Dynamic User Profiling for Search Personalisation

Thesis

DYNAMIC USER PROFILING FOR SEARCH PERSONALISATION

by

Thanh Tien Vu

Dissertation submitted in fulfilment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

School of Computing and Communications
Faculty of Science, Technology, Engineering and Mathematics
The Open University
Milton Keynes, United Kingdom

March 2017

ABSTRACT

The performance of a personalised search system largely depends upon the ability to build user profiles which accurately capture the user's search interests. However, many approaches to user profiling have neglected the dynamic nature of the user's search interests. That is, a user's search interests typically change in response to their interactions with the search system during the search period. Therefore, a profile built for previous searches might not reflect that user's current search interests.

A widely used type of profile represents the *topical interests* of the user. In these cases, a typical approach is to build a user profile using topics discussed in documents which the user has found relevant, and where the topics are obtained from a human-generated ontology or directory. However, a key limitation of these approaches is that many documents may not contain the topics covered in the ontology. Moreover, the human-generated ontology requires manual effort to determine the correct categories for each document.

In this research, we address these problems by proposing novel techniques for dynamically building user profiles which capture the user's search interests changing over time. Instead of using a human-generated ontology, we use a topic modelling technique (Latent Dirichlet Allocation) for unsupervised extraction of the topics from documents. To dynamically build user profiles, we make two important assumptions. First, that the group of users with whom a user shares a set of common interests may be *different* depending upon the particular topic of interest. Second, the more recently clicked/relevant documents tell us more about the user's *current* search interests.

To test these assumptions, we develop and implement dynamic user profiles, and then evaluate them on two search personalisation tasks. Our first chosen task is personalising search results returned by a Web search engine, and the second is the task of personalising query suggestions made by an Intranet search engine. We found that dynamic user profiles can significantly improve the ranking quality over well-established baselines.

ORIGINALITY STATEMENT

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at the Open University or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at the Open University or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.

. . . .
Thanh Tien Vu

ACKNOWLEDGMENTS

First and foremost, I am grateful to my PhD supervisors, Dawei Song and Alistair Willis, for always being positive and supportive. They agreed to start with me and patiently guided me throughout this long scientific journey. They have always been there for me to provide encouragement and helpful feedback on my research. To be honest, I would never have completed a PhD without them.

I am also grateful to Anne De Roeck and Stefan Rueger for their valuable suggestions and comments on this dissertation, and also Michael Philip Oakes and Paul Piwek for accepting to read my work and be examiners.

I would like to thank my former supervisor, Quang-Thuy Ha, who introduced me to Data Mining and encouraged me to pursue this PhD.

I highly appreciated the opportunities to work with Udo Kruschwitz, Son Ngoc Tran, Dat Quoc Nguyen and Jingfei Li in search personalisation, Dai Quoc Nguyen in sentiment analysis and Son Xuan Vu in image search. Also, thank you to Dat Quoc Nguyen for your encouragement which helped me to get through the tough times when writing this dissertation.

I thank the Open University for accepting my research proposal and funding my PhD. Thanks to Mike Richards for being such a kind third-party monitor. Thank you to Marian Petre and Robin Laney for arranging the PG forums which helped me to develop research skills.

A special thank you to David Bowers for his great advice and critical comments during my work, and especially for his fantastic sailing trips. His support was of inestimable value in bringing me back on the right track for my thesis writing.

Thank you to Giang Binh Tran and Dang Duc Pham for the fantastic time we spent together in developing Bitdealo.vn and then Tripi.vn which is a fast growing startup in tourism located in Vietnam.

Last but not least, I would like to thank my parents, Chien Van Vu and Hoa Thi Pham, for their unconditional love and constant support. I am thankful to my daughter Anh Hong Vu, who has given me the greatest motivation in finishing my work. Especially, I would like to thank Trang Thi Huyen Vu, my closest friend and my dearest wife who gave up her success to support mine by always being there for me. I sincerely appreciate their belief in me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

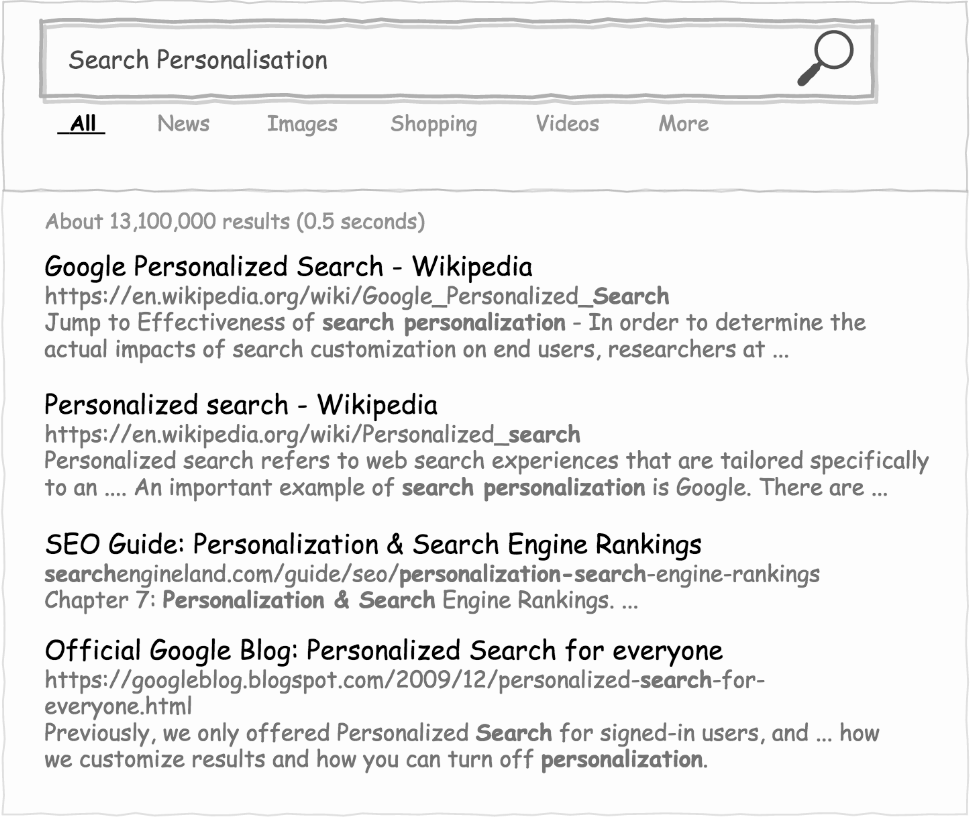
Information retrieval (IR) aims to find material of an unstructured nature which satisfies a searcher's information need from large collections (Manning et al., 2008). The primary focus of the IR field since the 1950s has been on *textual* documents, such as Web pages, emails, scholarly papers, and books. As these are normally in the form of free text, they are relatively unstructured compared to, for example, database records, such as hotel reservations or bank account records (Croft et al., 2009). Up until the 1990s, only a few people, such as reference librarians or professional searchers, dealt in obtaining information from information systems (Manning et al., 2008; Croft et al., 2009). Most people preferred finding out information from other people, for example by using travel agents to book their travel.

However, for the past twenty years, information retrieval activity has changed significantly, as seen by the arrival of the World Wide Web and Web search engines such as AOL, Google, and Bing. Searching the Web to access information, business or leisure, or to carry out online shopping is a daily activity for most people and some of the most popular uses of the Internet (Croft et al., 2009). According to InternetLiveStats[1], in 2016 Google received over 4.8 billion searches per day and 1.7 trillion searches per year worldwide. As IR technology has been playing a more and more important role in people's daily lives, it has attracted increasing attention from

[1] http://www.internetlivestats.com

companies and universities aiming to improve search by proposing better IR algorithms (Croft et al., 2009).

IR systems can be classified by the scale at which they operate. In the context of searching on the World Wide Web (*Web search*), the system provides search over billions of documents stored on millions of computers (Manning et al., 2008). Web search is by far the most common application involving IR and it also plays a crucial part in smaller scale applications in other domains (Croft et al., 2009). For instance, *Vertical search* is a specialised form of Web search in which the domain of the search is restricted to a particular topic. *Enterprise search* and *Domain-specific search* (such as Intranet searches) involve finding the required information from particular collections, such as a university's internal documents or a database of research articles. In these cases, the documents are typically stored on centralised file systems with one or a handful of dedicated machines providing search over the collection (Croft et al., 2009).



Search Personalisation

About 13,100,000 results (0.5 seconds)

Google Personalized Search - Wikipedia
https://en.wikipedia.org/wiki/Google_Personalized_**Search**
Jump to Effectiveness of **search personalization** - In order to determine the actual impacts of search customization on end users, researchers at ...

Personalized search - Wikipedia
https://en.wikipedia.org/wiki/Personalized_**search**
Personalized search refers to web search experiences that are tailored specifically to an .... An important example of **search personalization** is Google. There are ...

SEO Guide: Personalization & Search Engine Rankings
searchengineland.com/guide/seo/**personalization-search**-engine-rankings
Chapter 7: **Personalization & Search** Engine Rankings. ...

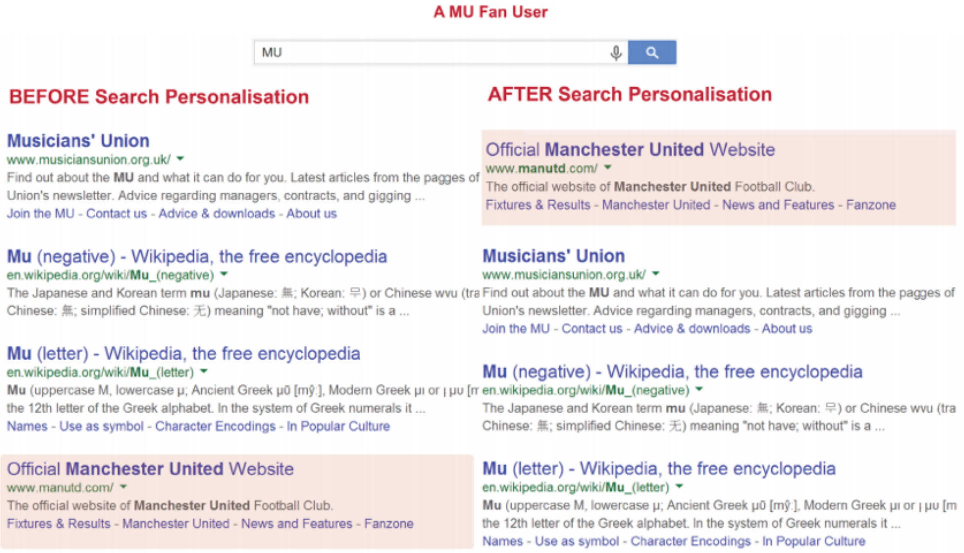
Official Google Blog: Personalized Search for everyone
https://googleblog.blogspot.com/2009/12/personalized-**search**-for-everyone.html
Previously, we only offered Personalized **Search** for signed-in users, and ... how we customize results and how you can turn off **personalization**.

Figure 1.1: Search result page returning after submitting a query "Search Personalisation"

A typical search scenario is that of a user submitting a *query* to an IR system. The query represents that user's information need, and the system then returns answers in the form of a *ranked document list*. For example, in Figure 1.1, after a user submits a query "Search Personalisation" to the search engine, the user receives a list of documents which the search engine believes address the user's information need. For each document within the list, the search engine also returns a small but relevant portion of the document in which the primary objects are highlighted.

The IR system needs algorithms that accurately perform the comparison between the query and the document in the system's stored collection to return the answers to the user's submitted query (Croft et al., 2009). One characteristic of classical IR systems (eg. Altavista, AOL search engines) is that these normally return the same result list to users who submitted the same input query, regardless of who submitted the query, why the query was submitted, where the query was submitted, or what other queries were submitted in the same search session (Croft et al., 2009). However, input queries are usually short and ambiguous (Dou et al., 2006). Different users might have different information needs even if they submitted identical queries. For instance, one user submitting an input query "MU" to a search engine may need information about the Manchester United Football Club, while another user submitting the same query may need information about the Musician's Union. Therefore, search systems which do not take user interests into account, may not always be able to satisfy the searchers with their desired information.

To tackle this problem, *search personalisation* has attracted increasing attention from both academia and industry, and is now an important feature of commercial search engines (Teevan et al., 2005; Dou et al., 2007; Teevan et al., 2009; Nanas et al., 2010; Bennett et al., 2012; Harvey et al., 2013; Liu, 2015; Yang et al., 2016; Cheng et al., 2016). Unlike classical IR methods, personalised search engines utilise the personal data of each user to tailor search results to that user depending not only on the input query but also on the user's search interests. Such personal data can be used to build a *user profile*. Figure 1.2 shows an artificial example of the effect of search personalisation. Knowing that the searcher is a Manchester United football club fan, the search system returned a Manchester United-related document at the top of the result list (on the right-hand side), which is different from the ranking of the results without personalisation (on the left-hand side).



A MU Fan User

MU

BEFORE Search Personalisation

Musicians' Union
www.musiciansunion.org.uk/
Find out about the **MU** and what it can do for you. Latest articles from the pagges of
Union's newsletter. Advice regarding managers, contracts, and gigging ...
Join the MU - Contact us - Advice & downloads - About us

Mu (negative) - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/**Mu**_(negative)
The Japanese and Korean term **mu** (Japanese: 無; Korean: 무) or Chinese wvu (tra
Chinese: 無; simplified Chinese: 无) meaning "not have; without" is a ...

Mu (letter) - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/**Mu**_(letter)
Mu (uppercase M, lowercase μ; Ancient Greek μῦ [mŷː], Modern Greek μι or μυ [m
the 12th letter of the Greek alphabet. In the system of Greek numerals it ...
Names - Use as symbol - Character Encodings - In Popular Culture

Official **Manchester United** Website
www.manutd.com/
The official website of **Manchester United** Football Club.
Fixtures & Results - Manchester United - News and Features - Fanzone

AFTER Search Personalisation

Official **Manchester United** Website
www.**manutd**.com/
The official website of **Manchester United** Football Club.
Fixtures & Results - Manchester United - News and Features - Fanzone

Musicians' Union
www.musiciansunion.org.uk/
Find out about the **MU** and what it can do for you. Latest articles from the pagges of
Union's newsletter. Advice regarding managers, contracts, and gigging ...
Join the MU - Contact us - Advice & downloads - About us

Mu (negative) - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/**Mu**_(negative)
The Japanese and Korean term **mu** (Japanese: 無; Korean: 무) or Chinese wvu (tra
Chinese: 無; simplified Chinese: 无) meaning "not have; without" is a ...

Mu (letter) - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/**Mu**_(letter)
Mu (uppercase M, lowercase μ; Ancient Greek μῦ [mŷː], Modern Greek μι or μυ [m
the 12th letter of the Greek alphabet. In the system of Greek numerals it ...
Names - Use as symbol - Character Encodings - In Popular Culture

Figure 1.2: A Manchester United FC fan submits the query "MU" to a search engine

The user profile is crucial to effective personalisation (Teevan et al., 2005, 2009). Recently, much research has been done on learning *user models* or *user profiles* to represent a user's interests and so personalise the user's search (Nanas et al., 2003; Sugiyama et al., 2004; Sieg et al., 2007; Bennett et al., 2012; Harvey et al., 2013; Liu, 2015; Yang et al., 2016). Previous work has also shown that improving the performance (that is, properties such as accuracy, speed and usability) of search engines which use personalisation depends on the *richness* of a user profile (Teevan et al., 2005, 2009), typically built from the documents relevant to the user.

A widely used type of a user profile represents topical interests of the user (Bennett et al., 2012; Sieg et al., 2007; Harvey et al., 2013; Raman et al., 2013) harvested from documents that are considered relevant to that user (e.g., those documents clicked by the user and for which the user's dwell time or viewing time is greater than 30 seconds) (Bennett et al., 2012; Harvey et al., 2013). The topics of a document can be obtained from a human-generated ontology, such as the Open Directory Project (ODP)[2,3]. However, this approach suffers from a limitation in that many documents do not appear in the online categorisation scheme. It also requires costly manual effort to determine the correct categories for each document (Bennett et al., 2012; Raman et al., 2013).

In addition, previous studies have largely ignored the dynamic nature of the user's interests; that is, that the user's search intent and interests may change during the searching time (Raman et al., 2013; White et al., 2013; Harvey et al., 2013). For instance, in June, the query "US Open" is more likely to be targeting the golf tournament, while in September it is the prominent keyword for a tennis tournament. In this dissertation, we argue that a user profile should be *dynamically* constructed using the topics discussed in the user's *relevant documents*, where a "relevant document" might be a document that a user has clicked on or viewed for a certain length of time. That is, a *dynamic user profile* is needed, which can quickly capture the user's current search interests which change from time to time. Furthermore, the topics discussed in the relevant documents should be learned using an unsupervised method without using the human effort (e.g., a topic modelling algorithm proposed by Blei et al. (2003)).

For building user profiles, a personalised search system could either ask the users to provide their interests explicitly (Chirita et al., 2005; Gauch et al., 2007) or infer the interest implicitly from each user's search and interaction history (Stamou and Ntoulas, 2009; Bennett et al., 2012; Cheng et al., 2016; Yang et al., 2016). The former method meets some problems. First, users may not wish to spend extra time and effort to provide the explicit information to the search system. Second, it is often difficult for users to define their contextual preferences accurately. In this research, therefore, we explore how to build user profiles implicitly using each user's search history (e.g., the user's submitted queries and clicked documents) as well as the search history of other users who share the same or similar search interests with the current user.

Finally, we evaluate the effectiveness of the dynamic profile by utilising the dynamic user profile to implement the search personalisation process in the context of Web search. In this scenario, when a user submits a query to a Web search engine, the original document list is *re-*

[2] http://www.dmoz.org/

[3] http://www.dmoztools.net/

ranked. This re-ranking can ensure that the most relevant documents to the user are returned at the top of the ranked list (Bennett et al., 2012; White et al., 2013; Yang et al., 2016). In addition, we apply our research on dynamic user profiling to another search scenario: *Intranet search*, which (similar to Enterprise search) is different from general Web search (Hawking, 2010). Intranet search is *domain-specific* and built to satisfy the user's information need related to a specific domain (e.g., the university's document corpus). While Web search users are typically interested in getting *some* relevant documents, Intranet search users are typically looking for a em particular document (such as a book, timetable, etc.) (Hawking, 2010; Croft et al., 2009; Manning et al., 2008). Moreover, an Intranet may not be fully indexed and accessible by Web search engines. For example, Web search engines cannot access and index those Intranet documents which require authorised logins. Hence Web search engines might not satisfy the Intranet user's information needs. The searcher, therefore, may need to use a specific Intranet search engine to locate the relevant documents.

1.2 Research questions

The general research question that motivates our research in this dissertation is:

> *How can we build user profiles dynamically to improve the performance of search personalisation?*

To make this question clearer, various terms need to be defined more explicitly. First, the user profile is *dynamic* if it can capture the way that a user's search interests dynamically change during the user's searching time. Second, *performance* can be thought of as the quality of search results returned to a searcher when she submits a query to a search engine. The best possible quality means that all relevant documents are ranked in the top of the result list. In this case, the user does not need to spend extra time skimming irrelevant documents or scrolling down with the mouse. In Chapter 2 (Literature Review), we will see that individual components for handling this research question already exist. However, two key aspects, including grouping users with shared interests *dynamically* to enrich the current user profile and building *temporal* user profiles using an unsupervised topic modelling method have not yet been investigated. The main purpose of this research is to close these gaps, contributing new dynamic user profiling methodology for the field of search personalisation.

We start our investigation by focusing on how to build a user profile using topics discussed in the user's relevant documents. We address the problems associated with the ontology-based method (Bennett et al., 2012; White et al., 2013), and answer the following question:

> **RQ 1** *How can we build a user profile which represents the user's topical search interests for search personalisation?*

In answering this question, we start employing an unsupervised topic modelling method (i.e., Latent Dirichlet Allocation (Blei et al., 2003)) to extract *latent topics* automatically from the user's relevant documents.[4] We then utilise the extracted topics from the relevant documents of each user to build that user's profile.

Recent studies (Dou et al., 2007; White et al., 2013) have indicated that a user profile could be enriched by using data from groups of users who share common interests with that user. Whilst being successful in improving the performance of Web search engines (e.g., Bing), these methods statically group users with shared interests using some predetermined criteria such as: common clicked documents (Dou et al., 2007) or search locations (White et al., 2013); they neglect the fact that users in a group may have different interests in different topics with respect to the input query. We argue that groups with shared interests should be dynamically constructed in response to the user's input query, which leads to the following question:

> **RQ 2** *How can we dynamically group users who share common interests for search personalisation?*

We answer this question by proposing a model for query-dependent user grouping. We use the user's input query as an indicator to group other users who share common interests with

[4] The only observable variables in the topic modelling method are the words or terms in the document collection. Other variables, such as the topics, are latent variables (Blei et al., 2003). Therefore, the extracted topics are referred to as the "latent topics".

the current user. Therefore, with different input queries, different groups of users with shared interests are formed.

Although user profiling has been extensively studied for search personalisation, little attention has been paid to the *temporal* aspects of the profiles. These reflect an important type of context, specifically, with more interactions with the search system, the user's interests may change over the search time. New topics of interest may gradually emerge while interest in some of existing topics may fade. Therefore, the profiling method should be able to build user profiles dynamically so as to capture the user's changing interests over time.

We argue that the more recent relevant documents express more about the user's *current* interests than the distant ones. We then propose a temporal user profiling methodology to capture the user's changing interests over time, and address the following question:

RQ 3 *How can we build temporal user profiles for search personalisation?*

For each user, we utilise the temporal user profiling method to build three temporal user profiles using the user's search history in different time intervals: in the whole search history, in the current searching day and the current search session. As we will see in Chapter 5, temporal user profiles help to improve the search personalisation significantly.

Next, we turn to the application of temporal user profiles to model temporal search tasks. Recent research has shown that the concept of a search *task* is better for considering how users attempt to address their *atomic information needs*[5] than the concept of a search *session* (Li et al., 2016). Moreover, mining and modelling search tasks help improve the performance of Web search personalisation (White et al., 2013; Wang et al., 2014). However, the dynamic nature of many search tasks (for example, that search tasks can be generalised or specified during the searching time) has been largely ignored in previous search task modelling studies. To handle these problems, we propose to utilise the temporal profiling methodology to model search tasks. We address the following question:

RQ 4 *How can we model search tasks for search personalisation?*

[5] An "atomic information need" expresses a single/particular information need of the user, such as looking to buy a ticket for a Premier League match. A search session may involve a number of different search tasks.

We answer the previous four research questions using the dynamic user profiling methodology mainly in the context of Web search. Finally, we utilise the dynamic user profiling methodology in another search scenario, that of query suggestion in Intranet search. Existing Intranet search approaches appear to follow a "one size fits all" strategy. That is, different users who submit the same query receive the same query suggestion list (Adeyanju et al., 2012; Albakour et al., 2011; Hawking, 2010).

However, even with domain-specific search engines developed in Intranet search, the same query may reflect different topics of interest, and users who have submitted the same query may have different search intentions. For example, a sociology student submitting the query "lecture notes" is likely to be more interested in sociology classes than maths classes. Moreover, users' interests and search intentions may be dynamically evolving depending on their interactions with the system (e.g., clicks on documents), and when the interactions are made during a search session (Bennett et al., 2012). We address these problems by employing the dynamic user profiling methodology to build two temporal user profiles. The first is a *click user profile* based on the documents that a user actually clicks on. The second is a *query user profile* based on the user's query modification history within the search session. We address the following question:

RQ 5 *How can we personalise query suggestion for Intranet search?*

In the context of the Intranet search, we assume that a user normally handles only one search task per search session. This assumption will be verified in Chapter 7 with a detailed data analysis in Appendix A. Therefore, we will build the two temporal user profiles using the user's search interactions (i.e., submitted queries and clicked documents) in the user's current search session.

After addressing the five research questions, we need to answer the question of whether the dynamic user profile helps to improve the *performance* of search personalisation. To answer this question, we adapt the dynamic profiles to handle two search personalisation tasks:

1. Personalised search results from the Bing search engine[6] for the research questions: **RQ 1** and **RQ 2** in Chapter 4, **RQ 3** in Chapter 5, and **RQ 4** in Chapter 6.

[6] http://www.bing.com/

2. Personalised query suggestions from the Essex Intranet search engine[7,8] for **RQ 5**

For both tasks, we can think of the personalisation tasks as re-ranking tasks, in which we use the user profiles to re-rank the original list (i.e., a search result list or a query suggestion list) returned by a basis ranker (e.g., the default ranker of Bing search engine). Specifically, we use the query logs of a search engine user to build that user's dynamic profile or profiles. In the first task, given an input query, the search system will return a list of the top n documents most relevant to the query. For each query, we adapt the dynamic user profile to re-rank the search result list returned by the Bing search engine. In the second task, given an input query, the search system will return a list of the top m suggested queries. For each query, we use the dynamic user profiles to re-rank the suggestion list returned by the Essex Intranet search engine. After re-ranking, we will achieve a higher search performance if more relevant documents to the queries are promoted to higher ranks (Bennett et al., 2012; White et al., 2013; Yang et al., 2016).

Thesis Aims: By investigating the questions above, the aims of this research are as follows:

- To develop effective models for constructing user profiles which can capture the user's dynamic search interests.
- Through this development, to investigate effective ways to apply the user profile to improve the performance of search personalisation in different scenarios. We apply to Web search and Intranet search, and to search result personalisation and personalised query suggestion.

In each of the research chapters (Chapters 4 - 7) we address the research question listed above. The answers are given in the conclusions of each chapter and are summarised in Chapter 8 of this dissertation. In the next sections, we summarise the contributions of this research, and describe the thesis outline and the origins of the material.

[7] That is, a search engine installed at the Website of the University of Essex

[8] http://search.essex.ac.uk/s/search.html

1.3 Contributions and Outline of the Thesis

1.3.1 Main contributions

We summarise the main contributions of this thesis as follows:

- **An approach to building topic-based user profiles.** A topic-based user profile represents topical search interests of that user, extracted from the user's relevant documents (e.g., the documents clicked by the user). To overcome the limitations of using a manually generated ontology, we propose to use an unsupervised topic modelling method to automatically extract topics from the user's relevant documents. Having obtained these topics, we propose a new approach to building user profiles using the extracted topics. The experimental results on a Bing query log collection demonstrate that our approach helps to improve the ranking quality of Bing's original search results.

- **An approach to dynamic grouping formation for search personalisation.** Previous research on search personalisation has shown that the performance of Web search engines can be improved by enriching a user's personal profile with information about other users with shared interests. In the existing approaches, groups of similar users are often determined statically, such as based on the common documents that users clicked on. However, these static group formation methods neglect the fact that users in a static group may have different interests in different topics. We, on the other hand, argue that common interest groups should be constructed dynamically in response to the user's input query. We propose an approach to dynamic grouping formation for search personalisation. Specifically, a user profile is enriched with information about other users dynamically grouped with respect to an input query. The experimental results on a Bing query log demonstrate that our approach significantly improves the ranking quality of Bing's original search results and achieves a better performance than the static grouping method.

- **An approach to building temporal user profiles for search personalisation.** We propose three temporal latent topic profiles for each user using the relevant documents with different time scales in the user's search history. We name the profiles as the *session* *profile*, the *daily profile* and the *long-term profile*, as they are built from the latent topics automatically learned from the documents within a search session, a day and a whole history, respectively. We then utilise the profiles to re-rank search results returned by the Bing search engine. Our experimental results demonstrate that our temporal profiles can significantly improve the ranking quality. The results further show a promising effect of temporal features in conjunction with different query types and query positions in a search session.

- **An approach to modelling search tasks using the temporal user profiling methodology for search personalisation.** Recent research has shown that mining and modelling search tasks can help to improve the performance of search personalisation. However, the previous studies largely ignored the dynamic nature of the search task. That is, with the change of time, the search intent and user interests may also change. We address this problem by modelling search tasks using the temporal user profiling methodology and latent topics which are automatically extracted from the documents considered relevant to the search task. We then utilise the modelled search task to re-rank search results returned by the Bing search engine. Empirical evaluation shows our proposed approach to have a promising performance.

- **An approach to personalised query suggestion on Intranet search with the temporal session user profiles.** Recent research has shown the usefulness of using collective user interaction data (e.g., query logs) to recommend query modification suggestions for Intranet search. However, most of the query suggestion approaches for Intranet search follow a "one size fits all" strategy, whereby different users who submit the identical query would get the same query suggestion list. This is problematic, as even with the same query, different users may have different topics of interest, which may change over time in response to the user's interaction with the system. Based on our dynamic user profiling methods, we address the problem by proposing a personalised query suggestion framework for Intranet search. For each search session, we construct two temporal user profiles: a *click user profile* which uses the user's clicked documents and a *query user* *profile* which uses the user's submitted queries. We then use these two profiles to re-rank the non-personalised query suggestion list returned by a state of the art query suggestion method for Intranet search. Experimental results on a large-scale query log show that our personalised framework significantly improves the quality of suggested queries.

1.3.2 Thesis outline

This section gives an overview of the content of each chapter in this thesis. Apart from the Introduction chapter, there are two chapters to summarise the main work related to user profiling for search personalisation and the main methodology to evaluate the performance of search personalisation. Following that, there are four chapters detailing our contributions. The conclusion chapter highlights our research findings.

- **Chapter 2: Literature Review** In this chapter, we review previous work on search personalisation, user modelling and evaluation approaches. We first focus on describing personalisation in general and search personalisation in particular. We then examine some general approaches to the user profile construction which is one of the most important tasks in search personalisation. After that, we describe search personalisation strategies. Finally, we review evaluation approaches for personalisation strategies.

- **Chapter 3: Experimental Methodology** In this chapter, we show some problems with the publicly available datasets for search personalisation. We then detail the alternative datasets for experiments and introduce the experimental setup that forms the basis of our empirical evaluations. Also, we describe some well-known evaluation metrics in IR as well as statistical significance tests for search personalisation ranking.

- **Chapter 4: Dynamic Group Formation for Search Personalisation** In this chapter, we propose an approach to building a single user profile which represents a user's topical interests. We then propose a framework for search personalisation which uses dynamic grouping method to enrich a user profile. For a user, the profile is constructed and dynamically enriched with information from other users whose interests are similar to the current user given a query. The experiments on re-ranking the search results returned by the Bing search engine demonstrate that our framework significantly improves the ranking quality.

- **Chapter 5: Temporal User Profiles for Search Personalisation** In this chapter, we present a study on the temporal aspects of building user profiles with latent topics learned from the documents. For each user, we use relevant documents at different time scales (i.e., the whole search history, a day and a search session) to build long-term, daily, and session profiles respectively. Each user profile is represented as a distribution over latent topics from which we extract the features and combine them with non-personalised features to learn a ranking function using a learning to rank approach. We performed a set of experiments to study the effectiveness of the temporal latent topic-based profiles.

- **Chapter 6: Modelling Search Tasks for Search Personalisation** In this chapter, we propose an approach to modelling search tasks using the temporal user profiling methodology (detailed in Chapter 5). Each search task is represented as a distribution over the topics from which we extract the personalised feature and combine it with non-personalised features to learn a ranking function using a learning to rank approach. Experiments on re-ranking search results returned by the Bing search engine show that the ranking quality is improved significantly.

- **Chapter 7: Personalised Query Suggestion for Intranet Search** In this chapter, we build two session-based temporal user profiles: a query user profile using the queries submitted by the user, and a click user profile using the documents clicked by the user. We then extract the personalised features using the two profiles and combine them with non-personalised features to learn a ranking model using a learning to rank approach. We then use the ranking model to re-rank the query suggestion list returned by a baseline well-performed query suggestion approach for Intranet search. Experiments on the query logs collected from the University of Essex's Intranet search engine show that the personalisation significantly improves the performance of query suggestion.

- **Chapter 8: Conclusion and Future Works** We highlight our findings as well as discuss possible future research directions.

1.3.3 Origins

The following publications form the basis of the chapters in this thesis:

- **Chapter 4** is based on Vu et al. (2014): "Improving Search Personalisation with Dynamic Group Formation", in *Proceedings of the 37th international ACM SIGIR conference on research and development in information retrieval, **SIGIR**, 2014.*
- **Chapter 5** is based on Vu et al. (2015a): "Temporal latent topic user profiles for search personalisation", in *Proceedings of the 37th European conference on IR Research, **ECIR**, 2015.*
- **Chapter 6** is based on Vu et al. (2015b): "Modelling Time-aware Search Tasks for Search Personalisation", in *Proceedings of the 24th International Conference on World Wide Web, **WWW**, 2015.*
- **Chapter 7** is based on Vu et al. (2017a): "Personalised Query Suggestion for Intranet Search with Temporal User Profiles", in *Proceedings of the ACM SIGIR Conference on Human Information Interaction and Retrieval, **CHIIR**, 2017.*

Also, this thesis draws on insights and experiences from the following material:

- Vu et al. (2017b): "Search Personalization with Embeddings", in *Proceedings of the 39th European Conference on Information Retrieval, **ECIR**, 2017.*

Chapter 2

Literature Review

The ultimate goal of this dissertation is to improve search by dynamically constructing user profiles from the user's relevant topics of interest, which reflect the user's changing interests over time. To understand the effectiveness of the dynamic user profile, we apply it to handle two typical search personalisation tasks (i.e., search result re-ranking and query suggestion). This chapter aims to cover three main areas: personalisation, user modelling and evaluation approaches. We first focus on a general description of search personalisation. We then examine a number of general approaches for constructing a user profile, one of the most important tasks in search personalisation. We next describe search personalisation strategies. Finally, we review evaluation approaches for personalisation strategies.

2.1 Search personalisation

From a user's perspective, the search process of most Web search engines is simple (Teevan et al., 2010). A user enters a few words into a search box, and the search engine returns a (possibly very long) list of results. Although the user's search interaction for the above query-response mode is simple, people use search engines for many complex tasks (Teevan et al., 2010; White et al., 2013; Wang et al., 2013), such as planning business trips and handling assignments. The challenge of a search engine is to identify the user's information need from a typically simple, short query and return documents satisfying that need (Teevan et al., 2010). However, it is very common that the user's short queries are ambiguous (Dou et al., 2006; Teevan et al., 2010; Aktolga et al., 2013). Different users often have different search needs, even with the same input query. Furthermore, even the same users may have different expectations for the same search query at different searching times. For example, for the query "London", a traveller with a business purpose may need different information from one with a leisure purpose. To handle those problems, *personalised* search engines utilise the personal data of each user to tailor search results to that user. The results may depend not only on the input query but also on the user's search interest (such as the context of the query). Such personal data can be used to construct a *user profile*. Recently, *Search Personalisation* has attracted increasing attention from both academia (Teevan et al., 2009, 2010; Bennett et al., 2012; Hassan and White, 2013; Shokouhi et al., 2013; White et al., 2013; White and Awadallah, 2015; Yan et al., 2014; Ustinovskiy et al., 2015; Salehi et al., 2015; Yang et al., 2016; Lofgren et al., 2016) and industry (e.g., Google, Bing).

For example, Salehi et al. (2015) examined the Google search personalisation of academic search results, and how much difference personalisation made to students' learning and educational outcomes. They found that a majority of university students use Google search engine as an educational tool, with 83% of participants finding search engines an important or very important learning resource. They also found that the difference between personalised and non-personalised search results are quite significant. Almost half the links in the first page of the search results were completely different and common links between two lists mainly appeared in a different order. Likewise, Hannak et al. (2013) showed that on average, 11.7% of search results showed differences due to personalisation on Google, with the difference increasing for the less relevant documents. They also found that personalisation was more highly employed for queries related to political issues, news, and local businesses.

Recent years have witnessed the emergence of proactive systems, such as Google Now and Microsoft Cortana. In these systems, the user's relevant content is presented to that user based on her context (e.g., search interactions, locations, etc.) without a query. Shokouhi and Guo (2015) studied user interactions with information cards. They then proposed a supervised model to re-rank the proactive cards based on the user's context and history. They showed that using the user's reactive search history (i.e., the user's search interactions with a query-based search engine) could significantly improve the card ranking.

Central to the above search personalisation methods are *user profiling*, where user profiling is the process of building a user profile which represents the user's current search interest and *personalisation strategies*, where a personalisation strategy is a process to personalise (e.g., re-rank) the responses returned by a query using the user profile. Recent research has shown that the user profile is crucial to effective personalisation, and the performance of search personalisation depends on the *richness* of a user profile (Teevan et al., 2005, 2009). In the next section, we review the current research on user profiling, and in Section 2.3, we describe typical personalisation strategies.

2.2 User profiling

Typically, search personalisation systems pass through two stages to construct a user profile, which are *information gathering* and *information representing*, respectively (Ghorab et al., 2012). While the former aims to collect personal information from each user, the latter aims to represent the user profile using the information that was gathered about that user.

2.2.1 Information gathering

Overview

This section of the literature review focuses on information gathering, the first stage of user profiling for search personalisation. The information gathering stage comes from the fact that the available information for a search personalisation system determines the way we build a user profile. Based on the different approaches of obtaining the user information as well as the different sources and types of user information, we discuss two different aspects as follows:

- **Information gathering methods** the first aspect is to gather user information. The user information can be collected either implicitly without any extra effort from the user or explicitly where the user has to provide the information to the search system explicitly.

- **Source of information** the second aspect is the source of information. Information might be stored either at the server-side or client-side. Also, the privacy issues, as well as some problems (e.g., "filter bubble") with search personalisation, are highlighted.

Information gathering methods

As mentioned in Gauch et al. (2007), the personal information of each user can be collected either in an implicit or explicit manner. An implicit method silently gathers information about the user's search history and interactions with the system without any extra effort from the user. For implicit methods, the user's interests are inferred automatically from the collected information (e.g. the user's search history). For example, if a searcher has viewed many football-related documents, the search system might infer that the searcher is interested in the topic of football.

In an explicit method, on the other hand, the users themselves have to actively provide this information to the IR system in forms such as whether or not the results of a search are relevant, or whether or not they are interested a particular topic (Ghorab et al., 2012). Specifically, to build user profiles, IR systems might ask their users to explicitly select several topics (Chirita et al., 2005) or provide a set of positive documents for their topics of interests (Li and Zhong, 2006). Chirita et al. (2005) proposed an approach to building user profiles with users selecting topics which best fit their interests, for example, */Arts/Architecture/Experimental.* Similarly, the system proposed by Micarelli and Sciarrone (2004) obtained user information using a method of specifying initial criteria of "interest" and "non-interest" provided by the user. The user profile can also be constructed using positive or negative feedback about retrieval documents (Chen and Sycara, 1998; Asnicar and Tasso, 1997).

However, there are some drawbacks of the explicit methods. First, a user may not wish to spend extra time and effort to provide the information to the IR system. Secondly, she may sometimes input inconsistent or incorrect information. Moreover, it is very difficult for users to define their own contextual preferences accurately (Budzik and Hammond, 2000). Because of the dynamic nature of the user interest, explicit methods which depend on the user explicitly providing data are not able to quickly capture the user's interests.

Recently, research on search personalisation has witnessed the usage of external resources (e.g., Twitter) to build user profiles. Zhou et al. (2012) and Vallet et al. (2010) have applied social networks, social tagging applications, blogs, etc. to collect user profiles and so gained significant improvement in the performance of their IR systems. Younus et al. (2014) took into account various features of a user's Twitter account, including both the posts and the social network, to build the user profile for web search personalisation. After that, the Twitter-based user profile was used to *re-rank* the list of search results returned by the non-personalised search system. Experimental results on *CiteData* (Harpale et al., 2010) showed that the profile helps to improve the ranking quality significantly. Khodaei et al. (2015) used social information (e.g., like, share, recommend, friendships) to personalise the search results generated for each user. They mentioned that although the user's social actions sometimes seem irrelevant to the search results, these actions are really useful for personalisation. Likewise, Kacem et al. (2014) also used a user's social activities to represent the time-sensitive user profile as a vector of weighted terms.

However, using external resources (e.g., Twitter network) is very difficult for a search engine to exploit at scale because these resources are not available for every user. Furthermore, because the search process is anonymised, correctly mapping a searcher to her external resource, such as Twitter account, is a challenge, and involves serious privacy concerns.

Therefore, recent research has mainly focused on implicit methods in which information is gathered without any effort from the user. The users' search history, such as submitted queries and clicked results, is one of the main resources for building user profiles (Speretta and Gauch, 2005; Sieg et al., 2007; Dou et al., 2007; Bennett et al., 2012; Cai et al., 2017). Speretta and Gauch (2005) and Sieg et al. (2007) built user profiles by classifying each user's search history into concepts and topics and then re-ranking the list of results based on the similarity between a Web page and a user profile. Bennett et al. (2012) proposed an approach to building user profiles using the topics extracted from the user's satisfied (SAT) clicks. The SAT click is implicitly collected from the user's search history, where the SAT criterion is that the user remains on the clicked document for at least 30 seconds or the clicked document is the final one viewed in a search session (Fox et al., 2005). Instead of using the content of input queries and clicked documents, the user's search behaviour, such as his dwell time (viewing-time) on the clicked documents (Cai et al., 2017) and the cursor movement (Guo and Agichtein, 2012), was used to personalise the web search. Guo and Agichtein (2012) proposed a model to capture the patterns of searchers' behaviours such as the cursor movement and scrolling on a landing page for estimating document relevance. The authors also indicated that the model significantly increased the effectiveness of both gathering the information of each user and re-ranking the results using the estimated relevance.

As an alternative to a user's search history, Teevan et al. (2005) used other information such as documents and emails the user has read and created, to build user profiles. Teevan et al. also mentioned that a successful personalisation algorithm relies on a rich user profile. Moreover, using data from a group of users who share common interests has been widely used to enrich user profiles (Dou et al., 2007; Teevan et al., 2009; Aktolga and Allan, 2011; White et al., 2013). However, these methods group users statically using common clicks (Dou et al., 2007) and locations (White et al., 2013). Despite being successful in improving search results, the static grouping methods neglect the fact that users in a group may have different interests with respect to different topics.

