DEIM Forum 2009 A9-5

Rank Optimization of Personalized Search

Lin LI[†], Zhenglu YANG[††], and Masaru KITSUREGAWA[††]

† Dept. of Info. and Comm. Engineering, University of Tokyo, 4-6-1 Komaba, Meguro-ku, Tokyo, 153-8305 Japan
†† Institute of Industrial Science, University of Tokyo, 4-6-1 Komaba, Meguro-ku, Tokyo, 153-8305 Japan
E-mail: †{lilin,yangzl,kitsure}@tkl.iis.u-tokyo.ac.jp

Abstract Augmenting the global ranking based on the linkage structure of the Web is one of the popular approaches in data engineering community today for enhancing the search and ranking quality of Web information systems. This is typically done through automated learning of user interests and re-ranking of search results through semantic based personalization. In this paper, we propose a query context window (QCW) based framework for Selective uTilization of search history in personalized le$A$rning and re-Ranking (STAR). We conduct extensive experiments to compare our STAR approach with the popular directory-based search methods (e.g., Google Directory search) and the general model of most existing re-ranking schemes of personalized search. Our experimental results show that the proposed STAR framework can effectively capture user-specific query-dependent personalization and improve the accuracy of personalized search over existing approaches.

Key words Personalized search, search interests, hierarchical semantic similarity.

1. Introduction

Encoding human search experiences and personalizing the search result delivery through ranking optimization is a popular approach in recent data engineering field to enhancing the result quality of Web search and user exprience with the Web today. Although the general Web search today is still performed and delivered predominantly through search algorithms, e.g., Google's PageRank [17] based query independent ranking algorithms, the interests in improving global notion of importance in ranking search results by creating personalized view of importance have been growing over the recent years. We categorize the research efforts on personalized search into three classes of strategies: 1) query modification or augmentation [3], [26], 2) link-based score personalization [8], [9], [15], [17], [19], [22], and 3) search result re-ranking [4], [5], [12], [14], [26], [29], [30]. A general process of re-ranking is to devise efficient mechanisms to re-order the search result ranking using the global importance by personalized ranking criteria. Such criteria are typically derived from the modeling of users' search behavior and interests.

In this paper, we develop a rank optimization framework (STAR) that promotes Selective uTilization of search history for personalized le$A$rning and re-Ranking. Our STAR framework consists of three design principles and a suite of algorithms for learning and encoding user's short-term and long-term search interests and re-ranking of search results through a careful combination of recent and previous search histories. We show that even though short-term interests based personalization using the most recent search histories may be effective at times [15], [25], [26], it is generally unstable and fails to capture the changing behavior of the users. Furthermore, most of existing long-term interests based personalization using the entire recent and previous search histories fails to distinguish the relevant search history from irrelevant search history [4], [18], [30], making it harder to be an effective measure alone for search personalization.

Bearing in mind of these observations, our $STAR$ framework advocates three design principles for rank optimization. First, we devise a so-called query context window (QCW) model to capture the user's search behavior through a collection of her per-query based click-through data. Second, we develop a query-to-query similarity model to distinguish the relevant search memories of personalized search behavior from irrelevant ones in the QCW of each user, reducing the noises incurred by using either a recent fragment or the entire QCW. Third, we develop a fading memory based weight function to carefully combine the frequency of relevant search behavior (long term interests) with the most recent search behavior (short term interests). To show the effectiveness of our STAR framework in quality enhancement of personalized search, we propose length and depth based hierarchical semantic similarity metrics and compare the effectiveness of four re-ranking strategies: 1) naïve re-ranking that is query and time independent; 2) relevant search memory based re-ranking that is query dependent but time independent; 3) fading memory based re-ranking that is time dependent but query independent; and 4) hybrid re-ranking that is both query and time dependent. Our experiments show that the hybrid re-ranking scheme can effectively combine the previous and recent memories through a smooth and gradually fading memory based weighting function. More importantly, our experimental results show that the proposed STAR framework for personalized search and re-ranking can effectively capture user-specific query-dependent personalization preference and significantly improve the accuracy of personalized search over the popular directory-based search methods (e.g., Google Directory search) and the general model of most existing re-ranking schemes of personalized search. We also show the variations in performance among queries with different search goals.

The remainder of this paper is organized as follows. The overview of our STAR framework is presented in Section 2.. Then, we discuss building QCW based user profiles and designing re-rank strategies in Section 3. and Section 4. respectively. Experimental results will be given in Section 5.. Related works are reviewed in Section 6.. Finally, we conclude the paper in Section 7..

2. The STAR Framework Overview

The goal of the STAR framework is to design a semantic rich user profile model to capture the query context and the search behavior of each user and intelligently utilize such user profiles to enhance the quality of personalized search by effectively re-ranking of the search results returned from a general purpose search engine for a given query. Figure 1 gives a sketch of the STAR framework, consisting of three main components.

The first component is the text classification module that performs hierarchical Web page classification. The popular way is to classify the documents into a pre-exist directory-based ontology, such as Yahoo! Directory [11], ODP (http://dmoz.org) [27], to name a few. Some studies [1], [10], [16] preferred to building their own ontology. Thanks for the fact that hierarchical text classification is well studied in the field of text processing, in the first prototype design of our STAR framework, we directly utilize the classified search results from Google Directory search.

