Studying Search Shortcuts in a Query Log to Improve Retrieval Over Query Sessions

M-Dyaa Albakour[1], Franco Maria Nardini[2],
Ibrahim Adeyanju[3], and Udo Kruschwitz[1]

[1] School of Computer Science and Electronic Engineering, University of Essex, UK
malbak@essex.ac.uk

[2] ISTI–CNR, Pisa, Italy
francomaria.nardini@isti.cnr.it

[3] IDEAS Research Institute, The Robert Gordon University, UK

Abstract. In this paper, we study a state-of-the-art model to derive query suggestions from the search logs of a Web search engine in the context of its application for the session retrieval problem. Using the session track 2011 as an evaluation platform and the MSN 2006 logs, we analyze the characteristics of this model and how it can be optimized to better represent user information needs throughout the session.

1 Introduction

Derived from large query logs, Search Shortcuts have proven to be an effective model for query recommendation, especially for rare or unseen queries [3]. Recently, this model was employed to tackle the task in the session track of TREC 2011 [1]. Under this setup, the model can be useful by expanding the user query with Search Shortcuts relevant to the query and previous user interactions throughout the session. Using the MSN Logs 2006 to derive the shortcuts, the technique showed improvements with respect to one of the two evaluations carried on in the competition while it fails to improve over a simple baseline in the other one.

The Session Track overview document describes the methodologies that participants employed in their submitted runs [6, 5]. An analysis of these summaries and the related results achieved in the past two years illustrate the state of the art in search over session queries. In the 2010 edition, the methods employed included term weighting, pseudo-relevance feedback, query expansion, ranked list re-ranking and merging. Query expansion related features produce the more successful submissions. The second year has seen a near-complete overhaul of the track in terms of topic design, session data, and experimental evaluation. In the 2011 edition, the methods presented include pseudo-relevance feedback, query expansion, adaptive modelling, and semantic approaches.

In this paper, we have a closer look at the Search Shortcuts model employed in the the session track task of TREC 2011. In particular, we want to address these questions:

- What are the optimal parameters for the expansion process with the Search Shortcuts in the session track task of TREC 2011?
- How does the different Search Shortcuts ranking schemes and the session retrieval performances correlate? Is it the retrieval score or the frequency of the query in the log that matters?
- How does different "classes" of recommendation ("history-based" recommendation, "current-query-based" recommendation) affect the session track performances?

2 Query Expansion with Search Shortcuts

The Search Shortcuts recommender system has been effectively designed in [2]. It recommends queries by efficiently computing similarities between partial user sessions (the one currently being performed) and historical successful sessions recorded in a query log. Final queries of most similar successful sessions are suggested to users as Search Shortcuts. The algorithm is thus able to recommend queries by taking into account the activity performed by the user in the entire session (history-based suggestion). In particular, the algorithm works by computing the value of a scoring function δ, which for each successful session measures the similarity between its queries and a set of terms τ representing the user need expressed so far. Intuitively, this similarity measures how much a previously seen session overlaps with the current user need (the concatenation of terms τ serves as a bag-of-words model of the user need). Sessions are ranked according to δ scores and, from the subset of the top ranked sessions, it is possible to suggest their final queries. It is obvious that depending on how the function δ is chosen the algorithm provides different recommendations. In [2], authors opted for δ to be the similarity computed as in the BM25 metrics [9]. In particular, authors define the final recommendation scoring formula as a linear combination of the BM25 score and the frequency in the log of the recommendation they are suggesting. Given a query q and the set of possible suggestions S_q for q, for each $s \in S_q$, let V_s be the representation of the successful sessions ending with s. Therefore, the score of s is computed as follows:

$$\delta(q, s) = \alpha \times BM25(q, V_s) + (1 - \alpha) \times freq(s) \quad (1)$$

In our TREC Session Track 2011[1], we use the Search Shortcuts query recommender system as a term expansion technique for improving performances over query sessions. The task in the Session Track 2011 involves generating 4 ranked lists (RL1, RL2, RL3, RL4) for each session in a total of 76 user sessions provided to the participants. Here we explain how we generate each list with our Search Shortcut approach.

RL1 is the required ranked list of results for the 'current query' in a given session without taking into account previous interactions of the user. This is the baseline ranked list on which we wish to improve.

[1] Session Track 2011 guidelines and terminology are available at http://ir.cis.udel.edu/sessions/guidelines.html.

RL2 is the required ranked list of results for the 'current query' by taking into account previous queries in the session. We built it by using an expansion of the current query made by the terms composing the first three recommendations provided by the method for the *entire session* (a query composing of the current query and all previous queries in the session) and by uniformly weighting them after performing a stopword removal step.

RL3 is the required ranked list of results for the 'current query' by taking into account previous queries and displayed documents to the user in the session. It was built starting by the first ten recommendations generated for each current query of the sessions provided. We use the terms composing them to produce an expanded query representation. After removing stopwords, we develop a term weighting scheme based on how many times any given expansion term appears in the snippets of the documents returned within the session. The final weight is thus computed by dividing the frequency of the single expansion term with the sum of the frequencies of the expansion terms over all the returned documents.

