SAC:G:From Context to Query

Jörg Schlötterer
University of Passau
Innstrasse 33a, Passau, Germany
joerg.schloetterer@uni-passau.de

ABSTRACT

Web surfers often face the need for additional material beyond the page, they are currently reading. To support web surfers in finding related material, we present a proactive retrieval approach, which is applicable to any search system that supports keyword queries. Our approach identifies the relevant context, detects an information need and automatically constructs a query to express this information need. We show, that with the presented approach, automatic queries can be constructed, which perform better in terms of precision (0.40) than the best queries, users are able to formulate (0.37).

1. PROBLEM AND MOTIVATION

Reading 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. In accordance with the user-based information seeking model of Marchionini & White [10], the required steps to turn the above process into automatic, proactive retrieval consist of:

1. *identifying the relevant context*
2. *recognizing an information need*
3. *expressing this information need* (as query to the retrieval system)

We address all of these three steps, with a special emphasis on the third step, i.e. automatically constructing a query.

2. BACKGROUND AND RELATED WORK

Proactive retrieval was first made popular by Rhodes as Just-in-Time Retrieval [14]. Recently, Rhodes' research has been continued under the topic of zero effort queries [1] with special emphasis on mobile applications [8]. Zero effort queries require minimal, ideally no effort from the user in expressing her information need and obtaining relevant results. While earlier work [15, 9, 2] focused on document retrieval, a wide variety of content and media types is taken into account in more recent work [20]. However, most of the systems either treat the retrieval system as an integral part of the application or focus on domain-specific sets of information needs.

We present an approach, that is agnostic of the underlying retrieval system. 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 context. More formally, we want to optimize the function $f = g \circ h : C \to R$, where C is the context and R the result set, towards relevant results. The mapping from the context C to a query Q is defined by $h : C \to Q$ and the retrieval of results by $g : Q \to R$. Less formal, the whole process is defined as $C \xrightarrow{h} Q \xrightarrow{g} R$. 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 context ($f : C \to R$). In this paper, we treat the search engine as black box and focus on the query side of retrieval. This means, we have no influence on g, but rather seek to optimize h in terms of optimizing $f = g \circ h$.

3. APPROACH AND UNIQUENESS

The unique feature of our approach is treating the retrieval engine as a black box - the key factor is constructing a query, which yields results relevant to a piece of text. While we describe the approach for a web setting, the core of finding a good query is equally applicable in any scenario, where related material to a piece of text is desirable. This material can be of any kind, such as scientific documents, cultural images or selling products, as long at is exposed via a search engine.

In a web setting, a browser extension allows to proactively present resources to users, while observing the user's context and adapting resources to the respective context [17]. In this setting, the observable context dimensions encompass first of all the web pages visited and in addition information like the user's location. As a proof of concept, we developed a mobile application, which integrates multiple context dimensions [18], but focus on the textual content of web pages in this paper. This textual context can be subdivided into five levels of granularity (from fine-grained to coarse): *terms*, *phrases*, *paragraphs*, *pages* and *sessions*.

Table 1 provides an overview of our approach for each of the steps, from identifying the relevant context to recognizing and expressing an information need, on the different levels of context granularity. In the following, we briefly sketch each step on every level and describe the expression of the information need on paragraph level in detail. We omit the term level, as it is captured by our approach for the phrase level by treating terms as single term phrases. While we conduct proactive retrieval on the phrase and paragraph level,

Table 1: Overview on steps 1-3 for each contextual granularity level. Proactive retrieval is conducted on the phrase and paragraph level, while the page and session level serve as supporting input for the lower levels.

context granularity	detection method	information need detection	information need formulation	information need representation
phrase	text selection	TRUE (for selection)	CRF Model	terms
paragraph	web browser focus area	topic overlap with user profile	NER + filtering keyword detection	terms + entities
page	NONE	topic overlap with user profile	main topic extraction	entity
session (sequence of pages)	topic similarity navigation patterns	session clusters	main topic extraction	entities
(1) *identify the relevant context*		(2) *recognize an information need*	(3) *express this information need*	

we utilize the outcome of the more coarse-grained page and session level as supporting features for the aforementioned levels.

3.1 Phrase Level

Relevant context identification: The most accurate way to identify the phrase currently read by the user is eye tracking - browser events, such as mouse movements or scroll position yield only limited accuracy [6]. Thus, we rely on explicit user interaction in this case, i.e. a text selection, which is a strong indicator for reading focus [6].

