Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

ABSTRACT

TASK-BASED USER PROFILING FOR QUERY REFINEMENT (TOQUE)

by
Chao Xu

Advisor: Dr. Yi-fang Brook Wu

The information needs of search engine users vary in complexity. Some simple needs can be satisfied by using a single query, while complicated ones require a series of queries spanning a period of time. A search task, consisting of a sequence of search queries serving the same information need, can be treated as an atomic unit for modeling user's search preferences and has been applied in improving the accuracy of search results. However, existing studies on user search tasks mainly focus on applying user's interests in re-ranking search results. Only few studies have examined the effects of utilizing search tasks to assist users in obtaining effective queries. Moreover, fewer existing studies have examined the dynamic characteristics of user's search interests within a search task. Furthermore, even fewer studies have examined approaches to selective personalization for candidate refined queries that are expected to benefit from its application. This study proposes a framework of modeling user's task-based dynamic search interests to address these issues and makes the following contributions. First, task identification: a cross-session based method is proposed to discover tasks by modeling the best-link structure of queries, based on the commonly shared clicked results. A graph-based representation method is introduced to improve the effectiveness of link prediction in a query sequence. Second, dynamic task-level search interest representation: a four-tuple user profiling model is introduced to

represent long- and short-term user interests extracted from search tasks and sessions. It models user's interests at the task level to re-rank candidate queries through modules of task identification and update. Third, selective personalization: a two-step personalization algorithm is proposed to improve the rankings of candidate queries for query refinement by assessing the task dependency via exploiting a latent task space. Experimental results show that the proposed TOQUE framework contributes to an increased precision of candidate queries and thus shortened search sessions.

TASK-BASED USER PROFILING FOR QUERY REFINEMENT (TOQUE)

by
Chao Xu

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Information Systems

Department of Information Systems

January 2016

.

APPROVAL PAGE

TASK-BASED USER PROFILING
FOR QUERY REFINEMENT (TOQUE)

Chao Xu

Dr. Yi-fang Brook Wu, Dissertation Advisor — Date
Associate Professor of Information Systems, NJIT

Dr. Yi Chen, Committee Member — Date
Associate Professor of Management, NJIT

Dr. Zhi Wei, Committee Member — Date
Associate Professor of Computer Science, NJIT

Dr. Songhua Xu, Committee Member — Date
Assistant Professor of Information Systems, NJIT

Dr. Lian Duan, Committee Member — Date
Assistant Professor of Information Systems and Business Analytics, Hofstra University

BIOGRAPHICAL SKETCH

Author: Chao Xu

Degree: Doctor of Philosophy

Date: January 2016

Undergraduate and Graduate Education:

- Doctor of Philosophy in Information Systems,
  New Jersey Institute of Technology, Newark, NJ, 2016

- Master of Science in Information Engineering,
  China University of Petroleum, Shandong, P. R. China, 2010

- Bachelor of Science in Electronic Information Engineering,
  Shandong University of Technology, Shandong, P. R. China, 2007

Major: Information Systems

Presentations and Publications:

Chao Xu, Mingzhu Zhu, Wei Xiong, Yi-Fang Brook Wu (2015). Graph-based Link Prediction in Cross-session Task Identification. *2015 International Conference on Data Mining (DMIN 2015)*, Las Vegas, NV.

Chao Xu, Mingzhu Zhu, Yanchi Liu, Yi-Fang Brook Wu (2014). Personalizing Query Refinement Based on Latent Tasks. *2014 International Conference on Data Mining (DMIN 2014)*, Las Vegas, NV.

Chao Xu, Mingzhu Zhu, Yanchi Liu, Yi-Fang Brook Wu (2014). User Profiling for Query Refinement. *20th Americas Conference on Information Systems (AMCIS 2014)*, Savannah, GA.

Chao Xu, Yi-Fang Brook Wu (2013). Task-based User Profiling for Personalized Query Refinement. *ACM and IEEE 2013 Joint Conference on Digital Libraries (JCDL 2013, Doctoral Consortium)*, Indianapolis, IN.

This dissertation is dedicated to my beloved family

To my parents, parents-in-law,
My beloved wife,
With whom I have shared
Many precious moments of my life.

致我挚爱的家人

ACKNOWLEDGMENT

I would like to express my deep and everlasting gratitude to my dissertation advisor, Dr. Yi-Fang Brook Wu, for her constant help and guidance while conducting my dissertation research. I would also like to especially thank all my other committee members, Dr. Songhua Xu, Dr. Lian Duan, Dr. Yi Chen, and Dr. Zhi Wei, who have provided excellent comments in the development of the research study and the analysis of experimental results. It has been a great honor to know the members of my committee, and I cannot thank them enough for their hard work and dedication to make this dissertation possible.

Finally, I would like to thank Dr. Mingzhu zhu, Dr. Regina Collins, Yanchi Liu, Chris Markson and Chong Wang from the Information Systems Department for their encouragement and guidance as I completed my research.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

Chapter **Page**

TABLE OF CONTENTS
(Continued)

Chapter **Page**

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES
(Continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

1.1 Background and Motivation

User's search interests derived from user logs are good indicators of their information needs and mostly applied in the re-ranking and/or personalization of search results. However, only few studies have examined the benefits of applying user's interests to assist them in obtaining effective queries. To retrieve better search results, it is important for users to choose appropriate keywords to describe their search intentions, which is not easy when users are unfamiliar with a search topic. Current studies (Guo et al., 2008; Hassan & White, 2012) have shown that users often provide short queries. Short queries, however, are usually ambiguous and may not accurately represent users' search intentions. Many studies have been conducted aiming at helping users build more effective queries. Among them, query refinement developed by Wang and Zhai (2008) is defined as a process of generating a candidate query list based on the original queries of a user. By generating more effective queries, query refinement helps users reformulate ill-formed queries to enhance the relevance of search results.

Current search engines usually apply query refinement to complement the search results page with a candidate query list. This list of queries is usually placed on the left column or at the bottom of the search results page (in Google, Yahoo, and Bing). These queries help users find and explore information related to their original query. As shown in Figure 1.1, for example, once a user inputs the query "Java", Yahoo will return candidate queries including "java script", "java games", "java runtime environment", "adobe", and

"flash". Among the traditional methods of query refinement, Mutual Information (MI) is widely used. Using MI, the words that have a high probability to co-occur with the words in the original query, are extracted and used to replace the original words. Based on MI, some studies (Bar-Yossef & Kraus, 2011; Wang & Zhai, 2008) have added context information for each query to improve the performance of predicting users' information needs. One study (Bing, Lam, & Wong, 2011) adopts a topic model based method to extract latent topics from user search histories to assess the semantic dependency among the words within a candidate query. Yet one issue is that they fail to consider users' diverse search intentions. In Figure 1.1, if two users having different interests such as coffee and programming language input the same query "java", the current query refinement approaches provide the same candidate query list to both users. However, this query list is irrelevant to the user who wants to find information about coffee rather than programming language. Thus, providing different candidate query lists based on each individual's search interests will be more beneficial than providing a generalized candidate query list. This work centers on the development of an effective framework for individualized query refinement. The goal is to provide more effective candidate queries, and thus user's information needs will be satisfied faster.

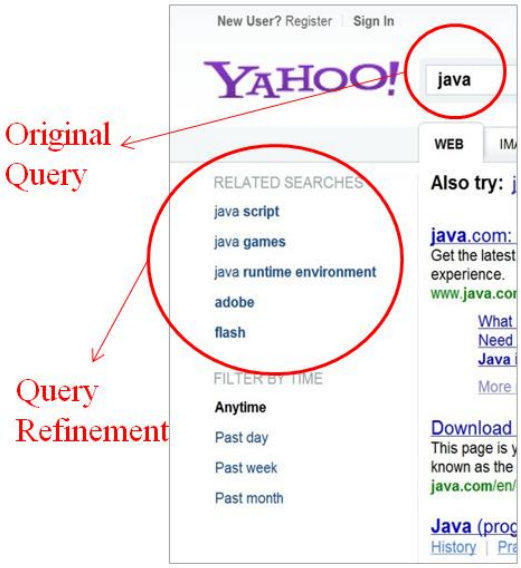


Figure 1.1 Example of query refinement.

It is apparent that, providing individualized candidate queries requires user's search interests as input. However, White et al. (2009) have shown that most of the users are reluctant to provide any explicit feedback on search results and their interests. Therefore, a personalized search engine intended for a large audience must learn user's preferences implicitly without requiring any explicit feedback from the user. Given the lack of explicit user feedback, in this research, learning and updating user's dynamic interests automatically based on her/his past click history is one of the main research problems.

Although personalization is a notable goal, studies (Ahn et al., 2007; Dou, Song, & Wen, 2007) have proven that applying personalization by learning users' interests from past search histories is not always effective in aiding users in satisfying their information needs. This is due to the lack of examining and modeling user's searching contexts and activities (Ahn et al., 2007; Rose & Levinson, 2004) in the personalization process. In fact, it is crucial to restrict personalization only to queries that benefit from its application.

Task-oriented user search behavior analysis is a popular method to analyze user's search activities based on the session information obtained by segmenting query sequences using the time interval between queries (Ahn et al., 2008). However, only few studies have examined how to apply it into modeling user's dynamic search interests. As such, in this research, a task-based personalization algorithm is proposed which selectively employs personalization techniques for queries that are expected to benefit from the user's prior search history.

A robust algorithm to learn user's interests is vital to such a framework. Rocchio is a very well-known algorithm for relevance feedback (Rocchio, 1971). It has been applied in learning user's interests in query refinement by using user's positive and negative feedback for learning new interests and unlearning old interests respectively. The algorithm, using information from user's relevance feedback, works on a bag-of-words representation. The bag-of-words representation is a list of attribute-value pairs (feature vector), where an attribute represents a feature (e.g., a word in a text document) and its value indicates the feature weight. Although systems that adopt bag-of-words based representation are effective in learning users' general interests, this representation cannot adapt to users' abrupt interest changes flexibly, because it assumes that the user's interests change at a constant rate.

Generally speaking, user's interests can be divided into two types: long- and short-term interests (Deng, King, & Lyu, 2009). Long-term interests indicate a user's general preferences (Haveliwala, 2002), which are formed gradually over the long run and are stable after they converge. By contrast, short-term interests are unstable by nature. The bag-of-words representation, however, cannot adapt to both types of search interests at the

same time. Thus, in this study, a fine-grained model which supports both long- and short-term interest learning and updating is needed to improve user-interest representation.

1.2 Scope of the Study

This research aims to study three main research objectives: 1) extracting user's task information based on their past search histories; 2) learning user's dynamic interests through the development of task-based user profiles according to their queries and click data; and 3) utilizing the learned interests to personalize query refinement for improving the precision of candidate query list. To achieve these, a fine-grained task identification method is introduced to extract search tasks by modeling the best-link structure of queries within each user search session. Moreover, query ambiguity is always a main problem in providing effective candidate queries. To solve it, Latent Dirichlet Allocation (LDA) is adopted to learn user's interests in the topic space. Based on LDA, a multi-descriptor based user profiling method is proposed to learn and predict user's dynamic search interests, including long- and short-term interests, in which a descriptor is a list of attribute-value pairs. Since adapting to a user's interests consists of learning new (positive) interests and unlearning old (negative) interests, positive and negative interests are modeled for both long- and short-term interest model respectively, thus resulting in a four-tuple descriptor (i.e., positive long-term, negative long-term, positive short-term, and negative short-term descriptor) representation. Furthermore, a personalization algorithm is proposed to utilize user's task-based search interests to personalize the candidate queries generated by traditional query refinement algorithms. In summary, to conduct such a study, three major research questions need to be investigated:

Research Question No.1:

> Is the proposed best-link based task identification method which incorporates latent structure of queries more effective than baselines?

Research Question No.2:

> Is the multi-descriptor based user profiling method more effective in learning user's dynamic search interests than bag-of-words based methods?

Research Question No.3:

> Is the proposed personalization algorithm effective in improving the performance of traditional query refinement methods?

1.3 Overview of the Research

To answer these three major research questions, Task-based user prOfiling for QUEry refinement (TOQUE) is proposed based on modeling user's long- and short-term interests within tasks and sessions respectively for query refinement. Figure 1.2 shows four major components of TOQUE.

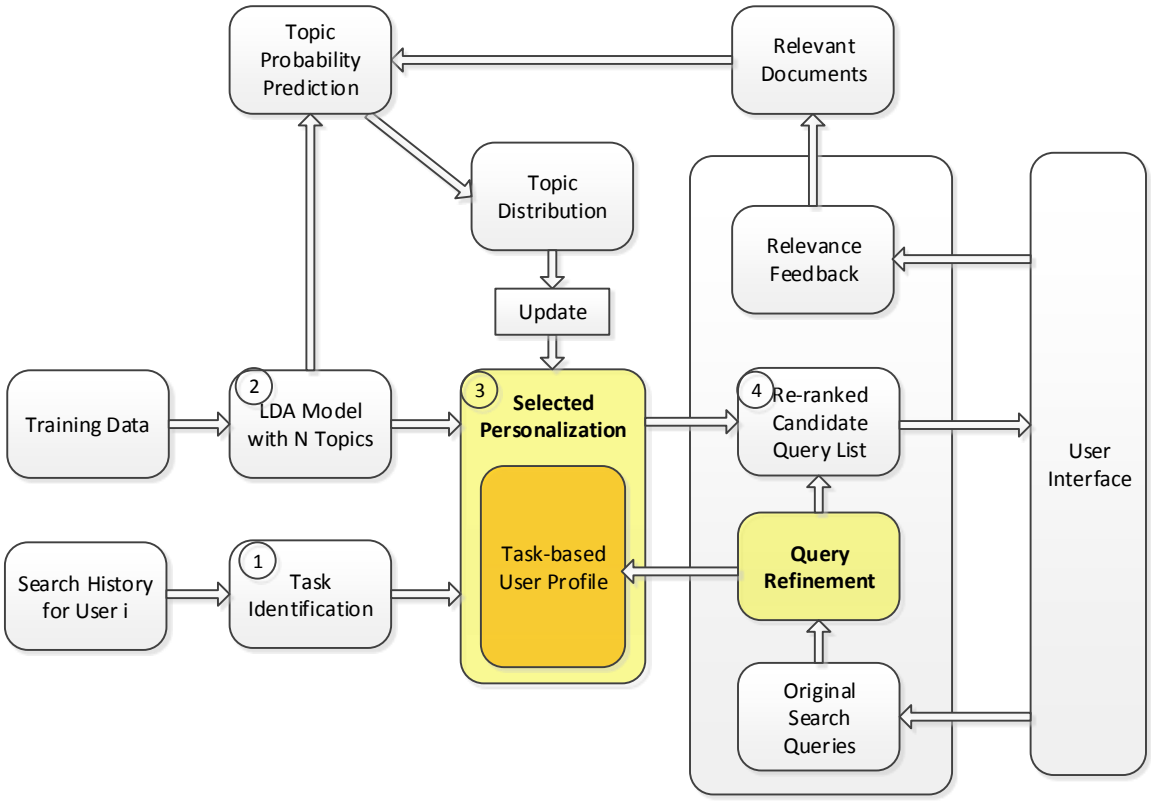


Figure 1.2 Framework of task-based personalization for query refinement (TOQUE).

The first major component, "Task Identification", aims to extract task information from a user's search history. From the training log dataset, session boundaries are identified using the time interval between queries. After the session segmentation, a best-link model is conducted for task identification within each search session. Task information plays two important roles in the framework. First, the task information is used as basic unit of modeling user's search interests. Specifically, the relevance feedback within a search task is used to model the long-term user interests, whereas the one within the current user search session is used to model the short-term user interests. These user interests are recorded into the user profile, which are represented by a four-tuple topic descriptor. Second, the task information is also used to selectively apply user's search interests on his current search activity.

In the second major component, “LDA Model with N Topics”, a portion of the relevance feedback documents in AOL search log are used for training a topic model (i.e., LDA). To avoid the over-fitting issue, these clicked URLs are excluded from the dataset which are used in other components of the framework. Then, pseudo-documents (Bing, Lam, & Wong, 2011) are created by combining all the queries which are connected to a same URL. Then the pseudo-documents are utilized, rather than the original clicked documents, for training the LDA model. The trained model is then used to represent user’s long- and short-term interests in the third component, and the relevant feedback in the fourth component.

The third major component, “Selected Personalization”, keeps a four-tuple descriptor based user profile, in which user’s interests are learned for each particular task. Since user’s interests can change dynamically, the user profile should capture user’s long- and short-term interests separately. Using the trained LDA, the user’s relevance feedback is first represented with topic distributions. Then Rocchio algorithm is adopted to update the existing user profile using user’s relevance feedback. Combined with the search activity information, the relevance feedback of a search task and the current search session are added to the long- and short-term descriptor separately.

The fourth major component, “Re-ranked Candidate Query List”, aims to apply related user’s interests to personalize query refinement, extract user’s current search interests, and update the task-based user profile. Once a user inputs a query, candidate queries generated from the traditional query refinement method are grouped into categories. Each candidate query is represented as a topic distribution, and each category is represented as the summation of these topic distributions. Then each category is compared

with the existing tasks to determine which task they should be assigned to. These candidate queries are then re-ranked according to the KL divergence similarity value between each of the m categories and each task-based user's interest, both of which are represented by the LDA model. If the value is above the predefined threshold, no personalization will be applied. Otherwise, the candidate queries will be re-ranked based on the user's search interests.

1.4 Organization of the Dissertation

The remainder of this study is organized as follows. Chapter 2 provides a review of related studies. It presents the background of query refinement and an overview of search activities (i.e., search session and search task). It also discusses topic model and user modeling methods. Chapter 3 introduces a best-link model of task identification. To enhance the pairwise link prediction, a graph-based representation method is proposed for comparing the contextual similarity between two queries. Chapter 4 theorizes a four-tuple topic descriptor user profiling to learn and analyze long- and short-term user interests, by adopting LDA to extract user's search interests from their relevance feedback. Chapter 5 discusses a two-step personalization method of query refinement using user's task-based search interests. Chapter 6 summarizes this study and discusses the limitations. Chapter 7 illustrates the discussion and contributions.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Query refinement, as a technique to improve user's ill-formed queries, has been widely adopted in current search engines. Yet none of them considers the user's diverse search interests and different users can use the same search query for different information needs. Search logs, as a valuable data source for extracting user's search interests, have been utilized in many personalization applications. Yet personalization is not always helpful since users can have different search interests in various search activities. It is necessary to restrict the personalization to the objects which will benefit from its applications. Therefore, this study attempts at finding ways of applying user's search interests in achieving selectively personalization of query refinement. This chapter introduces the background information on query refinement, search activity modeling, topic model (e.g., LDA) and log-based user profiling.

2.2 Query Refinement

Studies have shown that most queries are short so they cannot express the user's true search intentions. Query refinement, known as a process of providing users with a candidate query list based on their original queries, has attracted much attention on reducing the ambiguity of the users' queries.

More recently, studies on query refinement based on Mutual Information (MI) have been proposed. One study (Wang & Zhai, 2008) adopts context information of user queries to improve traditional MI algorithm, known as Context-based Mutual Information (CMI),

to predict users' search interests. However, studies (Lucchese et al. 2011; Kotov et al., 2011) have shown that users' explicit judgments for the same queries differ greatly. Most of the current search engines do not provide tailored candidate search queries, which provides the potential for the personalization of query refinement (Chen et al., 2010).

