Webis at the TREC 2012 Session Track

Matthias Hagen, Martin Potthast, Matthias Busse,
Jakob Gomoll, Jannis Harder, 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 2012 Session track. Our runs focus on three research questions: (1) distinguishing low risk sessions where we want to involve session knowledge from those where we don't, (2) examining conservative query expansion (only few expansion terms based on keywords from previous queries and seen/clicked documents/titles/snippets), and (3) incorporating knowledge from other users' similar sessions. Altogether, especially similar sessions seem to help improving retrieval performance in our experiments.

1 Introduction

The TREC 2012 Session track in its third year again focused on techniques for user experience improvement during a web *search session*—the set of queries submitted for the same information need. The underlying assumption of the track is 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, if some "sub"-information need remains open, or even if some new need evolved during studying the first results, she will hardly browse all the items in the ranked list of the very first query but instead submit different queries until she is satisfied or decides to give up. The idea then is to use the observable interaction scheme (e.g., clicked documents and dwell times) to gain knowledge about what the user is looking for and to apply this knowledge to help improve the retrieval for the final query. The task design had four steps which increased the available knowledge of the previous interactions: (RL1) only the last query string is given, (RL2) additionally the strings of the previous queries from the session are given, (RL3) additionally the top-10 results with snippets for the previous queries are given, (RL4) additionally clicked results and respective dwell times for the previous queries are given.

With this increased knowledge our framework also evolves in four steps: (RL1) query used as is, (RL2) keywords from previous queries as potential query expansion candidates, (RL3) additionally keywords from the shown snippets, titles, and the whole documents as expansion candidates, (RL4) only keywords from clicked results as expansion candidates. This strategy is rather similar to our approaches from the two previous years [HSV10,HGMS11] but with one big difference. This year we decided—based on careful analysis of our last year's runs—that we should be much more conservative when adding terms to a query. Basically it turned out that in the last year we added way too many terms and also that treating all sessions with the same strategy is not the best idea. For instance, when only two queries are available and the last one is a specialization or generalization of the previous one, we believe that not much can be learnt from such few available interactions. In such cases, a low risk strategy (that we want to develop) would not change the query but probably just leave out previously seen/clicked results. Our goal is to develop a strategy that only interferes a user's querying process by applying session knowledge in low risk situations when the chance of harming the user's search experience is low. In high risk cases (e.g., when not much interaction information is available), the idea is to mainly trust the underlying retrieval system.

The research questions we are dealing with are threefold. First, we want to examine the effect of distinguishing between sessions where session knowledge in form of query expansion should be applied (low risk) from those where this is not the case (high risk). Second, we want to expand with very few terms only; avoiding overlong (and thus time-consuming) queries. Third, we want to examine whether similar sessions from other users are a better source for query expansion terms than the previous interactions of the same user. Therefore, we manually created additional sessions on topics with only one or two sessions in the provided session data set (for the other sessions, we used the TREC sessions on the same/similar topics).

Altogether it turns out that with respect to retrieval performance the similar sessions are a better resource than previous interactions from the same session.

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 runs are given in Section 4. A discussion and some concluding remarks follow in Section 5.

2 Retrieval systems

All our three runs are for the full ClueWeb09 corpus (category A). One of our three runs uses the language modeling based Indri search engine provided by the Carnegie Mellon University.[1] The two other runs use our own ClueWeb search engine ChatNoir (French for black cat) [PHS$^+$12] which is mainly based on the classic BM25F retrieval model [RZT04] and an approximate proximity feature with variable width buckets [ELM11].

For all three runs we removed results with spam ranks of 30 or less according to the spam rank list provided by the University of Waterloo [CSC11]. Note that a spam rank of 30 means that only 30% of the ClueWeb have a higher probability of being spam. Hence, we only return results from the 70% portion of the ClueWeb with lower probability of being spam pages. Apart from that, we did not further tune any weighting schemes or other parameters of the search engines used.

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

3 Our three runs

We briefly describe the ideas underlying our three runs. As for RL1, all runs simply use the query as is but for RL2–RL4 we employ different query expansion and result list processing strategies. For all results lists, we exclude previously clicked documents of the same session as the task descriptions said that such documents will be ignored during retrieval performance measurement. We did this even for RL1–RL3 where the click information is not officially available to not treat RL4 in an unfair way (i.e., allowing previously clicked results (that probably are relevant) in RL1–RL3 and then excluding them from RL4 might result in a worse performance of RL4).

Besides the treatment of already clicked documents, all our runs also share the following query preprocessing and empty result list handling schemes.

Query preprocessing For all runs, we preprocess the query strings by lowercasing all keywords, removing punctuation and double white spaces, by removing stopwords, by replacing the keywords `wiki` or `facts` with `wikipedia`, and by spelling correction via the Bing API.