Information need recognition: Given a text selection, we assume an information need implicitly.

Information need expression: To gather ground truth data, we conducted an experiment, in which we had users select arbitrary pieces of text in web pages and issue queries to find resources relevant to that selection. It turned out, that most terms in the users' queries were already contained in the corresponding text selection [19]. This means, that the necessary information to construct a query is already contained in the selection and we only need to extract the relevant terms. To predict the relevant terms, we apply a linear chain conditional random field model (CRF). This model predicts a sequence of labels for a given input sequence, conditioned on the input features, i.e., it models the probability of the output variables (the labels) conditioned on the observed variables (the input features). We opted for a CRF, as it is advantageous over a Hidden Markov Model in efficiently accounting for dependencies among input features. The results are depicted in section 4.2.

3.2 Paragraph Level

Relevant context identification: To determine the relevant paragraph in a web page (i.e. the user's focused paragraph), two steps need to be accomplished: First, actual text passages (i.e., the paragraphs) need to be separated from navigational menus, advertisements, etc. Second, the paragraph in focus needs to be determined.

For the first step, we follow Cai et al. [3] in terms of a tag-tree independent approach. Opposed to them, we are not interested in the actual visual building blocks of a page, but only in the textual paragraphs. Hence, in favor of minimizing the computational effort, we apply a heuristic, which is based on a fixed length threshold of DOM text nodes and separates paragraphs from other page elements. For the second step, viewport, scroll position and mouse movements serve as indicators to determine the focused paragraph [6].

Information need recognition: We analyze the topical coherence of the paragraph and the user profile, to figure out, if the user is familiar with the topic of the paragraph or not. The latter reveals an information need.

Information need expression: A paragraph P is represented by its sequence of words and the corresponding query Q by a set of keywords, which provide a compact representation of the paragraph. Furthermore, 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 retrieval systems, which expose their contents via a search API, support boolean queries. In addition, boolean queries can be easily transformed into keyword queries, while the opposite is not possible in general. Combining all keywords with either OR or AND yields under- or over-specified queries, in particular, when the set of keywords grows. While the problem of over-specified queries is obvious (no results), the problem of disjunctive queries is less obvious: results, which are triggered by a single keyword only, may not fit the topic of the paragraph very well and hence are quite unrelated. Moreover, results which are triggered by a single keyword can suppress results that are related to several keywords. The conjunctive normal form (CNF) provides means to formulate highly precise queries, which are not over-specified though, and hence yield results. However, finding the optimal query for arbitrary combinations of keywords in CNF is NP-hard and hence intractable in general. Therefore, we propose to formulate a boolean query in CNF 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. In the following, we describe how both parts, i.e., the main topic and the keywords, of the proposed query structure can be extracted and how the resulting query can be optimized.

Extraction of Keywords.

Keyword extraction algorithms, that represent the keywords in terms of a subset of terms from the original text are readily available in the literature [11, 16]. However, query log analysis research revealed, that over 71% of (user generated) search queries contain named entities [4]. In addition, named entities have been show to be beneficial to query segmentation [5], 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]. However, this may miss some important terms to construct a good query. Therefore, we propose to gather additional keywords in a similar fashion as for the phrase level (c.f. section 3.1) and combine them with the extracted named entities.

Extraction of Main Topic.

To extract the main topic, we utilize Doc2Vec [7]. Based on Word2Vec [12], Doc2Vec produces a vector, given a sentence or document. Hence, we use the entire input paragraph and infer a vector representation given a Doc2Vec model created on a Wikipedia corpus. We compare this vector with the Doc2Vec representations of all named entities extracted from the paragraph via DBpedia spotlight (i.e. also Wikipedia pages) by computing the cosine similarity. We restrict the similarity computation to the extracted named entities, instead of all possible entities contained in Wikipedia, for performance reasons. 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.

Selection of Keywords.

Having extracted the keywords and main topic, the baseline approach to a query in the defined CNF is to use all of the extracted named entities as keywords. However, this may still yield some irrelevant results, not least as the named entity extraction is not perfectly accurate. The optimal query can be obtained with a brute-force approach, testing all keyword combinations. Knowing the optimal query, we know the optimal selection of keywords to use in this query. Based on this knowledge, we can train a classifier, that predicts whether a keyword should be used in the query or not.

