Utilizing External Resources for Enriching Information Retrieval

Jinming Min
School of Computing
Dublin City University

Submitted for Degree of Doctor of Philosophy
Supervisor: Prof Gareth J.F. Jones

September 2017

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: 

(Candidate) ID No: 58115366

Date:

Acknowledgements

I would like to acknowledge my supervisor Professor Gareth J.F. Jones, for all his help and advice during my studies at DCU. It was Gareth who introduced me the methods of doing research in the right way and made me to be a better researcher. Since I am not a native English speaker, Gareth also gives me countless help in improving the English skills. During the gap in my studies, Gareth gave me continuous support to encourage me to keep working on the thesis. Without Gareth's continuous support, there would have been no the thesis.

During my time at the School of Computing, it was good to work with Dr. Johannes Leveling. Johannes gave me much help in the beginning time of the Ph.D. program. He helped me to analyse the research results in the right way. The help from Johannes was very important for several research papers. These papers have been developed into several chapters in this thesis.

My thanks to the Centre for Digital Content and Media Innovation (ADAPT, formerly CNGL), which funded me for the Ph.D. program. The program not only provided the funding, but also give me lots of useful working experiences. ADAPT is a world-class working place where I was able to meet many experienced researchers from different backgrounds. These experiences will be very helpful in my future career.

I also thank for Microsoft Ireland which gave me the opportunity to serve as an intern. Microsoft may be the best place to work on computer science in the world.

In the end, all the courage to persevere in the thesis is from my wife and parents. Without their solid support, I could have given up at any time. Without their continuous support, there would be no this thesis. Thank you, my family!

Contents

List of Figures

List of Tables

Abstract

Jinming Min

Utilizing External Resources for Enriching Information Retrieval

Information retrieval (IR) seeks to support users in finding information relevant to their information needs. One obstacle for many IR algorithms to achieve better results in many IR tasks is that there is insufficient information available to enable relevant content to be identified. For example, users typically enter very short queries, in text-based image retrieval where textual annotations often describe the content of the images inadequately, or there is insufficient user log data for personalization of the search process. This thesis explores the problem of inadequate data in IR tasks. We propose methods for Enriching Information Retrieval (ENIR) which address various challenges relating to insufficient data in IR. Applying standard methods to address these problems can face unexpected challenges. For example, standard query expansion methods assume that the target collection contains sufficient data to be able to identify relevant terms to add to the original query to improve retrieval effectiveness. In the case of short documents, this assumption is not valid. One strategy to address this problem is document side expansion which has been largely overlooked in the past research. Similarly, topic modeling in personalized search often lacks the knowledge required to form adequate models leading to mismatch problems when trying to apply these models to

improve search. This thesis focuses on methods of ENIR for tasks affected by problems of insufficient data. To achieve ENIR, our overall solution is to include external resources for ENIR. This research focuses on developing methods for two typical ENIR tasks: text-based image retrieval and personalized web data search.

In this research, the main relevant areas within existing IR research are relevance feedback and personalized modeling. ENIR is shown to be effective to augment existing knowledge in these classical areas. The areas of relevance feedback and personalized modeling are strongly correlated since user modeling and document modeling in personalized retrieval enrich the data from both sides of the query and document, which is similar to query and document expansion in relevance feedback. Enriching IR is the key challenge in these areas for IR. By addressing these two research areas, this thesis provides a prototype for an external resource based search solution. The experimental results show external resources can play a key role in enriching IR.

Chapter 1

Introduction

Information retrieval (IR) is one of the major research topics in computer science. IR seeks to find material (conventionally documents) of an unstructured nature (conventionally text) that satisfies a user information need from within large collections (conventionally stored on computers). In a typical IR framework, the user inputs a query to the system, the IR system returns a ranked list related to this user query with items ranked in decreasing order of likelihood of relevance to the information need. Early IR systems were commonly used by professional librarians or academic researchers. With the emergence and rapid growth of the Internet, the most visible IR application - web search engine - appeared. Web search has transformed IR to be a core technology for online users to locate information to support them in their daily activities. In state-of-the-art IR, three important changes have occurred: users of IR systems have changed from qualified librarians or scholarly researchers to the average online user; the data size of IR systems has moved from small document sets to collections of huge size; the rapid growth of the Internet has

increased the range of types of data to be searched by IR applications, such as metadata search, mutimedia search, personalized search, social search and etc.

The Internet is one of the main areas of application for IR technologies. The Internet is not only a place for authoritative sources to publish information, it is also a place for everyone to share their own information with others. The information published by average users is usually not as complete and comprehensive as the information from authoritative sources. These new data types bring challenges for IR technologies. Although the data size is typically huge, one obvious characteristic of Web data is that there is no rich context to make it self-supported. Human beings are able to understand incomplete information which is not explicitly written by its author if they have the relevant background knowledge, but this is hard for computers. Without sufficient context and background information for incomplete Web data, classical IR algorithms developed based on full length articles will typically achieve weaker performance on these new data types. Thus new IR approaches are needed to help computers find relevant information for web data which is not informative enough. Motivated by these challenges for IR, we propose the topic of this thesis research of enriching IR applications using external resources.

The main interest of this thesis is to enrich IR using information from external resources. One obstacle for many IR algorithms to achieve better results is that the data is too sparse to enable the relevant content to be identified. For example, users typically enter very short queries in text-based image retrieval, in which textual annotations often describe the content of the images

Image	Annotation
	Downhill mountain bike

Figure 1.1: Example of sparse data.

inadequately. In Figure 1.1, the annotation only gives a simple description, which is a very typical situation for online images. But more information is needed for effective and reliable text-based image retrieval. With a richer annotation of the image, we can know for example where the image was taken and who the person in the image is, etc. This inadequate labelling of online images potentially results in poor performance of standard IR algorithms to find useful information for users. Addressing sparse data problems by using external information is the main focus of this thesis.

1.1 Motivation and Applications

Typical IR algorithms assume there is sufficient information available in the user queries and the target documents to enable effective retrieval. This is due to many IR algorithms being developed for tasks such as retrieval of news

articles for benchmark tasks such as the TREC Ad-Hoc evaluation tasks [1]. In this kind of search task, the queries and target documents usually contain enough and precise information to describe the user intent and the document content. But in many real-world IR tasks, this assumption may not be true. The rapid growth of the Internet brings new types of data for IR. In many of these new IR applications, the data can be too sparse to describe itself. Several typical IR tasks where a sparse data problem occurs are:

Image Search Image annotations usually contain very few terms to describe the content of the images. Text-based image retrieval relies on these sparse annotations to find relevant images. This is still the mainstream solution to the image search task.

Video Search Similar to image search, text-based video search relies heavily on sparse text annotations of videos.

Micro-blog Search Micro-blog documents are usually composed of very short sentences which do not provide enough detail for effective search.

Social Network Search In typical social networks, there are usually no complete texts which describe the events being referred to since users only use very simple text to describe their activities.

Chat Messages Search Dialogue in online conversations usually resemble spoken sentences, and are usually short and incomplete. The background information is known for the person in the conversation, but is not available to the computer.

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

SMS Search Short messages sent on a mobile devices usually contain very short messages from mobile users without context.

Alternative strategy for improving IR effectiveness where the search task is adequately described is personalization. The purpose of personalization is to provide different retrieval results for different users by exploring knowledge of the interests of the specific user making the search. In personalized retrieval tasks, a key step is to build a user model based on the user's historical data. However, the data available is usually too sparse to create a rich and comprehensive model of the user's interests. This is a typical sparse data problem in IR. Thus, in personalized retrieval, one challenge is how to provide personalized search results with sparse user historical data.

In the meantime, the growth of the Internet provides opportunities to resolve sparse data problems in IR since it provides large amounts of data which could be utilized in the retrieval process. In the last ten years, Internet data has rapidly grown into many billions of web pages. Table 1.1 shows the estimated number of web pages from Google's web index.

Table 1.1: Size of Google Index.

Year	Estimated Number of Web Pages
2005	11.50 billion
2009	25.21 billion
2012	55.00 billion

The huge size of Internet data provides an opportunity to resolve the sparse data problem in IR research. Internet data may contain suitable content to address the incomplete information in the target documents in IR tasks; Internet data may contain content relevant to the user information need suitable

for enriching the user query; and online reference sites contain references to many topics of general human knowledge, extracting information from this data may be used to enrich the queries or documents on the same topics in many IR tasks where the sparse data problem is present.

For example, the well-known Wikipedia archive contains a large number of articles relating to general human knowledge. Wikipedia can potentially be used as a general resource for resolving the sparse data problem in search tasks. Several other online reference sites are shown in Table 1.2 [1]. These reference sites can also potentially be very useful resources for enriching data used in IR tasks. PageRank, Alexa Rank and the number of monthly visitors shown in Table 1.2 suggest that these sites are popular sites for online users seeking to acquire general information to satisfy their search interests.

Table 1.2: Wikipedia in Reference Sites

	PageRank	Alexa Rank	Monthly Visitors
Wikipedia	**9**	**8**	**41,422,790**
Answers.com	7	309	10,607,121
HowStuffWorks	8	1,416	3,240,959
Encyclopaedia Britanica	8	4,370	1,329,460
Infoplease	7	4,692	1,463,272

1.2 Hypothesis

Based on the motivation of the previous section, we know that one of the critical challenges for IR in many retrieval tasks is the sparse data problem

[1] The data are collected in January, 2008

and that Internet data provides a potential opportunity to relieve it. Our research aims to utilize external resources to resolve the sparse data problem in IR tasks. Usually, IR tasks contain three kinds of data: user queries, target documents, and user historical data. We conduct research into utilizing external resources in all these parts. We select two tasks as the focus for our research: text-based image retrieval and personalized web data search. These two tasks are selected due to the challenges of the sparse data problem in the three components of these tasks: the query, the target documents and the user data. Text-based image retrieval is a typical task where the target corpus and the user queries do not contain adequate information, and personalized web search is a typical task where there is the lack of user historical information. We propose methods to utilize external resources to enrich IR in these two tasks. To utilize external resources for user queries and target documents, our methods use the classical method of relevance feedback. For personalized search, we expand the user data from external resources before building the user search interests model. Thus the main hypotheses of the thesis can be summarized as follows:

- External resources can be incorporated into the relevance feedback process to provide better feedback information for retrieval tasks with the sparse data problem. The enrichment could be helpful from both the query and document sides of the retrieval process.

- External resources can enrich user historical data, thus enabling user models based on user historical data to model more user search interests. This will help the retrieval system to provide effective personalized

search results to an individual user.

Based on these hypotheses, we specify our research described in this thesis into the research questions in Section 1.3.