Many approaches have adopted the user's session or task information to improve the accuracy of query refinement. For example, Luxenburger et al. (2008) propose a personalization framework, Matching Task Profiles (MTP), to utilize the user's past similar search task information for helping the user to satisfy his current information need. Although they consider the task information for personalization, they do not examine the user's dynamic search interests within a search task. Bing et al. (2011) propose a personalization system, Latent Topic Investigation (LTI), to extract the latent topics from users' search histories and assess the semantic consistency of words in the query. Then they utilize session information to construct a Markov graph to generate candidate queries. However, they do not consider the task information in modeling user's search interests.

In this study, a personalization framework is proposed to modeling user's dynamic search interests within the search tasks by using a multi-descriptor based user profiling method. Then the candidate queries are re-ranked by a two-step personalization algorithm considering the latent task consistency using a graphical model.

2.3 Search Activity Modeling

User search contexts based analysis is found to be effective for learning user's search interests (White, Bailey, & Chen, 2009) and for improving the performance of ranking the

search results. Prior research efforts in examining user search contexts can be categorized into two directions: search session and search task.

A search session, as defined by (Boldi et al., 2008), is a sequence of queries issued by a single user within a specific time limit. The related queries of the same session often correspond to the same search goal, i.e., information need. Based on this assumption, He et al. (2002) propose to group queries into search sessions through detecting the topic shifts among queries. Hassan et al. (2012) adopt topic models to extract session-level search goals. It is concluded that the method of examining user search activities through search sessions outperforms the traditional approaches that are only based on relevance feedback. Piwowarski et al. (2009) model a hierarchy of users' search activities through a layered Bayesian network to identify distinct patterns of users' search behaviors. They use classification methods to learn the latent connection between a clicked document and the user's relevance assessment of that document without using the document content. Mei et al. (2009) propose a framework of studying the sequences of users' search activities and an algorithm of segmenting the query stream into goals.

Recently, several studies have noticed the necessity of going beyond the session boundary and examining user's information needs in a task. For example, Spink et al. (2006) indicate that multi-tasking behavior occurs frequently in which users switch search tasks within a short period of time. Lucchese et al. (2011) model task-based sessions to extract multiple tasks from the search session. Meanwhile, Hassan and White (2012) indicate that a search task can be complex and span a number of search sessions. To tackle this, they propose a method to generate a task tour which comprises a set of related search tasks. Kotov et al. (2011) explicitly define the cross-session task as the one extending over

multiple sessions and corresponding to a certain high-level search intent. To extract cross-session tasks, Jones et al. (2008) have built classifiers to identify task boundaries and pairs of queries belonging to the same task. Agichtein et al. (2012) have examined the cross-session based task identification by using a binary classification method and have found that different types of tasks have different life spans. Besides, a few studies (Anick, 2003; Shen et al., 2006; Liao et al., 2012; Dhillon, Sellamanickam, & Selvaraj, 2011) have proven the effectiveness of classifying queries and web pages into search tasks on improving the search performance. Although they prove that the search task information contributes to the improvement of search performance, all of them have two main issues. The first issue is that they define the search task manually. The fixed number of search tasks is not suited to predict the user's future search activities – since it will be an incomplete representation, if the number is too small; and noises will occur, if the number is too large. The second issue is that existing classification-based methods rely on human annotated dataset for training models, which is not applicable when only few manual annotations are available. To tackle these issues, in this study, the cross-session based task identification is modeled as a link prediction problem rather than a binary classification problem.

The advantage of this study is that the latent dependencies between queries within each task are modeled explicitly. Furthermore, this study extends previous works in two ways: 1) search tasks and sessions are utilized as the contextual information for modeling the user's long- and short-term interests, respectively; 2) search task information is integrated into the proposed personalization algorithm to improve the effectiveness of traditional query refinement methods.

2.4 Topic Model

The classic way of modeling the user's interests is word-based representation. For example, in the vector space model, the user's interest is represented as a word-value pair vector in which the value is calculated through the TF-IDF method. The features of the vector are the terms occurred in the click data, thus such vectors are also called "term vectors". The computing cost is high due to the high dimension of the term vectors. By contrast, the topic model is a type of probabilistic model supporting the idea that each document is a mixture of multiple topics. Compared to word-based models, the topic model reduces the computing dimensions enormously, i.e., from the term space to the topic space, but still preserves the essential statistical relationships between terms in the documents.

As one of the most widely adopted topic models, LDA has attracted many efforts in the past few years. It is widely used to solve text mining problems such as co-author mining. For example, LDA is adopted to construct an author topic model by representing an author with a probability distribution over topics (Song et al., 2007). Thus a paper with multiple authors can be represented by a mixture of the distributions of these authors. Rosen-Zvi et al. (2010) extend LDA to analyze the relationship between users in a social-network dataset.

Different from the studies described above, in TOQUE, LDA is adopted to model users' search interests of topics using a click graph (Deng et al., 2009; Yi & Maghoul, 2009), which is generated from users' queries and clicked URLs. From the graph, pseudo-documents (Bing, Lam, & Wang, 2011) are extracted by combining all the queries

which are connected to a particular URL. Users' interests are constructed from these pseudo-documents which are represented by topic distributions via LDA.

2.5 Log-based User Profiling

Current search engines, which are designed to satisfy general user information needs, have low performance in terms of tailoring search results for individual users, because explicit relevance judgments for the same queries differ significantly between users. To tackle it, several studies adopt a user profile to learn user's search interests and tailor search results using their search histories, in which the user profile is represented as a weighted vector of topics or keywords. For example, Gauch et al. (2003) propose to represent the user profile as a concept vector using an existing reference ontology, Magellan. Each concept has a weight indicating the score of user's search interests in this concept. The user's clicked documents are first classified into one or multiple concepts contained in the reference ontology. Then the values of concepts in the user profile are updated based on the relevance of the user's clicked documents which is determined by the user's browsing time. In a search activity, results are re-ranked based on the similarities between each pair of the user's interests and the search result. Speretta and Gauch (2005) denote the user profile as a weighted concept hierarchy, which incorporates all the categories of the top three levels of the ODP (i.e., Open Directory Project) taxonomy. In the hierarchy, each concept has a weight representing the user's interests in the specific category. These weights are assigned by using the user's clicked documents which are classified into the related categories. Specifically, a list of concepts with associated weights is generated which are learned from the user's issued queries, snippets and document contents. Then

each search result is represented using a document profile in the same vector format as the user profile. The final result list is re-ranked by comparing the pairwise similarity between each document and the user profile. Sieg et al. (2007) introduce a method of denoting the user profile as an instance of the ODP taxonomy. Each concept in the user profile is associated with a value indicating the user's interest with one as the initial value. A spreading algorithm is proposed to maintain the user's interest scores on the ODP categories based on his/her ongoing behavior, i.e., frequency of visits to a page, the amount of time spent on the page, and user actions such as bookmarking a page.

Recently, several studies have noticed the significance of incorporating the structural information in denoting user profiles. For example, Li and Kitsuregawa (2007) represent the user profile as an ontology hierarchy. Google Directory is used as the predefined taxonomy to construct user profiles. Besides, a user topic tree is proposed to maintain and update his interests. Specifically, each node in the user topic tree indicates a topic in the Google Directory, and each topic is associated with a value based on the number of times the node has been visited. Then two operations, i.e., "adding" and "deleting", are incorporated to update the structure and contents of the user profile based on user's clicking behaviors. Besides, Xu et al. (2007) propose to generate a hierarchical user profile using frequent terms. In the hierarchy, generated terms with higher frequency are placed at higher levels, while specific terms with lower frequency are placed at lower levels. Moreover, two rules are introduced to generate the relationship between the frequent terms, i.e., combining the similar terms related to the same interest and describing the parent-child relationship between terms.

Furthermore, several systems have attempted to build complicated user profiles by integrating several keyword vectors within a single profile. For example, WebMate (Chen & Sycara, 1998) uses multiple keyword vectors for each user interest. Kohlschutter et al. (2006) propose to use two taxonomies, ODP and Del.icio.us, for faceted Web search. Web pages are first classified into different categories using personalized PageRank. The classified Web pages are then compared with the explicit user category preferences from the user profiles to provide a personalized ranking on the search results. Several other studies (Ahn et al., 2007; Luxenburger, Elbassuoni, & Weikum, 2008; Liu, Belkin, & Cole, 2012) have explored innovative methods for long-term user profiling, such as network-based profiles.

Although existing studies have improved search performance by modeling user's interests using a user profile, they still suffer from two main problems. First, it is observed that most existing concept-based user profiling methods rely on a predefined taxonomy (ODP or Google Directory) to determine a user's topical preferences. However, most existing taxonomies require extra human efforts to maintain and update the categories in the taxonomies. Second, current user profile representations cannot adapt to abrupt changes in the user's interests, because they assume that the user's search interest change at a constant rate (Ahmed et al., 2011). The method proposed in this study differs from previous work in two main aspects: 1) the user's long- and short-term search interests are modeled respectively using both the user's positive and negative relevance feedback; 2) user's interests are learned and updated for each search task, so that the personalization can be restricted to related queries which benefit from its application.

2.6 Summary

Current query refinement does not consider the user's diverse search intentions, i.e., if two users having different interests input the same query, the same candidate query list is provided to both of them. None of the studies examine the effectiveness of applying the user's search interests in re-ranking the candidate queries of query refinement.

Moreover, task-oriented user search behavior analysis is a popular method to predict the user's search interests, which has been well studied in categorizing the search results. Yet none of the studies examine how to apply search task information into modeling the user's dynamic search interests for selectively applying personalization.

Finally, the traditional user profile is built using the bag-of-words based representation which cannot capture user's dynamic search interest changes. It is valuable to examine a multi-descriptor based user profiling method to learn the user's long- and short-term interests separately and simultaneously.

CHAPTER 3

TASK IDENTIFICATION

3.1 Introduction

Search engine users' information needs span a broad spectrum (Hassan & White, 2012). Simple needs, such as homepage finding, can mostly be satisfied via a single query; but users may also issue a series of queries, collect, filter, and synthesize information from multiple sources to solve a complex task, e.g., computer fixing, travel planning, etc. For example, in Figure 3.1, if a user's laptop is broken, and he wants to find the solution on the internet, usually, he will search a query first, such as "Thinkpad T410 broken", and then go through search results. If the user fails to find relevant information, he would most likely revise his query. This iterative process, as shown in Figure 3.2, will keep running until the user finds his solution or gives up his search activity.

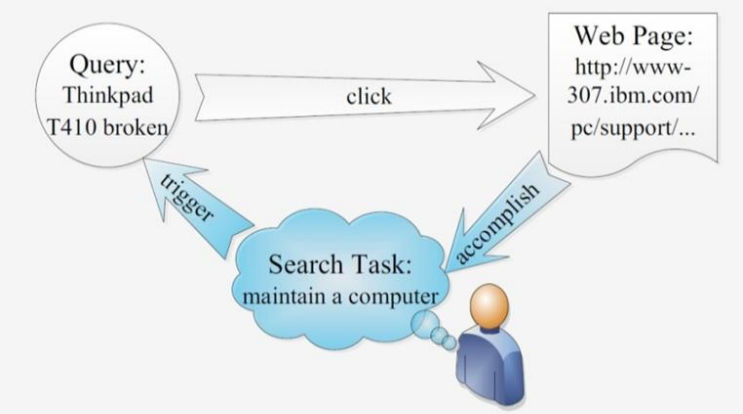


Figure 3.1 Example of a search task.

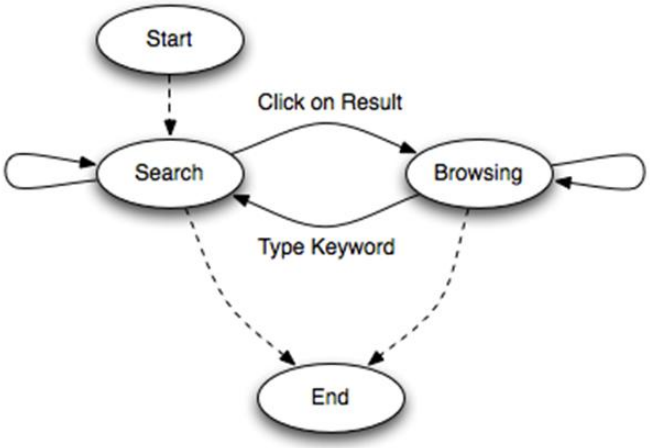


Figure 3.2 Iterative process of a user's search behavior in a search task.

To comprehensively and accurately understand these needs from recorded actions in the user logs, it is necessary to associate relevant queries together. The primary mechanisms for segmenting the logged query streams are session based. In practice, sessions are segmented using inactivity timeouts between user actions (Kotov et al., 2011). Then similar sessions are grouped together to represent a search task. Yet, in the log dataset, many tasks span multiple search sessions (Ahn et al., 2008; Ji et al., 2011), which suggests values in studying and improving task identification methods within a search session. Recently, there has been significant research on identifying tasks within these sessions, e.g., Lucchese et al. (2011) propose the concept of a "task-based session": a cluster of queries within the same session serves a particular common search intention. However, those methods are not a valid criterion for identifying the semantic structure among queries. In this chapter, a best-link model is introduced for discovering search tasks by modeling the latent link structure of queries in the search log.

3.2 Methods

3.2.1 Task Identification

Search logs are proven as a valuable data resource for analyzing the user's search activities and information needs. In this chapter, the AOL search log dataset is examined to model 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 a_i, 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 (Zhou et al., 2009).

A search session is usually considered the basic unit of information in search log analysis (Tan, Shen, & Zhai, 2006). In a search engine which works in the session mode, the user's current search activities are recorded and past search data in the same session such as queries and clicks are used to update user's current search results. A search session is defined as a sequence of search activities $S = \{< a_k , q_k , t_k , c_k , r_k > \ldots < a_j , q_j , t_j , c_j , r_j >\}$ 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. Short inactivity timeouts between user actions are applied as a means of demarcating session boundaries (Boldi et al., 2008). In the field of session segmentation, the relations between queries are categorized as Topic Continuation and Topic Shift. In Figure 3.3, query q_1 and q_2 are semantically related, so they should be grouped in the same session and the relation between them is Topic Continuation. On the contrary, q_2 and q_3 have no semantic relation, so the relation between them is Topic Shift, which generates a session boundary. However, since topic shift is difficult to detect, in

practice, user inactivity periods are adopted to segment the search session. The time interval within a search session should be less than a threshold σ (where σ is set at 25 minutes according to an empirical study). Table 3.1 shows a sample of segmented sessions.

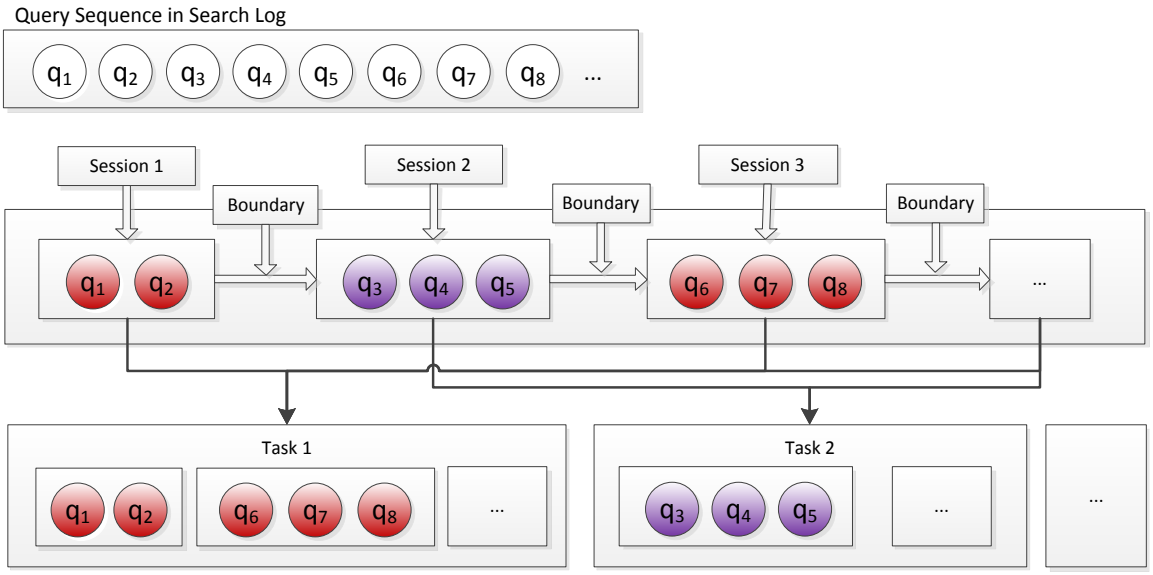


Figure 3.3 Task identification by grouping similar search sessions.

Table 3.1 Sample of Session Segmentation

User_ID	Query	QueryTime	Clicked_URL	Rank
382351	apple warranty	2006-04-24 22:00:21	http://www.superwarehouse.com	6
382351	ipod questions	2006-04-24 22:17:42	http://www.maclink.co.uk	1
382351	dogwood festival	2006-04-29 21:46:30	http://www.fayettevilledogwoodfestival.com	5
382351	myrtle beach map	2006-05-29 22:58:09	http://travel.yahoo.com	3
382351	cherry grove south carolina	2006-05-29 23:03:03	http://www.tripadvisor.com	4
382351	cherry grove south carolina	2006-05-29 23:03:03	http://www.cherrygrovebeachhouses.com	9
382351	body kits for civic	2006-05-30 20:03:12	http://www.modacar.com	2
382351	motley crue jackets	2006-03-01 17:41:26	http://www.motley.com	9
382351	ticketmaster	2006-03-16 14:40:40	http://www.ticketmaster.com	1

Search engine users have various search intentions. Addressing complex information needs usually requires a user to issue a series of queries, spanning a period of time and over multiple search sessions. Moreover, a user may open multiple web browsers and work on several search tasks at the same time. Thus, accurately identifying search tasks in a user's search session is difficult. In this chapter, the user's search activity is examined at the task level based on the session information.

In most of the existing studies (Lucchese et al., 2011; Mei et al., 2009), a search task is one or multiple sessions that corresponds to a distinct information need. The task is extracted based on the segmented session information, as shown in Figure 3.3, which is also used as the unit for extracting user interests. These methods are referred to as over-session based task identification, because the task information is constructed upon the session units. One obvious problem is that it oversimplifies user's search activity by assuming that users only work on the same search task within a short-period of time. Yet people might work on different search tasks at the same time.

To tackle this problem, a fine-grained task identification method, which is also called the cross-session based task identification method, is proposed in this study. As shown in Figure 3.4, search queries within a search session are segmented into sets of queries which are formed to achieve specific search tasks. Each set of queries is called a sub-task. Then, after examining all search sessions of the user, search queries related to a particular search task are identified by grouping similar sub-tasks together.