The second component is the context aware learning of user's search behavior. We utilize the per-query based click-through data to capture query dependent context and search behavior and develop the query context window (QCW) model to encode such leaning process. By automatically generating QCW based user query profiles, the user learning module automatically captures the query dependent context of user search behavior. For example, our approach focuses on the user's visited search results (Web pages) which supply us with not only what kind of content a user is interested in (topics) but also how much the user is interested in them (click frequency).

The third component is the query and time dependent, hybrid re-ranking scheme that produces a new user-centric, query dependent rank list for each user query through three step process. First, it selects the relevant click records from the entire QCW of a user through the query-to-query similarity analysis. Second, it combines the recent search memories with the previous search memories through applying a fading memory based weighting function over the selected QCW click records of a user. Finally, it employs hierarchical semantic similarity measures to compute the personalized ranking of the search results returned from a general search engine. In the subsequent sections we will focus on the technical detail of the user learning module and the re-rank module.

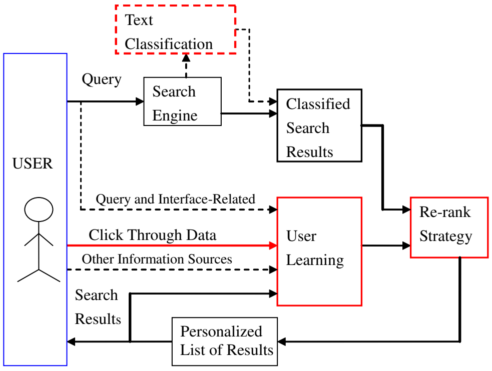


Fig. 1 Overview of the STAR framework

3. QCW Based User Learning Module

Our STAR framework devises the query context window (QCW) to encode the user specific and query dependent search behavior. Given a user, her query context window consists of m query-dependent context click records, denoted as $u_1, u_2, \ldots, u_m$. Each click record in the QCW is composed of the submitted query, the topics associated with the click search results, the click frequency of each topic, and the returned search results of the given query. The topics are extracted from Google Directory, structured as a hierarchical tree, so that each click record has its own tree. This topic tree records the click behavior of a user on a specific query, which can tell us what kind of topics a user is interested in. The click frequency is an interest score representing how much the user is interested in this topic. The topic trees in all click records store user's interests. To help us choose relevant QCW click records given an input query, the search results $(P1, P2, \ldots, PN)$ responding to a past query are stored in the Search Result Buffer (SRB). The technical details of click record selection are in the next section. Moreover, we implement each QCW as a queue. The tail of the queue holds most recently requested queries, while its head holds the least recently requested queries. When a new query is submitted, the corresponding record is added to the tail of the queue and the user model (QCW) is updated accordingly. This queue keeps the chronological order of different click records, which can easily differ the recent and old search histories for re-ranking strategies.

Figure 2 shows an example of QCW with three context records, each corresponds to one query and its context encoding of the query dependent click-through data. For example, a user inputs a query "Disneyland" to Google Directory search engine, and then she clicks some search results. Record 3 in Figure 2 will store the input query "Disneyland" as a root node followed by the clicked topics. The search results are kept in the SRB. Node F is represented by the $[ThemeParks, 6]$ which means the user has clicked some search results associated with the topic "$ThemeParks$" six times

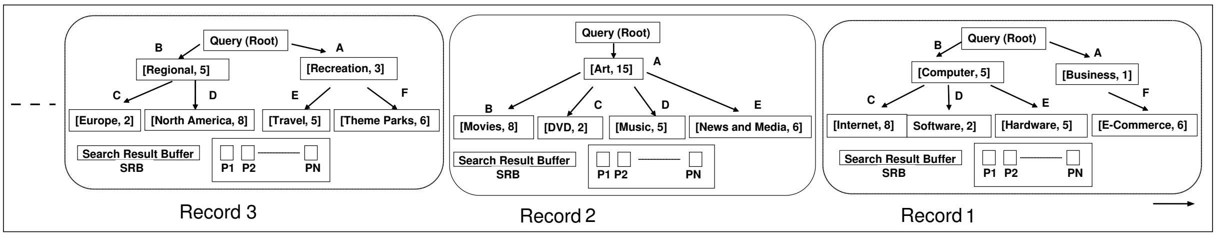


Fig. 2 Query Context Window: click records are queued up in a chronological order

in this search. In addition, for each topic, we store the top four depth of its full path in Google Directory in a record. For example, the node F is actually stored as the $[\backslash Recreation \setminus ThemeParks]$.

4. The Re-rank Module

The QCW based re-ranking module needs to address three key challenges. First, how to select relevant context records from the entire QCW given a user query. In other words, given a user and her current query, do all the context records in the QCW of the user equally reflect the user's current search interest? If not, which click records have higher probability in reflecting the current interests? Obviously if the user's current search interest is related to her short-term interests, the most recent context records are most likely to be useful. If the current user's search interest happens to be related also to her long-term interests, which means that the user has clicked the related topics often in the past, then both previous and recent memories are useful. Thus the second challenge is whether all the selected query-relevant context records play the same role in re-ranking the search results of the current query. If not, how to determine the weights of the selection of query-relevant context records? Finally, given the weighting function that combines previous and recent memories relevant to the current query, what is the most effective mechanism to compute the similarity of the current query results with the selected QCW click records and how should we re-order the search results according to the similarity measures? In the subsequent sections, we will address each of these three challenges in detail.