1.3 Overview of the Thesis

Currently there is no comprehensive study into the utilization of external resource in IR applications. Although there exists empirical research on utilizing external resource in some areas such as query expansion, it fails to provide systemic conclusions for the topic of external resources use in IR. This thesis addresses a number of critical problems regarding utilizing external resources to enrich IR research. It aims to establish the potential of external resources for IR techniques, such as relevance feedback and personalized web data search. The thesis focuses on two typical IR tasks where insufficient data occurs: text-based image retrieval and personalized web data search. The research content of the thesis is summarized in Table 1.3:

Table 1.3: Structure of the thesis.

Research tasks	Algorithms	
Image search	chap 3: query expansion	chap 4: document expansion
Personalization	chap 5: term model	chap 6: topic model

We separate the research into four parts with an overview in the following subsections. In the beginning of each subsection, several research questions are listed which the thesis aims to answer.

1.3.1 Query Expansion

Query expansion is a classical solution to the query/document mismatch problem in IR method [Xu & Croft, 1996]. The basic assumption of a typical query expansion method is that the target corpus contains sufficient data to enrich the original query to form a longer query. Results in previous research conclude that longer queries achieve better retrieval effectiveness in various IR tasks [Buckley *et al.*, 1994b; Rocchio, 1971]. But in many new IR tasks such as short document retrieval, this assumption may be not true due to the lack of information in the target corpus.

In this part of our research, we aim to discover whether utilizing external resources performs well for IR tasks with sparse data problem. We conduct our research by answering the following research questions:

- How does query expansion perform for retrieval tasks with sparse information? The purpose of this research question is intended to find the limitation of the classical query expansion on IR tasks with the sparse data problem.

- Is query expansion from the target collection or query expansion from an external collection? The purpose of this research question is intended to discover whether the utilization of external resources plays a positive role for IR tasks with the sparse data problem.

- Are classical query expansion methods the best for query expansion using external resources? The purpose of this research question is intended to discover whether alternative methods can be utilized with

external resources for IR tasks with the sparse data problem to produce results better than those achieved using classical methods.

In this part of the thesis, we explore the utilization of external resources on a classical query expansion algorithms. For a text-based image search task, we propose to compare external query expansion with classical query expansion. Furthermore, we propose a definition-based query expansion method to utilize Wikipedia as the external resource. This method not only utilizes the overall external corpus as the resource to enrich the original query, it also utilizes the knowledge of definition documents which directly explain the key concept of the user query.

1.3.2 Document Expansion

Document expansion has been a less investigated topic in IR research. There are some negative reports about the utilization of document expansion in TREC search tasks [Billerbeck & Zobel, December 2005]. These show that document expansion for news articles research does not yield significant improvement for retrieval effectiveness. While in our research, the problem in IR we want to resolve is the sparse data problem. In this setting, document expansion may show different behavior to that found for the TREC search tasks. Some research questions we are addressing in this topic are listed as:

- Is document expansion effective on retrieval tasks with sparse information? The purpose of this research question is to test whether document expansion from external resources can help to improve retrieval effectiveness for IR tasks with the sparse data problem.

- Is using the whole document as a query to find relevant documents the optimal approach for document expansion research? For this research question, we aim to find the most effective way to form queries for document expansion in our research.

In this part of the thesis, we investigate document expansion for the text-based image retrieval task. Rather than using document expansion from the target corpus, we introduce external document expansion from Wikipedia. A typical document expansion algorithm uses the whole document as the query to find relevant documents in external corpus [Singhal & Pereira, 1999a]. We introduce a method we refer to as *document reduction* to select the most important terms in a document to form a document "query". This query is sent to the external resource to identify the best feedback documents for the original document. A new expanded document is formed by combining the original document terms with the feedback terms obtained from the top ranked external documents.

1.3.3 Term Model on Personalization

Personalization is an important topic in IR since next generation search system targets seeks to improve their effectiveness by providing different search results to individual users. In personalized IR, the most important component is to model the user's search interests and the target documents into the same knowledge base. Since using external resources for personalized search task is a new topic, we need to answer the following research questions to test whether the external resources can be helpful for improving the retrieval

effectiveness for personalized search task:

- Can widely available external resources be used for effective personalized IR? For this research question, we propose to find a method to utilize external resources for building user models in personalized search task.
- Is there a simple and effective solution to a general personalized web data search task? For this research question, we aim to propose a simple method to utilize external resources for ranking documents in personalized search tasks.

In this chapter, we propose an external resource based knowledge system to model the user search data and web documents on the term level. User search interests and web documents can be presented as vectors of terms. Thus how much the target web documents are interesting to the user can be described by the similarity of the user interests vector and the document vector in the same knowledge base.

1.3.4 Topic Modeling in Personalization

Topic modeling is a major breakthrough in recent research in Natural Language Processing (NLP), and has been applied widely in IR tasks [Wei & Croft, 2006]. Modeling a document into topics is a fundamental problem in NLP research. In personalization, one key step is to model the user's search interests from their historical documents. Topics can be used to model the user's search interests. Building a topic model from the user's historical documents is a natural way for modeling user search interests. Thus, the technique

of the topic model is naturally potentially useful in personalization tasks. The challenge in a personalization task is the lack of user data to model the user's search interests. In this research, we propose to utilize external resources for building topic models from user data in a personalized search task. Some research questions we are addressing in this research are:

- Can a topic modeling framework be used to model the user and documents for personalized search task? The purpose of this research is to seek to find a way to utilize topic modeling for user modeling in personalized search task.
- Can external resources be utilized in user modeling and document modeling? The purpose of this research is to seek to utilize external resources for user modeling in personalized search task.
- Can external resources based user models be used to effectively rank documents in a personalized search task in a learning-to-rank framework? The purpose of this research question is to investigate the utilization of external resources for ranking documents in a personalized search task.

We propose a method to update the user model from an external resource - a web collection. The similarity between the updated user model and document model can be used as metrics to evaluate the topic relevance between the user and the document. All the user models and the document model in our search are produced using a standard topic model algorithm - Latent Dirichlet Allocation.

In chapter 2, we present background information related to our thesis, and survey the related work. After chapter 2, we then proceed in subsequent chapters with each of the four main parts of our investigation: query expansion, document expansion, term modeling on personalization, topic modeling on personalization.

Chapter 2

Background and Review

As introduced in Chapter 1, this thesis is focused on utilizing external resources to resolve the sparse data problem in IR. In this chapter, we give a more formal introduction to this problem and review past research on this topic. To introduce the sparse data problem in IR, we review different aspects of IR data including user queries, target documents, and user historical data. To review past research on resolving the sparse data problem, we analyse the advances of the past research and use this to motivate our research.

In past research in IR, there has been no significant work which explicitly aims to resolve the sparse data problem by using external resources. Related work to our research can be found in works on relevance feedback [Rocchio, 1971; Ide, 1968; Robertson, 1991], personalized search [Liu *et al.*, 2002; Pretschner & Gauch, 1999] and utilizing external resources in IR [Diaz & Metzler, 2006]. We review this existing work, build connections between it and then, based on this work to motivate our research on the utilization of external resources on the sparse data problem in IR.

This chapter is structured as follows: Section 2.1 introduces previous research on addressing the sparse data problem in IR, especially focusing on work on relevance feedback and personalized search. Section 2.2 introduces existing methods used for utilizing the external resources in IR. Section 2.3 summarizes previous research of resolving sparse data problem and illustrates the opportunity of our proposed methods for this problem and Section 2.4 provides a summary of this chapter.

2.1 Review of Sparse Data Problem in IR Research

Classical IR uses the statistics of natural language to build retrieval models. The classical probabilistic retrieval model Okapi BM25 [Robertson & Spärck Jones, 1994] relies on a variant of Term Frequency (TF) and Inverse Document Frequency (IDF) as the key components. However, a problem in many IR tasks is that a lack of sufficient text data means that IR model may not be trained effectively which can harm the retrieval results. An example is that the values of TF for many terms could all be 1 in a short documents retrieval task. For the short user query (the usual case for most IR tasks), TF for most query terms could be 1. Thus TF cannot make an effective contribution to weighting the importance of terms in a user query or document in these situations. Also in many IR tasks, the data can be sparse where the content is not described fully, which can lead to a mismatch problem between the query and the document. Although the document and query may be relevant to each other in this case, the retrieval models could still fail to match them due to the relevant queries and documents using different vocabulary to describe

themselves. The query/document mismatch problem can be viewed as a form of sparse data problem. Without sufficient data, the mismatch problem could happen. In this situation, a user query may not contain the term in the relevant target document, and the relevant target document may not contain the term found in the user query.

The sparse data problem happens in many places of IR tasks such as user queries, target documents and user historical search data. The reasons for such sparse data problem can be:

- User queries are usually short and sparse. This is a feature of many IR tasks, particularly for non-professional searchers.
- Target documents are short in some IR tasks. This happens for many emerging IR tasks such as the text-based image retrieval tasks, social network retrieval tasks such as the search tasks on the Twitter.com, or Facebook.com where the user posts are usually very short.
- Historical user search data is not sufficiently complete to describe the full extent of the user's search interests. This happens in many personalized search tasks where there is not sufficient user historical data available.

The sparse data problem in user queries and target documents can increase the likelihood and impact of query/document mismatch. The sparse data problem in user historical data can lead to the user/document topic mismatch, since the user historical data fails to record the user's search interests, resulting in the the calculation of the personal relevance between the user and target documents may not be reliable.

To resolve these mismatch problems arising from the sparse data problem, a classical strategy to adopt is relevance feedback. In the next section, we review the past efforts in using relevance feedback to resolve the sparse data problem in IR. Also, we review research on personalized search where the sparse data problem has been less noted. We review methods to utilize external resources in IR research which motivate our work described in this thesis where we seek to utilize external resources to resolve the sparse data problem in IR.

2.1.1 Relevance Feedback as a Solution to the Sparse Data Problem

In modern IR systems, the user is typically asked to input a simple text query to describe his information need. The query is sent to the IR system to conduct an initial retrieval, in response to which a ranked list of potentially relevant documents is returned to the user. Several problems can be observed to occur with this basic approach to using an IR system:

- The user may not be knowledgeable about the subject of their information need to form a useful query to describe his information need.
- The formed query may be too short to describe the user's information need sufficiently to reliably identify potentially relevant documents.
- Relevant documents in the target collection may use a different vocabulary to describe the content of the user's information need than that used by the user in the query.

These problems can be summarized as the sparse data problem in the user queries and target documents of IR tasks. Relevance Feedback (RF) in IR was introduced as a mechanism which seeks to relieve these problems [Rocchio, 1971; Ide, 1968]. The basic idea of RF is that the user is asked to provide relevance judgments for the top-ranked documents after the initial retrieval run. This feedback information is combined with the initial query to revise the query and/or the parameter of the IR system prior to carrying out a second retrieval run. With the expectation of helping to identify relevant documents more effectively, The RF process can be conducted iteratively until the user's information need is satisfied or there is no further improvement in retrieval effectiveness.

