Task-Based User Profiling for Personalized Query Refinement

Chao Xu
Department of Information System
New Jersey Institute of Technology
cx26@njit.edu

ABSTRACT

Existing studies on user search tasks focus mainly on applying user interests in personalized search results. However, few studies examined the effect of utilizing search tasks to assist users to design effective queries through query refinement. Moreover, none of the existing studies have examined the dynamic characteristics of users and their search task. This paper proposes a novel method for user profiling which addressed the two issues above. Our approach used a two-descriptor to represent long-term and short-term user interests extracted from search sessions within a search task. It modeled the user's interests at the task level to improve the rankings of the candidate queries for query refinement through modules of task identification and updating. Experimental results showed that our task-based user-profiling method contributes to an increased precision of search results, and it produced more accurate refined queries, thereby resulting in shorter search sessions.

Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information

Search and Retrieval—Query Formulation, Search Process

General Terms

User Interests, Algorithms, Performance, Experimentation

Keywords

Query Refinement, Search Task, Search Log, Personalization

1. INTRODUCTION

The user interests derived from user logs are good indicators of user's information needs and are mostly applied in the re-ranking and/or personalization of search results. However, few studies examined the benefits of applying user interests to assist users to design queries that are more effective. Because user's diverse backgrounds and search goals affect their information needs, it is very important for users to choose appropriate keywords to describe their search intentions better. Current studies [1] have shown that users usually provide short queries. Short queries are usually ambiguous and may not have enough contexts to represent accurately the user's search intent. Many studies have been conducted aiming at helping users to build better queries. One of the famous methods is query refinement, which is a process of generating a candidate query list on the basis of the original queries of a user. By generating more effective queries, query refinement helps users to reformulate ill-formed queries and hence enhances the relevance of search results. Although numerous query refinement methods can help users to issue more effective queries than their original queries, these methods do not consider the diversity of user search intentions. For example, if two users having different interests input the same query, then the current query refinement approaches provide the same candidate query list to both users. Therefore, providing candidate query lists according to individual user interest is more beneficial than providing a generalized candidate query list. In this paper, we propose an approach for identifying and applying user interests to personalize query refinement.

Studies have shown that the vast majority of users are reluctant to provide any explicit feedback on search results and their interests [2]. Therefore, our proposed approach automatically learns user interests by using the past click history of users and applies the learned interests to analyze the user information needs. Meanwhile, studies proved that applying personalization is not always appropriate for aiding the user in accomplishing their information needs. This is due to the lack of examining and modeling the user's searching contexts and activities besides the relevance feedback [3, 4]. Consider the activity, for example, of a user who wants to purchase a coffee of a certain brand. Suppose the user has done two search tasks including purchasing a coffee machine and learning how to make good coffee. For systems which do not distinguish search tasks, they will learn that the user's interests related to coffee from his search activities. Although the user's interests are only related to coffee machines and making coffee, it would be wrong for the system to apply the learned user's interests to filter out those returned documents that are relevant to the brand of coffee. However, if now the two activities the user has done are finding companies that sell coffee and how to make good coffee, then the former search task is much more relevant to the current search activity of purchasing the coffee of a special brand. From this point of view, it is important to obtain the extent to which these two kinds of activities are relevant to the current search query. Task-oriented user search behavior analysis is a popular method for analyzing user search contexts and activities. In this paper, we propose a task-oriented personalization method that employs personalization techniques selectively for queries that are expected to benefit from the prior history information. The proposed method can detect whether the user search and browsing history is appropriate for helping users with satisfying his/her current information need.