Some systems collect the user information using the combination of both explicit and implicit approaches. Google's Web search engine is an example of such a system. In addition to implicitly using the user information from search logs, the system also uses the explicit profile of the logged in user for search personalisation. Specifically, Hannak et al. (2013) developed a methodology for measuring personalisation in web search results and applying this method to measure the extent of personalisation on Google's Web search. They found that Google personalises the search results to users being logged in to Google using the user's explicit profile as well as the user's implicit information (e.g., the clicked documents). Tao et al. (2011) proposed a personalised model to build user profiles from both global analysis and local analysis. The global analysis uses existing global knowledge bases, such as WordNet[1], digital libraries, online categorisations, and Wikipedia to build a user profile. On the other hand, the local analysis builds a user profile with the user's local information or behaviour, such as her stored documents, browsed web pages, and composed/received emails. User profiles were defined as the interesting topics of a user's information need. The users first selected positive and negative subjects for a given topic. Then for the given topic, the user's feedback subjects are used to construct an ontology.

Source of information

The source of user information can be either client-side or server-side. On the client-side based search personalisation, all documents edited or viewed by users through some processes on client-side can be captured using models proposed by Dumais et al. (2003); Teevan et al. (2005); Buscher et al. (2008, 2009). Buscher et al. (2008) used feedback generated from eye-trackers on the segment level (e.g., for specific passages of a document) to expand the input query. The authors showed that this kind of feedback improves search accuracy. However, as eye-trackers with sufficient precision are too expensive, such feedback data will not be available in a common workplace in the near future. Buscher et al. (2009) described a method of using segment-level display time from mouse moving and scrolling as implicit feedback. By comparing this feedback data with the data from eye trackers, they found that both segment-level display time can be as valuable as eye-tracking-based feedback in both re-ranking of search results and query expansion. Joachims et al. (2005) analysed searchers' decision processes via gaze tracking and compared the implicit feedback from search result clicks against manual relevance judgements. They found that clicks are informative but biased, as users tend to favour results at higher rank positions. The relative result preferences inferred from clicks still better represent searchers' true preferences.

Mouse cursor activity, such as cursor hovering and scrolling was used by Huang et al. (2012) to extend searcher models. However, with this client-side method, users may feel unsafe installing such software plug-ins. Furthermore, if a user changes the computer used for search, there are problems with keeping the contexts consistent. Also, without any information about a user, the

[1] https://wordnet.princeton.edu/

server needs to send all or a large number of results to the client to enable local personalisation. This communication overhead could slow down the speed of searching, as well as potentially requiring a lot of mobile data, making the service unattractive for mobile devices. Moreover, it requires the service to put its personalisation algorithms on the client. This may not be desirable for the service provider as the algorithms are normally proprietary and the key component to compete with other server providers (Nitesh et al., 2015).

Alternatively, several research studies (Speretta and Gauch, 2005; Bennett et al., 2012; Shokouhi, 2013; White et al., 2013; White, 2014; Kong et al., 2015; Yang et al., 2016), and other commercial IR systems such as Google, Bing, Yandex maintain and process the history of a user's interaction with the system on the server side (e.g. query logs). The server-side based systems have some advantages. First, users do not have to install new software. Second, even if a user changes the computer or uses many other devices such as tablets, smartphones, etc., the system (e.g. Google) can still synchronise the searching history to build a consistent profile. Moreover, it is easier to record larger data sets since the system can maintain the search histories of millions of users. Therefore, there is much research using the query logs to build personalised models (Bennett et al., 2012; Shokouhi, 2013; White et al., 2013; White, 2014; Kong et al., 2015; Yang et al., 2016). Bennett et al. (2012) used a proprietary dataset comprising anonymised logs of users of Bing search engine from July to August 2011 to model user profiles. The logs contained a unique user identifier, a search session identifier, the query, the top-10 URLs returned by the search engine for the query, and clicks on the result list. However, as searchers may not be willing for their search history to be stored (Nitesh et al., 2015), the server-side systems meet privacy requirements. Kobsa et al. (2016) investigated how characteristics of the personalisation provider influence users' attitudes towards personalisation and their resulting disclosure behaviour (i.e., related to privacy issues). They suggested that users sometimes consider the unintended and unauthorised use of their personal information. However, when the users grant an application to access their personal data, they rather consider how well the application can satisfy their information needs.

To handle these problems on the server-side and client-side based systems, existing approaches use two techniques. First, personalisation is done by the server or the personalisation proxy and not by the client. Second, the client sends limited information about the user profile with its request. The key requirement of these systems is to obfuscate the user profiles before sending them out properly (e.g., generalisation or noise addition) (Nitesh et al., 2015). Although we will not discuss privacy issues in this dissertation, Nitesh et al. (2015) proposed some valuable techniques for respecting the user's anonymity in server-side solutions by encoding a user's profile in a compact and privacy-preserving way.

Gathering the user's search interaction from the server side meets a challenge of multiple users searching on the same machine. This is common, with 56% of machine identifiers comprising search activity from multiple searchers (White et al., 2014). To handle this problem, White and Awadallah (2015) proposed an attribution-based personalisation process where they targeted individual searchers on shared devices. The process contains three phases:

1. search activity attribution (i.e., assigning observed search activity to an individual searcher),
2. attribution of newly-observed search activity to the correct searcher, and
3. application of the user's interest model to personalisation.

Experimental results on a dataset provided under contract by the Internet analytic company *comScore*[2] showed that the performance of search personalisation is significantly enhanced over the original ranking. Singla et al. (2014) applied the models learned from *comScore* data to predict and assign a user's identifier to search actions from among a set of Bing search engine logs. To do that, they filtered the *comScore* search activity to the search engine from which they obtained search interaction logs so that they could directly apply the models trained from *comScore* logs to the search logs. Because *comScore* data is not publicly available, we will not be able to attempt to reproduce the proposed methods. Furthermore, this data type is also not available in our datasets (detailed in Chapter 3). Therefore, we will not address the problem of multiple users searching on the same machine in this research.

[2] http://www.comscore.com/

An interesting problem of search personalisation is the so-called *"filter bubble"*, in which personalisation filters out information that disagrees with user viewpoints (Pariser, 2011). In other words, the search personalisation process can tailor search results to the user's preferences, *irrespective of the truth* (White and Horvitz, 2015). White and Horvitz (2015) presented a valuable methodology to measure searchers' beliefs and confidence about the efficacy of treatment before, during, and after search episodes. They showed the influence of prior beliefs and confidence at the end of search sessions. For example, people accept factually incorrect or unsupported information because it reinforces a particular belief they hold (White, 2013, 2014). They then built predictive models to estimate post-search beliefs using sets of features about behaviour and content. They also described a way in which modelling search beliefs enables richer models for recommendation or personalisation. Other types of biases include "ranking position bias", "popularity bias", "caption attractiveness bias", "domain bias", which are different from the biased beliefs about task outcomes and roles of the search engines in reinforcing those beliefs (White, 2013, 2014; White and Horvitz, 2015).

Summary

This section reviewed a number of approaches in the literature to the information gathering stage. Table 2.1 shows a summary of these approaches together with some publications.

2.2.2 Information representing

Overview

This section focuses on the second stage of user profiling for search personalisation, that is information *representing*. How to represent a user profile is a key component in search personalisation. The user profile maintains the user's information at an individualised level, especially on the terms that represent user's search interests. The user interests could be long-term (Dou et al., 2007; Harvey et al., 2013; Teevan et al., 2005) or short-term (White et al., 2010, 2013). Moreover, a user profile can be represented by a term-based model (Nanas et al., 2003) or a topic-based model (Bennett et al., 2012). Regarding the different approaches to maintaining the

Table 2.1: Summary of the information gathering stage

Information gathering approach	**Source of information**	**Example publications**
Implicit	Server-side	Speretta and Gauch (2005); Bennett et al. (2012); White et al. (2013); Yang et al. (2016)
Implicit	Client-side	Teevan et al. (2005); Chirita et al. (2007); Huang et al. (2012)
Implicit & Explicit	Server-side	Tao et al. (2011); Hannak et al. (2013)
Explicit	Client-side	Chirita et al. (2005); Micarelli and Sciarrone (2004); Chen and Sycara (1998); Asnicar and Tasso (1997)

user's search interests, and to represent a user profile, in this section, we discuss two criteria:

User search interest maintenance This first criterion is to maintain the user's search interest. The user's search interests could be long-term or short-term.

User information representation This second criterion, which might be regarded as the most important criterion with respect to user profiling, is to represent the user information.

User search interest maintenance

Long-term interests, in the context of IR systems, represent stable interests that can be exhibited for a long (permanent) time in the user's search history (Ghorab et al., 2012; Bennett et al., 2012). For example, a computer science student might have a long-term interest in programming. The long-term interests have been shown to be helpful for improving the quality of search results (Dou et al., 2007; Bennett et al., 2012; Harvey et al., 2013). Dou et al. (2007) and Harvey et al. (2013) have proposed methods to model the user's long-term interests using all of that user's search history, and use those long-term interests to help enhance future searches. This is done by analysing the text of all the user's submitted queries and clicked results from the user's *entire* search history, and extracting frequent terms or topics from them. The frequent terms or topics are then used to adapt the future queries or search results so that documents which appear to be more relevant to the user are ranked and displayed at higher ranks. Alternatively, the frequent terms or topics can also be extracted from other personal data such as computer files and emails (Chirita et al., 2007; Teevan et al., 2005). Then terms or topics can be used for adapting future queries or re-ranking (re-order) lists of results (Harvey et al., 2013; Teevan et al., 2005).

Short-term interests, on the other hand, represent temporary interests that may be satisfied by searches over a relatively short search period (e.g., in one or some continuous search sessions) (Ghorab et al., 2012; Bennett et al., 2012). For example, our computer science student might be interested in the final score of that day's football match. Short-term interests might be obtained from the submitted queries, and the clicked documents in a search session and used to personalise the search within the session (Shen et al., 2005; Ruthven, 2003; White et al., 2010; Ustinovskiy and Serdyukov, 2013). When there is insufficient data about the current user, the search behaviour of other related users may be beneficial. White et al. (2013) proposed the model of users' on-task search behaviour. The authors described a method of mining the user's search history to find other users working on similar tasks to the current user. This behaviour can be used to identify Web pages and so improve the ranking of the search results.

Ustinovskiy and Serdyukov (2013) used the short-term browsing context[3] to personalise the initial queries of search sessions, as well as applying the technique to new users whose profile lacks both long- and short-term contextual information. Similarly, Kong et al. (2015) developed a model to predict user search intent using pre-search contexts, such as news articles, or emails that the user had only recently read. Experimental results on a large-scale dataset[4] claim that the model can successfully predict the user search intent with the pre-search context. However, these works meet a challenge as the user pre-search context and browsing logs are not always available: this can be a violation of user privacy, and users are often not willing to share their pre-search context with search engines. As obtaining the pre-search context and browsing logs

[3] The web pages visited by a user before formulating a query q in the same session are called *browsing context*

[4] The dataset is not publicly available.

at scale are not practical, we will not use this type of data for user profiling in this research.

Some research studies model both the users' long-term and short-term interests (Sugiyama et al., 2004; Bennett et al., 2012). For example, Sugiyama et al. (2004) proposed a personalised search approach based on both long-term and short-term interests. The system monitored the user's browsing activity at the client-side. Therefore, the system can capture both short-term and long-term of user interests from terms available in the browsed Web pages. In their system, short-term and long-term user interests are stored separately in different user models. Similarly, Bennett et al. (2012) studied the interaction between long-term and short-term interests and found that the long-term interests provided advantages at the start of a search session while short-term interests played a more important role in the extended search session. Moreover, the combination of short-term and long-term interactions outperformed using either alone. Eickhoff et al. (2013) proposed a model for personalising atypical web search sessions in which the users' information need is outside their regular areas of interest. They proposed a method to identify whether a search session is atypical or not. They then applied three types of user profile (the user's historical search record, the searches in the particular session and the aggregation of the two) proposed by Bennett et al. (2012) to personalise the results returned by Bing. They found that the short-term (session) profile achieved the largest improvement.

User profile representation

Because the user profile is intended to represent the user's search interests, it has to satisfy two significant criteria (Nannas, 2003):

1. The user might be interested in more than one topic in parallel. Even in the context of the user's short-term interest, if there is a single, general topic of interest, it may consist of related subtopics. Therefore, a user profile should be able to represent multiple topics of interest.

2. The user's search interests might change over time. That is, some topics may emerge or fade based on changes in the user's information needs. Therefore, we need to pay attention to the *dynamism* of user profiles to respond to the user's interests, which can change over

time.

There exist numerous techniques for developing user profiles: the most popular ones are *term-based* and *topic-based* approaches (Gauch et al., 2007; Ghorab et al., 2012).

For the term-based approach, a user's interests can be represented as feature vectors, which are vectors of terms and associated weights, where a "term" is a word, or a phrase mined from the user's search history (for example, the user's submitted queries or clicked documents). The weights might be determined using a term weighting scheme, such as *TF* (term-frequency), *TF.IDF* (term frequency multiplied by inverse document frequency), or *BM25* (a.k.a, Okapi *BM25*). A vector-based profile can then be constructed from one or more of these weighted vectors. For example, Shen et al. (2005) used only one vector to model a user's short-term interests. Sugiyama et al. (2004) modelled a user profile using two vectors, one for the user's short-term interests and one for the user's long-term interests. In a search context, a user is typically interested in many topics, in which the terms in two different topics might overlap. Using a single term-based vector to model the user's search interests may not be appropriate, as the vector can contain a wide variety of ungrouped or unclustered terms. In order to handle this obstacle, some research has represented the user interests using multiple vectors, with one vector for a topic of interest or cluster of interests (Mc Gowan, 2003). In Mc Gowan (2003), the authors grouped terms together in unlabelled clusters using unsupervised text clustering techniques.

A further drawback of the term-based method is that with the assumption of independence between terms, it is not able to model the term correlations. For example, "*black* dogs and *white* cats" is treated in the same way as "*white* dogs and *black* cats". To remedy this problem, Nanas et al. (2003) proposed a methodology to build a concept hierarchy representation of user profiles which capture the term dependence property (the order of terms in the text). In the first step, terms that do not relate to the underlying topic of interest are filtered. In the second step, the link and the weight between two terms are calculated using a "sliding window", i.e., a span of continuous words that slides through each document's text. In the final step, suitable thresholds are used to group terms into three different layers. However, the method is timeconsuming in creating user profiles because of the large size of vocabulary used across all the documents. Furthermore, the described method uses all the user specified/relevant documents and treating the documents equally to build that user profile. It neglects the fact that the more recently relevant documents tell us more about the user's current search interests.

To handle the problems of the term-based approach, such as the difficulty of representing a user's different topics of interests, various topic-based methods have recently been explored in personalised search (Chirita et al., 2005; Bennett et al., 2012; Sontag et al., 2012; Harvey et al., 2013; White et al., 2013). In these methods, the user's search interests can be represented as a probability distribution over topics, in which the weight assigned to each topic represents how interested the user is in that topic (Chirita et al., 2005; Bennett et al., 2012; Harvey et al., 2013). Typically, the topic-based approaches contain two steps. The first step is to extract the topics from the user's search history. The second step is to use the extracted topics to build that user's profile, in which the user's search interests are represented.

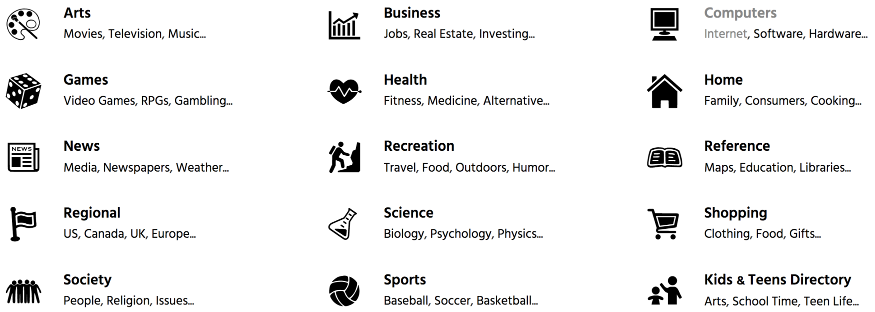
In the first step, topics are derived from the user's search history using some knowledge source, such as the Open Directory Project (ODP)[5,6] (Chirita et al., 2005; Bennett et al., 2012), or using a topic modelling technique (Sontag et al., 2012; Harvey et al., 2013). These can then be represented as a vector of weighted topics. Knowledge sources such as concept hierarchies (eg. ODP) or general knowledge repositories (eg. Wikipedia) are developed manually. The Open Directory Project (ODP) is one of the widely used knowledge models in search personalisation research (Chirita et al., 2005; Bennett et al., 2012; White et al., 2013; Yang et al., 2016). The reason is that it is the largest, most comprehensive human-generated directory of the Web[7] and has been built and maintained by hundreds of thousands of volunteer editors. ODP can be considered as a simple ontology with only one relation between its nodes. This kind of relation is an *"is-a"* or *"has-a"* relation between the parent category and a child category (e.g., *Science* → *Biology*) (Gauch et al., 2007). Figure 2.1 shows the two levels of categories in English. Figure 2.2 shows the topics derived from ODP for a Webpage. Specifically, Chirita

[5] http://www.dmoz.org

[6] http://www.dmoztools.net/

[7] http://www.dmoztools.net/docs/en/about.html

et al. (2005), Bennett et al. (2012) and White et al. (2013) mapped the user's search interests onto a set of topics, which were extracted from the Open Directory Project (ODP). Bennett et al. (2012) used a mechanism known as *decaying* to capture the dynamism of user profiles. To capture how interests change over time, Bennett et al. (2012) gave higher weights to more recently clicked documents. However, this ODP-based approach suffers from a limitation that many documents may not appear in the online categorisation scheme. Moreover, it requires costly manual effort to determine the correct categories for each document.



Arts
Movies, Television, Music...

Business
Jobs, Real Estate, Investing...

Computers
Internet, Software, Hardware...

Games
Video Games, RPGs, Gambling...

Health
Fitness, Medicine, Alternative...

Home
Family, Consumers, Cooking...

News
Media, Newspapers, Weather...

Recreation
Travel, Food, Outdoors, Humor...

Reference
Maps, Education, Libraries...

Regional
US, Canada, UK, Europe...

Science
Biology, Psychology, Physics...

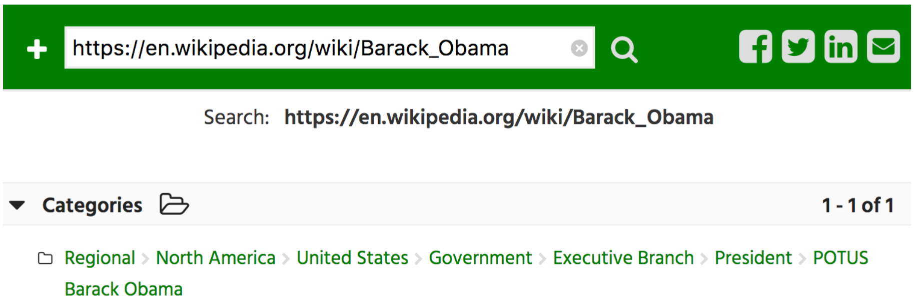
Shopping
Clothing, Food, Gifts...

Society
People, Religion, Issues...

Sports
Baseball, Soccer, Basketball...

Kids & Teens Directory
Arts, School Time, Teen Life...

Figure 2.1: ODP with two levels of categories



https://en.wikipedia.org/wiki/Barack_Obama

Search: **https://en.wikipedia.org/wiki/Barack_Obama**

Categories **1 - 1 of 1**

Regional > North America > United States > Government > Executive Branch > President > POTUS
Barack Obama

Figure 2.2: An example of topics derived from ODP

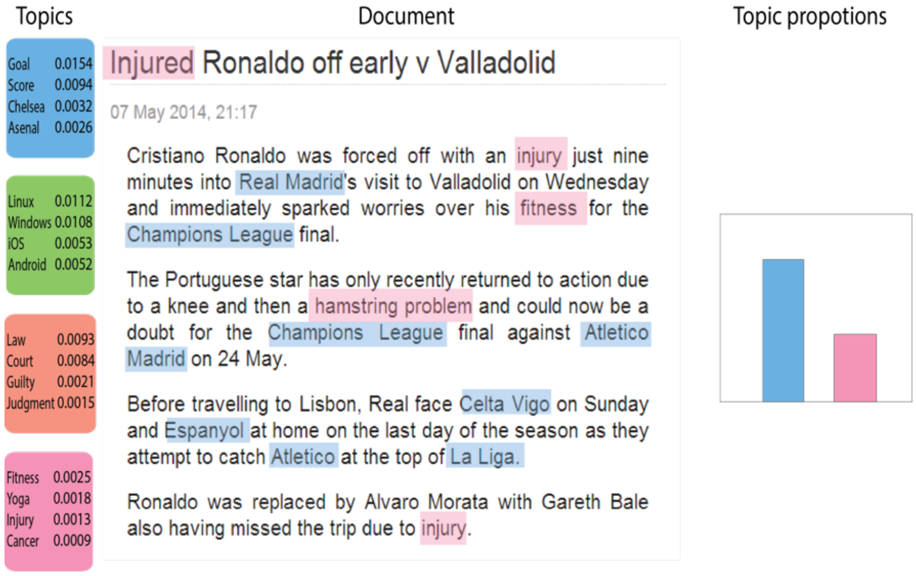
To overcome the problems with approaches based on knowledge sources, topic modelling techniques have been developed to automatically derive topics from documents (Blei et al., 2003). Human effort is not needed to extract topics from documents in these techniques. Harvey et al. (2013) applied an extension of Latent Dirichlet Allocation (Blei et al., 2003) to determine these topics. This means that the topic space is determined based purely on the user's search history (i.e., relevant documents extracted from query logs) and does not require human involvement to define the topics. However, the topic modelling techniques proposed by Harvey et al. (2013) and Sontag et al. (2012) did not pay attention to the dynamism of user profiles.

For the rest of the research, we extend these ideas of topic modelling to build user profiles dynamically which can quickly adjust to the user's changing search interests. Specifically, we employ Latent Dirichlet Allocation (LDA) (Blei et al., 2003) as the topic modelling technique to extract the topics from documents. Among many topic modelling techniques, LDA is fast and simple to implement, and performs with a good performance in the output topics (Phan et al., 2008; Blei et al., 2003; Blei, 2012). We introduce LDA briefly in the next section.

Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation is a topic modelling approach proposed by Blei et al. (2003). The intuition behind LDA is that documents describe multiple topics. For example, the article in Figure 2.3, entitled "Injured Ronaldo off early v Valladolid", is related to the injury of the football player Cristiano Ronaldo. In the figure, we have (manually) highlighted different words that are used in the article. Words about football, such as "Real Madrid" and "Champions League", are highlighted in blue; words about health, such as "injury" and "fitness", are highlighted in pink. If we highlight every word (except "to" or "on", which contain little topical content) in the article, one would see that this article blends the topics of health and football in different proportions.

LDA is a statistical model of document collections that tries to capture that intuition (Blei, 2012). Specifically, Blei (2012) defined a topic formally as a distribution over a fixed vocabulary. For example, the health topic has words related to health with high probability. Technically, the model assumes that the topics are generated first, before the documents. That is, to train the model, we have to specify the number of topics (denoted as n) as a prior parameter of the training process. After that, from the document collection as the input of LDA algorithm, the output will be n topics. Each topic is described as a distribution over a fixed vocabulary, and each word has a different proportion on the topic. For example, in Figure 2.4, the output topics



Injured Ronaldo off early v Valladolid

07 May 2014, 21:17

Cristiano Ronaldo was forced off with an injury just nine minutes into Real Madrid's visit to Valladolid on Wednesday and immediately sparked worries over his fitness for the Champions League final.

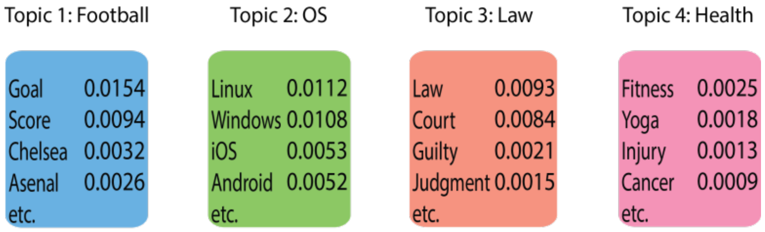
The Portuguese star has only recently returned to action due to a knee and then a hamstring problem and could now be a doubt for the Champions League final against Atletico Madrid on 24 May.

Before travelling to Lisbon, Real face Celta Vigo on Sunday and Espanyol at home on the last day of the season as they attempt to catch Atletico at the top of La Liga.

Ronaldo was replaced by Alvaro Morata with Gareth Bale also having missed the trip due to injury.

Figure 2.3: The distribution over the topic set of a football-related news

of LDA are Football, Health, Law and OS. The Football topic would give high proportions (i.e. probabilities) to words like “Score” and “Goal” while the Operating System topic gives high proportions (i.e. probabilities) to “Windows” and “Linux”.



Topic 1: Football		Topic 2: OS		Topic 3: Law		Topic 4: Health	
Goal	0.0154	Linux	0.0112	Law	0.0093	Fitness	0.0025
Score	0.0094	Windows	0.0108	Court	0.0084	Yoga	0.0018
Chelsea	0.0032	iOS	0.0053	Guilty	0.0021	Injury	0.0013
Asenal	0.0026	Android	0.0052	Judgment	0.0015	Cancer	0.0009
etc.		etc.		etc.		etc.	

Figure 2.4: The learned topic set of LDA

After using LDA to build the topics, each document in the collection is also described as a *distribution* over topics, in which topics appear in different proportions in each document. The proportion of each topic indicates how relevant the document is to the topic. Figure 2.3 shows the distribution over topics of the football-related news. The document has high proportions on the football topic and the health topic because the vocabulary of the document is relevant to both topics. However, the document is more relevant to football than to health because the number of football-related words (9) is greater than the number of health-related words (5).

Summary

This section reviewed a number of approaches in the literature to the information representing stage. Table 2.2 shows a summary of these approaches together with some publications.

Table 2.2: Summary of the information representing stage

User search interest maintenance	**User profile representation**	**Example publications**
Long-term	Term-based	Nanas et al. (2003); Mc Gowan (2003); Dou et al. (2007)
Short-term	Term-based	Shen et al. (2005)
Long-term & Short-term	Term-based	Sugiyama et al. (2004)
Long-term	Topic-based	Chirita et al. (2005); Teevan et al. (2005); Sontag et al. (2012); Harvey et al. (2013)
Short-term	Topic-based	White et al. (2013, 2010); Ustinovskiy and Serdyukov (2013)
Long-term & Short-term	Topic-based	Bennett et al. (2012); Eickhoff et al. (2013)

2.2.3 Summary and discussion

This section provided a literature review on user profiling. To build a user profile, we first need to gather the personal information of each user. We then use the personal data to build the user profile. In the first step, the user information can be gathered either explicitly (for example, via questionnaires) or implicitly (for example, by analysing query logs). The information can also be gathered either on the server-side or the client-side. In the second step, representing the user profile, either term-based or topic-based methods can be used.

Because explicit methods of gathering users' personal information suffer from the problems listed in section 2.2.1, in this dissertation we will propose an implicit method to extract the relevant document of each user from query logs. In our method, we use the query logs from a search engine (i.e. server-side) because of the advantages of the server-side methods as described in Section 2.2.1. With the pros and cons of the topic-based method analysed at the end of Section 2.2.2, we employ a topic-based method to represent user profiles which satisfy the two criteria described in Section 2.2.2, that is, multiple topics and the dynamism of user profiles. However, instead of using human-generated ontologies such as ODP, we will apply a topic modelling algorithm, LDA, to automatically learn latent topics from a document set.

Also, as described in Section 2.2.1, the performance of search personalisation depends on the richness of each user profile. A certain user profile can be enriched by the information of other users who share common interests with respect to the current user. In some research, the users with shared common interests are statically grouped (e.g. based on the number of common clicks between different users). However, the users in a static group can have different interests in different topics. Therefore, in Chapter 4, we focus on dynamically grouping users who share common interests with respect to different topics of interest. As a result, different input queries should return different groups of users who share common interests. Furthermore, to capture the dynamic nature of a user's search interest, in Chapter 5, we propose a means of building temporal user profiles, and study the interaction between the long- and short-term profiles.

2.3 Personalisation strategies

Having looked at methods for building user profiles, we can now focus on how to employ these user profiles for search personalisation. Typically, search personalisation can be performed by either adapting the search results (e.g. re-ranking the result list, so that more relevant results are displayed higher in the list) or adapting the input query (e.g. automatically or semi-automatically modifying, suggesting or auto-completing the query terms to obtain a better description of the user's information need).

2.3.1 Result re-ranking

Result re-ranking approaches are commonly used in search personalisation. The approach takes place after an original result list has been retrieved by the system. A further ranking step is performed to re-order the returned documents to meet some certain criterion (e.g. those documents judged more relevant to the user are displayed at a higher rank in the result list). Figure 2.5 shows an example of the result re-ranking approach. In the example, the user is interested in topic 1 and topic 2. After re-ranking, the documents related to the topics 1 and 2 are promoted to higher ranks. The re-ranking components proposed in the literature are often wrapped around a major commercial search engine (such as Google or Bing) and do not apply re-ranking on the full original result list returned by the search engine (Dou et al., 2007; Bennett et al., 2012; White et al., 2013; Cai et al., 2017). In particular, the re-ranking process only applies to the top n returned results.

The re-ranking approaches can be classified into two strategies. The first one consists of ranking *fusion* methods. The second one is to use a *learning-to-rank* (L2R) approach, which requires additional training of a ranking function. For both strategies, a personalised score which measures the similarity between the user profile and a returned document is calculated. After that, the personalised score is utilised to re-rank the original document list. In the fusion-based strategy, the personalised score is used directly to re-rank without the need of a training step (Dou et al., 2007; Li et al., 2014; Vu et al., 2014). The L2R strategy, on the other hand, uses the personalised score as a feature of an L2R framework (Bennett et al., 2012; White et al., 2013). In this case, we need a training dataset to train an L2R model; and then apply the model for the test dataset. The L2R strategy is widely used in previous research on search personalisation (Bennett et al., 2012; White et al., 2013), as well as winning Track 1 of the 2010 Yahoo! Learning to Rank Challenge.

Dou et al. (2007) proposed a re-ranking method for search results returned by the MSN search engine. In their first step, for each input query, they downloaded the top 50 search results from the MSN search engine. They then computed the personalised score for each Web page in the result list and re-ranked the search results based on the personal score to get a new ranked list.

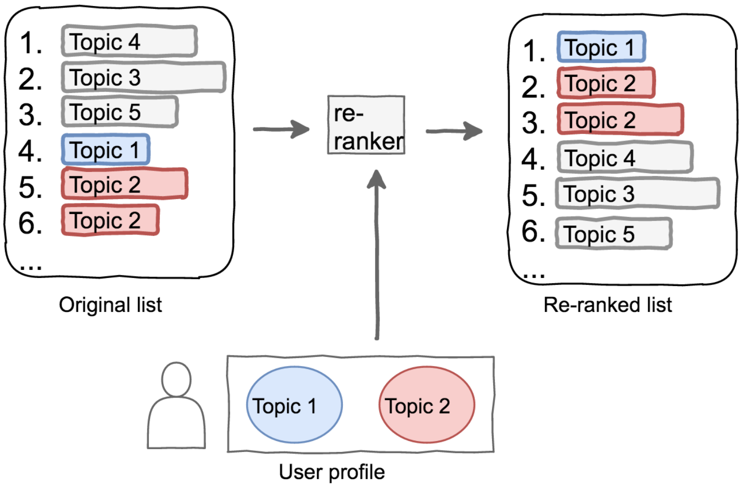


Figure 2.5: Search result re-ranking using the user profile

Finally, they combined the new rank with the original rank of each Web page using Borda's ranking fusion method (de Borda, 1781; Dwork et al., 2001). In particular, assuming that the original and the new rankings of a Web page are r_o and r_n respectively, the final ranking based on the ranking fusion method is $r = r_o + r_n$. The authors also mentioned that they used the rank-based ranking fusion method because they were unable to get the relevance scores between an input query and a Web page from the MSN search engine.

In a similar approach, Vu et al. (2014) proposed a re-ranking method which is wrapped around a collection of search logs from the Bing search engine. Vu et al. (2014) first downloaded the top 10 search results from Bing for each input query. Then the personal score p between a user profile and each Web page in the search results was calculated. After that, instead of using the ranking fusion method as in Dou et al. (2007), the personal score p was directly combined with the original ranking r_o to get a final score $f = \frac{p}{r_o}$. Finally, the scores are re-ranked to obtain a new ranked list. Vu et al. (2014) also showed that the personal score p only indicates how a certain user is interested in a Web page. Therefore, to obtain how the Web page is topically related to the input query, the original ranking r_o was used as an estimate.

In contrast to the previous approaches of using a simple ranking function, Bennett et al. (2012) applied a machine learning algorithm for re-ranking the top 10 results of each query returned by the Bing search engine. They first extracted the features of each document in the result list. The feature list was predefined by the authors and classified into three groups which are Query-Doc-User Features (e.g. click count overall user's history to the document), Query Features (e.g. Original ranking returned from Bing), and Query-History Features (e.g. Number of sessions containing this query). However, in their re-ranking method, a training set is required to train the ranking function before applying it to re-rank search results.

Social media data has also been used for result re-ranking in search personalisation. Vallet et al. (2010) studied how the ranking of search results can be improved if the users' social media information can be used in the re-ranking step. They combined the results retrieved from the Yahoo search engine with each user's tags extracted from del.icio.us. The returned results were then re-ranked based on the fact that both users and documents can be represented by associated sets of tags. An advantage of this approach is that it is independent of a specific search engine, and can be used in any search engine. However, not all web pages returned by search engines are tagged in the del.icio.us data set. As a result, this approach can suffer from sparsity challenges.

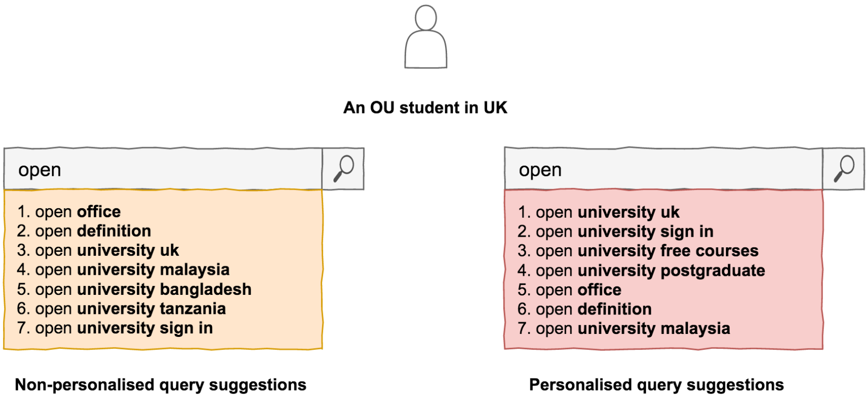
Also, while these approaches typically improve average performance of search results relative to simple baselines, they often ignore the fact that the new models can hurt performance on many queries (Wang et al., 2012). Dou et al. (2007) also pointed out that personalised search can lead to a significant improvement over common web search on some queries, but it has little effect on other queries. It even harms search accuracy under some situations (e.g. queries which showed less variation among individuals). Dou et al. used the notion of *query click entropy* to identify which types of query are best fit for search personalisation. For a given query, a smaller query click entropy value indicates that there is more agreement between users on clicking a small number of web pages. They found that if the click entropy is small, the personalisation process can even deteriorate the search performance. Large click entropy indicates a variety of Web pages clicked for the query. This might mean:

- A user has to select several pages to satisfy this query, which means the query is an informational query. Personalisation can help to filter the pages that are more relevant to users by making use of historical selections.

- Different users have different interests on this query, which means the query is an ambiguous query. In such cases, personalisation has a greater potential to provide different Web pages to different users.

2.3.2 Query adaptation

Query adaptation refers to the technique of expanding the terms of the input query with other terms to retrieve more relevant results (Manning et al., 2008). Figure 2.6 shows an example of the query suggestion lists with and without personalisation for the query “Open”. If the search system knows that the user is an OU student in the UK, it might suggest all the OU-related queries at the top of the suggestion list. Chirita et al. (2007) proposed approaches for query expansion, in which the expanding terms are extracted from each user’s Personal Information Repository (i.e. the personal collection of text documents, emails, cached Web pages, etc.). The authors conducted a set of experiments to determine the number of terms for expansion. They suggested that the decision should be dynamically based on each input query’s features such as query length, query clarity, etc.


An OU student in UK
Non-personalised query suggestions
Personalised query suggestions

open

1. open **office**
2. open **definition**
3. open **university uk**
4. open **university malaysia**
5. open **university bangladesh**
6. open **university tanzania**
7. open **university sign in**

open

1. open **university uk**
2. open **university sign in**
3. open **university free courses**
4. open **university postgraduate**
5. open **office**
6. open **definition**
7. open **university malaysia**

Figure 2.6: Query adaptation with/without personalisation

Zhou et al. (2012) proposed a system in which an individual user model is constructed based on the terms from the annotations and resources the user has marked on the del.icio.us tagging system. A statistical tag-topic model is defined to identify the most relevant terms in the user model to the user’s query and then those terms are used to expand the input query. In practice, each term in the user profile has an associated weighting score calculated based on its relationship with other terms in the profile and terms extracted from top-ranked documents. After calculation, terms with highest scores are chosen to expand the original query.

In some systems, query adaptation is performed by rewriting the whole query based on a set of rules maintained in the user model. For example, Koutrika and Ioannidis (2004) proposed a rule-based query rewriting process for personalising structured search across a database of movie information. The system substitutes the submitted query with multiple queries using a set of rules. These rules are based on the user's individual movie preferences. For example, suppose that a certain user is known to prefer movies of type "action". If that user enters a source query that requests a list of movies in a certain year, then the system can replace the source query with a query that seeks a movie list of the type "action" in that year.

In the study carried out by Yin et al. (2009), the authors used machine learning techniques to learn the similarities between queries in the logs, and exploit these similarities for query adaptation. They argued that traditional pseudo-relevance feedback has two drawbacks: (1) processing the full text of feedback documents (as opposed to processing only the snippets) obtained in the initial retrieval round is less effective; (2) not all feedback documents are guaranteed to be relevant, thus, some bad terms might be extracted from them (i.e. terms that may be harmful to retrieval effectiveness). The authors addressed these two issues by: (1) using the text of snippets instead of documents, which is further supported by the idea that, before clicking on results, users actually examine the result snippets in order to get a hint of how far a document is relevant to their information need; and (2) only selecting snippets that exceed a certain score threshold, where scores are assigned to snippets based on their rank and their similarity to the source query and similar target queries in the logs.

Also, query suggestion/auto-completion is one of the most prominent features of recent search engines both in general search engines (such as Web search engines) and for more specialised applications (such as Intranet search engines) (Cai and de Rijke, 2016b). Typically, for query suggestion, given a user's input query, one recommends a list of related queries to the given query (Adeyanju et al., 2012; Cai and de Rijke, 2016b; Carpineto and Romano, 2012; Kato et al., 2013). The main purpose of the query suggestion is to enable the user to formulate a query that better expresses his or her information need. Determining the best suggestion list for a query remains a challenge as this depends on the combination of multiple factors (e.g., the user's searching patterns and interests)(Adeyanju et al., 2012). However, recent research has shown that users prefer having suggestions regardless of their usefulness (Adeyanju et al., 2012; Ruthven, 2003). Typical query suggestion approaches use two main resources to produce the query suggestion list. These are the document collection (Xu and Croft, 1996; Carmel et al., 2002) and the user's own query logs (Adeyanju et al., 2012; Cai et al., 2014; Cai and de Rijke, 2016a; Shokouhi, 2013; Shokouhi and Guo, 2015), where the user's search interaction data such as submitted queries and clicked documents can be taken into account.

The approaches that utilise the document collection discover relationships between *terms* using either the whole document collection (global) (Dang and Croft, 2010) or those returned search results as relevant to a query (local) (Carmel et al., 2002). The top n terms related to the input query are then recommended for query refinement or automatically used for query expansion (Adeyanju et al., 2012). These approaches, therefore, return the same suggestions for all users. However, the intent of certain queries may vary significantly across users. For example, while "open university course" and "open office" might both be suggested to a searcher who has entered the initial search term "open". The former is more relevant to a student who wants to take some courses, and the latter is likely to be issued by a user seeking a piece of software (Figure 2.6). To handle this problem, query logs are a promising source for query suggestion since the terms used by a user in their search logs may indicate which interpretation of a query that user is likely to prefer (Adeyanju et al., 2012). Much recent research on query suggestion has used query logs to derive query suggestions (Adeyanju et al., 2012; Cai et al., 2014; Cai and de Rijke, 2016a; Shokouhi, 2013). From the query logs, the suggestion list can be generated either *locally* using only each user's search behaviour (Cai et al., 2014; Cai and de Rijke, 2016a; Shokouhi, 2013) or *globally* using the search behaviour of a group of similar users or all users (Adeyanju et al., 2012). For example, Shokouhi et al. (2013) proposed a supervised framework for personalising auto-completion ranking. In the framework, the completion suggestions are re-ranked based on each user's profile. A user's profile, in this case, is built from that user's age, gender, location and short-term and long-term search histories. All queries submitted in the current search session are used to define short-term history features. Otherwise, the entire search history of that user is considered to define long-term history features.

Most of these approaches are based on queries submitted to general Web search engines, such as AOL (Cai et al., 2014; Cai and de Rijke, 2016a) and Bing (Shokouhi, 2013), and so present a very broad view of the world. Little attention has been paid to the task of Intranet search. This is similar in some ways to enterprise search, but quite different from general Web search (Hawking, 2010). Recent research on query suggestion has shown the usefulness of query logs for the Intranet search (Adeyanju et al., 2012; Albakour et al., 2011). Adeyanju et al. (2012) proposed a method to adapt the concept of the *subsumption* hierarchy (Sanderson and Croft, 1999) with query reformulations observed in the query logs of the Intranet search. Although the proposed method successfully improved the quality of query suggestions, the suggestion list is derived globally using the query logs of all users. This technique, therefore, returns the same query suggestion list for all users since they are not dependent on each user's search behaviours.