RL4 is the required ranked list of results for the 'current query' by taking into account previous queries in the session, displayed and clicked documents with their dwell times. We used two different approaches for determining RL4. The first approach for building RL4 use only clicked documents. It was produced starting from the weights computed within RL3 and by adding to the terms appearing in the snippets of the clicked documents a boosting factor. The rationale of this is to promote terms that have been clicked by the user. This boosting factor depends on the frequency (with repetitions) of the given term in the set of the clicked documents divided by the total frequency of the expansion terms that are present in the set of the clicked documents. The second approach used for building RL4 use also dwell time. It consists of adding to the weights produced for RL3 a new weight obtained by exploiting the dwell time of each clicked document. Here, the rationale is to promote terms appearing in clicked documents proportionally to the time that it has been visualized. We do this on a term basis. We thus select a candidate expansion term, we check if it is part of the snippet of one or more clicked documents and we compute its weight as a sum of its dwell time within the session divided by the total dwelling time of all the documents within the session. We then sum the weights obtained to the weights referring to RL3. Finally, we normalize over all the expansion terms to obtain a set of weights that sum to one.

The retrieval process makes use of an existing Indri[2] index of the ClueWeb09 dataset. The Indri search engine [7, 8] uses language modeling probabilities. The current query of each session is linearly combined with the expansion generated using Search Shortcuts. Figure 1 shows a generated Indri query for session 2 (RL3).

In our Session Track 2011 submission [1], for all the sessions and each ranked list we arbitrarily chose the values 0.7 and 0.3 for both components of the expansion, i.e., the original query and the weighted expansion set of terms produced by the Search Shortcuts method.

[2] http://lemurproject.org/indri.php

```
#weight(
0.7 #combine(event planning college)
0.3 #weight(0.16 #combine(college) 0.01 #combine(fashion) 0.36
#combine(event) 0.4 #combine(planning) 0.01 #combine(conference) 0.05
#combine(online) 0.01 #combine(management))
)
```

Fig. 1. An example of an Session Track 2011 query expanded by means of Search Shortcuts.

3 Experiments

3.1 Tuning the Expansion Process

The retrieval performances of our proposed method depend on some parameters affecting both the generation of the query expansion and the retrieval of the expanded query. Starting from our Session Track 2011 results and using the relevant judgments provided by NIST assessors to evaluate the participating runs, we analyse how retrieval performances change by varying the set of parameters affecting: i) how the expansion set is built, ii) how results are retrieved. In particular, we vary: i) the numbers of suggestions used for building the expansion, ii) a filtering threshold working on term weights, iii) the combination of the retrieval weights associated to both the components of the expansion (i.e., the original query and the weighted expansion set) of the Indri query. Performances are evaluated in terms of nDCG@10 against all subtopics (differences with respect to the TREC paper [1] are due to a different set of parameters used for retrieving suggestions).

The number of suggestions generated by the Search Shortcuts engine affects the retrieval performances as their terms are used within the expanded representation of each query. Varying the number of suggestions, we test the retrieval performances of our method by building the weighted expansion set using 3, 5, 10 or 20 recommendations per query. The best performances (in terms of nDCG@10) are obtained by using 10 recommendations. In particular, while both RL3, RL4 (click) and RL4 (time) increase their performances by increasing the number of recommendations from 5 to 10, only RL4 (time) shows the highest sensitivity to this parameter (from 0.3631 obtained to 0.3812 using 10 recommendations). Results also show that a higher number of recommendations (20) used for building the expansion set decrease the retrieval performances. Possible reason of that is the "noise" (i.e., recommendations that are poorly correlated with the original query) starting to appear when the list of recommendations used becomes long.

We also perform an evaluation of the retrieval performances of our method after introducing a filtering threshold on the expansion terms. The rationale of this choice is to understand if terms presenting a very low expansion weight (as an example, the term "conference" in Figure 1) act as "noise" degrading the whole retrieval performances of the query. We test four different values of the threshold: 0 (no threshold), 0.05, 0.1, 0.15. In all the runs, the best performances are obtained when no filtering threshold is used. In addition, performances pro-

portionally degrade as the filtering threshold value increase. This suggests that low-scored expansions may be important in modelling the user session.

A third analysis is carried on the retrieval weights associated with both the components of the expansion (i.e., the original query and the weighted expansion set) of the Indri query (see Figure 1). We test five combinations of the parameters: (0.5, 0.5), (0.6, 0.4), (0.7, 0.3), (0.8, 0.2), (0.9, 0.1), where the first one refers to the original query while the second one refers to the weighted expansion set. The optimal retrieval performances (in terms of nDCG@10) for RL2 (0.3683) can be achieved by using (0.8, 0.2), while the best performance for RL3 (0.3774) is achieved by using (0.7, 0.3). RL4 (click) shows its best behavior (0.3811) when the combination (0.6, 0.4) is used, while the best value for RL4 (time) (0.3812) is achieved by using the combination (0.7, 0.3). Results show that is possible to obtain better retrieval performances by using an appropriate weighting scheme for both the components of the expansion (i.e., the original query and the weighted expansion set) of each run.

