Expert Systems with Applications xxx (2011) xxx–xxx

Contents lists available at ScienceDirect

Expert Systems with Applications

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

How do users describe their information need: Query recommendation based on snippet click model

Yiqun Liu *, Junwei Miao, Min Zhang, Shaoping Ma, Liyun Ru

State Key Lab of Intelligent Technology & Systems, Tsinghua National Laboratory for Information Science and Technology, CS &T Department, Tsinghua University, Beijing 100084, PR China

ARTICLE INFO

Keywords:
Web data mining
Query recommendation
User behavior analysis
Click-through data

ABSTRACT

Query recommendation helps users to describe their information needs more clearly so that search engines can return appropriate answers and meet their needs. State-of-the-art researches prove that the use of users' behavior information helps to improve query recommendation performance. Instead of finding the most similar terms previous users queried, we focus on how to detect users' actual information need based on their search behaviors. The key idea of this paper is that although the clicked documents are not always relevant to users' queries, the snippets which lead them to the click most probably meet their information needs. Based on analysis into large-scale practical search behavior log data, two snippet click behavior models are constructed and corresponding query recommendation algorithms are proposed. Experimental results based on two widely-used commercial search engines' click-through data prove that the proposed algorithms outperform practical recommendation methods of these two search engines. To the best of our knowledge, this is the first time that snippet click models are proposed for query recommendation task.

1. Introduction

Although search engines gain much success in helping users to obtain Web information in a most convenient way (http://www.comscore.com), this "keyword-based" user interface causes lots of troubles in search process. With analysis into AltaVista search engine's query logs, Silverstein Marais, Henzinger, and Moricz (1999) found that the average length of user queries is 2.35 terms. Jansen, Spink, Bateman, and Saracevic (1998) also concluded by analyzing query logs that most user queries are short (around two terms per query). A short list of keywords is not always a good descriptor of the information needs of search users because it may have ambiguities either in content or in information need. In order to help users to reorganize their short, ill-formed, and possibly ambiguous queries, search engines develop query recommendation function.

Query recommendation (as shown in Fig. 1) has been widely adopted by search users as an important way in finding information effectively. According to CNNIC search behavior survey report (CNNIC, 2009), 78.2% users will change their queries (mostly by adopting search engine's recommendation function) if they cannot obtain satisfactory results with the current query. When we look into search behavior logs collected by a famous commercial search engine in China (details to be described in Section 3), we found that 15.36% query sessions contain clicks on query recommendation links. Users adopt query recommendation function to clarify their information needs without taking efforts in inputting new queries. Therefore, it is important for search engines to provide high-quality recommendations which can represent users' exact information needs.

Currently, most commercial search engines and lots of research work (Baeza-Yates, Hurtado, & Mendoza, 2004; Baeza-Yates & Tiberi, 2007; Cucerzan & White, 2007; Fonseca, Golgher, De Moura, & Ziviani, 2003; Liu, 2008; Wen, Nie, & Zhang, 2001; Zaiane & Strilets, 2002) focus on how to recommend queries based on users' previous query and click behaviors. The idea is to locate popular queries which are similar with the current query either in content (Baeza-Yates et al., 2004; Baeza-Yates & Tiberi, 2007; Fonseca et al., 2003; Wen et al., 2001; Zaiane & Strilets, 2002) or in click context[1] (Cucerzan & White, 2007; Fonseca et al., 2003; Liu & Sun, 2008; Wen et al., 2001; Zaiane & Strilets, 2002). This method suggests user to adopt a similar and frequently-adopted queries to finish his search task. The major problem with this kind of recommendation methods is that it lacks understanding of users' actual information needs. It does not take current users' search intent into consideration;

* Corresponding author. Tel.: +86 10 62796672.
E-mail address: yiqunliu@tsinghua.edu.cn (Y. Liu).

[1] Here click context refers to the documents ever clicked by users for this query. In these researches if two queries share similar click context, it is supposed that they are similar and relevant.

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.04.188

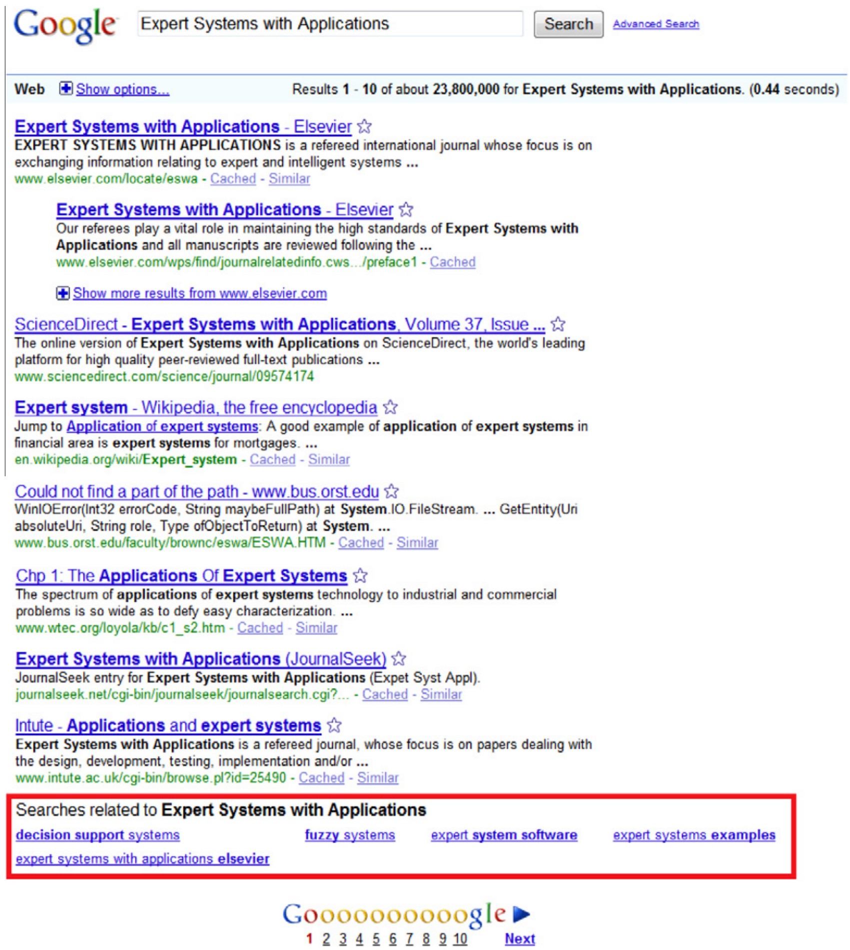

Fig. 1. Query recommendation function provided by commercial search engines (take query "Expert Systems with Applications" in Google.com for example, red box show query recommendations called "related searches"). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

instead, it supposes that he shares similar interests with other users who propose similar queries. This assumption is correct for most hot queries. However, it sometimes fails to give proper recommendations, especially for low-frequency queries because there are not so many candidate queries for them. Table 1 shows recommendation results for the query "WWW 2010" in three most widely-adopted search engines in China. We can see that most of these recommendation results focus on other events which also happen in the year 2010 (such as Shanghai Expo 2010 and Civil service examination in 2010). None of them is related to the World Wide Web conference because the query is a low-frequency one and few other people propose similar queries with the same meaning.