2.3.3 Summary and discussion

In this section, we summarise the personalisation strategies described in the recent state-of-the-art research. Table 2.3 shows a summary of these approaches together with some publications. This section focuses mainly on different techniques for query adaptation and result re-ranking. In our research, we use both the result re-ranking and query adaptation techniques for our search personalisation model. For the query adaptation technique, we focus on the problem of the personalised query suggestion for Intranet search.

Table 2.3: Summary of personalisation strategies

Result re-ranking	**Query adaptation**
Dou et al. (2007); Bennett et al. (2012); Vu et al. (2014); White et al. (2013)	Chirita et al. (2007); Zhou et al. (2012); Yin et al. (2009); Shokouhi et al. (2013)

2.4 Evaluation approaches

In this section, we review approaches to the evaluation of the effectiveness and efficiency of personalised search. In the area of information retrieval research, a common quantitative evaluation approach is to compare the performance of a proposed approach to a baseline search system. The evaluation methods can be classified to either a controlled setting or a realistic setting (Ghorab et al., 2012). The former method evaluates a personalisation approach using a small number of users and tasks (Granka et al., 2004; Sugiyama et al., 2004; Teevan et al., 2005; Moshfeghi and Jose, 2013). The advantage of such setting is that it allows establishing control groups and conducts a richer evaluation of usability aspects. The latter approaches base their evaluation on a large amount of data from a realistic setting (e.g. a major search engine) (Dou et al., 2007; Bennett et al., 2012; Shokouhi et al., 2013).

For example, Teevan et al. (2005) used a controlled setting method to compare their personalised search system to a baseline system. They constructed a study of 15 participants to evaluate the top 50 web pages returned by the MSN search engine for approximately 10 self-selected queries. Relevance judgements were performed in a non-binary manner, where returned web pages were judged on a three-level scale: highly relevant, relevant, or not relevant to the query. Moreover, to avoid bias to the particular participants, the returned results were presented randomly. To measure the ranking quality, they used Discounted Cumulative Gain (DCG). DCG, which will be described in Chapter 3, is a measure that gives more weight to highly ranked documents and allows incorporation of different relevance levels by giving them different gain values. The results showed that the personalised system significantly outperformed the MSN search system.

A similarly controlled setting method was also used in Moshfeghi and Jose (2013) to show the effectiveness of their proposed model on identifying relevant documents by combining affective, physiological and behavioural features with dwell time. The dwell time of a user on a clicked document is the time that the user spent in viewing the document. The authors constructed a study on 24 subjects to carry several search tasks. The results indicated that the combination of the features and dwell time can significantly improve the relevant data prediction.

Also, a number of studies in the literature have conducted experiments on realistic data sets (e.g. query logs from a commercial search engine). For example, Dou et al. (2007) evaluated their personalisation models on a realistic experimental setting, using large-scale query logs from the MSN search engine for 12 days in August 2006. A random sample of 10,000 distinct users (identified by "Cookie GUID") from the users in the United States on 19th August 2006 was drawn. Information about these users, along with their click-through logs, are extracted as a dataset. The click-through logs dataset contains the search history of users, and also records the returned documents from the search engine with respect to a query. A record of which documents had been clicked by individual users and how long each user viewed the document were also stored. The author argued that because users are randomly sampled, this dataset could reflect the characteristics of the entire logs. The ground truth data set was constructed using an assumption that the clicked documents are the relevant documents of each user. The two metrics used by Dou et al. (2007) for retrieval evaluation were Ranking scoring and Average rank, which will be described in more detail in Chapter 3.

In Dou et al. (2007), the authors assumed that all clicked Web pages are relevant. However, this assumption may not be correct because the clicks with short dwell time ("quick backs") are unlikely to be relevant (Fox et al., 2005). To handle this problem, the experiments carried out by Bennett et al. (2012) were also conducted in a realistic setting. (Bennett et al., 2012) extracted a dataset containing 8 weeks of query logs from the Bing search engine from July to August 2011. In their research, instead of assuming that all clicked documents represent relevant data, they defined a satisfied result click using the SAT criteria proposed by Fox et al. (2005) as either a click followed by no further clicks for at least 30 seconds, or the last result clicked in the search session. Logs were collected from the Bing search engine where other personalisation supports were disabled, in order not to bias their results with other personalisation signals. The authors evaluated their personalisation models using Mean Average Precision (MAP), which is a single-valued metric that serves as an overall figure for directly comparing different retrieval systems (Manning et al., 2008). MAP is calculated as the average Precision at k values of all testing queries, which will be described in Chapter 3.

Discussion

A key challenge for in-lab settings are that they do not yield large datasets of search logs. Moreover, researchers need to spend more time in constructing controlled setting experiments. However, controlled settings do provide some advantages, such as the ability to experiments with different settings using the same group of participants. Also, evaluating personalisation approaches using realistic settings can handle the challenges of in-lab setting. However, the realistic dataset are usually accessible only through collaborations with search engine companies.

2.5 Conclusion

This chapter has reviewed research on search personalisation covering search personalisation (Section 2.1), user profiling (Section 2.2), personalisation strategies (Section 2.3) and evaluation methodologies (Section 2.4). The literature indicates that the performance of search personalisation depends heavily on the richness of the user profile. Moreover, the user profile should be built in a dynamic way to capture the user's search interest that changes over time. From the literature reviewed in this chapter, in this thesis, we aim to answer the following question:

> *How can we build user profiles dynamically to improve the performance of search personalisation?.*

To handle this question, two potential research directions might be employed. The first would be to dynamically leverage the information of users who share common interests to enrich a certain user profile (Chapter 4). The second one would be to build temporal user profiles (Chapters 5 - 7).

This chapter has reviewed the literature on user profiling for search personalisation and its corresponding approaches. In the next chapter, we introduce the experimental methodology including the datasets, the evaluation metrics, and the significance test used in our experiments.

Chapter 3

Experimental Methodology

Typically, search personalisation approaches deal with a *re-ranking* problem (Teevan et al., 2005; Dou et al., 2007; Teevan et al., 2009; Nanas et al., 2010; Bennett et al., 2012; Harvey et al., 2013; Liu, 2015; Yang et al., 2016; Cheng et al., 2016). As shown in Figure 3.1, search personalisation models focus on *re-ranking* the original ranked list that is initially returned by a basic ranker when a user submits a query. The ranked list is a list of either returned documents, suggested queries for the Web search personalisation task or personalised query suggestion task. In the flow shown in figure 3.1, *List 2*, the re-ranked list, contains the top items from the initial list, *List 1*. The order of those items in the re-ranked list is usually different from that in the initial list. Thus, we can think of search personalisation approaches as *re-rankers*. In our thesis, the dominant measures used for the re-ranking is dynamic user profiles. These are either dynamic user grouping-centred or temporal user profiling-related.

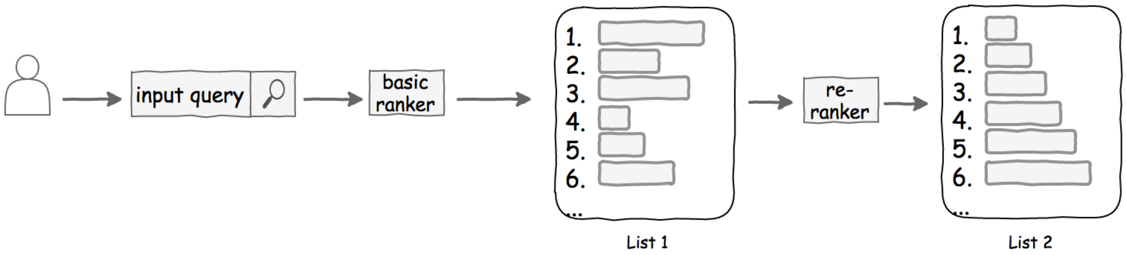


Figure 3.1: General flow of a search personalisation approach

3.1 Datasets

The availability of test collections plays a critical role in successful experimental research in IR. There are a number of academic communities providing test collections to evaluate the performance of retrieval algorithms. Among these, Text REtrieval Conference (TREC)[1] and the NII Test Collection for IR systems project (NTCIR)[2] are well known. Often these datasets are not suitable for search personalisation because they do not contain enough personal data about the users (such as their identifiers). The most suitable dataset for investigating search personalisation is the session track dataset[3] which is provided by TREC. The dataset contains information about a number of search sessions, and for each of these contains all of the previous queries, clicks, ranked results, and dwell times. However, because the user associated with a search session is not provided, we cannot identify whether two sessions belong to a single user or not. Thus, we cannot build the long-term profile of a user which expresses the user's long-term interest.

To handle this problem, we use query log dataset from Bing as an alternative. Each log entry consists of an anonymous *user identifier*, a submitted query, the top 10 returned URLs and clicked results along with the user's dwell time. The availability of user identifiers allows us to extract all search logs of each user so that we can build the long-term profile of a user using the user's search logs. In this thesis, we mainly use Bing's query logs to perform our experiments to study the effectiveness of dynamic features in building a user profile in Chapters 4 - 6. The experiments also allow us to study whether dynamic user profiles help to improve the performance of web search using the Bing search engine?

Also, in Chapter 7, we approach personalising query suggestions for Intranet search with dynamic user profiling. We use another query log collection from the Essex Intranet search engine[4,5]. Each log sample contains a session identifier, the event type (e.g., a query or a click), an auto-increment query id, the event content (e.g., query text, click URL), and the event

[1] http://trec.nist.gov

[2] http://research.nii.ac.jp/ntcir

[3] http://ir.cis.udel.edu/sessions/

[4] That is, a search engine installed on the Website of the University of Essex

[5] http://search.essex.ac.uk/s/search.html

time-stamp.

For our experiments in Chapters 4 - 6 we use two Bing datasets and in Chapter 7 we use the dataset from the Essex intranet. Basic statistics of the three query log collections are presented in Table 3.1[6]. The first Bing dataset contains queries sampled between 01 July 2012 and 15 July 2012 from 106 anonymous users. The second Bing dataset contains queries sampled in 4 weeks from 01 July 2012 to 28 July 2012 from 1166 anonymous users. We use the first (small) dataset in Chapter 4 and the second (bigger) dataset in Chapters 5, 6. The Essex dataset is a large-scale log collection from the Essex Intranet search engine over 2 years from 01 January 2012 to 31 December 2013.

Table 3.1: Basic statistics of the three query log collections. We note that the user information (i.e., user identifier) is not available in the Essex dataset.

Statistic	Bing_1	Bing_2	Essex
Language	English	English	English
Start Date	01-07-2012	01-07-2012	01-01-2012
End Date	15-07-2012	28-08-2012	31-12-2013
Days	15	28	730
Queries	17,947	520,010	1,416,929
Users	106	1,166	-
Sessions	4,724	94,972	735,804
Clicks	24,041	433,277	930,242

3.2 Evaluation metrics

Retrieval evaluation metrics play a crucial role in the evaluation of retrieval systems. Retrieval evaluation is the process of systematically associating quantitative measures with the results produced by an IR system in response to a set of user queries (Baeza-Yates and Ribeiro-Neto, 2011). Retrieval evaluation metrics should reflect the relevance of the results to the users. A

[6] A more detailed analysis of the data can be found in Chapters 4 - 7 and in Appendix A.

common approach to calculating evaluation metrics is to compare the results produced by the search system with results suggested by humans for the same set of queries. Here, the retrieval evaluation metrics concern the quality of the results, not the efficiency performance of the system, such as how fast queries are processed.

In this section, we detail the evaluation metrics that we use to evaluate the effectiveness of our dynamic user profiles in personalisation tasks. For a complete overview, we use the following evaluation metrics:

- Precision at rank k ($P@k$)
- Mean Average Precision (MAP)
- Mean Reciprocal Rank (MRR)
- Normalized Discounted Cumulative Gain ($nDCG@k$)
- Inverse Average Rank (IAR) (Dou et al., 2007; Li et al., 2014)
- Personalisation Gain (*P-Gain*) (Harvey et al., 2013)[7]

These are well known and widely used IR evaluation metrics (Baeza-Yates and Ribeiro-Neto, 2011; Manning et al., 2008). We describe those metrics in detail in the context of Web search personalisation [8] as follows:

Figure 3.2 shows an example of a returned document list before (left) and after (right) re-ranking. R means that the document is relevant (e.g., clicked). Each list contains the same set of documents, but in different orders.

[7] Note that, in Chapter 4, we only use Inverse Average Rank (IAR) and Personalisation Gain (*P-Gain*) because we follow the experimental setting for a *fusion* based re-ranking method from (Dou et al., 2007; Harvey et al., 2013) to compare our dynamic group formation to a static one (Dou et al., 2007). The other metrics are used extensively in Chapters 5 - 7 following the experimental setting used by Bennett et al. (2012) for a *learning-to-rank* based re-ranking method.

[8] In the context of personalised query suggestion; the metrics are calculated using the suggested query list instead of the document list as in the context of Web search personalisation

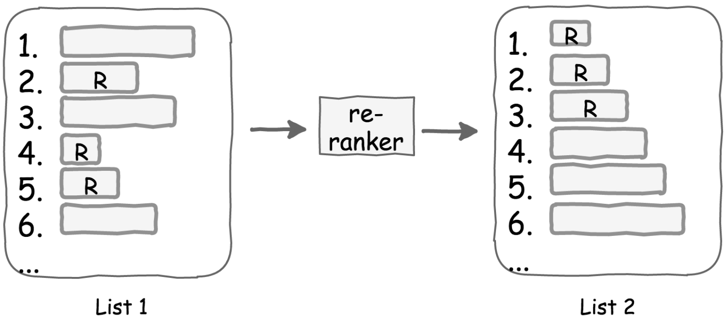


Figure 3.2: An example of a document list before and after re-ranking. R means that the document is relevant to the user.

3.2.1 Precision at rank k

The *precision at rank k* ($P@k$) metric gives the percentage of relevant documents within the top k returned documents. In web search related tasks this metric is widely considered important, as users usually only look at the top k returned documents of a ranked list. It is defined as:

$$P@k = \frac{\sum_{r=1}^{k} rel(r)}{k} \tag{3.1}$$

where $rel(r)$ is a binary function that indicates whether or not the document at rank r is relevant:

$$rel(r) = \begin{cases} 1, & \text{if } r \in R \\ 0, & \text{otherwise} \end{cases}$$

where R is the set of *relevant* documents for a given query (the set of documents marked as relevant in the training and test sets). In Figure 3.2, $P@5 = 3/5 = 0.6$ for both lists. However, if we consider only the first three results, *List 1* gives a much lower precision at rank 3 ($P@3 = 0.333$) than *List 2* ($P@3 = 1$). We optimise $P@k$ in a certain context in which we consider the relevance of top k ranked documents instead of the whole returned document list. One drawback of this metric is that we do not know which order of the k document list is better. For example, in Figure 3.2, $P@k$ is the same for both list 1 and list 2 if $k \geq 5$, but one can see that in most cases, *List 2* would be preferable to *List 1* as all the relevant documents are ranked nearer the top. Therefore, typically, this metric is only used to compare subsets of the returned lists (e.g., top 1, 3, 5 returned documents) but not all N items of the returned list. Furthermore, the metric cannot measure the *recall* of search results in the top k list (what proportion of the relevant documents).

3.2.2 Mean average precision

Mean average precision (*MAP*) is a widely used metric that measures both the precision and recall of search results. For each relevant document in the returned document list, we take the precision at the rank of that document. These are summed over the precision values and divided by the total number of relevant documents. This gives us the average precision (AP) for a query:

$$AP = \frac{\sum_{k=1}^{N} P@k \times rel(k)}{|R|}$$

where N is the number of returned documents ($N = 10$ in Bing's query logs and $N = 12$ in Essex's query logs). In Figure 3.2, we have

$$AP = \frac{0 \times 0 + 0.5 \times 1 + 0.3 \times 0 + 0.5 \times 1 + 0.6 \times 1 + 0.5 \times 0}{3} = 0.53$$

for *List 1* and

$$AP = \frac{1 \times 1 + 1 \times 1 + 1 \times 1 + 0.75 \times 0 + 0.6 \times 0 + 0.5 \times 0}{3} = 1$$

for *List 2*. Taking the mean of AP values over a set of test queries gives the mean average precision (MAP) for a system on that set of queries. In other words, the MAP score of a retrieval system can be obtained by averaging the AP values of all the test queries. Unlike $P@k$, the MAP metric can be used to compare the qualities of two *full* lists before and after re-ranking. We optimise the MAP metric in general search contexts in which relevant documents are expectedly ranked higher than irrelevant documents.

$$MAP = \frac{1}{|Q|} \sum_{i}^{Q} AP_i$$

where AP_i is the average precision of the query i. Q is the set of test queries and $|Q|$ is the size of Q.

3.2.3 Mean reciprocal rank

Mean reciprocal rank (*MRR*) is a metric that measures the ranking quality of the first relevant document in a returned document list. Given a query, we calculate the reciprocal of the rank (*RR*) at which the first relevant document was retrieved. This gives us the reciprocal rank (*RR*) for the query:

$$RR = \frac{1}{rank_r}$$

where $rank_r$ is the rank of the first relevant document. *RR* is 1 if the first document in the retrieved list is relevant, it is 0.5 if the first document in the list is irrelevant, but the second is relevant, and so on. In Figure 3.2, $RR = 0.5$ for *List 1* and $RR = 1$ for *List 2*. When we average *RR* values across a set of test queries, the measure is called the Mean Reciprocal Rank (*MRR*):

$$MRR = \frac{1}{|Q|}\sum_{i}^{Q} RR_i$$

where RR_i is the reciprocal rank of the query i. Q is the set of test queries and $|Q|$ is the size of Q. We might want to optimise *MRR* in cases where we only care about the first relevant document, such as the "I'm feeling lucky" function of the Google search engine, in which, instead of viewing the result list, you directly go to the first returned result. *MRR* is equivalent to *MAP* when the number of relevant documents in the returned list for calculating *MAP* is only one.

3.2.4 Normalised discounted cumulative gain at rank k

Normalized discounted cumulative gain (*nDCG@k*), like precision at k, is evaluated over the top k returned results. This metric can take into account both binary (e.g., Bad, Good) and non-binary notion of relevance (e.g., Bad, Fair, Good, Excellent, etc.) at rank k. Given a ranked result set of documents S and an ideal ordering of the same set of documents O, the discounted cumulative gain (*DCG*) at a particular rank threshold k is defined as:

$$DCG(S,k) = \sum_{j=1}^{k} \frac{2^{r(j)} - 1}{\log(1+j)}$$

where $r(j)$ is the judgement function (for example, $(0 = \mathit{Bad}, 1 = \mathit{Fair}, 2 = \mathit{Good}, 3 = \mathit{Excellent})$ at rank j in set S. For example, in Figure 3.3, $r(1) = 0$, $r(2) = 1$, etc., and

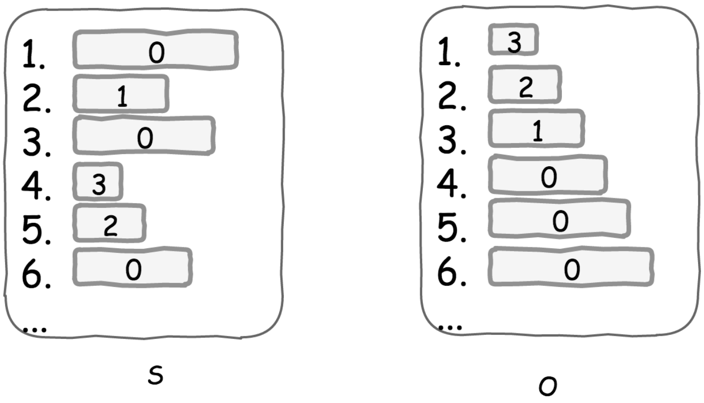


Figure 3.3: An example of a non-binary judgement function of relevance (0 = Bad, 1 = Fair, 2 = Good, 3 = Excellent). S is a ranked result set of documents. O is an ideal ordering of the same set of documents.

$DCG(S, 5) = 1/\log(3) + 7/\log(5) + 3/\log(6)$ for S. The ideally ordered set O contains all documents rated for the given query sorted descending by the judgement value ($DCG(O, 5) = 7/\log(2)+3/\log(3)+1/\log(4)$ for O). Then the normalized discounted cumulative gain ($nDCG@k$) at a particular rank threshold k is defined as:

$$nDCG(S,k) = \frac{DCG(S,k)}{DCG(O,k)}$$

$nDCG(S, k)$ discounts the contribution of a document to the score as its rank increases. Higher $nDCG(S, k)$ values correspond to better correlation with human judgements. $nDCG(S, k)$ value at rank threshold k when the set S is clear from the context is often written as $nDCG@k$. For example,

$$nDCG@5 = \frac{1/\log(3) + 7/\log(5) + 3/\log(6)}{7/\log(2) + 3/\log(3) + 1/\log(4)} = 0.51$$

for S (see Figure 3.3).

In addition to the above metrics, in Chapter 4, we propose to use two further metrics. These are:

3.2.5 Inverse Average Rank

The average rank (AR) over a set of test queries Q is defined as (Dou et al., 2007):

$$AR = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{|R_q|} \sum_{p \in R_q} rank_p \tag{3.2}$$

where Q is a set of test queries. $|Q|$ is the size of Q. R_q is a set of relevant web pages for a test query q; $rank_p$ is the rank of a page p. A smaller AR indicates a better overall quality of the ranked results, in which the relevant documents are returned at top of the result list(Dou et al., 2007). For example, if the test query set contains only one query as in Figure 3.2. $AR = 1/3 \times (2 + 4 + 5) = 3.67$ for the *list 1* and $AR = 1/3 \times (1 + 2 + 3) = 2.00$ for the *list 2*. For ease of use, in our thesis, we define an *Inverse Average Rank* (IAR) metric as:

$$IAR = \frac{1}{AR} \tag{3.3}$$

In the case shown in Figure 3.2, IARs are 0.273 and 0.5 for *List 1* and *List 2*, respectively. The higher IAR score indicates the better ranking quality. IAR has a similar effect to MAP as both of them take into account of the rank of all relevant documents in the result list.

3.2.6 Personalisation Gain

Personalisation Gain (P-Gain) shows how stably the personalisation improves the ranking performance over a baseline across all test queries (Harvey et al., 2013). The metric compares the number of relevant web pages promoted to a higher rank against the number of relevant pages obtaining lower ranking after using the personalisation algorithm. A higher positive P-Gain value indicates a better overall robustness of a personalisation algorithm in term of improving performance over the baseline.

$$P\text{-}Gain = \frac{\#better - \#worse}{\#better + \#worse} \tag{3.4}$$

where $\#better$ is the number of relevant web pages promoted to a higher rank after re-ranking. $\#worse$ is the number of relevant web pages promoted to a lower rank. In Figure 3.2, $\#better$ and $\#worse$ are 2 and 1, respectively. Thus, $P\text{-}Gain$ is $\frac{2-1}{2+1} = 0.33$.

3.3 Significance test

Researchers in IR commonly use three main statistical significance tests, those being Student's paired t-test (Gosset, 1904), the Wilcoxon signed rank test (Wilcoxon, 1945) and the sign test (Karas and Savage, 1967). The reasons for using significance testing is to determine whether observed differences are significant or just the matter of chance. In Chapters 4 - 7, we propose dynamic user profiling approaches which should improve the performances on the personalisation tasks in the chapters. To test whether our proposed approaches do show improvements, we compare their scores to baseline scores using the evaluation metrics described in the previous section. When comparing two runs, we want to test for significant differences between them. To this end, we usually use a two-tailed paired t-test. Gosset (1904) shows that in practice there is no difference between the t-test and the randomisation test.

For the paired t-test to be valid, the differences between two runs need to be approximately normally distributed (Lumley et al., 2002). Using the Central Limit Theorem[9] which states that the distribution of the sum (or mean or difference) of a sufficiently large number ($N > 100$) of independent, identically distributed variables will be approximately normal, regardless of the underlying distribution (Lumley et al., 2002). From Table 3.1, the numbers of queries are large ($N > 17,000$) for all three datasets. Therefore, following the Central Limit Theorem, the paired t-test is valid in our experiments. Figure 3.4 shows the histogram of the differences between average precision values of the Bing default ranker and after re-ranking using the long-term user profile (Chapter 5). We can see that the differences are approximately normally distributed. Moreover, the paired t-test has been widely used in search personalisation (Bennett et al., 2012; White et al., 2013; Yang et al., 2016).

The statistical significance of observed differences between the performance of two runs for significance levels $\alpha = 0.01$ and $\alpha = 0.05$ is shown in the experimental results of this thesis when necessary and appropriate, the former being stronger than the latter.

[9] https://en.wikipedia.org/wiki/Central_limit_theorem

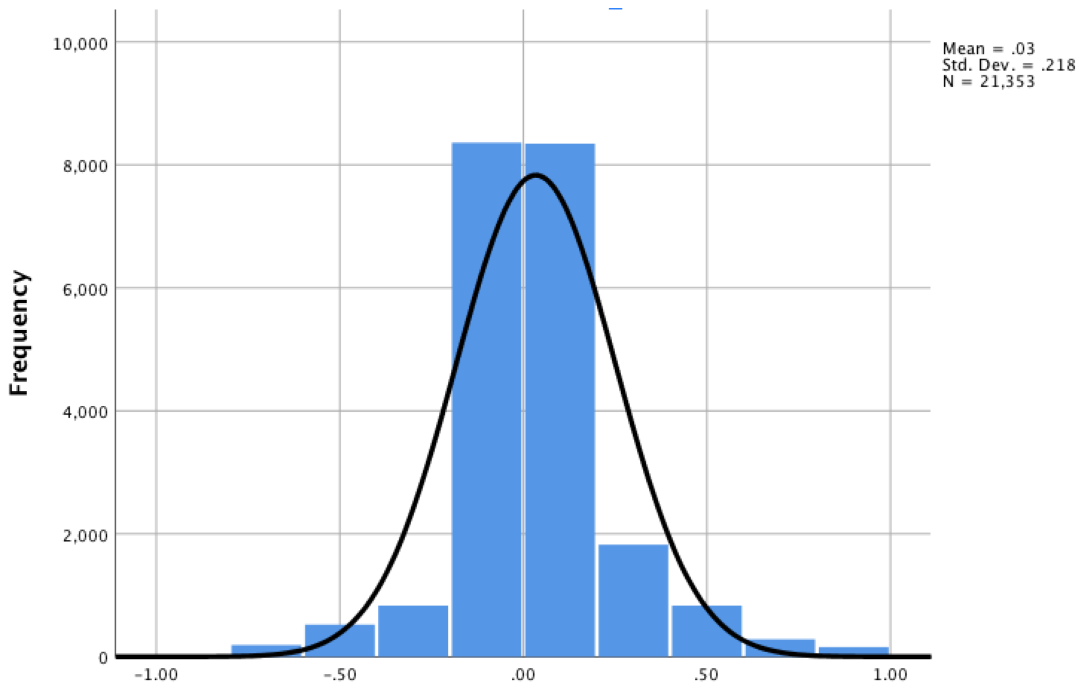


Figure 3.4: The histogram of differences between average precision values of the Bing default ranker and after re-ranking using the long-term user profile (Chapter 5). The differences are transformed with the scale of 0.2.

3.4 Summary

In this chapter, we have detailed the methodology for evaluating search personalisation using dynamic user profiles. In particular, we have presented the benchmark query log collections used for search personalisation. We have also introduced the most widely used metrics as well as the significance test for search personalisation. In the next chapter, we will start our investigation on dynamic group formation for search personalisation.

Chapter 4

Dynamic Group Formation for Search Personalisation

In this chapter, we begin our research and attempt to answer the research questions detailed in the introduction chapter. We explore how to improve the performance of search personalisation using dynamic group formation. To handle this task, we start with building a user profile representing the user's topical search interests.

Unlike classical search methods (that is, methods which rely only on comparison the contents of the query and the document collection), personalised search systems use personal data about a user to tailor search results to that specific user. This information can be considered as a *user profile*. A widely used type of user profile represents the topical interests of the user (Sieg et al., 2007; Bennett et al., 2012; Harvey et al., 2013; Raman et al., 2013; White et al., 2013; Yan et al., 2014). A typical approach is to build user profiles using the main topics discussed in the relevant documents (Sieg et al., 2007; Bennett et al., 2012; Harvey et al., 2013; White et al., 2013; Yan et al., 2014; Vu et al., 2015b,a). The topics covered by a document can be obtained from a human-generated online ontology, such as the Open Directory Project (ODP)[1,2] (Sieg et al., 2007; Bennett et al., 2012; White et al., 2013; Yan et al., 2014). However, this approach suffers from the limitation that many documents may not appear in the online categorisation

[1] http://www.dmoz.org/

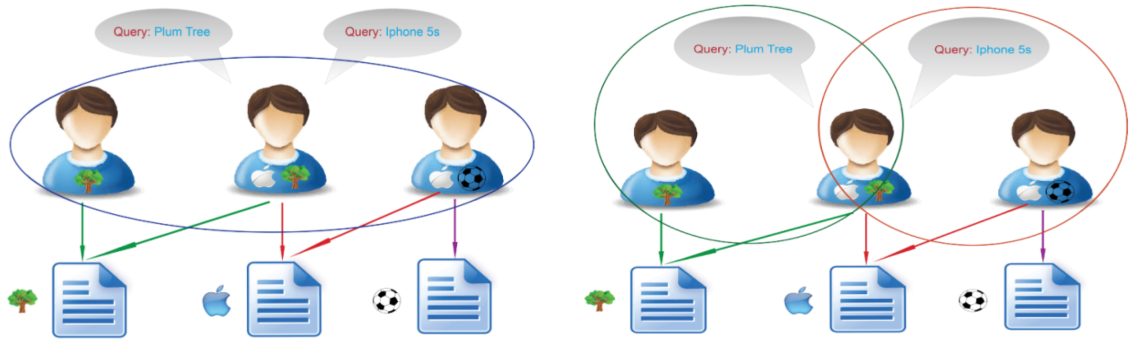
[2] http://www.dmoztools.net/

scheme. Moreover, it requires expensive manual effort to determine the correct categories for each document (Harvey et al., 2013; Vu et al., 2014).

In this chapter, we first look at these problems of ontology-based methods (Bennett et al., 2012; White et al., 2013; Yan et al., 2014) and attempt to answer the following question:

> **RQ 1** *How can we build a user profile which represents the user's topical search interests for search personalisation?*

Recent research has shown that the performance of web search engines can be improved by enriching a user's profile with information about other users who share some common interests (Teevan et al., 2005, 2009; Dou et al., 2007; White et al., 2013). In the existing approaches, groups of similar users are often statically determined, e.g., based on the common documents that users clicked (Dou et al., 2007; White et al., 2013). However, these static grouping methods are query-independent and neglect the fact that users in a group may have different interests with respect to different queries and topics. An example of static group formation using common clicks is shown in Figure 4.1 (*left*). In this case, regardless of the input query (either "plum tree" or "iPhone 5s") from the second user (the current user who is submitting the queries), the first and third users are classified in the same group with the second user.



(Static Group) (Dynamic Group)

Figure 4.1: Static versus Dynamic Group Formation

In this chapter, we will show how groups with shared interests can be dynamically constructed in response to the user's input query. An example of dynamic group formation is shown in Figure 4.1 (*right*), in which the first and second users are classified into a group with respect to the query “plum tree”. Similarly, the third and the second users are grouped in another group with respect to the query “iPhone 5s”. This leads to the second research question which we will attempt to address:

> **RQ 2** *How can we dynamically group users who share common interests for search personalisation?*

More specifically, the group construction is *dynamic* if the group is constructed in an online mode *after* the query is issued by a current user. That is, different input queries should return different groups of users who share some common interests with the current user. The motivation can be approached from many different angles as detailed in the literature review. The question can be divided into three more specific sub-questions that are given below:

1. *How can we dynamically group users who share common interests?*
2. *How can we enrich a user profile with group information?*
3. *Can enriched user profiles help to improve search performance and even perform better than the use of static group formation?*

To answer these research questions, we propose a model for query-dependent user grouping and use the dynamic group information to enrich user profiles for search personalisation. First, we build user profiles based on relevant documents extracted from query logs over a topic space. We utilise an unsupervised topic model to automatically derive topics instead of using a human-generated ontology as in Bennett et al. (2012); White et al. (2013). Specifically, we employ Latent Dirichlet Allocation (Blei et al., 2003) to automatically extract *latent topics* from the user's relevant documents (that is, documents which the user previously clicked on, and remained on with a dwell time of at least 30 seconds). We then utilise the extracted topics from the relevant documents of each user to build that user's profile. Moreover, instead of assuming that all clicked documents are relevant (as assumed in work such as (Dou et al., 2007) and (Raman et al., 2013)), we use the Satisfied (SAT) Clicks criteria (Fox et al., 2005) to identify SAT clicked documents. After that, we introduce a novel method to dynamically group users who have similar interests given the input query of the current user.

After grouping users who share common interests with the current user, we aim to access the performance of our dynamic grouping method by using the group data to enrich the current user profile. We then use the enriched user profile to personalise the search results returned by the Bing web search engine. For comparison, we consider the original ranking of Bing and two other comparative baselines. Our first comparative baseline is to use the user profile without being enriched, and the second is to use statically enriched profiles. As we will see below, our dynamic grouping method significantly outperforms these comparative baselines.

4.1 Personalisation Framework

We start this section with building a topic-based user profile in Section 4.1.1 using topics automatically extracted from that user's relevant documents. We then detail the process of dynamically grouping users who share similar interests with respect to an input query in Section 4.1.2. We finally use the group information to personalise the search results returned by a commercial search engine in Section 4.1.3.

4.1.1 Building a user profile

We infer user interests (i.e. topic-based profiles) implicitly using the relevant data extracted from query logs. The proposed approach consists of three steps.

The first step is to extract the relevant data of each user from their query logs. A log entity consists of an anonymous user-identifier, a submitted query, top-n returned URLs, and clicked results along with the user's dwell times (i.e., the time the user spent on that clicked result). Because a click with a short dwell time ("quick back") tends to be irrelevant data (Bennett et al., 2012), we use the SAT criteria (Fox et al., 2005) to identify satisfied (SAT) clicks (relevant data) from the query logs. A click is classed as a SAT click if it is either a click with a dwell time of at least 30 seconds or the last result click in a search session. Table 4.1 shows the returned documents of a log entry. There are three documents on which the user clicked. However, the document ranked 8^{th} is the only SAT click because it satisfies the SAT criteria. As a "session", we use the common approach of demarcating session boundaries by at least 30 minutes of user inactivity (Kotov et al., 2011; Bennett et al., 2012; Liao et al., 2012; Raman et al., 2013). Table 4.2 shows an example of demarcating two sessions from the query logs.

Table 4.1: Returned URLs in a log entry

Returned Results	**Rank**	**Click?**	**Time**	**SAT click?**
alphaonespf.com	1	-	-	-
youtube.com/watch?v=...	2	√	11s	-
batonrouge.craigslist.org/lab...	3	√	7s	-
manta.com/c/mt47rmw/...	4	-	-	-
...				
tigerdoppings.com/rant...	8	√	132s	√
...				

Table 4.2: An example of demarcating session boundaries

Time	**Query**	**Session**
7/1/2012 10:24:06 AM	baton rouge backpage	1
7/1/2012 10:47:34 AM	Than Merrill	1
7/1/2012 11:03:24 AM	does attic foil insulation work	1
7/1/2012 11:10:10 AM	baton rouge radiant barrier installers	1
7/1/2012 11:21:53 AM	baton rouge spray foam	1
7/1/2012 11:29:18 AM	baton rouge website company	1
7/1/2012 11:29:30 AM	baton rouge web design	1
70 minutes of user inactivity		
7/1/2012 12:39:49 AM	erin berry babin	2
7/1/2012 12:41:51 AM	bbq guys	2

In the second step, we apply the Latent Dirichlet Allocation (LDA) algorithm (Blei et al., 2003) to extract topics from the relevant data (i.e., SAT clicked documents) as described in section 2.2.2. After applying LDA to infer the topics from the documents in the collection, we obtain n topics, in which n is pre-defined as an input to the LDA model. Each topic is described as a distribution over a fixed vocabulary, and each word has a different proportion on the topic. For example, a topic about "football" would give high probabilities to words like "score" and "goal", and low probabilities to "Windows" and "Linux". Moreover, each document is described as a distribution over topics, in which each topic contributes to the document in different proportions. The proportion of each topic indicates how relevant that document is to the topic. For example, a document about a football match would give a high probability to the topic about "football" and a small probability to the topic about "OS".

In the final step, we use the topic distribution of each relevant document extracted from the second step to build a user profile. We model each user as a set of relevant documents. Then, because each document can be modelled as a distribution over topics, we can use a chain rule to model each user as a distribution over topics where each topic has a proportionately different contribution to the user's profile. The proportion of each topic in the user profile indicates how much the user is interested in the topic. We note that a user profile is updated on-the-fly (dynamically) using new relevant documents in the user's current search session.

In figure 4.2, we give an example of how to construct a topic-based user profile from the user's relevant documents. The user has three relevant documents extracted from his search history by applying the SAT criteria. After using LDA to construct topics (in this case, "Football", "Health", "Law" and "OS-Operating System"), each document is described as a distribution over topics. Then, the topic-based profile of the user is constructed as a distribution over the topic set, in which the proportion of each topic is calculated as the mean of the proportions of the topic on the three relevant documents (e.g. $User\_Health = (0.18 + 0.02 + 0.68)/3 = 0.29$). The user distribution over the topic set also indicates that the user is most interested in the "Health" topic because the proportion of the topic is highest (i.e., 0.29).

We describe the approach of constructing a user profile formally as follows. Let Z, W and D be random variables representing a topic, a word and a document respectively. The identity of a user who uses the search engine is a random variable, which we denote by U. Let z, w, d and u denote instances of Z, W, D and U respectively. Thus, for instance, the probability that U takes value u is denoted $p(u)$. After training the LDA model, we get two distributions $P(W|Z)$

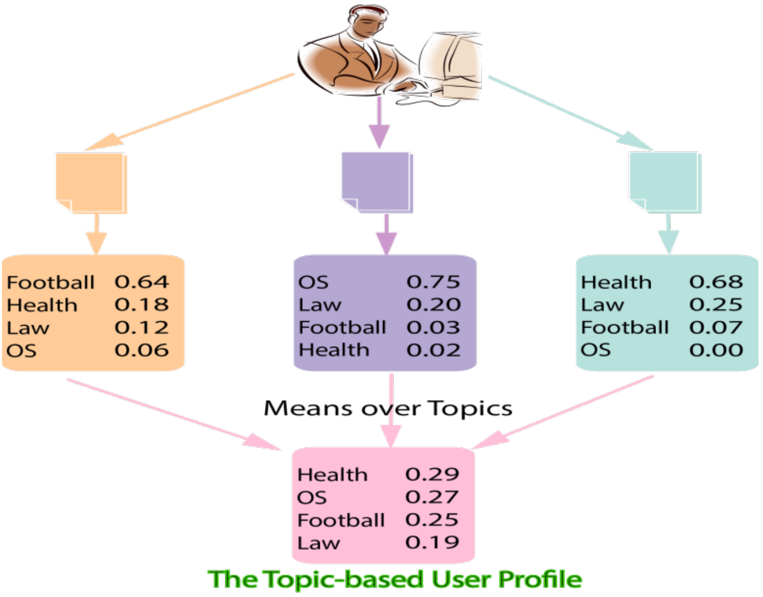


Figure 4.2: Building a user profile

and $P(Z|D)$. $P(W|Z)$ defines a distribution of words for each topic, which shows how relevant they are to the topic. $P(Z|D)$ defines a distribution of these latent topics for each document, which shows how relevant they are to the document.

We define the extent to which a user u is interested in a latent topic z as a conditional probability:

$$p(z|u) = \frac{1}{|SAT(u)|} \sum_{d \in SAT(u)} p(z|d) \tag{4.1}$$

Here, $SAT(u)$ is a set of SAT clicked documents from user u, $|SAT(u)|$ is the size of that set, and $p(z|d)$ is the probability of topic z given the document d. $P(Z|U)$ shows how interested each user is in the latent topics mentioned in his or her SAT clicked documents.

4.1.2 Query-dependent user grouping

In this section, we address the question of how to group users dynamically, based on their input queries. In particular, given a user, different input queries could result in different user groups. The dynamic grouping method contains three steps.

1. The first step is to build the shared user profile between two users using the common relevant documents between these two users.
2. In the second step, with each input query, we calculate the similarity between that input query and a shared user profile. This gives a final score which indicates how topics of interest are shared between the two users given that query.
3. In the final step, we use the score as a similarity measure to form a group of the K-nearest users who share common interests with the current user for the given input query.

In the first step, with a user u, we apply the same technique to build a user profile described in the previous section to construct the shared interest profile between this user and another user v. However, instead of using the SAT clicks of a single user, we use the common SAT clicks of the two users to build the shared interest profile.

Figure 4.3 illustrates the construction of a shared user profile as a distribution over the topics. The shared user profile of u and v is a distribution over topics, in which the proportion of each topic indicates how likely it is that the two users share common interests on the topic. Although the OS topic is only the second most interesting topic of u (Figure 4.2), u and v share the most interest on this topic (corresponding with the highest probability of 0.38 as shown in figure 4.3). After this step, we have all shared user profiles of the user u and another user v. As shown in figure 4.4, the user u shares his or her interests with three other users v_1, v_2 and v_3. In this example, u shares the most interest on "Health", "Law" and "OS" with v_1, v_2 and v_3 respectively (based on the shared user profiles).