In the remaining of the paper, we use calligraphic upper-case alphabets to represent sets. The elements of a set are denoted by lower case alphabets. For example, $\mathcal{U}$ is the set of click records in QCW and u_i is an element (click record) of $\mathcal{U}$. $|\mathcal{U}|$ is the cardinality of the set $\mathcal{U}$.

4.1 Selecting Relevant Click Records

Given a new input query, we first select the relevant QCW click records where the encoded queries are similar to the current input query by using a query-to-query similarity measure. Estimating the similarity (relatedness) between queries has a long history in traditional Information Retrieval [6], [21], [32]. It is still hot and active in various topics of Web Information Retrieval [2], [7], [31]. One of the lessons learned in the Information Retrieval area is that there are various similarity and specificity measures as well as various ways of combining them. Up to now it has not been possible to prove that any of these measures outperforms all others in a large set of experiments [33].

In our case, it is intuitive to use the term overlap between two queries as a similarity measure (e.g., previous queries having common terms with the input query are naturally recommended as alternatives). However, only a couple of keywords are used in defining Web queries [31]. It is possible that queries may be identical or phrased differently with different terms but for the same information needs. Consider the example of two queries, "IRS (Internal Revenue Service) form" and "file taxes online". Although they have no terms in common, both of them concern the application of filing taxes. The similarity between the two queries can be induced from the overlap of the two lists of search results (URLs) returned. Clearly, the query-result-vectors present a better similarity metric than query term-vectors [21]. As thus, using the query result URL to encode each search result, we formally define the query-to-query similarity measure as follows:

$$Q(q^{u_i}, q_{in}) = \frac{\mathcal{SRB}^{u_i} \cap \mathcal{P}^{q_{in}}}{\mathcal{SRB}^{u_i} \cup \mathcal{P}^{q_{in}}}. \tag{1}$$

Given the past query q^{u_i} in the click record u_i of a QCW-based user profile and the current input query q_{in}, we can get the URL set of search results of q^{u_i} from the search result buffer $\mathcal{SRB}^{u_i}$ of the click record u_i, and the URL set of search results $\mathcal{P}^{q_{in}}$ of q_{in} from the current search. The similarity between the two queries is estimated to the fraction of the intersection of the two URL sets (i.e., $\mathcal{SRB}^{u_i}$ and $\mathcal{P}^{q_{in}}$). Our query-to-query similarity measure states that the more URLs two queries have in common in their result sets, the more similar they are. The value of the defined similarity between two queries lies in the range [0, 1]: 1 if they have exactly the same URLs, and 0 if they have no URLs in common. In our experiments, URL similarity is measured by their host name. Though our URL-based query to query similarity measure is simple and intuitive, our experiments show that it can effectively extract relevant click records from QCW. We would like to note that our STAR framework can easily incorporate other similarity and specificity measures. Due to the space constraint and the fact that this paper focuses on combining query-to-query similarity with long and short term memory functions to improve the re-ranking of search results, we omit the further discussion on more complex query to query similarity measures.

4.2 Weighing Relevant Click Records

Equation 1 answers the first question that is how to select relevant context records from the entire QCW given a user query, which can select semantic similar queries, and then the user context with these queries in QCW is relevant to the current user's information need phrased as the input query. Here, we address the second challenge that is whether all the selected query-relevant context records play the same role in search results re-ranking. These selected click records are the collection of user's previous and recent search behaviors which reflect her interests. We assume that the user's interests will gradually decay as time goes on, so we assign more weights to more recent QCW click records and decreasing weights to older QCW click records to further improve the accuracy of the personalized search using a fading memory based weight function, defined as follows:

$$F(u_i) = e^{-\frac{log2}{hf \cdot |\mathcal{U}|} \cdot (|\mathcal{U}| - i)}\,, \tag{2}$$

where hf is a half fading parameter. In our experiments, hf is set in the range [0.1, 1]. After the click record u_i is selected as relevant according to the similarity between its q^{u_i} and the current input query q_{in}, its effect on the quality of personalized search (i.e., $F(u_i)$) depends on its temporal order. For example, if the click record u_i is located in the middle of the whole QCW (i.e., in the center of the oldest memory and the most recent memory, namely i=$|\mathcal{U}|$/2), its effect will be reduced by 1/2 when hf=0.5. With increasing the value of hf, the rate of fading becomes slow and the weights on previous memories increase. In our STAR framework, this fading memory function is a key metric to unify the user's long-term and short-term interests encoded in the QCW click records by assigning different weights to these click records appearing in different temporal order.

4.3 Capturing Search Interests

We have defined the query-to-query similarity and the fading memory based weight function for selecting QCW click records related to the current search interest of a user. After the relevant QCW click records and their weights are determined, the topics in these QCW click records are reflecting the user's current search interests. Now we can devise a re-ranking mechanism to re-order the search results by putting those that are more similar to the selected topics closer to the top of the final re-ordered rank list. In other words, given one of search results of the input query q_{in}, (e.g., $p_k^{q_{in}}$, the kth search result) and the set of weighted relevant QCW click records, we calculate the similarity between them. The higher similarity score $p_k^{q_{in}}$ is getting in comparison with the results of historical queries of the same user, the higher position it will be placed in the final ranked result list. In our STAR framework, the topics in the relevant QCW click records are structured in a semantic concept hierarchy as shown in Figure 2. Hierarchical similarity measures can be used to assess the similarity between the related topics and the search results of the given query.