In order to solve these problems and generate better query recommendation results, we have to find out how users describe their information needs so that these information needs can be utilized to organize queries with more exact meanings. When we look into users' interaction process with search engines, we find that users' click through behavior probably contains traces of their information needs. When user clicks a certain search result, it does not necessary mean that he is interested with the result because he/she has not viewed the result document yet. However, we can assume that he/she is interested in the snippets of the corresponding results because these snippets are actually shown to and read by users. Therefore, our assumption is that users' information needs are described in their interaction with search engines, more specifically, in the snippets of the results which they ever clicked.

Table 1
Top five query recommendation results for the query "WWW2010" (a Web related conference) in three most frequently used search engines in China (Baidu, Google and Sogou).

#	Baidu	Google China	Sogou
1	pes2010 (*a popular computer game*)	2010 国家公务员职位表 (*National civil service positions for* 2010)	2010 年国家公务员 (*National civil service exam in* 2010)
2	qq2010 (*a software*)	2010 年国家公务员报名 (*National civil service exam registration in* 2010)	2010 发型 (*fashion hair styles in* 2010)
3	实况 2010 (*a popular computer game*)	2010 国家公务员报名 (*National civil service exam registration in* 2010)	2010 年考研报名 (*Graduate entrance exam in* 2010)
4	实况足球 2010 (*a popular computer game*)		2010 公务员报名 (*civil service exam registration in* 2010)
5	卡巴斯基 2010 (*Kaparsky* 2010, *a software*)		2010 公务员考试 (*civil service exam in* 2010)

Please cite this article in press as: Liu, Y., et al. How do users describe their information need: Query recommendation based on snippet click model. *J. Cities* (2011), doi:10.1016/j.eswa.2011.04.188

With this assumption, we propose a query recommendation framework based on snippet click model. We analyze the nature of query recommendation process and construct two snippet click models. These models make use of clicked snippets' content features to extract keywords which can describe user's information need.

Compared with state-of-the-art techniques which require content or click context comparison with previous queries, this framework only involves content and click-through analysis of the snippets that are clicked by users. This kind of information is usually recorded in search engine click-through logs and the framework is thus easy and efficient for practical applications.

In summary, the contributions of the paper are:

(1) We propose a query recommendation framework in which keywords are recommended because of their appearance in clicked snippets instead of similarity with previous queries.
(2) We analyze the nature of query recommendation process from user's perspective. Two snippet click models and corresponding recommendation algorithms are presented based this analysis.
(3) Differently from previous human annotation based evaluation framework, we evaluate query recommendation performance based on practical search engine's click-through logs. Both click-through rate and user click amount are used to prove effectiveness of the proposed algorithms.

The rest of the paper is organized as follows: Section 2 gives a brief review of related work in query recommendation and click-through models. Section 3 analyzes the nature of query recommendation process and proposes two snippet click-through models. Corresponding algorithms are also presented in this Section. Experimental results are presented in Section 4 by performance evaluation on large scale search engines' click-through logs. In the end, there is the conclusion and a discussion of future work.

2. Related work

2.1. Query recommendation techniques

As the only interface for users to access Web pages, queries are one of the most important factors that affect the performance of search engines. Although users' information needs are complicated, their queries are usually simple, short and possibly ambiguous. Queries are simple because users are unwilling or unable to organize complicated queries which can describe their information needs more clearly. This phenomenon causes a major challenge in current Web search techniques, which is the understanding of user's information need behind queries (Broder, 2002). It is quite difficult for search engines to understand information need because only queries and click through behavior data can be utilized. Therefore, query recommendation technique is proposed to present users with a list of possible query choices whose information needs are relatively more clear to search engines. By this means, users can clarify their information need by clicking recommendation query links instead of inputting new queries.

Most state-of-the-art query recommendation techniques focus on recommending queries which were previously proposed by users. These queries should be both popular and similar with the current query. It should be popular so that the current user probably likes it. It should also be similar with the current query because similar queries are likely to represent similar information needs.

From the point of view of how to define "similarity" between queries, most previous query recommendation techniques can be grouped into two categories.

The first category of techniques adopts content-based similarity measures to find similar queries. It comes from the traditional method of query expansion (*QE* for short) (Baeza-Yates & Ribeiro-Neto, 1999, chap. 3) in information retrieval researches. The major difference lies that *QE* usually utilize dictionary or pseudo relevance feedback to get expansion words; while query recommendation refers to query log data. Wen et al. (2001) adopt string matching features to locate similar queries. Zaiane and Strilets (2002) also utilized content similarity to recommend similar queries. The work of Fonseca et al. (2003) presented a method to discover related queries by looking into the queries proposed by previous users in a similar query sessions. While Baeza-Yates (Baeza-Yates et al., 2004; Baeza-Yates & Tiberi, 2007) uses the content of user's historical preferences which is recorded in query logs to describe the semantic meaning of the current query.

For the second category of techniques, similarity is usually calculated by analyzing the click context of different queries. Most commercial search engines collect users' click through logs which record both query and click behavior of search users. Therefore, it is possible to represent a query as the set of URLs that are clicked by users. Wen et al. (2001), Zaiane and Strilets (2002) and Cucerzan and White (2007) all adopted this method to measure the similarity of queries besides the content similarity method. Besides this representation method, researches such as (Liu & Sun, 2008) also adopt bipartite network to describe the relation between queries and the clicked URLs. According to this work, with clustering technique in the bipartite network it is possible to extract possible "hidden" query pairs which are similar to each other.

Many previous researches adopt either or both of these two methods to measure the similarity between queries. Here the central problem is how to model the information need associated to a query. In content-based methods, information need is supposed to be contained in query content; while in click-context-based methods, information need is represented as the set of URLs clicked by users. We can see that information need cannot be clearly described by the query content because it is short, simple and possibly ambiguous. It is not clearly described by the clicked URLs either, because URLs are not actually representing information needs explicitly.

Differently from these methods, we look into users' search engine interaction process to find out how users describe their information needs. We find that information need is closely related with the snippets of the results clicked by users. By constructing a snippet click model, we extract user's information need from the content of the clicked snippets and generate recommendations directly from these contents. By this means we hope to generate query recommendations which are better at representing users' information need and preferred by search users.

2.2. Search engine user behavior models

In recent years, search engine user behavior analysis has been receiving much attention in Web search area. Several approaches are proposed to mining relevant information from click-through data and some applications are implemented based on the wisdom of crowds, e.g. re-ranking search results (Agichtein, Brill, & Dumais, 2006), learning ranking strategies (Dou, Song, Yuan, & Wen, 2008; Radlins & Joachim, 2005) and evaluating engine performances (Cen, Liu, Zhang, Ru, & Ma, 2009; Liu, Fu, Zhang, Ma, & Ru, 2007).