Moreover, most of the profiling techniques usually perform single-descriptor representation to model the general search interests of users. These techniques are effective in learning user interests because they assumed that user interests change at a constant rate. However, user's interests are not static but dynamically changing. The methods ignoring the dynamic characteristics of such interests cannot adapt to the abrupt changes in user interests. Learning the dynamics of user interests is closely related to learning long-term and short-term models [5]. Long-term user interests represent the general preference of users. These interests are formed gradually over the long run and are stable after they converge. By contrast, short-term interests are unstable by nature. Short-term interests are inevitable and have commonplace in real search activities. A single-descriptor representation cannot adapt to both types of interests simultaneously. Although a system could be designed to adapt to changes in short-term user interests by maintaining a fixed amount of recent feedbacks [6], it might ignore learned long-term user interest. These two problems indicate a need to develop a representation that can trade-off the shortcomings and benefits between long-term and short-term user interest models. In this paper, we used a two-descriptor model to represent the long-term and short-term user interests for each task generated from user search histories.

A number of studies argued that user long-term and short-term interests could be calculated by time interval. For example, one study [6] denoted the final interests as sum of the weighted user interests for the past weeks, months, and all time. Although this method separated user interests from the time periods, it did not examine user interests in terms of different search contexts, such as search session and search task. The proposed method models short-term and long-term user interests at the level of search session and search task, respectively. This approach has the following main advantages: 1) Distinguishing user interests in different periods, and 2) Keeping user interests distinguishable at a unit search activity.

The goal of the study is to answer three research questions: (a) Does applying the user's interests extracted from the search log generate more effective queries? (b) Is the TOQUE project effective on using different methods as the module of query refinement? (c) Does the task-based user profiling method improve the performance of query refinement by shortening user search sessions?

This study has made the following contributions:

- extract tasks as contextual information for application in creating user-interest profiles;
- propose a four-descriptor model to learn and analyze long-term and short-term user interests;
- propose two strategies of extracting the implicit relevance feedback of users by examining the query reformulations of users;
- propose a method for generating pseudo-documents to obtain user relevance feedbacks.

The rest of the study is organized as follows. Section 2 summarizes studies that are similar to the current research. Section 3 presents task generation, user-profiling method, and candidate term generation for query refinement. Section 4 introduces the dataset, experimental design, and evaluation methods used in this study. Section 5 discusses the evaluation results for the performance comparison of the proposed system with the baseline systems. Finally, Section 6 summarizes the main contributions and future works of this study.

2. Related Work

2.1 Search Task

Search logs are viewed as rich resources for user search activities. A large number of studies have focused on methods for classifying queries into separate search goals to extract user search intentions [7]. The relationship between user information needs and original queries is also widely studied [8]. User search contexts are found to contribute much information for analyzing user search interests [9] and for improving the performance of search results ranking. By analyzing user search activities over time, we can identify queries that contain user information needs.

Most of the studies have focused on search behavior analysis within a single search session. A search session, as defined in [10], is a sequence of queries issued by a single user within a specific time limit. The related queries of the same session often refer to the same search goal or the same search activity. One study [11] proposed an algorithm to group queries into search sessions by detecting the topic shifts among queries. Another study [12] adopted topic models to extract search goals on session level. The method of examining user search activities through search sessions outperforms the traditional approaches that are based on relevance feedback. Piwowarski et al. [13] provided a hierarchy of user search activities through using a layered Bayesian network to identify the distinct patterns of user search behaviors. They used classification methods to learn the connection of latent state distribution of clicked documents to assess the relevance of the document without considering their content. Another study [14] proposed a framework for studying the sequences of the user search activities. They proposed an algorithm to segment the query streams into goals and missions.

2.2 User Profile Representation

The explicit judgments of users for the same queries differ significantly. However, the current search engines, which are designed to satisfy general user information needs, have low performance in terms of tailoring individual results. This situation provides potential for personalization [15].

The user profile is a weighted vector of keywords that is applied in the majority of personalized-information search systems. Several systems have attempted to build complicated user profiles by integrating several keyword vectors within a single profile. For example, WebMate [16] uses multiple keyword vectors for each user interest. A number of projects have explored innovative methods for long-term user profiling, such as networked profiles or ontology-based profiles.

