Webis at the TREC 2011 Sessions Track

Matthias Hagen , Jan Grassegger , Maximilian Michel, and Benno Stein

Bauhaus-Universität Weimar
99421 Weimar, Germany
<first name>.<last name>@uni-weimar.de

Abstract In this paper we give a brief overview of the Webis group's participation in the TREC 2011 Sessions track with an extended version of our last year's approach [HSV10]. The basic idea can be described as a conservative query expansion based on terms used in previous queries or terms contained in clicked snippets. Furthermore, a query's result set is reduced by removing documents shown for previous queries or documents containing important terms from non-clicked snippets.

1 Introduction

The TREC 2011 Sessions track in its second year offered the opportunity to apply techniques for user experience improvement during a web *search session*—the set of consecutive queries submitted for the same information need. The main idea of our approach for this track is inspired by assuming the following interaction scheme during such sessions. The user comes up with a set of (in her opinion) appropriate keywords—or keyphrases—for a given information need. She submits a query containing some of these keywords and gets back a ranked result list. If the user does not find a match for her information need among the first results, she will hardly browse all the items but submit different queries based on her keywords until she is satisfied or decides to give up.

The track itself had the task to improve retrieval performance for a given query by using knowledge of the user's previous queries and interactions (e.g., clicked documents and dwell times). The task design had four steps which increased the available knowledge of the previous interactions: (1) only the last query string given, (2) additionally given the strings of the previous queries from the session, (3) additionally the top-10 results with snippets for the previous queries given, (4) additionally clicked results and dwell times given for the previous queries.

With this increased knowledge our framework also evolves in four steps: (1) query used as is, (2) a promising query is formulated from the keywords of all queries (basically this is a query expansion of the last query with only terms from previous queries), (3) additionally two keyphrases extracted from the combination of the shown snippets are used to exclude documents that contain these keywords (here, we assume that the user did not click any documents because the seen snippets indicated irrelevance of the shown results), (4) two extracted keyphrases from the combination of the clicked snippets are used to further expand the promising query and from the non-clicked snippets two keywords are extracted that were treated as in case (3) .

Our underlying assumption for using the promising query perspective is that such a query containing as many of the user's keywords as possible, while returning a reasonable number of results, best describes the user's information need. The rationale for requiring a reasonable number of results per query deserves closer consideration. Queries with empty result pages are useless and the same often applies to queries returning only a handful of hits. This gives a lower bound on the number of desired results. But there is also an upper bound since the number of results a user will consider for a single query is usually constrained by a processing capacity k, determined by the user's reading time etc. If the user faces a query with millions of hits, she can only check a fraction of the results—typically the top-ranked ones. Relevant entries below are missed. Based on the User-over-Ranking hypothesis [SH11], we argue that the best queries are the ones that are sufficiently specific to not return millions of hits—but also not just one or two. For such queries the user can check the complete result list and will not miss any potential match for her information need due to search engine ranking issues that she cannot influence. Hence, from the user's perspective, a promising query contains a possible description of the information need and offers the chance to check all the results. Previous experiments for the PROMISING QUERY framework showed that such queries perform well in TREC style experiments [SH11] and that they might be a tool to support users stuck in search sessions [SH10,HS11]. Unfortunately, the results of our runs on this year's TREC Sessions track suggest that the PROMISING QUERY framework as we applied it basically got worse results with increased session knowledge compared to less knowledge.

The paper is organized as follows. In Section 2, we describe the basic retrieval systems underlying our three runs. The applied query formulation and result set post-processing are explained in Section 3. Achieved experimental results of our systems are given in Section 4. A discussion and some concluding remarks follow in Section 5.