There are a set of topics $\mathcal{T}^{u_i}$ in the click record u_i and each search result is classified to a topic, so we first compute the hierarchical similarity score between each topic $t_j^{u_i}$ in $\mathcal{T}^{u_i}$ and the topic of $p_k^{q_{in}}$, and then combine all the scores of the topics in $\mathcal{T}^{u_i}$. We exploit the structural similarity among the related topics by considering the length-depth hybrid hierarchical similarity measures (i.e., Equation 7-8) [13] since the length based (i.e., Equation 3-4) and the depth based (i.e., Equation 5-6) have their own shortcomings.

Let h be the depth of the subsumer (the deepest node common to two nodes), l be the shortest path length between two topics, and M be the maximum depth of topic directory possessed by a QCW click record.

1) Length-based Topic Similarity Measure

The length-based measure is intuitive and considers the shortest path length between two topics (nodes) alone. We present the length-based measure in its linear and exponential form as follows:

$$L1 \equiv HS(t_j^{u_i}, p_k^{q_{in}}) \;=\; 2 \cdot M - l\,, \tag{3}$$

$$L2 \equiv HS(t_j^{u_i}, p_k^{q_{in}}) \;=\; e^{-0.25 \cdot l}\,. \tag{4}$$

Clearly, Equation 3 is a linear function of the shortest path length between two topics (nodes) and Equation 4 measures the same using a nonlinear function. Consider the example in Figure 2, given node A and node B in Record 1 of Figure 2, the length between them is 2, so their similarity values computed by $L1$ and $L2$ are 6 and 0.6065 respectively. The main drawback of the naïve length-based similarity is that it overlooks the depth of the subsumer.

2) Depth-based Topic Similarity Measure

An obvious alternative topic similarity measure is to consider the depth of subsumer information instead. The following two depth based equations use linear and nonlinear functions respectively to measure the topic similarity of the current user search query with her previous queries, defined as:

$$D1 \equiv HS(t_j^{u_i}, p_k^{q_{in}}) \;=\; 0.05 \cdot (2 \cdot M - l) + h\,, \tag{5}$$

$$D2 \equiv HS(t_j^{u_i}, p_k^{q_{in}}) \;=\; \frac{e^{0.15 \cdot h} - e^{-0.15 \cdot h}}{e^{0.15 \cdot h} + e^{-0.15 \cdot h}}\,. \tag{6}$$

Although Equation 5 uses the length in its definition, it gives relatively much heavier weight on the depth (1 vs. 0.05). Therefore, we classify it into depth-based metrics. Equation 6 is the transformation of the depth of the subsumer through a nonlinear function. Using node A and node B in Record 1 as an example again, the depth of their subsumer is 1 (i.e., $Query(Root)$ node), so the similarity values computed by $D1$ and $D2$ are 1.3 and 0.1489 respectively. It is easy to understand that using the depth information alone is also not optimal. Consequently, we consider the combination of length and depth used in the hierarchical semantic similarity.

3) Length and Depth Combined Topic Similarity Measure

Motivated by the strength and weakness of length and depth based approaches, we use a careful combination of depth and length based similarity measure defined as follows:

$$C1 \equiv HS(t_j^{u_i}, p_k^{q_{in}}) \;=\; \frac{2 \cdot h}{l + 2 \cdot h}\,, \tag{7}$$

$$C2 \equiv HS(t_j^{u_i}, p_k^{q_{in}}) \, = \, e^{-0.2 \cdot l} \cdot \frac{e^{0.6 \cdot h} - e^{-0.6 \cdot h}}{e^{0.6 \cdot h} + e^{-0.6 \cdot h}} \, . \qquad (8)$$

Equation 7 is a simple linear transformation function of the length and the depth, while Equation 8 transfers the length and the depth by a nonlinear function and then combines them by multiplication. The similarity values between node A and node B in Record 1 calculated by $C1$ and $C2$ are 0.36 and 0.5 respectively. We expect that the combination can work well for common cases since using depth or length alone has its shortcomings in some conditions. The parameters used in these equations weighs the depth and length. In this paper, we do not address how to get the best weights which is well discussed in [13].

The above six equations define the similarity between two topics (nodes), which we use, in combination with the query to query similarity measure, to capture the similarity of a user's current search behavior with her previous search behavior. A QCW click record may record more than one topic depending a user's click behavior. We further define the similarity between a QCW click record u_i and a search result $p_k^{q_{in}}$ as:

$$S(u_i, p_k^{q_{in}}) = \frac{1}{|\mathcal{T}^{u_i}|} \sum_{t_j^{u_i} \in \mathcal{T}^{u_i}} \frac{HS(t_j^{u_i} . p_k^{q_{in}}) \cdot c_j^{u_i}}{\sum_{c_j^{u_i} \in \mathcal{C}^{u_i}} c_j^{u_i}} \, , \qquad (9)$$