Joachims, Granka, Pan, Hembrooke, and Gay (2005) and Joachims et al. (2007) analyzed users' decision processes in Web search using eye-tracking and compared implicit feedback against manual relevance judgments. They proved that the documents clicked by users contain kind of implicit feedback information. Although this kind of feedback information cannot be directly adopted as absolute relevant judgment, it is possible to utilize this information to improve search performances.

Agichtein, Brill, Dumais, and Ragno (2006) proposed an idea of aggregating information from many unreliable user search session traces instead of treating each user as an individual "expert". They pointed out that user behaviors were only probabilistically related to explicit relevance judgments and preferences. Dou et al. (2008) studied the problem of using aggregate click-through log, and found that although some individual user clicks were unreliable, the aggregation of a large number of user clicks provided a valuable indicator of relevance preference. Agrawal, Halverson, Kenthapadi, Mishra, and Tsaparas (2009) proposed a method of transforming clicks into weighted, directed graphs and devised a function for finding cuts in these graphs that induce a labeling. Recently, Craswell, Zoeter, Taylor, and Ramsey (2008) and Guo, Liu, and Wang (2009) drew a cascade model, in which users view results from top to bottom and leave as soon as they see a worthwhile document, for explaining position bias of user behavior. All of these researches used statistics features to mine relevance information with extensive user interaction data for one query.

Inspired by these findings, some query recommendation works also tried to suggest related queries by extracting information from clicked documents because these documents are expected to contain user's preference and relevance judgments. Cucerzan and White (2007) propose a method to suggest queries based on mining into post-query browsing behaviors. They referred to these post-query behaviors as "search trails" and regarded queries with same landing pages (the ending pages of search trails) as similar queries. Bilenko and White (2008) further adopt this method to improve the performance of relevance ranking. However, they still use landing pages as a representation of queries and adopt queries with same or similar landing pages as query recommendations. Differently from this method, we look into the snippet of clicked documents and extract recommendation candidates from these snippets directly.

3. Snippet click model

As stated in Section 1, our query recommendation framework is based on the assumption that users' information needs are described in their interaction with search engines, more specifically, in snippets of the results which they ever clicked. This assumption comes from the phenomenon that when user clicks a certain search result, it does not necessarily mean that he is interested with the result because he/she has not viewed the result document yet. It is probably that he/she is interested in the snippets of the corresponding results because these snippets are actually shown to and read by users. According to this assumption, we construct the snippet click based query recommendation framework. The proposed framework and its differences with the previous methods are compared in Fig. 2.

We can see that this framework is based on a snippet click model which tries to extract keywords appearing in users' clicked snippets as recommendations. In this section we will explain this framework in detail and answer the following questions: What is the nature of query recommendation?, How do users describe their information needs while interacting with search engines?, Why clicked snippets contain information that is necessary for query recommendation and How can we extract this kind of information (in the form of keywords) from clicked snippets.

3.1. Query recommendation process

According to Baeza-Yates et al. (2004), query recommendation is the method which is adopted to suggest alternative queries to users in order to help them to specify alternative related queries in their search process. In our opinion, the users not only specify alternative related queries but also try to express their information need in the form of query recommendations. Therefore, search engine should recommend queries which are most likely to represent users' information needs.

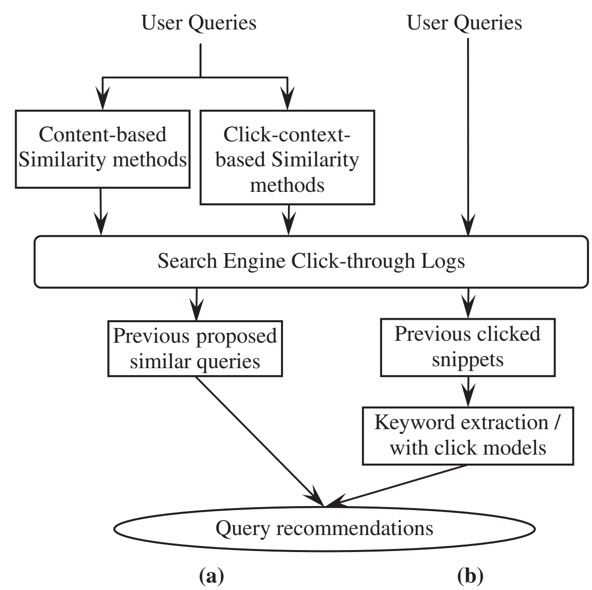

Fig. 2. Framework of the proposed query recommendation method (b) and its comparison with previous methods mentioned in Section 2 (a).

For query recommendation task, we try to give a ranking of queries which are related to the original proposed query and are better at representing users' information needs than that query. Let *Need* denote user's actual information need, and let *Query* denote the original query. Then the process in which users express their information need through queries can be shown as Fig. 3.

We can see from Fig. 3 that *Need* is implicit for search engines, while *Query* is explicitly proposed to search engines and result list is generated according to *Query*. This phenomenon is decided by current search engine's interaction method in which information need is represented by query proposed to search engines. According to language model for information retrieval and Web search,

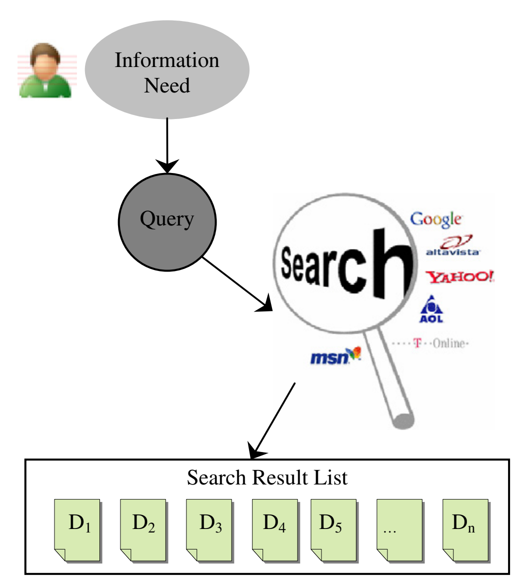

Fig. 3. User's search process in which queries are used to express information needs.

Please cite this article in press as: Liu, Y., et al. How do users describe their information need: Query recommendation based on snippet click model. *J. Cities* (2011), doi:10.1016/j.eswa.2011.04.188

documents in search result list are ranked by $P(D_i|Query)$, where D_i is a certain document in the result list. With Bayesian formula, $P(D_i|Query)$ can be written as:

$$P(D_i|Query) \approx P(Query|D_i)P(D_i) \tag{1}$$

Because $P(D_i)$ is usually supposed to be equal for all D_i, documents in search result list are actually ranked by $P(Query|D_i)$.

In ideal circumstance, *Query* can represent *Need* and the result list can be regarded as a ranking of

$$P(Query|D_i) \approx P(Need|D_i) \tag{2}$$

However, in practical Web search environment *Query* is driven by *Need* but *Query* always cannot cover the full and exact meaning of *Need*. Therefore, query recommendation is needed to help users to reorganize their queries and better describe *Need*.