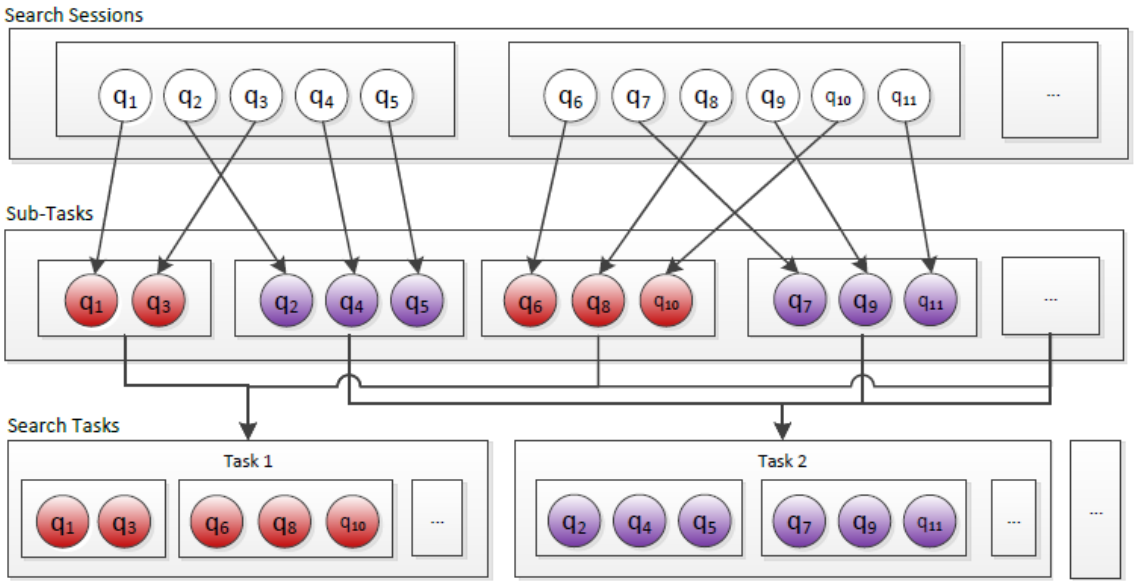


Figure 3.4 Task identification by grouping similar sub-tasks.

Some studies (Ji et al., 2011; Kotov et al., 2011) adopt supervised methods to label search tasks using a pairwise classification method. However, pairwise prediction might not be consistent. For example, two pairs of queries: (query q_i and q_j), (query q_i and q_k) are predicted to be in the same task, while query q_j and q_k are not. Meanwhile, some studies (Lucchese et al. 2011; Luxenburger, Elbassuoni, & Weikum, 2008) use an external dataset such as the Open Directory Project or Wikipedia in the task grouping process. However, there are disadvantages of these approaches. Because the labels and categories are generated from an external dataset, the total number of labels or categories of search tasks are fixed rather than adaptive to the user's dynamic search interests. In fact, it is usually the case that most users have multiple information needs and they are dynamically changing (Widyantoro, Ioerger, & Yen, 1999). Therefore, in TOQUE, an unsupervised method, cross-session based best-link model, is proposed to generate sub-tasks from each search

session from the training set, and a graph-based representation method is introduced for calculating the pairwise similarity of two queries for sub-task grouping automatically.

q_1 = “Apple MacBook” q_2 = “Tesla Car”

q_3 = ”Musk Tesla Patent” q_4 = ”MacBook Pro”

q_5 = ”Open-source Patent” q_6 = ”Battery”

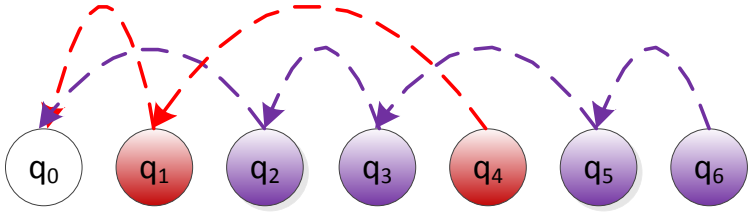


Figure 3.5 Latent task structure identified by best-link model.

The idea of best-link structure is illustrated in Figure 3.5. It can be noticed that the best-link defines a hierarchical tree structure of “strong” connections among the queries: rooted in the fake query q_0, and each sub-tree of q_0 corresponds to one specific search sub-task in a search session. For a new query, it can only belong to a previous search task or be the first query of a new task. Therefore, the temporal order provides a helpful signal to explore the dependency between queries.

As a result, the dependency among the queries belonging to the same sub-task is explicitly encoded by the latent best-link structure: as shown in Figure 3.5, predicting “Tesla Car” and “Musk Tesla Patent”, “Open-source Patent” and “Battery” belonging to the same task would immediately lead to the conclusion that all these four queries are in the same task, even though “Tesla Car” and “Open-source Patent” are not directly connected to each other.

Specifically, given a query sequence Q = {q_1, q_2, … , q_m} within a search session, h is introduced to denote the latent best-link structure. h(q_i , q_j) indicates the existence of a link between q_i and q_j as following:

$$h(q_i,q_j) = \begin{cases} 1, & \varphi(q_i,q_j) > \gamma \\ 0, & otherwise \end{cases} \quad (3.1)$$

where h(q_i , q_j) =1, if query q_i and q_j are directly connected in h; and otherwise, h(q_i, q_j) = 0. To model the first query of a new search task, i.e., the query that does not have a strong connection with any previous queries, a fake query q_0 is added at the beginning of each search session. All the queries connecting to q_0 would be treated as the initial query of a new search task. Besides, it is enforced so that a query can only link to another query in the past, or formally,

$$\sum_{i=0}^{j-1} h(q_i,q_j) = 1, \forall j \geq 1 \quad (3.2)$$

Note that the best-link model is conducted within each search session to generate a list of subtasks, and similar subtasks are grouped together as a search task using the hierarchical clustering method.

3.2.2 Link Prediction

To achieve the latent structure h(q_i, q_j) defined in Equation 3.1, φ(q_i, q_j) should be determined first. The pairwise similarity between relevant feedback documents is adopted for calculating the similarity between two queries. Specifically, the queries resulting in no click action are defined as invalid queries, such as q_3, q_4 and q_6. By contrast, the queries resulting in at least one clicked result are defined as valid queries, such as q_2 and q_5. All invalid queries are ignored in this study as are in one previous study (Bing, Lam, & Wong, 2011). For example, in Figure 3.6, to determine if q_2 and q_5 belong to the same task, two

similarities between the relevant feedback documents of these two queries are calculated, including sim($d_{2,1}$, $d_{5,3}$) and sim($d_{2,1}$, $d_{5,5}$), where $d_{2,1}$ denotes the first retrieved document of q_2, sim() represents the similarity of a pair of queries. Then q_2 and q_5 are segmented into the same task if at least sim($d_{2,1}$, $d_{5,3}$) or sim($d_{2,1}$, $d_{5,5}$) is bigger than the γ as indicated in Equation 3.1.

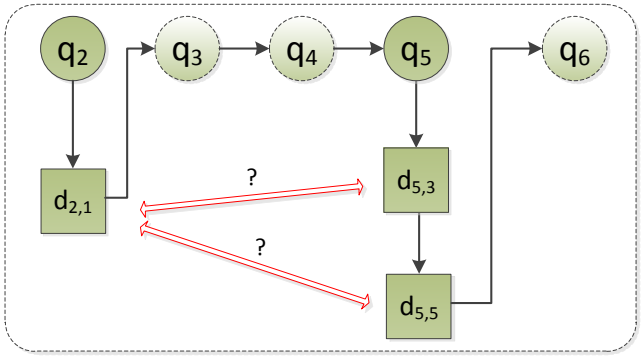


Figure 3.6 Example of the pairwise similarity.

However, there are two problems of calculating the above pairwise similarity using the original page contents, including data noise and data scarcity (Wu et al., 2006). This is due to the fact that many relevant documents contain other non-pertinent information such as advertisements, causing difficulty in summarizing their latent meanings. Futhermore, for a search log dataset, such as AOL, it does not contain snippets, but URLs that might not point to a live site anymore, or of which the content might have been changed after the dataset was created. To tackle this problem, a two-step graph-based representation method is proposed for predicting the pairwise similarity between the relevance feedback documents of two different search queries.

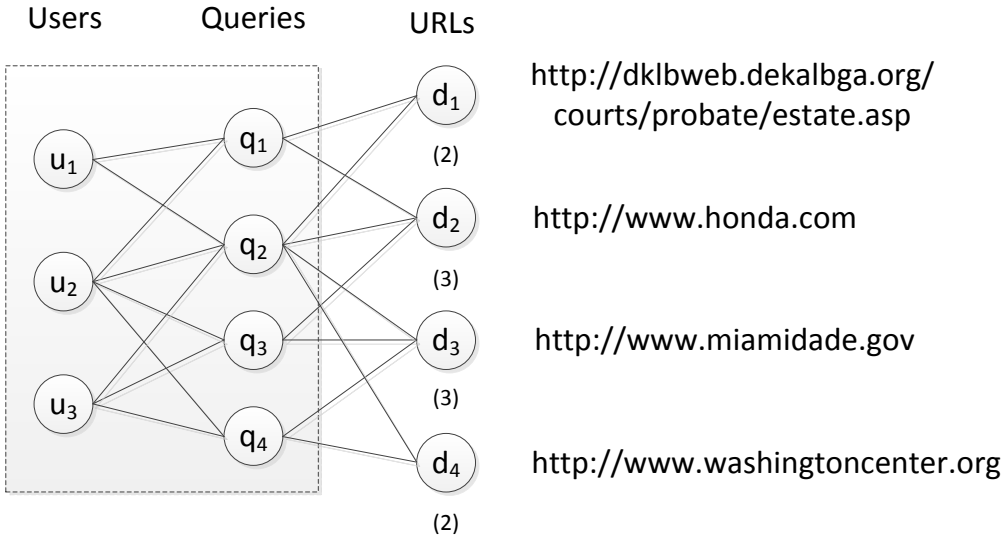


Figure 3.7 Example of the URL connection within AOL log.

First, a click graph is constructed for generating the pseudo-document of each clicked URL. An example of a click graph with four queries and four URLs is shown in Figure 3.7. The edges of the graph capture the relationships between the queries and the URLs. Since different users may use different queries to arrive at a particular web page, it is proposed to generate a pseudo-document for each URL by combining all its connected queries in this graph. For example, two different queries (q_1 and q_2) from two different users (u_1 and u_2) are connected to the same URL. The queries (q_1 and q_2) are then combined to represent the pseudo-content.

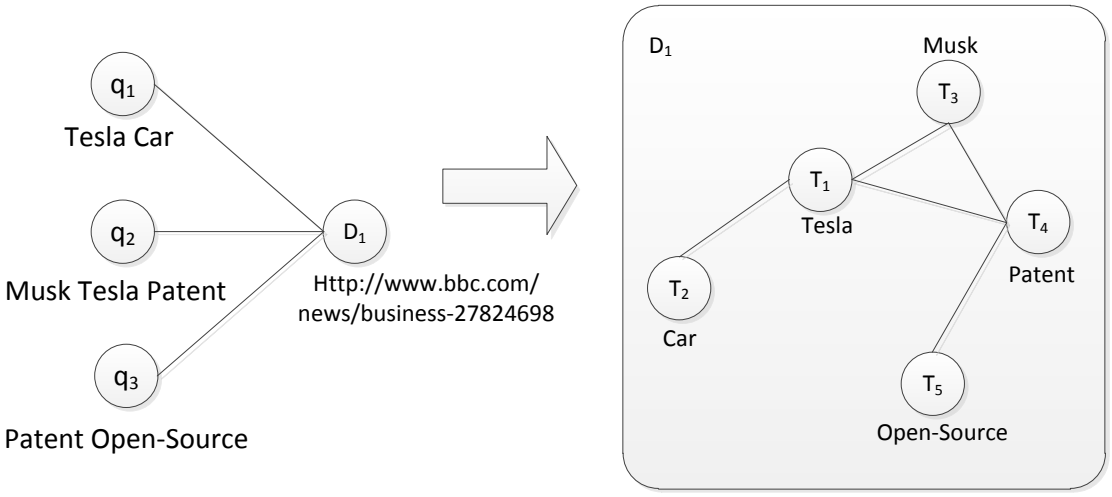


Figure 3.8 Graph-based representation of a relevance feedback document.

Second, simply adopting a bag-of-words to represent the content of a pseudo-document will lose the structural semantic information. To tackle it, a graph-based representation is proposed. Specifically, the unique terms, denoted as $\{T_i\}$, are extracted from the pseudo-document. For example, as shown in Figure 3.8, there are five unique terms within the pseudo-content of D_1, including T_1: "Tesla", T_2: "Car", T_3: "Musk", T_4: "Patent", and T_5: "Open-Source". Afterwards, a pairwise examination is automatically conducted within each query string to determine the existence of a binary non-directional edge between two terms. For example, in Figure 3.8, T_1 and T_2 are connected with an edge because they are in the same query q_1; T_2 and T_3 are not connected because no query in D_1 contains both of them. Then each pseudo-document is represented as a graph G = (N, E), where N denotes the nodes (unique terms) and E denotes the edges. Finally, given two semantic graphs $G_1 = (N_1, E_1)$ and $G_2 = (N_2, E_2)$ constructed for two relevance feedback documents, an existing graph similarity measure is adopted to estimate their semantic relatedness as in (Fan et al., 2010).

Given two graphs G_1 and G_2, p-homomorphism is defined as follows: G_1 is said to be p-homomorphism to G_2 if there exists a mapping δ from N_1 to N_2, such that for each node n in N_1 and m in N_2: (1) if mat(n, m) > θ, then $\delta(n) \rightarrow m$, where mat(n, m) indicates the similarity between node n and m; and (2) for each (n, n') in E_1, there exists a non-empty path (m /.../ m') in G_2 such that δ (n') $\rightarrow$ m' (i.e., each edge from n to n' is mapped to a path emanating from m and ending in m'). The mapping function $\delta(n)$ in the above is referred to as a p-homomorphism mapping from G_1 to G_2.

In practice, the similarity between two graphs G_1 and G_2 is calculated even if they are not 1-1 p-homomorphism to each other. Two frequently used metrics for measuring pairwise graph similarity is the maximum cardinality and overall similarity. In this study, the maximum cardinality algorithm is adopted to measure the pairwise graph similarity because of its high efficiency. Maximum cardinality gives a quantitative measure of the similarity between two graphs, which is in the range of [0, 1], by calculating the number of nodes in G_1 that map to G_2. Let δ be a p-homomorphism mapping from a sub-graph $G'_1 = (N'_1, E'_1)$ of G_1 to G_2. The cardinality of δ is then defined as Card (δ) = $| N'_1|/| N_1|$.

Algorithm 1 shows the procedure used in the proposed method for finding the p-homomorphism mapping meeting the above conditions. The algorithm first constructs a matching list L where for each node n_1 in G_1, $L(n_1)$ collects nodes n_2 in G_2 such that mat(n_1, n_2)> θ. The path information of G_2 is calculated and stored in W. G'_1 and G'_2 represents the sub-graph of G_1 and G_2 whose nodes come from L. The method uses a matrix W to store the path information between nodes in G'_2. For example, $W(n_2, n'_2) = 1$, if there is a path between n_2 and n'_2; $W(n_2, n'_2) = 0$, otherwise. Note that W is an asymmetric matrix because the concept of path here is directional.

ALGORITHM 1: Finding matching sub-graphs with a p-homomorphism mapping

Input: Graph $G_1(N_1, E_1)$ and $G_2(N_2, E_2)$
Output: A mapping δ_m from G_1 to G_2
for *node* $n_1 \in N_1$ *of graph* G_1 **do**
  $L(n_1) = \{n_2 \in N_2, Sim(n_1, n_2) > \theta\}$;
  $H(n_1) = \{n'_1 \in N_1, (n_1, n'_1) \in E_1 > \theta\}$;
end
for *each node pair* (n_2, n'_2) *in* G_2 **do**
  if *Path*(n_2, n'_2) *exists in* G_2 **then**
    $W[n_2][n'_2] = 1$;
  end
  else
    $W[n_2][n'_2] = 0$;
  end
end
$\delta_m = \emptyset$;
while $len(L) > len(\delta_m)$ **do**
  $\delta_m = \delta_m$ + IterativeMatching(G_1, G_2, L, W, H);
end
return δ_m
  procedure ITERATIVEMATCHING
    Input: G_1, G_2, L, W, H
    Output: A mapping δ from G_1 to G_2
    $\delta = \emptyset$;
  for *each node* n_1 *in* L *and a node* n_2 *from* $L(n_1)$ **do**
      $L'(n'_1) = \emptyset$
      /* prune the matching nodes for n_1's neighbors */
      for *each node* n'_1 *in* $\cap L$ **do**
        for *any node* n'_2 *in* $L(n'_1)$ *such that* $W[n_2][n'_2] = 0$ **do**
          $L(n'_1) = L(n'_1) - \{n'_2\}$;
          $L'(n'_1) = L'(n'_1) + \{n'_2\}$;
        end
      end
    end
    δ_1=IterativeMatching(G_1, G_2, L, W, H);
    δ_2=IterativeMatching(G_1, G_2, L', W, H);
    $\delta = \max(\delta_1, \delta_2)$
    return δ
  end procedure

After executing the above procedure, the method adopts a greedy matching method to find the optimal matching p-homomorphism sub-graph between G'_1 and G'_2. For this purpose, the IterativeMatching procedure takes the current list L as its input. It computes a p-homomorphism mapping from G'_1 to G'_2. Specifically, for each pair of n_1 in L and $L(n_1)$, it updates the L list by pruning the neighbors (i.e., the nodes connected to n_1) of n_1 whose connections to n_1 cannot be mapped as a path in G_2 according to W. After the pruning process, the pruned nodes and their mapping nodes in G_2 will not be deleted but rather be stored in a new list L'. Procedure IterativeMatching then iteratively computes the

p-homomorphism mapping δ_1 and δ_2 for L and L' respectively. The method then selects the larger one between δ_1 and δ_2 as its output. At last, the corresponding maximally weighted cardinality of the δ mapping is used as the estimated pairwise graph similarity.

3.3 Experiments

3.3.1 Dataset and Evaluation Methods

Lucchese et al. (2011) develop a Web application that helps human assessors manually identify the optimal set of user tasks from the AOL query log. They produce a ground truth for evaluating any automatic user task discovery method, which is also publicly available at "http://miles.isti.cnr.it/~tolomei/downloads/aol-task-ground-truth.tar.gz". It contains 554 search tasks with average 2.57 queries per task in total. 143 cross-session tasks are contained in this dataset. In this experiment, this dataset was adopted as the ground truth for comparing the performance of the proposed task identification method and baselines.

To evaluate the performance of the proposed task identification method, it is necessary to measure the degree of consistency between the ground truth and search tasks generated by our algorithms. Specifically, both classification- and similarity-oriented measures (Lucchese et al., 2009) are adopted in this experiment. A predicted task indicates the user task where a query was assigned by a specific algorithm, while a true task indicates the user task where the same query was in the ground truth.

Classification-oriented approaches measure how closely predicted tasks match true tasks. F1 is one of the most popular measures in this category, as it combines both precision and recall. In this study, precision measures the fraction of queries that are assigned to a

user task and that are actually part of that user task. Instead, recall measures how many queries are assigned to a user task among all the queries that are really contained in that user task. Globally, F1 evaluates the extent to which a user task contains only and all the queries that are actually part of it. Two notations, $p_{i,j}$ and $r_{i,j}$, are introduced to represent the precision and recall of predicted task i with respect to true task j, then F1 corresponds to the following weighted harmonic mean of $p_{i,j}$ and $r_{i,j}$.

$$F1 = 2 \times p_{i,j} \times r_{i,j} / (p_{i,j} + r_{i,j}) \quad (3.3)$$

Similarity-oriented measures consider pairs of objects instead of single objects. Let T be the sets of predicted tasks of true tasks S. For each true task in S, four values are computed, including: 1) t_n --- number of query pairs that are in different true tasks and in different predicted tasks (true negatives); 2) t_p --- number of query pairs that are in the same true task and in the same predicted tasks (true positives); 3) f_n --- number of query pairs that are in the same true task but in different predicted tasks (false negatives); 4) f_p --- number of query pairs that are in different true tasks but in the same predicted task (false positives). Then, two different measures are adopted as following:

Rand index:

$$R(T) = (t_n + t_p) \ / \ (t_n + f_p + f_n + t_p) \quad (3.4)$$

Jaccard index:

$$J(T) = t_p / (f_p + f_n + t_p) \quad (3.5)$$

3.3.2 Experimental Design

An experiment was conducted to compare the performance of the proposed task identification methods including best-link method (BL) and best-link with graph-based representation method (BL-G). The difference is that BL adopts the bag-of-words method for representing the feature of the pseudo-document, while BL-G uses proposed graph-based representation method for modeling rich semantic features.