When the user activity provides relevance judgments in this way, this process is referred to as *explicit* relevance feedback [White *et al.*, 2002]. Since users are often reluctant to provide relevance information in this way, the top-ranked documents can be assumed all to be relevant to the user query. This fully automatic process is called *blind* (*pseudo*) relevance feedback. Since blind RF will often assume that non-relevant documents are relevant as well as relevant ones, its effectiveness is on average lower than that of explicit RF.

Early work on relevance feedback can be found in [Rocchio, 1971; Ide, 1968]. The Rocchio algorithm for relevance feedback was first implemented in the SMART system around 1970 [Salton, 1971]. The SMART system is based on the Vector Space Model (VSM) which is one of the earliest information retrieval models [Salton *et al.*, 1975]. In the VSM models, the user query and the target documents are modeled as vectors, and the similarity between the user query and target document is calculated as the cosine similarity between

the vectors of the query and the document.

In the Rocchio algorithm, after the initial retrieval run a refined query is formed from three parts: the initial user query vector Q_0, the judged relevant documents vector D_r, the judged non-relevant document vector D_{nr}. Three parameters α, β, γ are used to combine these three vectors. The new query can be described shown in Equation 2.1:

$$\overrightarrow{Q} = \alpha * \overrightarrow{Q_0} + \beta * \frac{1}{|D_r|} * \sum_{\overrightarrow{D_j} \in D_r} \overrightarrow{D_j} - \gamma * \frac{1}{|D_{nr}|} * \sum_{\overrightarrow{D_k} \in D_{nr}} \overrightarrow{D_k} \tag{2.1}$$

In Equation 2.1, D_r is the number of judged relevant documents and D_j is a judged relevant document, and D_{nr} is the number of judged non-relevant documents and D_k is a judged non-relevant document. The Rocchio model can be explained as the newly expanded query is strengthening the information from the initial query and the judged relevant documents, while reducing the contribution of information from the judged non-relevant documents. The Rocchio algorithm is based on the VSM and all terms in judged relevant documents are considered as additions for the query. This is early work in the utilization of terms in the judged documents to improve the representation of the user query. Thus the modified query contains the new terms from the judged relevant documents and it has bigger chance to match the relevant documents which do not contain the original terms in the original query. The feedback source is usually the target document collection in these experiments. This process can be viewed as the process of utilizing the information from the target documents to resolve the sparse data problem in the query side in IR process.

More recently the probabilistic models have become the mainstream ranking IR models, and relevance feedback has also been interpreted within the framework of probabilistic models. As the Rocchio algorithm suggests, it is a natural choice to update the user query with terms found in the relevant documents from an initial retrieval run. In the framework of probabilistic models, these questions have formed the core consideration in the RF process:

1. How many terms in judged relevant documents (feedback terms) should be added to the user query?

2. How to weight these feedback terms to make a better query?

For the first question, a natural way to expand the user query is to add all the terms of relevant documents into the user query, but this is not a good strategy due to the curse of dimensionality [Rijsbergen, 1979]. Thus selecting some good terms for query expansion is a more reasonable way.

For the second question, a natural approach to weighting feedback terms from judged relevant documents is to rank all these terms using an existing term weighting method in the retrieval process. But analysis reveals that the term weighting methods of retrieval and relevance feedback should be different since their aims are different [Robertson, 1991]. As described in [Robertson, 1991], this is due to these two processes addressing different purposes: one is about term selection and a measure for this, and the other is about weighting the term in the revised query for ranking documents after feedback. Different questions should be resolved by different methods. Based on the argument, a term weighting score a_t of term t for relevance feedback has

been proposed in [Robertson, 1991] as:

$$a_t = w_t(p_t - q_t) \tag{2.2}$$

In Equation 2.2, p_t is the probability that a given relevant document is assigned the term t, q_t is the equivalent non-relevant probability. w_t is the term weight of t in the retrieval process. q_t is usually much smaller than p_t, and can be ignored. p_t can be estimated by r/R. r is the number of known relevant documents term t occurs in, R is the number of known relevant document for a request. R is equal for all the feedback terms, thus the Offer Weight (OW) of the feedback terms can be described as:

$$OW = r * w_t \tag{2.3}$$

Based on the proposed offer weight score, a comprehensive experimental investigation of using this method for relevance feedback was conducted in [Jones *et al.*, 2000]. In [Jones *et al.*, 2000], the results of series of experiments can be broadly summarized with the following conclusion: when query expansion from retrieved relevant documents is significantly better than no expansion is performed; massive expansion terms does not provide better results than modest managed expansion runs; blind relevance feedback can get comparable results compared to the explicit relevant feedback method; blind relevance feedback gets better results compared to the no expansion runs. This research provides us with a very solid methodology to utilize relevance feedback in resolving the sparse data problem in IR. This method has been shown to be effective, but remains unclear when using external resources as

the feedback source.

Much other research on relevance feedback has focused on answering the question of when RF can improve retrieval effectiveness. It has been demonstrated that it was important to expand the query in addition to re-weighting the terms, with most improvement coming for query expansion [Harman, 1992]. This work also suggests that queries can be expanded using only 20 selected terms, rather than all terms from the retrieved relevant documents, and if these terms are selected using a suitable method, significant performance improvements over no relevance feedback condition can be expected.

An experimental investigation to test a modified Rocchio relevance feedback approach on a TREC test collection [Buckley *et al.*, 1994a] showed that the recall-precision effectiveness varied linearly with the log of a number of terms added to the query from the relevant documents. Recall-precision also appeared to vary linearly with the log of the number of known relevant documents. The overall improvement in retrieval effectiveness (using MAP as the evaluation metric) arising from the application of relevance feedback are impressive, ranging from 19% to 38% depending on the number of known relevant documents using in the relevance feedback process.

Motivated by the hypothesis that query expansion terms should only be sought from the most relevant areas of a document, an investigation explored the use of document summaries in query expansion [Lam-Adesina & Jones, 2001]. In their experiments, using the Okapi BM25 model with the TREC-8 ad hoc retrieval task, query expansion using document summaries was shown to be considerably more effective than using full-document expansion. Later work showed that blind RF can be substituted for explicit evidence from hu-

man judgment in RF [White *et al.*, 2002]. The experimental results showed the automatic RF performed as well as the explicit system with human judgment in the process of RF.

Subsequent research in RF has asked interesting questions based on the previous research, with new work being motivated by the rapid growth in the application of machine leaning technologies in IR. These questions include:

- How can the parameter settings in RF be set automatically?
- How can the best feedback documents at the top of the ranked retrieval documents be chosen automatically?

In a typical process of query expansion, there are several free parameters that need to be set. One of the most important parameters is the coefficient between the original query terms and feedback query expansion terms. A query-regularized mixture model for PRF which automatically adjusts coefficient for feedback terms was introduced in [Tao & Zhai, 2006]. In this model, the feedback documents are assumed to be generated from a mixture model. Each feedback document was generated from a linear combination of a feedback document topic model and a background document topic model. In the process of linear combination, the parameters were different for each feedback document. The EM algorithm was used to estimate these parameters for the mixture model. Experimental results showed that this approach outperformed the standard language model for IR with feedback.

Three heuristics are used to adjust the coefficients for feedback information in [Lv & Zhai, 2009]: the more discriminative the query is, the more drift

tolerant it is likely to be, and thus, it is safe to utilize more feedback information; the less discriminative feedback documents could be trusted more; if the divergence between a query and its feedback documents is large, this can mean that the query does not represent relevant documents well, thus it may need a larger feedback coefficient. By adaptive RF, the experimental results on several TREC retrieval tasks can be improved by 1.12% to 4.12% by the criterion of MAP compared to the RF method with fixed coefficients [Lv & Zhai, 2009].

A problem in blind RF is that the assumed top feedback terms may not actually be relevant, while the default option in blind RF has generally been to assume that all top ranked feedback terms are relevant. Research on how to choose good feedback terms has been carried out [Cao *et al.*, 2008; Lv & Zhai, 2010]. Automatic method for selecting good feedback terms was introduced in [Cao *et al.*, 2008]. In this research, supervised learning was utilized to classify good feedback terms from the bad ones. Results showed significant improvement on three TREC collections. A method named the *positional relevance model* was introduced in [Lv & Zhai, 2010]. It demonstrated that not all feedback terms were relevant to the user query since the feedback documents may contain more than one topic and some topics were not relevant to the user topic. Their model was based on the assumption that the words closer to query words were more likely to be related to the query topic. The experimental results on two large data-sets show effective and robust results compared to the classical RF method.

The query-document mismatch problem which RF seeks to resolve is explained as an uncertainty problem in [Collins-Thompson & Callan, 2007].

Here the uncertainty means that the user's information need may be vague or incompletely specified by these queries. Even if the query is perfectly specified, the language in the collection documents is inherently complex and ambiguous and matching such language effectively is a formidable problem by itself. This work focused on the hypothesis that estimating the uncertainty in feedback was useful and led to better individual feedback models and more robust combined models. They proposed a method for estimating uncertainty associated with an individual feedback model in terms of a posterior distribution over language models. This work estimates a posterior distribution for the feedback model by resampling a given query's top-retrieved documents, using the posterior mean or mode as the enhanced feedback model. The idea behind this work is that the original feedback documents may not be good enough to represent the best feedback information since some feedback documents are not similar to the overall feedback information. Thus an estimated distribution from the original feedback documents can help to produce better feedback documents which are more relevant to the overall feedback information. These new feedback documents are then utilized for RF and more robust results have been gained in various TREC collections especially for the results of P@10 [Collins-Thompson & Callan, 2007].

In summary, RF provides a solid methodology to ameliorate the sparse data problem in many IR tasks. One limitation of past RF research is that these methods focus on utilizing the information from the target corpus to enrich the query information for resolving the sparse data problem. Less attention has been paid to resolving the problem from the document side or obtaining feedback information from external sources. Emerging new tasks

require new ways to utilize RF. In our research, we propose to utilize external resources in the process of RF from both query side and document side.

2.1.2 Sparse Data Problem in Personalized Search

Our research focuses on resolving the sparse data problem in IR. One way in which we propose to address this problem is by expanding existing work on RF to use external resources for query expansion and document expansion. The other aspect of IR data that we propose to explore is the use of user historical data, which is frequently used in personalized search. In this section, we review previous research on personalized search and motivate our research on this topic.

Personalization is an important trend in the modern IR systems. In this section, we introduce the topic of personalized search, the history of research investigating the personalization search and the state-of-the-art research in this area. With this background information, we then introduce the sparse data problem within personalized search.

For IR systems, personalization aims to provide search results adapted for a specific user such that the results are likely to be of interest to this user. The reason for the personalized search is that even when users enter the same search query, their search intent can be different. The reasons for this phenomenon can arise from various situations:

- The background of search users can be different, and their intent can be different when using the same search query. For example, a query such as "football", could mean looking for information about soccer or

information about American football.