After obtaining the shared user profile between the user u and another user v, in the second step, we calculate the topic similarity between the two users with respect to an input query. We assume that the word order in an input query is not important, so we describe the input query as a bag of words. A topic returned from LDA is also modelled as a set of words. Then, each input query can be modelled as a distribution over the topics, in which each probability value describes the proportion of the query in each topic. For instance, in Figure 4.5, the query "windows 10" is described as a distribution over the topics, in which the corresponding value of each topic is the proportion of "Windows 10" in that topic. The proportion of the query in a

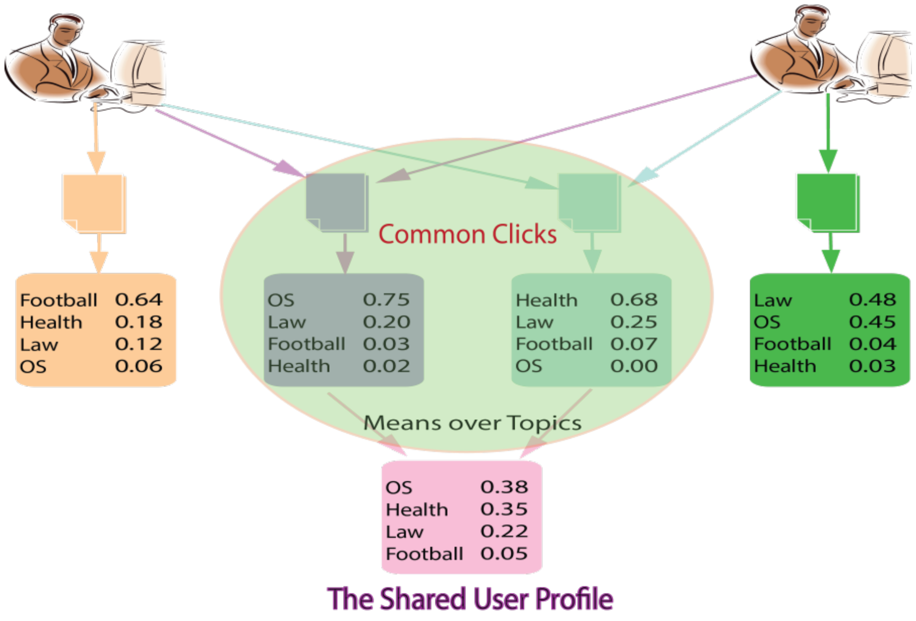


Figure 4.3: Building a shared user profile

topic is extracted from the LDA topic distributions. From Figure 4.5, the query “windows 10” is more relevant to the “OS” topic than to, say, the “Law” topic.

After that, we measure the similarity between a shared user profile and an input query using either a probability-based method (such as JensenShannon divergence (Lin, 1991)) or a vector-based function (such as cosine similarity). Because both the shared profile and the input query are presented as $P(Z|q)$ and $P(Z|sp)$ distributions over topics Z, respectively, we can measure the similarity, *SharedSim*, between a shared profile and an input query using a probability-based method. Among many probability-based methods, Jensen-Shannon divergence ($D_{JS}\lfloor.||.\rfloor$) is a useful symmetric measure of the distance between two probability distributions (Lee, 1999), and has been widely used in IR research (Bennett et al., 2012). Because JensenShannon divergence is a method of measuring the difference of two distributions (Lin, 1991), for ease of use, we define *SharedSim* as $-D_{JS}\lfloor.||.\rfloor$ as follows:

$$SharedSim(q|sp) = -D_{JS}\lfloor Q||SP\rfloor = -\frac{1}{2}D_{KL}\lfloor Q||M\rfloor - \frac{1}{2}D_{KL}\lfloor SP||M\rfloor \tag{4.2}$$

where $D_{KL}\lfloor.||.\rfloor$ is the Kullback-Leiber divergence and $M = \frac{1}{2}(Q + SP)$.

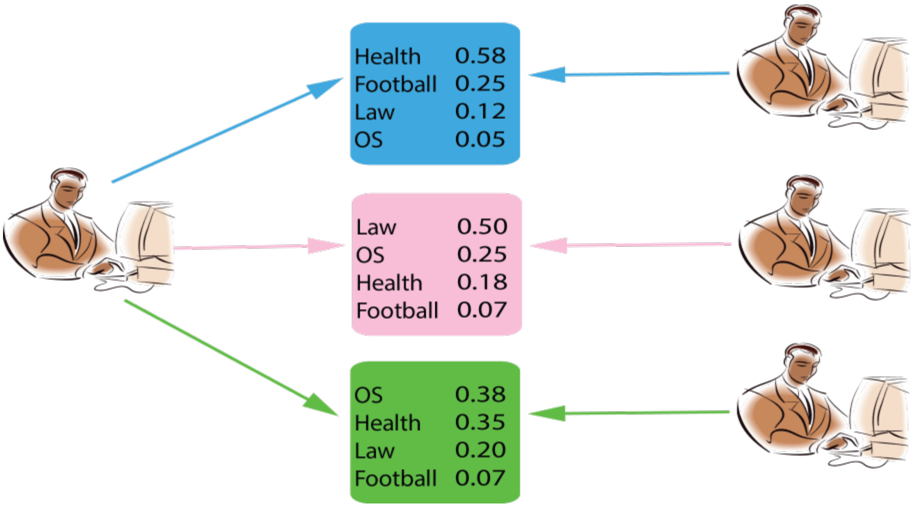


Figure 4.4: The three shared profiles between the user u and other users v_1, v_2 and v_3

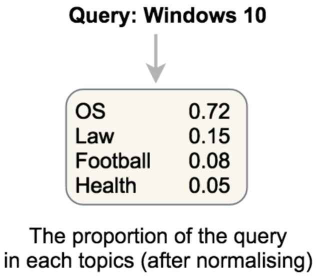

Query: Windows 10
OS 0.72
Law 0.15
Football 0.08
Health 0.05
The proportion of the query
in each topics (after normalising)

Figure 4.5: The distribution of the query "Windows 10" over the topic space

Now, for each shared profile sp between u and v, and an input query q, we get a similarity score $SharedSim(q|sp)$. The higher the score is, the more interest the user v shares with the user u given the query q. Figure 4.6 shows an example of *SharedSim* scores between the input query "windows 10" (Figure 4.5) and the three shared profiles (Figure 4.4). The third profile has the highest *SharedSim* score with the input query which indicates that the user v_3 shares the most common interest with the user u with respect to the query.

In the final step, using the similarity scores from the second step, we identify the K-nearest users to the user u (ie. those users who have the highest similarity score) based on their shared common interests given the input query q. As shown in Figure 4.6, if we set $K = 2$, the third and the second users (i.e., v_3, v_2) are the two nearest users who share common interests with

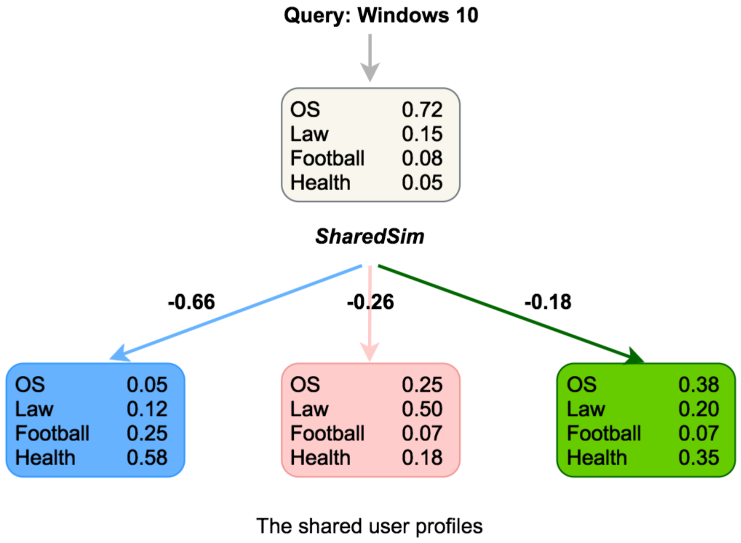


Figure 4.6: The similarity scores (i.e., *SharedSim*) between the input query and the shared profiles

the current user u because the similarity scores are -0.18 and -0.26, respectively.

4.1.3 Re-ranking search results using group information

After obtaining the K-nearest users of the user u (denoted as G_u), we first use the data from the K users to enrich the current user profile. We then utilise the enriched user profile to re-rank the original list of documents returned by the Bing search engine.

To enrich the current user profile, we apply the same technique to construct a single user profile, as shown in section 4.1.1. However, instead of using the relevant documents, we use the profiles of the K-nearest users together with the current user's profile to construct an enriched user profile (p_e). The enriched profile is described as a distribution over topics, in which the proportion of each topic is calculated as an average of those of the topic over the K users together with the current user.

$$p_e(z|u) = \frac{p(z|u) + \sum_{v \in G_u} p(z|v)}{K+1} \tag{4.3}$$

Figure 4.7 shows an example of an enriched user profile. The profile is enriched *dynamically* because the K-nearest users are extracted using the input query at the query time. After enriching the profile, we can see that the interest of the current user in the "Health" and "OS" topics are larger while the user interest in the "Football" and "Law" topics are decreased in comparison with those in the original user profile.

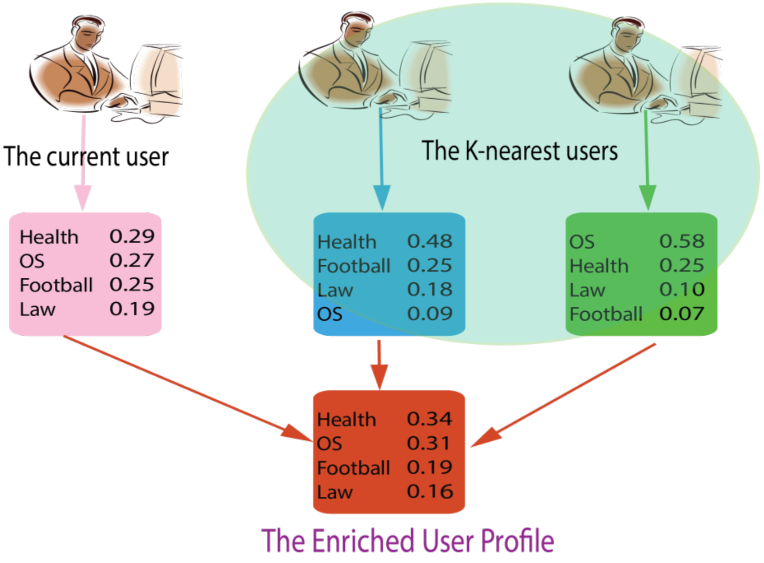


Figure 4.7: An enriched user profile

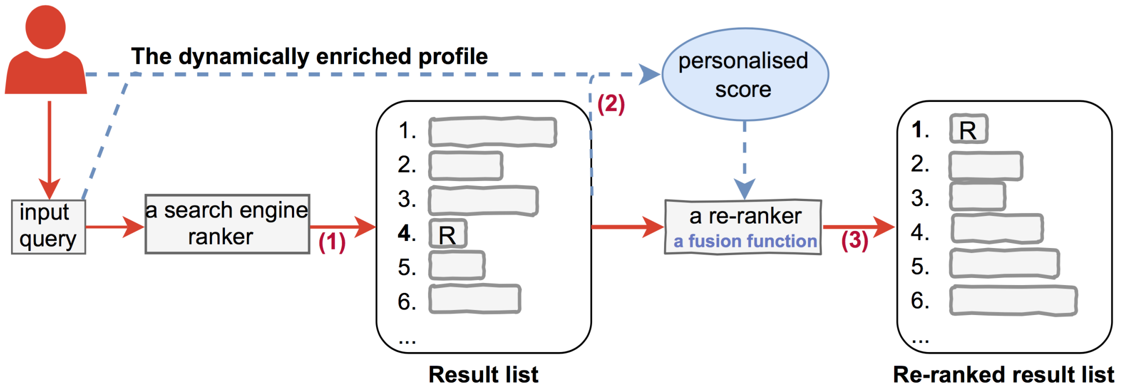


Figure 4.8: The general process of re-ranking. R means that the document is relevant to the user

We can now use the enriched user profile to re-rank the original list of documents returned by a search engine. Figure 4.8 shows an overview of how the enriched user profile is used to re-rank the original document list returned by the search engine. After re-ranking the document list, we expect that the more relevant documents will be promoted to higher ranks. For each query q of the current user u, the detailed steps are as follows:

1. Download the top n ranked search results (as in query logs) returned from the search engine for the query. We denote a downloaded web page as d and its rank in the search result list as $r(q, d)$.

2. Compute a personalised score for each web page d given the current user u as $sim(d|u)$. Since both d and u are modelled as a distribution over topics, we calculate $sim(d|u)$ using JensenShannon divergence as described in section 4.1.2. To use the group information, we use the enriched profile p_e instead of the original profile of the current user.

3. Combine the personalised score $sim(d|u)$ and the original rank $r(q, d)$, to get a final score as:

$$f(d|u, q) = \frac{sim(d|u)}{r(q, d)} \tag{4.4}$$

As we do not have access to the original relevance score between the query q and the returned document d given by the baseline search engine, we use $1/r(q, d)$ as an estimate. We combine $sim(d|u)$ and $r(q, d)$, as they reflect different aspects in ranking documents. A higher final score indicates that the document is more relevant to the user and should be ranked higher. We then re-rank the returned documents to obtain a new ranked list. For example, in figure 4.8, after the re-ranking step, the ranks of a relevant document changes from the 4^{th} place to the 1^{st} place. Then, we will compare the two ranked lists (before and after re-ranking) using IR evaluation metrics detailed in Chapter 3.

4.2 Experimental Setup

4.2.1 Evaluation metrics

We use two measurement metrics to evaluate a personalised search approach, which are: inverse average rank (*IAR*) and personalisation gain (*P-Gain*) as detailed in Chapter 3. As mentioned in chapter 3, the higher *IAR* score indicates a better ranking quality while a higher positive *P-Gain* value indicates a better overall robustness of a personalisation algorithm in terms of improving the performance over the baseline. As shown in figure 4.8, the relevant set P_s for the query also consists of only one relevant document. Before search personalisation, the relevant document is ranked 4^{th}; thus $IAR = 1/4 = 0.25$. After search personalisation, the relevant document is ranked 1^{st}; thus $IAR = 1/1 = 1$. Figure 4.8 also indicates a case in which the more relevant document has been re-ranked to higher up the list, so that after re-ranking the relevant document is promoted from the 4^{th} rank to the 1^{st} rank. Accordingly, $P\text{-}Gain = \frac{1-0}{1+0} = 1$.

4.2.2 Dataset and evaluation methodology

The dataset used in our experiments is a sample of query logs from the Bing search engine for 15 days from the 01^{st} to the 15^{th} July 2012. The query logs contain search data from 106 anonymous users. A log entity consists of an anonymous user identifier, a query, the top-10 returned URLs for that query, and clicked results along with the user's dwell time. For each URL, we use the main content (i.e. title and body) for training the LDA topic model.

For evaluation, we use the SAT criteria to identify the satisfied clicks from the query logs. We then split the dataset into training and test sets. The training set contains the log data for the first 10 days, and the test set contains the log data for the remaining 5 days. Table 4.3 shows some statistics for the query logs. We also consider the SAT clicks as the ground truth of the test data. In our experiments, we evaluate our proposed method by comparing the original ranked list given by the Bing search engine and the re-ranked list given by our methods with the evaluation metrics defined in Section 4.1.3.

In addition to reporting the overall performance, we also analyse the results with respect to the concept of query click entropy (Dou et al., 2007) as a direct indication of query click

Table 4.3: Basic statistics of the dataset

Item	ALL	Training	Test
#days	15	10	5
#users	106	106	106
#queries	17,947	11,695	6,252
#distinct queries	8,008	5,237	3,102
#clicks	24,041	15,688	8,353
#SAT clicks	16,166	10,607	5,559
#SAT clicks/#queries	0.9008	0.9069	0.8892

variation.

$$QueryClickEntropy(q) = \sum_{d \in U(q)} -p(d|q) \log_2 p(d|q) \tag{4.5}$$

Here $U(q)$ is a collection of web pages which are clicked for the distinct query q, and $p(d|q)$ is the percentage of the clicks on document d among all the clicks for q. A smaller query click entropy value indicates that more agreement between users on clicking a small number of web pages with respect to that query (Dou et al., 2007). Dou et al. also pointed out that if the query click entropy is small, the personalisation process can even deteriorate the search performance. In the experimental data, about 80% of the queries have a low click entropy (where "low" is between 0 and 1), 15% of the queries have a click entropy between 1 and 2, and about 5% of the queries have a higher click entropy (greater than 2) (Figure 4.9).

4.3 Experimental results

In the experiment, we compare the performances of the baseline and three personalisation strategies which we have called S_Profile, S_Group and D_Group. The baseline is the originally ranked results from the Bing search engine. S_Profile is a personalisation approach using the current user profile, S_Group uses the profile enriched with information from static grouping $p(u, v)$ (as shown in Figure 4.10, we use only the number of common clicks between two use to group users and are regardless of the input query), and D_Group is enriched with information from dynamic

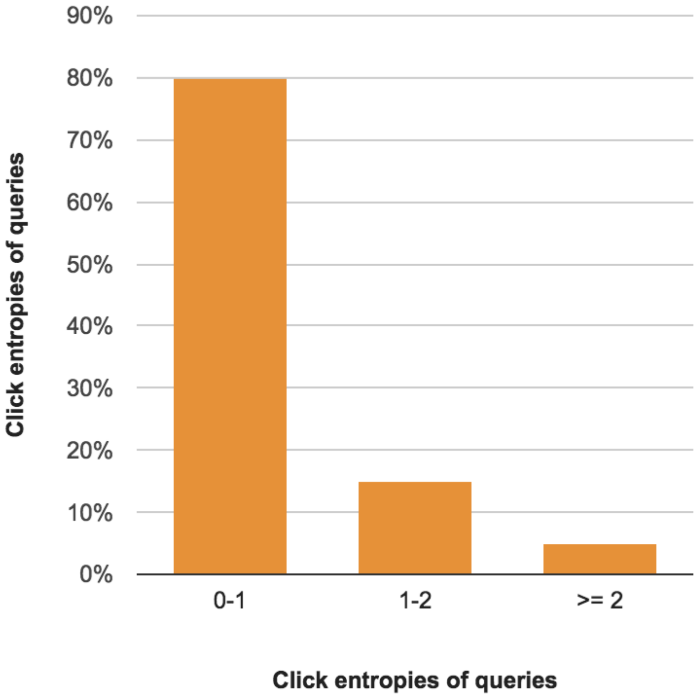


Figure 4.9: Query click entropies on our experimental dataset

grouping $p(u, v|q)$.[3,4]

For training the LDA model, we employ the Mallet implementation (McCallum, 2002) of the LDA model. We also observe that the choice of hyper-parameter has little impact on overall performance. Therefore, in this work, we set the number of topics as 100 and the hyper-parameters as in Blei et al. (2003). Since the number of the anonymous users is relatively small (106), we set the number of nearest neighbours $K = 5$ for both S_Group and D_Group.

4.3.1 Overall performance

In this section, we analyse the experimental results of the personalised strategies regarding *IAR* and *P-Gain*. Table 4.4 shows that all personalisation strategies can lead to improvements over

[3] We use the training set to initialise the user profiles as well as the shared user profiles. We also use the training set to statically group users who share common interests

[4] We use the test set to compare the performances of the personalisation strategies after re-ranking. We also use the satisfied clicks extracted from the test set to update the user profiles as well as the shared user profiles.

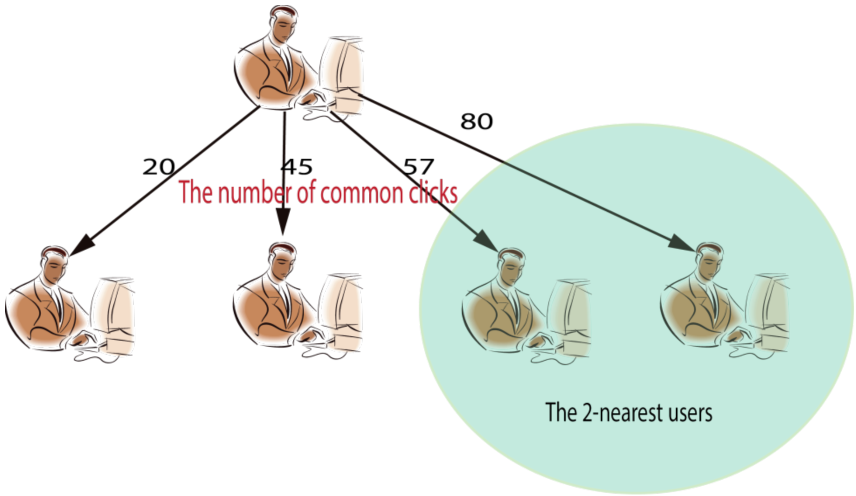


Figure 4.10: The static grouping method

the baseline (i.e., all reported changes are positive in *IAR* and *P-Gain* values). Interestingly, the D_Group method has the highest improvement ($p < 0.01$ with the paired t-test)[5] of 8.12% over the baseline (using the *IAR* metric). S_Group and S_Profile methods also have significant improvements of 7.64% and 5.94% respectively ($p < 0.01$) over the baseline. This shows that latent topic-based personalisation methods obtain better web search performance. We note that the relevant (SAT) documents were largely (more than 75%) in the first half of the original lists , the improvements after re-ranking were not based on chance.

Table 4.4 also shows that the Group-based methods (S_Group and D_Group) improve the *IAR* and *P-Gain* values by at least 1.60% and 80.37% respectively over the S_Profile ($p < 0.01$)[6]. Consistent with Teevan et al. (2009), our result confirms that the information from the group of users who share some common interests is helpful in building better user profiles.

The dynamic grouping method (D_Group) leads to improvements of 0.45% and 14.22% in *IAR* and *P-Gain* respectively over the static grouping method (S_Group) ($p = 2.76E - 4$). Furthermore, in Table 4.5, even though D_Group method leads to only two more improved ranks of relevant documents than S_Group, the former has much (41) fewer worse ranks. This suggests that dynamic group information with respect to an input query could improve performance in

[5] The exact p values are reported in Table B.1, Appendix B.

[6] $p = 3.41E - 5$ for S_Group and $p = 4.88E - 16$ for D_Group, respectively.

Table 4.4: Overall performance of the methods

Strategy	*IAR*	*P-Gain*
Baseline	*0.3473*	-
S_Profile	0.3679	0.1579
vs. Baseline	*+5.94%*	-
S_Group	0.3738	0.2848
vs. Baseline	*+7.64%*	-
vs. S_Profile	*+1.60%*	*+80.37%*
D_Group	**0.3755**	**0.3253**
vs. Baseline	*+8.12%*	-
vs. S_Profile	*+2.07%*	*+106.02%*
vs. S_Group	*+0.45%*	*+14.22%*

Table 4.5: Numbers of better and worse ranks in comparison with the baseline and *P-Gain*

Strategy	#Better	#Worse	*P-Gain*
S_Profile	913	664	0.1579
S_Group	882	491	0.2848
D_Group	884	**450**	**0.3253**

term of robustness over static group information, especially for reducing the number of incorrect re-rankings.

A remark on efficiency issues

In LDA, both training and static group formation are done offline, and the dynamic group formation is done partially offline.[7] Thus the online processing of S_Profile, D_Group and S_Group methods is reasonably efficient for the small dataset. The average processing time per query is about **0.70** milliseconds for S_Profile and S_Group, and **1.09** milliseconds for D_Group. In fu-

[7] We initially construct the shared user profiles from section 4.1.2 offline. After that, we update a shared profile online using new relevant documents to the current user in the current search session.

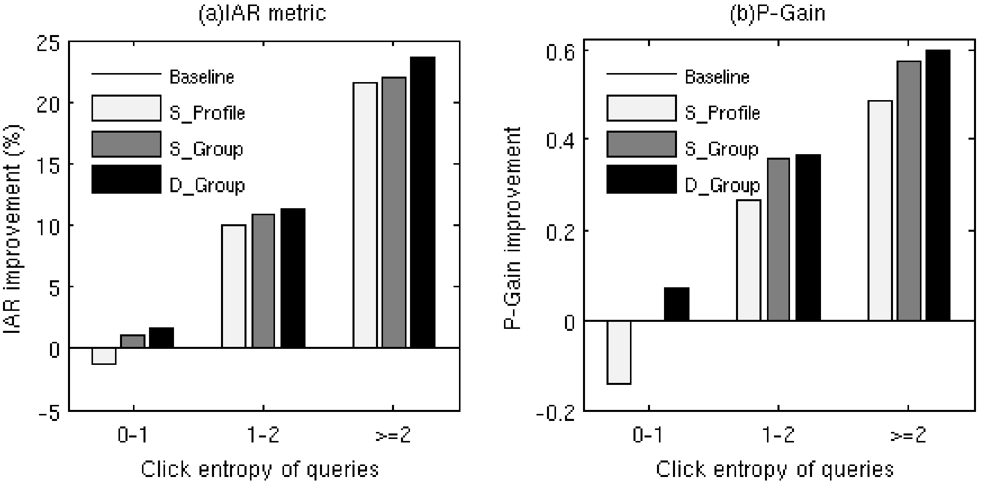


Figure 4.11: Search performance improvements over the baseline with different click entropies

ture work, we aim to further improve the efficiency of the D_Group method by applying parallel programming.

4.3.2 Performance on different query click entropies

In this section, we evaluate the search performance on different click entropies (figure 4.11). In common with Dou et al. (2007), we find that for queries with small query click entropy, the personalisation methods do not improve, and can even worsen, the search performance. Specifically, with query click entropies between 0 and 1, the *IAR* of the S_Profile method is 1.31% lower than the baseline's. However, improvements are achieved by using the group-based methods (S_Group and D_Group). As can be seen in Figure 4.11, the improvement of personalised search performance increases significantly when the click entropy becomes larger, especially with click entropies greater than 1. Furthermore, the personalisation methods achieve highest improvements when click entropies are no less than 2. In this case, all three personalisation methods have improvements of more than 22% in *IAR* score over the baseline. These results indicate that the higher click entropy is, the better performance the search personalisation is likely to achieve. Moreover, in general D_Group performs better than S_Group, and both D_Group and S_Group methods outperform S_Profle.

Table 4.6: The performances of D_Group method over the different group sizes

Group Size	*IAR*	*P-Gain*
0/S_Profile	*0.3679*	*0.1579*
1	0.3726	0.2613
2	0.3734	0.2651
3	0.3740	0.2930
4	0.3744	0.3130
5	**0.3755**	**0.3253**

4.3.3 Performance on different group sizes

We also investigate the impact of the group size on the performance of the D_Group method. Table 4.6 shows the search performance using the dynamic grouping method against different numbers of nearest neighbours. Due to the small size of the dataset (106 users), forty-five (42.5%) users in the dataset have common clicks with no more than five other users. Therefore, we test the number of similar users from 1 to 5 in this experiment. The results show that in general, the more users share common interests with the current user, the higher is the performance that the D_Group tends to achieve.

This indicates that the information from user groups is useful. Even with only one other user in the group, the performance of the D_Group method achieves improvements of 1.28% (*IAR*) and 65.48% (*P-Gain*) over S_Profile where user profiles are not enriched by group information. With five nearest users, the D_Group method achieved the highest performance: improvements of 2.07% in *IAR* and 106% in *P-Gain* over S_Profile ($p = 4.88E - 16$).

4.4 Conclusions

In this chapter, we have presented a framework for search personalisation using a dynamic grouping method to enrich a user profile. For each user, we built the user profile using all the user's relevant documents and treating the relevant documents equally. This profile is *dynamic* because the profile is updated "on-the-fly" using the user's relevant documents extracted from the interactions between that user and the Bing search engine. The profile is dynamically enriched with information from other users whose interests are similar to the user given a query. Applying it to web search, we use the enriched profile to re-rank search results. We performed a set of experiments to study the effectiveness of the enriched profiles. Our experiments provide answers to the main research questions raised at the beginning of this chapter:

> **RQ 1** *How can we build a user profile which represents the user's topical search interests for search personalisation?*

To answer this question, we set up a personalisation baseline (S_Profile) in which we utilised a single user profile (without enriching) to re-rank the search result returned by the Bing web search engine. Our experimental results show that S_Profile helps to improve the search performance significantly with a relative improvement of 5.94% over the default ranker of Bing. Our experimental results also show that S_Profile produces the highest number of better rank in comparison with the default ranker.

> **RQ 2** *How can we dynamically group users who share common interests for search personalisation?*

Our experimental results demonstrated that the dynamically enriched profile could stably and significantly improve the ranking quality over the competitive ranker of the original search engine and the individual as well as statically enriched profiles. Other experimental results confirmed the impact of the query click entropies. That is, the higher click entropy is, the better performance the search personalisation is likely to achieve. Finally, the experimental results on the user group sizes indicated that we achieved better search performances using data from more users with shared interests.

Although a single user profile (i.e., S_Profile) helped to improve the search performance overall, we built the profile without considering temporal features (i.e., the time of documents being clicked and viewed). It leads to the fact that the profile is too broad and might not fully express the user's current search interests. Figure 4.12 shows that a user is more and more interested in the "OS" topic as she has recently clicked on documents more relevant to the

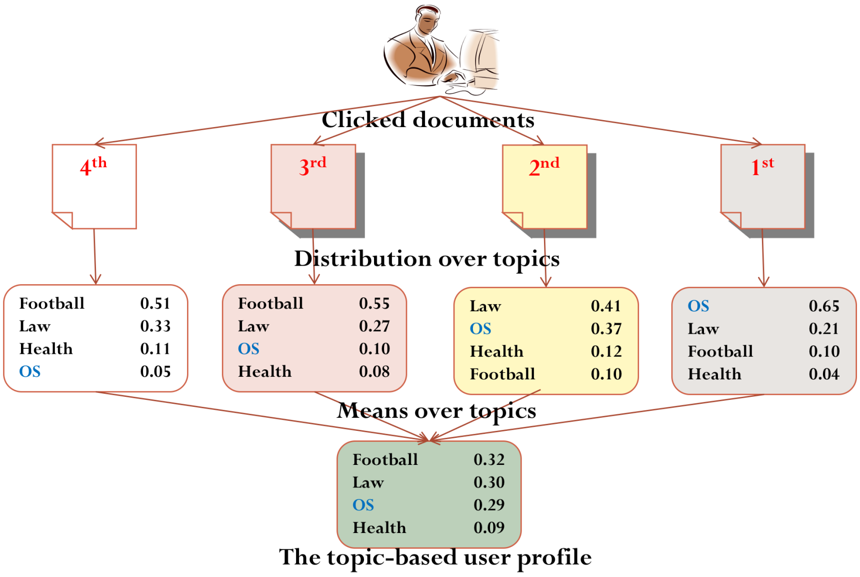


Figure 4.12: An example of building a user profile which does not quickly represent the user's search interest. The smaller order of each document shows that the document is more recently clicked

OS topic. Unfortunately, the user profile does not quickly represent the user interest as it still expresses that "football" is the most interesting topic.

Also, the single profile even harmed the performance with small query click entropies (Figure 4.11). Moreover, with more interactions with the search system, the user's interests may change over searching time. New topics of interest may gradually emerge while interest in some of existing topics may fade (Nanas et al., 2003; Bennett et al., 2012). Therefore, the modelling method should be able to build user profiles in such a more dynamical way that can quickly capture the user's interests. We assume that the more recent relevant document expresses more about the user's interests than a distant one. In next chapters, we will extend the user profiling model to capture the user's interests that change over time.

Chapter 5

Temporal User Profiles for Search Personalisation

In chapter 4, we introduced a user profiling method using Latent Dirichlet Allocation (Blei et al., 2003) for unsupervised extraction of the topics from documents. The user profile built using this method is *dynamic* as it is updated "on-the-fly" when the system observes the user's new relevant documents. Figure 5.1 shows an example of a user profile dynamically updated using the user's clicked documents. To build the user profile, we treated all the user's relevant documents equally. However, we now argue that treating all relevant documents as equally important is not an appropriate method to built a user profile, as the user profile is normally too broad. This method may also not be sufficiently responsive to capture the user's current information needs. In Figure 5.2, we illustrate the case where a user is more and more interested in the Operating System ("OS") topic as she has recently clicked on documents (i.e., $1^{st}, 2^{nd}$ documents) relevant to that topic. However, by treating all the four documents equally, the user profile cannot represent the current interest of that user on the "OS" topic and shows that the most interesting topic is still "football".

Furthermore, we did not pay attention to the *temporal* aspects in the topic profiles, which reflect an important type of search context. That is, the user's interests may change over searching time (Bennett et al., 2012). For instance, the query "US Open" is more likely to be

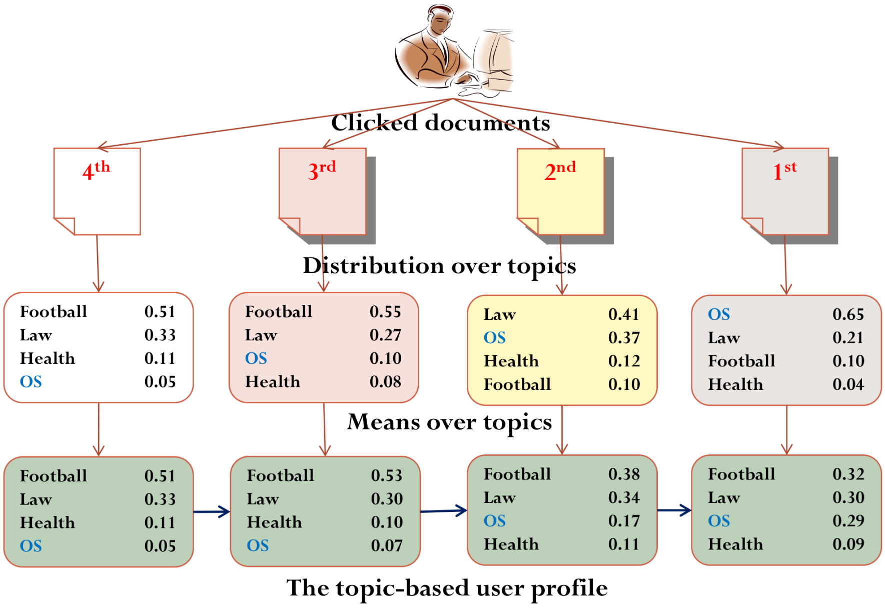


Figure 5.1: A user profile is dynamically updated using the user's clicked documents

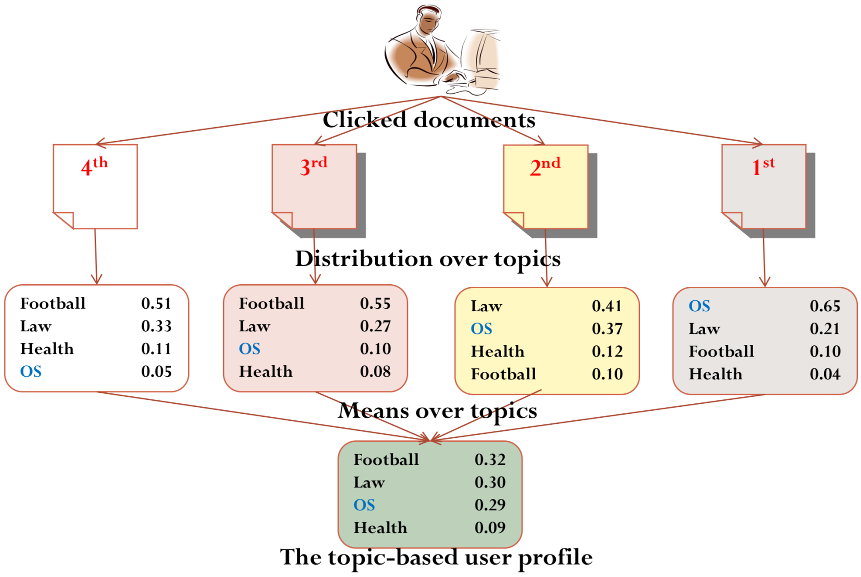


Figure 5.2: An example of building a user profile, in which all the clicked documents are treated equally. The smaller order of each document shows that the document is more recently clicked

targeting the golf tournament in June, while in September it is the prominent keyword for a tennis tournament. We explore how these problems can be handled by addressing the following research question:

RQ 3 *How can we build temporal user profiles for search personalisation?*

The above general question can be divided into three specific sub-questions:

1. *How can we build temporal profiles using topics extracted from relevant documents?*
2. *How do temporal aspects affect the re-ranking quality?*
3. *Can the temporal profiles help to improve the search performance and perform better than the non-temporal user profiles?*

To answer this main research question, we propose a temporal profiling approach to building user profiles from latent topics. We then carry out a study on the effectiveness of temporal features in learning the topical interest of a user, with application to search results re-ranking. Further, with more interactions with the search system, the user's search interests may change over time. To capture this characteristic, the profiling method should be able to build user profiles in a dynamic way that can quickly adapt to the user's current search interests. To this end, we construct three temporal latent topic profiles for each user using the relevant documents with different time scales in the user's search history. We name the profiles as the *session profile*, the *daily profile* and the *long-term profile*, as they are built from the topics extracted from the documents within a search session, a day and a whole history respectively. We note that the three profiles represent the user interest on different time scales (from short-term to long-term). In the experiments, we utilise the profiles to re-rank search results returned by the Bing search engine. Our experimental results demonstrate that our temporal profiles can significantly improve the ranking quality. The results also show a promising effect of temporal features in correlation with click entropies and query positions in a search session.

In contrast to Bennett et al. (2012) and White et al. (2013), we apply LDA to automatically derive the latent topics from the user's relevant documents. As well as building long-term and short-term profiles (as suggested by Bennett et al. (2012)), we also build the daily user profile from the user's interactions in the current searching day. Furthermore, in contrast to Harvey et al. (2013) and Vu et al. (2014) in building a single long-term user profile by treating all the interactions equally, we propose three temporal user profiles (i.e., long-term, daily and session profiles) which can represent both long-term and short-term user interests. Our long-term profile is also different from that in Chapter 4 in terms of considering the dwell-time/viewing-time of each relevant document (Section 5.1.2). We then investigate the effectiveness of the proposed profiles in search personalisation.

The rest of this chapter is structured as follows. Section 5.1 describes our personalisation framework for building the temporal profiles and using the profiles to re-rank the returned result list. In section 5.2, we describe our experimental setting. We then report the results in section 5.3 and conclude the chapter in Section 5.4.

5.1 Personalisation framework

We start by briefly describing how to extract topics from the user's relevant documents using LDA in section 5.1.1. We then detail the process of temporal user profiling to build long-term, daily and session user profiles in section 5.1.2. Finally, in section 5.1.3 we show how the temporal user profiles can be utilised in a learning-to-rank mechanism to personalise the search results returned by a search engine.

5.1.1 Extracting topics from relevant documents

We briefly describe the method to extract topics from relevant documents, which was initially proposed in the previous chapter. We first extract the relevant data of each user from the query logs. A log entry consists of an anonymous user-identifier, a submitted query, top-10 returned URLs and clicked results along with the user's dwell time on these results. We use the SAT criteria detailed in Fox et al. (2005) to identify satisfied (SAT) clicks (as relevant data) from the query logs as either a click with a dwell time of at least 30 seconds or the last result click in a search session. To identify a "session", similar to Chapter 4, we use the common approach of demarcating session boundaries by 30 minutes of user inactivity (Kotov et al., 2011; Bennett et al., 2012; Liao et al., 2012; Raman et al., 2013). After that, we employ LDA (Blei et al., 2003) to extract latent topics (Z) from the SAT clicked documents (D) of all users.

5.1.2 Building temporal user profiles

Building a User Profile We now utilise the topics extracted from a user's relevant document to build that user profile. In Chapter 4, we equally treated all the user's relevant documents to build the user profile. However, in this chapter, we assume that the more recently clicked (relevant) documents express more about the user's current search interests. To capture this criterion, we introduce a weighting scheme to assign a higher weight to the more recently clicked documents.

Specifically, we denote the user variable as U. Let u denote an instance of U. We build a user profile based on the topics of the user's relevant documents (e.g., clicked documents satisfying some predefined criteria). Let $D_u = \{d_1, d_2, .., d_n\}$ be a relevant document set for the user u. Because each document d_i is modelled as a distribution over the topics Z, by using the chain rule each user u can be modelled as a distribution over the topics Z (i.e., $P(Z|U)$) where each topic z makes a proportionately different contribution to the profile of the user u. The proportion of each topic on the user indicates how interested the user is in that topic. Formally, the probability of a topic z given u is defined as a combination of probabilities of z given a relevant document $d_i \in D_u$ as:

$$p(z|u) = \sum_{d_i \in D_u} \lambda_i p(z|d_i) \tag{5.1}$$

Here, λ_i is the weighting parameter which indicates how important the document d_i is in building the profile of the user u. $\sum_i \lambda_i = 1$ to guarantee that $\sum_z p(z|u) = 1$. The simple approach as used in Chapter 4 is to treat relevant documents equally when calculating $p(z|u)$. It means that $\lambda_1 = \lambda_2 = ... = \lambda_n = \frac{1}{|D_u|}$. Therefore, we have

$$p(z|u) = \frac{1}{|D_u|} \sum_{d_i \in D_u} p(z|d_i) \tag{5.2}$$

Temporal weighting Due to dynamic interactions of the user with the search system, the search intent and user interests will change over time. New topics of interest may gradually emerge while interest in some of the existing topics may fade. We assume that the more recent relevant documents could express more about the user interest than the distant ones. This characteristic can be captured by introducing a decay function (White et al., 2010; Bennett et al., 2012). In this chapter, instead of treating all the relevant documents equally as in chapter 4, we model λ_i as an exponential decay function of t_{d_i}, which is the time the user u clicked on the document d_i, as:

$$\lambda_i = \frac{1}{K}\alpha^{t_{d_i}-1} \tag{5.3}$$

where $K = \sum_{d_i} \alpha^{t_{d_i}-1}$ is a normalisation factor, with $t_{d_i} = 1$ indicating that d_i is the most recent relevant (SAT clicked) document. α is the decay parameter ranging from 0 to 1 (that is, $0 < \alpha \leq 1$), which is set by using a validation set in our experiments. By applying Eq. 5.3 to Eq. 5.1, we have

$$p(z|u) = \frac{1}{K}\sum\nolimits_{d_i \in D_u} \alpha^{t_{d_i}-1} p(z|d_i) \tag{5.4}$$

In the Figure 5.3, we show an example of how to construct a temporal topic-based user profile from the user's relevant documents. The user has four relevant documents extracted from her search history (i.e. query logs) by applying the SAT criteria as in Chapter 4. After using LDA to construct topics (i.e. "Football", "Health", "Law" and "OS-Operating System"), each document is described as a distribution over the topics. Then, the session profile of the user is constructed as a distribution over the topic set, in which the proportion of each topic is calculated using Eq. 5.7 with the decay parameter $\alpha = 0.9$.