Three baselines methods were adopted in this experiment, including one over-session based method and two cross-session based methods. The over-session based method (OS) is proposed by Luxenburger et al. (2008) who adopt a hierarchical clustering method to identify tasks in which the atomic units to be clustered are past sessions. The two best performing cross-session based methods are from the study conducted by Lucchese et al. (2011), i.e., QC_wcc and QC_htc. Specifically, QC_wcc performs clustering by dropping "weak edges" among queries and extracting the connected components as tasks. QC_htc assumes a cluster of queries can be well represented by only the chronologically first query and last query in the cluster; therefore only the similarity of the first and last queries of two clusters is considered in the agglomerative clustering.

The annotated log dataset was randomly split into a training set with 270 annotated search tasks, and a test set with the other 270 annotated tasks. The parameters in each model were tuned by a 5-fold cross-validation on the training set. All baselines and our methods were trained on the same training set.

3.3.3 Experimental Results

Figure 3.9 shows the performance comparisons between proposed methods and baselines.

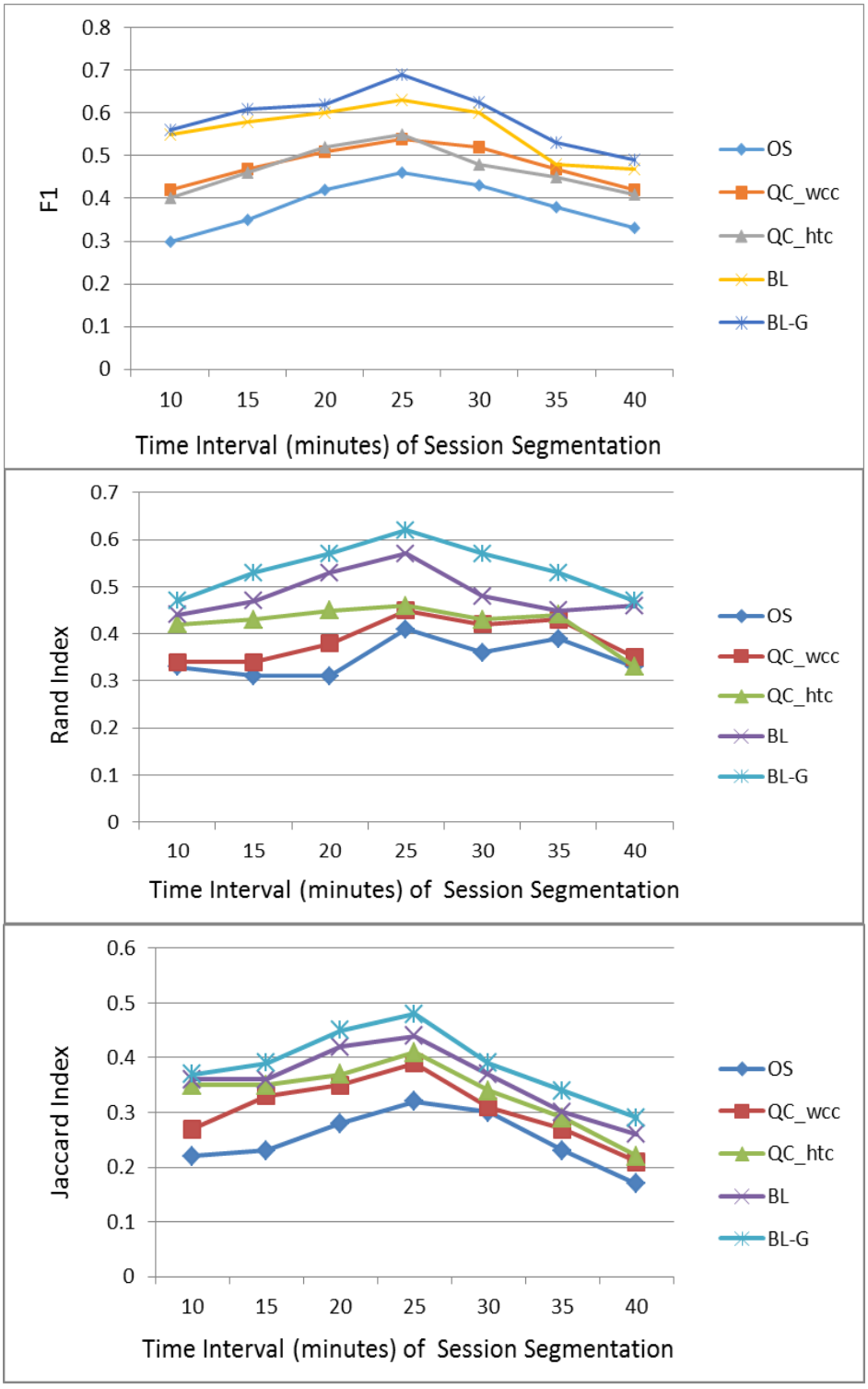


Figure 3.9 Performance comparisons between proposed methods and baselines.

It was first observed that the session boundary does impact the performance of all compared task identification methods. Most of them achieve the highest performance on these three evaluation metrics when the time interval is set at 25 minutes, which is consistent with existing studies (Kotov et al., 2011; Lucchese et al., 2011). The proposed methods BL and BL-G outperformed QC_wcc and QC_htc significantly in all three metrics. The reason is that both QC_wcc and QC_htc target on predicting whether two queries represent the same task. However, the pairwise prediction cannot directly generate the task information and post-processing is required to obtain the tasks. Such a post-processing is independent from the classifier training therefore is not necessarily optimal.

In addition, the over-session based method (OS), performed much worse than the others especially on Rand Index and Jaccard Index metrics. The possible reason is that it assumes that users work on the same task within the entire duration of a search session which results in a high f_p value. Finally, BL-G performs better than BL, because BL-G utilizes the proposed graph-based representation to retain the semantic information.

Table 3.2 Performance Comparisons between Session-based and Non-session based Task Identification Methods

Task Identification Methods		Evaluation Metrics		
		F1	Rand Index	Jaccard Index
Non-session based	BL-NoSS	0.560	0.478	0.422
	BL-G-NoSS	0.603	0.539	0.439
Session-based	BL	0.628	0.571	0.446
	BL-G	**0.695**	**0.619**	**0.483**

So far, the proposed best-link model for task identification is conducted within the entire duration of a search session. One interesting question is whether the session

information is contributive in the proposed best-link method. Table 3.2 illustrates the performance comparisons between the best-link methods within the search session and the ones without using the session data (denoted as BL-NoSS and BL-G-NoSS respectively). Note that both BL and BL-G were optimized by setting session interval at 25 minutes. It was observed that the proposed methods performed much better when using the session data. For example, the F1 scores of BL and BL-G were 0.628 and 0.695, whereas those of BL-NoSS and BL-G-NoSS were 0.560 and 0.603. The major reason for these performance differences is that the session information sets a temporal boundary for identifying the latent link structure of queries from the same search task. This temporal boundary prevents the predicted error made in previous session from affecting the prediction accuracy in the current session. Furthermore, the fact that BL-G and BL-G-NoSS outperformed BL and BL-NoSS respectively, indicates that the proposed graph-based representation for query similarity computation is more effective.

3.4 Summary

Users switch search tasks frequently during their search activities, thus developing methods to extract these tasks from historical data is an important problem. In this chapter, a two-step cross-session based method is presented for extracting search tasks. First, a best-link model is introduced which is capable of learning query connections from the user's search activities. Second, a graph-based representation method is proposed to estimate the contextual pairwise similarity of queries. Then an experiment using a publicly available annotated dataset is conducted to demonstrate the superior performance of our method in identifying search tasks versus a number of state-of-the-art algorithms. The

results are promising and pave the way for further works, including user modeling and task based personalization.

CHAPTER 4

FOUR-TUPLE DESCRIPTOR BASED USER PROFILING

4.1 Introduction

Modeling users' search interests has been a popular research topic. Most of the profiling techniques utilize a descriptor representation to model the general search interests of a user. For example, Downey et al. (2008) adopt a bag-of-words based method to learn the user's interests by adjusting the weights of features, such as the TF-IDF value of keywords. Bennett et al. (2012) introduce another feature descriptor, in which a classifier is used to predict the likelihood that a specific feature, such as a word, is interesting to a particular user. Systems that adopt bag-of-words based representations are effective at learning users' long-term interests, because their prediction accuracy improves substantially using only a small amount of feedback. However, this representation cannot flexibly adapt to the abrupt change of users' short-term interests, because it assumes that user interests change at a constant rate. Although a system could be designed to enhance how it models user's short-term interests by maintaining a fixed number of the most recent feedback (Downey et al., 2008), it would ignore the stable long-term interests easily. This problem indicates a need to develop a representation which can balance the shortcomings and benefits between the long- and short-term interest models. In this chapter, a four-tuple descriptor model is introduced to represent and learn the long- (positive and negative) and short-term (positive and negative) user interests for each task generated from the user's past search histories.

Moreover, the classic form of a user profile is a weighted vector of keywords. Yet the computing cost is high due to the high dimension of a keywords vector. In this chapter, a topic model based user profiling method is introduced. The topic model (e.g., LDA) has been accepted as an effective and efficient approach for text modeling. It is a theoretical model supporting the idea that each document is a mixture of multiple topics, where each topic is a mixture of multiple words. Compared to the vector space model representing a document with terms and weights such as TF-IDF value (Qiu & Cho, 2006), the topic model reduces the computing dimensions enormously and still preserves the essential statistical relationships.

4.2 Methods

4.2.1 Training an LDA Model

In TOQUE, topic models for feature selection are utilized by transforming the document representation from a term vector into a topic vector. In natural language processing, a topic model is a type of statistical model for discovering the abstract "topics" that occur in a collection of documents. LDA is one of the most popular topic models that allow documents to have a mixture of topics (Blei, Ng, & Jordan, 2003). In text processing, LDA model allows sets of documents to be represented by latent topics which consist of different terms that are semantically similar. In this research, whether the topics discovered by LDA are useful for modeling user's interests is explored.

Using LDA, the topic distribution of a document along with the probability that the document belongs to each of the discovered topics can be derived. Then a document can be represented as a topic vector by using each of the LDA discovered topics as a feature and

the probability as the corresponding feature weight. Once the dataset is pre-processed, LDA is used to cluster the documents into topic groups. After topic extraction, a document d_i in our data set will be represented as a topic vector V_i:

$$V_i = (p(t_1 \mid d_i), \ldots p(t_j \mid d_i), \ldots p(t_k \mid d_i)) \quad (4.1)$$

where k is the total number of the topics, and $p(t_j \mid d_i)$ denotes the probability that document d_i is assigned to topic t_j by LDA.

MALLET, a Java-based package for statistical NLP, is used to carry out the topic modeling.

4.2.2 Representing User's Task-based Interests

The classic form of a user profile is a weighted term vector. Yet the computing cost is high due to the high dimension of the term vector space. In TOQUE, building user profiles based on the topic model, which reduces the computing dimension from term space to topic space is proposed. Specifically, a task-based user profiling method is introduced as a technique of constructing user profile through modeling user's long- and short-term search interests for each search task.

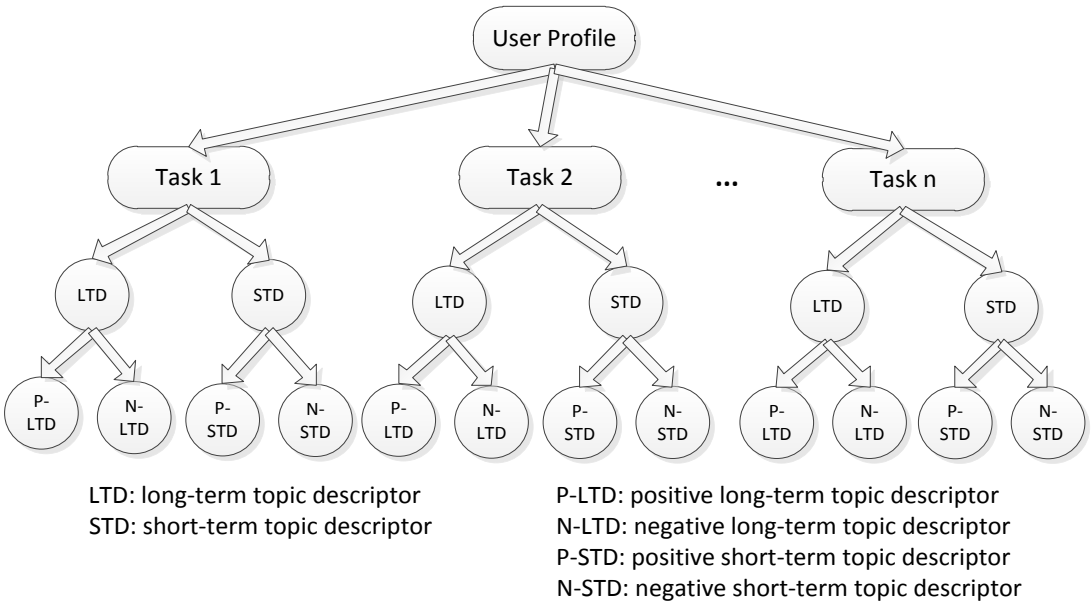


Figure 4.1 Representation of the task-based user profiling.

Figure 4.1 provides the representation of the task-based user profile that is proposed in this study. In a search task, user's interests are modeled by a four-tuple topic descriptor (TD). Each TD is represented by two descriptors, i.e., long-term topic descriptor (LTD) and short-term topic descriptor (STD). The long-term user interests are modeled by two descriptors, i.e., positive long-term topic descriptor (P-LTD) and negative long-term topic descriptor (N-LTD). Similarly, short-term user interests are modeled by positive short-term topic descriptor (P-STD) and negative short-term topic descriptor (N-STD). Then, user's interests can be represented by the following two-level descriptor:

$$TD = < LTD<P\text{-}LTD, N\text{-}LTD>, STD<P\text{-}STD, N\text{-}STD>> \quad (4.2)$$

This representation aims at preserving the feature vectors of relevant and non-relevant documents, thus enabling separate measurements of similarities between topics of the user's positive and negative interests. The degree of a user's interest in a candidate query is computed by subtracting the user interest values in negative descriptors from the one in positive descriptors. The relevance feedback within a search task is used to

model the user's long-term interests, whereas the user's current search session is used to model the user's short-term interests.

Rocchio algorithm is the most well-known relevance feedback algorithm and is widely used in information retrieval. Equation 4.3 represents the general form of the query refinement using Rocchio algorithm (Qiu & Cho, 2006).

$$Q_{i+1} = Q_i + a\sum_{pos} D_{pos} / n_{pos} - (1- a)\sum_{neg} D_{neg} / n_{neg} \quad (4.3)$$

where Q_i indicates original user's interest in the query Q, Q_{i+1} indicates the updated user's interest in the query Q, D_{pos} denotes a relevant document for Q_i, D_{neg} denotes an irrelevant document for Q_i, n_{pos} is the number of relevant documents, and n_{neg} is the number of irrelevant documents.

In this study, Rocchio algorithm is adopted to learn the user relevance feedback in a four-tuple descriptor model. For example, P-LTD and N-LTD are updated by the following equations:

$$P\text{-}LTD_{new} = P\text{-}LTD_{old} + D_{pos} - D_{neg} \quad (4.4)$$

$$N\text{-}LTD_{new} = N\text{-}LTD_{old} + D_{pos} - D_{neg} \quad (4.5)$$

where D_{pos} is the positive relevance feedback, and D_{neg} is the negative relevance feedback. The user's long-term interest in a query Q is represented by ILTD(Q), which is expressed as follows:

$$ILTD(Q) = \alpha\, SIM(Q, P\text{-}LTD) - (1- \alpha)\, SIM(Q, N\text{-}LTD) \quad (4.6)$$

where $\alpha \in (0, 1)$, α is the weight of the positive long-term interest, and $(1- \alpha)$ is the weight of the negative long-term interest. SIM(Q, P-LTD) represents the similarity between the query Q and P-LTD. Similarly, P-STD and N-STD are updated by the following equations:

$$P\text{-}STD_{new} = P\text{-}STD_{old} + D_{pos} - D_{neg} \quad (4.7)$$

$$\text{N-STD}_{new} = \text{N-STD}_{old} + D_{pos} - D_{neg} \quad (4.8)$$

The short-term user interest in a query Q is represented by ISTD(Q), which is expressed as follows:

$$\text{ISTD}(Q) = \beta\, \text{SIM}(Q, \text{P-STD}) - (1-\beta)\, \text{SIM}(Q, \text{N-STD}) \quad (4.9)$$

where $\beta \in (0, 1)$, β is the positive short-term interest weight, and $(1-\beta)$ is the weight of the negative short-term interest. The final interest in a query Q is given by

$$\text{ITD}(Q) = \gamma\, \text{ILTD}(Q) + (1-\gamma)\, \text{ISTD}(Q) \quad (4.10)$$

where $\gamma \in (0, 1)$, γ is the long-term interest weight, and $(1-\gamma)$ is the short-term interest weight.

The relevance feedback within a search task is used to model user's long-term interests, which is calculated by Equation 4.4 and 4.5. Similarly, the relevance feedback of the current search session is used to model user's short-term interests, which is calculated by Equation 4.7 and 4.8. Note that all P-LTD, N-LTD, P-STD, and N-STD are represented through a topic distribution as indicated in Equation 4.1.

4.3 Experiments

4.3.1 Dataset

The dataset adopted in the study is the AOL log, which is publicly available at "http://www.infochimps.com/datasets/aol-search-data". The AOL dataset is adopted because it is the latest accessible public data set on the internet. It is a query log from a standard search engine (AOL.com) and widely used in the web research related studies. 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. The dataset contains a large amount of noise, such as typographical errors. Raw data preprocessing was conducted similar to that described in (Bing, Lam, & Wong, 2011). First, host navigation queries, such as "www.msn.com" and "www.bbc.com", were removed. Second, queries with non-alphabetical characters were removed as well. Third, stop words were removed from the queries. After duplication removal and data cleaning, it resulted in 642,371 unique users, 4,224,165 unique queries, and 1,343,302 unique clicked URLs in total.

Note that the session information can be obtained by a temporal method introduced in (Bing, Lam, & Wong, 2011), where the time interval between queries within a session was less than 25 minutes. Every two consecutive queries within the same session should share at least one term. The users who have less than 100 sessions in the whole AOL dataset were removed. After session division, the dataset was split 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. Task identification was conducted as introduced in Section 3.2. Pseudo-documents (Ji et al., 2011) were constructed for each URL contained in the training and test sets. These pseudo-documents were used to represent the content of each clicked URL in the AOL dataset.