Systems performing single-descriptor representation are effective in learning user interests because their prediction accuracy can improve substantially by using only a small amount of feedback. However, this representation cannot adapt to abrupt changes in user interests because such system assumes that user interests change at a constant rate [17].

Table 1 Sample of session division.

ID	Query	Query Time	Clicked URL	Rank
2178	*honda accord check engine light*	*2006-03-31 12:25:39*	*http://townhall-talk.edmunds.com*	*2*
2178	*honda accord check engine light*	*2006-03-31 12:25:39*	*http://www.faqfarm.com*	*1*
2178	*people search*	*2006-04-05 19:56:57*	*http://people.yahoo.com*	*1*
2178	*sprint.com*	*2006-04-05 21:23:40*	*http://www.sprint.com*	*1*
2178	*bare minerals make up*	*2006-04-07 15:36:02*	*http://www.bareminerals.com*	*1*
2178	*bare minerals make up*	*2006-04-07 15:36:02*	*http://www.essentialdayspa.com*	*3*
2178	*bare minerals make up*	*2006-04-07 15:36:02*	*http://bareescentuals.qvc.com*	*6*
2178	*amc painters crossing*	*2006-04-07 18:14:43*	*http://www.mrmovietimes.com*	*1*

3. Methods

3.1 Clustering Sessions into Tasks

Search logs are proven valuable data resource for analyzing user search activities and information needs. In this study, we examined the America Online, Inc. (AOL) search log dataset to model the dynamic search interests and preferences of users. A search log is a dataset that records user search activities, which can be denoted by the vector $<a_i, q_i, t_i, c_i, r_i>$, where a_i is the identifier of the user, q_i is the query submitted by the user, t_i is the time of the user activity, c_i is the click on the relevant result returned for q_i, and r_i is the rank position of c_i [20].

User actions include queries and result clicks. A search session is considered the basic unit of information in search log analysis [21]. A search session is defined as a sequence of search activity S $=\{<u_k, q_k, f_k, t_k>\ldots<u_j, q_j, f_j, t_j>\}$ issued by a single user within a specific time limit.

Methods of extracting relevant sessions from search logs should examine all queries issued by a user. Users tend to issue related queries within a short period and stop their search activity for a period. Therefore, time intervals are significant in detecting session boundaries [6]. In this study, we adopt user inactivity periods to segment the search session. The time interval within a search session should be less than 10 min. Table 1 shows a sample of segmented sessions.

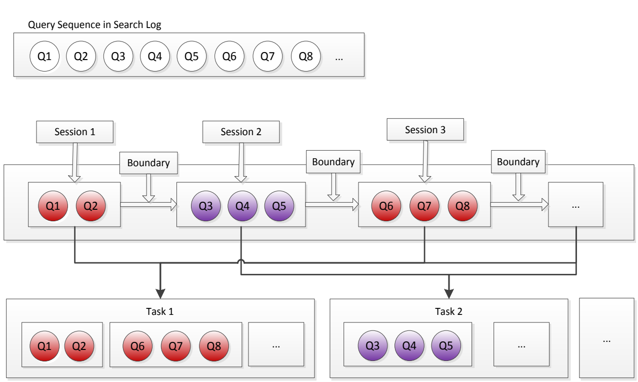

Figure 1 Example of Search Tasks.

Task is defined as a unit of session that corresponds to a search intent [22]. We extract task on the basis of the segmented session information, which is also used as the unit for extracting user interests. We described methods of extracting task-based user interests in the next section.

A number of studies [12] have adopted supervised methods in labeling search tasks by using an external dataset, such as the Open Directory Project. However, this approach has two disadvantages. First, the amount of candidate-refined queries generated for some categories may be small because the labels and categories are generated from an external dataset. Second, the total number of labels or categories of search tasks are fixed rather than adaptive to the dynamic search interests of users. However, in reality, most users have multiple information needs that are dynamically changing [17]. Thus, we adopt an unsupervised method called *hierarchical clustering* to generate tasks from the training set. Furthermore, we propose an algorithm to update the existing tasks and add new ones automatically. Figure 1 demonstrates the session division and task extraction from the raw query sequences of search logs.