where each topic $t_j^{u_i}$ in $\mathcal{T}^{u_i}$ is weighted by its corresponding $c_j^{u_i}$ representing the interest score of the topic j in u_i. The larger an interest score is, the more interested the user is in one topic. In this paper, we use click frequency as the interest score. We obtain a normalized version by dividing this score by the sum of the interest scores of all the topics in u_i. Then we sum all the normalized weighted hierarchical similarity scores of the topics in u_i with a search result. Moreover, the number of topics stored in each click record depends on user's click behavior. The more clicked topics in a click record, the click record may gain larger similarity scores. Due to the collective strength of these topics, although each clicked topic in it may have a relatively small score, the sum of these scores will effect the re-ranking quality. We further normalize the sum of hierarchical similarity scores through dividing it by the number of topics stored in a click record $|\mathcal{T}^{u_i}|$.

4.4 QCW Based Re-ranking

In this section we will describe how to use all the selected relevant QCW click records of the given user to re-order the search results of her current query. To provide a better understanding of the effects of different factors on the quality of our hybrid re-ranking optimization, we consider rhe following four re-ranking strategies.

1) Strategy 1 $-$ Query and time independent scheme

$$S_1(\mathcal{U}, p_k^{q_{in}}) = \frac{1}{|\mathcal{U}|} \sum_{u_i \in \mathcal{U}} S(u_i, p_k^{q_{in}}) \, . \qquad (10)$$

"Strategy 1" is query and time independent, a naïve strategy, which defines an equal weighting strategy. Click records of different past queries are assigned equal weights regardless of the current input query. The similarity scores of past queries with a search result $p_k^{q_{in}}$ are summed together and divided by the number of click records ($|\mathcal{U}|$) in $\mathcal{U}$. There is no selection of relevant click records and no temporal order based weighting in "Strategy 1", which means all the past histories (click records) are related to a user's current query. This simple weighting schemes suffers from the problems that it produces a global but weak description of user's current search interests. As we discussed in Section 1., the entire QCW includes noisy memories unrelated to the current query and only those that are related to the current search interests are important. We should therefore assign much weight on them, and ignore other noisy memories of QCW. Most of re-ranking based Web search personalization methods in the literature [4], [5], [12], [14], [18], [26], [27], [30] have commonly used all available user context to get some improvement. "Strategy 1" can represent the general idea of these methods, compared with the following three strategies.

2)Strategy 2 $-$ Query dependent scheme

$$S_2(\mathcal{U}, p_k^{q_{in}}) = \frac{1}{|\mathcal{U}|} \sum_{u_i \in \mathcal{U}} Q(q^{u_i}, q_{in}) \cdot S(u_i, p_k^{q_{in}}) \, . \qquad (11)$$

We define the "Strategy 2" as a query dependent and time independent strategy, which is selective about which click records of QCW to use according to the current query q_{in} by using the query-to-query similarity $Q(q^{u_i}, q_{in})$ to weight these click records. Tan et al. [29] did preliminary discussion on query-dependent selection of user profile. However, their work is in the context of only exploiting long-term search histories of users and ignores the changes of user's interests with time. For example, if a Web user used the query "Python" to get information on snake ago, and now she is interested in Python programming language and search related programming skills on the Web. When she inputs "Python" on a search engine again, it is reasonable to think that the recent search histories on Python in the field of computer science are more important than the previously clicked Web pages on snakes.

3)Strategy 3 $-$ Time dependent scheme

$$S_3(\mathcal{U}, p_k^{q_{in}}) = \frac{1}{|\mathcal{U}|} \sum_{u_i \in \mathcal{U}} F(u_i) \cdot S(u_i, p_k^{q_{in}}) \, . \qquad (12)$$

"Strategy 3" strengthens recent memories and weakens the effect of previous memories by applying the fading function $F(u_i)$ to each QCW click record without the selection of relevant contexts in terms of the input query like "Strategy 2". If hf is set to a very small value, the previous memories cannot have an influential effect on re-ranking. Then for simplicity we can think that in this case the quality of "Strategy 3" is largely based on the recent memories and ignores the previous memories even if these old memories are related to the current query. Researches [15], [25], [26] emphasize that the most recent search is most directly close to the user's current information need, which can be regarded as a special case where hf is close to zero in “Strategy 3”. The retrieval quality improved by their approaches are heavily relied on accurately detecting session boundaries, such that only those searches within the session are used as relevant search histories. Properly finding session boundaries is non-trivial, so they determined the session boundary by manual or using 30 minutes, a well-known threshold. Based on these discussions, we think the combination of query-dependent and time-dependent strategy would be more effective in a general Web search.

4) Strategy 4 − Query and time dependent scheme

$$S_4(\mathcal{U}, p_k^{q_{in}}) = \frac{1}{|\mathcal{U}|} \sum_{u_i \in \mathcal{U}} F(u_i) \cdot Q(q^{u_i}, q_{in}) \cdot S(u_i, p_k^{q_{in}}) . \quad (13)$$

“Strategy 4” is both query and time dependent, a hybrid strategy. As we know, users have their own characteristics of search behavior. If a user always likes phrasing general queries which cover a number of various topics, her whole search histories would be useful like “Strategy 1”. If a user major in some field, e.g., Information Retrieval, likes to frequently search the latest technical reports , this interest can be captured and then the search histories on IR can be extracted from her whole histories using “Strategy 2”. If all the searches of a user are related to different information needs and there is no relatedness between these information needs, we cannot learn her long-term interests which should be consistent and accumulated by experiences over a long time period. In this case, recent-memory based strategy like the time dependent “Strategy 3” is effective. If the query input by a user may have no relevance with her search history, the current general search engines are doing a good job and then no personalization is needed. To handle the most general case where we have many kinds of Web users and users will show different search behaviors, “Strategy 4” is designed to select relevant click records $Q(q^{u_i}, q_{in})$, but also assign greater weights to the more recent click records $F(u_i)$.

