Learning to Shorten Query Sessions

Cristina Ioana Muntean,
Franco Maria Nardini,
Fabrizio Silvestri
ISTI–CNR, Pisa, Italy
{name.surname}@isti.cnr.it

Marcin Sydow
Institute of Computer Science,
Polish Academy of Sciences and
Polish-Japanese Institute
of Information Technology
Warsaw, Poland
msyd@ipipan.waw.pl

ABSTRACT

We propose the use of learning to rank techniques to shorten query sessions by maximizing the probability that the query we predict is the "final" query of the current search session. We present a preliminary evaluation showing that this approach is a promising research direction.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Search process*

Keywords

Query Prediction, GBRT, Learning to Rank

1. INTRODUCTION

Web search engines deeply rely on query recommendation techniques to help users formulate their queries. In this paper, we go a step beyond query recommendations, by proposing a way to shorten query sessions. We introduce the *final query prediction* problem, as the problem of predicting the most likely "final" query of the current search session, i.e. a topically coherent sequence of queries of a user, by means of machine learning techniques. We assume this query to be an effective representation of the user information need [2]. While query recommendation approaches aim at producing a list of relevant and useful suggestions for a given query, here we aim at avoiding the reformulation process that often can be found in sessions, thus shortening them. Our approach comprises a sort of "I am feeling lucky" service that predicts one query as the best representation of an information need. We believe such a service can add value to the query processing of a modern Web search engine. As an example, the predicted query can be used directly in the retrieval of results, instead of using the original user query.

2. RELATED WORK

The *final query prediction* problem can be seen as a successive step of the query recommendation problem, which is recently approached by means of learning to rank techniques. The goal is to determine the relevance of a suggestion with respect to a query by exploiting many available signals. For example, Ozertem *et al.* [3], propose to learn the probability that a user may find a follow-up query useful and relevant, given his initial query, and demonstrate, with offline (based on human judges) and online (based on millions of user interactions) evaluations, the superiority of their approach over some important baselines. Santos *et al.* [4] rank query suggestions based on a structured representation of queries based on common sessions and clicks to overcome data sparsity for long-tail queries. This representation can be exploited along multiple ranking criteria, that could be used as features for learning to rank query suggestions. Experiments on the TREC Web track data show the effectiveness of the approach in adhoc and diversity search.

To the best of our knowledge, shortening query sessions has been poorly investigated so far. This paper proposes a preliminary formulation and a possible research approach to the problem.

3. PROBLEM STATEMENT & SOLUTION

Our work relies on the hypothesis that "final" queries in a query log are effective representations of user information needs. This assumption has been studied in the past. Broccolo *et al.* validate this hypothesis [2] showing that final session queries ending with at least a click on a result *are usually close to the achievement of the user information goal.*

Given a current query q_i, the *final query prediction* problem aims at predicting a final query q_f that is an effective representation of the information need behind q_i. We propose to learn an optimal ranking function $h : X \rightarrow Y$, where the input space X is a vector representation of a candidate prediction q_f. We employ the query features shown in Table 1 to map each q_f on the input space X. Features can be classified in three different sets: "session", "query pair", and "final query" features. Each of them is a set of signals representing different aspects of the user search activity. In particular, while "session" features aim at representing the behavior of the user in the current search session, "query pair" features model the characteristics of the current query w.r.t each candidate query q_f and "final query" features describe the properties of the candidate query q_f. To be noted, we rely on an effective method for mining sessions (i.e., sequences of queries referring to the same information need) from the query log [5], that is crucial due to the presence of multi-tasking activity in the user search behavior.

Moreover, the output space Y for the learning problem consists of a set of ground-truth labels determining the relevance of each query prediction. As our aim here is to learn a model that predicts the most likely "final" query for a query q_i in the user session, we select as training examples for the

WWW 2013 Companion, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2038-2/13/05.

session only final queries from the query log. We consider the final query q_f of the current session as a positive example, whereas other final queries represent negative examples.

Labels per session for positive and negative examples are thus assigned according to:

$$y_{q_f} = \begin{cases} 1, & \text{if } q_f \text{ is the final query of the session;} \\ 0, & \text{otherwise.} \end{cases}$$

We employ GBRT [6], state-of-the-art in web search ranking, as the learning to rank algorithm. It consists in an ensemble of regression trees for determining the predicted value of the label assigned to each candidate query. The loss function used for optimizing the learning is the root mean square error between the label assigned to each example in the training set, i.e., y_{q_f}, and its predicted value, $h(q_f)$.