3.2 Constructing User Profiles

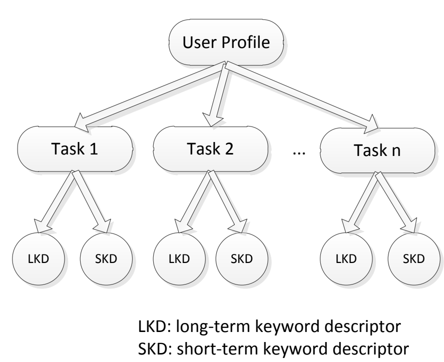

Figure 2 Task-based User Profile.

Figure 2 shows the user profile proposed in this study. The proposed approach is a task-oriented method for user profiling, in which the user interests are learned within the context of a particular search task. In a search task, user interests are modeled by a keyword descriptor (KD) which is represented by a term-weight vector, as shown in Equation 1.

$$KD = << t, w_1 >< t_2, w_2 > \ldots < t_n, w_n >> \quad (1)$$

Each KD is represented by two descriptors, namely, long-term keyword descriptor (LKD) and short-term keyword descriptor (SKD). User interests can be represented by the following two-level descriptor:

$$KD = < LKD, SKD > \quad (2)$$

This separation aims to preserve the feature vectors of relevant and non-relevant documents, thus enabling the separate measurement of the similarities between subjects of interest and the positive and negative interests. The degree of interest of users in a subject is computed by subtracting the user interest when negative descriptors are used from that when positive descriptors are used. The relevance feedback of all session data within a search task is used to model the long-term user interests, whereas the current user search session is used to model the short-term user interests.

A large number of algorithms can effectively obtain user relevance feedback. The Rocchio algorithm is one of the most famous algorithms and is widely used in information retrieval. Equation 3 represents the general form of the query refinement of the Rocchio algorithm during the relevance feedback process [17].

$$Q_{i+1} = Q_i + \alpha \sum_{pos} D_j / n_{pos} - \beta \sum_{neg} D_j / n_{neg} \quad (3)$$

where $\alpha + \beta = 1$, n_{pos} is the number of relevant documents, and n_{neg} is the number of non-relevant documents.

In this paper, we simply apply this algorithm to learning user's long-term and short-term interests. Long-term interests are modeled by a long-term descriptor LKD, which is updated using the following learning rule adopted from the Rocchio algorithm:

$$LKD_{new} = LKD_{old} + bD_{pos} - (1 - b) D_{neg} \quad (5)$$

Given a query Q, the interest in Q is denoted by $I_{LTD}(Q)$,

$$I_{LKD}(Q) = SIM(Q, LKD) \quad (6)$$

A short-term interest model is learned from the following formula:

$$SKD_{new} = SKD_{old} + cD_{pos} - (1 - c) D_{neg} \quad (7)$$

Given a query Q, the short-term interest in Q is denoted by $I_{SKD}(Q)$, which is calculated by using Equation 8:

$$I_{SKD}(Q) = SIM(Q, SKD) \quad (8)$$

Given a query Q, the combination of long-term and short-term interest is denoted by $I_{TD,}$ which is defined by

$$I_{TD} = \mu I_{LKD}(Q) + (1-\mu) I_{SKD}(Q) \quad (9)$$

where μ determines the impact weight of long-term interest in this mixture interest.

Considering that the negative (irrelevant) documents are not available in the AOL dataset, D_{neg} is not included in the calculation of LKD and SKD.

