Supporting Web Surfers in Finding Related Material in Digital Library Repositories

Jörg Schlötterer, Christin Seifert, and Michael Granitzer

University of Passau,
Innstraße 32, 94032 Passau, Germany
{joerg.schloetterer,christin.seifert,michael.granitzer}@uni-passau.de
http://www.uni-passau.de

Abstract. Web surfers often face the need for additional information beyond the page they are currently reading. While such related material is available in digital library repositories, finding it within these repositories can be a challenging task. In order to ease the burden for the user, we present an approach to construct queries automatically from a textual paragraph. Named entities from the paragraph and a query scheme, which includes the topic of the paragraph form the two pillars of this approach, which is applicable to any search system, that supports keyword queries. Evaluation results point towards users not being able to find optimal queries and needing support in doing so.

Keywords: Just-in-Time Retrieval, Zero Effort Queries, User Study

1 Introduction

Reading a paragraph in a web page often triggers the need for additional information. Then a user has to visit a digital library or general search engine and express this information need as a query in order to retrieve related resources. Proactive retrieval simplifies this process by presenting related material according to the current context without explicit user interaction. This approach was first made popular by Rhodes as Just-in-Time Retrieval [7] and has recently been continued under the topic of zero effort queries [1]. Zero effort queries require minimal, ideally no effort from the user in expressing her information need and obtaining relevant results. However, most of the existing systems either treat the retrieval system as an integral part of the application or focus on domain-specific sets of information needs [9]. We present a proactive retrieval approach that is agnostic of the underlying retrieval system and can be applied to any search system whose contents are searchable via keyword queries. The de-coupling from the retrieval engine is achieved by focusing on the query-side of retrieval: our aim is to construct queries that yield results relevant to the current paragraph.

2 Problem Definition

The aim of our work is to find relevant results for a paragraph of text. More formally, we define the paragraph as P, the query as Q and the result set as

R. The mapping from a paragraph to a query is defined by $h : P \rightarrow Q$ and the retrieval of results by $g : Q \rightarrow R$. We then aim to optimize the function $f = g \circ h : P \rightarrow R$ towards relevant results. Less formal, the whole process is defined as $P \xrightarrow{h} Q \xrightarrow{g} R$. In just-in-time retrieval systems, that treat the search engine as integral part, f is not a composition of g and h, but results are retrieved directly according to the paragraph ($f : P \rightarrow R$). In this paper, we treat the search engine as black box and focus on the query side of retrieval. That is, we have no influence on g, but rather seek to optimize $f = g \circ h$ by optimizing h.

3 Approach

Query Representation P is represented by its sequence of words and Q by a set of keywords, which provide a compact representation of the paragraph. Q can be represented in two principled ways: either as a keyword query or as a boolean query. We use the boolean representation, as it provides richer expressiveness and most digital libraries, which expose their contents via a search API, support boolean queries. In order to avoid over- or underspecified queries, we propose to formulate a boolean query of the following structure:

("main topic") AND ("keyword 1" OR "keyword 2" OR ...)

where the main topic is defined as the overall topic of the paragraph and the right part of the conjunction are additional keywords. This way, we can be sure, that a keyword triggers only results which are connected to the overall topic of the paragraph. Even though, from the perspective of the search engine, all of the query terms are keywords, we will refer to the left part of the conjunction as *main topic* and to the right part as *keywords* in the further course.

Extraction of Keywords Keyword extraction algorithms, that represent the keywords in terms of a subset of terms from the original text are available in the literature [5, 8]. However, query log analysis research revealed, that over 71% of (user generated) search queries contain named entities [2]. In addition, named entities have been shown to be beneficial to query segmentation [3], a technique that is used to optimize queries. Also, named entity extraction can be seen as some kind of keyword extraction task, as the original text is represented by a smaller set of terms. Therefore, we base our query generation on named entities, which are obtained via DBpedia Spotlight[1].

Extraction of the Main Topic To extract the main topic, we utilize Doc2Vec [4]. Based on Word2Vec [6], Doc2Vec produces a word embedding vector, given a sentence or document. Hence, we use the entire input paragraph and compute a vector representation given a Doc2Vec model created on a Wikipedia corpus. Each entity in DBpedia spotlight also corresponds to a Wikipedia page, from which we obtain the Doc2Vec representation of the extracted entities. We compare the Doc2Vec representations of the paragraph and extracted entities by

[1] `http://spotlight.dbpedia.org/`

Table 1. Comparison of optimal, automatic and best user queries.

	own index			original index		
	optimal	automatic	USER*	optimal	automatic	USER*
precision	0.55	0.34	0.37	0.23	0.29	0.33
recall	0.49	0.33	0.37	0.22	0.25	0.33
F1-score	0.49	0.31	0.35	0.21	0.24	0.31

computing the cosine similarity. The named entity with the highest similarity to the input paragraph represents the main topic. The remaining named entities are used as keywords in the right part of the boolean conjunctive query.