2 Retrieval system

One of our three runs is based on using the Indri search engine for the ClueWeb that is provided by the Carnegie Mellon University[1]. Our other two runs use our own ClueWeb search engine called Chat Noir (French for black cat). Chat Noir is based on the BM25F retrieval model [RZT04] (including the anchor text list provided by the University of Twente[2] and the PageRank list provided by the Carnegie Mellon University[3]). For all three runs we removed results from the ranked lists that have spam ranks smaller than 70% (meaning that at most 70% of the ClueWeb have a higher probability of being spam) according to the spam rank list provided by the University of Waterloo[4]. Thus we only return results from a 30% fraction of the ClueWeb that have the lowest "probability" of being spam pages.

[1] http://boston.lti.cs.cmu.edu:8085/clueweb09/search/

[2] http://wwwhome.cs.utwente.nl/~hiemstra/2010/anchor-text-for-clueweb09-category-a.html

[3] http://boston.lti.cs.cmu.edu/clueweb09/wiki/tiki-index.php?page=PageRank

[4] http://durum0.uwaterloo.ca/clueweb09spam/

We did not further tune the engines used as we are not primarily interested in designing a system with the best performance for single queries. Our main interest is whether our ideas for improving the retrieval via session knowledge have any positive effect with the above fairly standard retrieval models.

3 Query processing

In the real retrieval phase we treat the search engines of the different runs as black boxes and just work with the results lists.

3.1 The baseline run

Our baseline run uses our own search engine Chat Noir. As for RL1, the query is processed as is. As for RL2 we adopt the PROMISING QUERY framework as follows. The final query is expanded by keywords from previous queries of the same session as long as the number of results stays above 100. This can be viewed as kind of a conservative query expansion method that trusts the user's keywords and knowledge of the topic more than automatically finding good expansion terms.

RL3 of our basic run uses the same query as RL2 but removes all results shown in the top-10 for the previous queries. The rationale for this is the assumption that the user has already seen all these results and judged them as not relevant.

As for RL4, we use the query of RL2 and further expand it by two head noun phrases extracted from the concatenated text of all the clicked snippets of previous queries. The head nouns were extracted using the system of Barker and Cornacchia [BC00]. The rationale is that these head nouns were contained in previously clicked results such that we assume that the user judges results with these head nouns as highly relevant.

We also extract two head noun phrases from the concatenation of all the not-clicked results of the previous queries and post-processed the result list of the query for RL4 by removing all documents that contain the two head noun phrases of the not-clicked results. The rationale here is that we assume that the user judged snippets containing these head nouns as not relevant. We ensured that the extracted head nouns for clicked and not-clicked snippets did not overlap.

3.2 Weighting and improved post-processing

Our second run also uses our own search engine Chat Noir. As for RL1, the query is processed as is. The queries for RL2–RL4 are similar to the ones of the baseline run but with term weighting: (1) terms appearing in the current query and also in previous queries get a basis weight of 1.0, (2) terms appearing only in the current query get a doubled weight of 2.0, (3) terms appearing only in previous queries get a weight of only 0.5. These weights are meant to reflect the importance of the terms for the current query—terms from the current query are the most important and terms only present in previous queries are the least important.

3.3 Indri as the search engine

Our final run uses the Indri ClueWeb search engine provided by the Carnegie Mellon University. We used the same queries and term weighting we described for our second run.

4 Evaluation

The evaluation for the Sessions track is done by comparing the four ranked lists with respect to several retrieval performance measures. Our runs' nDCG@10 performances are given in Table 1.

Table 1. Results for **nDCG@10** averaged over all 76 topics with all subtopics as the relevance criterion.

	RL1	RL2	RL3	RL4
Our baseline Char Noir run	0.2291	0.1808	0.1863	0.1624
Our weighted Chat Noir run	0.2386	0.2021	0.1919	0.1758
Our Indri run	0.3207	0.2487	0.2473	0.2567
Median of all runs	0.3056	0.3106	0.3084	0.3263
Maximum of all runs	0.3663	0.4061	0.4086	0.4320

As can be seen from Table 1, our best run's performance is worse than the median of all runs. However, what is even more important: none of our runs benefit from increased session knowledge. Further investigating the performance per individual session, our PROMISING QUERY framework based runs only improve retrieval performance in around 25 cases, compared to the simple query from RL1, while for the other 50 sessions the performance is decreased.