The second dataset is a subset of the Reuters-21578 1.0 test collection (available at "https://archive.ics.uci.edu/ml/machine-learning-databases/reuters21578-mld/"). The original dataset consists of 135 topics and 21,578 stories obtained from the Reuters newswire in 1987. Out of these stories, 12,902 stories are divided into one or multiple categories, which are divided as 9,603 stories for the training set and 3,299 stories for the test set according to ModApte split. Because the experiment is designed to evaluate the

model's adaptability to changing topics, five categories were randomly picked up which exist in both the training and test set, and each category contains at least 100 stories. The five categories are TRADE, CRUDE, SUGAR, COFFEE, and ACQ.

4.3.2 Parameter Selection

Before evaluating the performance of proposed method, it is needed to examine the performance of the proposed four-tuple descriptor model by tuning three important parameters: interest impact weight (i.e., α, β, γ) of P-LTD, P-STD, and LTD. The effectiveness of the user-profiling method was measured by evaluating the performance of the method on a learning activity. A learning activity was used to simulate changes in a user interest among tasks. For simplicity, ">>" was used to represent the task transition within the learning activity. For example, if the user had an initial interest on the task of buying a laptop (labeled as T1, short for Task One) and then shifted this interest to the task of finding a Spanish restaurant (labeled as T2, short for Task Two), then the interest change can be described as [T1] >> [!T1, T2], which represented two phases of interest learning. "!T1" indicates unlearning user's interest in task T1. In this case, changing the interest consisted of learning a new interest in T2 and unlearning an old interest in T1. In this experiment, an activity was designed to simulate changes of user's interests from one task to another, which is described as follows:

Learning Activity 1: [T1] >> [!T1, T2] >> [!T2, T3] >> [!T3, T4] >> [!T4, T5].

The proposed user-profiling model can be measured by cycles of evaluations. Each cycle involves 1) learning relevance feedback from the clicked documents during the query sequence of a session and 2) measuring the accuracy by analyzing the user's interests at the

end of each session. Each learning phrase, such as [T1], consisted of 10 cycles of interest learning and accuracy measurement since 10 sessions in sequence were randomly selected for each task from the same user. Note that the queries with at least one clicked document were used for evaluation, and the clicked URLs of these queries were used as the relevance feedback.

To identify the parameters, five users with more than 50 sessions were randomly chosen in our AOL training set. For each task, the information on the first 10 sessions was used to learn user's interests, while the information on the next 30 sessions was used to evaluate the performance of the proposed four-tuple descriptor model. The accuracy is defined as:

$$\text{Accuracy} = \frac{\text{numbers of interested documents}}{30} \tag{4.11}$$

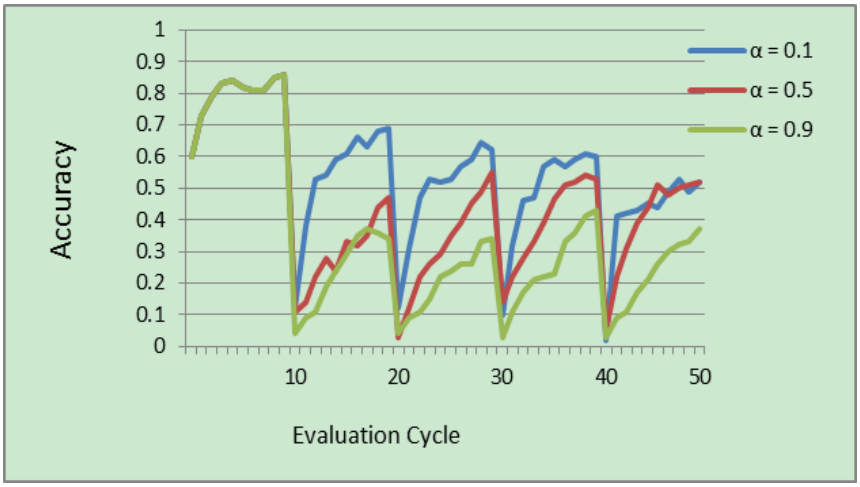


Figure 4.2 Performance of LTD in the learning activity 1.

Figure 4.2 presents the performance of LTD in the learning activity 1 with various values of α (0.1, 0.5, and 0.9). By varying the interest impact weight α, it is concluded that LTD achieved the highest average accuracy when α was set at 0.1. As shown in the figure,

the accuracy of LTD in matching user's interests increases steadily within each task (10 cycles of evaluation) which is caused by the accumulation of learned interests of the user. However, the LTD model suffers a sharp decrease of accuracy at each task transition. Although the model can learn a user's interests, the model is incapable of unlearning the old interests quickly when the user shifts to a new task. This outcome also results in the decrease in accuracy from one phase to another. For example, in the first learning phase, the accuracy is stable at around 0.81, whereas the accuracy drops to 0.68 during the second phase and to 0.52 during the fifth phase.

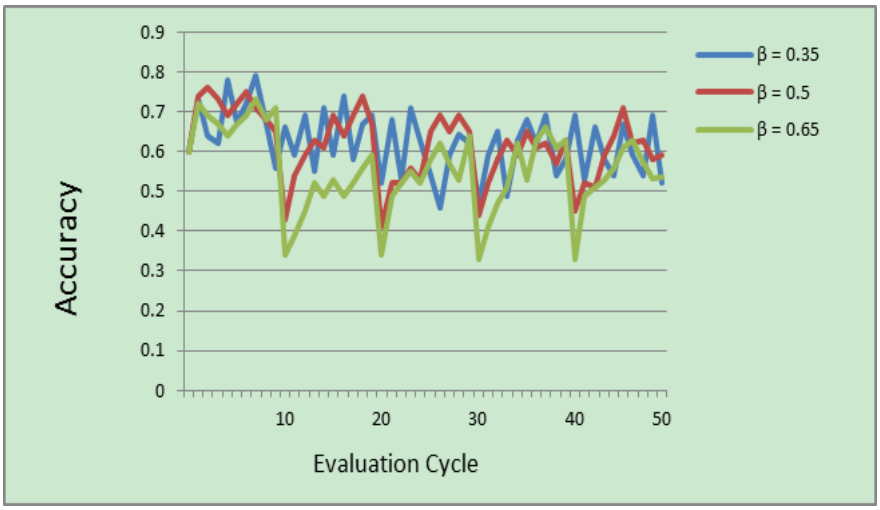


Figure 4.3 Performance of STD in the learning activity 1.

The ability of the model to learn short-term user interests was examined within a session boundary. For a specific session, the relevant pseudo-documents of the clicked URLs were used to simulate the short-term user interests. The initial short-term interest vector was set to the zero vector and updated with all relevant pseudo-documents of the clicked URLs' queries within the same session. The KL divergence was computed between each pair of the short-term user interest vector and each of the pseudo-documents in the

corpus. The only difference between learning long- and short-term interest is that the short-term interest is learned within a session instead of across sessions.

Figure 4.3 shows the performance of STD on the test learning activity, with varying values of β (0.35, 0.5, and 0.65). Given that the STD does not have a memory of former session interests, its accuracy in matching user's interests fluctuates greatly compared with the performance of LTD in Figure 4.2. STD does not learn the user interest as stably as LTD does. However, STD exhibits stable accuracy during task transitions, particularly when β is 0.35. This result indicates that STD possesses better adaptability to interest changes. By varying the learning rate β, it is found that the highest average accuracy of the STD model is obtained when β is set at 0.35.

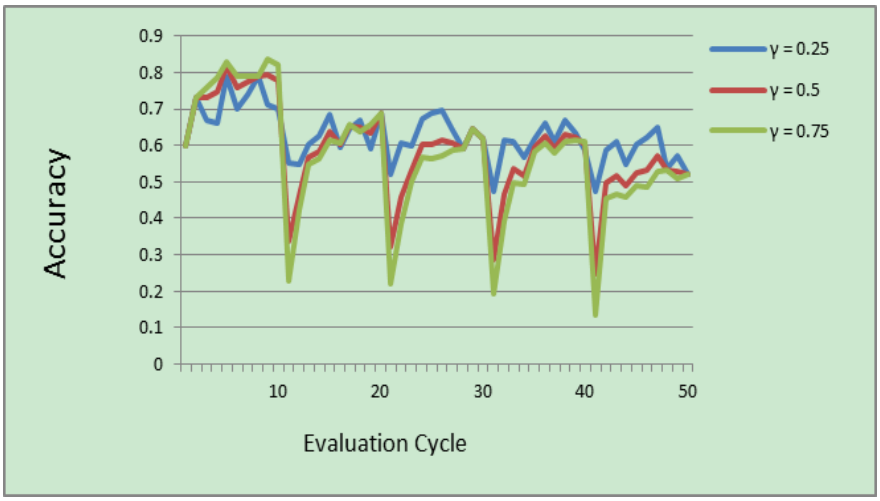


Figure 4.4 Performance of TD in the learning activity 1.

The performance of TD was examined based on the above evaluations of LTD and STD by setting the parameters of α as 0.1 and β as 0.35, thus maximizing the learning ability and prediction of user interests. The interest weight γ was used to control the effect of LTD and STD in the TD. As shown in Figure 4.4, the system performance obtained is stable and has adaptive accuracy when the parameter γ is set to 0.25. TD outperforms LTD

in unlearning older interests, and is superior than STD in matching user interest. Thus, TD overcomes the weaknesses of both LTD and STD.

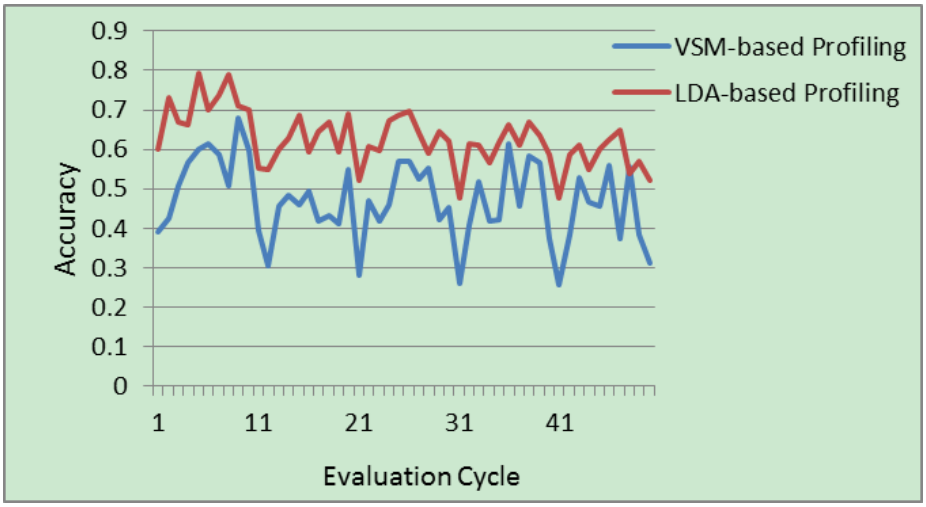


Figure 4.5 Performance comparison between LDA- and VSM-based profiling in learning activity 1.

All above experiments were conducted using LDA for feature representation. One question would be whether the LDA-based user profiling method is more effective than the baseline, VSM-based one. Figure 4.5 shows the performance comparison between the LDA-based profiling method and the VSM-based one. Both methods were set with optimum parameters. Specifically, the LDA-based method was set with parameters of α as 0.1, β as 0.35, and γ as 0.25, and the VSM-based one was set with parameters of α as 0.15, β as 0.38, and γ as 0.21. As shown, the LDA-based profiling method outperformed the VSM-based one significantly.

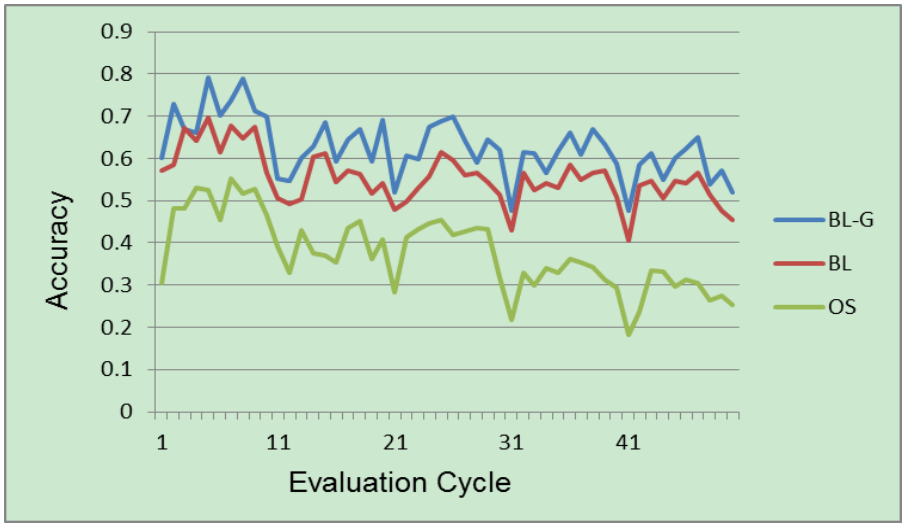


Figure 4.6 Performance comparisons between user profiling methods using various task identification methods (i.e., BL-G, BL, and OS).

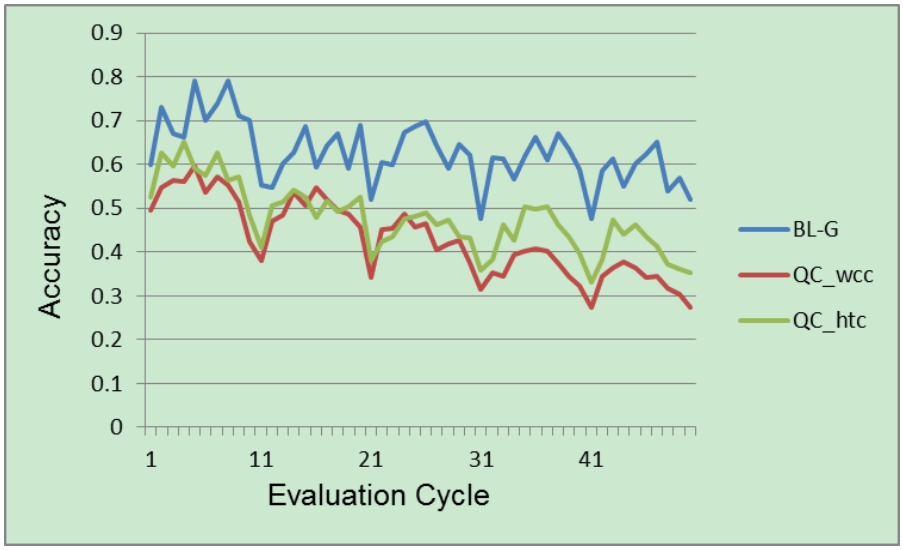


Figure 4.7 Performance comparisons between user profiling methods using various task identification methods (i.e., BL-G, QC_wcc, and QC_htc).

Note that the performance of the proposed user profiling method was evaluated using the task information extracted by the proposed best-link task identification algorithm introduced in Chapter 3. It is also necessary to compare the user profiling methods using different task identification methods. Specifically, the impacts of the proposed BL-G and

BL were compared with other three baselines, including OS, QC_wcc, and QC_htc. In Figure 4.6, the performance of the proposed four-descriptor user profiling method using the proposed task identification methods (i.e., BL-G and BL) perform much better than the one of using the OS method. The reason is that the OS method assumes all queries within a search session are serving for the same search task. Therefore, the relevance feedback of queries from other tasks is incorporated in the user's search interests for the current search task, which results in a lower accuracy of task identification. This observation reinforces our assumption that user's search interests of other search tasks might not be helpful to predict user's current search activity. Besides, it is noticed that even though QC_wcc and QC_htc performed as well as the proposed method, BL, at the beginning of each learning phrase, their performance dropped down as the evolution cycle increased. The possible reason is that both QC_wcc and QC_htc accumulate the predictive error as the search task grows. By contrast, the proposed methods pretend the task identification error from spanning across sub-tasks since the proposed best-link model is conducted within the scope of a search session.

4.3.3 Experimental Design

So far, the pseudo-documents, instead of the original URL contents, were used as user's relevant feedback. Another question is whether the proposed LDA-based profiling method is more effective than the VSM-based one if the original document contents were adopted. Considering that the AOL dataset doesn't provide the document contents, an alternative, the Reuters dataset, was adopted to compare the performance between the LDA- and VSM-based methods on learning and updating user's long- and short-term interests. Reuters dataset was adopted because, in this dataset, each document is assigned with an

explicit topic label (e.g., TRADE, CRUDE, SUGAR, COFFEE, ACQ, etc.), which are widely used for evaluating the performance of learning models.

In this experiment, another two learning activities are designed to measure the effectiveness of both methods. The difficulty level of a learning activity is determined by the number of topics that must be learned at a time, and the changing degree of interests (number of categories) that occurs between two learning phases. Learning a larger number of interest categories at a single time and adapting to a significant change of topics of interest are considered more difficult learning problems. The following are descriptions of these two learning activities used in the experiments.

Learning Activity 2: {TRADE} >> {!TRADE, COFFEE} >> {!COFFEE, CRUDE} >> {!CRUDE, SUGAR}

Learning Activity 3: {TRADE, COFFEE} >> {!TRADE, COFFEE, CRUDE} >> {!COFFEE, CRUDE, SUGAR}

The proposed user-profiling model is measured by cycles of evaluations. At each cycle of evaluation, all the clicked documents were ranked according to their similarity values with the current user interests by using the KL-divergence algorithm. The top 10 ranked documents were examined using the precision@10, which is defined as follows:

$$\text{precision @ 10} = \frac{\text{numbers of interested documents}}{10} \tag{4.12}$$

4.3.4 Experimental Results

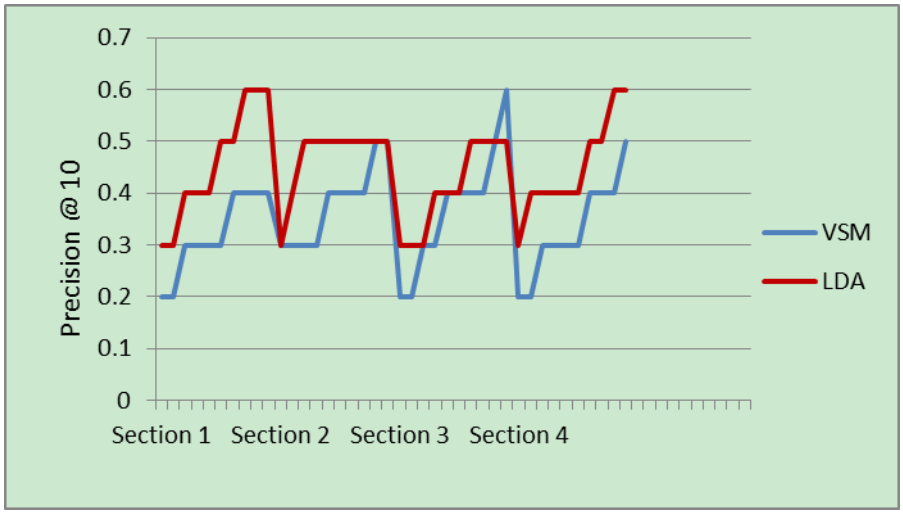


Figure 4.8 Performance comparison between LDA- and VSM-based profiling in learning activity 2.