Empty result lists Whenever a query does not return any results (e.g., when expanded with too many keywords), we remove keywords until the result set is not empty anymore. The removal is done in a PROMISING QUERY framework style [SH11,HS11] that was also employed in our two previous Session track approaches [HSV10,HGMS11]. Note however that this year we use the framework only for handling empty result lists and not during the query expansion phase itself.

3.1 Run webiscnqe

In the webiscnqe run, we employ ChatNoir as the retrieval system and case based query expansion (low risk vs. high risk) using the user's previous interactions within the same session.

Query expansion As for RL2–RL4, we compare the current query q to each previous query q' of the same session to "classify" the potential of including interactions for that query (keywords, clicks, etc.) as session knowledge. When q is a repetition, specialization, or generalization of q' (i.e., $q = q'$, $q \subset q'$ or $q \supset q'$), we do nothing except not showing previously clicked results again. In case of a repetition, we additionally do not show documents that contain more than two of the ten most frequent phrases extracted from the shown top-10 results of q' (extraction via the repeated string keyphrase extractor [Tse98]). In all other cases, we add q' to a set Q. For RL2, we extract from Q at most two keyphrases not present in q (using the phrase frequency for ranking) and add them to q. For RL3, we additionally have a string T containing the concatenated titles of all the shown results, a string S containing the concatenated shown snippets, and a string R containing the concatenated strings of the complete documents in the shown results. From each of R, S, and T, we extract at most one keyphrase and add it to q (via the repeated string keyphrase extractor again). As for RL4, we only have the clicked results in the strings R, S, and T with the assumption that these are relevant to the user and the non-clicked ones are not.

Keyphrase processing The extracted keyphrases are cleaned before adding them to q. We remove html artifacts like `add new comment` or `top of page`, double white spaces are omitted, keyphrases shorter than four characters or with more than three keywords are removed, keyphrases that contain URL artifacts like `http`, `www`, or `.com` are removed, and keyphrases too similar to others are removed (Levenshtein distance smaller than 4).

Term weighting Terms in the expanded query q are weighted according to their origin: original terms (weight of 2.0) are more important than the ones from Q (weight of 0.6) which again are more important than the ones from R, S, or T (weight of 0.2 for R and 0.1 for S and T). The idea underlying the weights is that we want to trust the user more than the retrieval system (which is somewhat "responsible" for R, S, and T) as any expansion strategies bear the risk of misunderstanding the user intent.

Result list postprocessing The result list of the potentially expanded query q is treated in a postprocessing phase as follows.

Aspect sessions. For sessions that query different aspects of the same underlying concept (e.g., Session 5 on the `pocono mountains`), we add the Wikipedia article on the concept to the top-100 documents in the result list if it is not contained (even when it was clicked before!). Note that we add it to the top-100 as these will be later reordered. Our idea for the aspect sessions is that we believe that the user potentially did miss that different aspects are covered in the Wikipedia article when it was shown before (when it was not shown, it might be a good document anyway).

Wikipedia. Using the Wikipedia baseline query segmentation algorithm [HPBS12], we find the most important titles of Wikipedia articles in q (if there are any). For titles with at least two keywords, we add the respective Wikipedia articles to the top-100 documents in the result list if they are not contained. Finally, all the Wikipedia articles in the top-100 (including the ones for aspect sessions) are moved to the top spots of the result list respecting their previous relative order. The underlying idea here is that often Wikipedia articles on concepts contained in a query should be relevant for the query. Note that this point is related to the RGU-ISTI-Essex team's 2011 baseline that adds the term `wikipedia` to any query [ANA+11]. However, in our case we even support queries that themselves do not match any Wikipedia article.

Clicks from similar queries. Each query q' in any of the TREC sessions is checked for overlap with the current query q. Whenever at least 2/3 of the terms of q' are contained in q, the counters for the clicked results of q' in a click table C (containing how often users clicked on documents) are increased. Finally, the two most clicked documents in C that were clicked by at least two users are added to the ranks 3 and 4 of the result list of q. Due to the requirement of known clicks, this technique is only available for RL4. The underlying idea is to exploit the clicks of others but trust the first two results of the expanded query q (note however, that the first two results could also be Wikipedia articles from the previous postprocessing steps).

Long documents. We remove documents with a length greater than 7000 words as our pre-tests showed such documents to mostly be spam.

Duplicate documents. We remove documents whose 5-gram cosine similarity (tf weights) to a document ranked above it is greater than 0.98 as we view such documents as duplicates and do not want to show the same content twice (even though this might improve retrieval scores when the duplicates are all relevant).