Given one of the four strategies, a new relevant score will be calculated for each of search results. We output the list of the search results in order of their assigned scores. We use a concrete example to further discern the four re-ranking strategies. Recall Figure 2, given the query “Disneyland”, we assume that the relevant click records are Record 2 and Record 3 in Figure 2 which includes topics like “Theme Parks”, “Travel”, “Music” and so on. This assumption means that the query-to-query similarity scores (Equation 1) of the two click records are not zero. “Strategy 1” will use all the click records to re-rank search results, so that if there are unrelated Web pages about “Computer” or “Business” in the search results, they will be considered related to the input query, thus lowering the retrieval accuracy. “Strategy 2” with the help of Equation 1 can identify Record 2 and Record 3 as relevant click records, thus expelling the noisy Record 1 from the re-ranking scheme. Although “Strategy 3” gives less weights to Record 1 than Record 2 and Record 3, Record 1 is not the relevant click record. Thus, this strategy will be interfered by Record 1 and cannot benefit much from the relevant Record 2 and Record 3. Based on the selected relevant click records, “Strategy 4” can not only select the relevant Record 2 and Record 3, but also assign greater weight on Record 3 than Record 2 by using Equation 2 because Record 3 is more recently added into QCW than Record 2. In the following experiments, we will evaluate the effectiveness of the four re-rank strategies utilized in our STAR framework.

5. Experiments

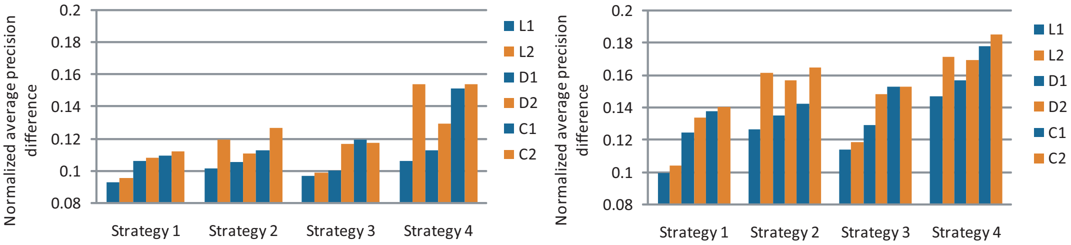
5.1 Experiment Setup

The goal of this paper is to achieve a personalized ranking by scoring the similarity between a user profile and the returned search results. Instead of creating our own Web search engine, we retrieve results from Google Directory search engine and use them as a baseline in the following evaluation. Moreover, as discussed in [23], informational queries (IQ) are such queries where the user does not have a special page in mind and intends to find out Web pages related to a topic. We further classified the goal of IQ into three categories: new IQ, semi-new IQ, and repeated IQ. A query is a new IQ if a user never searches such a topic before. It means that we cannot get the relevant search histories. A semi-new IQ has similar topical contents with some of the user’s search histories. A repeated IQ refers to the query by which the user has already obtained the desired information, and is searching for it again. The following experiments will evaluate the performances of the semi-new and repeated IQs since our STAR framework wants to use the previous relevant memories to enhance the current search. For new IQs, collaborative information retrieval will be an interesting direction in our future work. The evaluation of our framework is a challenge because currently there are no suitable query log data sets as a public benchmark. We created our own two data sets: a real one and a synthetic one.

The real data set [12] was collected over a ten-day period (From October 23rd, 2006, to November 1st, 2006). Twelve users are invited to search through our framework and judge whether the clicked results are relevant or not. They are graduate students (5 females and 7 males). Users were asked to input search queries related to their professional knowledge in the first four days, and search queries related to their hobbies in the next three days. Then, in the last three days, each user is requested to repeat some searches with the queries entered in the previous days. We got a log of about 300 queries averaging 25 queries per subject and about 1200 records of the pages the users clicked in total. The size of this real data set is relatively small because the click data collection and users’ judgments are labor intensive.

