Neurocomputing 167 (2015) 195–208

Contents lists available at ScienceDirect

Neurocomputing

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

Recommending high-utility search engine queries via a query-recommending model

JianGuo Wang [a,*], Joshua Zhexue Huang [b], Jiafeng Guo [c], Yanyan Lan [c]

[a] Shenzhen Key laboratory of High Performance Data Mining, Shenzhen Institutes of Advanced Technology Chinese Academy of Sciences, Shenzhen College of Advanced Technology, University of Chinese Academy of Sciences, 1068 Xueyuan Avenue, Shenzhen University Town, Shenzhen 518055, PR China

[b] College of Computer Science and Software Engineering, Shenzhen University, 3688 Nanhai Avenue, Shenzhen 518060, PR China

[c] Institute of Computing Technology, Chinese Academy of Sciences, 6 Kexueyuan South Road, Beijing 100190, PR China

ARTICLE INFO

Article history:
Received 17 February 2015
Received in revised form
23 April 2015
Accepted 27 April 2015
Communicated by M. Wang
Available online 13 May 2015

Keywords:
Query recommendation
Query log analysis
Query ranking
Recommendation methods

ABSTRACT

Query recommendation technology is of great importance for search engines, because it can assist users to find the information they require. Many query recommendation algorithms have been proposed, but they all aim to recommend similar queries and cannot guarantee the usefulness of the recommended queries. In this paper, we argue that it is more important to recommend high-utility queries, i.e., queries that would induce users to search for more useful information. For this purpose, we propose a query-recommending model to rank candidate queries according to their utilities and to recommend those that are useful to users. The query-recommending model ranks a candidate query by assessing the joint probability that the query is selected by the user, that the obtained search results are subsequently clicked by the user, and that the clicked search results ultimately satisfy the user's information need. Three utilities were defined to solve the model: query-level utility, representing the attractiveness of a query to the user; perceived utility, measuring the user's probability of clicking on the search results; and posterior utility, measuring the useful information obtained by the user from the clicked search results. The methods that were used to compute these three utilities from the query log data are presented. The experimental results that were obtained by using real query log data demonstrated that the proposed query-recommending model outperformed six other baseline methods in generating more useful recommendations.

1. Introduction

The explosive growth of web information has not only created a crucial challenge for search engines to handle large scale data, but also increased the difficulty for a user to manage his information need. It has become increasingly difficult for a user to compose a precise query to present his search intent. To alleviate users' reformulation burden, it is common practice for search engines to provide some types of query recommendations.

The existing query recommendation methods used in search engine all focus on recommending queries that are similar to user's initial query. This means that, for any given query q, its candidate queries $\{q_1, q_2, \dots, q_m\}$ presented for recommendation are ranked based on similarities between the candidate queries and q, i.e., $S(q, q_i)$ where S is computed from the log data of queries q and q_i. The top $k(<m)$ most similar queries are selected and recommended to the user and listed on the results page of q. Different log data are used to compute the similarity between two queries, such as commonly clicked URLs [1–3] and queries that were consecutively reformulated in the same search session [4–6].

However, the ultimate goal of query recommendation is to assist users to reformulate useful queries so that they can find desired information successfully and quickly. The existing similarity-based query recommendation approaches are apparently not directly toward this goal and cannot guarantee the usefulness of recommendations. For example, given a user's initial query "iPhone available time market", which tends to be used to find "when does it start to sell an iPhone in the market", he will be recommended "available market", "total available market", and "iPhone market analysis" by Google and "your available time", "times available to work", and "total available market" by Bing.[1] Obviously, these recommended queries are all similar to the initial query, especially in terms of text, but they are not useful in helping him find the release date of newest iPhone.

* Corresponding author.
E-mail address: wangjg@siat.ac.cn (J. Wang).

[1] All real examples are collected on 20 April 2015.

http://dx.doi.org/10.1016/j.neucom.2015.04.076

Therefore, in this paper, we argue that it is more important and beneficial to directly recommend queries with high utility, i.e., queries that can better satisfy users' information needs. Formally, query utility is defined as the useful information that a user can obtain from the search results of the query according to his initial information need. By recommending high utility query, we actually emphasize users' post-click satisfaction, i.e., whether users will be satisfied by the recommendation after clicking its search results. We argue that users' post-click satisfaction reflects the true effectiveness of query recommendation.

This paper proposes a probabilistic query-recommending model (QRM) with the aim of recommending high utility queries to search engine users. We also propose that the criteria for the usefulness of a candidate query are that the user selects it, chooses from its search results, and feel satisfied with the chosen outcome. Therefore, QRM functions by ranking a candidate query on the joint probability of three events, namely that the query is selected by the user, that the search results of the query are clicked by the user, and that the clicked search results meet the user's information requirement. Accordingly, we define three utilities: query-level utility, perceived utility, and posterior utility, to solve this model. Actually, the paradigm our approach used to rank candidate queries can be applied to other recommendation scenarios such as video recommendation, image recommendation and tag recommendation. This is another contribution of our work.