For example,

$$User\_OS = \frac{0.9^3 \times 0.05 + 0.9^2 \times 0.13 + 0.9^1 \times 0.37 + 0.9^0 \times 0.65}{0.9^3 + 0.9^2 + 0.9^1 + 0.9^0} = 0.31$$

It can be seen that the user's relevant documents are more and more related to the "OS" topic (that is, the proportion of the "OS" contribution to the relevant documents increases from 0.05 to 0.65, becoming the topic in which the user is most interested). From figure 5.3, the temporal user profile's distribution over the topic set indicates that the user is most interested in the

"OS" topic because the proportion of the topic is highest (i.e., 0.31). However, from the non-temporal profile[1](on the bottom-right of figure 5.3), the proportion of the topic OS is ranked third (i.e., 0.28). Thus the temporal user profile is able to represent the user evolving interests more accurately than the non-temporal user profile (ranked *first* versus ranked *third*).

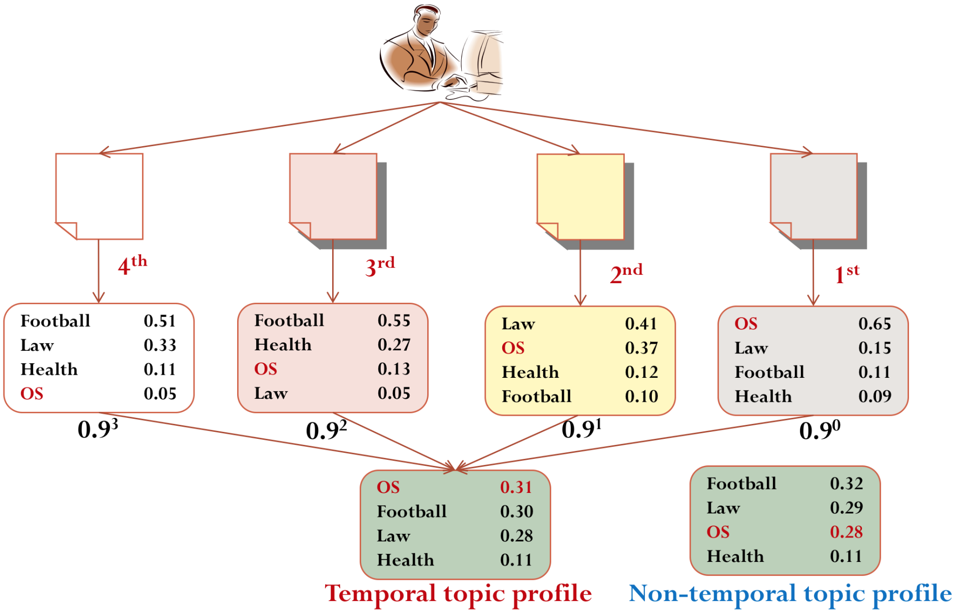


Figure 5.3: Building a temporal user profile

Discussion

In the previous chapter, we built only a single topic-based user profile (i.e., a long-term profile). This work treated all the relevant documents equally and used the user's whole search history to construct the profile. In this chapter, however, we treat the relevant documents temporally based on the viewing time of the user on the document. Furthermore, a single long-term profile cannot dynamically represent the short-term interest of a user in a search session or on a specific day.

For example, for a user having a strong law background, the long-term profile of the user has

[1] Here, we apply the same technique detailed in Chapter 4 to build the non-temporal profile: all the relevant documents are treated equally by using the long-term profile with $\alpha = 1$

been constructed from thousands of law-related documents. On the first day of the Euro 2016, even though she submitted Euro-related queries (e.g., "France Euro Squad") and clicked on the Euro related documents, the updated long-term profile cannot change promptly to express the football interest and does not seem to help to personalise the Euro related queries.

Therefore, apart from the long-term profile, we model two other shorter-term profiles, namely *daily* and *session* profiles using the user's relevant documents in the current searching day and current search session respectively. Here, the long-term profile represents the long-term interest of the user. The session profile describes the provisional interest of the current user. The daily profile indicates the user interest over a searching day. Finally, we construct the three user profiles using different relevant document sets in different time intervals as follows:

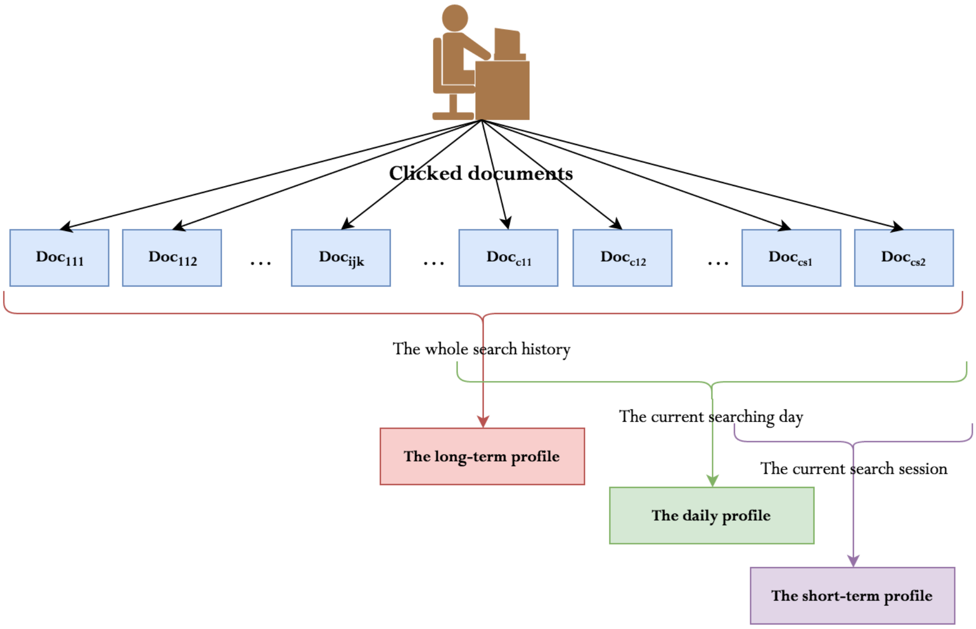


Figure 5.4: Building the long-term, daily and session user profiles from the user's search history. Doc_{ijk} indicates the user clicked on that document on the i^{th} day, in the j^{th} search session on that day; and it is the k^{th} click on the j^{th} search session. $i = 1$ indicates the first day the user used the search system. $i = c$ and $j = s$ indicate the current search day and the current search session, respectively.

Long-term Profile We build the long-term user profile of u using relevant documents D_w extracted from the user's whole search history as follows:

$$p_w(z|u) = \frac{1}{K} \sum\nolimits_{d_i \in D_w} \alpha^{t_{d_i}-1} p(z|d_i) \tag{5.5}$$

Daily Profile We build the daily user profile of u using relevant documents D_d extracted from the search history of u in the current day as follows:

$$p_d(z|u) = \frac{1}{K} \sum\nolimits_{d_i \in D_d} \alpha^{t_{d_i}-1} p(z|d_i) \tag{5.6}$$

Session Profile We build the session user profile of u using relevant documents D_s extracted from the current search session of u as follows:

$$p_s(z|u) = \frac{1}{K} \sum\nolimits_{d_i \in D_s} \alpha^{t_{d_i}-1} p(z|d_i) \tag{5.7}$$

Figure 5.4 shows how we use the relevant documents extracted from the search history of a user in different time intervals (i.e., the whole search history, the current searching day and the current search session) to build her long-term, daily and session profiles, respectively.

5.1.3 Re-ranking search results using user profiles

Having built the temporal user profiles, we can utilise the user profiles to re-rank the original list of documents returned by a search engine. Figure 5.5 gives an overview of using user profiles to re-rank the original document list. After re-ranking the document list, we expect that the relevant documents will be promoted to higher ranks. The detailed steps are as follows:

1. For each query, the top n ranked search results (as recorded in a query log collection) returned by the search engine are downloaded. We denote a downloaded web page as d and its rank in the search result list as $r(d)$.

2. We then compute a similarity measure, $Sim(d|p)$, between each web page d and user profile p as in Chapter 4. Because both d and p are modelled as D, P distributions over topic

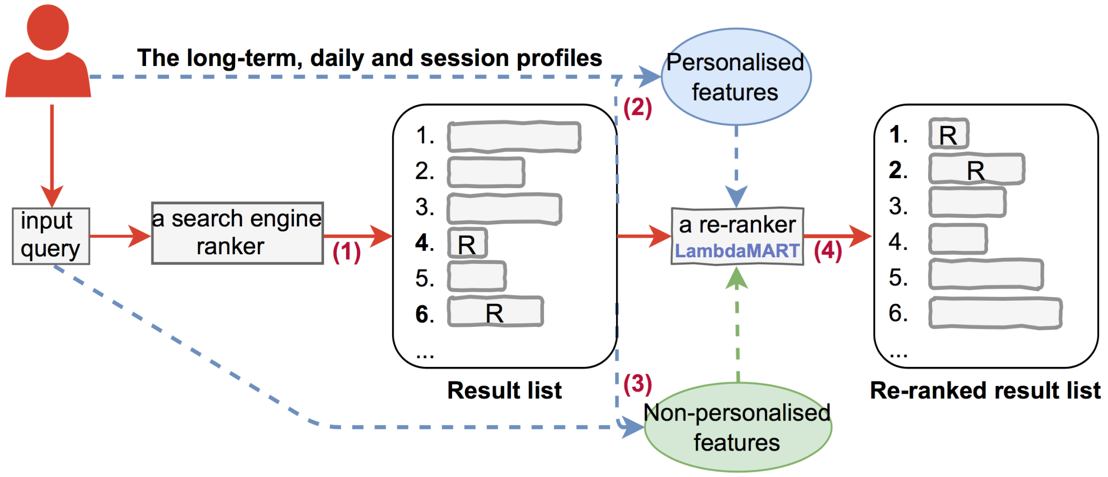


Figure 5.5: The general process of re-ranking. R means that the document is relevant to the user

Z, respectively, we use Jensen-Shannon divergence ($D_{JS}\lfloor.||.\rfloor$) to measure the similarity between the two probability distributions. Jensen-Shannon divergence is a popular method of measuring the divergence (similarity) between two distributions, (as detailed in Chapter 4) which we use as:

$$Sim(d|p) = -D_{JS}\lfloor D||P\rfloor = -\frac{1}{2}D_{KL}\lfloor D||M\rfloor - \frac{1}{2}D_{KL}\lfloor P||M\rfloor \tag{5.8}$$

Here, $D_{KL}\lfloor.||.\rfloor$ is the Kullback-Leiber divergence and $M = \frac{1}{2}(D+P)$. After this step, we receive three personalised scores, denoted as $f_w = Sim(d|p_w)$, $f_d = Sim(d|p_d)$, and $f_s = Sim(d|p_s)$, with respect to long-term, daily, and session profiles respectively. We consider the three scores as the personalised features of the document d.

3. The personalised features only represent the user interest on a returned document. Therefore, apart from these features, we also extract other non-personalised features of input query q and the search result d. The full description of these features is presented in table 5.1.

4. The reason is that the simple fusion-based ranking function in Chapter 4 can only capture the relationship between a single profile and a non-personalised feature (i.e., *DocRank*). It

Table 5.1: Summary of the document features

Feature	**Description**
Personalised Features	
LongTermScore	The similarity score between the document and the long-term profile
DailyScore	The similarity score between the document and the daily profile
SessionScore	The similarity score between the document and the session profile
Non-personalised Features	
DocRank	Rank of the document on the original returned list
QuerySim	The cosine similarity score between the current query and the previous query
QueryNo	Total number of queries that have been submitted to the Search Engine

is not suitable for capturing the relationships between the personalised and non-personalised features detailed in table 5.1. To handle this problem, we employ a learning-to-rank method to automatically learn a ranking function for these proposed features using a training dataset (Section 2.3). After extracting the document features, to re-rank the top n returned URLs instead of using a simple ranking function (as in Chapter 4), we employ a learning to rank algorithm. Our chosen algorithm to train ranking models is LambdaMART (Burges et al., 2006): among many learning to rank algorithms, LambdaMART has been regarded as one of the best-performing algorithms (Burges, 2010) and has been chosen as the base learning algorithm in various state of the art approaches to search personalisation[2] (Bennett et al., 2012; Shokouhi et al., 2013; Song et al., 2014; Wang et al., 2014). Although we use LambdaMART in our experiments, any reasonable learning-to-rank algorithm should produce similar results as our proposed features are insensitive to ranking algorithms.

[2] Indeed, an ensemble of LambdaMART rankers won Track 1 of the 2010 Yahoo! *Learning to Rank Challenge* (Chapelle et al., 2010).

5.2 Experimental methodology

5.2.1 Dataset and evaluation methodology

Dataset In the experiment, we evaluate the approaches using the search results produced by the Bing search engine. As described in Chapter 3, the dataset used in our experiments is the query logs of 1166 anonymous users over four weeks, from 01^{st} July 2012 to 28^{th} July 2012. Each sample in the query logs consists of: an anonymous user identifier, an input query, the query time, top 10 returned URLs and clicked results along with the user's dwell time. We also download the content of these URLs for learning the topics.

The whole dataset was partitioned into profiling, training and test sets. The profiling set is used to build the long-term user profile, the training set is for training the ranking model using LambdaMART, and the test set is used for evaluation of the approaches. In particular, the profiling set contains the log data for the first 13 days; the training set contains the query logs for the next 2 days, and the test set contains the log data for the remaining 13 days. Table 5.2 shows the basic statistics on the three datasets.

Table 5.2: Basic statistics of the evaluation search log set

Item	ALL	Profiling	Training	Test
#day	28	13	2	13
#query	520,010	240,066	29,834	250,110
#distinct query	176,029	85,641	12,112	89,445
#search session	94,972	43,462	5,655	45,886
#click	433,277	200,119	25,805	207,353
#SAT click	334,227	154,753	19,513	159,961
#SAT click/#query	0.6427	0.6446	0.6541	0.6760

Evaluation Methodology For evaluation, we use the SAT criteria (Fox et al., 2005) to identify the satisfied clicks (SAT click) from the query logs. We assign a positive (relevant) label to a returned URL if it is a SAT click. In common with (Bennett et al., 2012), we also assign a positive label to an URL if it is a SAT click in one of the repeated or modified queries in the same search session[3]. The remainder of the top-10 URLs are each assigned negative (irrelevant) labels. We use the rank positions of the positive labelled URLs as the ground truth to evaluate the search performance before and after re-ranking. We also apply some simple pre-processing on these datasets as follows. We first remove from the dataset, any queries whose positive label set is empty as we cannot evaluate the performance on those queries after re-ranking using the evaluation metrics detailed in Chapter 3.[4] Next, we discard the domain-related queries (e.g. Facebook, Youtube). Finally, we normalise the relevance features (both personalised and non-personalised features) to have mean of zero and standard deviation of one (i.e., z-score) from the training set.

5.2.2 Experimental settings

Personalisation Methods and Baselines We empirically investigate the effect of different temporal aspects in latent topic-based personalisation by using the three proposed profiles and their combination to generate the following features:

1. LongTermScore from the long-term profile (LON)
2. DailyScore from the daily profile (DAI)
3. SessionScore from the session profile (SES)
4. AllScore from the combination of all three profiles (ALL)

We further combine these features with the non-personalised features to enrich the personalisations with relevant information from all users. As mentioned earlier, our first baseline, named as *Default*, is the search results (ranking of URLs) returned by the commercial search engine, where we obtain the log data. The second baseline we would like to compare with is the combination of non-personalised features and the topic features as in Chapter 4, which does not take the temporal features into account. We named the second baseline *Non-temporal*.

[3] q' is a modification of q if the returned URLs (top 10) of q' contains at least one SAT click of q.

[4] We note that this might not be representative of a real world scenario in which a user can get the answer directly from the result list without any click, such as, weather related queries.

We now present the setting of LDA and LambdaMART for learning the topics and for learning the ranking function respectively. Note that in order to make a fair comparison we use the same topic distributions for all personalisation approaches and baselines.

LDA & LambdaMART We train the LDA model on the relevant documents extracted from the query logs, as detailed in section 5.1.1. For model selection, we apply the tuning approach proposed by Wallach et al. (2009) to automatically refine the hyper-parameters α and β. The number of topics is decided by using a held-out validation set which consists of 10% of all the relevant documents. The selected number of topics is the one that gives the lowest perplexity value. We also use the validation set to select the temporal weighting parameter α. Table 5.3 shows the ten most probable topical words in topics trained using LDA.

Table 5.3: The ten most probable topical words in topics trained using LDA

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
earth	march	mary	news	intelligence
space	june	william	facebook	artificial
characters	april	james	read	computer
make	july	john	blog	human
robot	october	thomas	comments	neural
moon	january	lee	post	brain
sun	february	elizabeth	twitter	problems
star	november	george	share	problem
solar	august	robert	latest	neurons
planet	september	mae	media	function

The ranking function is learned using LambdaMART. After getting the features from the approaches, we randomly extract 10% of the training set for validation. We used the default setting for LambdaMART's prior parameters[5]. We follow the same model selection process as

[5] Specifically, number of leaves = 10, minimum documents per leaf = 200, number of trees = 100 and learning rate = 0.15.

in Bennett et al. (2012); Shokouhi et al. (2013).

Evaluation metrics The evaluation is based on the comparison between our personalised approaches and the baselines. We use four evaluation metrics which are: Mean Average Precision (*MAP*), Precision at k (*P@k*), Mean Reciprocal Rank (*MRR*) and Normalized Discounted Cumulative Gain at k (*nDCG@k*) (detailed in Chapter 3). These are standard metrics which have been widely used for performance evaluation in document ranking (Manning et al., 2008). For each evaluation metric, a higher value indicates a better quality of ranking.

5.3 Experimental results

5.3.1 Overall performance

In this experiment, we analyse the effect of temporal aspects on latent topic profiles as proposed in Section 5.1 using the six metrics: *MAP*, *P@1*, *P@3*, *MRR*, *nDCG@5* and *nDCG@10*. Table 5.4 shows promising results when the temporal features are used to build user profiles. One can see that all three temporal profiles (i.e., the session, daily and long-term profiles) have led to improvements in the original ranking and the use of the non-temporal profile. In particular, the combination of all features (ALL) achieves the highest performance. This interesting result shows that a comprehensive user profile should capture different temporal aspects of the user's history (i.e., it should capture both long-term and short-term temporal aspects). We note that the improvements over the baselines reported in table 5.4 are all significant with paired t-test of $p < 0.01$.[6] The MAP value of the Bing default ranker was 0.7494 out of 1 indicating that the relevant documents were largely (more than 82%) in the first half of the original lists. Therefore, the improvements after re-ranking were not based on chance.

In the comparison between the temporal profiles, table 5.4 shows that the session profile (SES) achieves better performance than the daily profile (DAI). It also shows that the daily profile (DAI) gains an advantage over the long-term profile (LON). This indicates that the short-term profiles capture more details of user interest than the longer ones. The results are

[6] The exact p values are reported in Tables B.2 and B.3, Appendix B.

Table 5.4: Overall performance of the methods. The differences between the baselines and the four models of using the temporal profiles are all statistically significant according to a paired t-test ($p < 0.01$)

Models	*MAP*	*P@1*	*P@3*	*MRR*	*nDCG@5*	*nDCG@10*
Default	*0.7494*	*0.6471*	*0.3320*	*0.7699*	*0.7805*	*0.8197*
Non-temporal	*0.7460*	*0.6464*	*0.3289*	*0.7683*	*0.7751*	*0.8175*
LON	0.7577	0.6601	0.3377	0.7813	0.7911	0.8267
DAI	0.7760	0.6897	0.3473	0.8016	0.8080	0.8406
SES	0.7936	0.7207	0.3537	0.8214	0.8238	0.8540
ALL	**0.7964**	**0.7283**	**0.3543**	**0.8254**	**0.8251**	**0.8563**

also consistent with what has been found in Bennett et al. (2012). The difference is that our profiles are based on the learned latent topics while they use the ODP.

5.3.2 Performances on different query click entropies

In search personalisation, click entropy plays an important role in deciding the search performance. Dou et al. (2007) have argued that a small click entropy may deteriorate the quality of the search results. Similar to Chapter 4, the click entropy of a query is defined as:

$$QueryClickEntropy(q) = \sum\nolimits_{d \in D_q} -p(d|q) \log_2 p(d|q) \tag{5.9}$$

Here D_q is a collection of web pages which are clicked for the distinct query q, and $p(d|q)$ is the percentage of the clicks on document d among all the clicks for q. A smaller query click entropy value indicates more agreement between users who click a small range of web pages. In this chapter, we are also interested in investigating the effect of the click entropy on the performance of the temporal latent topic profiles. Figure 5.6 shows the distribution over click entropies in the experimental data. In the experimental data, about 67.25% and 16.34% queries have a low click entropy from 0 to 0.5 and from 0.5 to 1 respectively; 10.05% and 3.95% queries have a click entropy from 1 to 1.5 and from 1.5 to 2 respectively; and only 2.41% queries have

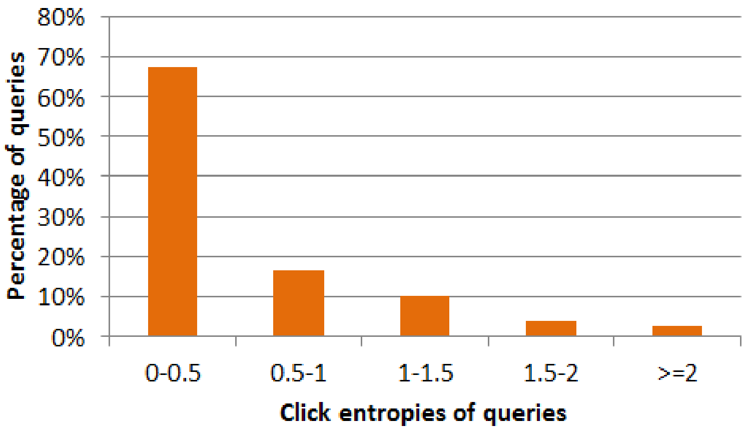


Figure 5.6: Distribution of query click entropy

a high click entropy (≥ 2).

In Figure 5.7, we show the improvement of the temporal profiles over the *Default* ranking from the search engine in term of *MAP* metric for different magnitudes of click entropy. Here the statistical significance is also guaranteed with the use of the paired t-test ($p < 0.01$).[7] The results show that when users have more agreement over clicked documents, with respect to a smaller value of click entropy, the re-ranking performance is only slightly improved. For example, with click entropy between 0 and 0.5, the improvement of the *MAP* metric from the long-term profile is only 0.39%, in comparison with the original search engine. One may see that the effectiveness of the temporal profiles increases proportionally according to the value of click entropy. In particular, the improvement of personalised search performance increases significantly when the click entropy becomes larger, especially with click entropies ≥ 0.5, and the highest improvements are achieved when click entropies are ≥ 2. This result contributes a case study on temporal latent topic profiles to the study of click entropy for personalisation besides the non-temporal latent topic profile (Chapter 4).

[7] The exact p values are detailed in Table B.4, Appendix B.

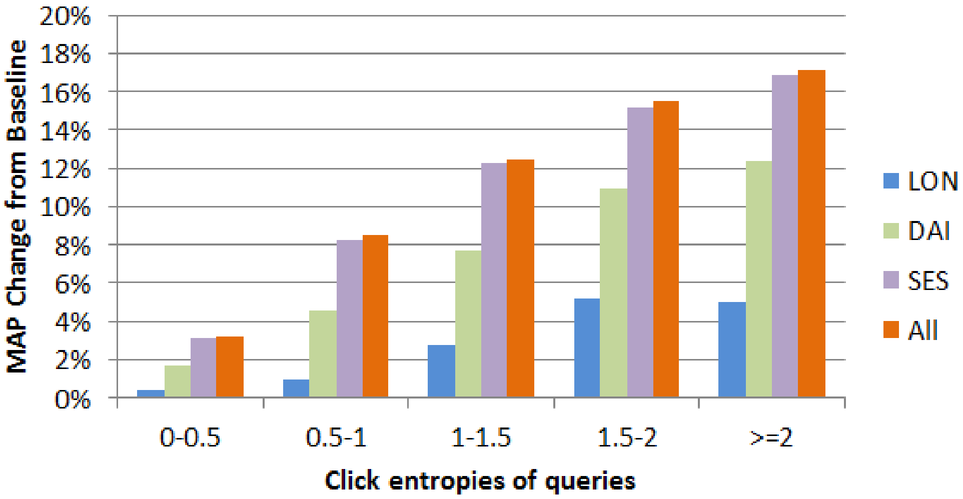


Figure 5.7: Search performance improvements over *Default* with different click entropies

5.3.3 Performances on different query positions

A query usually has a broader influence in a search session than only returning a list of URLs. The position of a query in a search session is also important because it may be fine-tuned by a user after the unsatisfactory results from previous queries. Therefore, in order to get into the insights of the user's information need, a search engine should take into account the position of an input query in a search session. In this experiment, we aim to study whether the position of a query has any effect on the performance of the temporal latent topic profiles. For each session, we label the queries by their positions during the search. The first five queries are numbered from one to five according to the order of the time that they have been entered to the search engine, the remaining queries are labelled as ≥ 6, similarly as in Bennett et al. (2012).

We show the *MAP* performances of the temporal latent topic profiles for different query positions in Figure 5.8. From the *MAP* values, we can see that the first query always received higher satisfaction than the others. It shows that the advanced search engine where we extracted the logs has managed to produce reasonably relevant results at the first query. The higher query positions achieve a smaller value of *MAP* in a search session, which can be explained as users tend to search for supplementary information after the first query, and that the latter queries are so similar to the previous one that the search results contain many URLs which have already

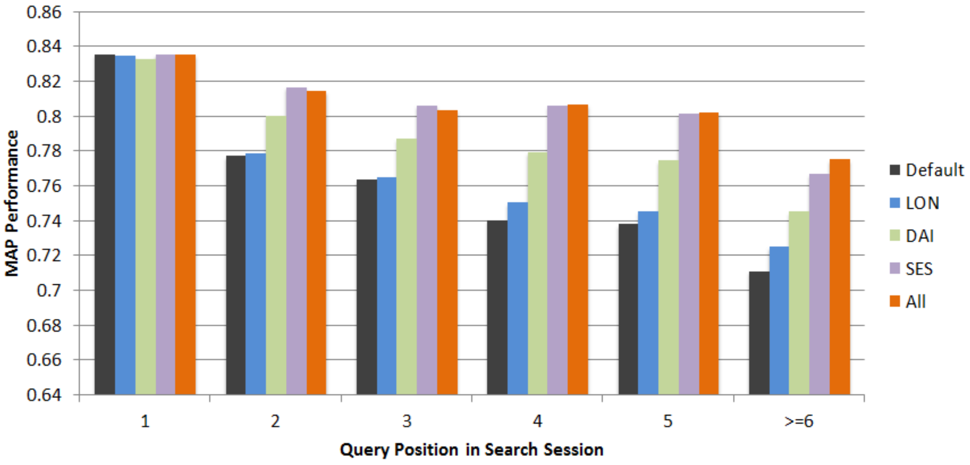


Figure 5.8: Performances of the methods by position of query in search session

appeared in the previous search result. Our result is consistent to what has been mentioned in White et al. (2013).

Note that we cannot build a session profile for the first query because there is no previously observed relevant document for the query. For long-term and daily profiles, we found that their search performances are similar to the search engine performance of the first query. This can be explained by the fact that the single long-term and daily profiles are diverse and cannot sufficiently represent the user recent interests for the first query. Furthermore, as shown in Figure 5.8, the search engine satisfies most the user's information need for the first query (MAP value of 0.8353 out of 1). However, for the next queries in the search session, the temporal latent topic profiles show a significant improvement. It shows that temporal profiles can quickly adapt to represent the user interest. For example, the session profile achieves the highest performance in the second and the third queries in a session while the combination of profiles outperforms the other models on the queries from the fourth positions. This new result is interesting because it shows that the temporal features can help tuning the search performance in further queries which has not been done successfully by the original search engine.

5.4 Conclusions

We have presented a study on the temporal aspects of building user profiles with latent topics learned from the relevant documents. For each user, we used relevant documents at different time scales to build long-term, daily, and session profiles. Each user profile is represented as a distribution over latent topics from which we extract the features and combine them with non-personalised features to learn a ranking function using LambdaMART. We performed a set of experiments to study the effectiveness of the temporal latent topic-based profiles. Our experiments provide answers to the main research question raised at the beginning of this chapter:

RQ 3 *How can we build temporal user profiles for search personalisation?*

To answer the question, we worked with the larger-scale query logs of 1166 searchers from the Bing web search engine detailed in Chapter 3. The results showed that the temporal user profiles help improve the search performance over the competitive ranker of the Bing search engine and the non-temporal user profile. We also found that the session profile captures the most interests of a user and is able to generate helpful features for learning the re-ranking function. The best performance was achieved by the combination of all three temporal profiles, indicating that a good personalisation should take into account all temporal aspects from user's search history. Other experimental results confirmed that the impact of the query's click entropy on the temporal latent topic profiles is similar to that on the non-temporal user profiles. Finally, another interesting finding is the usefulness of the temporal profile in tuning the search results for the next queries in a session.

The experimental results indicate that the session user profile helps to improve the search performance over the competitive ranker of the Bing search engine and even performed better than the longer-term profiles (i.e., long-term and daily profiles). However, in the context of the web search logs, a user may handle several *search tasks* within a search session and submit several queries within a search task (Liao et al., 2012). As described in Liao et al. (2012), a search task is an atomic (single) user information need. Table 5.5 shows an example of a search session containing two different search tasks (one is related to *gnats*, and another one is related to *lamb consumption*).

Table 5.5: An example of two search tasks within a search session

No.	**Query**	**Click**	**Query Time**	**Task ID**
1	gnats	ehow.com/about...gnats	12:33:17	1
2	types of gnats	ehow.com/info...gnats	12:34:53	1
3	buffalo gnats	thetelegraph.com/...gnats	12:36:24	1
4	*lamb consumption in US*	chacha.com...	12:57:58	2
5	lamb consumption in UK	guardian.co.uk...	12:58:59	2

In this case, a single temporal user profile for the search session might not quickly capture the user's interests in the current search task, and even harm the search performance. Specifically, at the time the user submitted the query "lamb consumption in US", the user profile was built using all the user's relevant documents related to "gnat". The user profile cannot capture the user's current interests in the "lamb consumption" topic. This might even harm the search performance if we use the profile to re-rank the search result list returned by the search engine for the query "lamb consumption in US". We assume that the user profile should be built using the information of the user's current search task. To handle this problem, in the next chapter, we employ our current temporal user profiling methodology to model the user's search tasks.

Chapter 6

Modelling Search Tasks for Search Personalisation

In Chapter 5, we introduced an approach to building three temporal user profiles (i.e., long-term profile, daily profile and session profile) using the user's historical search interactions in different search intervals. We then applied the temporal profiles to re-rank search results returned by the Bing search engine using a learning-to-rank mechanism. The experimental results showed that the temporal session profile, which is constructed using the user's interactions in that *search session*, helps most to improve the ranking quality significantly. However, in the web search context, a user might handle several *search tasks* within a search session and submit several queries within a search task (Liao et al., 2012). A *search task*, that is an atomic information need, is defined (Liao et al., 2012; White et al., 2013; Wang et al., 2014; Li et al., 2016) as a set of queries serving the same *information need*.

Table 6.1 shows an example of a search session in which the user submitted queries and clicked on documents to handle two different search tasks (i.e., related to gnats and lamb consumption, respectively). In the example, if we only build a single profile for the search session, that profile might not adapt quickly enough to capture the user's current search interests. For example, we assume that the fourth query (i.e., "lamb consumption in US") is the current query. But because the temporal session profile (detailed in Chapter 5) is built using the previously relevant documents (i.e., related to the "gnats" task) in the search session, that profile does not capture the current user interest in the "lamb consumption" task. This might even harm the search performance if we use that session profile to personalise search results for the current query "lamb consumption in US". In this chapter, we assume that the user profile should be better built in response to the user current search task rather than the user current search session.

Table 6.1: An example of two search tasks within a search session

No.	**Query**	**Click**	**Query Time**	**Task ID**
1	gnats	ehow.com/about...gnats	12:33:17	1
2	types of gnats	ehow.com/info...gnats	12:34:53	1
3	buffalo gnats	thetelegraph.com/...gnats	12:36:24	1
4	*lamb consumption in US*	chacha.com...	12:57:58	2
5	lamb consumption in UK	guardian.co.uk...	12:58:59	2

Recent research has shown that mining and modelling the search tasks can help improve the performance of web search personalisation (White et al., 2013; Wang et al., 2014). However, the dynamic nature of search tasks is largely ignored in previous search task modelling studies (e.g., the search task intent may be generalised or specialised over the searching time), that is, they treated all the user's relevant documents *equally* (White et al., 2013). Figure 6.1 shows an example of a search task related to the "bluebell" flower, in which the user's search intent was shifting from "what bluebell flower is?" to "where a nearby bluebell wood is?"

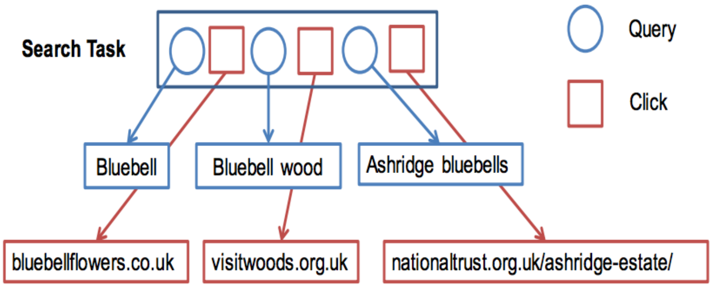


Figure 6.1: A "bluebell" related search task

To tackle these problems, we argue that the user's search task should be modelled in a dynamic way so that it can quickly capture the user's current search interests. Accordingly, this chapter aims at answering the following question:

RQ 4 *How can we model search tasks for search personalisation?*

To answer this research question, we propose a personalisation framework in which we adapt the temporal profiling methodology (e.g., the temporal weighting scheme) proposed in Chapter 5 to model search tasks. Specifically, we utilise latent topics automatically derived by LDA from the *relevant documents* of a user's *search task* to model that search task. In the experiments, we utilise the temporal search task to re-rank the result list returned by the Bing search engine. Experimental results show that modelling search task using the temporal user profiling methodology helps to improve the ranking quality significantly over the static search task (without using the temporal aspects) as well as the session-based user profile.

The remainder of this chapter is structured as follows. Section 6.1 describes our personalisation framework for modelling the temporal search tasks and using the temporal search tasks to re-rank the result list returned by the Bing search engine. In Section 6.2, we describe the experiment setting. We then report the results in Section 6.3 and conclude the chapter in Section 6.4.

6.1 Personalisation framework

Our main focus in this chapter is to model the user's search tasks. First, we utilise a state-of-the-art approach, Query Task Clustering, to extract search tasks from sessions. After obtaining all the search tasks, we extend the temporal user profiling methodology in Chapter 5 to model each search task in Section 6.1.2. The search task is finally utilised in a learning-to-rank mechanism to personalise the search results returned by a search engine in Section 6.1.3.

6.1.1 Clustering search tasks in a search session

To identify search tasks from each search session, we apply a state-of-the-art approach, that is, Query Task Clustering (QTC) (Liao et al., 2012), which has the desirable feature of extracting interleaved tasks within a session.

In general, the method works in two steps: first, it measures the similarity between query pairs to build an undirected graph of queries within each search session. Second, queries within a search session are grouped into tasks via a clustering algorithm. Figure 6.2 shows an example of a search session which is classified into two interleaved search tasks using QTC.

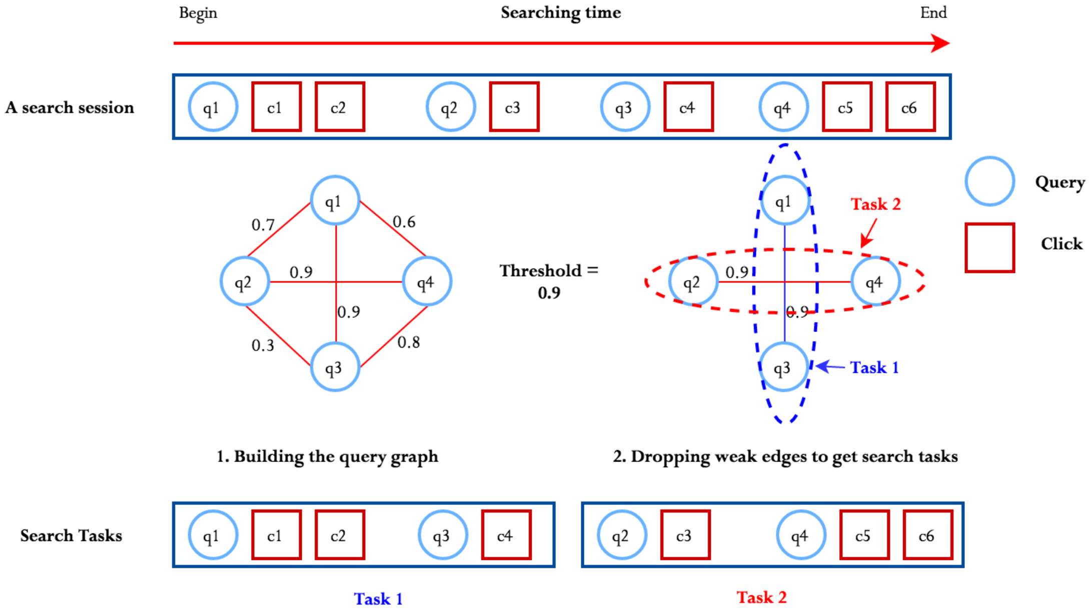


Figure 6.2: Identifying search tasks from a search session using QTC

To calculate the similarities of a query pair, QTC used time and word-based features and applied a supervised learning approach to learning the weight of the features (Liao et al., 2012). Specifically, they constructed a labelled dataset by asking human annotators to assign binary labels to a set of randomly sampled query pairs. A query pair has a positive label if these annotators think they are related using the predefined criteria (e.g., they are repeated). On the other hand, a negative label is assigned if the two queries are not related or contain different atomic information need (e.g., “Seattle city” and “space needle”). A supervised classifier is then learnt to determine the weight of the features. Table 6.1.1 details the QTC features of query pairs and the learned weights of these features, which are also used in our experiments to extract search tasks from a search session.

Table 6.2: The QTC features of query pairs and corresponding weights

Feature Description	Learned Weight
Temporal features	
timediff_1: time difference in seconds	-0.1121
timediff_2: category for 1/5/10/30 minutes	-0.0623
Word features	
lv_1: Levenshtein distance of two queries	0.0106
lv_2: lv_1 after removing stop-words	-0.1951
prec_1: average rate of common terms	-0.2870
prec_2: prec_1 after removing stop words	1.2058
prec_3: prec_1 (If term A contains B, A=B)	0.5292
rate_s: rate of common characters from left	1.6318
rate_e: rate of common characters from right	0.4014
rate_l: rate of longest common substring	0.4941
b_1: 1 if one query contains another, else 0	0.6361

Using the learned query similarity function, QTC then constructs a undirected graph of queries within a search session. The vertices of the graph are queries and the edges are the similarity scores between queries. By dropping the edges where the similarities are smaller than a threshold (0.9 in our case), we get all search tasks as connected queries of the graph together with the user clicked documents of these queries.

6.1.2 Building a temporal search task

After identifying search tasks within sessions, we need to model the tasks in a dynamic way so that the user's current search interest is quickly captured and represented. Because the search intent and user interests may change during a search task, we propose to use the method of building temporal user profiles (Chapter 5) to build the temporal representation of the search task. The search result click behaviour is readily available in the query logs. We use that click information of a search task to model the representation of that task.

Specifically, for each search task, we extract the relevant data of that task. After that, we employ Latent Dirichlet Allocation (LDA) to automatically derive latent topics (Z) from the relevant documents of all search tasks.

By utilising the temporal user profiling methodology in Chapter 5 (i.e., the temporal weighting scheme), we build the temporal representation of a search task as a multinomial distribution over topic Z. Formally, we denote the search task set as S. Let s denote an instance of S. Let $D_s = \{d_1, d_2, .., d_n\}$ be a relevant document set for the search task s. We model the task s (given D_s) as a distribution over the topics Z (i.e. $P(Z|s)$. Furthermore, since the search task intent may change over time, the more recent relevant documents express more about the current search intent than the distant ones. This characteristic can be captured by introducing a decay function as in Chapter 5. In this chapter, we model the probability of a topic z given s (i.e., $p(z|s)$) as a mixture of probabilities of z given relevant document $d_i \in D_s$ as follows:

$$p(z|s) = \frac{1}{K} \sum_{d_i \in D_s} \alpha^{t_{d_i}-1} p(z|d_i) \tag{6.1}$$

where $\alpha^{t_{d_i}}$ is a exponential decay function of t_{d_i}; t_{d_i} is the time the searcher clicked on the document d_i within s; $t_{d_i} = n$ indicates that d_i is the n^{th} recent relevant document; and $K = \sum_{d_i} \alpha^{t_{d_i}-1}$ is a normalisation factor. α is the decay parameter ($0 < \alpha \leq 1$). $\alpha = 1$ means that all relevant documents are treated equally without considering the temporal aspect (i.e., the viewed time).