Given a list of query recommendations $\{QS_1, QS_2, QS_3, \ldots, QS_n\}$, whether these suggestions are selected is decided by their accordance with *Need*. $P(QS_i|Need)$ can be used to describe whether QS_i would be selected by users because it represents the possibility of generating QS_i with user's actual information need *Need*. As a result, the task of query recommendation can be regarded as a process of finding QS^* satisfying $QS^* = \arg\max_i P(QS_i|Need)$.

From Fig. 3 we can see that *Need* cannot be directly obtained from users, therefore, most previous researches in query recommendation try to find QS^* satisfying $QS^* = \arg\max_i P(QS_i|Query)$ instead of $QS^* = \arg\max_i P(QS_i|Need)$. That is why these previous methods focus on recommending queries which are similar with the proposed query either in content or click context. However, because query recommendations are supposed to help users to better describe their *Need*, it is believed that recommendations will be clicked by users only if the original query does not describe the exact meaning of *Need* very well. Here comes the problem, if the proposed query cannot describe the exact meaning of *Need*, how can we suppose the recommendations that are similar with this query are better at expressing users' *Need*?

In order to solve this problem, we have to find a better way to describe users' information needs. Therefore, users' interaction process with search engines should be looked into to extract possible traces of the information need.

3.2. User's interaction with search engines

According to previous researches introduced in Section 2.2, user clicks can be regarded as a kind of implicit feedback information for the relevance of the clicked documents. However, this kind of feedback information is often referred to as "not reliable" because it is believed that user's behavior always contains noises and biases.

When we look into user's interaction process with search engines shown in Fig. 4, it is possible to find out why user clicks can only be regarded as unreliable information sources.

From Fig. 4, we can see that users have not navigated to the result document when he/she decides to click the result. This decision is not made by reading the content of the page (users have not read it yet); instead, it is made by reading the snippet of the page shown in the result list (users actually read it). The snippet does not necessarily represent the full and exact meaning of the original document. Therefore, the documents clicked by users are not always relevant to the proposed queries. It explains why user click is an "unreliable" feedback information source because it can only be regarded as user's preference for the snippet instead of the whole document.

Although this information source is "unreliable" for relevance feedback, it can be useful for the query recommendation task. We can see from the interaction process that although the documents clicked do not necessarily meet users' information needs,

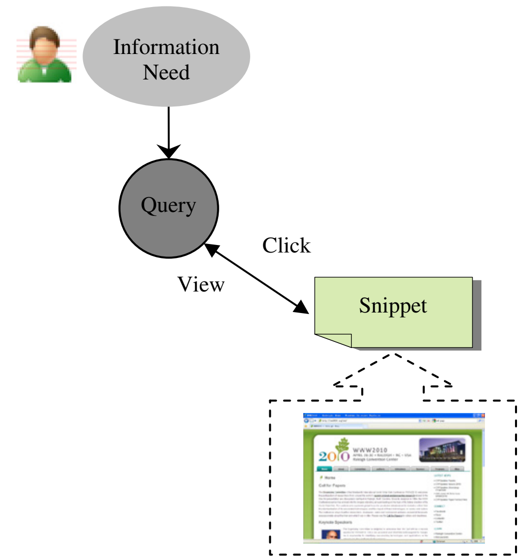

Fig. 4. User's interaction process with search engines.

the corresponding snippets are likely to contain contents that are useful for users. Users are interested in the content of snippet because it contains keywords that are related to their information needs. If we can locate these keywords, they can be adopted as recommendations because they can be used to describe users' actual information needs.

Therefore, the major idea of our query recommendation framework is to locate keywords that appear in snippets clicked by users and can describe users' information need. Differently from previous recommendation methods, it relies on information extracted from users' result click-through process instead of the historical queries proposed by other users.

3.3. Snippet click models

We believe that users click a certain document because he/she actually views its corresponding snippet and also expects this document to meet his/her information need. Therefore, the probability of clicking a certain document is decided by both whether user views the snippet and whether user is interested in it. That is:

$$P(click_i) = P(click_i, view_i) = P(view_i)P(click_i|view_i) \tag{3}$$

Because user can only view the snippet of the document before he/she actually click on the result, we can see that the probability of clicking is decided by whether user is interested in the snippet of the result document; in other words, by whether this snippet meet user's information need. So we get the following equation:

$$P(click_i|view_i) = P(snippet_i|Need) \tag{4}$$

Substituting expressions (4) into (3), we obtain:

$$P(click_i) = P(view_i)P(click_i|view_i) = P(view_i)P(snippet_i|Need)$$

Then we can get:

$$P(snippet_i|Need) = \frac{P(click_i)}{P(view_i)} \tag{5}$$

From Eq. (5), we can see that users' information need is described in the snippets that they click. We can derive different models from this equation which are corresponding to different kinds of recommendation methods. Here we propose both a global scale and a local scale snippet click model. They can both be adopted to finish the task of query recommendation.

3.3.1. A global scale snippet click model

In the global scale model, we treat all the clicked snippets for a certain query as a whole "snippet document". Therefore, for all clicked snippets, we get the following equation:

$$\sum_i P(snippet_i|Need) = \sum_i \frac{P(click_i)}{P(view_i)} \tag{6}$$

In Eq. (6), $P(click_i)$ can be estimated with a maximum likelihood method. It is difficult to estimate exact values of $P(view_i)$ directly, but we can adopt search engine user behavior models such as the one proposed in Agichtein et al. (2006) to give a rough estimation or we can just assume that $P(view)$ follows a uniform distribution. Therefore, for a given query Q and a given set of query logs, the right side of Eq. (6) has a fixed value C. That is:

$$\sum_i P(snippet_i|Need) = C \tag{7}$$

Specifically, if we suppose that $P(view_i)$ follow a uniform distribution for all i, then we get:

$$\sum_i P(snippet_i|Need) = \sum_i \frac{P(click_i)}{P(view_i)} \propto \sum_i P(click_i) = 1 \tag{8}$$

From Eq. (7) we can see that *Need* is related to all clicked snippet in the result list. Therefore, a possible method for estimating *Need* is to find the reprehensive component in the clicked snippets. That is, to extract a keyword list from the snippets. Different methods can be adopted to finish the keyword extraction task but discussing performance of these methods is not a key point of this paper. Therefore, we utilized a simple TF-based model to extract keyword lists from the snippets. For each keyword in the snippets, the recommendation candidates are those with the largest TFs where TF is defined as:

$$TF(w) = \sum_i (appearances\ of\ \ w\ in\ snippet_i) \tag{9}$$

We do not involve a DF factor in this keyword extraction process because that requires construction of a large scale background corpus which is difficult to obtain. In Section 4 we show that this simple keyword extraction method can generate promising results for query recommendation tasks.

With Eqs. (8) and (9), we propose the corresponding recommendation algorithm (Algorithm 1) as follows:

Algorithm 1. Query recommendation based on global scale snippet click model

Recommendation (Original query Q, Click through log *LOG*)
1. Find all documents clicked for Q in *LOG* and form a document set called D;
2. Extract all snippets of D for query Q using search engine interfaces and form a snippet set called S;
3. For snippet set S, extract N keywords according to Eq. (9) or other keyword extraction algorithms;
4. Return these N keywords as recommendation words.