5.2 Evaluation Measure

Precision is a standard measure in the field of information retrieval. We calculate a normalized precision because we test 30 queries. First, the average precision of a single query is defined



(a) Semi-new IQ (b) Repeated IQ

Fig. 3 The improvement difference value of our strategy on real data set

Table 1 The improvement percentage of our strategy

Semi-new IQ (%)					Repeated IQ (%)				
Measure	Strategy 1	Strategy 2	Strategy 3	Strategy 4	Measure	Strategy 1	Strategy 2	Strategy 3	Strategy 4
L1	21.17	23.10	22.04	**24.11**	L1	40.27	51.43	46.18	**59.67**
L2	21.72	27.07	22.50	**35.00**	L2	42.36	65.43	47.99	**69.53**
D1	24.14	23.93	22.83	**25.60**	D1	50.42	54.85	52.41	**63.54**
D2	24.61	25.61	26.49	**29.34**	D2	54.19	63.67	60.20	**68.54**
C1	24.92	25.71	**27.13**	**34.35**	C1	55.95	57.72	61.92	**72.16**
C2	**25.50**	**28.78**	**26.64**	**34.88**	C2	**56.93**	**66.71**	**61.92**	**75.00**

as:

$$P@N = \frac{\sum_{p_k^{q_{in}} \in \mathcal{P}^{q_{in}} \& k \leqq N} sp_k^{q_{in}}}{N}, \quad (14)$$

$$AvgP = \frac{\sum_{k=1}^{N} P@k \cdot sp_k^{q_{in}}}{\sum_{p_k^{q_{in}} \in \mathcal{P}^{q_{in}}} sp_k^{q_{in}}}. \quad (15)$$

$sp_k^{q_{in}}$ is the user's judgment on the k_{th} search result of the query q_{in} and it has two values, 0 for "irrelevant" and 1 for "relevant". $P@N$ evaluates the performance at a given cut-off rank, considering only the top N results returned by the system (e.g., N=15). For a single query, average precision is defined as the average of the $P@k$ values for all relevant documents (k=1, $\cdots$, N). We divide the sum of the $AvgP$ values of all the queries by the number of testing queries (i.e., 30), which represents the $normalized\ average\ precision$ and is used as one of our evaluation measures.

5.3 Results and Discussions

In the real data set, the queries in the last three days are regarded as repeated IQs. The first seven-day click-through data is divided into two parts (odd-day and even-day) as semi-new informational searches. One is for setting up the QCW user profile and the other is for re-ranking search results based on the learned user profile, and then the two parts are exchanged to run the evaluation once again. Here, we set M to be 5 and $hf \cdot |\mathcal{U}|$ to be 20.

In Figure 3 and Table 1, we summarize the performance of the proposed four re-rank strategies according to different hierarchical semantic measures. "Normalized average precision difference" means the difference value between our strategy and the baseline and "Normalized average precision % " represents the improvement percentage of our strategy over the baseline. The experimental results show that the proposed user-context aware re-rank strategies are more effective than the baseline. "Strategy 1", representing the general idea of most existing personalized re-ranking schemes, is inferior to other three strategies. Among the proposed four re-rank strategies, the "Strategy 4" broadly shows the best performance. The "Strategy 2" with selective utilization of user profiles, averagely produces better results than the "Strategy 1" and "Strategy 3". In Figure 3 and Table 1 the improvement of repeated IQs in is more obvious than those of semi-new IQs. The larger improvement of repeated IQs shows that our re-rank strategies can effectively retrieve the Web pages previously clicked by users since these queries have been submitted before and user's click behavior has been stored in our QCW.

Moreover, in Figure 3 and in Table 1 we observed that the similarity measures using nonlinear transformation function (i.e., $L2$, $D2$, and $C2$ shown in orange columns) generally produce better performance that the similarity measures using linear transformation (i.e., $L1$, $D1$, and $C1$ shown in blue columns). $C2$, the combination of length and depth with nonlinear transformation, generates the highest improvement among all the six measures. Length-based nolinear measure $L2$ largely increases its performance in "Strategy 2" and "Strategy 4". The two strategies using Equation 1 select relevant click records given an input query and filter some irrelevant records. Therefore, using length information alone can gain comparable performance with $C2$. In a word, "Strategy 4" with $C2$ produces the largest improvement, e.g., its improvements over baseline are 34.88% and 75% for semi-new IQs and repeated IQs respectively.

From the results, we can say that re-ranking of search results through semantic based personalization actually can enhance the general search. We also confirm that there are two critical factors: (1) the query-to-query similarity which captures the long term search interests of a user (query dependent), and (2) the most recent search interest which reflects the short term search behavior of a user (time dependent). The two factors indicate that both short-term and long-term memories contribute to the improvement.

6. Related Work

In this section we give a brief overview of some related works in the literature of personalized search. There are two kinds of context information we can use to model search experience and capture user search histories. One is short-term context, which emphasizes that the most recent search is most directly close to the user's current information need [15], [25], [26]. Successive searches in a session usually have the same information need. Detecting a session boundary, however, is a difficult task. The other is long-term context, which generally assumes that users will hold their interests over a relatively long time. It means that any search in the past may have some effect on the current search [4], [14], [18], [30]. These studies commonly used all available contexts as a whole to improve the search result quality and ranking. Preliminary discussion on this problem in [29] is in the context of only exploiting long-term search history of users. In addition, several researchers have used taxonomic hierarchy (a simple directory-based ontology) is used to represent user's interests in the Web search [4], [10], [16], [18], [20], [24]. However, very few has taken into consideration the hierarchical structure of the directory-based ontology when calculating similarity values between current search of a user and her search history. Chirita et al. [4] using hierarchical semantic measure, however, required users to manually select topics they are interested in. A unique characteristics of our STAR framework is the development of a selective use of personalized search history and a combination of long term and short term user search histories in rank optimization of personalized search.

