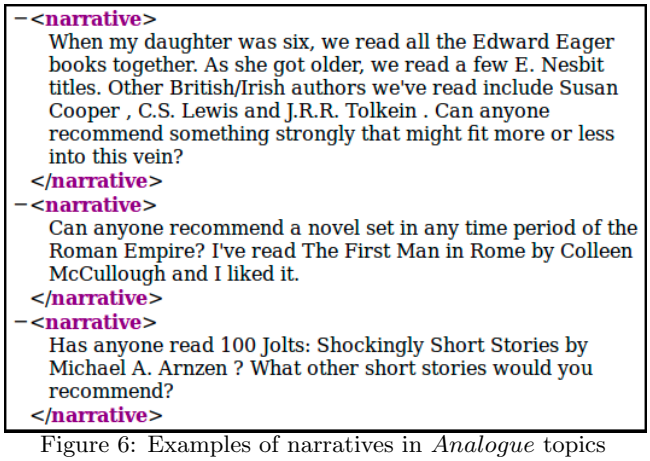

Figure 6: Examples of narratives in *Analogue* topics

Using Likeliness scores (in InL2_DGD_MnRtg) to rerank retrieved documents decreases significantly the baseline efficiency for the two topic sets. This means that ratings given by users don't provide any improvement for the reranking performances.

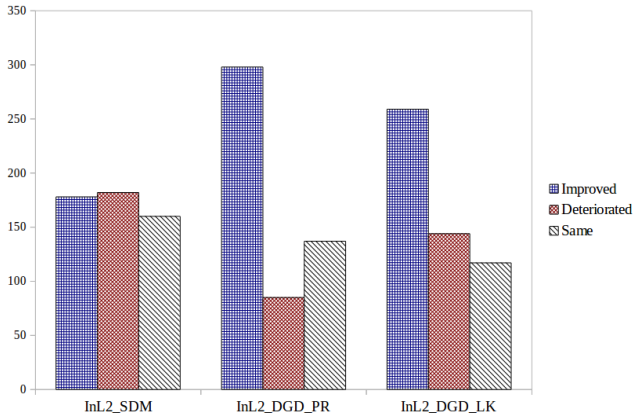

Figure 7: Histograms that demonstrate and compare the number of improved, deteriorated and same results' topics using the proposed approaches for MAP measure. (Baseline: InL2)

Figure 7 compares the number of improved, deteriorated and same results' topics between the baseline (InL2) and the proposed retrieval systems in term of MAP measure. The proposed systems based on DGD graph provide the highest number of improved topics compared with the combination of IR systems. More precisely, using PageRank to rerank document produces better results in term of improved topics. This results prove the positive impact of linked structure on document retrieval systems for book recommendation.

The depicted results confirm that we are starting with competitive baseline, suggesting that improvements contribute

Table 1: Experimental results. The runs are ranked according to nDCG@10. (∗) denotes significance according to Wilcoxon test [9]. In all cases, all of our tests produced two-sided p-value, $\alpha = 0.05$.

	Analogue **topics**				*Non-Analogue* **topics**			
Run	nDCG@10	Recip Rank	MAP	P@10	nDCG@10	Recip Rank	MAP	P@10
InL2	**0.1099**	**0.267**	**0.072**	**0.078**	**0.138**	**0.207**	**0.117**	**0.0579**
InL2_SDM	0.1115 (+1%*)	0.271 (+1%*)	0.073 (+0.6%)	0.079 (+1%*)	0.147(+6%*)	0.222(+7%*)	0.124(+5%*)	0.0630(+8%*)
InL2_DGD_PR	0.1111 (+1%*)	0.277 (+3%*)	0.068 (−5%*)	0.082 (+12%)	0.127(−7%*)	0.206(−0.6%*)	0.102(−12%*)	0.0570(−1%*)
InL2_DGD_LK	0.1043 (−5%)	0.275 (+2%)	0.064(−11%*)	0.082(+5%)	0.130(−5%)	0.214(+3%*)	0.100(−14%*)	0.0676(+16%)

by combining output retrieval systems and social link analysis are indeed meaningful.

7. HUMANITIES AND SOCIAL SCIENCES COLLECTION: GRAPH MODELING AND RECOMMENDATION

We tested the proposed approach of recommendation based on linked documents on Revues.org[10] collection. Revues.org is one of the four platforms of OpenEdition[11] portal dedicated to electronic resources in the humanities and social sciences (books, journals, research blogs, and academic announcements). Revues.org was founded in 1999 and today it hosts over 400 online journals, i.e. 149000 articles, proceedings ans editorials.