By this algorithm, a list of words can be generated as recommendation keywords for query Q. We should point out that sometimes these keywords may not be directly adopted as recommendations because they should be combined with the original query to form complete information need. For example, keyword "download" may be returned for query "Live messenger", it should be combined with the original query to form a complete query recommendation word "Live messenger download". However, these keywords (even when they cannot be directly adopted as recommendations) are supposed to meet users' information needs.

3.3.2. A local scale snippet click model

Differently with the global scale model, in a local scale snippet click model each snippet is considered separately. With the bag-of-words model, a certain clicked snippet can be represented by a set of keywords (each with different TF values). Therefore, the left side of Eq. (5) can be written as:

$$P(snippet_i|Need) = P(\{w_{i1}, w_{i2}, \ldots, w_{ik}, \ldots\}|Need) \tag{10}$$

If we suppose that each word's appearance is independent from each other given users' information need, then we can get:

$$\begin{aligned} P(snippet_i|Need) &= P(\{w_{i1}, w_{i2}, \ldots, w_{ik}, \ldots\}|Need) \\ &= \prod_k (P(w_{ik}|Need))^{TF_{ik}} \end{aligned} \tag{11}$$

Here TF_{ik} is the term frequency of word w_{ik} in $snippet_i$. If we use logarithmic function on both sides of Eq. (11), we get:

$$\log(P(snippet_i|Need)) = \sum_k TF_{ik} \cdot \log(P(w_{ik}|Need)) \tag{12}$$

If we substitute (5) into (12), we obtain the following equation:

$$\sum_k TF_{ik} \cdot \log(P(w_{ik}|Need)) = \log\left(\frac{P(click_i)}{P(view_i)}\right) \tag{13}$$

In Eq. (13), the right side is a fixed value according to Section 3.3.1; while the left side contains unknown variables $P(w_{ik}|Need)$. These variables can be regarded as the probabilities of words w_{ik} describing user's information need. If we can get values of these variables, we simply adopt words with the largest $P(w_{ik}|Need)$s as recommendation candidates.

Eq. (13) is a set of linear equations, so lots of methods such as Gaussian elimination method[2] can be adopted to solve these equations. The matrix format of Eq. (13) is:

$$\begin{bmatrix} TF_{1,1} & TF_{1,2} & \cdots & TF_{1,m} \\ TF_{2,1} & TF_{2,2} & \ldots & TF_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ TF_{n,1} & TF_{n,2} & \ldots & TF_{n,m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_m \end{bmatrix} \tag{14}$$

In Eq. (14), m is the number of keywords in the clicked snippets while n is the number of clicked snippets. x_i is used to represent $P(w_{ik}|Need)$ and C_i represents the value of $\log(P(click_i)/P(view_i))$. We can see that m is usually larger than n because it can be supposed that each snippet contains at least one unique keyword.

These equations can be solved only if m equals to n, so we have to discard several keywords to reduce the number of keywords m. For practical use, we can discard the keywords which only appears in documents with lowest $P(Click)$ values because these keywords are unlikely to be interesting for users (if not, the $P(Click)$ value will not be low). After discarding a number of keywords in Eq. (14), we are able to estimate the value of $P(w_{ik}|Need)$ for the remaining keywords and those with the largest $P(w_{ik}|Need)$ values are selected as recommendations.

Another problem while solving the equations in (14) is data sparsity. Not all keywords appear in each clicked snippet, therefore many TF_{ij} in Eq. (14) are with zero values. In order to avoid this data sparsity problem and estimate $P(w_{ik}|Need)$ correctly, we adopt smoothing technique while solving (14). Katz smoothing (Chen & Goodman, 1998) is applied for this smoothing task because it is one of the most widely-used and effective techniques. With this kind of smoothing, TFs are recalculated according to (15) to replace TFs in Eq. (14)

$$TF'_{i,j} = \begin{cases} \alpha \cdot \sum_k TF_{i,k} & TF_{i,j} = 0 \\ (1-\alpha) \cdot TF_{i,k} & TF_{i,j} > 0 \end{cases} \tag{15}$$

[2] http://en.wikipedia.org/wiki/Gaussian_elimination.

Please cite this article in press as: Liu, Y., et al. How do users describe their information need: Query recommendation based on snippet click model. *J. Cities* (2011), doi:10.1016/j.eswa.2011.04.188

Here α is a smoothing factor and assigned the value of 0.05 in our experimental researches.

After solving equations in (14), we obtain each keyword's probability of describing user's information need. Then we can choose keywords with the largest probabilities as query recommendations and suggest them to users. We can give the corresponding algorithm as follows:

Algorithm 2. Query recommendation based on local scale snippet click model

Recommendation (Original query *Q*, Click through log *LOG*)
1. Find all documents clicked for *Q* in *LOG* and form a document set called *D*;
2. Extract all snippets of *D* for query *Q* using search engine interfaces and form a snippet set called *S*;
3. Recommendation candidate set *CANDIDATE* = { };
4. For each snippet S_i in *S*,
   if $P(Click_i)$ > threshold *T*
     put all words into *CANDIDATE*;
5. For all words in *CANDIDATE*, form equations *E* according to (14) and (15)
6. Solve *E* according to Gaussian elimination or other methods
7. Select *N* keywords with the largest $P(w_{ik}|Need)$ values;
8. Return these *N* keywords as recommendation words.

Similar with Algorithm 1, these *N* keywords should be combined with the original query to form complete query recommendations. There is a parameter *T* in Algorithm 2 which is the threshold of *P*(*Click*) value. We do not put keywords into *CANDIDATE* if they only appear in snippets with *P*(*Click*) values lower than *T*. By this means, we construct equations according to the local scale model and make sure that they can be solved.

3.3.3. Assigning weights for different snippet compositions

Besides the global and local scale snippet click models, the structure inside a document's snippet can also be utilized to improve the performance of query recommendations. We know that for each snippet of a given document, it is composed of both a "title" part and an "abstract" part as shown in Fig. 5.

From Fig. 5 we can see that snippets are composed of two parts and these parts play different parts in users' interaction process with search engines. Usually the title part is shown in larger font and different colors (as a hyperlink); and it is likely that users pay more attention to this part of snippets. Therefore, it is reasonable to assign different weights to words that appear in different compositions of a snippet.

We can introduce a "title weight" parameter λ to both Eqs. (9) and (14); then the TF factor in these two equations can be rewritten as follows:

$$TF'' = \lambda \cdot TF(title) + (1 - \lambda) \cdot TF(abstract) \tag{16}$$

We can see that λ should be assigned a value between 0 and 1. A larger λ value means assigning more weight to the title part, and vise versa. When we substitute (16) into (9) and (14), we can assign different weights for word's appearance in title and abstract parts of a snippet. Experimental results with different λ values will be shown in Section 4.