7. Conclusions

We presented a STAR framework for selective utilization of user search behaviors for personalized learning and re-ranking. We designed a novel user search profile called query context window (QCW) to record the search behavior of a user. We developed a query-to-query similarity model and the fading memory based weight function. We showed how our STAR framework carefully chose and weighed the relevant click records as useful user context given an input query and how we applied hierarchical semantic similarity measures in our re-rank strategies. The experimental results show that STAR approach to personalized search and re-ranking approach can effectively learn user-specific query-dependent personalization preference and significantly improve the accuracy of personalized search over the most existing personalized re-rankings.

References

[1] M. S. Aktas, M. A. Nacar, and F. Menczer. Personalizing PageRank based on domain profiles. In *WEBKDD*, 83–90, 2004.

[2] D. Beeferman and A. L. Berger. Agglomerative clustering of a search engine query log. In *KDD*, 407–416, 2000.

[3] P.-A. Chirita, C. S. Firan, and W. Nejdl. Personalized query expansion for the web. In *SIGIR*, 7–14, 2007.

[4] P. A. Chirita, W. Nejdl, R. Paiu, and C. Kohlschütter. Using ODP metadata to personalize search. In *SIGIR*, 178–185, 2005.

[5] Z. Dou, R. Song, and J.-R. Wen. A large-scale evaluation and analysis of personalized search strategies. In *WWW*, 581–590, 2007.

[6] L. Fitzpatrick and M. Dent. Automatic feedback using past queries: Social searching? In *SIGIR*, 306–313, 1997.

[7] N. S. Glance. Community search assistant. In *IUI*, 91–96, 2001.

[8] T. H. Haveliwala. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *IEEE Trans. Knowl. Data Eng.*, 15(4):784–796, 2003.

[9] G. Jeh and J. Widom. Scaling personalized web search. In *WWW*, 271–279, 2003.

[10] H. R. Kim and P. K. Chan. Learning implicit user interest hierarchy for context in personalization. In *IUI*, 101–108, 2003.

[11] Y. Labrou and T. W. Finin. Yahoo! As an ontology: Using Yahoo! categories to describe documents. In *CIKM*, 180–187, 1999.

[12] L. Li, Z. Yang, B. Wang, and M. Kitsuregawa. Dynamic adaptation strategies for long-term and short-term user profile to personalize search. In *APWeb/WAIM*, 228–240, 2007.

[13] Y. Li, Z. Bandar, and D. McLean. An approach for measuring semantic similarity between words using multiple information sources. *IEEE Trans. Knowl. Data Eng.*, 15(4):871–882, 2003.

[14] F. Liu, C. T. Yu, and W. Meng. Personalized web search for improving retrieval effectiveness. *IEEE Trans. Knowl. Data Eng.*, 16(1):28–40, 2004.

[15] Y. Lv, L. Sun, J. Zhang, J.-Y. Nie, W. Chen, and W. Zhang. An iterative implicit feedback approach to personalized search. In *ACL*, 585–592, 2006.

[16] N. Nanas, V. S. Uren, and A. N. D. Roeck. Building and applying a concept hierarchy representation of a user profile. In *SIGIR*, 198–204, 2003.

[17] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[18] A. Pretschner and S. Gauch. Ontology based personalized search. In *ICTAI*, 391–398, 1999.

[19] F. Qiu and J. Cho. Automatic identification of user interest for personalized search. In *WWW*, 727–736, 2006.

[20] H. rae Kim and P. K. Chan. Personalized ranking of search results with learned user interest hierarchies from bookmarks. In *WEBKDD*, 32–43, 2005.

[21] V. V. Raghavan and H. Sever. On the reuse of past optimal queries. In *SIGIR*, 344–350, 1995.

[22] M. Richardson and P. Domingos. The intelligent surfer: Probabilistic combination of link and content information in pagerank. In *NIPS*, 1441–1448, 2001.

[23] D. E. Rose and D. Levinson. Understanding user goals in web search. In *WWW*, 13–19, 2004.

[24] V. Schickel-Zuber and B. Faltings. Inferring user's preferences using ontologies. In *AAAI*, 1413–1418, 2006.

[25] X. Shen, B. Tan, and C. Zhai. Context-sensitive information retrieval using implicit feedback. In *SIGIR*, 43–50, 2005.

[26] X. Shen, B. Tan, and C. Zhai. Implicit user modeling for personalized search. In *CIKM*, 824–831, 2005.

[27] M. Speretta and S. Gauch. Personalized search based on user search histories. In *WI*, 622–628, 2005.

[28] K. Sugiyama, K. Hatano, and M. Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. In *WWW*, 675–684, 2004.

[29] B. Tan, X. Shen, and C. Zhai. Mining long-term search history to improve search accuracy. In *KDD*, 718–723, 2006.

[30] J. Teevan, S. T. Dumais, and E. Horvitz. Personalizing search via automated analysis of interests and activities. In *SIGIR*, 449–456, 2005.

[31] J.-R. Wen, J.-Y. Nie, and H. Zhang. Query clustering using user logs. *ACM Trans. Inf. Syst.*, 20(1):59–81, 2002.

[32] J. Xu and W. B. Croft. Query expansion using local and global document analysis. In *SIGIR*, 4–11, 1996.

[33] J. Zobel and A. Moffat. Exploring the similarity space. *SIGIR Forum*, 32(1):18–34, 1998.