- A query term can have many different meanings and different users can use the same query term, but intend a different meaning of the query term. For example, the query term "bank", could mean a river bank, a financial bank, or to bank an aeroplane.

- Search users have different levels of understanding of the search topics. Even when their search intent is the same, users can still have different interest or interpretation of relevance in the returned results.

For the query "machine translation", the top ten results returned by Google.com on a trial search run were: the Wikipedia page describing Machine Translation (MT)[1], the MT journal from Springer publisher[2], the free Translation service from WorldLingo[3], a statistical MT research website[4], and the MT engine from Foreignword[5], MT archive[6], MT system from SYSTRAN[7], MT page from Microsoft research[8], MT research from Google research[9] and MT system from Google.com[10]. While these are all related to the topic of machine translation, different users will often be interested in different results. A user who wants to find a free MT service is likely to be most interested in the machine translation systems available from WorldLingo, SYSTRAN, or Google; for an early

[1] https://en.wikipedia.org/wiki/machine_translation
[2] http://link.springer.com/journal/10590
[3] http://www.worldlingo.com/
[4] http://www.statmt.org/
[5] http://www.foreignword.com/
[6] http://www.mt-archive.info/
[7] http://www.systransoft.com/
[8] http://research.microsoft.com/en-us/projects/mt/
[9] http://research.google.com/pubs/MachineTranslation.html
[10] https://translate.google.com/

stage researcher who wants to gain a general introduction to machine translation, the Wikipedia page should be interesting; while for a senior researcher, maybe the links to Microsoft research, Google research, links from Springer or the archive of MT are better choices. The demands for different information when using the same query from different search users makes personalization a compelling challenge for IR research.

For online search services, personalization functions are already applied in our daily Internet usage. An example of personalized search can be found in Twitter.com. For the query "information retrieval", the twitter posts from the search function can be seen in Figure 2.1. The results are ranked by the time of twitter posts which contain the query terms. The newer posts are ranked higher than the older posts while these posts contain all of the query terms.

If we change the search function into the configuration of "from people you follow", the new search results are shown in Figure 2.2. The returned twitter posts are filtered only to include tweets from people which the search user is following. This is a very simple form of personalized search since the new results are only from people that has the search user has expressed interest in.

This search function could be more suitable for search users of twitter.com since people that the searcher is following could be providing the most useful information to the searcher. twitter.com also allows the search user to choose to personalize the search results or not. Another example of personalized search can be found on Twitter.com via its search from "near you" which recalls posts from twitter users near the searcher's location.

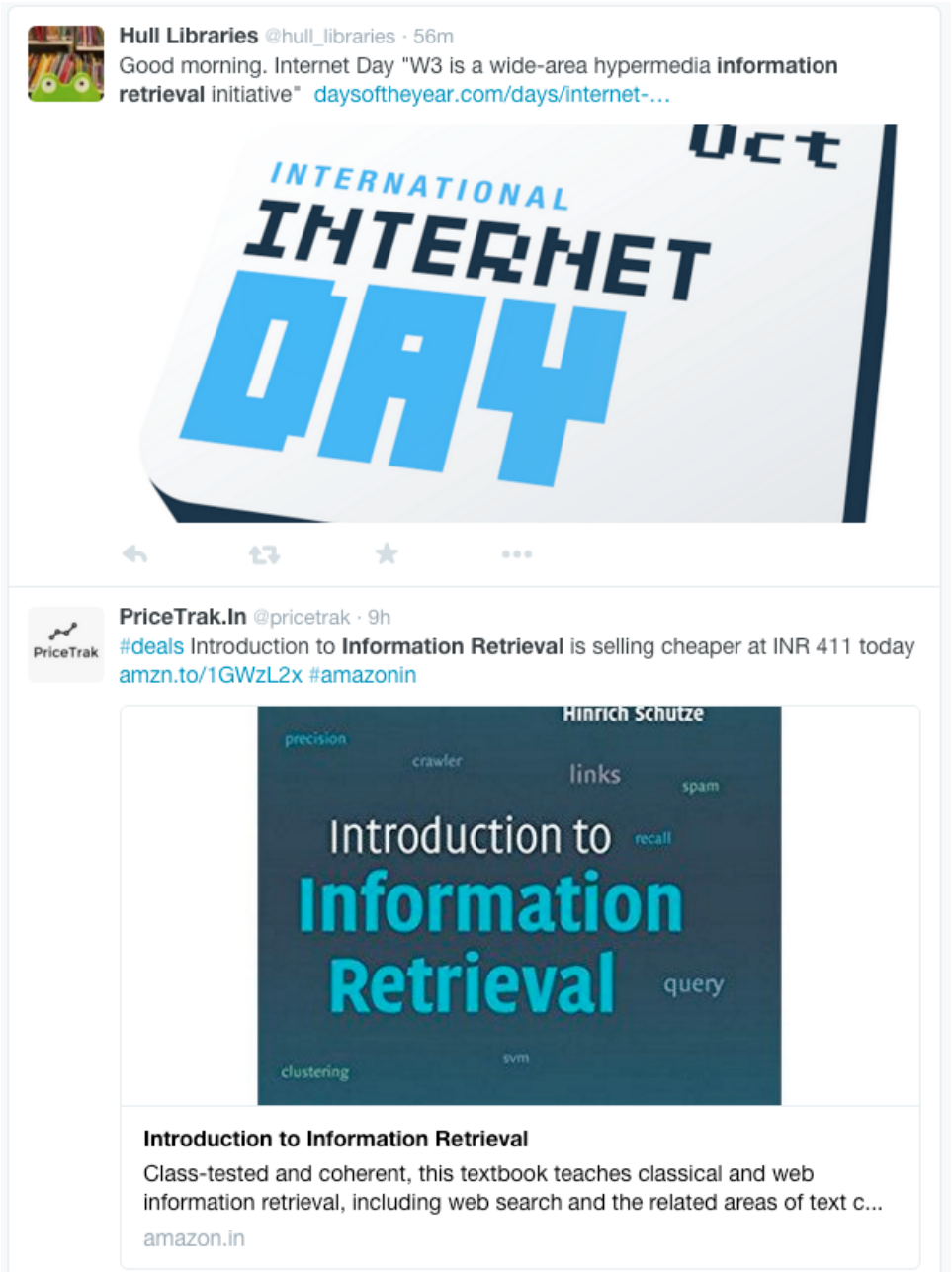

Figure 2.1: Search results form Twitter.com using the query “information retrieval".

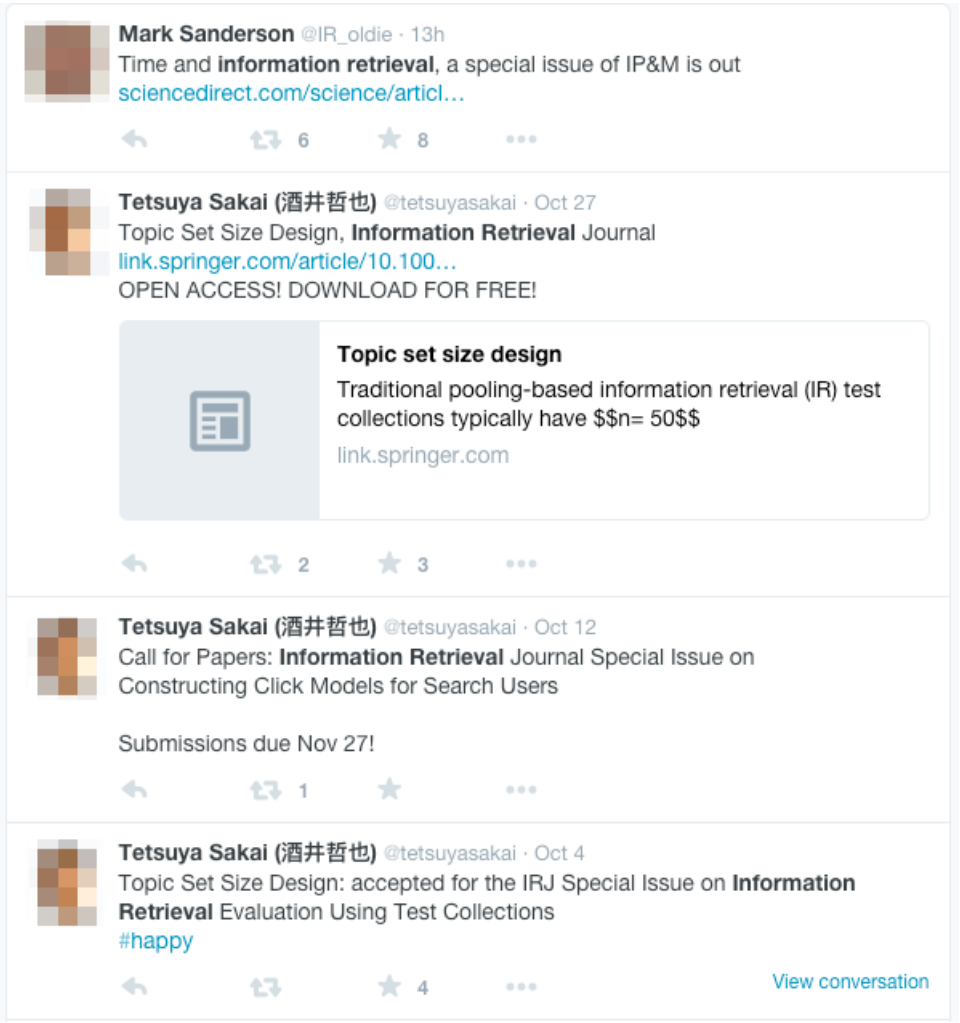

Figure 2.2: Personalized search results from Twitter.com using the query “information retrieval".

In the Twitter examples, the search system utilizes the user's social relationship and geographic information to personalize their search results. Many similar personalization services can be found in other online services such as Google and Facebook.

Classical IR research has focused on the similarity between the user's search query and potentially relevant items in the available document collection. There is no place within these classical IR models for the incorporation of personal preferences. The desire to incorporate elements of personal preference into the IR process introduces new challenges within IR research.

To personalize search results for a particular user, some quantification of the personal relevance between the user and the target documents is needed. To be able to calculate this, personal information relating to the current user is needed. A typical personalized search system contains a user modeling component which learns from the user's historical search activities and uses this information to personalize the search results for this particular user. The data produced by the user modeling component is called the *user model*. Each search user can be associated with a personal user model in a personalized search system. Some material in the user model may be captured in a registration form or a questionnaire which describes personal details of the user and their interest. More complex user models can be produced by incorporating details of the user's background information such as education level, the location of the user, their phone number, their familiarity with the topic of interest, etc.