4. Experiments and discussions

4.1. Experiment setups

In order to evaluate performance of the proposed query recommendation framework, we use practical search click-through data and compares performance of our method with current search engine's query recommendation performances. This evaluation method is different from most previous researches which adopt human-annotation based precision-recall metrics (Cucerzan & White, 2007; Liu & Sun, 2008). We believe that human annotation may not represent user's information need very well because it is difficult to tell what users actually want only by examining queries. The process also costs lots of time and human efforts and makes it almost impossible to construct a large-scale training/test set. Therefore, we choose a different evaluation process in which the performance of query recommendation is evaluated by how many percentages of users actually clicked these recommendations in practical environment.

With the help of a widely-used Chinese search engine, we collect click-through log data of two of the most popular search engines in China (www.baidu.com and www.sogou.com). These click-through logs were collected using Web browser toolbars which is often used to record anonymous click-through information from users' browsing behavior. Previous work such as (Bilenko & White, 2008) adopts this kind of click-through information to improve ranking performance. Another work by Liu, Cen, Zhang, Ma, and Ru (2008) proposed a Web spam identification algorithm based on this kind of user behavior data. Here we performed experiments on two different search engines' log data because we want to prove that the effectiveness of the proposed method is independent of search engine choices.

We collected this kind of click-through data from September 1st to 21st, 2009 and then only retained the queries that were requested at least 20 times (about once a day on average). By this means, we want to reduce possible noises from the log data and avoid loss of generality as well. Queries without query recommendation clicks were also discarded because we suppose that these queries describe information need clearly and no recommendations are needed. After that, we got 32,323 queries whose recommendations were clicked 691,806 times during this time period. Among these queries, 9000 were randomly sampled to form a experiment query set. Therefore, query set contains all kinds of queries with different frequencies and different amount of recommendation clicks. To the best of our knowledge, it is the largest and most practical experiment query set in query recommendation researches.

Both click-through rate and click amount are adopted as metrics to evaluate the performance of recommendation algorithms. In our work, click-through rate (CTR) is defined as the percentage of ever-clicked recommendations in all recommendations for a given query. Click amount is referred to as the number of clicks for a certain recommendation. CTR is used to evaluate whether a recommendation is clicked by users while click amount represents how many times a recommendation is clicked by users. If we want to prove effectiveness of the proposed algorithms, recommendations generated by these algorithms should have higher CTR values and larger click amounts than the other recommendations.

4.2. Effectiveness of the global scale snippet click model

For the global scale model, all clicked snippets for a certain query are treated as a whole. According to Eq. (8), user's information need is related with the snippets. Therefore, traditional keyword extraction algorithm can be adopted to generate recommendations for queries.

We employ the TF-based method (as shown in Eq. (9)) to finish the keyword extraction task. After the recommendation keywords were generated according to Algorithm 1 (with *N* = 10), we look into the click-through logs introduced in Section 4.1 and see whether these recommendations appear in search engines' recommendation lists. For all recommendations generated by search engines, some match our recommendation keywords while others

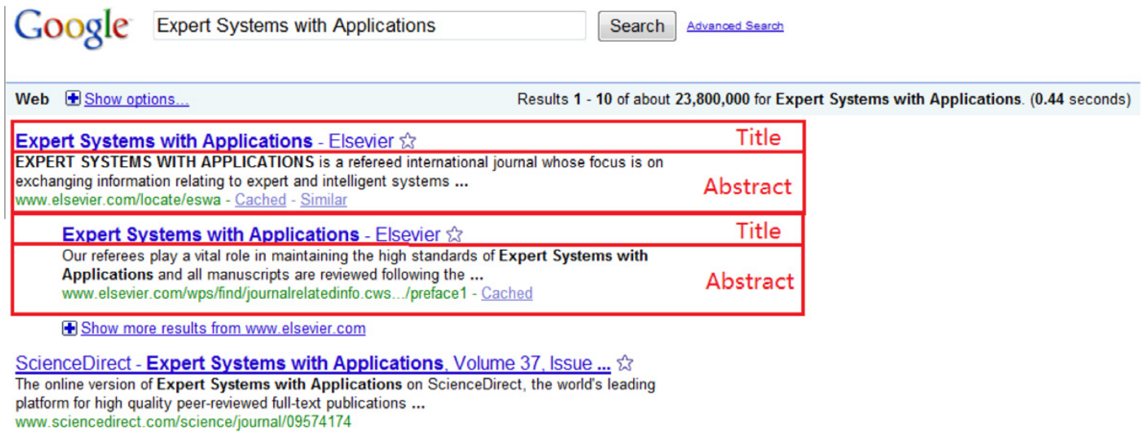

Fig. 5. Composition of a result document's snippet (take the "Expert Systems with Applications" query in Google for example).

not, so we can compare click-through rate of these two kinds of recommendations to see how our algorithm performs. The comparison result is shown in Fig. 6.

In our experiments, we found that 33.86% query recommendations given by Baidu.com and 27.34% recommendations given by Sogou.com match the keywords generated by Algorithm 1. These recommendations have a much higher click-through rate and much larger average amount of user clicks according to Fig. 6. It means that keywords extracted from the clicked snippets are preferred by users while selecting query recommendations. Therefore, users click a larger percentage of these recommendations and click each of them more frequently.

The experimental results show that the keywords generated by our query recommendation algorithm are more preferred by users than the others. About one third of the recommendations provided by Baidu and Sogou search engines match our algorithm results; while these matching results' CTR rates and user click amounts are much higher than the mismatching results.

From the results in Table 2 we can see that recommendations generated by Algorithm 1 are clicked more frequently than the other ones. A further experiment in Fig. 7 shows how the percentage of matching recommendations varies with the increase of user click frequencies.

We can see from Figs. 7 and 8 that for both search engines, there are more matching ones in the recommendations that are frequently clicked by users. Similar with the results shown in Fig. 6, it means that matching recommendations tend to be clicked more frequently than the mismatching ones. Another finding in Figs. 7 and 8 is that the percentage of matching recommendations in Baidu is larger than the percentage of Sogou. It accords with the stats shown above that 33.86% recommendations in Baidu match the results generated by Algorithm 1 while the percentage is only 27.34% in Sogou search engines. This difference comes from these two search engines' different recommendation generation methods.

In Section 3.3.3, we introduce Eq. (16) to assign different weights to title and abstract part of a snippet. A "title weight"

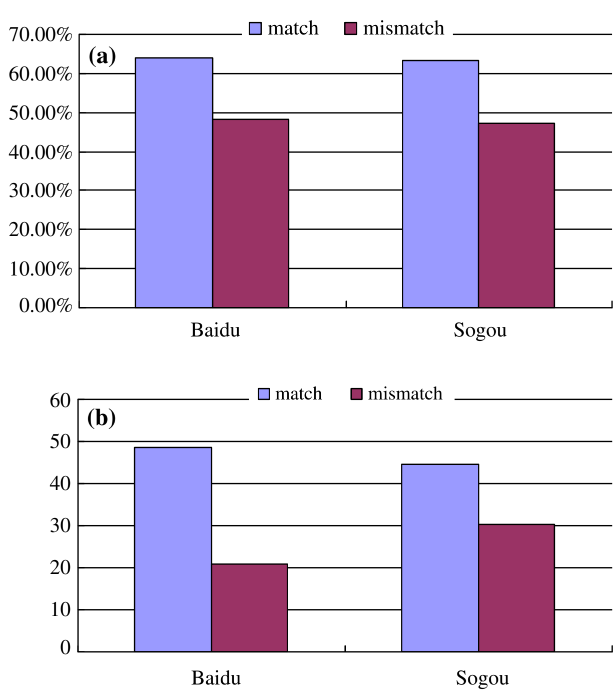

Fig. 6. Comparison of click-through rate (a) and average amount of user clicks (b) between the recommendations that matches/does not match recommendation results generated by Algorithm 1.