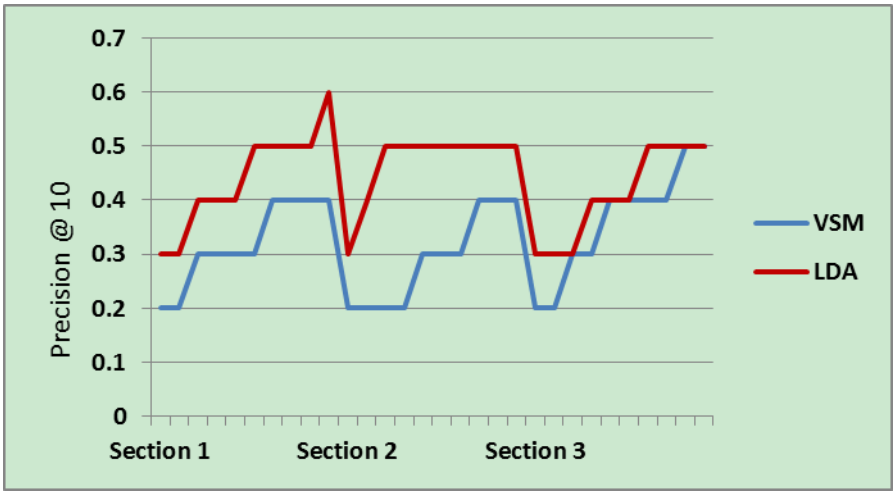


Figure 4.9 Performance comparison between LDA- and VSM-based profiling in learning activity 3.

The performance of TD was examined by setting the parameters of α as 0.1, β as 0.35 and γ as 0.25, for maximizing performance of the proposed model on learning user's interests. As shown in Figure 4.8, the performance of the LDA-based user profiling method outperformed the VSM-based user profiling significantly. Also, within each learning

phase, the performance of LDA-based profiling method increase gradually. By contrast, the VSM-based profiling method achieves a low performance which grows slowly during a learning phase. In other words, the LDA-based profiling method is capable of learning new interests and unlearning old interests more quickly than the baseline. This outcome also results in the increase of P@10 value from one phase to another. For example, in the second learning phase of the learning activity 2, the P@10 value of the LDA-based method is stable at around 0.5 while the baseline method achieves comparable performance at the very end of this learning phase. In the second learning phase of the learning activity 3, as shown in Figure 4.9, the P@10 of the LDA-based method achieves 0.5 at the beginning of the learning phase, while the one of the VSM-based method achieves 0.4 at the very end of this phase. Thus, LDA-based methods match the user's interests better during topic transitions.

4.4 Summary

In this chapter, a four-tuple descriptor based user profiling method was introduced which is adapted to learn and update dynamic user interests. This adaptability is achieved by modeling the user's long- and short-term interests using a four-tuple topic descriptor. Specifically, the LTD learns the user's long-term interests gradually, while the STD captures the abrupt change of the user's short-term interests. Both LTD and STD learn via both the positive and negative descriptors, controlling the interaction between positive and negative interests implicitly. The parameter selection process reveals that: 1) the LTD learns a user's general preferences effectively, which are formed gradually over the long run; 2) the STD adapts to user's abrupt interest change more effectively than LTD, but it is

unstable by nature; 3) the TD incorporates the advantages of both LTD and STD, which are learning the user's interests stably and adapting to the user's interest change effectively. Moreover, LDA is adopted, instead of bag-of-words based method (e.g., VSM), to learn the user's interests. The experimental results show that the LDA-based user profiling method outperforms VSM-based one on both AOL and Reuters datasets.

CHAPTER 5

PERSONALIZATION OF QUERY REFINEMENT

5.1 Introduction

Current studies (Guo et al., 2008; Wang & Zhai, 2008) show that many queries from users are short, which might not sufficient to represent users' search intentions. Therefore, it is significant to help users improve their original queries to represent their information needs better. There are studies aimed at helping users build more effective queries. Among them, query refinement is a process of generating a candidate query list based on each user's original query. The goal of query refinement is to reformulate ill-formed search queries to enhance the relevance of search results.

However, one common issue of traditional query refinement methods is that they fail to consider users' diverse search preferences. In this chapter, to tackle this problem, a two-step personalization method is introduced to utilizing user's task information to improve the effectiveness of candidate queries. First, a graphical model is presented to access the latent task dependency of terms in a query by exploiting a latent task space. Second, users' interests extracted from search logs are applied on personalization of query refinement by re-ranking the candidate query list. The objective of this personalization method is to satisfy the user's information needs faster by providing more effective candidate queries of query refinement for each individual. These queries are generated according to both the user's original search queries and the user's search interests. Therefore, the newly generated candidate query list will result in more relevant search results and user's information needs can be satisfied faster.

5.2 Methods

5.2.1 Candidate Query Terms Generation

Word co-occurrence has been well studied in query suggestion and query refinement research. The widely used co-occurrence based method, i.e., mutual information, is adopted to calculate the most likely candidate words for an original word, based on the assumption that different users may use words with similar meanings to describe the same resource. Specifically, for any two words, w_1 and w_2, MI can be computed using the following equation (Wang & Lochovsky, 2004):

$$I(w_1, w_2) = \sum_{T_{w_1}, T_{w_2} \in \{0,1\}} P(T_{w_1}, T_{w_2}) \log \frac{P(T_{w_1}, T_{w_2})}{P(T_{w_1}) P(T_{w_2})} \qquad (5.1)$$

where T_w is a binary random variable indicating whether the word w appears in a particular query set. For example, $P(T_{w_1} = 1, T_{w_2} = 1)$ is the proportion of the query sets which contain both w_1 and w_2.

5.2.2 Rescoring Candidate Queries using Task Information

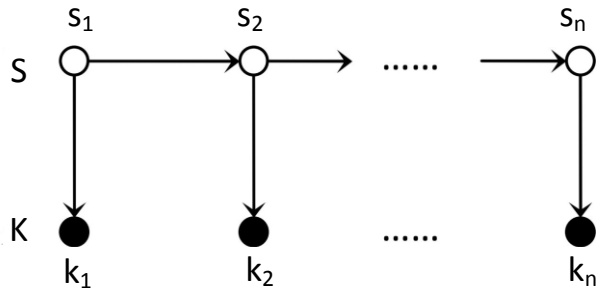


Figure 5.1 Latent task model for a candidate query.

A candidate query for query refinement is a sequence of keyword denoted as K: k_1, k_2, …, k_n, where n represents the position of the keyword within the query after stop words are removed. The latent task of k_1 is denoted as s_1, which is a task from the full task set S. Such a generative process is represented using a graphical model which is shown in Figure 5.1. The latent search tasks are unobservable and represented by empty nodes. The joint distribution of the term sequence denoted as $P(k_{1:n})$ is computed for scoring the candidate query. Let $s_{1:n}$ be the task sequence, the candidate query score can be computed as

$$P(k_{1:n}) = \sum_{s_{1:n}} P(k_{1:n}, s_{1:n}) \tag{5.2}$$

According to the dependency structure as shown in Figure 5.1, the marginal distribution of task sequence and keyword sequence can be computed as

$$P(k_{1:n}, s_{1:n}) = \prod_{w=1}^{n} P(k_w|s_w)P(s_1) \prod_{w=2}^{n} P(s_w|s_{w-1}) \tag{5.3}$$

where $P(k_w \mid s_w)$ denotes the probability that keyword k_W is generated by task s_w, and $p(s_w \mid s_{w-1})$ denotes the relationship between two search tasks. Such a relationship enables a means of governing the task context of neighboring keywords in a query.

The parameter $P(k_w \mid s_w)$ can be easily obtained via Equation 4.1 which is indicated in Section 4.2. As for the second parameter, $p(s_w \mid s_{w-1})$, the pairwise dependent probability is calculated as that task s_{w-1} is followed by s_w. Recall that the objective of query refinement is to provide more relevant candidate query in which the latent search task for each keyword should be consistent, because user's search intention is unique for each search query. To achieve it, the semantic similarity between each pair of search tasks is calculated as shown in Equation 5.4. That is, the probability is high if the two latent search tasks are similar, and vice-versa.

$$P(s_j|s_i) = \frac{sim(s_i, s_j)}{\sum_{s_o \in S} sim(s_i, s_o)} \tag{5.4}$$

where sim(s_i , s_j) is a similarity measure between task s_i and s_j. Specifically, the cosine similarity is adopted to calculate such a similarity as shown following.

$$sim(s_i, s_j) = \frac{\sum_{k_l} P(k_l|s_i)P(k_l|s_j)}{\sqrt{\sum_{k_l} P(k_l|s_i)^2}\sqrt{\sum_{k_l} P(k_l|s_j)^2}} \tag{5.5}$$

5.2.3 Assigning a Query to an Existing Task

This study proposes a personalization process for applying user's task-based search interests in query refinement, which is illustrated in Figure 5.2. Specifically, for a new search query, a candidate query list (i.e., l_1) is generated through an existing query refinement algorithm such as MI (mutual information) or CMI (context-based mutual information). The top 50 candidate queries are grouped into m categories using a hierarchical clustering algorithm (Lee, Liu, & Cho, 2005). Then top K tasks are retrieved by calculating the pairwise KL-divergence value between the current search task and each preexisting one. The category c_i is compared with a preexisting task s_j to determine whether it belongs to a historical search task by calculating the Kullback-Leibler (KL) divergence (Bigi, 2003), as shown in Equation 5.6.

$$KL(c_i||s_j) = \sum_t P(t|c_i)\ log\, P(t|c_i)/P(t|s_j) \tag{5.6}$$

where t represents the topics of the trained LDA model.

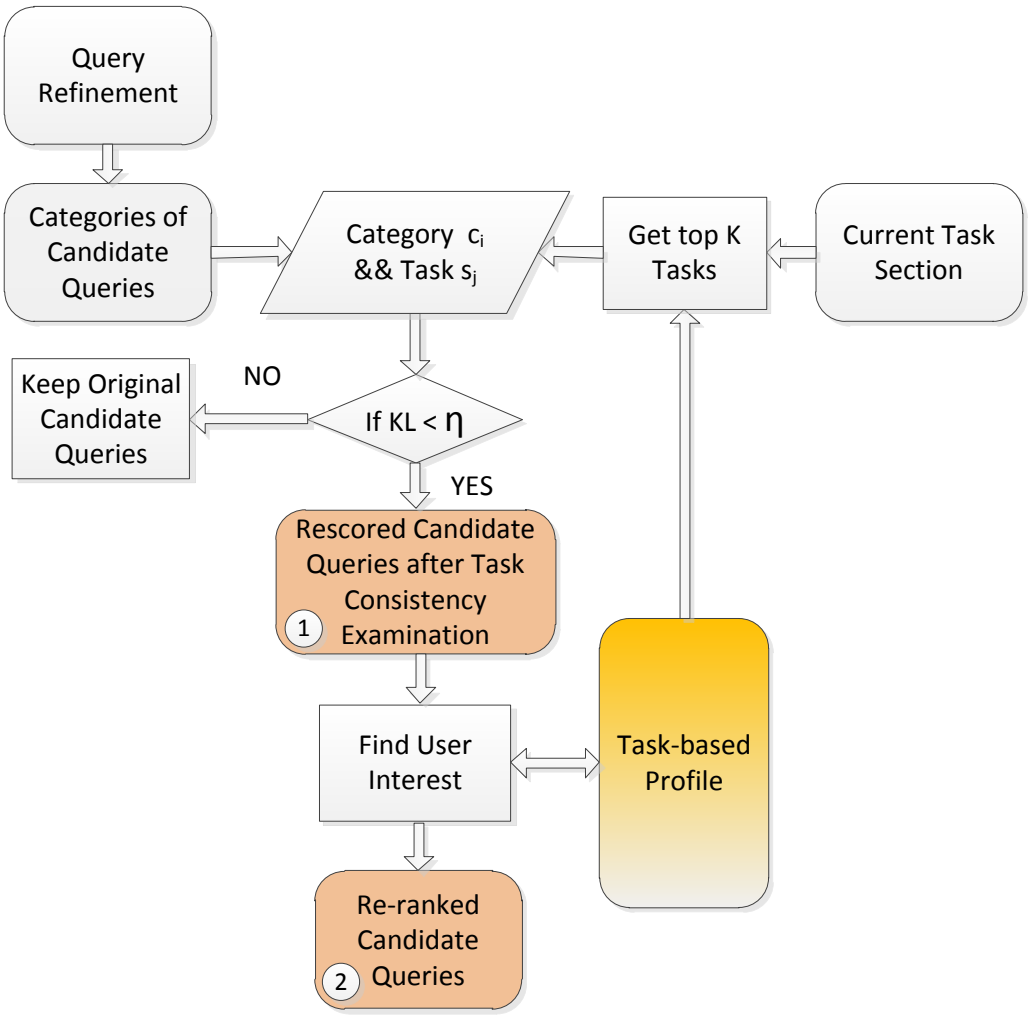


Figure 5.2: Personalization algorithm.

The current search activity is not assigned to an existing task, if the KL divergence between them is above the threshold η (the value of η is set at 0.12 regarding an empirical study). In this case, no personalization will be applied to the user's current search activity. Otherwise, a re-ranking of candidate queries is conducted based on the user profile. First, a latent task consistency score is calculated using the Equation 5.3, resulting in a rank list l_2. Second, a personalized score is computed for each candidate query using the personalization algorithm; subsequently, a new rank list l_3 is generated with respect to each user, sorted by descending personalized scores. Finally, the three ranks l_1, l_2, and l_3 are

merged using Borda's ranking fusion method (Dwork et al., 2001) and the candidate queries are sorted with the merged ranks.

5.2.4 Extracting User's Relevance Feedback

Wang & Zhai (2008) have proven that query reformulation activities from the search log is a good resource for extracting user's preferences. The queries issued later in a specific session are considered more important, compared to those issued earlier in the session. This hypothesis is that, after seeing the search results from earlier queries, users would either 1) revise queries to better characterize their information needs or 2) stop, if they are pleased with the search results. Based on this assumption, two implicit relevance feedback extraction methods, as shown in Figure 5.3, are proposed by distinguishing two different query reformulation behaviors, i.e. adding-word and removing-word behaviors (Huang & Efthimiadis, 2009). An adding-word behavior occurs when a new query is constructed by adding one word to its previous query. A removing-word behavior occurs when a new query is constructed by removing one word from its previous query.

The two proposed methods of extracting relevance feedback are defined as follows:

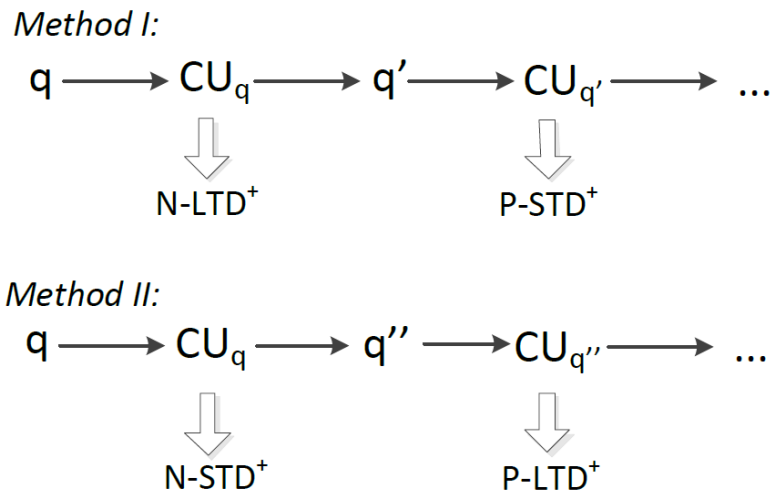


Figure 5.3 Methods of extracting Relevance Feedback.

Method I *(an approach to extract Relevance Feedback when Adding-word Behavior is observed): Assuming that the current query q' is resulted from an adding-word reformulation of its previous query q, then* CU_q *(the clicked URLs of q) are negative feedbacks on long-term user interests (updated in N-LTD), whereas* $CU_{q'}$ *are positive feedbacks on short-term user interests (updated in P-STD).*

In an adding-word reformulation, the latter query contains some words that do not exist in the former query. It is inferred that the previous query q is too general to represent the user's current information need. Therefore it should be beneficial to use the clicked documents of q as negative relevance feedback of long-term interests. Moreover, since the current query q' is the user's most recent query and is more specific than the previous query q, it should be used as the positive relevance feedback of short-term interests.

Method II *(an approach to extract Relevance Feedback when Removing-word Behavior is observed): Assuming that the current query q'' is a removing-word reformulation of its previous query q, then* CU_q *are negative feedbacks on*

short-term user interests (updated in N-STD), whereas $CU_{q''}$ are positive feedbacks on long-term user interests (updated in P-LTD).

In a removing-word reformulation, some words in the former query are removed from the later one. It is inferred that the previous query q is too specific to represent the user's current preference. Therefore the click documents of q should be used as negative relevance feedback of short-term interests. Moreover, since the current query q'' is the user's most recent query and is more general than the previous query, it should be used as the positive relevance feedback of long-term interests.

5.3 Experiments

5.3.1 Evaluation Methods

In this experiment, the AOL dataset is adopted. The data preprocessing process was conducted as introduced in Section 4.3.1. Our evaluation followed an existing study (Bing, Lam, & Wong, 2011) by utilizing the session information of query logs. In a search session, when a user feels unsatisfied with the results of the current query, he may refine the query and conduct a new search. When the user obtains satisfactory search results, he or she may stop searching and start a new search activity. Downey et al. (2008) have discussed the importance of the terminal URL. Therefore, based on this observation, a reliable evaluation can be conducted using the terminal URL information. The definitions of two kinds of queries, as mentioned in (Bing, Lam, & Wong, 2011), are defined as following:

DEFINITION 1 (Satisfied Query): *In a user session, the query resulting in at least one clicked URL and is located at the end of the session is called a satisfied query.*

DEFINITION 2 (Unsatisfied Query): *Any query which causes at least one URL clicked and located ahead of the satisfied query in the same user session is called a unsatisfied query.*

A set of first unsatisfied queries of sessions are collected as the input and their corresponding satisfied queries are used as the benchmark query set of the refinement task. The performance of the system is evaluated at the top m of the candidate query list. Accuracy is defined as the total number of successfully predicted satisfied queries divided by the total number of test queries. Considering that users are more likely to care about the top ranked candidate queries, the metrics P@K (Precision at K) is also adopted to evaluate the results, where K is the number of top queries given by the model.

5.3.2 Experimental Design

An experiment was conducted to compare the performance of the proposed personalization framework and two existing query refinement techniques, i.e., an MI model and a context-based mutual information (CMI) model (Bar-Yossef & Kraus, 2011). In this study, MI and CMI were used to generate the original candidate query list of query refinement. The proposed model was applied to re-rank these two candidate lists. Specifically, two personalized models, i.e., personalized mutual information model (P-MI) and personalized context-based mutual information model (P-CMI), were obtained after using MI and CMI, respectively, as query refinement modules of the proposed query refinement framework as shown in Figure 5.4. Besides, another two personalized baselines, i.e., a topic model based framework (LTI) (Bing, Lam, & Wong, 2011), and a task-based method (MTP) (Luxenburger, Elbassuoni, & Weikum, 2008), were adopted for performance comparison. Specifically, LTI is a framework of utilizing latent topic consistency within a query to

re-rank the candidate query list, which does not consider task information in modeling user's search interests. MTP is a framework of matching search task for personalization, which considers the task information but doesn't examine user's search interests in a search task. Note that query scoring and noise filtering were conducted respectively in each method.