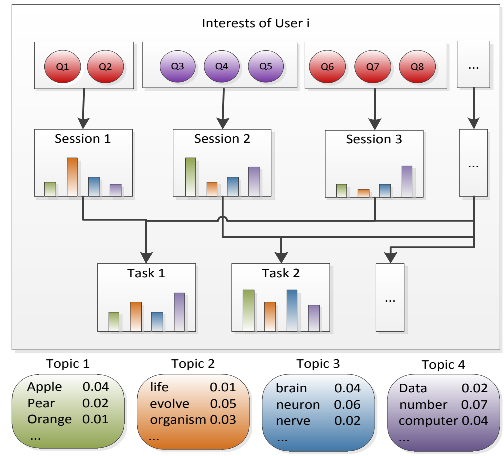

Figure 3 Example of User's Interests

Figure 3 presents an example of modeling the dynamics of user's interests within a search task. In the figure, we list a set of 4 topics. For each topic, we show a sample of words with probability values. We observe that we can filter out user's short interests of each search session. For example, in session 1, the user has a short-term interest in topic 1. Yet as we learn user's interests from session to session of task 1, we find that the user has an interest increase on topic 4 and a decrease on topic 1.

3.3 Modeling User Interest

In an actual search activity, the session data increase as the user inputs new queries. Determining whether the new query benefits from using learned task-oriented user interests is necessary. To address this problem, we propose an interest-selecting algorithm, as shown in Figure 4.

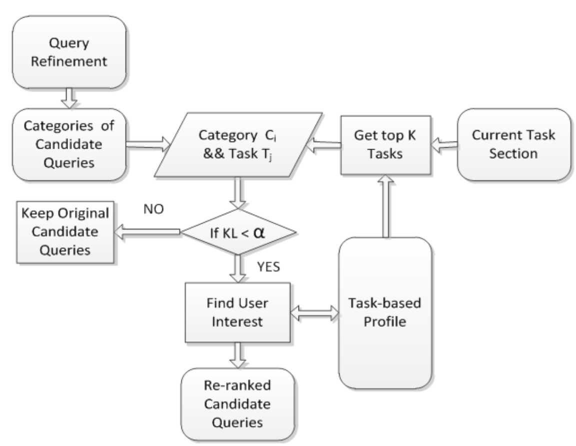

Figure 4: Interest-selecting algorithm.

For a new search query, a candidate query list is generated from the query refinement system. The top 50 candidate queries are grouped into *m* categories by using a hierarchy clustering algorithm. The category C_i is compared with an existing task T_j to determine whether this task belongs to a historical search task by calculating the Kullback–Leibler (KL) divergence [23] between each result cluster and task, as shown in Equation 11.

$$KL(C_i||T_j) = \sum_w P(w|C_i)\ log\ P(w|C_i)/P(w|T_j) \qquad (11)$$

Here, *w* represents the keyword used by the user.

The current search activity does not belong to any existing searching task if the KL divergence is above the predefined threshold α (α is set to 0.1 in this study). In this case, no personalization will be performed based on the current user search activity. Otherwise, we perform re-ranking of candidate queries via KD, which has the lowest α value to the current search task. An advantage of this method is that user interests are applied to those queries that benefit from its application.

3.4 Query Refinement

We adopt the most widely used co-occurrence development method, mutual information (MI), as the baseline system for generating a candidate query list for an original query. This method is used to calculate semantically similar candidate terms for an original term of a query. For any two words, *w1* and *w2*, MI can be computed by using the following Equation [26]:

$$I(w_1, w_2) = \sum_{T_{w_1}, T_{w_2} \varepsilon \{0,1\}}^{\infty} P(T_{w_1}, T_{w_2})\ log\ \frac{P(T_{w_1}, T_{w_2})}{P(T_{w_1})P(T_{w_2})} \qquad (12)$$

where T_{w_i} denotes the presence/absence of w_i in the pseudo-documents generated from the AOL dataset explained as follows.

3.5 System Framework

There are mainly three parts of the framework. First, sessions are extracted from the search-log training dataset. Each session is viewed as a pseudo-document represented by a keyword vector. After the hierarchical clustering algorithm is applied to the sessions, *K* clusters, which are represented by a keyword vector, are extracted. These tasks are then recorded into the user profile.