Figure 6.3 shows an example of modelling a search task. The user's current search task has three relevant documents extracted from her query logs by applying the SAT criteria. After using LDA to derive topics (i.e., "Flower", "Shopping", "Delivery" and "Place To Visit"), each document is modelled as a distribution over these topics. Then the temporal representation of the search task is constructed as a distribution over the topic set, in which the probability of each topic is calculated using Eq.6.1 with the decay parameter of $\alpha = 0.9$. For example, the "Place" topic has the highest value, from

$$\mathit{Task\_Place} = \frac{0.9^2 \times 0.13 + 0.9^1 \times 0.45 + 0.9^0 \times 0.57}{0.9^2 + 0.9^1 + 0.9^0} = 0.40$$

, which means that in the search task the user is currently most interested in the "Place" topic, which is the user's current search interest.

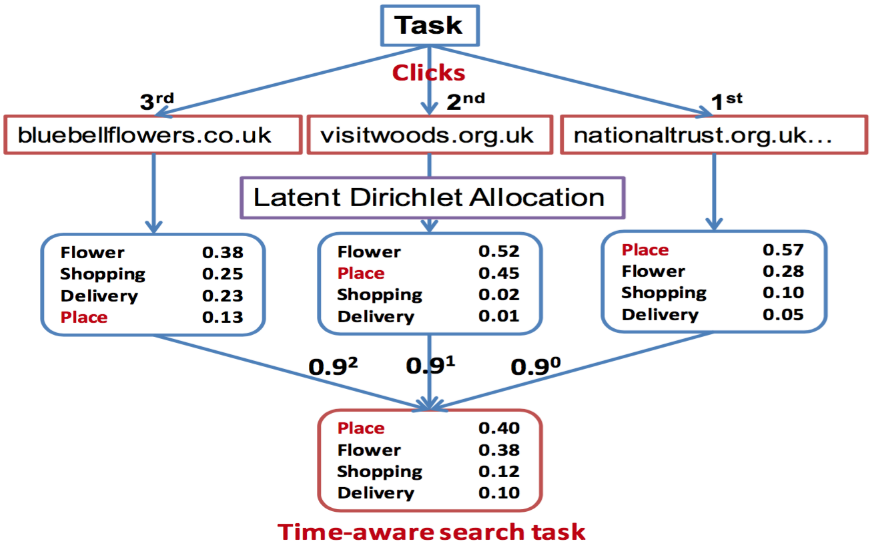


Figure 6.3: The temporal representation of a search task with the decay parameter $\alpha = 0.9$

6.1.3 Re-ranking search results using temporal search tasks

After modelling search tasks, we utilise the temporal representation of the search tasks to re-rank the original list of documents returned by a search engine. Figure 5.5 shows an overview of using search tasks to re-rank the original document list. After re-ranking the document list, we expect that the relevant documents will be promoted to higher ranks. For each query and a search task containing that query, the detailed steps are as follows:

1. For each input query q, we utilise the current search task s, to which the query belongs, to re-rank the first n documents returned by the Bing search engine. We denoted a downloaded web page d and its rank in the search list as $r(d)$. s represents the topical interests of the user in the current search task.

2. In a similar way to the descriptions in chapters 4 and 5, for each returned document d, we compute a similarity measure, *TaskScore*, between d and s . Specifically, as both

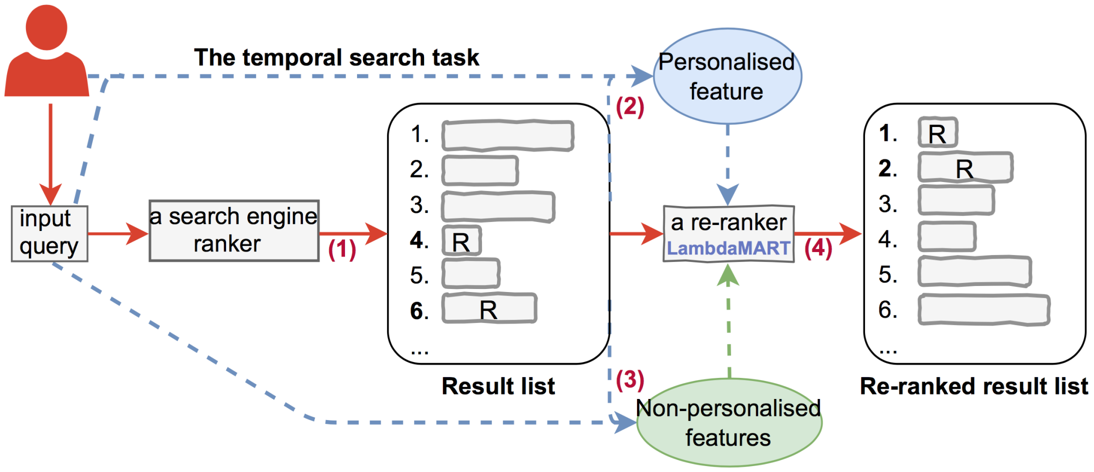


Figure 6.4: The general process of re-ranking. R means that the document is relevant to the user

d and s are models as $P(Z|d)$ and $P(Z|s)$ distributions over topic Z, respectively, we use Jensen-Shannon divergence to measure the similarity between the two distributions (i.e., $TaskScore = -D_{JS}\lfloor.||.\rfloor$) which is a popular method of measuring the divergence (similarity) between two distributions (as detailed in chapter 4), to measure the similarity between the two probability distributions as follows:

$$TaskScore(d|s) = -D_{JS}\lfloor D||S\rfloor = -\frac{1}{2}D_{KL}\lfloor D||M\rfloor - \frac{1}{2}D_{KL}\lfloor S||M\rfloor \tag{6.2}$$

Here $D_{KL}\lfloor.||.\rfloor$ is the Kullback-Leiber divergence and $M = \frac{1}{2}(D + S)$. Now for each returned document d and a search task s, we get the personalised score $TaskScore(d|s)$. We consider the score as the *personalised feature*.

3. The personalised features only represent the user interest on a returned document. Therefore, apart from that feature, we also extract other *non-personalised features* of q and d. Table 6.3 describes the features.

4. After extracting the document features, to re-rank the top n returned documents, we employ a learning-to-rank (L2R) algorithm (i.e., LambdaMART) to train ranking models as in Chapter 5. Among many L2R algorithms, LambdaMART has been chosen as the base

Table 6.3: Summary of the document features

Feature	**Description**
Personalised Feature	
TaskScore	The similarity score between the returned document and the search task
Non-personalised Features	
DocRank - $r(d)$	Rank of the document on the original returned list
QuerySessionSim	The cosine similarity score between the current query and the previous query in the same search session
QuerySessionNo	Total number of queries that have been submitted to the Search Engine in the search session
QueryTaskSim	The cosine similarity score between the current query and the previous query in the same search task
QueryTaskNo	Total number of queries that have been submitted to the Search Engine in the search task

learning algorithm in various state of the art approaches to search personalisation (Bennett et al., 2012; White et al., 2013). Although we use LambdaMART in our experiments, any reasonable L2R algorithm would likely produce similar results as our proposed features are insensitive to ranking algorithms.

6.2 Dataset and evaluation methodology

6.2.1 Dataset

We use the preprocessed Bing query logs that have been described in Chapter 3 and used in Chapter 5. A major difference from the dataset used in Chapter 5 is that in this chapter, we use the user behaviours in a search task (rather than in a search session) to identify the user's relevant document. Specifically, we use only those with a dwell time exceeding 30 seconds (SAT clicks) as in White et al. (2013)[1]. We assign a positive (relevant) label to a returned URL if it is a SAT click. The remainder of the top-10 URLs are assigned negative (irrelevant) labels. We use the rank positions of the positive labelled URLs as the ground truth to evaluate the search performance before and after re-ranking. As in Chapter 5, we apply a simple pre-processing on these datasets as follows: we remove the queries whose positive label set is empty from the dataset as we cannot evaluate the performance on those queries after re-ranking using the evaluation metrics detailed in Chapter 3. After that, we discard the domain-related queries (e.g.

[1] We did not use the last click in a *search session* as in Chapter 5 as it is based on the user behaviours in a search session rather than in a search task

facebook, youtube). Finally, similar to Chapter 5, we normalise the relevance features (both personalised and non-personalised features) to zero mean and a standard deviation of 1 (i.e., z-score).

The dataset is the four weeks of logs gathered from 01 July 2012 to 28 July 2012 of 1166 randomly-sampled users. We also download the content of these URLs for the learning of the topics. We then partition the whole dataset into the training and test sets. The training set contains the log data of the first 15 days (from 01 to 15 July 2012). The test set contains the log data of the 13 remaining days (from 16 to 28 July 2012). We use the Query Task Clustering approach for identifying search tasks within search sessions. There were about 95,000 search sessions and 231,000 search tasks from the dataset. This represents a mean of 2.43 tasks per session, and 2.25 queries per task which are higher than that reported in White et al. (2013) (i.e., 1.36 and 1.86, respectively). Figure 6.5 shows that around 90% of sessions have no more than three search tasks; specifically, more than 51% of sessions have only one search task. This shows that although most sessions comprise a single task, there are still a large number of sessions (nearly 49%) containing multiple tasks. Since using all in-session activity may result in a noisy relevance signal, we may need to consider in-session task boundaries. We then explore the effect of using full-session search activity (i.e., the temporal session user profile in Chapter 5) versus in-session task activity (i.e., the temporal representation of a search task). Table 6.4 shows the basic statistics on the dataset.

6.2.2 Experimental settings

Personalisation Method and Baselines We list all the models to be compared in Table 6.5. Specifically, we name our proposed re-ranking model with temporal search tasks as *TimeTask*. Our first baseline, named as *Default*, is the original ranking of URLs returned by the Bing search engine. Among the three temporal profiles proposed in Chapter 5, the session profile achieved the best search personalisation performance. Moreover, we need to explore the effect of in-session task activity versus full-session search activity. We then use the temporal session profile in Chapter 5 as our second baseline (denoted as *ShortTerm*). The third baseline we want

Table 6.4: Basic statistics of the evaluation search log set

Item	ALL	Training	Test
#day	28	15	13
#query	520,010	269,900	250,110
#distinct query	176,029	95,089	89,445
#search task	231,145	120,047	111,108
Avg. #query/task	2.25	2.25	2.25
#search session	94,972	49,100	45,886
Avg. #task/session	2.43	2.44	2.42
#click	433,277	225,924	207,353
#SAT click	326,768	170,332	156,436
Avg. #SAT click/query	0.63	0.63	0.63

to compare with is the non-temporal search task modelling method (that is, the decay parameter $\alpha = 1$) (denoted as *StaticTask*).

Table 6.5: An overview of personalisation method and baselines

Model	Description
Default	The default ranker of Bing search engine.
ShorTerm	Using the temporal session profile to re-rank the result list.
StaticTask	Using non-temporal search task modelling method to re-rank the result list.
TimeTask	Using the temporal search tasks to re-rank the search result list.

In the following, we present the setting of LDA and LambdaMART for learning the topics and for learning the ranking function respectively. Note that to make a fair comparison, we use the same topic distributions for all personalisation approaches and baselines.

LDA & LambdaMART As in chapter 5, we train the LDA model on the relevant documents extracted from the query logs, as mentioned in Section 6.1.2. For model selection, we apply the tuning approach (Wallach et al., 2009) to automatically refine the hyper-parameters. The

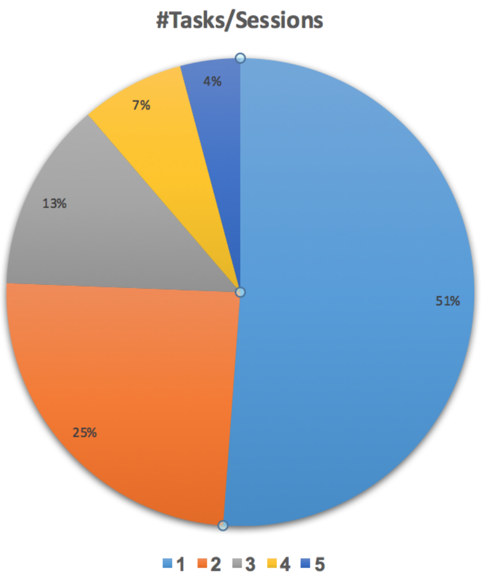


Figure 6.5: Number of search tasks in search sessions

number of topics is decided by using a held-out validation set which consists of 10% of all the relevant documents. The selected number of topics is the one that gives the lowest perplexity value. We also use the validation set to select the temporal weighting parameter α.

The ranking function is learned using LambdaMART. After getting the features from the approaches, we randomly extract 10% of the training set for validation. We used the default setting for LambdaMART's prior parameters[2]. We follow the same model selection process as in Bennett et al. (2012); Shokouhi et al. (2013).

Evaluation metrics The evaluation is based on the comparison between our personalised approaches and the baselines. We use four evaluation metrics which are: Mean Average Precision (MAP), Precision ($P@k$), Mean Reciprocal Rank (MRR) and Normalized Discounted Cumulative Gain at k ($nDCG@k$) (detailed in Chapter 3). These are standard metrics which have been widely used for performance evaluation in document ranking (Manning et al., 2008). For each evaluation metric, a higher value indicates a better ranking quality.

[2] Specifically, number of leaves = 10, minimum documents per leaf = 200, number of trees = 100 and learning rate = 0.15.

6.3 Experimental results

6.3.1 Overall performance

Table 6.6 reports the performance of each model using six metrics as in Chapter 5, that is *MAP*, *P@1*, *P@3*, *MRR*, *nDCG@5* and *nDCG@10* (detailed in Chapter 3). All differences of personalisation models (i.e., ShortTerm, StaticTask and TimeTask) with the *Default* baseline (i.e., Bing default ranker) are statistically significant according to the paired t-test ($p < 0.01$).[3] The MAP value of the Bing default ranker was 0.7369 out of 1 indicating that the relevant documents were largely (more than 80%) in the first half of the original lists. Therefore, the improvements after re-ranking were not based on chance.

Table 6.6 shows that using temporal search tasks, TimeTask achieves a better performance than the original ranking (Default) as well as other strong baselines including non-temporal search task baseline (StaticTask). Especially, the TimeTask achieves the best improvement on the *P@1* value indicating that temporal search tasks help improve the ranking quality of the first ranked document most. Table 6.6 also shows that StaticTask gains an advantage over ShortTerm.

Table 6.7 shows the numbers of better and worse ranks of relevant documents after re-ranking in comparison with the *Default* baseline. As we can see from Table 6.7, by modelling search tasks, the StaticTask model helps to improve the search performance over the ShortTerm model (with the temporal session profile) by significantly decreasing the number of worse ranks. Moreover, the TimeTask model helps to significantly improve the search performance over the ShortTerm model by both decreasing the number of worse ranks and increasing the number of better ranks.

6.3.2 Performances on different query click entropies

Similar to Chapter 5, we evaluate the effectiveness of the temporal search tasks with different query click entropy ranges. Figure 6.6 shows the relative improvement of personalisation models over the *Default* ranking from Bing search engine in term of *MAP* metric with the different

[3] The exact p values are reported in Table B.5, Appendix B.

Table 6.6: Overall performance of the methods. The differences between the baselines and the TimeTask model are all statistically significant according to the paired t-test ($p < 0.01$)

Model	*MAP*	*P@1*	*P@3*	*MRR*	*nDCG@5*	*nDCG@10*
Default	0.7369	0.6368	0.3239	0.7602	0.7649	0.8110
ShortTerm	0.7764	0.6968	0.3470	0.8051	0.8061	0.8415
StaticTask	0.7843	0.7136	0.3488	0.8150	0.8127	0.8477
TimeTask	0.7910	0.7305	0.3502	0.8242	0.8182	0.8530

Table 6.7: Numbers of better and worse ranks after re-ranking in comparison with *Default* and *P-Gain*

Model	#Better	#Worse	*P-Gain*
ShortTerm	11,760	4,989	0.404
StaticTask	11,669	2,779	0.615
TimeTask	13,130	3,087	0.619

magnitudes of click entropy. The statistical significance is guaranteed with the use of paired t-test ($p < 0.01$).[4] We can see that when users have more agreement over clicked documents, with respect to a smaller value of click entropy, the re-ranking performance is only slightly improved. For example, with click entropy between 0 and 0.5, the improvement of the *MAP* metric from the personalisation models are of only 4%, in comparison with the original search engine.

Figure 6.6 also shows that the differences between the task-based models (i.e., TimeTask and StaticTask) and the session-based model (i.e., ShortTerm) are also small with the value of click entropy between 0 and 0.5. However, the effectiveness of personalisation models in general and the task-based models, in particular, increases significantly with the higher click entropy (e.g., [1.5 - 2]). The temporal task-based model (i.e., TimeTask) achieves significantly better performance in term of the ranking quality than both StaticTask and ShortTerm models ($p < 0.01$).

[4] The exact p values are detailed in Table B.6, Appendix B.

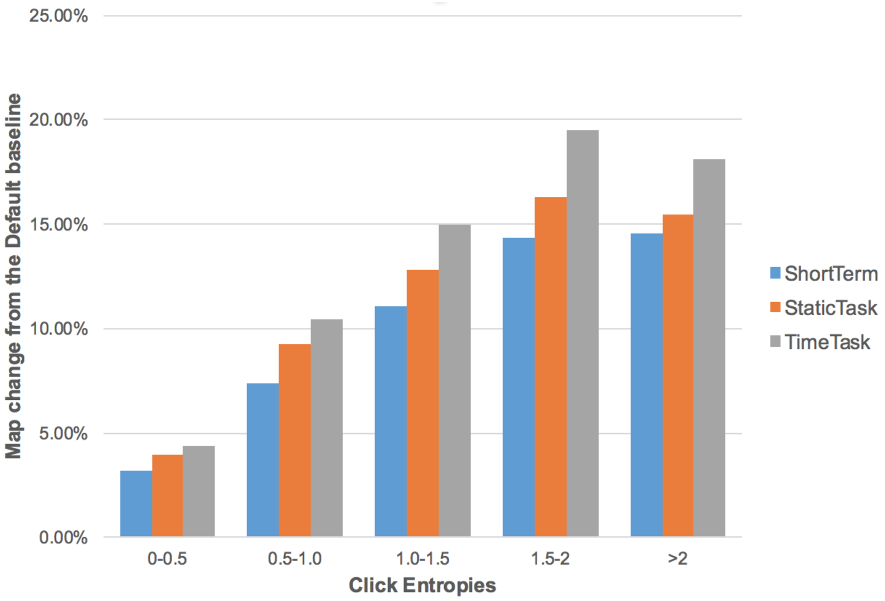


Figure 6.6: Search performance improvements over *Default* with different click entropies

6.4 Conclusions

In this chapter, we have proposed an extension of the temporal user profiling methodology in Chapter 5 to model search tasks extracted from search sessions. Each search task is represented as a distribution over the topics extracted from that task's relevant documents. We then extracted the task-based feature and combine it with non-personalised features to learn a ranking function using LambdaMART. We performed experiments on re-ranking search results returned by the Bing search engine. Our experiments provide answers to the main research question raised at the beginning of this chapter:

RQ 4 *How can we model search tasks for search personalisation?*

To answer the research question, we worked with the Bing query logs of 1166 anonymous users (detailed in Chapter 3) but using a different setting to extract the user's relevant documents. Specifically, we only used the clicked documents with a dwell-time of at least 30 seconds as the relevant documents[5]. In the experiments, we considered two comparative baselines using

[5] We did not use the last clicked document in a search session as in Chapter 5 as it is only the session behaviour rather than the task behaviour

non-temporal search tasks or temporal session user profiles. We found that the ranking quality is improved significantly over the default ranker of Bing. We also found that task-based profiles (i.e., *TimeTask* and *StaticTask*) help significantly improve the search performance over the session-based profile. Furthermore, by capturing the temporal aspect, the search task outperforms the non-temporal search task.

In Chapter 4, we focused on constructing groups of users with shared interests dynamically, and in chapter 5 we focused on building temporal user profiles. In this chapter, we considered temporal search tasks, where we use the temporal profiling methodology in chapter 5 to build temporal search tasks. For these proposed models, we apply them to improve the performance of a commercial web search engine (i.e., Bing) by re-ranking the result list. In the next chapter, we will investigate the adaptation and application of our dynamic profiling technique on another domain (a university Intranet) as well as a different task (personalised query suggestion).

Chapter 7

Personalised Query Suggestion for Intranet Search

In previous chapters, we have explored how to build dynamic user profiles and utilise the profiles to improve the performance of web search personalisation. In this chapter, we will investigate the application of the dynamic user profiles in another search domain, that of an intranet, as well as a new personalisation problem, that of personalised query suggestion. Query suggestion is an important feature in web search engines (e.g., Bing, Google) as well as in domain-specific search engines (e.g., Intranet search) (Adeyanju et al., 2012). Query suggestion helps users quickly refine the input query to better meet the user's information need by recommending possible terms to modify the original input query. Moreover, the Intranet search is fundamentally different from the general Web search, i.e., while web search users are normally interested in getting *some* relevant documents, Intranet search users are mainly looking for specific documents, books, etc. (Hawking, 2010). Moreover, the Intranet may not be fully indexed and accessible by web search engines. For example, web search engines cannot access and index those Intranet documents which require authorised logins. The searcher, therefore, may need the Intranet search engine to get her information need.

Using collective user interaction data (e.g., query logs) for query suggestions has been shown to be useful for Intranet search (Adeyanju et al., 2012; Albakour et al., 2011). Existing query

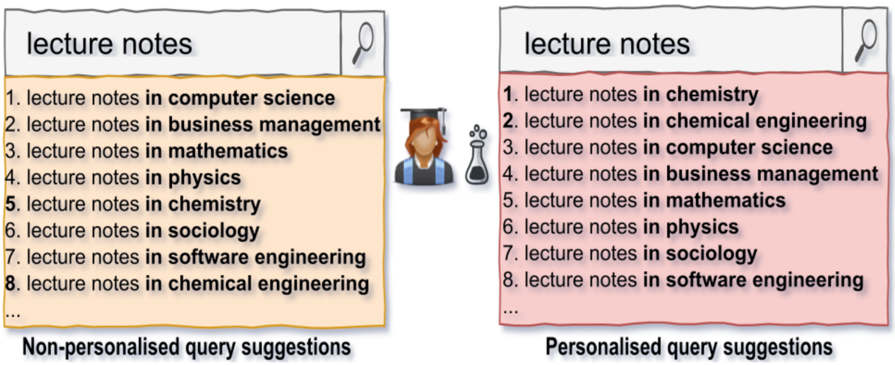

lecture notes
1. lecture notes in computer science
2. lecture notes in business management
3. lecture notes in mathematics
4. lecture notes in physics
5. lecture notes in chemistry
6. lecture notes in sociology
7. lecture notes in software engineering
8. lecture notes in chemical engineering
...
Non-personalised query suggestions
lecture notes
1. lecture notes in chemistry
2. lecture notes in chemical engineering
3. lecture notes in computer science
4. lecture notes in business management
5. lecture notes in mathematics
6. lecture notes in physics
7. lecture notes in sociology
8. lecture notes in software engineering
...
Personalised query suggestions

Figure 7.1: An example of query suggestion lists returned to a chemistry student who submits a query "Lecture Notes" without (left) and with (right) personalisation

suggestion approaches in Intranet search appear to follow a "one-size-fits-all" strategy (Adeyanju et al., 2012; Albakour et al., 2011). That is different users who submit the same query will get the same suggestion list. However, different users may have different topics of interest. Consequently, the users who have submitted the same query may have different search intentions. For example, a chemistry student submitting the query "lecture notes" is likely to be more interested in chemistry classes than computer science classes. We argue that query suggestion should be personalised in the context of Intranet search. Figure 7.1 shows an example of personalised query suggestions. If the system knows that the user is interested in the chemistry topic, it can suggest the query "lecture notes in chemistry" in the top of the suggestion list (right).

Table 7.1: Some basic statistics of the Bing dataset used in Chapters 5 and 6 and the Essex dataset used in this chapter.

Statistic	Bing	Essex
#Sessions	94,972	735,804
#Queries	520,010	1,416,929
#Queries/Session	**5.48**	**1.93**
#Clicks	433,277	930,242
#Clicks/Session	**4.56**	**1.26**

Figure 7.2 shows an example of a search session from an Intranet query log collection. Each

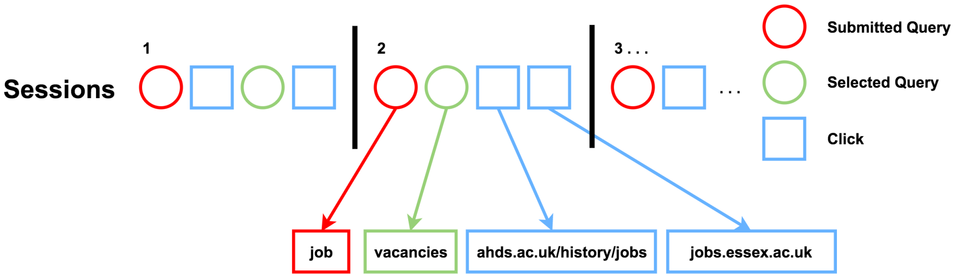


Figure 7.2: Search sessions in Intranet query logs

search session contains two types of event (i.e., queries and clicks). Table 7.1 shows some basic statistics of the Bing (i.e., Web search engine) dataset used in Chapters 5, 6 and the Essex (i.e., Intranet search engine) dataset used in this chapter. We can see that the numbers of queries and clicks per search session in the Intranet search engine are much smaller than that of the Web search engine (i.e., 1.93 versus 5.48 and 1.26 versus 4.56 for the numbers of queries and clicks, respectively). We, therefore, assume that when a user uses the Intranet search engine, she only handles one search task (see Chapter 6) in a search session. That is, the concept of a search session in Intranet search is equivalent to that of a search task. Accordingly, we are interested in answering the following question:

RQ 5 *How can we personalise query suggestion for Intranet search?*

To find the answer to the research question, we propose a unified framework to personalise query suggestion for Intranet search. Specifically, extending the temporal user profiling methodology in Chapters 5, 6, we use the interaction data of each user with the Intranet search engine during a search session to build two user profiles. The user profiles represent the user's topics of interests and may change over time in response to the user's interaction with the system. In our proposed framework, we build two temporal topic-based user profiles for each search session. The first is a *click user profile* based on the clicked documents. The second is a *query user profile* based on the user's query modification history within the search session. We then use the two profiles within a learning-to-rank framework to re-rank suggested queries generated by a non-personalised method for query suggestion on Intranet search (Adeyanju et al., 2012). Experimental results show that our approach significantly improves the query suggestion performance. Note that, in the context of Intranet search, the concepts of a search task and a search session are exchangeable. Moreover, we cannot build the longer-term user profiles (i.e., long-term and daily profiles as in Chapter 5) because the user identifiers are not available in the experimental dataset (i.e., the Essex search log collection).

The rest of this chapter is structured as follows. In Section 7.1 describes our personalisation framework for building the temporal profiles and using the profiles to personalise the query suggestion list. In Section 7.2, we describe our experiment setting and evaluation query logs. We then report the results in Section 7.3 and conclude the chapter in Section 7.4.

7.1 Personalised query suggestion framework

We start the section with building two temporal user profiles for each search session in Section 7.1.1. After that, in Section 7.1.2, we then utilise the temporal user profiles in a learning-to-rank (L2R) mechanism to personalise the query suggestion list returned by a state-of-the-art non-personalised query suggestion algorithm (Adeyanju et al., 2012).

7.1.1 Building temporal user profiles

Each search session contains two types of event (i.e., queries and clicks). Given a search session, we propose to build two temporal profiles for the specific user. These are a click user profile (denoted as $\mathit{profile}(C)$), built using the user's clicked documents, and a query user profile (denoted as $\mathit{profile}(Q)$), built using the user's submitted queries within the session. The click profile has been extensively used in other search personalisation methods (Bennett et al., 2012; Harvey et al., 2013; Vu et al., 2014), so we expect that the query profile will enrich the representation of the user's search interests.

Since a user's interests and search intentions may change over time, the more recently clicked documents and submitted queries could better represent the user's current interests. In this chapter, we propose to use a *decay function* to capture this characteristic as in Chapters 5 and 6.

Extracting Topics from Clicked Documents

Unlike the Bing dataset, we do not have the dwell-time (viewing-time) for each click in the Intranet dataset. We follow (Joachims et al., 2005) in considering that click information (e.g., clicked documents) is a good indicator of those documents' relevance to the user's interests. To build the user profiles, we use the topics discussed in the clicked documents. We first extract clicked documents from the Intranet search's query logs. After that, we employ Latent Dirichlet Allocation (LDA) (Blei et al., 2003) to automatically extract latent topics (denoted as Z) from the clicked documents (denoted as D). After training/inferring LDA model using the clicked documents, each document is described as a multinomial distribution over the topics (denoted as $P(Z|D)$), in which each topic is represented as a multinomial distribution over the entire vocabulary [1].

Building a Click User Profile

We represent the temporal click user profile as a multinomial distribution over the topics as in Chapter 5. Specifically, the user set is denoted as U. Let u be an instance of U. Let $D_c = \{d_{c_1}, d_{c_2}, ..., d_{c_n}\}$ be the set of clicked documents of the user u in the current search session. We define the click user profile of the user u (given the clicked document set D_c) as a distribution over topics Z (denoted as $P_C(Z|U)$). A higher value of $p_C(z|u)$ shows that the user u is more interested in the topic $z \in Z$. $p_C(z|u)$ is defined as a mixture of probabilities of z given $d_{c_i} \in D_c$ as follows:

$$p_C(z|u) = \sum\nolimits_{d_{c_i} \in D_c} \lambda_i p(z|d_{c_i}) \tag{7.1}$$

$\lambda_i = \frac{1}{N}\alpha^{t_{d_{c_i}}-1}$ is the exponential decay function of $t_{d_{c_i}}$, which is the order of the document d_{c_i} clicked by the user u in the search session. $t_{d_{c_i}} = 1$ indicates that d_{c_i} is the most recent clicked document; N is the normalisation factor. α is the decay parameter ($0 < \alpha \leq 1$).

Figure 7.3 shows an example of a click user profile, which is constructed using three clicked documents and a decay parameter $\alpha = 0.9$. We can see that the click profile can capture the user's topic of interest (i.e., "Topic 4").

[1] The set of words which appear in all the clicked documents

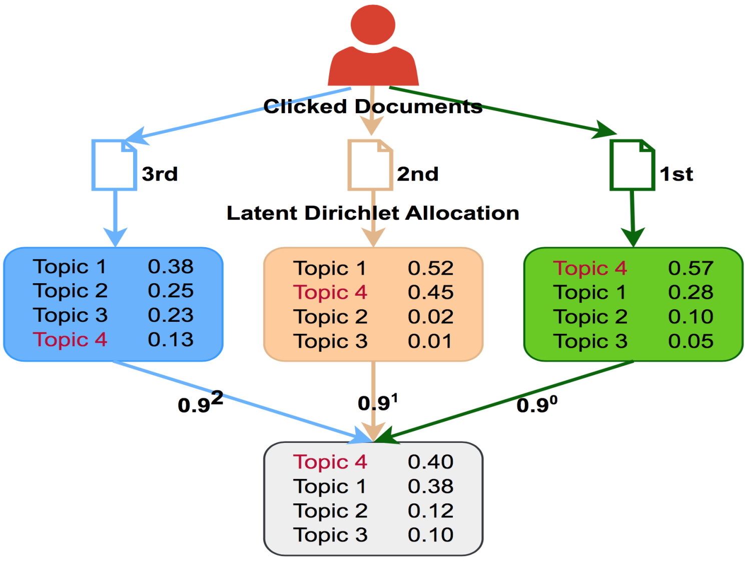


Figure 7.3: Temporal click user profile

Building a Query User Profile

Let $Q = \{q_1, q_2, ..., q_m\}$ be the submitted query set of u in the search session. We represent the temporal query user profile as a multinomial distribution over the topics Z. In order to do that, we, first, need to model each query q_i as a multinomial distribution over the topics Z. Because the number of Intranet documents is small and can be assumed to change less frequently than that of the Web search engine (Hawking, 2010; Adeyanju et al., 2012), we make a simplifying assumption of describing each query by the set of documents that contain the query words, denoted as $D_{q_i} = \{d_{i_1}, d_{i_2}, ..., d_{i_k}\}$. Then, each search query q_i (given the document set D_{q_i}) is modelled as a distribution over topics Z (denoted as $P(Z|q_i)$). The probability of a topic $z \in Z$ given $q_i \in Q$ (i.e., $p(z|q_i)$) is defined as a mixture of probabilities of z given a document $d_{i_j} \in D_{q_i}$ as follows:

$$p(z|q_i) = \sum\nolimits_{d_{i_j} \in D_{q_i}} \frac{1}{|D_{q_i}|} p(z|d_{i_j}) \tag{7.2}$$

Here, $|D_{q_i}|$ is the size of the document set D_{q_i}. Figure 7.4 shows an example of modelling a query as a distribution over topics (Z) using the documents related to that query.[2]

We then model the query user profile of the user u (given the query set Q) as a distribution over topics Z (denoted as $P_Q(Z|u)$). The probability of a topic z given u (i.e., $p_Q(z|u)$) is defined

[2] It is feasible in Intranet search as the size of document corpus is small. However, in our experiments, we use only top n results returned by the Intranet search engine to that query for the efficiency.

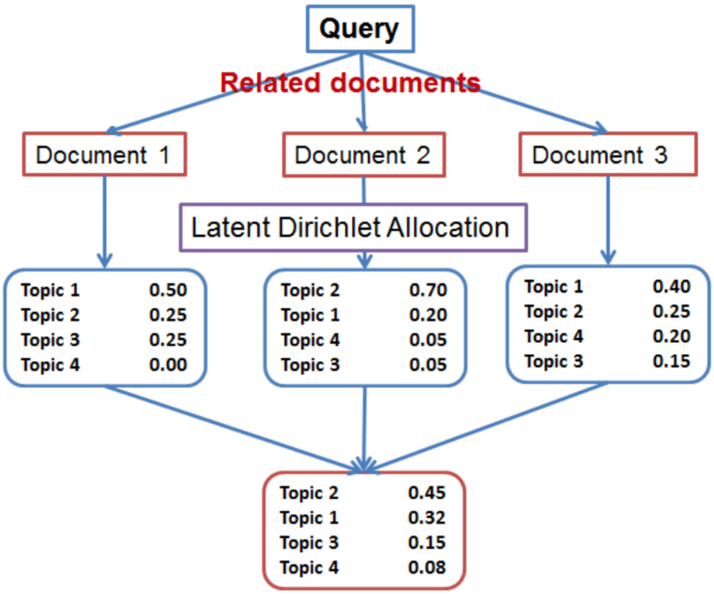


Figure 7.4: Topic-based query modelling using related documents

as a mixture of probabilities of z given query $q_i \in Q$ as follows:

$$p_Q(z|u) = \sum\nolimits_{q_i \in Q} \lambda_i p(z|q_i) \tag{7.3}$$

$p(z|q_i)$ is defined in Equation 7.2. Similar to the click user profile, $\lambda_i = \frac{1}{M}\alpha^{t_{q_i}-1}$ is the exponential decay function of t_{q_i}, which is the order of the query q_i submitted by the user u in the search session. $t_{q_i} = 1$ indicates that q_i is the most recent query; M is the normalisation factor. α is the decay parameter $(0 \leq \alpha \leq 1)$.

Figure 7.5 shows an example of a query user profile, which is constructed using three submitted queries and a decay parameter $\alpha = 0.9$. We can see that the user is more and more interested in the "Topic 4", and the query profile can capture that topic of interest.

7.1.2 Re-ranking suggested queries using user profiles

Figure 7.6 shows an example of the query suggestion function installed at the University of Essex Intranet. When the user submits a query (e.g., "webmail") to the Essex Intranet search function, the search function will return two ranked lists. The first list contains documents (10 per page), which are related to the input query returned by the original Essex ranker. The

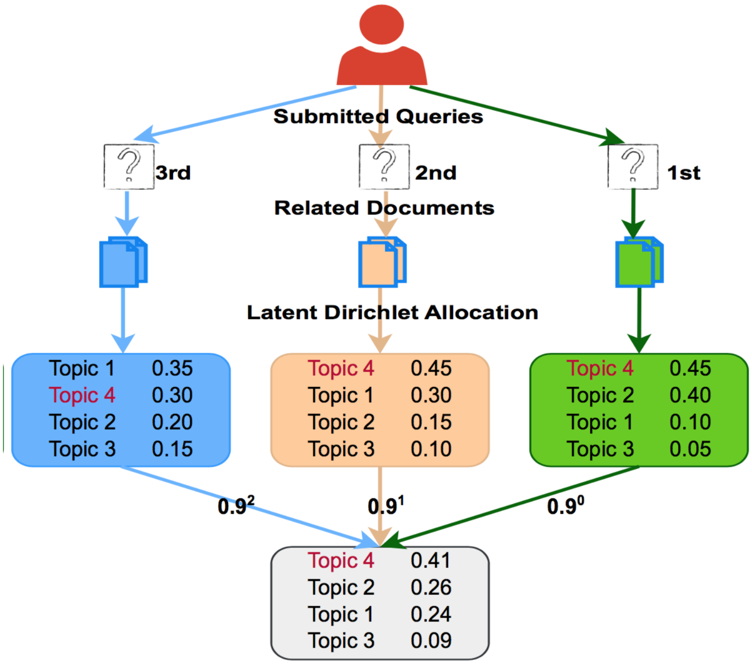
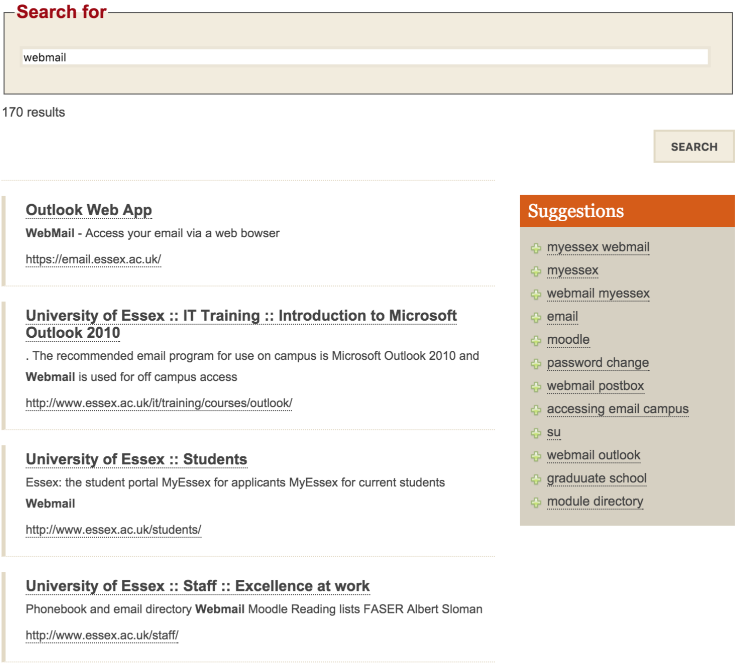


Figure 7.5: Temporal query user profile



Search for

webmail

170 results

Outlook Web App

WebMail - Access your email via a web bowser

https://email.essex.ac.uk/

University of Essex :: IT Training :: Introduction to Microsoft Outlook 2010

. The recommended email program for use on campus is Microsoft Outlook 2010 and **Webmail** is used for off campus access

http://www.essex.ac.uk/it/training/courses/outlook/

University of Essex :: Students

Essex: the student portal MyEssex for applicants MyEssex for current students **Webmail**

http://www.essex.ac.uk/students/

University of Essex :: Staff :: Excellence at work

Phonebook and email directory **Webmail** Moodle Reading lists FASER Albert Sloman

http://www.essex.ac.uk/staff/

Suggestions

- myessex webmail
- myessex
- webmail myessex
- email
- moodle
- password change
- webmail postbox
- accessing email campus
- su
- webmail outlook
- graduate school
- module directory

Figure 7.6: An example of the query suggestion function installed at the University of Essex Intranet

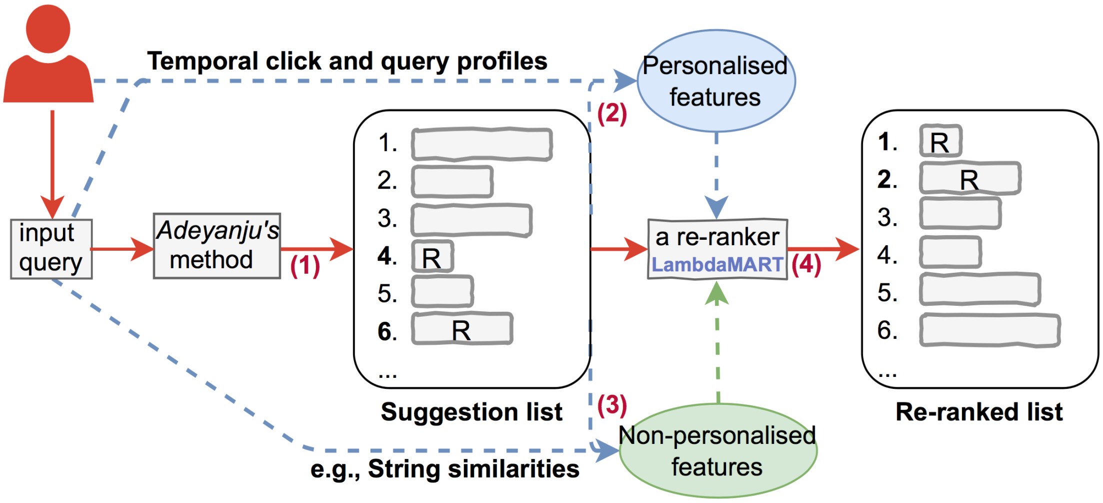


Figure 7.7: The general process of re-ranking. R means that the query is relevant to the user

second list contains at most 12 suggested queries, which are related to the input query returned by a state-of-the-art query suggestion algorithm (Albakour et al., 2011; Adeyanju et al., 2012).