We built a network of documents from ASp[12] journal. It publishes research articles, publication listings and reviews related to the field of English for Specific Purposes (ESP) for both teaching and research. The network contains 500 documents and 833 relationships which represent bibliographic citations. Each relationship is constructed using BILBO [12], the reference parsing software. BILBO is constructed with annotated corpora from Digital Humanities articles from OpenEdition Revues.org platform. It automatic annotates bibliographic references in the bibliography section of each document and obtains the corresponding DOI (Digital Object Identifier) via CrossRef[13] API if such an identifier exists.

Each node in the citation network have a set of properties (*ID* which is its URL, *type*, it can be article, editorial, review of book, etc., and readers' clicks number that we called *popularity*). The recommender system applied on this network takes as input user query, generally a small set of short keywords, and performs retrieval step using Solr[14] search engine. The system extend the returned results with documents in the citation network by using graph algorithms (neighborhood search and shortest path algorithm) as described in section5.1. After that, we rerank documents according to the popularity property of each document.

We tested the system manually for a small set of user queries, and found that for most queries, the results were satisfying.

8. CONCLUSION AND FUTURE WORK

In this paper, we proposed and evaluated approaches of document retrieval in the context of book recommendation. We used the test collection of CLEF Labs Social Book Search track and the proposed topics in 2014 divided into two classes *Analogue* and "*Non-Analogue*".

We presented the first approach that combines the outputs of probabilistic model (InL2) and Language Model (SDM) using a linear interpolation after normalizing scores of each retrieval system. We have shown a significant improvement of baseline results using this combination.

A novel approach was proposed, based on Directed Graph of Documents (DGD) constructed from social relationships. It exploits link structure to enrich the returned document list by traditional retrieval model (InL2). We performed a reranking method using PageRank and Likeliness of each retrieved document.

In the future, we would like to construct an evaluation corpora from Revues.org collection and develop an evaluation process similar to that of INEX SBS task. Another interesting extension of our work would be using the learning to rank techniques to automatically adjust the settings of re-ranking parameters.

9. ACKNOWLEDGMENT

This work was supported by the French program Investissements d'Avenir FSN and the French Région PACA under the projects InterTextes and Agoraweb.

10. REFERENCES

[1] M. Abolhassani and N. Fuhr. Applying the divergence from randomness approach for content-only search in XML documents. pages 409–419, 2004.

[2] G. Amati and C. J. Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, 20(4):357–389, Oct. 2002.

[3] G. Amati and C. J. van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, 20(4):357–389, October 2002.

[4] T. Beckers, N. Fuhr, N. Pharo, R. Nordlie, and K. N. Fachry. Overview and results of the INEX 2009 interactive track. In *Research and Advanced Technology for Digital Libraries, 14th European Conference, ECDL 2010, Glasgow, UK, September 6-10, 2010. Proceedings*, pages 409–412, 2010.

[5] N. J. Belkin, P. B. Kantor, E. A. Fox, and J. A. Shaw. Combining the evidence of multiple query representations for information retrieval. *Inf. Process. Manage.*, 31(3):431–448, 1995.

[6] C. Benkoussas, H. Hamdan, S. Albitar, A. Ollagnier, and P. Bellot. Collaborative filtering for book recommandation. In *Working Notes for CLEF 2014*

[10] http://www.revues.org/
[11] http://www.openedition.org
[12] http://www.openedition.org/6457
[13] http://www.crossref.org/
[14] http://lucene.apache.org/solr/

Conference, Sheffield, UK, September 15-18, 2014., pages 501–507, 2014.

[7] L. Bonnefoy, R. Deveaud, and P. Bellot. Do social information help book search? In P. Forner, J. Karlgren, and C. Womser-Hacker, editors, *CLEF (Online Working Notes/Labs/Workshop)*, 2012.

[8] H. Bouchard and J.-Y. Nie. ModÃÍles de langue appliquÃl's Ãă la recherche d'information contextuelle. In *CORIA*, pages 213–224. UniversitÃl' de Lyon, 2006.

[9] W. B. Croft. *Organizing and searching large files of document descriptions*. PhD thesis, Cambridge University, 1978.