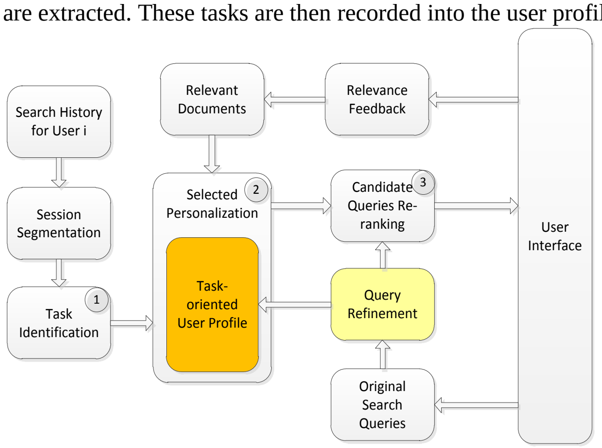

Figure 5: Task-based personalization framework.

Second, the user relevance feedback is learned through the Rocchio algorithm and then applied to update the existing user profile. Combined with the search activity information, relevance feedbacks are added separately to long-term and short-term descriptors.

Third, once the user inputs a query, the top *N* candidate queries generated from the traditional query refinement method are grouped into *m* categories, which are then compared with existing tasks T_j to determine whether the categories belong to an existed search task. The candidate queries are re-ranked according to the similarity value between categories and tasks (both of which are represented by keyword vectors) based on user interests. Finally, the original and new ranks are merged by using Borda's ranking fusion method [28].

4. Experimental Design

4.1 Data Sets and Preprocessing

The dataset we used was an AOL query log. The collection period began on 1 March 2006 and ended on 31 May 2006. This dataset contained 19,442,629 lines of click-through information, 657,426 unique user IDs, 4,802,520 unique queries, and 1,606,326 unique URLs.

Sessions boundaries were determined by using a two-step method [11]. Every two consecutive queries within the same session should share at least one term. The time interval within a session was less than 10 min. After session division, we split the dataset into training and test sets. The training set contained two-month-worth of search log data, whereas the test set contained one-month-worth of search log data. Pseudo-documents were constructed for each URL contained in the training and test set. These pseudo-documents were used to represent the content of each clicked URL in the AOL dataset.

4.2 Evaluation Method

Performing manual evaluation of the output of query refinement is time consuming and labor intensive; therefore, we evaluated the output by utilizing the session information of query logs in the same manner as [18]. In a specific search session, when a user feels unsatisfied with the results of the current query, the user may refine the query and run a new search. When the user obtains satisfactory search results, the user may stop searching and start a new search activity. On the basis of the discussion by Downey et al. [8] on the importance of the terminal URL, we can conduct a reliable evaluation by using the terminal URL information. We defined two types of queries, which are mentioned in [18], as follows:

Definition 1 (satisfied query): In a user session, the query causing at least one URL to be clicked and is located at the end of the session is called a satisfied query.

Definition 2 (unsatisfied query): Any query located just ahead of the satisfied query within the same user session is called an unsatisfied query.

We adopted the metric P@K (precision at K) to evaluate the results. The satisfied query for each session is counted as a relevant query in calculating the precision value. Here, *K* is the number of top queries given by the model. A maximum of 30 result queries were considered for each method.

4.3 Experimental Setup

The experiment analyzed the contributions of our proposed user-profiling method to two baseline systems, namely, an MI model and a context-based mutual information (CMI) model, which are presented in [19]. In this study, we used these two baseline methods to generate the original candidate query list of query refinement. We then applied the proposed model to re-rank these two candidate lists. Specifically, we obtained two personalized models, namely, personalized mutual information (P-MI) and personalized context-based mutual information (P-CMI) models, after using MI and CMI, respectively, as query refinement modules of the proposed query refinement framework.