After getting two user profiles for each search session, we use the two user profiles in a learning-to-rank mechanism to re-rank the query suggestion list returned by a state-of-the-art non-personalised query suggestion method proposed by Adeyanju et al. (2012), denoted as Adeyanju's. Specifically, Adeyanju's first constructs a domain knowledge structure in the form of a term subsumption hierarchy using both the Intranet document collection and collective users' query logs. Specifically, a term is either a unique keyword or an n-gram (phrases) extracted from the collection. Then, the term co-occurrence (i.e., the document frequency - df) is used to create the subsumption hierarchical tree. Let df be a function that return the document frequency of a term. A term 'x' is said to subsume 'y' if $df(x) > df(y)$ and $df(x, y)/df(y) \geq \gamma$. Figure 7.8 shows an example of the term subsumption hierarchy. One the hierarchy is built using the collection, it is updated using the actual search query logs collected by the Essex Intranet search engine (i.e., the Essex dataset).

Next, the suggestion list is generated using the top n terms most related to the query in the hierarchy. The link weight w between two terms 'x' and 'y' is set as $df(x, y)/df(x)$. In Figure 7.8 shows the weights for links between terms. Given a query such as term 'F' in Figure 7.8,

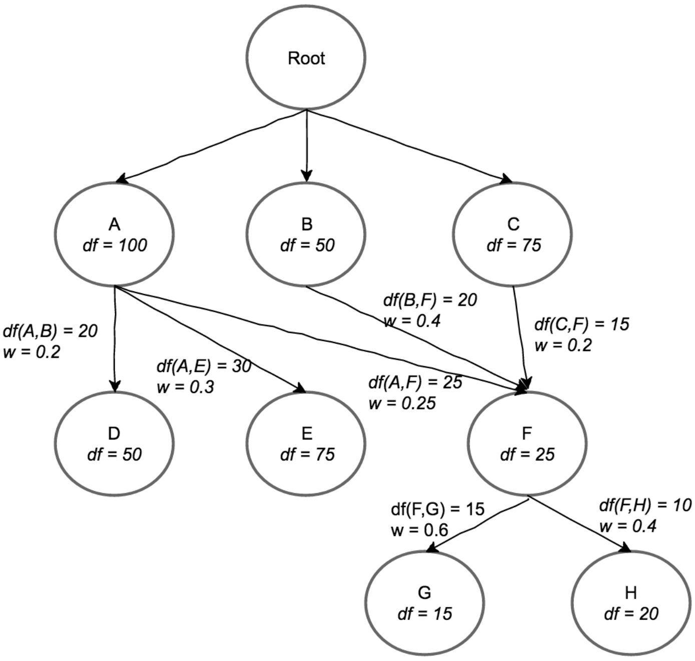


Figure 7.8: An example of the term subsumption hierarchy

Adeyanju's method will suggest a ranked list of terms 'G', 'H', 'B', 'A', 'C' based on the link weights (w).

Figure 7.7 shows an overview of using the two temporal user profiles to personalise the original query suggestion list returned by Adeyanju's. For each input query, our re-ranking method is as follows:

1. We generate the top n ranked suggested queries using the Adeyanju's method. We denote a suggested query as q_s.

2. As in chapters 4, 5 and 6, we then compute similarity scores between q_s and $profile(C)$, and between q_s and *profile*(Q). Both the suggested query q_s and a user profile (denoted as pf which is either *profile*(C) or *profile*(Q)) are modelled as distributions over topics Z (Section 7.1.1). To measure the similarity between q_s and the user profile pf, we use Jensen-Shannon divergence ($D_{JS}\lfloor.||.\rfloor$), which is a popular method of measuring the divergence (similarity) between two distributions (as detailed in Chapter 4), to measure the similarity between q_s and pf as follows:

$$Sim(q_s|pf) = -D_{JS}\lfloor Q||P \rfloor = -\big(\frac{1}{2}D_{KL}\lfloor Q||M \rfloor + \frac{1}{2}D_{KL}\lfloor P||M \rfloor\big) \tag{7.4}$$

Here, Q and P are distributions over topics of q_s and pf, respectively. $D_{KL}\lfloor.||.\rfloor$ is the Kullback-Leiber divergence and $M = \frac{1}{2}(Q+P)$. We consider the scores as the *personalised features.*

3. We also extract other *non-personalised features* of the input query q and the suggested query q_s. Table 7.2 shows the features extracted for re-ranking the suggestion list.

Table 7.2: The personalised query suggestion features

Feature	**Description**
Personalised Features	
ClickPersonalisedScore	The similarity score between the suggested query and the user click profile
QueryPersonalisedScore	The similarity score between the suggested query and the user query profile
Non-personalised Features	
QueryRank	Rank of the suggested query on the original list
QuerySim	The cosine similarity score between the current query and the previous query
QueryNo	Total number of queries that have been submitted to the Search Engine
SuggestedQueryCosine	The cosine similarity score between the current query and the suggested query
SuggestedQueryJaccard	The Jaccard distance score between the current query and the suggested query
SuggestedQueryEdit	The edit distance between the current query and the suggested query
SuggestedQueryLevenshtein	The Levenshtein distance between the current query and the suggested query
SuggestedQueryPreUsed	Whether the suggested query was used by the user in the same search session?

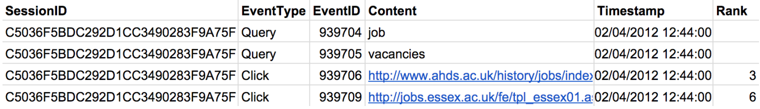
4. After extracting the query features, to re-rank the top n suggested queries, we employ LambdaMART (Burges et al., 2006) to train ranking models as in Chapters 5, 6.

7.2 Dataset and evaluation methodology

7.2.1 Dataset

The dataset used in our experiments contains large-scale[3] query logs collected from the search engine installed on the Intranet site of the University of Essex during the two years covering 1

[3] In that they represent the complete set of interactions with an Intranet search engine over a two-year period



SessionID	EventType	EventID	Content	Timestamp	Rank
C5036F5BDC292D1CC3490283F9A75F	Query	939704	job	02/04/2012 12:44:00	
C5036F5BDC292D1CC3490283F9A75F	Query	939705	vacancies	02/04/2012 12:44:00	
C5036F5BDC292D1CC3490283F9A75F	Click	939706	http://www.ahds.ac.uk/history/jobs/inde	02/04/2012 12:44:00	3
C5036F5BDC292D1CC3490283F9A75F	Click	939709	http://jobs.essex.ac.uk/fe/tpl_essex01.a	02/04/2012 12:44:00	6

Figure 7.9: A search session from the Essex logs

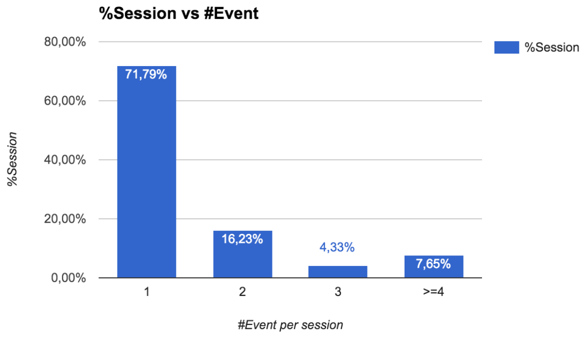


Figure 7.10: Number of events per search session

January 2012 - 31 December 2013 (i.e., the Essex dataset detailed in Chapter 3 (Experimental Methodology)). Each log sample contains a session identifier, the event type (i.e., a query or a click), an auto-increment id, the event content (i.e., query text, click URL), and the event time-stamp. Figure 7.9 shows a search session from the query logs. For each click URL, we download the content of the clicked document for the learning of the topics.

Figure 7.10 shows the distribution of the session length in term of the number of events per session. We see that more than 70% of the Essex search session contain only one event (either a query or a click) denoted as a single event session. The reason why the single event session happened frequently could be explained by the fact that instead of clicking on the returned document directly, the searcher uses the browser tab function to open the document (i.e., single click event sessions). It could also be that the user could get the direct answer from the document list after getting the search result list (i.e., single query event sessions). A reasonable reason is that after submitting a query to the intranet search function, the user could not get any suitable information from the returned result list, and then he or she abandoned the search (Huang et al., 2011; Diriye et al., 2012).

With those sessions, we could not justify whether the user got the needed information or not.

Moreover, we also could not apply our proposed method to personalise the query suggestion list. We employ a pre-processing step on the dataset to remove all the single search session. After this pre-processing step, 71% of the Essex search sessions (about 1.9 million) had been removed from the dataset. Moreover, the search function logs all user queries submitted or selected from the query suggestion list by the user. The user sometimes clicked on the search button with an empty query. After counting the number of the empty queries (about 50,000), we also removed all these query events from the logs.

We then analysed statistically the remainder of the Essex logs. Table 7.3 shows some basic log statistics from the remainder of the query logs. We see that although the number of search sessions decreased significantly from about 400,000 in 2012 to about 342,000 in 2013, the average number of user interactions increased slightly from 3.22 to 3.25. The average number of distinct queries was 3.75, which is similar to that of the commercial search engines (i.e., Bing). However, the average number of distinct URLs was 25.68, which is much higher than that of Bing.

Table 7.3: Basic statistics of the evaluation search logs

Item	2012	2013	Total
#search session	397,461	338,391	735,804
#event	1,263,179	1,083,992	2,347,171
#event/session	**3.22**	**3.25**	**3.23**
#query	757,645	659,284	1,416,929
#query/session	**1.91**	**1.95**	**1.93**
#click	505,534	424,708	930,242
#click/session	**1.27**	**1.26**	**1.26**

7.2.2 Evaluation methodology

Evaluation methodology For evaluation, we use AutoEval, an automated evaluation framework, which measures the performance of query suggestions automatically based on the actual query logs of an Intranet search (Albakour et al., 2011). Specifically, for each *query suggestion* *list*, we assign a positive label for a suggestion if it is an actual *refinement*, which is the next submitted query in the search session, and there is at least one user click on retrieved results after the refinement. In other words, we interpret the user click after a reformulation as the criterion of a relevant suggestion. The remainder of the suggestion list is assigned negative (irrelevant) labels. We use the rank positions of the positively labelled queries as an approximation of the ground truth to evaluate the performance of query suggestion before and after re-ranking.

We also follow the experimental methodology in Adeyanju et al. (2012), that is, the model is evaluated continuously at periodic intervals. Specifically, we use the logs in the week i for training the re-ranking model and the following week $i+1$ for testing the trained model; where, in our experiments, $1 \leq i \leq w$, the number of weeks in the test period.

7.2.3 Experimental settings

Our personalisation method and baselines We name our proposed re-ranking model as Ours. We choose two baselines to compare our work against. The *first baseline* is the Adeyanju's method (Adeyanju et al., 2012), which we reimplemented to generate the original suggestion list for *re-ranking*.

We use the session-based approach proposed by Bennett et al. (2012) as the *second baseline*. Specifically, in the baseline, we use only the click user profile (i.e., $\textit{profile}(C)$) together with the non-personalised features detailed in Table 7.2 to re-rank the suggestion list. We name the baseline as Click.

We note that Adeyanju's is a non-personalised approach and achieved a good performance of query suggestion on Intranet search. Click is a personalised approach and produced good performances in Web search personalisation (Bennett et al., 2012; Harvey et al., 2013; White et al., 2013; Vu et al., 2015a). Moreover, instead of the session-based approach by Bennett et al. (2012) as our personalised baseline, we could alternatively have used Shokouhi (2013) or Shokouhi and Guo (2015).

LDA & LambdaMART We train an LDA model on the clicked documents extracted from the query logs, as detailed in Section 7.1.1. The number of topics (i.e., 300 in our experiments) is decided by using a held-out validation set which consists of 10% of all clicked documents. The selected number of topics is the one that gives the lowest perplexity value. Table 7.4 shows the ten most probable topical words in topics trained using LDA. The decay parameter α for the two user profiles is set to 0.90 as in Chapter 5. The ranking function is learned using LambdaMART (Burges et al., 2006). We used the default setting for LambdaMART's prior parameters[4].

Table 7.4: The ten most probable topical words in topics trained using LDA

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
language	project	support	computer	housing
linguistic	management	development	science	home
department	planning	work	engineering	rent
english	strategic	issues	electronic	house
research	plan	provide	school	household
resource	university	training	research	property
modern	change	services	system	mortgage
committee	relu	community	professor	urban
sociolinguistic	knowledge	local	project	family
arabic	portal	process	industry	local

Evaluation metrics The evaluation is based on the comparison between our personalised approach and the baselines. We use the same set of evaluation metrics as Chapters 5 and 6 which are: Mean Average Precision (*MAP*), Precision (*P*@*k*), Mean Reciprocal Rank (*MRR*), and Normalized Discounted Cumulative Gain (*nDCG*@*k*). These are standard metrics which have been widely used for performance evaluation in document ranking (Manning et al., 2008). For each evaluation metric, a higher value indicates a better ranking quality.

[4] Number of leaves = 10, minimum documents per leaf = 200, number of trees = 100 and learning rate = 0.15

7.3 Experimental results

7.3.1 Overall performance

In this experiment, we analyse the effect of temporal user profiles proposed in Section 7.1.1 using six metrics: *MAP*, *P@1*, *P@5*, *MRR*, *nDCG@5* and *nDCG@10*. Table 7.5 shows promising results when user profiles are used to personalise the query suggestion list. We can see that even using only the click user profile, the Click method has led to the improvement of 7.22% on *MAP* over the Adeyanju's method. Especially, the combination of the query user profile and the click user profile (i.e., Both) achieves the highest improvement of 10.97% over Adeyanju's on *MAP* score. The improvements indicate that personalisation helps improve the query suggestion performance. The improvements over the Adeyanju's method are all significant with the paired t-test ($p < 0.01$).[5] The MAP value of Adeyanju's was 0.5440 out of 1 indicating that the relevant documents were largely (more than 76%) in the first half of the original lists. Therefore, the improvements after re-ranking were not based on chance.

Table 7.5: Overall performance of the methods. %rel denotes the relative improvement over Adeyanju's

Model	*MAP*	*P@1*	*P@5*	*MRR*	*nDCG@5*	*nDCG@10*
Adeyanju's	0.5440	0.4113	0.1823	0.5447	0.5714	0.6000
Click	0.5833	0.4271	0.1981	0.5839	0.6193	0.6583
%rel	*+7.22%*	*+3.84%*	*+8.67%*	*+7.22%*	*+8.38%*	*+9.73%*
Both	**0.6037**	**0.4526**	**0.2026**	**0.6043**	**0.6413**	**0.6780**
%rel	*+10.97%*	*+10.04%*	*+11.14%*	*+10.94%*	*+12.23%*	*+13.02%*

In the comparison between personalisation methods (i.e., Both and Click), Table 7.5 shows that using both the click and query user profiles (i.e., Both) significantly improves the suggestion quality over the Click baseline ($p < 0.01$). Interestingly, our method produces a significantly better quality of the first query in the suggestion list with the improvement of 5.97% on *P@1*

[5] The exact p values are reported in Table B.7, Appendix B.

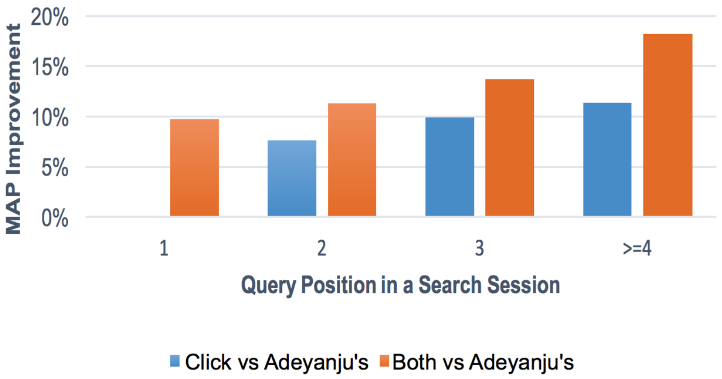


Figure 7.11: Relative performance improvements over Adeyanju's with different query positions. There is no result reported for Click with the first query because we cannot build the click profile as there is no previously clicked document

over Click. The improvements of Both over Click also indicate that the query user profile is highly important in the query suggestion task, especially in the quality of the first suggested query.

7.3.2 Performance on different query positions

With more submitted queries and clicked documents, we can build richer user profiles. In this experiment, we aim to study whether the position of a query in a search session has any effect on the performance of personalised query suggestion. For each search session, we label queries by their positions during the session. Because there are few sessions containing more than three queries (i.e., accounting for 7.65% of sessions in the query logs), we label the first three queries from one to three according to their orders in the search session; the remaining queries are labelled as ≥ 4.

For the first query, we cannot build the click user profile because there is no previously clicked document. However, we can still build the query user profile for the first query. We show the improvement in performance of the personalised methods over Adeyanju's in term of *MAP* metric with different query positions in Figure 7.11. Here the statistical significance is verified by the paired t-test ($p < 0.01$).[6] For the first query within a search session, our method, which can use only the query user profile, significantly improves the query suggestion performance over Adeyanju's. It again confirms the effectiveness of the query information on personalised query suggestion for Intranet search.

From the second query, we can build both the query and click user profiles. One can see that the higher the position of a query is, the greater the improvement in performance the personalised query suggestion can achieve. Specifically, from the query with high positions (i.e., ≥ 4), the improvements of Click and Both are 11.37% and 18.22%, respectively. Figure 7.11 also shows that Both outperforms Click significantly with the improvements of at least 3.45% ($p < 0.01$). It indicates that richer user profiles (by observing more clicked documents and submitted queries during the search session) help achieve better query suggestion performances. The findings offer future research directions that use user profiles which go beyond single sessions

7.3.3 Performance on different query lengths

The query length is defined by the number of words in the query (e.g., the query "University webmail" has the length of 2). The length of a query might give an indication as to how specific the information need of an individual user is (i.e., a longer query can typically be assumed to reflect a more specific information need).

In Figure 7.12, we show the distribution of the query length in the Essex query logs. It can be seen that most queries consisted of less than or equal to 3 terms. The percentage of the query containing either less than or equal to 3 terms was about 94% and stayed almost identical over the two years 2012 and 2013. Regardless of the year, there were very few queries containing more than 3 terms. The numbers for the two years were 5.6% and 5.8%, respectively. In this experiment, we aim to show the impact of personalisation on query suggestion with different query lengths. We label each query by its length, which is the number of words in the query (i.e., from one to three and ≥ 4 words because there are few queries containing more than three words (5.7% of queries in the query logs)).

[6] The exact p values are detailed in Table B.8, Appendix B.

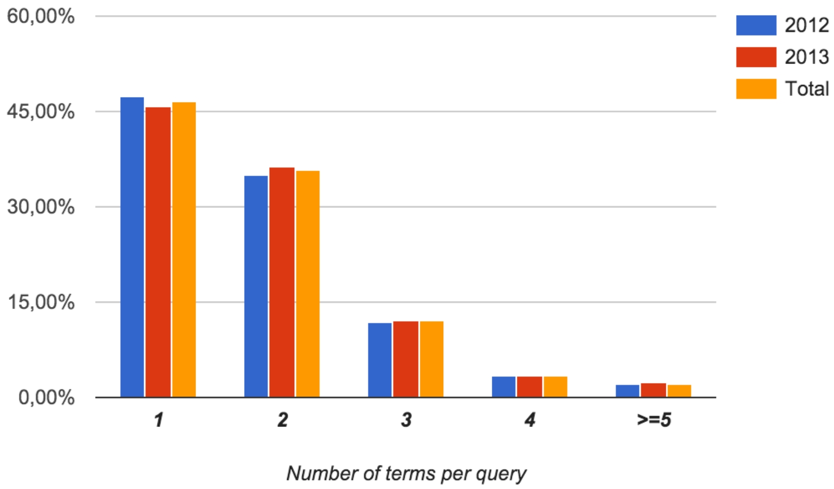


Figure 7.12: The percentage of the queries that contained n terms

Figure 7.13 shows the improvement in performance of Click and Both over Adeyanju's method in term of *MAP* with different query lengths. We see that personalisation methods achieve significantly better performances than the non-personalised method does ($p < 0.01$).[7] Even for short queries (length 1 and 2) which tend to be more generic, the Click and Both methods outperform the Adeyanju's method with an improvement of more than 6.11% and 9.09%, respectively. We see that the longer the query is, the higher the improvement personalised methods can achieve. Specifically, with a longer query (i.e., with length ≥ 4), the Click and Both methods yield the highest improvements, i.e., 29.82% and 53.85%, respectively. This indicates that the longer query would also get more benefit from the personalisation.

Figure 7.13 also indicates that by combining the query user profile with the click user profile, Both significantly improves the query suggestion performance over using only the click user profile ($p < 0.01$) . Specifically, the improvements are significantly high with the long query (i.e., containing at least three words) with the changes of 6.7% and 18.2% on the query containing three words and the query containing at least four words, respectively.

[7] The exact p values are detailed in Table B.9, Appendix B.

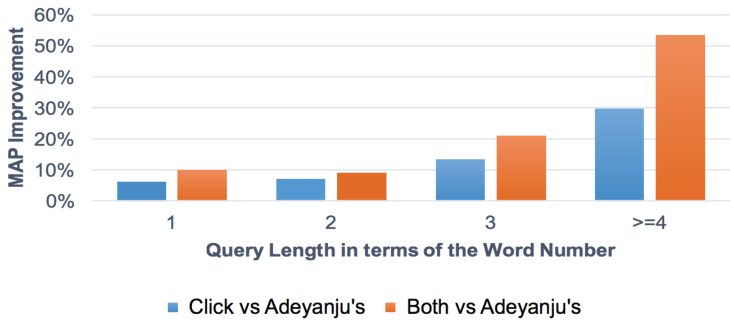


Figure 7.13: Relative performance improvements over Adeyanju's with different query lengths

7.4 Conclusions

In this chapter, we proposed a personalised query suggestion framework and showed how it performed on Intranet search. We built two session-specific temporal user profiles, a query user profile using the submitted queries, and a click user profile using the clicked documents. We then extracted the personalised features using the two profiles and combined them with non-personalised features to learn a ranking model using LamdaMART. Finally, we used the ranking model to re-rank the query suggestion list returned by a well-performing query suggestion approach for Intranet search. Our experiments provide answers to the main research question raised at the beginning of this chapter:

RQ 5 *How can we personalise query suggestion for Intranet search?*

To answer the research question, we worked with the Essex dataset collected from the search engine installed on the Website of the University of Essex. Experimental results show that personalisation significantly improved the query suggestion performance. Using both the click user profile and query user profile achieved the highest performance indicating that the personalised query suggestion for Intranet search should take into account both click and query information. Moreover, the positive impact of personalised query suggestions is more pronounced with longer queries and queries submitted later within a session.

Chapter 8

Conclusions and Future Work

Search personalisation in information retrieval has been extensively studied in recent years. It is one of the most important features of commercial search engines because it helps users quickly find the information they need. The performance of search personalisation depends on the richness of user profiles which represent the user's search interests. In this thesis, we started with a discussion of the particular and scientific obstacles that motivated our research on dynamic user profiling for search personalisation. In answering the research question "how we can build user profiles dynamically to improve the performance of search personalisation?", we found that dynamic user profiles help to improve the performance of search personalisation.

The four research chapters (4 - 7) of this thesis address the research question as follows: First, in Chapter 4, we focused on how to build a topic-based profile, and then improve the performance of search personalisation with dynamic group formation. In particular, to build a user profile, instead of using a human-generated online ontology, such as Open Directory Project (ODP), we proposed the use of a topic modelling method, namely, Latent Dirichlet Allocation (LDA) (Blei et al., 2003), to automatically extract topics from documents relevant to the user. We then grouped users who share common interests dynamically with respect to the user's input query. That is, with different input queries, we grouped different users who share common interests. After that, we used the information about these users with shared interests to enrich the current user profile.

In Chapter 5, we revisited the problem of building temporal user profiles for search personalisation using topics automatically learned from the user's relevant documents. We focused on building dynamic user profiles for search personalisation, but also examined the different temporal aspects of building three user profiles (i.e., a long-term profile, a daily profile and a short-term profile).

In Chapter 6, we turned to the problem of dynamic user profiling for search task modelling, and experimentally verified the effectiveness of temporal user profiling for modelling search tasks via a re-ranking scheme.

Finally, in Chapter 7, we focused on the problem of personalised query suggestion for Intranet search, and proposed two session-based user profiles: a user click profile and a user query profile. We then applied these in a re-ranking scheme to deal with the problem.

To conclude, this chapter first summarises our answers to the research questions set out in the Introduction chapter as well as key findings in Section 8.1. We then carve out an outlook on future research directions in Section 8.2.

8.1 Answers and key findings

The goal of our research is to improve the performance of search personalisation with dynamic user profiling techniques. In particular, in chapter 4, the research questions we addressed focused on building a topic-based user profile and whether we can improve the performance of search personalisation with dynamic group formation:

> **RQ 1** *How can we build a user profile which represents the user's topical search interests?*

We have proposed a methodology to build a user profile using topics learned from the user's relevant documents. We handled the problems with human-generated ontologies and directories (such as ODP) by employing a topic modelling technique (LDA) to automatically extract topics from the documents. The user profile is dynamic because it is updated "on-the-fly" when the search system observes more relevant documents.

Key findings Utilising the user profile to personalise search results returned by the Bing search engine, we found that the profile (S_Profile) helps to improve the search performance over the Bing default ranker.

> **RQ 2** *How can we dynamically group users who share common interests for search personalisation?*

In existing approaches, groups of similar users are often statically determined, for example by finding groupings based on the common documents that users clicked. However, these static grouping methods are query-independent and neglect the fact that users in a group may have different interests on different topics. To handle these problems, we used the current input query to dynamically group users who share common interests with the current user on that query. This leads to the fact that, with different input queries, the system will return different groups of users with shared common interests. We then utilised the enriched user profile to personalise the search results returned by the Bing search engine with a re-ranking scheme.

Key findings We found that the dynamically enriched profile can stably and significantly improve the ranking quality over the competitive ranker of the original search engine and the individual as well as statically enriched profiles. Other experimental results confirmed the impact of the query click entropies. That is, the higher click entropy is, the better performance the search personalisation is likely to achieve. Finally, the experimental results on the user group sizes indicated that we achieved better search performances using data from more users with shared interests.

Although the user profiles are enriched dynamically with respect to the input query, in chapter 4 we built a single user profile using all the user's relevant documents and treating the relevant documents equally. However, with more interactions with the search system, the user's interests may change over time, and the more recent relevant documents can express the user's current interests more accurately than distant ones. Therefore, the user modelling method should be able to build user profiles dynamically in such a way that the user's interests can be captured. Therefore, in order to evaluate the effectiveness of temporal aspects in personalisation, we took an approach to building three temporal user profiles (i.e., *a long-term user profile*, *a daily user profile* and *a session profile*) using the user's historical interactions in three typical time scales (i.e., the whole search history, a day and a search session, respectively). Regarding these new profiles, we addressed the following research question:

> **RQ 3** *How can we build temporal user profiles for search personalisation?*

In answering this question, we proposed a temporal weighting scheme to build temporal user profiles. In particular, we have presented a study on the temporal aspects of building user profiles with latent topics learned from the documents. For each user, we used relevant documents at different time scales to build long-term, daily, and session profiles. Each user profile is represented as a distribution over latent topics from which we extract the features and combine them with non-personalised features to learn a ranking function. We performed a set of experiments to study the effectiveness of the temporal latent topic-based profiles.

Key findings We found that the temporal user profiles help to improve the search performance regarding traditional relevance-oriented metrics like MAP, $P@k$ over the competitive ranker of the Bing search engine and the non-temporal user profile. We also found that the session profile captures the most current search interests of a user and can generate helpful features for learning the re-ranking function. The best performance was achieved by the combination of all three temporal profiles, indicating that a good personalisation should take into account all temporal aspects from user's search history. Other experimental results confirmed that the impact of the query click entropy on the temporal latent topic profiles is similar to that on the non-temporal latent topic profiles (Chapter 4). Finally, another interesting finding is the usefulness of the temporal profile in tuning the search results for the next queries in a session.

Although the effectiveness of the session user profile has been confirmed by the experiments on the Bing query logs, in the context of the web search logs, a user may handle several *search tasks* within a search session and submit several queries within a search task (Liao et al., 2012). Therefore, a single profile for each search session might not quickly capture the interests of the user's search tasks within that session. We argue that the user profile should be constructed using the information of the user's current search task.

Recent research has shown that mining and modelling the user's current search task help to improve the performance of search personalisation. However, previous studies largely ignored the dynamic nature of the search task; that is, with the change of time, the search intent and user interests may also change. To address this problem, we built the temporal task-based user profile using the historical interactions between the user and the search system during that search task. This leads to the following research question:

RQ 4 *How can we model search tasks for search personalisation?*

In answering this question, we utilised the temporal user profiling method to model the user's search tasks. In particular, we first extracted the user's search tasks from the user's current search session using a state-of-the-art method. Second, for each search task, we proposed an approach to modelling a search task using the user's relevant documents in that task. Each search task is represented as a distribution over the topics from which we extract the personalised feature and combine it with non-personalised features to learn a ranking function to re-rank search results returned by the Bing search engine.

Key findings We found that the ranking quality is improved significantly over the default ranker of the Bing search engine. We also found that task-based profiles (even a non-temporal one) help significantly improve the search performance over the session-based profile. In addition, by capturing the temporal aspect, the search task outperforms the non-temporal search task.

We proposed two dynamic profiling methods for search personalisation, those being dynamic group formation and temporal topic user profiles. We studied the effectiveness of the dynamic user profiles by applying them to re-rank the search results returned by the Bing search engine. Our experimental analysis shows that the dynamic user profiles help to significantly improve the performance of the default ranker of the Bing search engine as well as strong personalisation baselines. We now would like to see whether the dynamic user profiles can be applied to another personalisation application (e.g., query suggestion). In particular, we approach this problem by handling the following research question:

RQ 5 *How can we personalise query suggestion for Intranet search?*

Recent research has shown the usefulness of using collective user interaction data (such as query logs) to recommend query modification suggestions for Intranet search. However, most of the query suggestion approaches for Intranet search follow a “one size fits all” strategy, whereby different users who submit an identical query would get the same query suggestion list. This is problematic, as even with the same query, different users may have different topics of interest, which may change over time in response to the user’s interaction with the system. We have addressed the problem by extending our temporal session user profiles to personalise the query suggestion for Intranet search. For each search session, we constructed two temporal user profiles: *a click user profile* using the user’s clicked documents and *a query user profile* using the user’s submitted queries. We then used the two profiles to re-rank the non-personalised query suggestion list returned by a state of the art query suggestion method for Intranet search.

Key findings We found that our temporal user profiles significantly improve the quality of the query suggestion list. Moreover, using both the click user profile and query user profile achieved the highest performance indicating that the personalised query suggestion for Intranet search should take into account both click and query information. Moreover, the positive impact of personalised query suggestions is more pronounced with longer queries and queries submitted later within a session.

8.2 Future work

The work presented in this thesis provides insights and algorithms to build dynamic user profiles for search personalisation. Beyond the findings and conclusions summarised above, it opens up important directions for future work. In this final section, we identify some possible future directions.

As a common search personalisation approach involves re-ranking the top N candidates initially returned by a basis ranker, typically $N = 10$ (Bennett et al., 2012; White et al., 2013; Yang et al., 2016), it would be interesting to have a closer look at the top N candidates for

$N > 10$: how much we can gain from the good candidates that were ranked at lower ranks than 10? A large value of N could bring a possible completion in search personalisation. Moreover, the default ranker of the Bing search engine has been getting stronger and stronger. Our proposed personalisation frameworks help to significantly improve the performance over the default ranker in 2012. However, due to the constraint of available data, we do not know whether it can help to improve the performance of the current default ranker. Therefore, we aim to apply our dynamic user profiling methods to other large-scale and more recent datasets.

Also, current research has shown that mining and modelling task behaviour help to improve the performance of search personalisation (White et al., 2013). In their research, they focused on building rich models of the current user's task by grouping other users who have performed similar tasks, and leverage their on-task behaviour to improve search performance. However, in the research, they grouped the users statically and neglected the fact that different input queries should return different groups of users who have performed similar tasks. To handle this problem, we aim to apply our dynamic group formation method to dynamically find users with shared on-task behaviours using the user's current input query.

In chapter 4, we found that our dynamic grouping formation helps to significantly improve the performance of search personalisation. However, in this method, we treated the user's relevant documents to build a single user profile and the user's common relevant documents to build a shared user profile *equally* without considering the temporal aspects (described in chapter 5). We aim to apply the temporal user profiling methodology to handle these problems.

In our research, we have extensively used the LDA topic model to extract topics from documents as it is fast and scalable. In the future, we aim to apply other topic modelling algorithms, such as Dynamic Topic Models (Blei and Lafferty, 2006), Hierarchical Topic Models (Griffiths et al., 2004) or word/document embeddings techniques, such as word2vec (Mikolov et al., 2013), doc2vec (Le and Mikolov, 2014) to automatically learn the topics from documents. We can also improve topic models using word embeddings as in Nguyen et al. (2015a,b).

However, our research on dynamic user profiling is not limited to the Information Retrieval context. We can extend the dynamic user profiling methodology to other studies, such as the

Internet of Things (IOT) (Price et al., 2013; Bourgeois et al., 2014; Kummerfeld and Kay, 2017). Combining user profiling with cutting edge research in Smart Cities (Calvillo et al., 2016) could allow us to build dynamic user profiles using a user's daily activities, as well as with other information, such as weather patterns, peak and off-peak energy pricing rates and so on. This could help the user to maximise energy efficiency and reduce the costs of the household tasks (showering, laundry and so on).

Recently, "Fake News" has been a widespread topic of discussion, in particular following the 2016 US Presidential elections. The expression "Fake News" is generally used to refer to a story presented as a fact, but there is no factual basis (Allcott and Gentzkow, 2017). The term has become particularly prominent following the presidential elections, as it has been suggested (Allcott and Gentzkow, 2017) that the circulation of such stories on social media may have had an impact on voter's behaviour at the ballot box. Figure 8.1 shows a popular fake news available on the internet[1,2].

In fact, the current rapid spread of such fake news stories may be a product of excessive personalisation of search results and information presentation. If personalisation algorithms return search results based on only a user's search interests and those of their social group, inaccurate stories can spread widely irrespective of their truth (Pariser, 2011; White, 2013, 2014; White and Horvitz, 2015). This can result in the inability of users to inform themselves about the facts of a matter if the information presented to their is excessively skewed by the search interests of their social group (the so-called "echo chamber effect" (Colleoni et al., 2014; Flaxman et al., 2016)). Future work in this area could investigate how to balance a user's personalisation against her requirements for information from a broader set of sources, as well as possible mechanisms for considering the reliability of news items.

[1] http://yepsee.com/trump-offering-free-one-way-tickets-to-africa-mexico-for-those-who-wanna-leave-america-trending/

[2] https://twitter.com/NewssTrump/status/827439059832233985

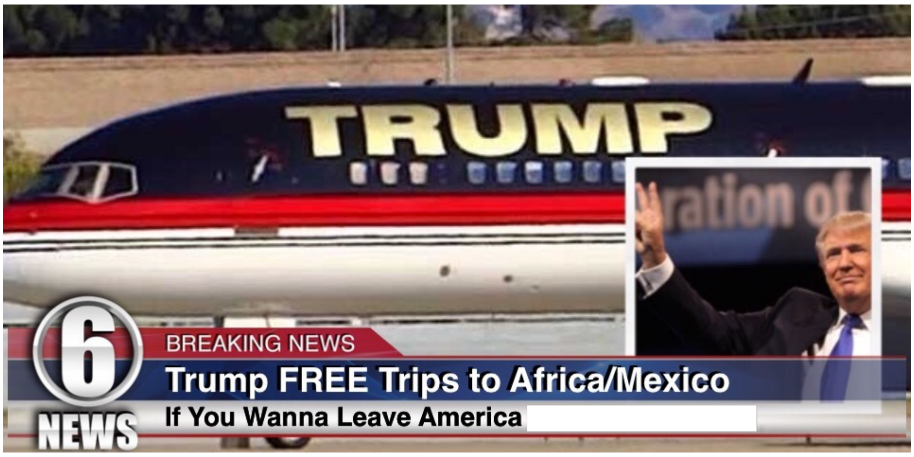


Figure 8.1: A popular "fake news" on the internet. Original fake post: *"Everyone says they want to go to Africa or Mexico well here's your chance! Make America great and go back to your country for free!" Trump*

Bibliography

Ibrahim Adepoju Adeyanju, Dawei Song, M-Dyaa Albakour, Udo Kruschwitz, Anne De Roeck, and Maria Fasli. Adaptation of the concept hierarchy model with search logs for query recommendation on intranets. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 5–14, 2012.

Elif Aktolga and James Allan. Reranking search results for sparse queries. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 173–182, 2011.

Elif Aktolga, Alpa Jain, and Emre Velipasaoglu. Building rich user search queries profiles. In *User Modeling, Adaptation, and Personalization: 21st International Conference, UMAP 2013, Rome, Italy, June 10-14, 2013 Proceedings*, pages 254–266, 2013.

M-Dyaa Albakour, Udo Kruschwitz, Nikolaos Nanas, Yunhyong Kim, Dawei Song, Maria Fasli, and Anne De Roeck. Autoeval: An evaluation methodology for evaluating query suggestions using query logs. In *Proceedings of the 33rd European Conference on IR Research on Advances in Information Retrieval*, pages 605–610, 2011.

Hunt Allcott and Matthew Gentzkow. Social media and fake news in the 2016 election. Technical report, National Bureau of Economic Research, 2017.

F. Asnicar and C. Tasso. ifweb: a prototype of user model-based intelligent agent for document filtering and navigation in the world wide web. In *Proceeding of 6th International Conference on User Modelling*, 1997.

Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. *Modern Information Retrieval - the concepts and technology behind search, Second edition.* Pearson Education Ltd., Harlow, England, 2011.

Paul N. Bennett, Ryen W. White, Wei Chu, Susan T. Dumais, Peter Bailey, Fedor Borisyuk, and Xiaoyuan Cui. Modeling the impact of short- and long-term behavior on search personalization. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 185–194, 2012.

David Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.

David M. Blei and John D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 113–120, 2006.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

Jacky Bourgeois, Janet Van Der Linden, Gerd Kortuem, Blaine A Price, and Christopher Rimmer. Conversations with my washing machine: an in-the-wild study of demand shifting with self-generated energy. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 459–470, 2014.

Jay Budzik and Kristian J. Hammond. User interactions with everyday applications as context for just-in-time information access. In *Proceedings of the 5th international conference on Intelligent user interfaces*, IUI '00, pages 44–51, 2000.

Christopher J C Burges. From ranknet to lambdarank to lambdamart : An overview. Technical report, Microsoft Research, 2010.

Christopher J. C. Burges, Robert Ragno, and Quoc V. Le. Learning to rank with non-smooth cost functions. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, NIPS'06, pages 193–200, 2006.

Georg Buscher, Andreas Dengel, and Ludger Van Elst. Query expansion using gaze-based feedback on the subdocument level. In *Proceedings of the 31st annual international ACM*

SIGIR conference on Research and development in information retrieval, pages 387–394, New York, New York, USA, 2008.

Georg Buscher, Ludger Van Elst, and Andreas Dengel. Segment-level display time as implicit feedback: A comparison to eye tracking. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 67–74, 2009.

Fei Cai and Maarten de Rijke. Selectively personalizing query auto-completion. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pages 993–996, 2016a.

Fei Cai and Maarten de Rijke. A survey of query auto completion in information retrieval. *Foundations and Trends in Information Retrieval*, 10(4):273–363, 2016b.

Fei Cai, Shangsong Liang, and Maarten de Rijke. Time-sensitive personalized query auto-completion. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, pages 1599–1608, 2014.

Fei Cai, Shuaiqiang Wang, and Maarten de Rijke. Behavior-based personalization in web search. *Journal of the Association for Information Science and Technology*, 68(4):855–868, April 2017.

C.F. Calvillo, A. Snchez-Miralles, and J. Villar. Energy management and planning in smart cities. *Renewable and Sustainable Energy Reviews*, 55(C):273–287, 2016.

David Carmel, Eitan Farchi, Yael Petruschka, and Aya Soffer. Automatic query wefinement using lexical affinities with maximal information gain. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, pages 283–290, 2002.

Claudio Carpineto and Giovanni Romano. A survey of automatic query expansion in information retrieval. *ACM Comput. Surv.*, 44(1):1:1–1:50, 2012.

O. Chapelle, Y. Chang, and T.-Y. Liu. The yahoo! learning to rank challenge. http://learningtorankchallenge.yahoo.com, 2010.

Liren Chen and Katia Sycara. Webmate: a personal agent for browsing and searching. In *Proceedings of the second international conference on Autonomous agents*, pages 132–139, 1998.

Zhiyong Cheng, Shen Jialie, and Steven C.H. Hoi. On effective personalized music retrieval by exploring online user behaviors. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pages 125–134, 2016.

Paul Alexandru Chirita, Wolfgang Nejdl, Raluca Paiu, and Christian Kohlschütter. Using odp metadata to personalize search. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, pages 178–185, 2005.

Paul Alexandru Chirita, Claudiu S. Firan, and Wolfgang Nejdl. Personalized query expansion for the web. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 7–14, 2007.

Elanor Colleoni, Alessandro Rozza, and Adam Arvidsson. Echo chamber or public sphere? predicting political orientation and measuring political homophily in twitter using big data. *Journal of Communication*, 64(2):317–332, 2014.

Bruce Croft, Donald Metzler, and Trevor Strohman. *Search Engines: Information Retrieval in Practice*. Addison-Wesley Publishing Company, 1st edition, 2009.