User modeling in personalized search aims to record the user's search interests from their explicit and implicit data. The explicit data could con-

tain the user's registered profile for any web services; the implicit data could contain data associated with their interactions with their information system. The most obvious implicit data for an online search system is the user's click-through documents and the historical queries for a search system.

In the following analysis, we summarize the user modeling methods introduced in previous research of personalized search, and explain how these user models are utilized to contribute to ranking methods in IR.

An early exploration of personalized search can be seen in [Pretschner & Gauch, 1999]. This work is one of the earliest studies of the construction of user profiles in a search system. The study examines ways to model a user's search interests and shows how these models can be deployed for more effective IR and information filtering. In this work, user profiles are created by periodically processing the user's web cache to extract the URLs of Web pages that they visited. A spider collects the identified Web pages, and the pages are then classified into the appropriate concept(s) in a reference ontology using a vector-space classifier.

In this work, the reference ontology includes 4,400 nodes. Each node is associated with a set of documents to represent the content of the node, and these documents can be merged into a super-document. Thus the user historical data can be compared with the super-documents of these nodes. For a user's surfed page, a vector of this page can be compared with the vectors of the super-documents of the nodes using a standard vector space IR model [Salton, 1988]. The nodes with the top similarity scores are assumed to be related to the browsed page.

To create the user profile automatically, the surfed pages are collected pe-

riodically. For the top five categories (each node is a category), the weight of the category (weight) is combined from the time a user spent on the page (time) and the length of the page (length). The weight between the surfed page and the node can be adjusted using the Equation 2.4. In Equation 2.4, $\gamma(d, c_i)$ is the similarity score between the super-document of the node c_i and the surfed document d computed using the vector space model.

$$\Delta l(d, c_i) = log\frac{time}{log(length)} \cdot \gamma(d, c_i) \tag{2.4}$$

The equation adjusts the weight between the surfed page and the categories. The more time the user has spent on the page, the higher the assigned weight; the longer the page, the lower the assigned weight. This can be justified since if the user spends more time on a page, they are likely to be more interested in this page, but this weight should be modulated by the length of the page since a longer page needs more time to read. Thus the top categories identified from the historical surfed web pages are stored as the user's profile for later use.

One research question that this work sought to answer is whether automatically generated user profiles created using this proposed method really indicate the user's search interests. To answer this question, one step was to validate the convergence of the user interests from the surfed pages. If the categories created using the user's surfed pages do not converge into several important ones, it means that the method does not produce converged user search interests. The experimental results show that for the user's surfed pages, the categories related to a user will converge into a fixed number. This

indicates the user's search interests can be identified after accumulating a certain amount of surfed pages. A further investigation asked whether these automatically produced categories really represent the search user's search interests. This was explored using a questionnaire. The user's answers to which showed nearly half of them agreed that the categories were accurate.

User profiles constructed in this way are used in re-ranking of the search results. The re-ranking process uses the personal relevance between the identified categories of most interest to the user and the target web pages to re-rank the results from an initial search run. The best personalized results show an 8% improvement in MAP compared to the initial search results without personalized re-ranking in a web search task.

The basic methodology adopted in this work has been widely adopted in later personalized search research. Some key ideas for personalized search arising from this work can be summarized as:

- The search system does not ask the user to input search interests explicitly for building the user profiles.
- Surfed web pages from the user play a key role in modeling the user's search interests.
- Automatically constructed user profiles are used to re-rank the general search results for the individual user.

Following this early research work, various methods of personalized search have been proposed. Some of these work focuses on utilizing knowledge systems to build user profiles automatically, frequently used knowledge re-

sources are ODP[1] [Liu *et al.*, 2002] and data collected from Folksonomy systems [Xu *et al.*, 2008b]; some work concentrates on building personalized PageRank for target corpus [Haveliwala, 2002]; other research focuses on grouping users in personalized search [Teevan *et al.*, 2009]. We introduce these methods in this section, and analyse the advantages and drawbacks of these methods.

ODP web categories were used to build user profiles for users in [Liu *et al.*, 2002]. ODP categorizes all websites into a comprehensive human-edited directory. An example directory of open source software in ODP is shown in Figure 2.3. For the top level category "Computers:Open Source:Software", it includes 15 sub-categories. Under each sub-category, it can include several sub-categories. The sub-categories can still include their own sub-categories. Thus the ODP system consists of many levels of categories. In the bottom level of the hierarchical system, there are the links to the websites.

To build a user profile, one user's search record can be saved as shown in the Figure 2.4. Each user query is categorized into one or more categories in the ODP system. Usually the categories used to map the user data are the top one and two level categories, since ODP contains many categories levels. The surfed web pages using the query are also associated with the corresponding categories in the user profile. An example is shown in Figure 2.4, the query "apple" belongs to the category "Food & Cooking" and the surfed pages are "page1" and "page2".

A general user profile for all users was utilized to smooth the category results. This general user profile uses all the user's data to produce an overall

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

Computers: Open Source: Software

- Application Servers
- Databases
- Development Tools
- Editors
- Games
- Graphics
- Groupware
- GUI
- Internet
- Music and Audio
- Office Suites
- Operating Systems
- Programming Languages
- Project Management
- Security Tools

Figure 2.3: An Example of an ODP Category.

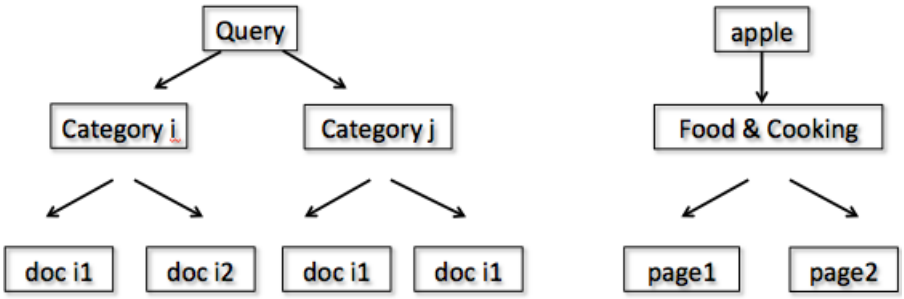

Figure 2.4: Model and Example of a Search Record.

profile. It can also be used to get a general relevance between the target web pages and the general user profile.

In this work, for each user query, the similarity between the user query and the general profile along with the similarity between the user query and user profile was computed. The general profile was extracted from the ODP web categories with each category associated with terms. This is similar to the work of [Pretschner & Gauch, 1999]. The similarity between the user query and the user profile can be also generated in this way. Also, several text categorization algorithms have been tested to classify the user query into categories. The experimental results showed that combining a general profile and a user profile can produce better personalized search results than using only a user profile. The results can be explained as shown that the general user profiles are used as a supplement for the individual user's profile for user modeling. It also reveals that the user profile may be not sufficient to record the user's search interests.

An attractive feature of this ODP method is that ODP is an existing web category system which can be utilized in the user modeling process for personalized search. One problem with the ODP system is that ODP has been developed by different online users, with the result that the categories are not well organized and some categories can be very broad while others are very narrow. Also the topics of different categories can overlap with each other, which is not good for identifying distinct user search interests.

The alternative to using the ODP web directory is to utilize a folksonomy for user modeling. A folksonomy is a system of classifications derived by collaboratively creating and managing tags to annotate and categorize content.

It is also known as collaborative tagging, social classification, social indexing, or social tagging. The use of folksonomy in personalized web search is explored in [Xu *et al.*, 2008b]. Several useful features can be brought into personalization by using a folksonomy: social annotations can provide category names, social annotations can be used as keywords, and it can introduce a collaborative link structure. In this work, the final query/document relevance $r(u, q, p)$ (u: user, q: query, p: web page) was a linear combination of the term relevance between query and documents, and the topic relevance between document and user as shown in Equation 2.5. $r_{term}(q, p)$ is the term relevance between the query and the web page, $r_{topic}(u, p)$ is the topic relevance between the user and page, and γ is a coefficient to adjust the weight of the linear combination.

$$r(u, q, p) = \gamma \cdot r_{term}(q, p) + (1 - \gamma) \cdot r_{topic}(u, p) \tag{2.5}$$

The topic relevance $r_{topic}(u, p)$ is computed using the vector space model as shown in Equation 2.6. $\vec{p_{ti}}$ and $\vec{u_{ti}}$ is the topic vector of the web page and the user. The dimension of the vectors is the number of tags in the folksonomy system and then each dimension in the vector represents a tag. For $\vec{p_{ti}}$, the weight in each dimension is the number of times that the web page contains this tag. For $\vec{u_{ti}}$, the weight in each dimension is the number of times that the user profile contains this tag. Experimental results show improvement in the search quality on a web search task [Xu *et al.*, 2008b].

$$sim_{topic}(p_i, u_j) = \frac{\vec{p_{ti}} \cdot \vec{u_{ti}}}{|\vec{p_{ti}}| \times |\vec{u_{ti}}|} \tag{2.6}$$

Similar work can be seen in [Braun *et al.*, 2008]. In this work, a system is built to record the user's click-through data on Web 2.0 websites such as *Youtube*, *Flick*, and *del.icio.us* to collect tags related to the user. Tags were used to rank the future search results from the same user. This research showed that a folksonomy is a good resource to build user profiles in personalized search when available. One problem in using a folksonomy to model the user's search interests is that the folksonomy only exists on some web data and it is not easy to collect relevant folksonomy data for a general personalized search task.

PageRank is an important algorithm in a web search engine. It provides a score for each web page based on how many outside links point to this page, which means how important for this web page on the overall Internet [Page *et al.*, 1999]. The original PageRank algorithm provides unified scores for websites, but different websites might mean different weights for different users. Thus it is interesting to develop a personalized version of PageRank scores for a personalized web search task. Going beyond the original PageRank, a topic sensitive PageRank is introduced in [Haveliwala, 2002]. A personalized web search method based on the personalized topic sensitive PageRank method was proposed in [Qiu & Cho, 2006]. Each user is associated with a topic distribution. Web search result ranking is based on the estimated user profile and the topic sensitive PageRank score. Their results show significant improvement compared to topic sensitive PageRank scores on data for 10 subjects collected from Google search history in a computer science department.

Past research has focused on small collections of data for personalized search research. With the fast progress of web search engines, the personal-

ized search research for large scale data has become the focus of state-of-the-art research. In large scale web search tasks, user logs become a key resource to build user models. User logs are the focus of much research on personalized web search [Sugiyama *et al.*, 2004; Speretta & Gauch, 2005; Wen *et al.*, 2009; Zhu *et al.*, 2010].

In [Speretta & Gauch, 2005], the study was conducted through three phases:

- Collecting information from users. All searches, for which at least one of the results was clicked were logged for each user.
- Creation of user profiles. Two different sources of information were identified for this purpose: all queries submitted for which at least one of the results was visited, and all visited snippets of web pages. Two profiles were created: one created from queries and one using the snippets.
- Evaluation: the created profiles were used to calculate a new rank of results browsed by users for a query. The new rank was used to compared with the Google's original ranked output.