We generated user profiles by randomly selecting 400 users with more than 50 sessions in our AOL training set. A total of 100 user sessions were used in the parameter determination experiment, whereas the sessions of the other 300 users were used to compare the effectiveness of the two pairs of systems mentioned above. In the second experiment, the first 25 sessions of each user were used to set up the initial task-based interests of users, and the next 25 sessions were used to learn the long- and short-term user interests and evaluate the effectiveness of the system. Note that for each session, the last query with at least one clicked document was used as the satisfied query for evaluation.

5. Results and Discussions

We determined the optimal value for the threshold α as mentioned in section 3.5 by varying its value with {0.1, 0.2, 0.3, 0.4, 0.5}. And we finally chose 0.3 so that the performance of selective personalization framework was maximized. The performance of both models is given in Figure 6. It can be observed that P-MI and P-CMI outperformed MI and CMI significantly when K is small. For example, The P@15 value of P-MI is 0.026, while the P@5 value of MI is 0.022. The P@5 value of P-CMI is 0.055, while the P@5 value of CMI is 0.052. The differences between the performances of our method and CTA are statistically significant ($p<0.05$). The major reason for the performance difference is that our proposed method scores a query taking into account the user's interest within a task level, which cannot be captured by the baseline approaches.

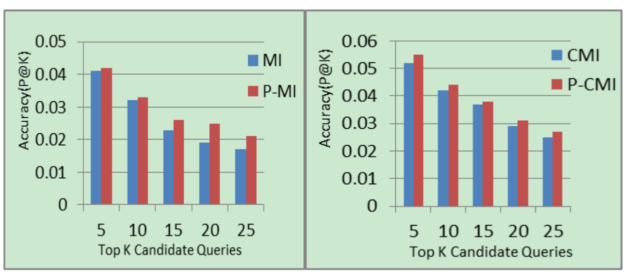

Figure 6: Comparison of scoring performance between MI and P-MI and between CMI and P-CMI.

Currently, the proposed model considers only individual user's search history. One future direction of this study is to improve our model such that it can utilize multiple users' search histories to take advantage of collaborative wisdom.

6. References

[1] J. Guo, G. Xu, H. Li, and X. Cheng, "A unified and discriminative model for query refinement," Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, Singapore, Singapore, 2008.

[2] F. Qiu and J. Cho, "Automatic identification of user interest for personalized search," Proceedings of the 15th international conference on World Wide Web, Edinburgh, Scotland, 2006.

[3] J. Luxenburger, S. Elbassuoni, and G. Weikum, "Task-aware search personalization," Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, Singapore, Singapore, 2008.

[4] J.-w. Ahn, P. Brusilovsky, J. Grady, D. He, and S. Y. Syn, "Open user profiles for adaptive news systems: help or harm?," Proceedings of the 16th international conference on World Wide Web, Banff, Alberta, Canada, 2007.

[5] D. H. Widyantoro, T. R. Ioerger, and J. Yen, "Learning user interest dynamics with a three-descriptor representation," Journal of the American Society for Information Science and Technology, vol. 52, pp. 212-225, 2001.

[6] A. Ahmed, Y. Low, M. Aly, V. Josifovski, and A. J. Smola, "Scalable distributed inference of dynamic user interests for behavioral targeting," Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, San Diego, California, USA, 2011.

[7] Y.-S. Chang, K.-Y. He, S. Yu, and W.-H. Lu, "Identifying User Goals from Web Search Results," Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, 2006.

[8] D. Downey, S. Dumais, D. Liebling, and E. Horvitz, "Understanding the relationship between searchers' queries and information goals," Proceedings of the 17th ACM conference on Information and knowledge management, Napa Valley, California, USA, 2008.

[9] R. W. White, P. Bailey, and L. Chen, "Predicting user interests from contextual information," Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, Boston, MA, USA, 2009.

[10] A. Kotov, P. N. Bennett, R. W. White, S. T. Dumais, and J. Teevan, "Modeling and analysis of cross-session search tasks," Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, Beijing, China, 2011.