Table 2
Performance improvement of recommendation results generated by Algorithm 1 compared with the other results provided by Baidu and Sogou.

	CTR improvement (%)	Click amount improvement (%)
Baidu	+32.80	+131.03
Sogou	+39.55	+47.27

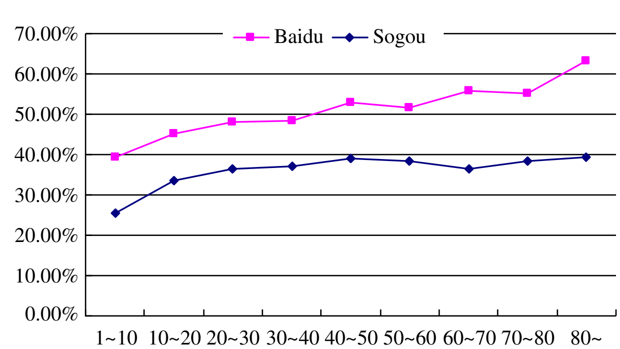

Fig. 7. Percentage of recommendations that matches results generated by Algorithm 1 with different click frequencies (value axis: percentage of matching recommendations; category axis: recommendations with different user click frequencies).

Please cite this article in press as: Liu, Y., et al. How do users describe their information need: Query recommendation based on snippet click model. *J. Cities* (2011), doi:10.1016/j.eswa.2011.04.188

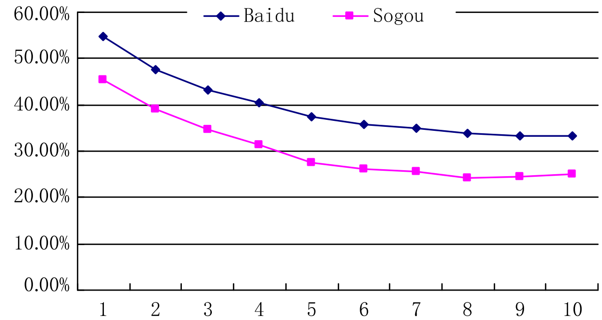

Fig. 8. Percentage of recommendations that matches results generated by Algorithm 1 with different click frequency rankings (value axis: percentage of matching recommendations; category axis: recommendations' ranking according to click frequency, each query's top ten clicked recommendations are considered in this experiment).

parameter λ is used to adjust the weight assigned to different parts. Experimental results in Fig. 9 show how the recommendation result varies with the parameter λ.

According to the results shown in Fig. 9, recommendation performances do not change much with different value of λ. With the increase of λ from 0 to 0.9, Baidu's CTR of matching keywords improves from 63.04% up to 64.54% while CTR of mismatching ones decreases from 52.41% down to 49.85%. The results of Sogou are similar with Baidu's according to Fig. 9. It is likely that the title part is more important for query recommendation task because the performance improves (higher CTR for matching result, lower CTR for mismatching ones) with increase in title weight. However, when λ is set to 1, which means no abstract information is involved, the performance is not as good as the performance with $\lambda = 0.9$ (higher CTR for mismatching results). Therefore, abstract part in snippets is also useful for recommendation task, although it is not as important as the title part.

4.3. Effectiveness of the local scale snippet click model

For the local scale model, each snippet is considered separately. With Algorithm 2, we are able to estimate the probability of a keyword in representing user's information need. With a similar estimation method as in Section 4.2, we compare the performance of matching and mismatching recommendation results with metrics of CTR and average click amount. The experimental results are shown in Fig. 10.

Similar with results shown in Fig. 6, from Fig. 10 we see that Algorithm 2 also gain better performance than the original recommendation results provided by search engines. Query recommendations generated by Algorithm 2 are more preferred by users

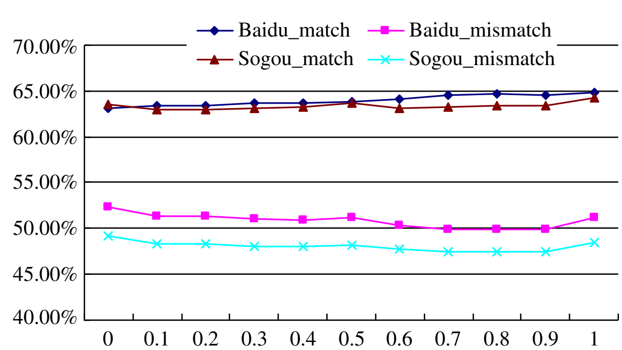

Fig. 9. Comparison of recommendation click through rate with different title weight parameter λ (value axis: click-through rate; category axis: λ).

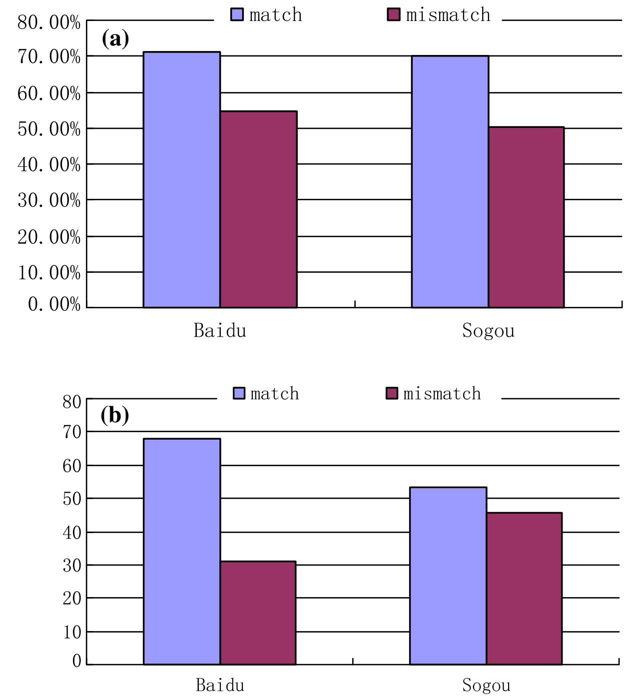

Fig. 10. Comparison of click-through rate (a) and average amount of user clicks (b) between the recommendations that matches/does not match recommendation results generated by Algorithm 2.

compared with the other recommendations either with CTR or the amount of user clicks.

From results shown in Figs. 6 and 10 we find that both Algorithm 1 and Algorithm 2 are effective in generating user-preferred recommendations. In order to compare performance of these two different kinds of snippet click models, in Table 3 we show the performance improvement of recommendation results provided by Algorithm 2 compared with the other results provided by Baidu and Sogou.