The evaluation was based on a personalized web search task and the data was collected from six users with 45 queries. The average rank of the user's click documents by this method was improved by 37% compared with Google's original rank. This work can be viewed as an early exploration of personalized search methods for web search.

Methods are applied to perform personalized query expansion from the individual user's logs in [Cui *et al.*, 2003]. The assumption is that the historical data from users contains relevant information with regard to the current

user query, thus this feedback can create a better user query to find better results on the target corpus. A client-side web search agent to perform implicit feedback and query expansion was described by [Shen *et al.*, 2005]. In this work, query expansion was based on previous queries and immediate result re-ranking based on click-through information. The main focus of this research was how to exploit the immediate and short-term search context to improve search. They presented a decision-theoretic framework for optimizing interactive information retrieval based on eager user model updating. A method for improving web queries by expanding them with terms collected from each user's personal information repository was proposed in [Chirita *et al.*, 2007], this implicitly personalized the search output. Their results show that some of these approaches perform very well, especially on ambiguous queries, producing a very strong increase in the quality of the output rankings of relevant documents.

While the basic methodology of personalized search has been proven to be effective, more problems have been addressed in the following research. One important problem is that the lack of user data may harm the user modeling and lead to the failure of personalized search. Later research tried to enrich the user data using data from other users. A typical method used the grouping of similar users [Teevan *et al.*, 2009], thus a group user model was used to model an individual user's search interests. Another way to enrich the user data is to get information from the user's friends in Internet social networks [Carmel *et al.*, 2009]. This problem is directly related with our thesis topic of the sparse data problem in IR.

The sparse data problem in user data is a less researched problem in per-

sonalized search tasks. Work on grouping user data for personalized search may be the earliest exploration to examine this problem [Teevan *et al.*, 2009]. Previous research on personalized search focused on methods for creating user profiles and how to use them to adjust document ranking. Less attention has been paid to the problem of whether sufficient user data is available to adopt this strategy. In a real search environment, collecting enough user data to build user profiles can be a challenge, since the user's historical data may not be sufficient to cover the user's search interests.

Grouping similar users' data is one method to enrich an individual user's data. This method is called *grouplization* in this work. Experimental results show that people show explicit similarity share similar search interests and intent, and it can be beneficial to group their historical data to get better personalized results. In this work, people from the same age range, same sex or same occupation are called an explicit group. Some common queries from the explicit group are more effective when using the grouplizaton method. These groups indicate the potential of sharing the same search interests between individuals. But the information for grouping users such as age, sex or occupation is usually hard to collect for general online search users. Thus a more common and easy to implement method to enrich the user historical data in personalized search is needed.

Social search has gained considerable attention in recent years. It can be viewed as a solution to the sparse data problem of user data in personalized search. A personalized social search method based on the user's social relations is described in [Carmel *et al.*, 2009]. In this work, search results are re-ranked according to their relations with individuals in the user's social

network. Three types of strategies were studied: a familiarity-based network of people related to the user through explicit familiarity connection, where familiarity means two individuals know each other; a similarity-based network of people similar to the user as reflected by their social activity, where similarity means two individuals having common activities; and an overall network that provides both relationship types. All these social based methods outperform a topic-based strategy which builds user profiles based on terms in their experiments.

Previous research into personalized search has demonstrated the effectiveness of personalization for IR and utilizes several knowledge systems to construct the user profiles [Pretschner & Gauch, 1999; Xu *et al.*, 2008b; Teevan *et al.*, 2009]. The lack of user data problem has also been investigated. The current solution is not general enough to be used in the general personalized search task. In our research, we propose to utilize external resources to update the user historical data which can be utilized in any common personalized search tasks. In the next subsection, we introduce and review previous methodologies to utilize external resources in IR tasks.

2.2 Utilizing External Resources in IR

A typical IR evaluation task contains a target corpus for retrieval, a set of user queries and the human relevance judgments for these user queries. We refer to a corpus other than the target corpus included in the retrieval process as an *external* corpus. In this section, we survey important work in the utilization of external resources in IR. The purpose of this section is to introduce methods

in previous research on the utilization of external resources in IR.

The work described in [Diaz & Metzler, 2006] introduces external corpus into the relevance model [Lavrenko & Croft, 2001]. Relevance models provide a framework for estimating a probability distribution, $\widehat{\theta}_Q$, over possible query terms, w, given a short query, Q. This work takes a Bayesian approach, as shown in Equation 2.10.

The query and the target documents are used to estimate a probability distribution $\widehat{\theta}_Q$ as Equation 2.7.

$$P(w|\widehat{\theta}_Q) \approx P(w|Q) \approx \frac{P(w,Q)}{P(Q)} \tag{2.7}$$

Since $P(Q)$ is the same for all w, this can be reduced to the form shown in Equation 2.8.

$$P(w|\widehat{\theta}_Q) \propto P(w,Q) \tag{2.8}$$

All the target documents with document model θ_D can then be produced as shown in Equation 2.9.

$$P(w,Q) = \int_{\theta_D} P(w|\theta_D)P(Q|\theta_D)P(\theta_D) \tag{2.9}$$

Equation 2.8 and 2.9 can be combined to produce Equation 2.10.

$$P(w|\widehat{\theta}_Q) \propto \int_{\theta_D} P(w|\theta_D)P(Q|\theta_D)P(\theta_D) \tag{2.10}$$

where θ_D is a document language model and $P(Q|\theta_D)$ is the query likelihood. The relevance model combines two models by linear interpolation as shown

in Equation 2.11.

$$P(w|\theta_Q) = \lambda P(w|\widetilde{\theta}_Q) + (1-\lambda)P(w|\widehat{\theta}_Q) \tag{2.11}$$

where $\widetilde{\theta}_Q$ is the maximum likelihood query estimate.

To build a query model that combines evidence from one or more collections, a mixture of relevance models is formed. This results in modifying Equation 2.11 to produce Equation 2.12.

$$P(w|\theta_Q) = \sum_{c \in C} P(c)P(w|\theta_Q, c) \tag{2.12}$$

where C is the set of collections and $P(w|\theta_Q)$ is the relevance model computed using collection c. Thus, two collections including the target corpus and an external corpus can be used to estimate the query model. When compared to traditional PRF techniques, external expansion is more stable across topics and up to 10% more effective in terms of MAP. This work utilizes the external resources as the source to estimate the language models of the query. This process is a very similar process to query expansion methods in RF research.

[He *et al.*, 2012] proposes a framework that combines both implicitly and explicitly represented sub-topics, and allows a flexible combination of multiple external resources in a transparent and unified manner. Specifically, a random walk based approach is used to estimate the similarities of the explicit subtopics mined from a number of heterogeneous resources: click logs, anchor text, and web n-grams. These similarities are then used to regularize the latent topics extracted from the top-ranked documents, the internal subtopics. Empirical results show that regularizing the latent topics extracted

from the right resource leads to improved diversification results, indicating that the proposed regularization with external resources forms better topic models. Click logs and anchor text are shown to be more effective resources than web n-grams under current experimental settings. Combining resources does not always lead to better results, but is found to achieve robust performance in all cases. This robustness is important for two reasons: it cannot be predicted which resources will be most effective for a given query, and it is not yet known how to reliably determine the optimal model parameters for building implicit topic models. By this method, the external resources are utilized to build topic models in the initial retrieved results.

Utilizing external resources for query expansion continues to be an attractive topic in IR research. [Bendersky *et al.*, 2012] presents a unified framework that automatically optimizes the combination of information sources used for effective query formulation. The proposed framework produces fully weighted and expanded queries that are both more effective and more compact than those produced by the current state-of-the-art query expansion and weighting methods. Empirical evaluation is reported for both newswire and web corpora. In all cases, the combination of multiple information sources for query formulation of multiple information sources for query formulation is found to be more effective than using any single source. The proposed query formulations are especially advantageous for large scale web corpora, where they also reduce the number of terms required for effective query expansion and improve the diversity of the retrieved results.

[Bouchoucha *et al.*, 2013] utilized the ConceptNet[1] as an external resource

[1]http://conceptnet.io/

to conduct query expansion. Expansion terms were selected from ConceptNet so as to cover as diverse aspects as possible. Diversifying query expansion has a very similar goal to result diversification. The expansion terms need to be diverse, or non-redundant. An approach similar to MMR (Maximal Marginal Relevance) can naturally be used. MMR is a method of SRD (Search Result Diversification) which tries to select documents that are dissimilar from the ones already selected. The use of MMR for diversity is shown in Equation 2.13.

$$MMR(D_i) = \lambda \cdot rel(D_i, Q) - (1 - \lambda) \cdot \max_{D_j \in S} sim(D_i, D_j) \tag{2.13}$$

where D_i is a candidate document from a collection, and S is the set of documents already selected. The parameter λ controls the trade-off between relevance and novelty. *rel* and *sim* determine respectively the relevance score of the candidate document to the query and its similarity to a selected document. In each step, MMR selects the document with the highest MMR score. Experiments were conducted on the ClueWeb09 dataset, using the test queries from the TREC 2009, 2010 and 2011 web tracks. The MAP values for these query sets improve from 0.160 to 0.206 compared to a baseline for SRD [Vargas *et al.*, 2013].

To summarize previous research into utilizing external resources in IR, it has focused on using external resources to improve the retrieval effectiveness by query expansion. The difference between these studies has been in the retrieval model used such as the language model retrieval method [Diaz & Metzler, 2006], or topic modeling [He *et al.*, 2012]. Also search result diver-

sification is a motivating factor to include external resources to bring more relevant topics to the query side using information from the external resources. The key reason that external resources work for result diversification is that the external resources tend to expand the queries from different aspects of its meaning than expanding only on the target document collection. This work demonstrates the effectiveness of utilizing external resources in IR tasks. For our research topic - the sparse data problem in IR, utilizing external resources may play an important role, since resolving the sparse data problem by adding more data from external resources is an attractive possibility. These existing successful applications motivate our research to utilize external resources to address the sparse data problem in IR tasks, since external resources have been shown to be effective in bringing useful information into the IR process.

2.3 Stepping-off to Our Research

From the findings of the previous work described in this chapter, several ongoing problems can be identified:

1. Previous research ignores retrieval tasks where there is a sparse data problem.

2. Most work assumes that the target corpus contains sufficient information for RF which is not true for some retrieval tasks.

3. Most work is focused on the query side and little research investigates how to solve sparse data problem from the document side.

4. In personalized search research, the sparse data problem in user historical data has been less noted.

Based on these points, this motivates us to conduct a research study into the utilization of external resources in IR to resolve the sparse data problem, including QE, Document Expansion (DE) and enriching the user data in personalized search task. RF from external resources for the sparse data problem is a less studied problem. In Chapter 3 and Chapter 4, we report a more detailed study of QE and DE using external resources.