Van Dang and Bruce W. Croft. Query reformulation using anchor text. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, pages 41–50, 2010.

Jean Charles de Borda. Mémoire sur les élections au scrutin. In *Histoire de l'Académie Royale des Sciences*. 1781.

Abdigani Diriye, Ryen White, Georg Buscher, and Susan Dumais. Leaving so soon?: Understanding and predicting web search abandonment rationales. In *Proceedings of the 21st*

ACM International Conference on Information and Knowledge Management, CIKM '12, pages 1025–1034, 2012.

Shen Dou, Sun Jian-Tao, Yang Qiang, and Chen Zheng. Building bridges for web query classification. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, page 131138, 2006.

Zhicheng Dou, Ruihua Song, and Ji-Rong Wen. A large-scale evaluation and analysis of personalized search strategies. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 581–590, 2007.

Susan Dumais, Edward Cutrell, JJ Cadiz, Gavin Jancke, Raman Sarin, and Daniel C. Robbins. Stuff i've seen: a system for personal information retrieval and re-use. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, pages 72–79, 2003.

Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 613–622, 2001.

Carsten Eickhoff, Kevyn Collins-Thompson, Paul N. Bennett, and Susan Dumais. Personalizing atypical web search sessions. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, pages 285–294, 2013.

Seth Flaxman, Sharad Goel, and Justin Rao. Filter bubbles, echo chambers, and online news consumption. *Public Opinion Quarterly*, pages 298–320, 2016.

Steve Fox, Kuldeep Karnawat, Mark Mydland, Susan Dumais, and Thomas White. Evaluating implicit measures to improve web search. *ACM Trans. Inf. Syst.*, 23(2):147–168, 2005.

Susan Gauch, Mirco Speretta, Aravind Chandramouli, and Alessandro Micarelli. User profiles for personalized information access. In *The Adaptive Web*, pages 54–89. Springer-Verlag, 2007.

M.Rami Ghorab, Dong Zhou, Alexander OConnor, and Vincent Wade. Personalised information retrieval: survey and classification. *User Modeling and User-Adapted Interaction*, pages 1–63, 2012.

William Sealy Gosset. The application of the law of error to the work of the brewery. *Guinness Internal Note*, 1904.

Laura A. Granka, Thorsten Joachims, and Geri Gay. Eye-tracking analysis of user behavior in www search. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 478–479, 2004.

Thomas L. Griffiths, Michael I. Jordan, Joshua B. Tenenbaum, and David M. Blei. Hierarchical topic models and the nested chinese restaurant process. In *Advances in Neural Information Processing Systems 16*, pages 17–24, 2004.

Qi Guo and Eugene Agichtein. Beyond dwell time: estimating document relevance from cursor movements and other post-click searcher behavior. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 569–578, 2012.

Aniko Hannak, Piotr Sapiezynski, Arash Molavi Kakhki, Balachander Krishnamurthy, David Lazer, Alan Mislove, and Christo Wilson. Measuring personalization of web search. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 527–538, 2013.

Abhay Harpale, Yiming Yang, Siddharth Gopal, Daqing He, and Zhen Yue. Citedata: A new multi-faceted dataset for evaluating personalized search performance. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 549–558, 2010.

Morgan Harvey, Fabio Crestani, and Mark J. Carman. Building user profiles from topic models for personalised search. In *Proceedings of the 22nd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '13, pages 2309–2314, 2013.

Ahmed Hassan and Ryen W. White. Personalized models of search satisfaction. In *Proceedings of the 22nd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '13, pages 2009–2018, 2013.

David Hawking. Enterprise search. In *Modern Information Retrieval, 2nd Ed.* 2010.

Jeff Huang, Ryen W. White, and Susan Dumais. No clicks, no problem: Using cursor movements to understand and improve search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 1225–1234, 2011.

Jeff Huang, Ryen W. White, Georg Buscher, and Kuansan Wang. Improving searcher models using mouse cursor activity. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '12, pages 195–204, 2012.

Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 154–161, 2005.

Ameni Kacem, Mohand Boughanem, and Rim Faiz. Time-sensitive user profile for optimizing search personlization. In *User Modeling, Adaptation, and Personalization: 22nd International Conference, UMAP 2014, Aalborg, Denmark, July 7-11, 2014. Proceedings*, pages 111–121, 2014.

J. Karas and I. Savage. Publications of frank wilcoxon. *Biometrics*, pages 1–10, 1967.

Makoto P. Kato, Tetsuya Sakai, and Katsumi Tanaka. When do people use query suggestion? a query suggestion log analysis. *Inf. Retr.*, 16(6):725–746, 2013.

Ali Khodaei, Sina Sohangir, and Cyrus Shahabi. Personalization of web search using social signals. In *Recommendation and Search in Social Networks*, pages 139–163, 2015.

Alfred Kobsa, Hichang Cho, and Bart P. Knijnenburg. The effect of personalization provider characteristics on privacy attitudes and behaviors: An elaboration likelihood model approach. *Journal of the Association for Information Science and Technology*, 67(11):2587–2606, 2016.

Weize Kong, Rui Li, Jie Luo, Aston Zhang, Yi Chang, and James Allan. Predicting search intent based on pre-search context. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 503–512, 2015.

Alexander Kotov, Paul N. Bennett, Ryen W. White, Susan T. Dumais, and Jaime Teevan. Modeling and analysis of cross-session search tasks. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 5–14, 2011.

Georgia Koutrika and Yannis Ioannidis. Rule-based query personalization in digital libraries. *International Journal on Digital Libraries*, 4(1):60–63, 2004.

Bob Kummerfeld and Judy Kay. User modeling for the internet of things. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, UMAP '17, pages 367–368, 2017.

Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. *CoRR*, abs/1405.4053, 2014.

Lillian Lee. Measures of distributional similarity. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 25–32, 1999.

Jingfei Li, Dawei Song, Peng Zhang, Ji-Rong Wen, and Zhicheng Dou. Personalizing web search results based on subspace projection. In *Proceedings of the 10th Asia Information Retrieval Societies Conference*, pages 160–171, 2014.

Liangda Li, Hongbo Deng, Yunlong He, Anlei Dong, Yi Chang, and Hongyuan Zha. Behavior driven topic transition for search task identification. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pages 555–565, 2016.

Yuefeng Li and Ning Zhong. Mining ontology for automatically acquiring web user information needs. *IEEE Trans. on Knowl. and Data Eng.*, 18(4):554–568, 2006.

Zhen Liao, Yang Song, Li-wei He, and Yalou Huang. Evaluating the effectiveness of search task trails. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 489–498, 2012.

J. Lin. Divergence measures based on the shannon entropy. *IEEE Trans. Inf. Theor.*, 37(1): 145–151, September 1991.

Xin Liu. Modeling users' dynamic preference for personalized recommendation. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI '15, pages 1785–1791, 2015.

Peter Lofgren, Siddhartha Banerjee, and Ashish Goel. Personalized pagerank estimation and search: A bidirectional approach. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, WSDM '16, pages 163–172, 2016.

Thomas Lumley, Paula Diehr, Scott Emerson, and Lu Chen. The importance of the normality assumption in large public health data sets. *Annual review of public health*, 23(1):151–169, 2002.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

John Paul Mc Gowan. A multiple model approach to personalised information access. Master's thesis, University College Dublin, 2003.

Andrew Kachites McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

Alessandro Micarelli and Filippo Sciarrone. Anatomy and empirical evaluation of an adaptive web-based information filtering system. *User Modeling and User-Adapted Interaction*, 14(2-3): 159–200, 2004.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013.

Yashar Moshfeghi and Joemon M. Jose. An effective implicit relevance feedback technique using affective, physiological and behavioural features. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 133–142, 2013.

Nikolaos Nanas, Victoria Uren, and Anne De Roeck. Building and applying a concept hierarchy representation of a user profile. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, pages 198–204, 2003.

Nikolaos Nanas, Manolis Vavalis, and Anne De Roeck. A network-based model for high-dimensional information filtering. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 202–209, 2010.

Nikolaos Nannas. *Towards Nootropia : a non-linear approach to adaptive document filtering.* PhD thesis, KMI, Open University, 2003.

Dat Quoc Nguyen, Richard Billingsley, Lan Du, and Mark Johnson. Improving topic models with latent feature word representations. *Transactions of the Association for Computational Linguistics*, 3:299–313, 2015a.

Dat Quoc Nguyen, Kairit Sirts, and Mark Johnson. Improving topic coherence with latent feature word representations in map estimation for topic modeling. In *Proceedings of the Australasian Language Technology Association Workshop 2015*, pages 116–121, 2015b.

Mor Nitesh, Riva Oriana, Nath Suman, and Kubiatowicz John. Bloom cookies: Web search personalization without user tracking. In *Proceedings of the 22nd Annual Network and Distributed System Security Symposium (NDSS '15)*, 2015.

Eli Pariser. *The Filter Bubble: What the Internet Is Hiding from You.* Penguin Group , The, 2011.

Xuan-Hieu Phan, Le-Minh Nguyen, and Susumu Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of the 17th international conference on World Wide Web*, pages 91–100, 2008.

Blaine A Price, Janet van der Linden, Jacky Bourgeois, and Gerd Kortuem. When looking out of the window is not enough: informing the design of in-home technologies for domestic energy microgeneration. *Proc. ICT4S*, pages 73–80, 2013.

Karthik Raman, Paul N. Bennett, and Kevyn Collins-Thompson. Toward whole-session relevance: Exploring intrinsic diversity in web search. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 463–472, 2013.

Ian Ruthven. Re-examining the potential effectiveness of interactive query expansion. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, pages 213–220, 2003.

Sara Salehi, Jia Tina Du, and Helen Ashman. Examining personalization in academic web search. In *Proceedings of the 26th ACM Conference on Hypertext and Social Media*, HT '15, pages 103–111, 2015.

Mark Sanderson and Bruce Croft. Deriving concept hierarchies from text. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 206–213, 1999.

Xuehua Shen, Bin Tan, and ChengXiang Zhai. Implicit user modeling for personalized search. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, CIKM '05, pages 824–831, 2005.

Milad Shokouhi. Learning to personalize query auto-completion. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 103–112, 2013.

Milad Shokouhi and Qi Guo. From queries to cards: Re-ranking proactive card recommendations based on reactive search history. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 695–704, 2015.

Milad Shokouhi, Ryen W. White, Paul Bennett, and Filip Radlinski. Fighting search engine amnesia: Reranking repeated results. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 273–282, 2013.

Ahu Sieg, Ahu Sieg, Bamshad Mobasher, Bamshad Mobasher, Robin Burke, and Robin Burke. Web search personalization with ontological user profiles. In *Proceedings of the 16th ACM conference on Conference on information and knowledge management*, pages 525–534, 2007.

Adish Singla, Ryen W. White, Ahmed Hassan, and Eric Horvitz. Enhancing personalization via search activity attribution. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '14, pages 1063–1066, 2014.

Yang Song, Xiaolin Shi, Ryen White, and Ahmed Hassan Awadallah. Context-aware web search abandonment prediction. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '14, pages 93–102, 2014.

David Sontag, Kevyn Collins-Thompson, Paul N. Bennett, Ryen W. White, Susan Dumais, and Bodo Billerbeck. Probabilistic models for personalizing web search. In *Proceedings of the fifth ACM international conference on Web search and data mining - WSDM '12*, pages 433–442, New York, New York, USA, 2012.

Micro Speretta and Susan Gauch. Personalized search based on user search histories. In *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, WI '05, pages 622–628, 2005.

Sofia Stamou and Alexandros Ntoulas. Search personalization through query and page topical analysis. *User Modeling and User-Adapted Interaction*, 19(1-2):5–33, 2009.

Kazunari Sugiyama, Kenji Hatano, and Masatoshi Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. In *Proceedings of the 13th international conference on World Wide Web*, WWW '04, pages 675–684, 2004.

Xiaohui Tao, Yuefeng Li, and Ning Zhong. A personalized ontology model for web information gathering. *IEEE Trans. Knowl. Data Eng.*, 23(4):496–511, 2011.

Jaime Teevan, Susan T. Dumais, and Eric Horvitz. Personalizing search via automated analysis of interests and activities. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, pages 449–456, 2005.

Jaime Teevan, Meredith Ringel Morris, and Steve Bush. Discovering and using groups to improve personalized search. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, WSDM '09, pages 15–24, 2009.

Jaime Teevan, Susan T. Dumais, and Eric Horvitz. Potential for personalization. *ACM Trans. Comput.-Hum. Interact.*, 17(1):4:1–4:31, 2010.

Yury Ustinovskiy and Pavel Serdyukov. Personalization of web-search using short-term browsing context. In *Proceedings of the 22nd ACM international conference on Conference on information and knowledge management*, CIKM '13, pages 1979–1988, 2013.

Yury Ustinovskiy, Gleb Gusev, and Pavel Serdyukov. An optimization framework for weighting implicit relevance labels for personalized web search. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, pages 1144–1154, 2015.

David Vallet, Iván Cantador, and Joemon M. Jose. Personalizing web search with folksonomy-based user and document profiles. In *Proceedings of the 32nd European conference on Advances in Information Retrieval*, ECIR'2010, pages 420–431, 2010.

Thanh Vu, Alistair Willis, Son N Tran, and Dawei Song. Temporal latent topic user profiles for search personalisation. In *Advances in Information Retrieval: 37th European Conference* *on IR Research, ECIR 2015, Vienna, Austria, March 29 - April 2, 2015. Proceedings*, pages 605–616, 2015a.

Thanh Vu, Alistair Willis, Udo Kruschwitz, and Dawei Song. Personalised query suggestion for intranet search with temporal user profiling. In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval*, CHIIR '17, pages 265–268, 2017a.

Thanh Tien Vu, Dawei Song, Alistair Willis, Son Ngoc Tran, and Jingfei Li. Improving search personalisation with dynamic group formation. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '14, pages 951–954, 2014.

Thanh Tien Vu, Alistair Willis, and Dawei Song. Modelling time-aware search tasks for search personalisation. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, pages 131–132, 2015b.

Thanh Tien Vu, Dat Quoc Nguyen, Mark Johnson, Dawei Song, and Alistair Willis. Search personalization with embeddings. In *Advances in Information Retrieval: 39th European Conference on IR Research, ECIR 2017, Aberdeen, UK, April 8-13, 2017, Proceedings*, pages 598–604, 2017b.

Hanna M. Wallach, David M. Mimno, and Andrew McCallum. Rethinking lda: Why priors matter. In *Advances in Neural Information Processing Systems 22*, pages 1973–1981, 2009.

Hongning Wang, Yang Song, Ming-Wei Chang, Xiaodong He, Ryen W. White, and Wei Chu. Learning to extract cross-session search tasks. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, pages 1353–1364, 2013.

Hongning Wang, Yang Song, Ming-Wei Chang, Xiaodong He, Ahmed Hassan, and Ryen W. White. Modeling action-level satisfaction for search task satisfaction prediction. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '14, pages 123–132, 2014.

Lidan Wang, Paul N. Bennett, and Kevyn Collins-Thompson. Robust ranking models via risk-sensitive optimization. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 761–770, 2012.

Ryen White. Beliefs and biases in web search. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 3–12, 2013.

Ryen W. White. Belief dynamics in web search. *Journal of the Association for Information Science and Technology*, 65(11):2165–2178, 2014.

Ryen W. White and Ahmed Hassan Awadallah. Personalizing search on shared devices. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 523–532, 2015.

Ryen W. White and Eric Horvitz. Belief dynamics and biases in web search. *ACM Trans. Inf. Syst.*, 33(4):18:1–18:46, 2015.

Ryen W. White, Paul N. Bennett, and Susan T. Dumais. Predicting short-term interests using activity-based search context. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 1009–1018, 2010.

Ryen W. White, Wei Chu, Ahmed Hassan, Xiaodong He, Yang Song, and Hongning Wang. Enhancing personalized search by mining and modeling task behavior. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 1411–1420, 2013.

Ryen W. White, Ahmed Hassan, Adish Singla, and Eric Horvitz. From devices to people: Attribution of search activity in multi-user settings. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 431–442, 2014.

F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, pages 80–83, 1945.

Jinxi Xu and W. Bruce Croft. Query expansion using local and global document analysis. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 4–11, 1996.

Jinyun Yan, Wei Chu, and Ryen W. White. Cohort modeling for enhanced personalized search. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '14, pages 505–514, 2014.

Liu Yang, Qi Guo, Yang Song, Sha Meng, Milad Shokouhi, Kieran McDonald, and W. Bruce Croft. Modeling user interests for zero-query ranking. In *Advances in Information Retrieval: 38th European Conference on IR Research, ECIR 2016*, pages 171–184, 2016.

Zhijun Yin, Milad Shokouhi, and Nick Craswell. Query expansion using external evidence. In *Proceedings of the 31st European Conference on IR Research on Advances in Information Retrieval*, ECIR '09, pages 362–374, 2009.

Arjumand Younus, Colm O'Riordan, and Gabriella Pasi. A language modeling approach to personalized search based on users' microblog behavior. In *Proceedings of the 36th European Conference on IR Research on Advances in Information Retrieval - Volume 8416*, ECIR 2014, pages 727–732, 2014.

Dong Zhou, Séamus Lawless, and Vincent Wade. Improving search via personalized query expansion using social media. *Inf. Retr.*, 15(3-4):218–242, 2012.

Appendix A

Data Analysis on the Essex Intranet Query Logs

A.1 User Interactions in Different Time Granularities

In this session, we do statistical analyses to show the change of the user's search pattern in different time granularities. We start with the user's search pattern during a day. Figure A.1 shows the average number of events in different hours during a day in the Essex query logs. From the Figure 10, we can see that the user interaction was highest during the lunchtime. This searching pattern of users during a day has a strong correlation with the working and studying habit of students and university staffs. Specifically, the average number of user search activities increased significantly from about 100 at 8 AM to reach a peak of more than 250 during the lunchtime. After that, the number decreased gradually to 100 at 11 PM.

Similarly, Figure A.2 shows the average number of user interactions in different days during a week. Because all of the classes at the Essex University are normally off during the weekend, the number of user interactions was lowest on Saturday and Sunday with nearly 1,900 and 2,100 search interactions, respectively. Figure A.3 and Figure A.4 show the average numbers of user interactions in different days and different months during a year, respectively. The number is again strongly correlated to the studying time of the university. It was highest in October (the

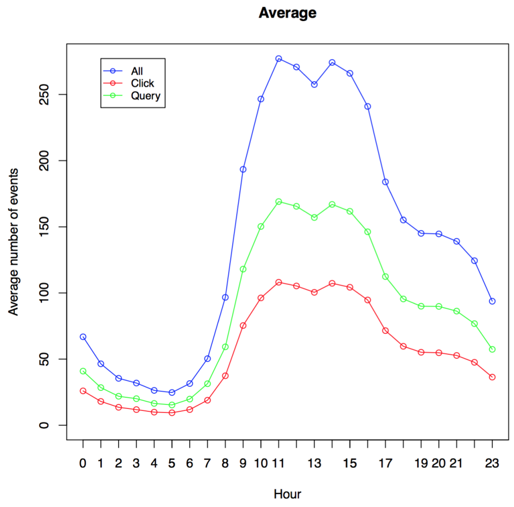


Figure A.1: The average number of events in different hours during a day in the Essex logs

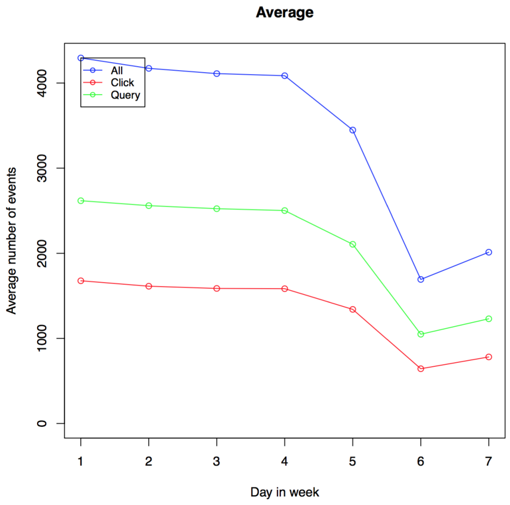


Figure A.2: The average number of events on different days during a week

Enrollment time) with more that 150,000 interactions and lowest in December (the Christmas time) with just below 60,000 interactions.

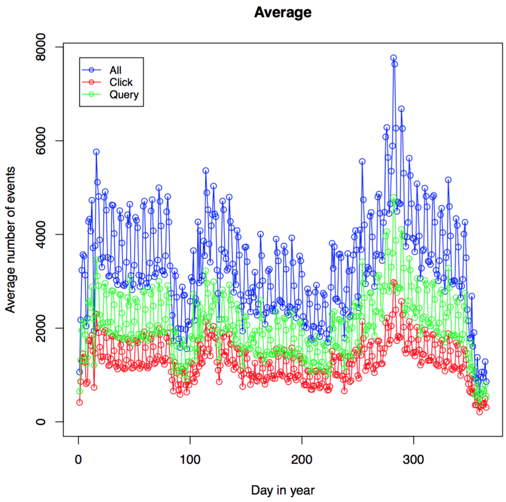
A.2 Query Analysis

Table A.1 shows the basic statistic of query events in the Essex data. The number of queries in 2013 decreased by 13%. However, the average number of terms per query went up from 1.80 in 2012 to 1.83 in 2013. Of about 1.4 million query events, more than 46% (about 660 thousand events) are single term queries (e.g., Webmail, Essex, etc.).

Table A.1: Basis statistics of query events

Item	2012	2013	Total
#queries	757,645	659,284	1,416,929
#single term queries	359,065	302,225	661,290
#average terms per query	1.80	1.83	1.81

In Figure A.5, we show the distribution of the query length in the Essex query logs. It can be seen that most queries consisted of less than or equal to 3 terms. The percentage of the



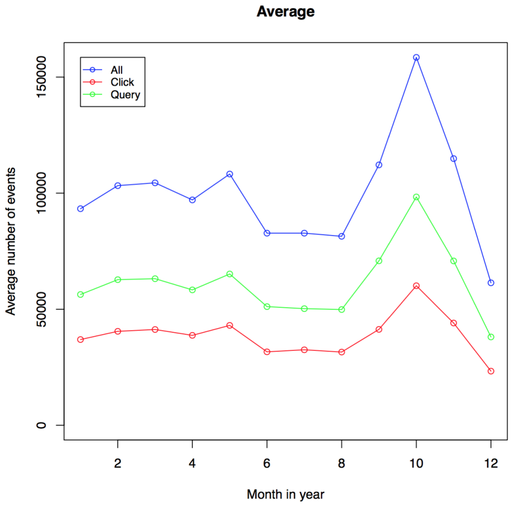


Figure A.3: The average number of events on different days during a year

Figure A.4: The average number of events in different months during a year

query containing either less than or equal to 3 terms are about 94% and almost identical over two years, 2012 and 2013. Regardless of the years, there were very few queries containing more than 3 terms. The numbers for the two years are 5.6% and 5.8%, respectively.

To understand to query term distribution, we build a word cloud of 200 most popular words appeared in the Essex query events (as shown in Figure A.6). The word size in the figure describes the popularity of the word (i.e., the big word means more popular than the smaller one. For example, the “Moodle” is more popular than “Timetable”.

To understand the question of why the user changed queries during a search session, we made a statistical analysis on the frequent query pairs. Table A.2 shows the top ten most popular query pairs in the Essex logs. The most popular one is the “accomodation → accommodation” pair, in which the user corrected the typo on “accomodation” query. However, in general, the user changed the input query during their search session because they either added more term to the previous query (e.g., “moodle → moodle university”) or explained the short query with a longer and more meaningful one (e.g., “ocs → coursework submission”).

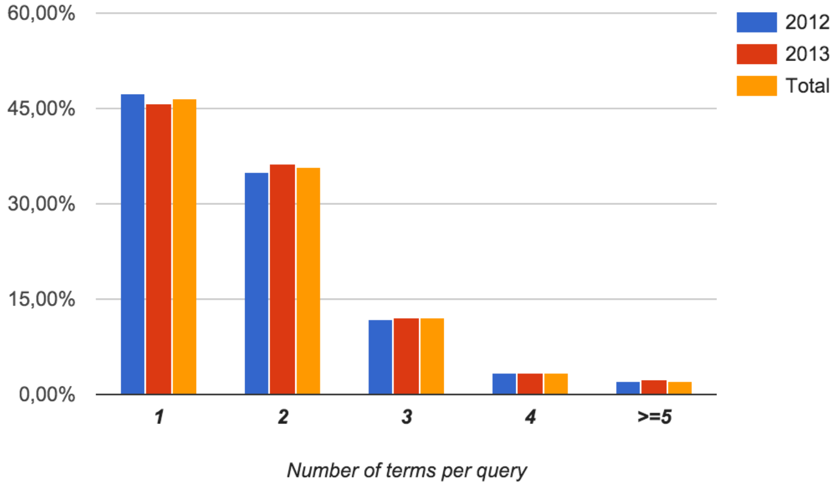


Figure A.5: The percentage of the queries that contained n terms

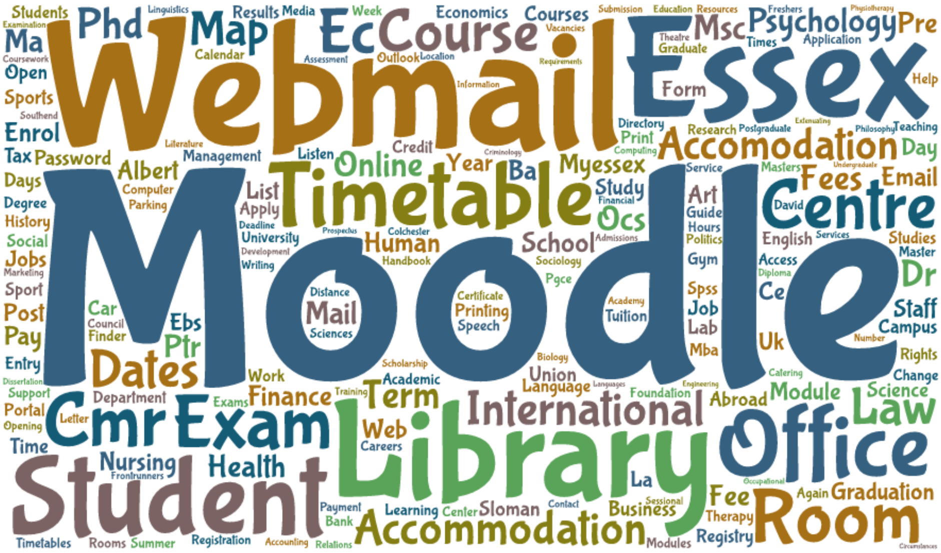


Figure A.6: Word cloud of the search query

Table A.2: Frequent query pairs

Query 1	Query 2	Frequency	Refinement Type
accomodation	accommodation	3377	Typo
moodle	moodle university	2106	Expansion
ocs	coursework submission	947	Explanation
map	campus map	841	Expansion
timetable	timetables	821	Expansion
nursing	adult nursing	797	Expansion
webmail	myessex webmail	648	Expansion
jobs	job vacancies	573	Explanation
accomodation	accomodation office	489	Expansion (typo)
ocs	online coursework	468	Explanation
coursework submission	online coursework submission	402	Expansion

A.3 Click Analysis

Table A.3 describes basic statistics of click events in the Essex query logs. As can be seen from the table, the number of click events went down significantly by 16% from more than 505,000 in 2012 to just below 425,000 in 2013. However, the average numbers of click per search session remained stable with 1.27 and 1.26 in the years of 2012 and 2013, respectively. The average number of click per URL decreased remarkably from 24.26 in 2012 to 16.73 in 2013. Similarly, the max number of click per URL reduced significantly from about 33,000 in 2012 (https://moodle.essex.ac.uk/) to only about 23,000 in 2013 (https://email.essex.ac.uk/) by 30%. Among 930,000 click events, there are only about 36,000 distinct clicked URL. Figure A.7 shows top 200 most popular URLs in term of number click per URL. The big URL is more popular than the smaller one. From the figure, we see that the clicked document is also domain-specific.

In Figure A.8, we describe the distribution of the number of click per URL. We can see that most URLs were clicked less that 15 times during two years 2012 and 2013. Specifically, in the Essex dataset, 35.56% and 15,74% were clicked only one and two times, respectively.

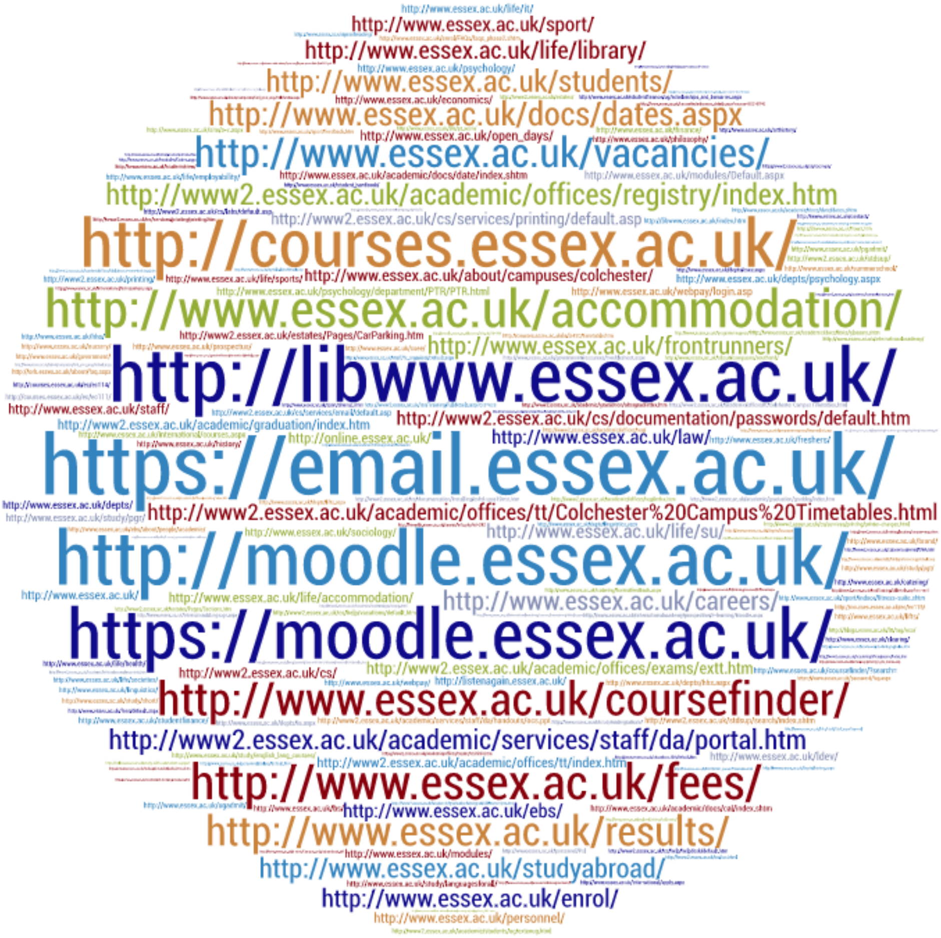

http://www.essex.ac.uk/life/it/
http://www.essex.ac.uk/sport/
http://www.essex.ac.uk/life/library/
http://www.essex.ac.uk/psychology/
http://www.essex.ac.uk/students/
http://www.essex.ac.uk/economics/
http://www.essex.ac.uk/docs/dates.aspx
http://www.essex.ac.uk/open_days/
http://www.essex.ac.uk/philosophy/
http://www.essex.ac.uk/vacancies/
http://www.essex.ac.uk/academic/docs/date/index.shtm
http://www.essex.ac.uk/modules/Default.aspx
http://www2.essex.ac.uk/academic/offices/registry/index.htm
http://www2.essex.ac.uk/cs/services/printing/default.asp
http://courses.essex.ac.uk/
http://www.essex.ac.uk/life/sports/
http://www.essex.ac.uk/about/campuses/colchester/
http://www.essex.ac.uk/depts/psychology.aspx
http://www.essex.ac.uk/psychology/department/PTR/PTR.html
http://www.essex.ac.uk/webpay/login.asp
http://www.essex.ac.uk/accommodation/
http://www2.essex.ac.uk/estates/Pages/CarParking.htm
http://www.essex.ac.uk/frontrunners/
http://libwww.essex.ac.uk/
http://www.essex.ac.uk/staff/
http://www2.essex.ac.uk/cs/documentation/passwords/default.htm
http://www2.essex.ac.uk/academic/graduation/index.htm
http://online.essex.ac.uk/
http://www.essex.ac.uk/law/
http://www.essex.ac.uk/freshers/
https://email.essex.ac.uk/
http://www.essex.ac.uk/depts/
http://www.essex.ac.uk/study/pgr/
http://www2.essex.ac.uk/academic/offices/tt/Colchester%20Campus%20Timetables.html
http://www.essex.ac.uk/sociology/
http://www.essex.ac.uk/life/su/
http://moodle.essex.ac.uk/
http://www.essex.ac.uk/
http://www.essex.ac.uk/life/accommodation/
http://www.essex.ac.uk/careers/
https://moodle.essex.ac.uk/
http://www2.essex.ac.uk/cs/
http://www2.essex.ac.uk/academic/offices/exams/extt.htm
http://listenagain.essex.ac.uk/
http://www.essex.ac.uk/coursefinder/
http://www2.essex.ac.uk/academic/services/staff/da/portal.htm
http://www2.essex.ac.uk/academic/offices/tt/index.htm
http://www.essex.ac.uk/ldev/
http://www.essex.ac.uk/fees/
http://www.essex.ac.uk/ebs/
http://www.essex.ac.uk/results/
http://www.essex.ac.uk/modules/
http://www.essex.ac.uk/studyabroad/
http://www.essex.ac.uk/enrol/
http://www.essex.ac.uk/personnel/

Figure A.7: Top 200 most popular URLs

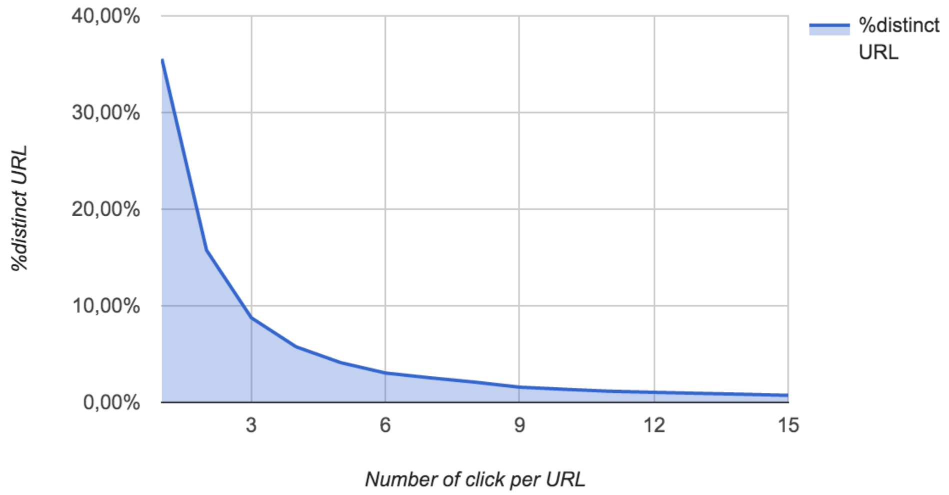


Figure A.8: The percentage of distinct URL vs the number of click per URL

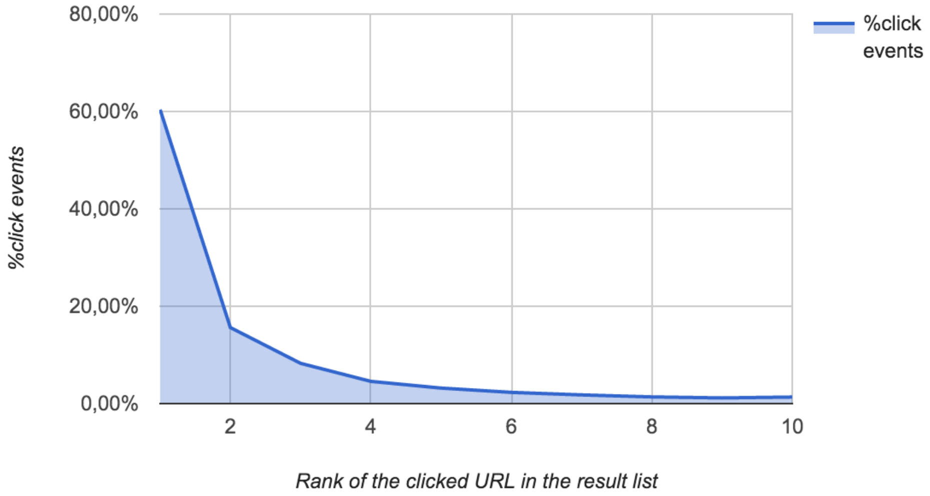


Figure A.9: The percentage of click events vs the rank of the clicked URL. As can be seen, most users did not bother to look beyond the result ranked lower than 4

Table A.3: Basic statistics of click events

Item	2012	2013	Total
#search session	397,461	338,391	735,804
#clicked url	505,534	424,708	930,242
#click/session	1.27	1.26	1.26
#distinct clicked url	20,842	25,389	36,222
Average #click/url	24.26	16.73	25.68
Max #click/url	33,675[1]	23,434[2]	55,415[3]

Figure A.9 indicates the distribution of the rank of clicked document. From the figure, we see that most users only clicked on the first and the second-ranked document. Specifically, the number of those clicks account for 76% click events. There are only 11% click events where the user clicked on the lower ranked document (i.e., ranked from 5 to 10). The decreasing viewing of the lower rank document could be explained by the fact that the user is biased in the document ranking, in which the user only pays attention to the higher ranked document.

Appendix B

P-values of Significance Test

Table B.1: p-values of the paired t-test comparing the search personalisation models with the Bing default ranker in Chapter 4 with the IAR metric

Model	*IAR*
S_Profile	2.39E-07
S_Group	3.88E-12
D_Group	1.97E-13

Table B.2: p-values of the paired t-test comparing the temporal models with the Bing default ranker in Chapter 5 with different metrics

Model	*MAP*	*P@1*	*P@3*	*MRR*	*nDCG@5*	*nDCG@10*
LON	5.25E-04	2.40E-04	1.00E-05	1.11E-05	1.03E-06	2.04E-07
DAI	1.31E-16	1.75E-14	5.06E-11	2.96E-20	5.67E-19	1.27E-21
SES	2.02E-46	3.51E-36	5.59E-19	2.04E-39	3.85E-41	4.81E-38
ALL	8.98E-47	2.33E-49	2.22E-28	6.21E-43	2.74E-43	7.35E-50

Table B.3: p-values of the paired t-test comparing the temporal models with the non-temporal model in Chapter 5 with different metrics

Model	*MAP*	*P@1*	*P@3*	*MRR*	*nDCG@5*	*nDCG@10*
LON	7.18E-10	4.41E-05	1.09E-03	2.34E-06	1.08E-06	2.70E-05
DAI	3.05E-13	2.15E-16	2.48E-11	2.47E-19	2.82E-22	3.68E-19
SES	6.36E-40	4.53E-31	1.71E-17	1.39E-37	1.32E-36	2.09E-44
ALL	5.58E-49	1.80E-42	8.30E-27	1.75E-54	4.30E-42	5.07E-43

Table B.4: p-values of the paired t-test comparing the temporal models with the non-temporal model in Chapter 5 with different query click entropies

Model	0-0.5	0.5-1.0	1.0-1.5	1.5-2.0	> 2.0
LON	2.95E-05	6.47E-04	5.55E-12	6.22E-29	2.71E-25
DAI	4.72E-09	2.14E-20	1.42E-35	6.42E-69	1.38E-68
SES	4.67E-24	2.87E-41	4.53E-58	3.31E-94	1.53E-102
ALL	2.60E-23	6.89E-58	3.13E-79	4.18E-104	4.11E-123

Table B.5: p-values of the paired t-test comparing the TimeTask model with other models in Chapter 6 with different metrics

Other Model	*MAP*	*P@1*	*P@3*	*MRR*	*nDCG@5*	*nDCG@10*
Default	1.36E-45	4.19E-45	1.17E-22	6.01E-47	3.25E-45	1.81E-45
ShortTerm	7.53E-17	6.34E-23	4.64E-06	9.48E-24	3.40E-13	7.15E-16
StaticTask	3.90E-07	8.13E-06	4.44E-03	8.29E-04	1.70E-04	1.69E-06

Table B.6: p-values of the paired t-test comparing the TimeTask model with other models in Chapter 6 with different query click entropies

Other Model	0-0.5	0.5-1.0	1.0-1.5	1.5-2.0	> 2.0
Default	8.45E-31	3.57E-55	5.70E-77	1.55E-109	2.20E-104
ShortTerm	2.14E-13	3.28E-20	2.92E-24	2.28E-43	3.88E-35
StaticTask	2.30E-03	6.01E-03	7.13E-06	5.27E-16	8.07E-20

Table B.7: p-values of the paired t-test comparing the personalisation models with the Adeyanju's model in Chapter 7 with different metrics

Model	*MAP*	*P@1*	*P@5*	*MRR*	*nDCG@5*	*nDCG@10*
Click	3.51E-35	6.25E-11	9.78E-33	1.38E-29	4.01E-33	1.15E-43
Both	2.81E-41	1.92E-13	6.46E-33	3.52E-32	2.15E-35	3.43E-68

Table B.8: p-values of the paired t-test comparing the personalisation models with the Adeyanju's model in Chapter 7 with different positions

Model	1	2	3	>= 4
Click	*NaN*	1.90E-28	7.78E-33	3.09E-34
Both	3.08E-38	1.10E-24	2.48E-30	1.05E-42

Table B.9: p-values of the paired t-test comparing the personalisation models with the Adeyanju's model in Chapter 7 with different lengths

Model	1	2	3	>= 4
Click	1.77E-32	7.34E-31	4.45E-43	2.42E-71
Both	2.88E-30	5.88E-21	2.37E-40	6.53E-94