3.2 Search Shortcuts Ranking Schemes

The ranking of the Search Shortcuts is given in Equation 1. The parameter α balances the importance given to either the frequency of the Search Shortcut or the retrieval score of similar sessions to user queries. We want to test the effect of the ranking schemes on the session retrieval performance. We test three values of α (1.0, 0.5, 0.0) which correspond to only BM25 ranking, equal weights on BM25 and frequency and only frequency ranking. We report the results of evaluation using all the subtopics criteria in Table 1. Ranking Search Shortcuts for query expansions work best with BM25 scoring while using only frequency harms the performance. Sessions which are more relevant to the user query are good candidates for query expansions to provide a better session retrieval performance. BM25 scoring for RL2 outperforms the other two schemes significantly. The increase in retrieval performance is not as significant over RL3 or RL4 which is expected as in RL3 and RL4 we use the same documents to filter the candidate expanding terms. Here we do not report the results for last subtopic evaluation, but we also found out that the BM25 scoring was the only model showing improvement.

Table 1. nDCG@10 values when assessing all subtopics; the arrows indicate improvement (↑) or decline (↓) against the previous results lists, the first arrow in a cell relates to RL1, the second arrow to RL2 and so on. Double arrows (⇑ / ⇓) indicates the comparison is statistically significant returning a two tailed t-test $value \leqslant 0.05$.

System	RL1	RL2	RL3	RL4
BM25(α=1.0)	0.3634	⇑ 0.3978	↑ ↑ 0.3981	⇑ ↑ ↑ 0.4035
BM25+f(α=0.5)	0.3634	↑ 0.3707	↑ ↑ 0.3965	⇑ ⇑ ↑ 0.3993
f(α=0.0)	0.3634	↓ 0.3579	↑ ⇑ 0.3854	⇑ ⇑ ⇑ 0.3971

3.3 Search Shortcuts with "No History"

In all the previous experiments the Search Shortcuts used for expansions are extracted using the entire session rather than the current query. We want to see the

impact of using only the last query to extract shortcuts on the session retrieval performance. The intuition here is that recommendation with "no history" may possibly improve the retrieval performance for the subtopics of the current query. However, the results in Table 2 does not support that claim. One explanation could be that it is adding more noise to the current user interest as it diversify the information needs. This behavior of the Search Shortcuts recommender system has been, in fact, already exploited within another application: Web search results diversification [4]. Here, the "no history" recommendation has been used for generating a set of possible meaning behind a given "ambiguous" query.

Table 2. nDCG@10 values when assessing last subtopics; the arrows have similar indication of arrows in Table 1.

System	RL1	RL2	RL3	RL4
History	0.2301	↑ 0.2412	↑ ↓ 0.2332	↑ ↓ ↑ 0.2369
No History	0.2301	↓ 0.203	⇓ ⇓ 0.1583	⇓ ⇓ ↑ 0.1633

4 Conclusion

In this paper, we conducted a thorough evaluation of using Search Shortcuts, mined from query logs, for query expansion to guide session retrieval. We optimized the expansion process and show that similar sessions are more useful than popular queries for that. Finally, we found that shortcuts which does not take into account previous actions of users have a negative impact on session retrieval.

Acknowledgments

This work has been partially funded by the EU projects: S-CUBE (EU-FP7-215483), ASSETS (CIP-ICT-PSP-250527), and INGEOCLOUDS (CIP-ICT-PSP-297300). This research is also part of the AutoAdapt research project. AutoAdapt is funded by EPSRC grants EP/F035357/1 and EP/F035705/1.

References

1. I. Adeyanju, F. M. Nardini, M.-D. Albakour, D. Song, and U. Kruschwitz. RGU-ISTI-Essex at TREC 2011 session track. In *TREC 2011 Proceedings*. NIST, 2011.
2. D. Broccolo, L. Marcon, F. M. Nardini, R. Perego, and F. Silvestri. Generating suggestions for queries in the long tail with an inverted index. *Information Processing and Management (IPM)*, 2011.
3. D. Broccolo, L. Marcon, F. M. Nardini, R. Perego, and F. Silvestri. Generating suggestions for queries in the long tail with an inverted index. *IP&M*, 48(2):326–339, Mar. 2012.
4. G. Capannini, F. M. Nardini, R. Perego, and F. Silvestri. Efficient diversification of web search results. *Proceedings of the VLDB, Volume 4, Issue 7*, April 2011.
5. E. Kanoulas, B. Carterette, P. Clough, and M. Sanderson. Session track 2010 overview. In E. M. Voorhees and L. P. Buckland, editors, *TREC*. NIST, 2011.
6. E. Kanoulas, B. Carterette, M. Hall, P. Clough, and M. Sanderson. Session track 2011 overview. In E. M. Voorhees and L. P. Buckland, editors, *TREC*. NIST, 2012.
7. V. Lavrenko and W. B. Croft. Relevance based language models. In *SIGIR'01*, pages 120–127, New York, USA, 2001. ACM.
8. D. Metzler and B. W. Croft. Combining the language model and inference network approaches to retrieval. *Information Processing and Management*, 40:735–750, 2004.
9. S. Robertson and H. Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009.