Although much work has been done on personalized search, there are still unsolved research problems in this area. Analysis of existing research reveals that personalized search has a sparse data problem due to the ambiguous and short queries and incomplete user historical data. Based on this analysis, we note several problems with personalized search:

- User logs may be insufficient to enrich the user query. Thus, the question arises, can we utilize external resources to enrich the user logs to overcome this problem?
- The second question is, can we utilize external resources in personalized search effectively?

Based on these research problems, we propose to examine the utilization of external resources to enrich user historical data in personalized search. This research is described in Chapter 5 and Chapter 6.

2.4 Summary

In this chapter, we introduced the problem of sparse data in IR. We surveyed existing research in RF and personalized search, and their efforts in resolving the sparse data problem. We also introduced previous research into utilizing external resources in IR. Based on state-of-the-art of these topics, we introduce research studies seeking to utilize external resources to resolve the sparse data problem in IR. In the following chapters, we begin to introduce our research work on these topics.

Chapter 3

Exploring External Resources in Query Expansion

Following the introduction to our research in Chapter 1, and the survey of related existing work in Chapter 2, we begin our investigation into the utilization of external resources in IR. We first investigate the potential for the use of external resources in Query Expansion (QE) which we refer to as External Query Expansion (EQE). In EQE, an external resource is used to augment the user's query as the source of feedback information. Our experiments on EQE are conducted on a text-based image retrieval task. We select this task due to the sparse data problem which arises because of the short document length in this task, where the textual description of these documents only consists of a small number of meta-data entities. In this research, we compare EQE to the standard Query Expansion (QE) from the target collection and also introduce a novel Definition-document based Relevance Feedback (DRF) method which seeks to fully utilize the external resources.

In this part of our research, we aim to find whether utilizing external resources in query expansion performs well for the IR tasks with sparse data problem. We conduct our research by answering the following research questions:

- How does the classical query expansion perform for retrieval tasks with sparse information?
- Which is better to compare query expansion from the target collection with query expansion from external collection?
- Is the classical query expansion algorithm the best for query expansion using external resources?

The structure of the chapter is as follows. Section 3.1 introduces the background and related work on utilizing external resources in QE. Section 3.2 describes our method to apply the RF method by utilizing an external resource. Section 3.3 proposes our new DRF method which seeks to improve the utilization of external resources in RF for IR. Section 3.5 and 3.6 discusses the work described in this chpater and summarizes our findings for this chapter.

3.1 Background and Related Work

As introduced in Chapter 2, previous research on RF has focused on the utilization of information from the target corpus. A key assumption of this approach is that the target corpus contains enough information to enrich the

user query to resolve the sparse data problem in the query side. For some emerging retrieval tasks, this assumption may not be true since for many IR tasks the target documents are short, with the result that they do not fully describe the topic of themselves. To resolve this problem, a potential solution is to get the information from an alternative external resources for the RF process. We refer to this approach tof RF as EQE.

A straightforward approach to EQE is to conduct an initial retrieval run on the external corpus, and then to select expansion terms from feedback documents selected from this initial run. The new query expanded is then applied to the target corpus to perform the second retrieval step. In this section, we review existing work on QE using external resources.

Early work on QE using external resources can be found in the TREC newswire retrieval tasks [Walker *et al.*, 1998; Robertson *et al.*, 1999; Robertson & Walker, 2000; Robertson *et al.*, 2000]. TREC newswire documents are usually long and comprehensive meaning that there is no sparse data problem to harm retrieval effectiveness. However, research at TREC showed that external QE can work well in tasks using TREC newswire collection. QE from a larger collection than target collection has been shown to be effective in TREC tasks [Robertson *et al.*, 2000]. Blind expansion using the TREC 1-5 showed a gain of 8% in MAP compared to a baseline without QE for the TREC 6 task. In this experiment, the TREC 6 data-set was the target collection and the TREC 1-5 datasets were used as a large external collection for RF. The TREC 6 data consists of newswire documents where the document length is usually long (average length of documents ranges from several hundred terms to several thousands terms) and the description of the content is thus quite detailed.

The relevance feedback from external collections proves to be effective in this task. EQE improves the MAP by 8% over a method without EQE for the TREC8 ad hoc task [Kwok, 2000]. The results of further experiments showed that EQE produces better results for shorter queries than for long queries [Kwok, 2000]. EQE was investigated in later work which modelled the QE process as a random walk process [Collins-Thompson & Callan, 2005]. In this work, the combination of an external collection and the target collection as evidence for QE achieved higher performance than using the target collection only on several TREC newswire tasks.

Since the TREC newswire collections do not suffer from the sparse data problem, EQE may not always work better than QE from the target corpus for all queries. Later research proposed a method to select use of the external collection or the target collection itself as the source of QE information for different queries [He & Ounis, 2007]. In this work, retrieval performance is estimated by a query performance predictor for each query. The external resource or target collection is chosen for QE based on the estimated performance. This adaptive QE method achieved the best result compared with QE from only one collection for two standard TREC web search tasks. Further research classified TREC topics into three categories based on an external corpus: i) entity queries; ii) ambiguous queries and iii) broader queries [Xu *et al.*, 2009]. Experimental results showed that use of an external resource helped to improve retrieval effectiveness for all three query types.

Current research lacks deep analysis of the reason why EQE works on some retrieval tasks and not on others. This gives us the opportunity for a detailed examination for tasks with a sparse data problem, and investigation

of the potential for QE using external resources.

3.2 Query Expansion from External Resources

In this section, we investigate QE from external resources on a text-based image retrieval task. The Dbpedia collection [1] is used as our external resource since it contains a broad coverage of topics and less noise information than full Wikipedia articles [2]. Our initial QE method employs the standard Okapi feedback method [Robertson, 1991; Robertson *et al.*, 1994] as described in Chapter 2.

Our experiments were conducted on the collection from the ImageCLEF WikipediaMM 2008 task [Theodora Tsikrika, 2008]. We selected this task to conduct the research on utilizing external resources for IR on the query side, since it is a typical task where the queries are short and the target documents are short as well. Some example queries can be seen in table 3.1. The example queries show that the queries are all very short. The length of the 75 queries in this collection ranges from 2 to 7 words. Short queries are usually unable to describe the search intent of the user in full details. This is a typical sparse data problem in IR tasks.

Table 3.1: Example Queries of WikipediaMM 2008.

1	blue flower	2	sea sunset
3	ferrari red	4	white cat
5	silver race car	6	potato chips
7	spider web	8	beach volleyball
9	surfing	10	portrait of Jintao Hu

[1] http://wiki.dbpedia.org/

[2] In the research of this chapter, we refer to Dbpedia when we use Wikipedia documents

The target collection includes 151,520 images with 75 queries and relevance data [Westerveld & van Zwol, 2007]. Each image is associated with a meta-data file as shown in Figure 3.1. As stated above, we also use the Dbpedia as the external resource for QE. Each document in the Dbpedia is the first paragraph of the corresponding Wikipedia document [1]. The English Dbpedia includes 2,452,726 documents (a version downloaded in Jan, 2009). We use this Dbpedia as the external resource for QE since:

1. It includes only the definition sentences of Wikipedia terms and contains less noise than full articles.

2. It covers a very broad range of general topics and should contain documents relevant to most general user queries.

3. The coverage of Wikipedia is expanding over time and thus making our method suitable to more user queries as it develops.

For each metadata file in the WikipediaMM collection, we remove the tags and leave the remaining text as the target document for retrieval. For the metadata file in Figure 3.1, the text "Australian_20note_back.jpg Australian $20 note back money" is used as the text for text based information retrieval. Several pre-processing steps were carried out for the experimental queries, the target documents (WikipediaMM corpus), and the external documents (Dbpedia):

- Punctuation removal

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


```
<?xml version="1.0"?>
<article>
    <name id="4">Australian_20note_back.jpg</name>
    <image xmlns:xlink="http://www.w3.org/1999/xlink"
           xlink:type="simple" xlink:actuate="onLoad"
           xlink:show="embed"
           xlink:href="../pictures/Australian_20note_back.jpg"
           id="4" part="images-50000">
           Australian_20note_back.jpg
    </image>
    <text>
        <wikitemplate parameters="1">
            <wikiparameter number="0" last="1">
            <value>Australian $20 note, back, money</value>
            </wikiparameter>
        </wikitemplate>
    </text>
</article>
```

Figure 3.1: Image with metadata example.

- Stopword removal (stop-word list from the SMART retrieval system [Salton, 1971])
- Stemming using the Porter stemming algorithm (implementation in the Lemur toolkit [1]) [van Rijsbergen *et al.*, 1980]

After pre-processing, the average length of data sources are as shown in Table 3.2. We use the Okapi BM25 model in the Lemur toolkit for the retrieval tasks.

Table 3.2: Data Average Length.

Data	Average Length (in terms)
Queries	2.8
Target Documents	24.4
English Wikipedia Abstract Documents	99.7

3.2.1 Results of Okapi Feedback Algorithm

In this section, we examine the effectiveness of QE using the method incorporating external resources introduced in previous section. We carried out the following experimental runs:

- Okapi retrieval model only (Run: Baseline)
- QE on the target corpus and the Okapi retrieval model on the target corpus (Run: QE)
- QE on the external resource and the Okapi retrieval model on the target corpus (Run: QEE)

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

- QE on the external resource, then QE on the target corpus, and then the Okapi retrieval model on target corpus (Run: QEE+QE)
- QE on the target corpus and then QE on the external resource, and then the Okapi retrieval model on the target corpus (Run: QE+QEE)

In our experiments, queries were expanded from the target corpus or external resource using Okapi feedback algorithm. The expansion terms can be the same as the original query terms. The following parameters were adjusted manually:

- the assumed number of relevant documents from the initial retrieval run (R)
- the number of feedback terms for expansion (k)
- the coefficient to adjust the weights of the original query terms and expansion feedback terms (coefficient)

To compare these different runs, we examine the results for the alternative methods using different parameters.