QRM model considers the attractiveness of queries to users, which has thus far been neglected by similarity-based approaches on query recommendation. We define a query-level utility to measure the attractiveness of candidate queries to users. The easier a user finds a recommendation to conceive of and understand, the more attractive it is. For example, given the initial query "online games", since the meaning of "online shooting games" is straightforward, it is easy for the user to understand. However, in contrast, some recommendations are difficult for the users to understand. The recommendation "neopets" tends to find an online virtual pet community, but users can hardly understand its intended purpose from the query itself without detailed interpretation. The query-level utility can significantly affect the users' behaviors in terms of choosing recommendations. In this paper, our approach recommends queries not only by perceived utility and posterior utility, but also by query-level utility. Compared to recommendations generated only by perceived utility and posterior utility, the recommendations generated using our approach can attract more users' clicks.

QRM model also considers the attractiveness of search results to users, which has also been neglected by similarity-based approaches. We define a perceived utility to measure the user's probability of clicking on the search results. Some other work such as [7] also studied the similar information, but they obtained perceived utility only by computing the proportion that the search results of a candidate query are clicked while we build a probabilistic model to learn the attractiveness of each clicked URL and then aggregate them to get the attractiveness of the search results. Compared to the simple method used by [7], we can learn better perceived utility with our probabilistic model.

QRM model can provide better post-click satisfaction to users. The post-click satisfaction reflects the effectiveness of query recommendation. However, traditional similarity-based query recommendation technologies have paid no attention to it. In contrast, our approach emphasizes the post-click satisfaction by employing posterior utility. Posterior utility measures the useful information that the user can obtain by the user from the clicked search results. Obviously, posterior utility contributes directly to post-click satisfaction. Compared to traditional similarity-based approaches, our approach has clear advantages. With high posterior utility, the clicked search results of recommended queries can provide more useful information for users.

To conduct query recommendation, we first compute the three utilities from the query log data for all candidate queries. We then rank the candidate queries according to three utilities, and recommend top k to the users. We have conducted a series of experiments on real query log data to evaluate the QRM model and compared its performance with seven other baseline methods in recommending high utility queries. The experimental results showed that the queries recommended by QRM were able to attract more clicks from the users and could entice users to select a larger number of relevant documents than those by six other baseline methods.

The remainder of this paper is organized as follows. Section 2 gives a brief review on related work. Section 3 defines the preliminaries on data representation. The QRM model and its solutions are introduced in Section 4. Section 5 presents experiments and results. In Section 6, we discuss the applications of our paradigm in other recommendation scenarios. Finally, conclusions and future work are given in Section 7.

2. Related work

In this section, we review three topics that are related to our work: similarity-based query recommendation, utility-based query recommendation and visual query recommendation.

2.1. Similarity-based query recommendation

Given a query q, the candidate queries $\{q_1, q_2, \ldots, q_m\}$ to q are ranked according to similarity function $S(q, q_i)$ where S is computed from the log data of queries q and q_i. The top $k(<m)$ most similar queries are recommended to the user and displayed at the bottom of the search results page of q. Different features extracted from log data were used to compute S.

First, the clicked URLs in the query log data were used to compute the similarity between two queries. This was done by first creating a query-URL bipartite graph from the clicked URLs in the query log data, which was then used to compute the similarities between queries. Ref. [1] used an agglomerative clustering algorithm to cluster queries and recommended similar queries in the same cluster. Ref. [8] applied two types of random walk processes to propagate the query similarity along the query-URL bipartite graph and this enabled them to obtain improved similarity scores between queries. Ref. [2] folded the query-URL bipartite graph to an affine graph and applied the hierarchical agglomerative clustering-based ranking method to recommend similar queries. Ref. [9] transformed the query-URL bipartite graph from undirected into a directed and applied a random walk to find queries similar to the initial query. Instead of a random walk, [3] used heat diffusion to model similar information propagation on the directed query-URL bipartite graph for query recommendation. Based on the features of queries extracted from the URL snippets, such as the title, summary of indexed documents, and user ID, [10] created a query-concept bipartite by using an agglomerative clustering algorithm to cluster queries for recommendation. This enabled them to obtain superior results than those in [1].

Search sessions, i.e., the sequence of queries issued by the same user in a given time period, were also used to compute the similarities of queries. Ref. [11] formulated search sessions as transactions of queries and applied association rule mining algorithms to find associated similar queries for recommendation. Ref. [12] represented each query as a vector of search sessions, the index of which recorded the occurrences of the query in that session. The similarity between two queries was computed from the two query vectors. Ref. [13] proposed to use a Mixture Variable