[10] R. GuillÃl'n. Gir with language modeling and dfr using terrier. In C. Peters, T. Deselaers, N. Ferro, J. Gonzalo, G. Jones, M. Kurimo, T. Mandl, A. PeÃśas, and V. Petras, editors, *Evaluating Systems for Multilingual and Multimodal Information Access*, volume 5706 of *Lecture Notes in Computer Science*, pages 822–829. Springer Berlin Heidelberg, 2009.

[11] K. Järvelin and J. Kekäläinen. Ir evaluation methods for retrieving highly relevant documents. In E. Yannakoudakis, N. Belkin, P. Ingwersen, and M.-K. Leong, editors, *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2000)*, pages 41–48, New York, NY, USA, 2000. ACM.

[12] Y.-M. Kim, P. Bellot, E. Faath, and M. Dacos. Automatic annotation of bibliographical references in digital humanities books, articles and blogs. In G. Kazai, C. Eickhoff, and P. Brusilovsky, editors, *BooksOnline*, pages 41–48. ACM, 2011.

[13] M. Koolen, T. Bogers, J. Kamps, G. Kazai, and M. Preminger. Overview of the INEX 2014 social book search track. In *Working Notes for CLEF 2014 Conference, Sheffield, UK, September 15-18, 2014.*, pages 462–479, 2014.

[14] O. Kurland and L. Lee. PageRank without hyperlinks: Structural re-ranking using links induced by language models. In *Proceedings of SIGIR*, pages 306–313, 2005.

[15] J. H. Lee. Combining multiple evidence from different properties of weighting schemes. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '95, pages 180–188, New York, NY, USA, 1995. ACM.

[16] J. Lin. Pagerank without hyperlinks: Reranking with pubmed related article networks for biomedical text retrieval. *BMC Bioinformatics*, 9(1), 2008.

[17] J. Lin. Pagerank without hyperlinks: Reranking with pubmed related article networks for biomedical text retrieval. *BMC Bioinformatics*, 9(1), 2008.

[18] D. Metzler and W. B. Croft. Combining the language model and inference network approaches to retrieval. *Inf. Process. Manage.*, 40(5):735–750, 2004.

[19] D. Metzler and W. B. 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.

[20] D. Metzler and W. B. Croft. A markov random field model for term dependencies. In R. A. Baeza-Yates, N. Ziviani, G. Marchionini, A. Moffat, and J. Tait, editors, *SIGIR*, pages 472–479. ACM, 2005.

[21] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and C. Lioma. Terrier: A High Performance and Scalable Information Retrieval Platform. In *Proceedings of ACM SIGIR'06 Workshop on Open Source Information Retrieval (OSIR 2006)*, 2006.

[22] I. Ounis, G. Amati, P. V., B. He, C. Macdonald, and Johnson. Terrier Information Retrieval Platform. In *Proceedings of the 27th European Conference on IR Research (ECIR 2005)*, volume 3408 of *Lecture Notes in Computer Science*, pages 517–519. Springer, 2005.

[23] I. Ounis, C. Lioma, C. Macdonald, and V. Plachouras. Research directions in terrier: a search engine for advanced retrieval on the web. *Novatica/UPGRADE Special Issue on Web Information Access*, 2007.

[24] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. In *Proceedings of the 7th International World Wide Web Conference*, pages 161–172, Brisbane, Australia, 1998.

[25] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

[26] V. Plachouras, B. He, and I. Ounis. University of glasgow at trec 2004: Experiments in web, robust, and terabyte tracks with terrier. In E. M. Voorhees and L. P. Buckland, editors, *TREC*, volume Special Publication 500-261. National Institute of Standards and Technology (NIST), 2004.

[27] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proc. SIGIR*, 1998.

[28] S. E. Robertson, C. J. van Rijsbergen, and M. F. Porter. Probabilistic models of indexing and searching. In *SIGIR*, pages 35–56, 1980.

[29] F. Song and W. Croft. A general language model for information retrieval. In *Proceedings of the SIGIR Conference on Information Retrieval*, 1999.

[30] T. Tao, X. Wang, Q. Mei, and C. Zhai. Language model information retrieval with document expansion. In R. C. Moore, J. A. Bilmes, J. Chu-Carroll, and M. Sanderson, editors, *HLT-NAACL*. The Association for Computational Linguistics, 2006.

[31] C. Zhai. *Statistical Language Models for Information Retrieval*. Synthesis Lectures on Human Language Technologies. Morgan and Claypool Publishers, 2008.