Session Features	
sumCoOccurrInSession	Cumulative popularity of queries in session
frequentQueries	# frequent queries in session
totalTime	Total session time
numberOfTermsTrend	Trend of the number query terms in session
avgNumClicks	Average # clicks in session
avgNumTerms	Average # terms in session
Query Pair Features	
trigramsInCommon	# tri-grams in common
bigramsInCommon	# bi-grams in common
termsInCommon	# terms in common
coOccurrFrequency	# co-occurrences of query pairs
Final Query Features	
queryLength	# characters in the query
queryNumberTerms	# terms in the query
queryPopularity	Frequency of the query
queryNumClicks	Total # clicks per query
queryPopularTermFreq	Frequency of top query term
successfullyEndingQuery	True iff query is final and clicked

Table 1: List of the three feature sets used to model each candidate query in the features space X.

In the training phase, examples for a set of training sessions are used to learn the GBRT. The model is then employed in the test phase to score test candidates. The input of the test phase is a list of candidate "final" queries for any given test session. The output of the test phase is a re-ranked list of candidates sorted by decreasing probability of being the "final" query of the test session.

4. PRELIMINARY RESULTS

We evaluate our approach on sessions devised from the MSN RFP 2006 query log ($\sim$ 15 mil. queries, 1 month) [5] . We preprocess the log by converting all the queries to lowercase, and by removing stop-words and punctuation/control characters. We then apply a session splitting technique based on the Query Flow Graph [1]. From the set of sessions obtained, we filter out sessions with 4 or less queries, considered already fairly short. The resulting set of sessions is then divided in training ($20,000$ sessions) and test set ($4,000$ sessions), used to measure the performance of our model.

Given a session $(q_1, \ldots, q_i, \ldots, q_n)$ of length n, we select q_i as the current query of the user. We use $(q_1, \ldots, q_i)$ to devise session features, (q_i, q_n) to compute query pair features and q_n to compute query features. In both training and test sets, the final query q_n of the session is the positive example. Accordingly, in the training, for each session we have 1 positive and 5 negative examples, whereas in test, we have 1 positive and 9 negative examples. The negative examples are sampled from the set of final queries in the entire query log, considered non relevant for the given session. This restrictive strategy allows us to measure the real performance of our predictor without being biased by the effectiveness of a specific candidate generator as, for example, a query recommender engine [2].

We learn a GBRT to re-rank suggestion candidates. We measure the percentage of sessions correctly predicted at different levels of the re-ranked list (P) and Mean Reciprocal Rank (MRR). We run the analysis by varying i, thus evaluating the importance of "session" features in the prediction.

i	P (MRR)			
	@1	@2	@3	@5
2	24.45 (0.24)	32.07 (0.28)	35.85 (0.29)	46.72 (0.31)
3	26.45 (0.26)	34.00 (0.30)	40.17 (0.32)	50.77 (0.34)
4	30.92 (0.30)	39.22 (0.35)	44.60 (0.36)	57.62 (0.39)

Table 2: Results of our proposed technique in terms of P (%) and MRR by varying i and the number of top re-ranked candidates considered.

For $i = 2$, we have two queries in the current session and we want to predict the final query. P@1 in this case is 24.45%, a promising result considering the small size of the session. By increasing i, we denote an increasing trend in P as well. For $i = 3$, P@1 is 26.45% and for $i = 4$, P@1 is 30.92%, due to the fact that more data in session features consolidate the learning, by adding valuable information. Such a tool becomes advantageous when the number of queries in session is rather small.

In conclusion, we used learning to rank for shortening query sessions. Preliminary results show that the technique is effective in predicting "final" queries. As future work, we plan to refine the model and to extend the set of features used in the prediction.

Acknowledgments. This work has been partially funded by InGeoClouds (CIP-ICT-PSP-2011-5, grant n. 297300) and MIDAS (EU FP7 STREP, grant n. 318786).

References

[1] P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, and S. Vigna. The query-flow graph: model and applications. In *Proc. CIKM'08*. ACM, 2008.

[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. *Inf. Process. Manage.*, 48(2):326–339, 2012.

[3] U. Ozertem, O. Chapelle, P. Donmez, and E. Velipasaoglu. Learning to suggest: a machine learning framework for ranking query suggestions. In *Proc. SIGIR'12*, New York, NY, USA, 2012. ACM.

[4] R. L. T. Santos, C. Macdonald, and I. Ounis. Learning to rank query suggestions for adhoc and diversity search. *Information Retrieval*, 2013.

[5] F. Silvestri. Mining query logs: Turning search usage data into knowledge. *Foundations and Trends in Information Retrieval*, 1(1-2):1–174, 2010.

[6] S. Tyree, K. Q. Weinberger, K. Agrawal, and J. Paykin. Parallel boosted regression trees for web search ranking. In *Proc. WWW '11*, pages 387–396, NY, USA, 2011. ACM.