3.2 Run webisindqe

The webisindqe run applies the same strategies as the webiscnqe run with two exceptions. First, the retrieval system is not ChatNoir but the online Indri search engine for the ClueWeb provided by the Carnegie Mellon University. Second, we employ query segmentation (i.e., highlighting phrases in the queries). In pre-tests we compared different segmentation strategies [HPSB10,HPSB11,HPBS12] and decided to use a hybrid query segmentation optimized for usage with Indri [HPBS12] (basically, no segmentation for strict noun phrase queries and a Wikipedia-based segmentation for all others). Hence, the term weighting now is also done on phrase level instead of keyword level.

3.3 Run webiscnse (A potentially manual run)

With the webiscnse run we want to examine the potential of knowledge from related sessions compared to the same session of the user that we used in our two other runs. A similar idea was employed last year by the RGU-ISTI-Essex team [ANA$^+$11]. They used the Microsoft RFP 2006 query log to extract similar sessions of other users. These sessions are then used as search shortcuts in the form of query expansion with keywords from them. However, the resulting differences in retrieval performance compared to unexpanded queries were not statistically significant. We think that this might have been due to the lack of related sessions extractable from the used query log. Note however that the authors did not give any numbers of how successful their search for related sessions in the RFP 2006 log was. We tried to use the AOL query log for the very same purpose (due to the non-availability of the RFP log at our site) and found only a handful of queries related to the sessions of this year's Session track. Instead of using the AOL log, we thus decided to use the sessions on the same topic from the released TREC sessions. However, there are topics with too few sessions where we think that related sessions might have a potential to be useful. Hence, for each of the sessions 1, 3, 8, 34, 38, 46, 53, 64, 66, 69, and 92 of the original 2012 Session track data set, we manually created three additional sessions using ChatNoir—making this run a potentially manual run in our opinion. With the manually created sessions, we can ensure to have related sessions at hand for all the cases where we want to apply them.

The run itself uses the ChatNoir search and is analogous to the webiscnqe run. However, the sets Q, R, S, and T are not populated via the previous interactions of the same user but via the similar sessions of other users (TREC released or manually created). Another difference is with respect to result list postprocessing using clicks from other sessions. In the webiscnse run we populate the very top of the ranking with all the clicked documents of other users (even the ones that were clicked only once). This rather aggressive change is meant to evaluate the potential of related clicks from other users.

4 Evaluation

The evaluation for the Session track is done by comparing the four ranked lists RL1–RL4 with respect to nDCG@10. Our runs' performances are given in Table 1.

Table 1. Results for nDCG@10 averaged over all 98 topics. Statistically significant improvements for our runs compared to RL1 are marked with a $\Uparrow$ (paired, two-sided t-test with $p < 0.05$). Other improvements compared to RL1 are marked with a $\uparrow$.

	RL1	RL2	RL3	RL4
Run webiscnqe	0.0865	0.1174 $\Uparrow$	0.1204 $\Uparrow$	0.1171 $\Uparrow$
Run webiscnse	0.1086	0.1220 $\Uparrow$	0.1401 $\Uparrow$	0.1796 $\Uparrow$
Run webisindqe	0.2053	0.2097 $\uparrow$	0.2102 $\uparrow$	0.2077 $\uparrow$
Median all systems	0.2455	0.1745	0.1900	0.2160
Max all systems	0.4642	0.4449	0.4759	0.4831

Analyzing the performances of our runs it stands out that both ChatNoir runs (even with their RL4 setting) still perform much worse than the basic Indri RL1 (and all are below the median of all runs). Furthermore, while the ChatNoir runs' RL2–RL4 can significantly improve performance over their respective RL1, this is not the case for Indri. Comparing the webiscnqe and the webisindqe runs which basically used the same strategy, it turns out that our query expansion ideas do not really work for the Indri retrieval model. The strategy is probably too conservative in the Indri case as the returned retrieval results often do not change.

Analyzing each of the ChatNoir runs in more detail, it turns out that our query expansion techniques are probably only responsible for the significant improvements from RL1 to RL2. For RL2 we only used terms provided in previous queries while for RL3 and RL4 a non-negligible amount of the retrieval improvement might be caused by implicitly incorporating the much better performing retrieval system used in the creation of the TREC sessions. Document titles, snippets, and clicks that we use for query expansion are heavily influenced by the original search engine as they stem from its top-10 results. That's why the RL3 and RL4 performances in the ChatNoir runs cannot really be compared to their respective RL1 performance in a fair way.

However, even though the improvements within each single ChatNoir run cannot be viewed as a good argument for the respective strategies, comparing both runs with each other is more reasonable (and more interesting). Note that for RL4, the webiscnse run manages a 50% performance improvement over webiscnqe using the same retrieval system. Hence, including clicks from other users' related sessions and incorporating other user's sessions for query expansion seems to have a much higher potential of improving retrieval effectiveness than using a single user's interactions alone.