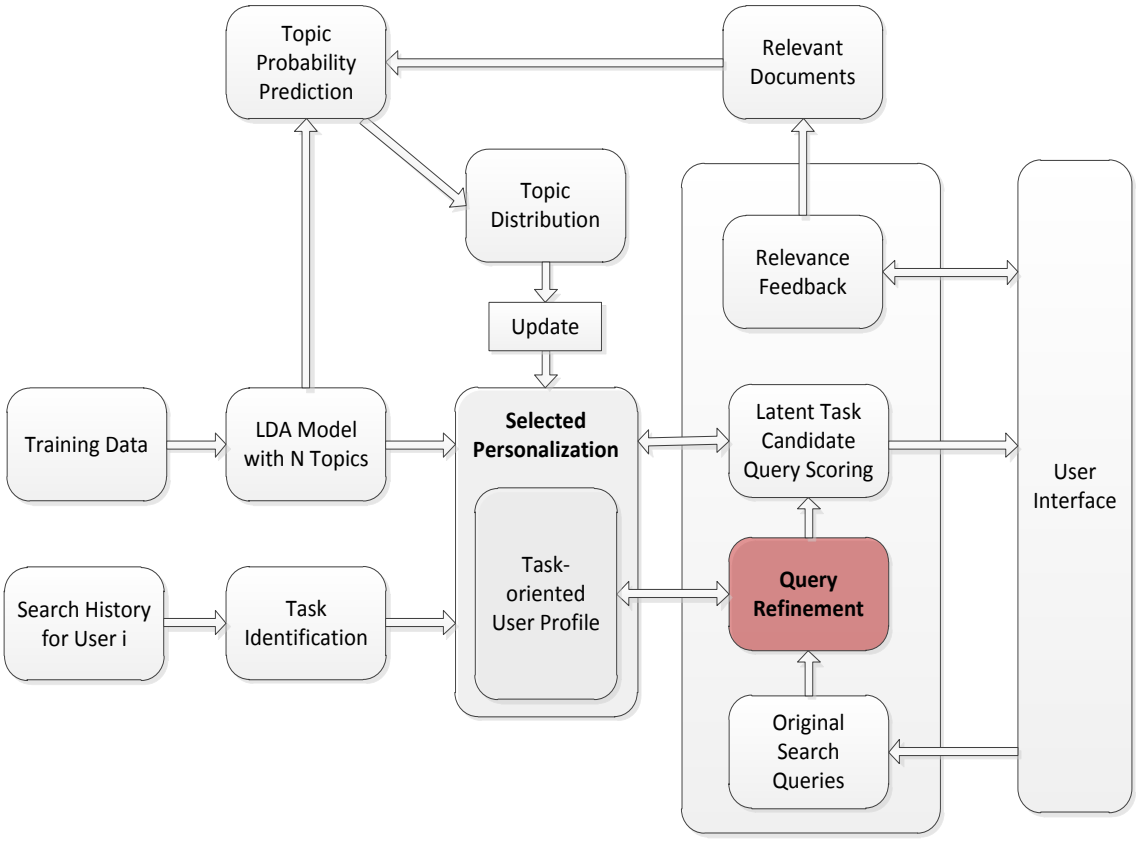


Figure 5.4 Framework of task-based personalization for query refinement.

User profiles were generated by randomly selecting 400 users with more than 50 sessions in the AOL training set. The first 25 sessions of each user were used to create the initial task-based interests of users, and the next 25 sessions were used to evaluate the effectiveness of the system. Sessions from 100 users were used in the parameter determination experiment, whereas the sessions of the other 300 users were used to

compare the effectiveness of all systems mentioned above. Note that for each session, the last query with at least one clicked document was used as the satisfied query for evaluation.

5.3.3 Experimental Results

5.3.3.1 Two-Step Rescoring Methods · Figure 5.5 shows the performance of two traditional query refinement methods (i.e., MI and CMI), proposed personalized methods (i.e., P-MI and P-CMI), and two baseline personalized methods (i.e., LTI and MTP).

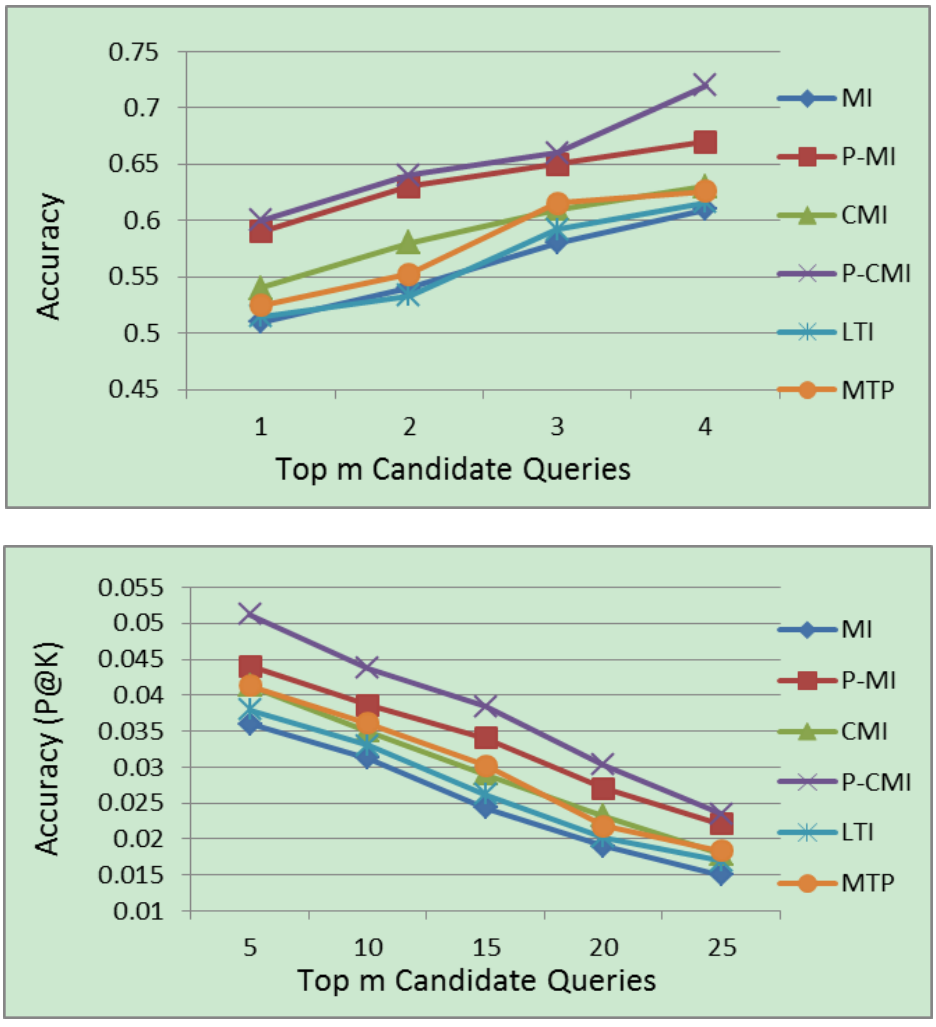


Figure 5.5 Performance comparisons among MI, P-MI, CMI, P-CMI, LTI, and MTP.

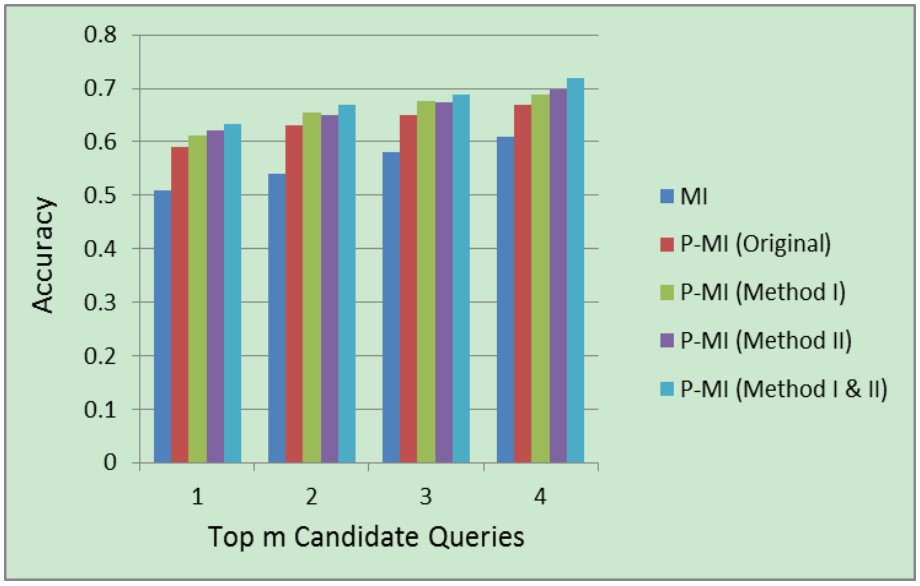
As shown, the proposed P-MI and P-CMI performed much better than MI and CMI respectively. For example, the accuracy values (at position 1) of P-MI and P-CMI were 0.59 and 0.60, whereas those of MI and CMI were 0.51 and 0.54. The P@5 value of P-MI and P-CMI were 0.043 and 0.051, whereas those of MI and CMI were 0.036 and 0.041. Within the four baselines, MTP and CMI outperformed other baseline method including MI and LTI. For example, The P@15 value of MTP and CMI were 0.030 and 0.029, whereas those of MI and LTI were 0.023 and 0.27. However, none of them performed as well as the proposed P-MI and P-CMI. Table 5.1 shows a sample of generated candidate queries by P-CMI.

Table 5.1 Sample of Experimental Results (P-CMI)

Original queries	Satisfied queries	Suggestions		
		Top 1	Top 2	Top 3
egyptian lentils	egyptian recipes	egyptian food	**egyptian recipes**	history lentils
pipe tobacco	pipe smoking	**pipe smoking**	pipe cigar	design tobacco
ford motor	ford parts	**ford parts**	ford hardware	Ford electronics
unclaimed funds	unclaimed money	**unclaimed money**	unclaimed investment	unclaimed investing
brownie cookies	brownie recipes	**brownie recipes**	brownie baking	brownie food
usps theft penalties	mail theft penalties	usps security penalties	**usps theft penalties**	mail theft penalties
casino phoenix	casino arizona	**casino arizona**	gambling phoenix	games phoenix
antique strollers	vintage strollers	**vintage strollers**	shopping strollers	history strollers
learning methods	teaching methods	education methods	**teaching methods**	tutorial methods
atlanta colleges	georgia colleges	atlanta school	**georgia colleges**	design colleges
wacky metaphors	funny metaphors	**funny metaphors**	cool metaphors	culture metaphors
strip poker	strip games	**strip games**	strip software	strip tools

5.3.3.2 Two Methods of Extracting Relevance Feedback ·The effectiveness of our proposed user-profiling model was evaluated for personalization of query refinement with different implicit feedback extraction methods. The values of three parameters were set (α at 0.1, β at 0.35, and γ at 0.25) to maximize the performance of the proposed user-profiling method introduced in Section 4.2. Figures 5.6 and 5.7 show the performance of the two pairs of experimental systems (MI with P-MI, CMI with P-CMI) under four different sets of relevance feedback extraction methods (described in Section 5.2.4), including: 1) "Original"– traditional relevance feedback extraction method (all clicked URLs are viewed as positive feedback, while all unclicked URLs are viewed as negative feedback); 2) "Method I"– applied when the user adds a word to the query; 3) "Method II"– applied when the user removes a word from the query; and 4) "Method I & II"– the combination of Method I and Method II.

It is observed that P-MI and P-CMI outperformed MI and CMI, even with the original method of extracting relevance feedback. For example, The P@10 value of P-MI was 0.038, whereas that of MI was 0.031. The P@10 value of P-CMI was 0.043, whereas that of CMI was 0.034. Both the performance difference between MI and P-MI and the one between CMI and P-CMI were statistically significant ($p < 0.05$). The major reason for this performance difference is that the proposed methods re-ranks candidate queries while considering user's interests within a task level, which cannot be captured by baseline approaches.



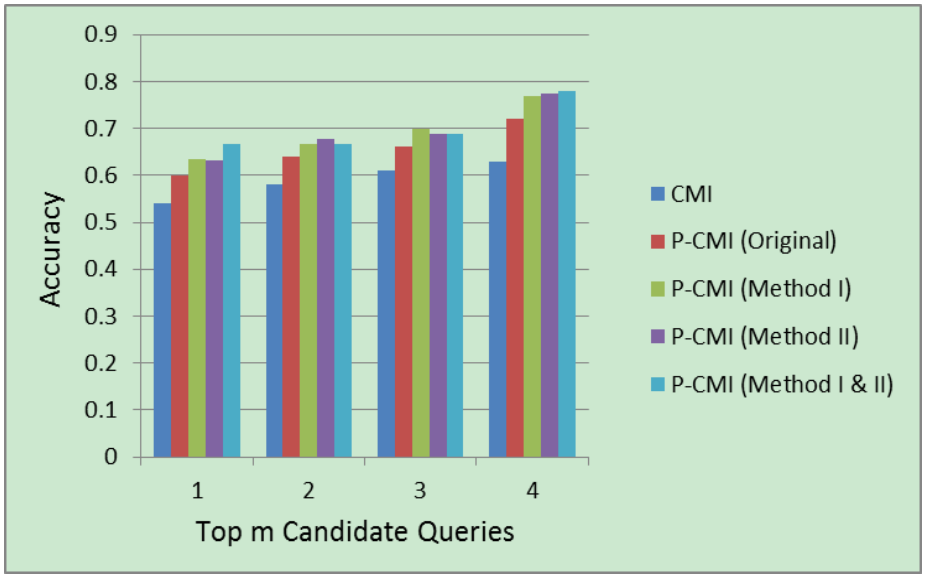
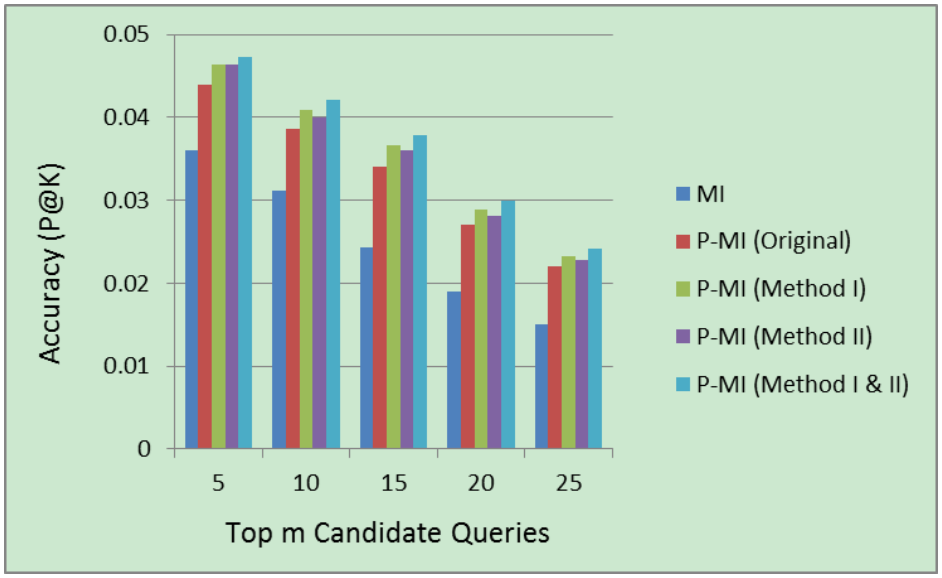


Figure 5.6 Comparison of scoring performance (Accuracy) between MI and P-MI, and between CMI and P-CMI.



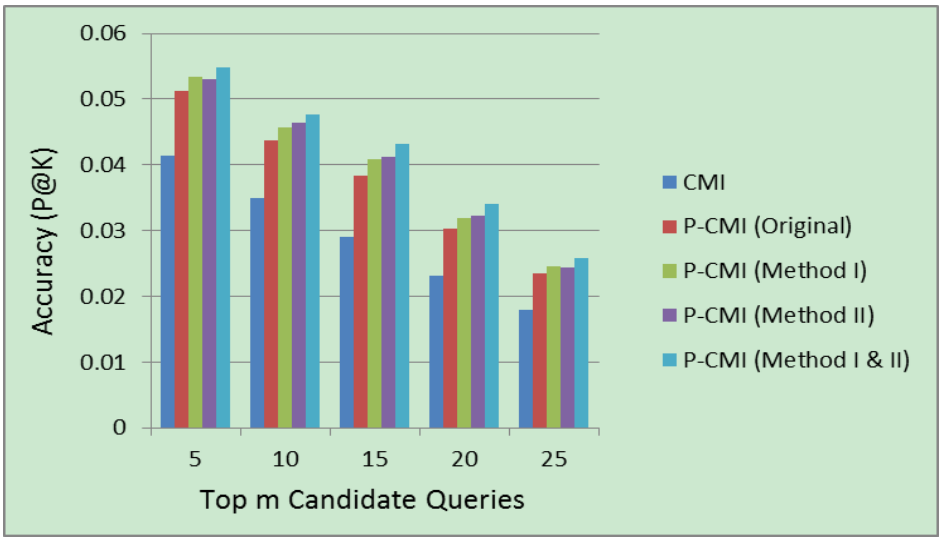


Figure 5.7 Comparison of scoring performance (P@K) between MI and P-MI, and between CMI and P-CMI.

The utilization of the Method I in P-MI improved precision of MI by 8.1% for precision@5, 9.7% for precision@10, 11.2% for precision@15, 21.1% for precision@20, and 14% for precision@25. These results confirm that the model can improve the learning accuracy of the user's information needs by maintaining and updating the P-STD and N-LTD. Because the proposed model performed best when γ was 0.25, the influence of

D_{pos} of STD contributes more to precision than the D_{neg} of LTD does. Therefore, P-STD is effective in learning dynamic interests of users.

The Method II is not as effective as the Method I in P-CMI. The accuracy of the top four candidate queries is not significantly improved, because the Method II is highly reliant on N-STD and P-LTD. Although N-STD helps to low-rank the irrelevant queries in the candidate query list, P-LTD alone is not sufficient to elevate the relevant query because P-LTD has poor adaptability to changes of interest.

5.4 Summary

In this chapter, a module is introduced for rescoring the candidate queries generated by the traditional query refinement techniques, and this module has two main components. First, a graphical model is proposed to score the candidate query which can detect and maintain the latent task consistency of terms in a query. Second, an algorithm is presented to determine if the user's past search task information has the potential benefits of re-ranking candidate queries for their current search activities. Moreover, two methods of extracting the implicit relevance feedback of users are proposed by examining the user's query reformulation behaviors. In the experiment, the influence of the proposed model on the system performance is examined by applying different methods of extracting the relevance feedback of users. Experimental results show that the task-based user modeling method increased the accuracy of traditional query refinement significantly.

CHAPTER 6

SUMMARY AND LIMITATIONS

6.1 Summary

The main objective of this research is to investigate how to achieve effective query refinement personalization, through modeling and applying user's task-based search interests in re-ranking candidate queries generated by traditional query refinement techniques.

In Chapter 3, a cross-session based query analysis method with a best-link model is proposed to improve the performance of task identification. Specifically, search queries within a search session are segmented into sub-tasks by using the best-link model to learn query connections from users' search activities. Then a graph-based representation method is utilized to calculate the contextual pairwise similarity of queries. Finally, Search tasks are identified by grouping similar sub-tasks from all search sessions together. Experimental results demonstrated that the proposed best-link task identification methods, i.e., BL and BL-G, outperformed the baselines significantly in all three evaluation metrics, i.e., F1, Rand index, and Jaccard index. Moreover, BL-G outperformed BL, which indicates that the proposed graph-based representation is more effective than the bag-of-words based approach in the best-link model. It was also observed that the session boundary did impact the performance of all compared task identification algorithms. Most of them achieved the highest performance on these three metrics when the time interval was set at 25 minutes.