3.3 Page Level

Relevant context identification: By design, only a single page can be the *active* page in a browser window. We consider the *active* page relevant, even though this may not be true in the (rare) situation of a split screen with several browser windows or tabs. But again, an eye tracker would be required to cover such scenarios perfectly.

Information need recognition: The decision whether to present additional resources is based on the topical coherence of the page and the user profile. As mentioned before, the information need recognition on page level does not directly trigger the retrieval of results, but serves as an additional indicator for the paragraph level. This decision was taken since a page always exhibits at least one paragraph, and the paragraph provides more fine-grained context.

Information need expression: The information need on page level is not directly expressed as a query, but serves as input feature for the information need expression on paragraph level. Our evaluation revealed, that the page topic is better suited as main topic in the query than the paragraph topic (c.f. section 4.4). Its extraction is analogous to the paragraph topic, described in section 3.2.

3.4 Session Level

Relevant context identification: The first indicator for session detection is the topical coherence of subsequently visited pages. In some cases, this is not sufficient. For example the pages of a ”reading online news” session may have diverse topics, but still belong to the same session. Preliminary experiments indicated that a small set of recurring sessions (such as the just mentioned ”reading online news”) constitute the main part of a user's browsing behaviour. This hypothesis is supported by the finding that few sites account for the majority of visits in a user's browsing history [13]. Therefore, we cluster the user's visited pages by their frequency, in order to identify features of recurring sessions.

Information need recognition: Recurring sessions typically do not exhibit an information need per se (but still might exhibit an information need on page level or below), as those are sessions such as ”visiting institutional pages”. Hence, we neglect recurring sessions and focus on rare ones. Indicators for an information need in the latter case are visits of a search engine in between other pages or textual input to a search form field on an arbitrary page.

Information need expression: The main topics from previously visited pages within a session can be used to expand the query on paragraph level, i.e. they provide additional keyword candidates for the right part of the query structure described in section 3.2.

4. RESULTS AND CONTRIBUTIONS

We conducted several user studies to evaluate the individual steps of the proposed approach. This section presents some of the results.

4.1 Identification of the Focused Paragraph

In a study with 77 participants, we evaluated the extraction and identification of the focused paragraph, as described in section 3.2. Therefore, participants were instructed to navigate to particular sections of Wikipedia pages. The extracted paragraphs of the page were highlighted and the paragraph identified as focused was pre-selected. Participants were then asked to adapt this pre-selection, if it was not correct. If the extraction of paragraphs was correct, but the identification of the focused paragraph failed, they could select another paragraph from the extracted ones. If already the extraction of paragraphs failed, they could indicate the correct paragraph with a text selection. Participants modified only 16% of the paragraphs extracted by our approach, i.e. 84% of the paragraphs were extracted correctly or meaningful from a user perspective. From the correctly extracted paragraphs, the focused paragraph was identified correctly in 65% of the cases. These results are quite promising, in particular, as they are based on a simple heuristic for performance reasons and browser events yield only limited accuracy. Even though these results are only valid for Wikipedia pages, we expect similar values for other pages, since the heuristic was not specifically tuned towards

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

Table 2: Accuracies [%] for query prediction from selected text. Cross-validated using splits over users, pages, and 10-fold random.

		feature set			trivial	
		i, c, t	i, t	c, t	rejector	acceptor
users	mean	76	77	75	51	49
	SD	15	15	18	35	35
pages	mean	82	83	82	71	29
	SD	6	6	7	8	8
10-fold	mean	89	88	84	71	29
	SD	1	2	1	2	2

i - the identity of a term, i.e. the term itself
c - whether the term begins with upper- or lowercase
t - POS tag

Wikipedia. We will evaluate the performance on arbitrary pages in a live system.

4.2 Query Construction - Phrase Level

As mentioned in section 3.1, in the majority of queries corresponding to a text selection, query terms are already contained in the selected text. On a set of 2449 selection-query pairs, we evaluated the performance of 29 feature combinations on a linear chain Conditional Random Field (CRF) model. Among the evaluated input features were for example the term itself (i), the part-of-speech tag (t), an indicator, whether the term is a stopword (s), an indicator, whether the term starts with upper- or lowercase (c), etc. The best performing feature combinations are shown in table 2. As the CRF model assigns a label to each term in the selection (identifying it as relevant or not relevant), accuracy is the ratio of correctly labeled terms to the total number of terms.

The standard deviations reveal, that the query behavior is stable over pages, but not over users. In fact half of the users incorporated the major part of the selection into their queries and the queries of the other half contained only a minority of the selection terms. Thus, prediction performance drops for the evaluation over users.