5 Discussion

With respect to our three research questions (1) risk-aware session handling, (2) conservative query expansion, and (3) incorporating other user's sessions, the obtained results are rather promising. First of all, risk-aware session handling seems a very safe way to go. Our approaches for the previous two Session tracks at TREC 2010 and TREC 2011 incorporated session knowledge via query expansion for every session without estimating potential or risk. This resulted in drastic performance losses for many topics while improving only a few others. In contrast, employing risk-aware strategies and often leaving the last query of a session untouched, we observed hardly any performance losses this year. Algorithmically "intervening" a user's search process only in low risk cases thus seems a promising research direction.

As for the conservative query expansion question (i.e., expansion with few terms only), the picture is not that clear. The ChatNoir runs achieve a statistically significant improvement from RL1 to RL2 (different to our last years' approaches that added many more terms) while the Indri run's performance is basically unchanged. Thus, Indri seems to require more aggressive expansion (i.e., adding more terms) than ChatNoir.

Incorporating other users' sessions yields very good results. Comparing RL4 of the ChatNoir run with the related sessions (webiscnse) to RL4 using only the same user's session (webiscnqe) achieves a 50% improvement in retrieval performance. However, as the initial retrieval performance of ChatNoir is rather low, it would be very interesting to test the same idea with other retrieval systems (e.g., Indri) as well.

Besides the above promising results there is one major "drawback" of all the developed techniques for the TREC Session track that also comes with our this year's contribution. So far the search session boundaries given in the TREC session data set are taken as granted. But in real life situations the system itself has to detect whether a user is still submitting queries for the same information need or not. This leads to the task of automatic session and mission detection techniques and we experimented with some state-of-the-art detection approaches [HSR11,HGS12] on the provided TREC sessions. Indeed, the detection methods divided a few of the sessions into smaller parts. We plan to further evaluate whether in these few cases using only the smaller "sub"-session containing the last query instead of the complete TREC provided session can yield any further improvements. A natural next step could then also be to develop a much more fine grained classification of when to apply what session support technique etc.

References

[ANA+11] Ibrahim Adeyanju, Franco Maria Nardini, M-Dyaa Albakour, Dawei Song, and Udo Kruschwitz. RGU-ISTI-Essex at TREC 2011 Session track. In *Proceedings of TREC 2011*.

[CSC11] Gordon V. Cormack, Mark D. Smucker, and Charles L. A. Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. *Information Retrieval*, 14(5):441–465, 2011.

[ELM11] Tamer Elsayed, Jimmy J. Lin, and Donald Metzler. When close enough is good enough: approximate positional indexes for efficient ranked retrieval. In *Proceedings of CIKM 2011*, pages 1993–1996.

[HGMS11] Matthias Hagen, Jan Graßegger, Maximilian Michel, and Benno Stein. Webis at the TREC 2011 Sessions track. In *Proceedings of TREC 2011*.

[HGS12] Matthias Hagen, Jakob Gomoll, and Benno Stein. Improved cascade for search mission detection. In *Proceedings of ECIR 2012 Workshop on Information Retrieval over Query Sessions*.

[HPBS12] Matthias Hagen, Martin Potthast, Anna Beyer, and Benno Stein. Towards optimum query segmentation: in doubt without. In *Proceedings of CIKM 2012*, pages 1015–1024.

[HPSB10] Matthias Hagen, Martin Potthast, Benno Stein, and Christof Bräutigam. The power of naïve query segmentation. In *Proceedings of SIGIR 2010*, pages 797–798.

[HPSB11] Matthias Hagen, Martin Potthast, Benno Stein, and Christof Bräutigam. Query segmentation revisited. In *Proceedings of WWW 2011*, pages 97–106.

[HS11] Matthias Hagen and Benno Stein. Applying the user-over-ranking hypothesis to query formulation. In *Proceedings of ICTIR 2011*, pages 225–237.

[HSR11] Matthias Hagen, Benno Stein, and Tino Rüb. Query session detection as a cascade. In *Proceedings of CIKM 2011*, pages 147–152.

[HSV10] Matthias Hagen, Benno Stein, and Michael Völske. Webis at the TREC 2010 Sessions track. In *Proceedings of TREC 2010*.

[PHS$^+$12] Martin Potthast, Matthias Hagen, Benno Stein, Jan Graßegger, Maximilian Michel, Martin Tippmann, and Clement Welsch. ChatNoir: a search engine for the ClueWeb09 corpus. In *Proceedings of SIGIR 2012*, page 1004.

[RZT04] Stephen E. Robertson, Hugo Zaragoza, and Michael J. Taylor. Simple BM25 extension to multiple weighted fields. In *Proceedings of CIKM 2004*, pages 42–49.

[SH11] Benno Stein and Matthias Hagen. Introducing the user-over-ranking hypothesis. In *Proceedings of ECIR 2011*, pages 503–509.

[Tse98] Yuen-Hsien Tseng. Multilingual keyword extraction for term suggestion. In *Proceedings of SIGIR 1998*, pages 377–378.