One possibility to circumvent this behavior of our system could be to check in a preprocessing step whether our framework might be able to improve the retrieval performance. An idea could be to apply some session detection approach like [HSR11] and check whether all the queries from a TREC session are also assigned to a session by the detection method or whether only some of the previous queries would be assigned to the same session as the last query. In the latter case, the last query could only be expanded in a PROMISING QUERY manner with terms from the queries assigned to the same session and not with terms from all previous queries given in the TREC session. This might improve the overall retrieval performance as then the results could be more focused on the real intent of the last query and the related queries submitted before. We plan to further investigate this possibility in future work.

5 Discussion

As can be seen from the evaluation, our best run performs worse than the median of all systems. As we were not primarily interested in the best overall system our focus

is on the performance for the different steps of available session knowledge. Unfortunately, on average our ideas seem to decrease performance: the PROMISING QUERY framework in the form we applied it to this year's Session track sessions seems not applicable to increase retrieval performance via session knowledge.

An idea for improvement could be to check (in a preprocessing step) whether the PROMISING QUERY framework might be able to improve the retrieval performance. This could for instance be done via some session detection approach like [HSR11]. The query expansion could then only involve terms from the session that the detection method outputs and not with all queries from the provided TREC session.

An additional idea could be to also apply query segmentation [HPSB11] to the queries in order to increase the system performance via phrase based search.

References

[BC00] Ken Barker and Nadia Cornacchia. Using noun phrase heads to extract document keyphrases. In *Advances in Artificial Intelligence, 13th Biennial Conference of the Canadian Society for Computational Studies of Intelligence, AI 2000, Montréal, Quebec, Canada, May 14-17, 2000, Proceedings*, pages 40–52, 2000.

[HPSB11] Matthias Hagen, Martin Potthast, Benno Stein, and Christof Bräutigam. Query Segmentation Revisited. In Sadagopan Srinivasan, Krithi Ramamritham, Arun Kumar, M. P. Ravindra, Elisa Bertino, and Ravi Kumar, editors, *20th International Conference on World Wide Web (WWW 11)*, pages 97–106. ACM, March 2011.

[HS11] Matthias Hagen and Benno Stein. Applying the User-over-Ranking Hypothesis to Query Formulation. In *Advances in Information Retrieval Theory. Third International Conference on the Theory of Information Retrieval (ICTIR 11)*, volume 6931 of *Lecture Notes in Computer Science*, pages 225–237. Springer, 2011.

[HSR11] Matthias Hagen, Benno Stein, and Tino Rüb. Query Session Detection as a Cascade. In Bettina Berendt, Arjen de Vries, Wenfei Fan, Craig Macdonald, Iadh Ounis, and Ian Ruthven, editors, *20th ACM International Conference on Information and Knowledge Management (CIKM 11)*, pages 147–152. ACM, 2011.

[HSV10] Matthias Hagen, Benno Stein, and Michael Völske. Webis at the TREC 2010 Sessions Track. In Ellen M. Voorhees and Lori P. Buckland, editors, *19th International Text Retrieval Conference (TREC 10)*. National Institute of Standards and Technology (NIST), 2010.

[RZT04] Stephen Robertson, Hugo Zaragoza, and Michael Taylor. Simple BM25 extension to multiple weighted fields. In *CIKM'04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 42–49, New York, NY, USA, 2004. ACM Press.

[SH10] Benno Stein and Matthias Hagen. Making the Most of a Web Search Session. In *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT 10)*, pages 90–97, Los Alamitos, CA, USA, August 2010. IEEE.

[SH11] Benno Stein and Matthias Hagen. Introducing the User-over-Ranking Hypothesis. In *Advances in Information Retrieval. 33rd European Conference on IR Resarch (ECIR 11)*, volume 6611 of *Lecture Notes in Computer Science*, pages 503–509, Berlin Heidelberg New York, April 2011. Springer.