Incorporating a term itself as a feature (i, c, t & i, t) leads to the best results, but this may not generalize well due to the limited vocabulary in the dataset. Nevertheless, feature combinations without the words provide similar results as well (e.g., the combination of case-identifier and POS-tag, c, t) and thus are the better option.

4.3 Query Construction - Paragraph Level

In the same study as mentioned in section 4.1, after identification of the focused paragraph (and a potential refinement by the user), an automatic query was generated with the approach described in section 3.2 (main topic and keywords - MTAK). Users then had to rate the results of this query. Afterwards, they were asked to adapt the query in order to retrieve better results until they were either satisfied with the results, a timing constraint was met or they were sure that the content collection of the search engine does not contain relevant results. At most, the top-10 results were retrieved via the search APIs of Mendeley and Europeana and interleaved via Round-Robin.

With the results retrieved during the study, we populated an own ElasticSearch index, on which further evaluations were carried out. With a brute-force approach, we determined the optimal queries achievable, based on the extracted keywords. Therefore, if the set of candidate (i.e. extracted) keywords was sufficiently small, we ran queries with all possible combinations of candidate keywords against our index and chose the best performing query in terms of F1-score as optimal query (MTBK). Otherwise, we selected promising candidates and evaluated all their combinations. The threshold for the maximum candidate set size was set to 20, as at most 20 results were retrieved. Hence, every candidate can at most contribute one unique result and contributions of further candidates would not show up in the list. Accordingly, the main principle of the selection process was to remove candidates, which did not yield additional positive results, compared to the results retrieved by the remaining candidates. Still, the combinations to test amount to $\sum_{k=0}^{n} \binom{n}{k}$, where n is the number of candidates and k is the size of the combination set to be tested. This equation is equal to 2^n and over one million combinations to test for a set of 20 candidates. The set of optimal queries was then used to train a decision-tree classifier, in order to predict, which of the extracted keywords should be used in a query (MTPK). For the user generated queries, we considered only the best query, a user was able to formulate (USER*): if the initial automatic query outperformed all subsequent modifications by the user, it was chosen as the best query a user could formulate.

Table 3: Performance of MTBK, MTAK, MTPK and USER* on our own index.

	MTBK	MTAK	MTPK	USER*
precision	0.55	0.34	0.40	0.37
recall	0.49	0.33	0.33	0.37
F1-score	0.49	0.31	0.33	0.35

The performance of the four query construction techniques is depicted in table 3. Considering, that MTAK is restricted to the extracted main topic and keywords, while users can provide arbitrary values for these two, MTAK already performs surprisingly well (0.31) compared to the users (0.35). In terms of precision, the predicted queries (MTPK) perform even better (0.40) than the best queries users were able to formulate (USER*, 0.37). Favoring precision over recall, since we aim to present only relevant results to the user and not bother her with irrelevant results, this is a remarkable result, in particular, as users had more freedom in query formulation.

We also trained a CRF model in a similar way as described in section 4.2, yielding a macro-averaged accuracy of 0.90 over a 10-fold cross-validation random split. We plan to combine the terms extracted by this model with the extracted named entities, in order to increase the recall.

4.4 Main Topic Extraction

In order to determine, how well the extracted main topic is suited as main topic in the query, we evaluate, how often users modified the extracted main topic in the automatically generated queries mentioned in the previous section. When evaluating the modifications, we need to consider two factors: (i) modifications, which do not result in an improvement and (ii) non-modifications, for which we do not know the reasons. The latter occurs, when the automatic query is not followed by a user query and the user did not provide a reason for it, or the reason was a time out. Another reason for not modifying the query is, that the user is perfectly satisfied with the results, but in this case, we can conclude that the main topic is well suited. If we take the two factors into account, the suggested main topic is appropriate for 79% of the queries.

As the main topic is represented by a Wikipedia article title and the evaluation was carried out on Wikipedia pages, we can easily compare the main topic extracted from the paragraph with the topic of the page. In 41% of the queries, both were the same. Moreover, the suggested main topic was different from the page topic in 81% of the queries where the main topic was changed and the change resulted in an improvement in 63%. Also, the extracted keyword with the page topic was set as new main topic for 25% of the queries. These findings suggest, that in general, the topic of the whole page is well suited to be used as main topic in the query. Therefore, under the assumption that the main topic is extracted correctly, main topic extraction should be based on the whole page, rather than the focused paragraph.