[11] D. He, A. Göker, and D. J. Harper, "Combining evidence for automatic Web session identification," Information Processing & Management, vol. 38, pp. 727-742, 2002.

[12] A. Hassan, R. Jones, and K. L. Klinkner, "Beyond DCG: user behavior as a predictor of a successful search," Proceedings of the third ACM international conference on Web search and data mining, New York, New York, USA, 2010.

[13] B. Piwowarski, G. Dupret, and R. Jones, "Mining user web search activity with layered bayesian networks or how to capture a click in its context," Proceedings of the Second ACM International Conference on Web Search and Data Mining, Barcelona, Spain, 2009.

[14] Q. Mei, K. Klinkner, R. Kumar, and A. Tomkins, "An analysis framework for search sequences," Proceedings of the 18th ACM conference on Information and knowledge management, Hong Kong, China, 2009.

[15] C. Chen, M. Yang, S. Li, T. Zhao, and H. Qi, "Predicting query potential for personalization, classification or regression?," Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, Geneva, Switzerland, 2010.

[16] L. Chen and K. Sycara, "WebMate: a personal agent for browsing and searching," Proceedings of the second international conference on Autonomous agents, Minneapolis, Minnesota, United States, 1998.

[17] D. H. Widyantoro, T. R. Ioerger, and J. Yen, "An adaptive algorithm for learning changes in user interests," Proceedings of the eighth international conference on Information and knowledge management, Kansas City, Missouri, United States, 1999.

[18] L. Bing, W. Lam, and T.-L. Wong, "Using query log and social tagging to refine queries based on latent topics," Proceedings of the 20th ACM international conference on Information and knowledge management, Glasgow, Scotland, UK, 2011.

[19] X. Wang and C. Zhai, "Mining term association patterns from search logs for effective query reformulation," Proceedings of the 17th ACM conference on Information and knowledge management, Napa Valley, California, USA, 2008.

[20] B. Zhou, D. Jiang, J. Pei, and H. Li, "OLAP on search logs: an infrastructure supporting data-driven applications in search engines," Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, Paris, France, 2009.

[21] B. Tan, X. Shen, and C. Zhai, "Mining long-term search history to improve search accuracy," Proceedings of the 12th ACM SIGKDD

international conference on Knowledge discovery and data mining, Philadelphia, PA, USA, 2006.
[22] J. Luxenburger, S. Elbassuoni, and G. Weikum, "Matching task profiles and user needs in personalized web search," Proceedings of the 17th ACM conference on Information and knowledge management, Napa Valley, California, USA, 2008.
[23] B. Bigi, "Using Kullback-Leibler distance for text categorization," Proceedings of the 25th European conference on IR research, Pisa, Italy, 2003.
[24] D. Jiang, K. W.-T. Leung, and W. Ng, "Context-aware search personalization with concept preference," Proceedings of the 20th ACM international conference on Information and knowledge management, Glasgow, Scotland, UK, 2011.
[25] T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay, "Evaluating the accuracy of implicit feedback from clicks and query reformulations in Web search," ACM Transaction of Information System, vol. 25, p. 7, 2007.
[26] G. Wang and F. H. Lochovsky, "Feature selection with conditional mutual information maximin in text categorization," Proceedings of the thirteenth ACM international conference on Information and knowledge management, Washington, D.C., USA, 2004.
[27] H. C. Wu, R. W. P. Luk, K. F. Wong, and K. L. Kwok, "Probabilistic document-context based relevance feedback with limited relevance judgments," Proceedings of the 15th ACM international conference on Information and knowledge management, Arlington, Virginia, USA, 2006.
[28] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar, "Rank aggregation methods for the Web," Proceedings of the 10th international conference on World Wide Web, Hong Kong, Hong Kong, 2001.
[29] Wetzker, Robert, Zimmermann, Carsten, & Bauckhage, Christian. Analyzing Social Bookmarking Systems: A del.icio.us Cookbook. Study the Mining Social Data (MSoDa) Workshop Proceedings. 2008.