In Chapter 4, a four-tuple descriptor model is introduced to represent and learn the long-term (positive and negative) and short-term (positive and negative) user interests for each task generated from past user search histories. Experimental results indicated that the TD model outperformed both LTD and STD significantly. Although the performance of LTD increased gradually as learning user's interests within each learning phase, it suffered a sharp decrease of accuracy at each learning phase transition. The reason is that the LTD model is incapable of unlearning the old interests quickly when the user shifts to a new search interest. By contrast, STD possesses better adaptability to interest changes during task transitions. But STD does not learn the user interest as stably as LTD does. Given that STD does not have an accumulation of former session interests, its accuracy in matching user interest fluctuated greatly compared with the performance of LTD. TD overcomes the weaknesses of both LTD and STD. Thus, it outperforms LTD in unlearning older interests and is superior than STD in matching user interest.

In Chapter 5, a two-step personalization method is proposed to re-rank candidate queries generated by traditional query refinement methods. First, a graphical model is used to access the latent task dependency of terms in a candidate query by exploiting the latent task consistency value. Second, a personalization algorithm is proposed to selectively applying users' task-based search interests on personalization of query refinement by re-ranking the candidate query list. Experimental results demonstrated that the proposed P-MI and P-CMI performed much better than the baselines. Specifically, P-MI and P-CMI performed much better than the traditional query refinement baselines, i.e., MI and CMI, because the proposed methods improved the relevance of candidate queries using user's task-based dynamic search interests. Moreover, both P-MI and P-CMI outperformed other

two personalized baseline methods including LTI and MTP. The major reason for this performance difference is that the proposed methods scored a query while taking into account the user's interests within a task level, which cannot be captured by either LTI or MTP. This result also indicates that the user's relevance feedback within a session or task is useful in generating the satisfied query of the user.

6.2 Limitations

6.2.1 Cold Start Problem

Cold start problem is an issue of the proposed framework using the AOL log dataset. When a user is new to the system or just starts to conduct a search task, he or she might not have enough search history to be learned for re-ranking the candidate query list. One solution is to apply the regular query refinement process without personalization in the beginning.

6.2.2 AOL Dataset Limitation

In this research, the AOL dataset was used to analyze user's search interests, because AOL search log was recorded by one of the most famous search engines. There are two main limitations on using the AOL dataset for this study. First, this dataset only covers a 3-month period, which is not a very long time for learning user's long-term search interests. Second, the dataset size is small so that the number of users who have more than 50 search sessions for the experiments in Section 5.3.2 is limited.

However, the AOL dataset is still adopted in this research because it is the only publicly accessible English log dataset. Moreover, a ground truth dataset that labels user search task information is vital to this research. Lucchese et al. (2011) create such a dataset based on the AOL search log. Using Lucchese et al's dataset requires using the AOL

dataset as raw data. These two factors make AOL search log dataset the only suitable dataset for this research.

6.2.3 Ground Truth Limitations

In Section 3.3, a human annotated ground truth dataset of search tasks is used for evaluating the proposed task identification algorithm. This ground truth is relatively small in size, which contains 554 search tasks in total with average 2.57 queries per task. In other words, this dataset only contains the search data from a small group of users. Generalizability might be an issue.

Moreover, in Section 5.3, it was assumed that, in a search session, the user's last original query with at least one clicked document is considered a satisfied query and adopted as the ground truth to evaluate the performance of personalized query refinement. This is not necessarily the case, since users' search behavior is complex and they may end a search session with unsatisfied queries even though a search result is clicked. However, considering that the explicit user's satisfaction information is not available in the AOL dataset, this assumption is adopted as a compromise since it is widely adopted in existing studies.

6.3 Summary

This chapter first presents the summary of this research, including TOQUE framework, experimental design, and results. It continues to describe the limitations of this study.

CHAPTER 7

DISCUSSION AND CONTRIBUTIONS

7.1 Discussion

7.1.1 Balancing Interest Weights of LTD and STD

As discussed in Chapter 4, although the TD model outperformed LTD and STD in learning user's dynamic search, there is a tradeoff between interest weight γ and $(1 - \gamma)$ determining the importance of LTD and STD in TD respectively, depending on various factors, e.g., the frequency of the user's search activity. For example, as for the users who conduct search activities daily, the high weight of LTD would be effective because LTD can keep track of the user's gradually accumulated long-term interests with continuous search data. By contrast, as for the users who conduct search activities only several times a month, a high weight of LTD will result in less precise candidate query lists, because the change of user's long-term interests may not be captured by the model due to the data scarcity issue. In this case, STD should be assigned with a high weight to effectively learn the user's current search interests.

Most search engines will have a mixture of users. Therefore, it is crucial to select a γ value that ensures the combined search effectiveness for all users is optimal. It is recommended to start with a balanced LTD and STD, namely $\gamma = 0.5$. As more search history is gathered, optimizing γ using an approach similar to that in Chapter 4 periodically and using clicked candidate queries as ground truth is prudent. However, the optimization of γ is out of the scope of this study.

7.1.2 Finding Top k Related Tasks

In the proposed personalization algorithm in Chapter 5, the number of top k similar tasks to be compared with the candidate query categories can also vary. In this study, k is set to 15 based on an empirical study. If k is too large, irrelevant tasks might be added to the comparison, which undermines user intent extraction. It will also increase the number of the pairwise similarity calculations, which leads to higher computational cost. If k is too small, fewer tasks are added as related tasks and less information can be extracted from the user's previous search tasks, especially when the user's search data are collected over a short period of time. In practice, empirical efforts need to be conducted to obtain optimal results. For example, a large training dataset can be divided into multiple user groups and each group only consists of users with similar length of search history. Then, k can be optimized under each user group regarding the user's search experience.

7.1.3 Collecting Relevance Feedback

In TOQUE, user's clicked URLs are adopted as relevance feedback for learning user's search interests. Specifically, the queries with at least one clicked document were used for evaluation, and the clicked URLs of these queries were used as the relevance feedback. However, there are accidental clicks on the search results. In this case, the clicked URLs may have nothing to do with the user's search interests. In practice, a minimum number of clicked search results as a threshold can be defined for each query, because the more clicks a user makes in the search results for a query, the more likely the user is really interested in tasks or interests associated with the query. However, if the threshold is too high, it will filter out some valuable queries and clicks. Therefore, a trade-off should be considered when collecting the relevance feedback. Another possible solution is that, those accidental

clicks can be detected and removed by examining the user's browsing behaviors, i.e., the amount of time spent on the page and user actions such as scrolling the mouse and bookmarking a page. However, this is out of the scope of the study.

7.1.4 Computational Complexity

A computational complexity issue results from TOQUE due to its nature as a personalized retrieval system using the LDA model. As the size of document collection increases, TOQUE will require more computing resources for indexing documents. More computing power will be required to re-train a topic model and topic distribution for all documents in the database. However, the search efficiency should not be much affected, because this training/generating process can be conducted offline.

7.2 Contributions

Query refinement, a well-known information retrieval technique, has been proven effective to reformulate ill-formed queries to enhance the relevance of search results. However, current studies of query refinement do not consider users' diverse search intentions. TOQUE bridges this gap by utilizing task-based user profiles to improve the precision of candidate queries. Specifically, AOL search log was examined to model search interests of users: task and session information were extracted as contextual information for user interest modeling. The candidate queries were re-ranked based on user's task-based search interests. As a result, the effectiveness of the candidate query list was improved. The outcomes of the research activities make the following contributions.

7.2.1 A Framework of Query Refinement Personalization

TOQUE is of great value to improve the effectiveness of traditional query refinement techniques. Instead of simply exploring the alternative words of high lexical or topic similarities with the words in the original query, TOQUE focuses on generating candidate queries which are most related to the user's search interests. Coupled with proposed personalization algorithm, this framework is highly valuable to filter out a great deal of candidate queries which do not meet the user's search interest and preference.

7.2.2 A Four-tuple Descriptor based User Profiling Model

Learning user's search interests is challenging in current Web environment because user's search interests are diverse and changing over time. TOQUE adopts a four-tuple topic descriptor representation of user profiling, which models the user's interests at the task level to improve the rankings of the candidate queries for query refinement. When more user's search history is collected, their search session and task information are built incrementally. These search context information is valuable not only for identifying user's current search activity but also for applying user's search interest intelligently to improve the performance of query refinement.

Moreover, the experimental results not only determine the effectiveness of the framework, but also provide parameter tuning for user interest modeling. For example, when the user has a totally new interest, his historical search interests will not be applicable for personalization of query refinement, which might influence the performance of the system. By dividing and modeling user's long- and short-term search interests, this research is of high value to solve this problem through adjusting the weights of LTD and

STD, thus informing the IR communities on the relationship between LTD/STD and the learning rate of user's interests.

7.2.3 A Best-link Model with Graph-based Representation

In this research, a cross-session based method is proposed to identify search tasks in user's search history. Specifically, a best-link model is introduced to generate the latent term structure within a candidate query. Moreover, a graph-based representation is proposed to explicitly represent user's relevance feedback as a semantic graph. Then, the pairwise similarity of relevance feedback from adjacent queries is calculated using an existing graph similarity measure. The resultant effectiveness of grouping related queries for each search task is significantly improved.

7.3 Summary

In this chapter, the discussion of this research, including balancing interest weights of LTD and STD, finding top k related tasks, collecting relevance feedback, and computational complexity are illustrated. Then the main contributions of this study are also summarized.

REFERENCES

Agichtein, E., White, R. W., Dumais, S. T., & Bennet, P. N. (2012) *Search, interrupted: understanding and predicting search task continuation*. Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, Portland, Oregon, USA

Ahmed, A., Low, Y., Aly, M., Josifovski, V., & Smola, A. J. (2011). *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.

Ahn, J.-w., Brusilovsky, P., Grady, J., He, D., & Syn, S. Y. (2007). *Open user profiles for adaptive news systems: help or harm?* Proceedings of the 16th International Conference on World Wide Web, Banff, Alberta, Canada.

Ahn, J.-w., Brusilovsky, P., He, D., Grady, J., & Li, Q. (2008). *Personalized web exploration with task models*. Proceedings of the 17th International Conference on World Wide Web, Beijing, China.

Anick, P. (2003). *Using terminological feedback for web search refinement: a log-based study*. Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, Toronto, Canada.

Bar-Yossef, Z., & Kraus, N. (2011). *Context-sensitive query auto-completion*. Proceedings of the 20th International Conference on World Wide Web, Hyderabad, India.

Bennett, P. N., White, R. W., Chu, W., Dumais, S. T., Bailey, P., Borisyuk, F. & Cui, X. (2012). *Modeling the impact of short- and long-term behavior on search personalization.* Proceedings of the 35th international ACM SIGIR conference on research and development in information retrieval, Portland, Oregon, USA.

Bigi, B. (2003). *Using Kullback-Leibler distance for text categorization*. Proceedings of the 25th European Conference on IR research, Pisa, Italy.

Bing, L., Lam, W., & Wong, T.-L. (2011). *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.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). *Latent dirichlet allocation*. Journal of Machine Learning Research, *3*, 993-1022.

Boldi, P., Bonchi, F., Castillo, C., Donato, D., Gionis, A., & Vigna, S. (2008). *The query-flow graph: model and applications*. Proceedings of the 17th ACM Conference on Information and Knowledge Management, Napa Valley, California, USA.

Chang, Y.-S., He, K.-Y., Yu, S., & Lu, W.-H. (2006). *Identifying User Goals from Web Search Results*. Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, Hong Kong, Hong Kong.

Chen, L., & Sycara, K. (1998). *WebMate: a personal agent for browsing and searching*. Proceedings of the Second International Conference on Autonomous Agents, New York, NY, USA.

Chen, C., Yang, M., Li, S., Zhao, T., & Qi, H. (2010). *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.

Deng, H., King, I., & Lyu, M. R. (2009). *Entropy-biased models for query representation on the click graph*. Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval, Boston, MA, USA.

Dhillon, P. S., Sellamanickam, S., & Selvaraj, S. K. (2011). *Semi-supervised multi-task learning of structured prediction models for web information extraction*. Proceedings of the 20th ACM International Conference on Information and Knowledge Management, Glasgow, Scotland, UK.

Dou, Z., Song, R., & Wen, J.-R. (2007). *A large-scale evaluation and analysis of personalized search strategies*. Proceedings of the 16th International Conference on World Wide Web, Banff, Alberta, Canada.

Downey, D., Dumais, S., Liebling, D., & Horvitz, E. (2008). *Understanding the relationship between searchers' queries and information goals*. Proceedings of the 17th ACM Conference on Information and Knowledge Management, Napa Valley, California, USA.

Dwork, C., Kumar, R., Naor, M., & Sivakumar, D. (2001). *Rank aggregation methods for the Web*. Proceedings of the 10th International Conference on World Wide Web, Hong Kong, Hong Kong.

Fan, W., Li, J., Ma, S. Wang, H., & Wu, Y. (2010). *Graph homomorphism revisited for graph matching*. Proceedings of the VLDB Endowment. 3, 1-2, 1161-1172.

Gauch, S., Chaffee, J., & Pretschner, A. (2003). *Ontology-based personalized search and browsing*. Web Intelligence and Agent Systems,1(3), 219-234.

Guo, J., Xu, G., Li, H., & Cheng, X. (2008). *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.

Hassan, A., Jones, R., & Klinkner, K. L. (2010). *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.

Hassan, A., & White, R. W. (2012). *Task tours: helping users tackle complex search tasks*. Proceedings of the 21st ACM International Conference on Information and Knowledge Management, Maui, Hawaii, USA.

Haveliwala, T. H. (2002). *Topic-sensitive PageRank*. Proceedings of the 11th International Conference on World Wide Web, Honolulu, Hawaii, USA.

He, D., Göker, & Harper, D. J. (2002). *Combining evidence for automatic web session identification*. Information Processing and Management, 38(5), 727-742.

Huang, J. & Efthimiadis. E. N. (2009). *Analyzing and evaluating query reformulation strategies in web search logs*. Proceedings of the 18th ACM Conference on Information and Knowledge Management. New York, NY, USA, 77-86.

Ji, M., Yan, J., Gu, S., Han, J., He, X., Zhang, W. V., & Chen, Z. (2011). *Learning search tasks in queries and web pages via graph regularization*. Proceedings of the 34th international ACM SIGIR conference on research and development in Information retrieval, Beijing, China.

Jones, R., & Klinkner, K. L. (2008). *Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs*. Proceedings of the 17th ACM Conference on Information and Knowledge Management, Napa Valley, California, USA.

Kohlschütter, C., Chirita, P. A., & Nejdl, W. (2006). *Using link analysis to identify aspects in faceted web search*. Proceedings of the 29th International ACM SIGIR Conference on Research and Development in Information Retrieval, Faceted Search Workshop, Seattle, Washington, USA.

Kotov, A., Bennett, P. N., White, R. W., Dumais, S. T., & Teevan, J. (2011). *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.

Lee, U., Liu, Z., & Cho, J. (2005). *Automatic identification of user goals in Web search*. Proceedings of the 14th International Conference on World Wide Web, Chiba, Japan.

Li, L., Kitsuregawa, M. (2007). *Personalizing web search via modelling adaptive user profile*. Proceedings of Distant Early Warning Conference.

Liao, Z., Song, Y., He, L., & Huang, Y. (2012). *Evaluating the effectiveness of search task trails*. Proceedings of the 21st International Conference on World Wide Web.

Liu, C. Belkin, N. J, & Cole, M. J. (2012). *Personalization of search results using interaction behaviors in search sessions*. Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA.

Lucchese, C., Orlando, S., Perego, R., Silvestri, F., & Tolomei, G. (2011). *Identifying task-based sessions in search engine query logs*. Proceedings of the fourth ACM International Conference on Web Search and Data Mining, Hong Kong, China.

Luxenburger, J., Elbassuoni, S., & Weikum, G. (2008). *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.

Mei, Q., Klinkner, K., Kumar, R., & Tomkins, A. (2009). *An analysis framework for search sequences*. Proceedings of the 18th ACM Conference on Information and Knowledge Management, Hong Kong, China.

Piwowarski, B., Dupret, G., & Jones, R. (2009). *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.

Qiu, F., & Cho, J. (2006). *Automatic identification of user interest for personalized search*. Proceedings of the 15th International Conference on World Wide Web, Edinburgh, Scotland.

Rocchio, J. J. (1971). *Relevance feedback in information retrieval*. In The SMART Retrieval System: Experiments in Automatic Document Processing, pages 313-323, Englewood Cliffs, NJ, USA.

Rose, D. E., & Levinson, D. (2004). *Understanding user goals in web search*. Proceedings of the 13th International Conference on World Wide Web, New York, NY, USA.

Rosen-Zvi, M., Chemudugunta, C., Griffiths, T., Smyth, P., & Steyvers, M. (2010). *Learning author-topic models from text corpora*. ACM Transactions On Information Systems, *28*(1), 1-38.

Sadikov, E., Madhavan, J., Wang, L., & Halevy, A. (2010). *Clustering query refinements by user intent*. Proceedings of the 19th International Conference on World Wide Web, Raleigh, North Carolina, USA.

Shen, D., Sun, J., Yang, Q., & Chen, Z. (2006). *Building bridges for web query classification*. Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA.

Sieg, A., Mobasher, B., & Burke, R. (2007). *Web search personalization with ontological user profiles*. Proceedings of the 16th ACM Conference on Conference on Information and Knowledge Management, Lisboa, Portugal.

Song, Y., Huang, J., Councill, I. G., Li, J., & Giles, C. L. (2007). *Generative models for name disambiguation.* Proceedings of the 16th International Conference on World Wide Web, Banff, Alberta, Canada.

Speretta, M., & Gauch, S. (2005). *Personalized search based on user search histories*. Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence.

Spink, A., Park, M., Jansen, B. J., & Pedersen, J. (2006). *Multitasking during web search sessions*. Information Processing and Management.

Tan, B., & Peng, F. (2008). *Unsupervised query segmentation using generative language models and wikipedia*. Proceedings of the 17th International Conference on World Wide Web, Beijing, China.

Tan, B., Shen, X., & Zhai, C. (2006). *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.

Wang, G., & Lochovsky, F. H. (2004). *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.

Wang, X., & Zhai, C. (2008). *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.

White, R. W., Bailey, P., & Chen, L. (2009). *Predicting user interests from contextual information*. Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval, Boston, MA, USA.

Widyantoro, D. H., Ioerger, T. R., & Yen, J. (1999). *An adaptive algorithm for learning changes in user interests*. Proceedings of the eighth International Conference on Information and Knowledge Management, Kansas City, Missouri, USA.

Wu, H. C., Luk, R. W. P., Wong, K. F., & Kwok, K. L. (2006). *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.

Xu, Y., Wang, K., Zhang, B., & Chen, Z. (2007). *Privacy-enhancing personalized web search*. Proceedings of the 16th International Conference on World Wide Web, 591-600.

Yi, J., & Maghoul, F. (2009). *Query clustering using click-through graph*. Proceedings of the 18th International Conference on World Wide Web, Madrid, Spain.

Zhou, B., Jiang, D., Pei, J., & Li, H. (2009). *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.