4.5 Contributions

We proposed a search engine agnostic just-in-time retrieval approach. Therefore, we described how the steps from identification of the relevant context to construction of a query can be accomplished on different levels of context granularity. In particular, we introduced a new query model to construct queries, which yield additional resources for a paragraph. We further provide means to obtain the individual components of this model and show that queries can be constructed which perform better in terms of precision than the best queries, users are able to formulate.

5. 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.

6. REFERENCES

[1] J. Allan, B. Croft, A. Moffat, and M. Sanderson. Frontiers, challenges, and opportunities for information retrieval: Report from SWIRL 2012. *SIGIR Forum*, 46(1):2–32, May 2012.

[2] J. Budzik and K. Hammond. Watson: Anticipating and contextualizing information needs. In *62nd annual meeting of the american society for information science*, pages 727–740, 1999.

[3] D. Cai, S. Yu, J.-R. Wen, and W.-Y. Ma. Extracting content structure for web pages based on visual representation. In *Proc. of the 5th Asia-Pacific Web Conf. on Web Technologies and Applications*, APWeb'03, pages 406–417, Berlin, Heidelberg, 2003. Springer-Verlag.

[4] J. Guo, G. Xu, X. Cheng, and H. Li. Named entity recognition in query. In *Proc. of the 32nd Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, SIGIR '09, pages 267–274. ACM, 2009.

[5] M. Hagen, M. Potthast, A. Beyer, and B. Stein. Towards optimum query segmentation: In doubt without. In *Proc. of the 21st ACM Int. Conf. on Information and Knowledge Management*, CIKM '12, pages 1015–1024. ACM, 2012.

[6] D. Hauger, A. Paramythis, and S. Weibelzahl. Using browser interaction data to determine page reading behavior. In *UMAP'11*, pages 147–158. Springer-Verlag, 2011.

[7] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*, 2014.

[8] R. Lee and K. Sumiya. Zero-effort search and integration model for augmented web applications. In *Proc. of the 9th Int. Conf. on Web Engineering*, ICWE '9, pages 330–339. Springer-Verlag, 2009.

[9] H. Lieberman. Autonomous interface agents. In *Proc. of the ACM SIGCHI Conf. on Human Factors in Compu. Sys.*, CHI '97, pages 67–74. ACM, 1997.

[10] G. Marchionini and R. White. Find what you need, understand what you find. *Int. J. Hum. Comput. Interaction*, 23(3):205–237, 2007.

[11] R. Mihalcea and P. Tarau. Textrank: Bringing order into texts. In *Proc. of the Conf. on Empirical Methods in Natural Language Processing*, 2004.

[12] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

[13] H. Obendorf, H. Weinreich, E. Herder, and M. Mayer. Web page revisitation revisited: Implications of a long-term click-stream study of browser usage. In *CHI '07*, pages 597–606. ACM, 2007.

[14] B. J. Rhodes. *Just-In-Time Information Retrieval.* PhD thesis, Massachusetts Institute of Technology, 2000.

[15] B. J. Rhodes and P. Maes. Just-in-time information retrieval agents. *IBM Syst. J.*, 39(3-4):685–704, July 2000.

[16] S. Rose, D. Engel, N. Cramer, and W. Cowley. *Automatic Keyword Extraction from Individual Documents*, pages 1–20. John Wiley & Sons, Ltd, 2010.

[17] J. Schlötterer, C. Seifert, and M. Granitzer. Web-based just-in-time retrieval for cultural content. In *PATCH '14: Proc. of the 7th International ACM Workshop on Personalized Access to Cultural Heritage*, 2 2014.

[18] J. Schlötterer, C. Seifert, W. Lutz, and M. Granitzer. From context-aware to context-based: Mobile just-in-time retrieval of cultural heritage objects. In *Advances in Information Retrieval - 37th European Conf. on IR Research, ECIR 2015, Vienna, Austria, March 29 - April 2, 2015. Proc.*, pages 805–808, 2015.

[19] C. Seifert, J. Schlötterer, and M. Granitzer. Towards a feature-rich data set for personalized access to long-tail content. In *SAC '15*. ACM, 2015.

[20] M. Shokouhi and Q. Guo. From queries to cards: Re-ranking proactive card recommendations based on reactive search history. In *Proc. of the 38th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, SIGIR '15, pages 695–704, 2015.