Information Processing and Management 52 (2016) 628–643

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

Learning from homologous queries and semantically related terms for query auto completion

Fei Cai [a,b,*], Maarten de Rijke [b]

[a] Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Hunan, China
[b] Informatics Institute, University of Amsterdam, Amsterdam, The Netherlands

ARTICLE INFO

Article history:
Received 27 April 2015
Revised 26 November 2015
Accepted 1 December 2015
Available online 12 January 2016

Keywords:
Query auto completion
Semantics
Query suggestion
Learning to rank

ABSTRACT

Query auto completion (QAC) models recommend possible queries to web search users when they start typing a query prefix. Most of today's QAC models rank candidate queries by popularity (i.e., frequency), and in doing so they tend to follow a strict query matching policy when counting the queries. That is, they ignore the contributions from so-called homologous queries, queries with the same terms but ordered differently or queries that expand the original query. Importantly, homologous queries often express a remarkably similar search intent. Moreover, today's QAC approaches often ignore semantically related terms. We argue that users are prone to combine semantically related terms when generating queries.

We propose a learning to rank-based QAC approach, where, for the first time, features derived from homologous queries and semantically related terms are introduced. In particular, we consider: (i) the observed and predicted popularity of homologous queries for a query candidate; and (ii) the semantic relatedness of pairs of terms inside a query and pairs of queries inside a session. We quantify the improvement of the proposed new features using two large-scale real-world query logs and show that the mean reciprocal rank and the success rate can be improved by up to 9% over state-of-the-art QAC models.

1. Introduction

Query auto completion (QAC), a popular feature of modern search engines, is offered to help users formulate a query when they have an intent in mind but not a clear way to express it. The typical query completion service of a modern search engine takes a few initial characters entered by the user and returns matching queries to automatically complete the search clue. Where offered, query completion is heavily used by visitors and highly influential on search results (Mitra, Shokouhi, Radlinski, & Hofmann, 2014).

Unlike query recommendation or query suggestion, auto-completed queries strictly start with an initially typed prefix (Cai, Liang, & de Rijke, 2014b). Most previous work on QAC is centered around the Most Popular Completion (MPC) approach, which ranks QAC candidates by query popularity, i.e., frequency, collected either from historical logs (Bar-Yossef & Kraus, 2011; Whiting & Jose, 2014) or from future predictions (Shokouhi & Radinsky, 2012; Whiting & Jose, 2014). In the latter

* Corresponding author at: Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Hunan, China. Tel.: +31 687591248.
E-mail address: f.cai@uva.nl, caifei1104@gmail.com (F. Cai).

http://dx.doi.org/10.1016/j.ipm.2015.12.008
0306-4573/

	SessionID	UserID	Query	Time
1	821174	1662425	google	20060408, 17:02:46
2	821174	1662425	evanescence	20060408, 17:04:21
3	821174	1662425	ulitimate guitar	20060408, 17:05:13
4	821174	1662425	evanescence videos	20060408, 17:09:44
5	821174	1662425	evanescence videos	20060408, 17:16:23
6	821174	1662425	music videos	20060408, 17:17:31

Fig. 1. An AOL session example.

1	music
2	music downloads
3	music videos
4	music lyrics
5	music video codes
6	music codes
7	music new
8	music download
9	musicians friend
10	mustang

(a) Ranked by frequency of candidates.

1	music
2	music videos
3	music download
4	music new
5	music codes
6	music downloads
7	music lyrics
8	mustang
9	music video codes
10	musicians friend

(b) Reranked by frequency of homologous queries.

1	music videos
2	music video codes
3	music downloads
4	music new
5	musicians friend
6	music codes
7	music download
8	music lyrics
9	music
10	mustang

(c) Reranked by semantic similarity.

Fig. 2. Ranked lists of QAC candidates for the prefix "mus".

case, methods from time series analysis are put to work to predict the query frequency (Cai et al., 2014b; Shokouhi & Radinsky, 2012; Whiting & Jose, 2014).

We propose to complement these time- and popularity-based QAC models with two methods based on lexical variations. First of all, popularity-based QAC models invariably count the query volumes following a strict query matching policy, thereby ignoring the contributions from so-called *homologous queries*, i.e., (1) queries with the same terms as the candidate query but in a different order and (2) queries that extend the candidate query. Formally, we define the following two types of homologous queries for a given query $q = (term_1, term_2, \ldots, term_m)$: (1) Given q, a *super query* of q is a query $s_q = (term_1, term_2, ..., term_m, term_{m+1}, \ldots, term_L)$ that extends q; (2) A *pseudo-identical query* for q is a query p_q that is a permutation of q. To a certain extent, homologous queries express similar search intents. For instance, at the time of writing (late 2014), for the two queries "*Chile SIGIR*" and "*SIGIR Chile*" (a pseudo-identical query of "*Chile SIGIR*"), the same SERPs should probably be returned. And the SERPs for "*Chile SIGIR*" and "*Chile SIGIR 2015*" (a super query of "*Chile SIGIR*") should probably overlap to a very large degree. Based on these examples, we hypothesize that it is advantageous to consider homologous queries as a context resource for QAC.

QAC features inferred from homologous queries are one important innovation that we study in this paper. A second way of using lexical variations for QAC that we propose is based on semantically related terms. As discussed in the literature, a user's search history usually reveals their search intent, often expressed by the queries or clicked documents. For instance, Shokouhi (2013) studies the similarity between a QAC candidate and previous queries in both the short-term and long-term history for reranking QAC candidates. And Jiang, Ke, Chien, and Cheng (2014) infer features from users' reformulation behavior for reranking QAC candidates. We exploit a similar intuition but operationalize it differently, by considering the semantic relatedness of terms in a QAC candidate and of terms from a QAC candidate and queries previously submitted in the same session. Let us give an example. Consider Fig. 1, which contains a session from the well-known AOL query log.

For the sake of the example, let us assume that we have not yet seen the last query (query 6, "music videos") and that it is in fact the initial segment "mus" of this query for which we want to recommend completions. A regular baseline based on query frequency is likely to rank the completion "music" first, as shown in Fig. 2a. If we consider the observed frequency of homologous queries for a candidate, we would return the list seen in Fig. 2b, which is a reranked version of the list in Fig. 2a. Clearly, the queries "music" and "music video" gain more benefits from homologous queries than others as they are now ranked at the top. But if we look in the user's search session (e.g., at query 4 and 5 in Fig. 1), we would see that "videos" is semantically closely related to earlier queries. Based on this insight, the query "music videos" in Fig. 2a is a more sensible completion. Considering the semantic similarity of terms both inside a candidate and of queries inside a session can generate another reranked QAC list shown in Fig. 2c. We can see semantically close queries, e.g., "music videos" and "music video codes," have now been to the top of the list.

Motivated by the examples above, we study the potential of homologous queries and semantic relatedness for improving state-of-the-art QAC methods. In particular, in addition to effective popularity-based features of QAC candidates, extended with time-based features and features of user reformulation behavior, we consider time- and popularity-based features for