Compared with the stats in Table 2, we see that these two algorithms have different recommendation performances. For Baidu's results, Algorithm 2's CTR and click amount improvements are not as much as Algorithm 1's. For Sogou, Algorithm 2's CTR improvement is larger while click amount improvement is smaller compared with Algorithm 1. However, both algorithms prove to be more effective than the results provided by Baidu and Sogou search engines.

From the comparisons we found that although the computation cost of Algorithm 2 is more than that of Algorithm 1 (Algorithm 2 involves equation solving), the performance of Algorithm 2 does not show significant improvement compared with Algorithm 1. We expect Algorithm 2 to provide more reasonable results because it utilizes each snippet's click-through information. However, when we look into differences between recommendations generated by these two algorithms, we found that the ranking of the top-ranked results are not the same, but the top-ranked result sets share a lot in common. That is why these two algorithms'

Table 3
Performance improvement of recommendation results generated by Algorithm 2 compared with the other results provided by Baidu and Sogou.

	CTR improvement (%)	Click amount improvement (%)
Baidu	+29.67	+120.45
Sogou	+39.55	+16.38

Please cite this article in press as: Liu, Y., et al. How do users describe their information need: Query recommendation based on snippet click model. *J. Cities* (2011), doi:10.1016/j.eswa.2011.04.188

performance differences are not so much because both algorithms recommend very similar top-ranked result sets.

5. Conclusions and future work

Most query recommendation approaches focus on locating previously proposed queries which are similar to the current query either in content or in click context. Different from these previous methods, we propose a query recommendation framework which tries to extract user's information need from click-through logs. Based on analysis into user's interaction process with search engines, we found that user's information needs are likely to be represented in the clicked snippets. Two snippet click models are constructed according to this finding and corresponding algorithms are presented. According to click-through analysis in large scale search engine logs, the proposed algorithms prove to be more effective than the recommendations provided by practical search engines.

In the near future, we hope to extend this framework to embody previous proposed content and click context features. We also plan to work on a unified query correction/recommendation model for Web search engines based on our findings in this paper.

Acknowledgements

This work was supported by Natural Science Foundation (60736044, 60903107) and Research Fund for the Doctoral Program of Higher Education of China (20090002120005).

References

Agichtein, E., Brill, E., & Dumais, S. (2006). Improving web search ranking by incorporating user behavior information. In *Proceedings of SIGIR'06* (pp. 19–26). New York, NY: ACM.

Agichtein, E., Brill, E., Dumais, S., & Ragno, R. (2006). Learning user interaction models for predicting web search result preferences. In *Proceedings of SIGIR'06* (pp. 3–10). New York, NY: ACM.

Agrawal, R., Halverson, A., Kenthapadi, K., Mishra, N., & Tsaparas, P. (2009). Generating labels from clicks. In *Proceedings of WSDM'09* (pp. 172–181). New York, NY: ACM.

Baeza-Yates, R., & Tiberi, A. (2007). Extracting semantic relations from query logs. In *Proceedings of KDD*.

Baeza-Yates, R., Hurtado, C., & Mendoza, M. (2004). Query recommendation using query logs in search engines. In *EDBT* 2004 *workshop on current trends in database technology*.

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern information retrieval*. Addison-Wesley. pp. 75–79.

Bilenko, M., & White, R. W. (2008). Mining the search trails of surfing crowds: Identifying relevant websites from user activity. In *Proceeding of the* 17*th WWW conference* (pp. 51–60).

Broder, A. (2002). A taxonomy of web search. *SIGIR Forum – Fall 2002, 36*(2).

Cen, R., Liu, Y., Zhang, M., Ru, L., & Ma, S. (2009). Automatic search engine performance evaluation with the wisdom of crowds. In *Asia information retrieval symposium*.

Chen, S. F., & Goodman, J. F. (1998). *An empirical study of smoothing techniques for language modeling*. Technical report TR-10-98. Computer Science Group, Harvard University.

China Internet Network Information Center, CNNIC. (2009). CNNIC search behavior survey report. <http://research.cnnic.cn/html/1253600840d1370.html>.

Craswell, N., Zoeter, O., Taylor, M., & Ramsey, B. (2008). An experimental comparison of click position-bias models. In *Proceedings of WSDM'08* (pp. 87–94). New York, NY: ACM.

Cucerzan, S., & White, R. W. (2007). Query suggestion based on user landing pages. In *Proceedings of SIGIR* 2007, *Amsterdam, Netherland*.

Dou, Z., Song, R., Yuan, X., & Wen, J. (2008). Are click-through data adequate for learning web search rankings? In *Proceeding of CIKM'08* (pp. 73–82). New York, NY: ACM.

Fonseca, B. M., Golgher, P. B., De Moura, E. S., & Ziviani, N. (2003). Using association rules to discovery search engines related queries. In *First Latin American Web congress (LAWEB'*03*), Santiago, Chile*.

Guo, F., Liu, C., & Wang, Y. M. (2009). Efficient multiple-click models in web search. In *Proceedings of WSDM'09* (pp. 124–131). New York, NY: ACM.

Jansen, M., Spink, A., Bateman, J., & Saracevic, T. (1998). Real life information retrieval a study of user queries on the web. *ACM SIGIR Forum, 32*(1), 5–17.

Joachims, T., Granka, L., Pan, B., Hembrooke, H., & Gay, G. (2005). Accurately interpreting click through data as implicit feedback. In *Proceedings of SIGIR'05* (pp. 154–161). New York, NY: ACM.

Joachims, T., Granka, L., Pan, B., Hembrooke, H., Radlinski, F., & Gay, G. (2007). Evaluating the accuracy of implicit feedback from clicks and query reformulations in Web search. *ACM Transactions on Information Systems, 25*(2). Article 7.

Liu, Z., & Sun, M. (2008). Asymmetrical query recommendation method based on bipartite network resource allocation. In: *Proceedings of WWW* 2008, *Beijing China*.

Liu, Y., Cen, R., Zhang, M., Ma, S., & Ru, L. (2008). Identifying web spam with user behavior analysis. In *The* 4*th international workshop on adversarial information retrieval on the Web. AIRWeb'*08 (pp. 9–16). New York, NY: ACM.

Liu, Y., Fu, Y., Zhang, M., Ma, S., & Ru, L. (2007). Automatic search engine performance evaluation with click-through data analysis. In *Proceedings of WWW'07* (pp. 1133–1134). New York, NY: ACM.

Radlinski, F., & Joachims, T. (2005). Query chains: Learning to rank from implicit feedback. In *Proceedings of the ACM conference on knowledge discovery and data mining (KDD)*.

Silverstein, C., Marais, H., Henzinger, M., & Moricz, M. (1999). Analysis of a very large Web search engine query log. *SIGIR Forum, 33*(1), 6–12.

Wen, J., Nie, J., & Zhang, H. (2001). Clustering user queries of a search engine. In *Proceedings of the* 10*th international World Wide Web conference, W3C* (pp. 162–168).

Zaiane, R., & Strilets, A. (2002). Finding similar queries to satisfy searches based on query traces. In *Proceedings of the international workshop on efficient web-based information systems (EWIS), September,* 2002*, Montpellier, France*.