4 Evaluation

We evaluated the approach by means of a user study with university students. Due to space constraints, we only report the principle setup of the study[2]. For a given piece of text, i.e. a paragraph, an automatic query was generated by the approach described in the previous section and participants had to rate the results. Then, participants were asked to adapt the query, in order to retrieve more relevant results. The evaluable study data consisted of 251 paragraphs and 558 associated queries, performed by 69 users.

The collection of a repository is subject to change (new items may be added or the ranking may change), which would render future comparability of the results infeasible. To counter for this fact, we set up an Elasticsearch[3] index with the results retrieved during the study. On this index, we collected the optimal queries, that can be posed, based on the extracted main topic, keywords and query scheme as defined in Sec. 3. In principle, we collected those queries with a brute-force approach, testing all possible combinations of keywords and choosing the best performing in terms of F1-score as optimal query.

Results We evaluated the query quality of our approach for automatic queries with all extracted keywords and the structure described in Sec. 3 (automatic), the optimal queries and the best queries, users were able to formulate (USER*) on the original data and our own index with the results depicted in Table 1. The reported precision-, recall- and F1-scores are macro averaged over all queries.

For the evaluation of user queries, we took the best query, a user was able to formulate for a paragraph (USER*). This means, that if the initial automatic query scored better in terms of F1-score than all subsequent modifications by the user, we take the initial query. As can be seen from the table, the performance of automatic and user queries (USER*) is quite low, with the user generated

[2] The setup is documented in the project repository `https://github.com/schloett/p2q` alongside with further material, like the log files collected during the user study

[3] `https://www.elastic.co/products/elasticsearch`

queries performing slightly better (0.35) than the automatic queries (0.31). If we consider, that automatic queries are restricted to the extracted main topic and keywords, while users can provide arbitrary values for these two, the automatic queries still perform quite well. As one would expect, the performance on our own index is higher, as the amount of potential false positives is reduced. Also, the performance of optimal queries (0.21) on the original index is lower than on our custom index (0.49) (and even lower than the performance of automatic queries and USER*), since the optimal queries trigger results, which are not contained in the original result sets. Hence, as we do not have relevance feedback for those results, they are treated as irrelevant, even though they may be relevant in fact.

5 Summary

In this paper, we presented a search-engine agnostic approach for the automatic generation of boolean queries from a paragraph, based on named entities. We evaluated the performance against optimal achievable queries and the best queries, users were able to formulate. Results indicate, that users are able to formulate better queries than the automatic approach, but still below the optimal achievable performance and hence need support in query formulation.

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.

References

1. Allan, J., Croft, B., Moffat, A., Sanderson, M.: Frontiers, challenges, and opportunities for information retrieval: Report from SWIRL 2012. SIGIR Forum 46(1), 2–32 (May 2012)
2. Guo, J., Xu, G., Cheng, X., Li, H.: Named entity recognition in query. In: SIGIR '09. pp. 267–274. ACM (2009)
3. Hagen, M., Potthast, M., Beyer, A., Stein, B.: Towards optimum query segmentation: In doubt without. In: CIKM '12. pp. 1015–1024. ACM (2012)
4. Le, Q.V., Mikolov, T.: Distributed representations of sentences and documents. In: ICML '14. pp. 1188–1196 (2014)
5. Mihalcea, R., Tarau, P.: Textrank: Bringing order into texts. In: EMNLP '04 (2004)
6. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NIPS '13, pp. 3111–3119. Curran Associates, Inc. (2013)
7. Rhodes, B.J.: Just-In-Time Information Retrieval. Ph.D. thesis, Massachusetts Institute of Technology (2000)
8. Rose, S., Engel, D., Cramer, N., Cowley, W.: Automatic Keyword Extraction from Individual Documents, pp. 1–20. John Wiley & Sons, Ltd (2010)
9. Shokouhi, M., Guo, Q.: From queries to cards: Re-ranking proactive card recommendations based on reactive search history. In: SIGIR '15. pp. 695–704 (2015)