3.2.2 Comparing QE and QEE

To compare the QE and QEE methods, we tested them under different parameter settings. First we examined standard QE under different parameters. We show results for QE using different numbers of feedback documents (R) with different numbers of expansion terms (k). The results are shown in Figure 3.2. We make the following observations:

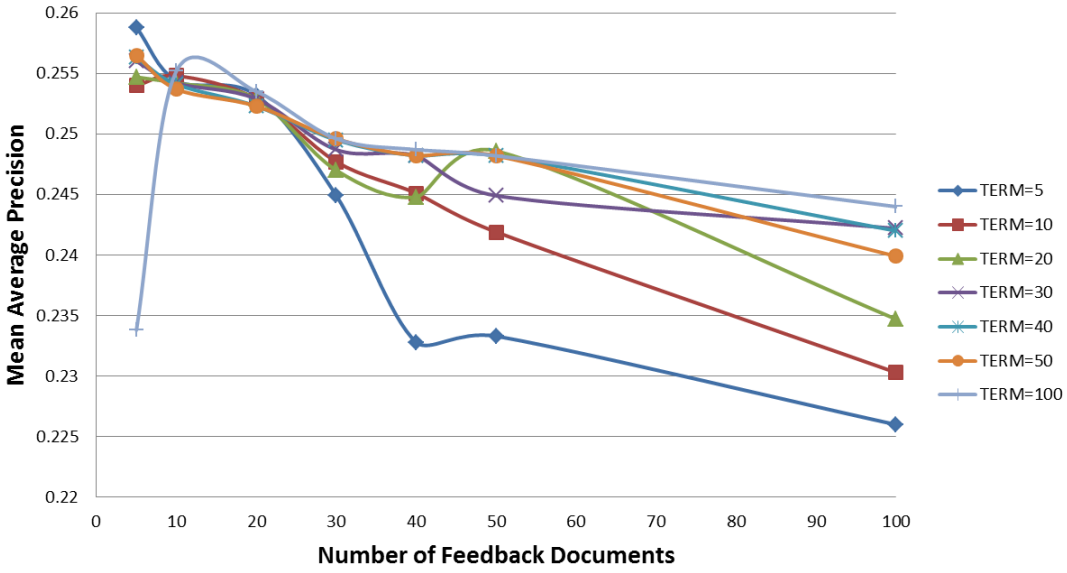

Figure 3.2: Results for QE for the WikipediaMM test collection using a fixed number of feedback terms.

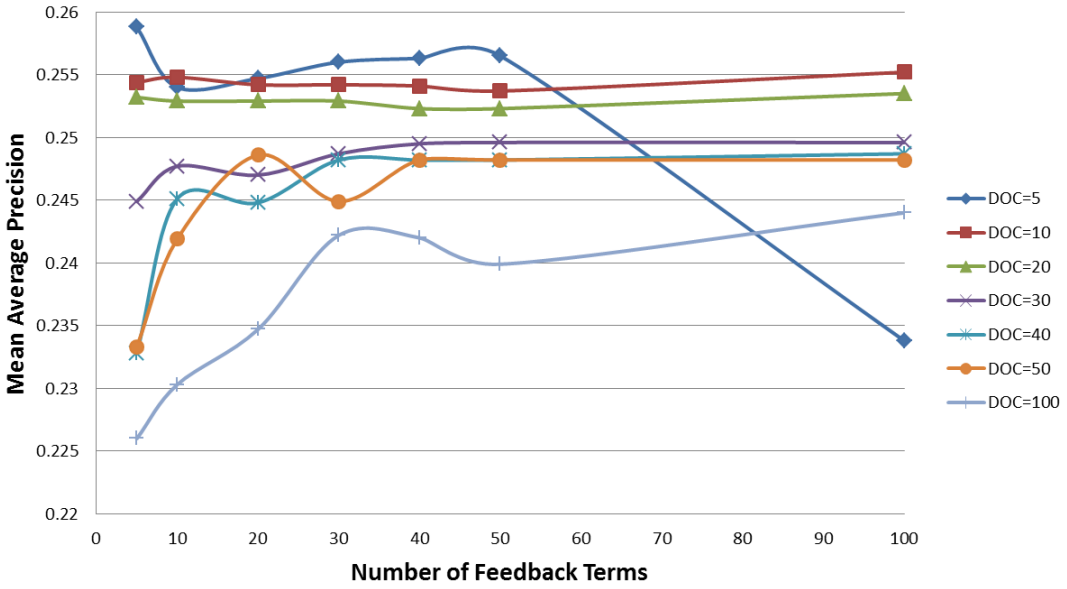

Figure 3.3: Results for QE for WikipediaMM collection using a fixed number of feedback documents.

1. The best result is obtained when setting $R = 5$ and $k = 5$.

2. When fixing the number of expansion terms, the MAP value decreases when more feedback documents are added.

These findings suggest that bringing more feedback documents into the RF process does not help to improve retrieval effectiveness in this case. Our results indicate when using QE on short length documents retrieval tasks, more feedback documents do not add more useful information into the RF process. Thus it does not help selection of good expansion terms for QE. These findings are different from the previous conclusions of experiments on the TREC collections [Buckley *et al.*, 1994a; Robertson *et al.*, 1994]. In the earlier TREC experiments [Buckley *et al.*, 1994a], the more feedback terms added from relevant documents, the better the recall-precision, up to a steady-state value. For the Okapi feedback method [Robertson *et al.*, 1994], the previous results show that adding a reasonable large number of feedback terms will benefit to the results (such as term number as 60).

We show results using various numbers of expansion terms when the number of feedback documents is fixed in Figure 3.3. From Figure 3.3, we can see that the results indicate that more expansion terms do not change the retrieval effectiveness very much. When we use only 5 good feedback documents, adding too many expansion terms also hurts the final results ($k > 50$). Using 100 feedback documents and 5 expansion terms gives the worst result for QE method in our experiments. Also when using many nonrelevant documents for QE ($R = 100$), more expansion terms could help to improve the final retrieval effectiveness. This evidence indicates that QE only needs

good expansion terms in the good feedback documents for this short-length documents retrieval task. The results indicate that the target corpus provides limited useful feedback documents and expansion terms for short-length documents retrieval task.

We hypethesis that external resources may help to relieve data sparse problem in IR tasks better than the search target collection. We test the results of the QEE method under various parameter settings. First we examine the effect of different numbers of feedback documents when using a fixed number of expansion terms. The results of this experiment are shown in Figure 3.4. Comparing Figure 3.4 with Figure 3.2, the observable difference is that the MAP scores in Figure 3.4 do not decrease with the addition of more feedback documents from the external resources when using a fixed number of expansion terms, as was the case in Figure 3.2. The difference of QE and QEE is illusrated in Figure 3.5 using the fixed number of feedback terms set at 5. This observation can be explained as the external resource containing more useful information relevant to the query than the target corpus. Thus adding more feedback documents does not add significantly more noisy documents into the process of relevance feedback. However, the best result of the QEE method does not outperform the best result of the QE method (the best results of different methods are shown in Talbe 3.3).

We also show results for different numbers of expansion terms when using a fixed number of feedback documents in Figure 3.6. The results show that adding too many expansion terms from external resources harms the retrieval. This indicates that the expansion terms should be limited to a reasonable number when using external resources.

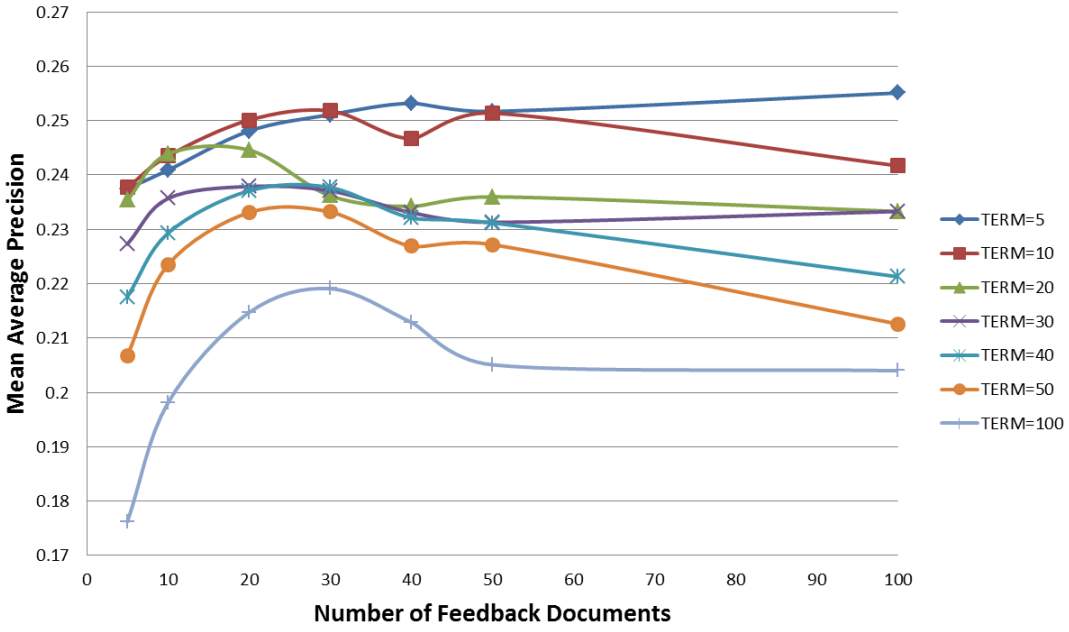

Figure 3.4: Results for QEE for the WikipediaMM test collection using fixed number of feedback terms.

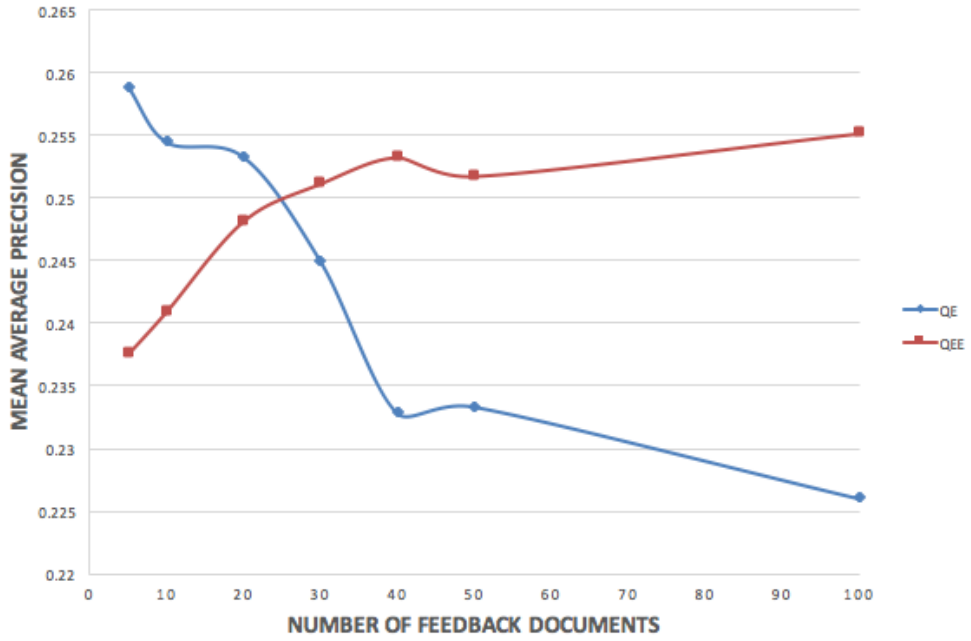

Figure 3.5: Comparision of QE and QEE using fixed number of feedback terms 5.

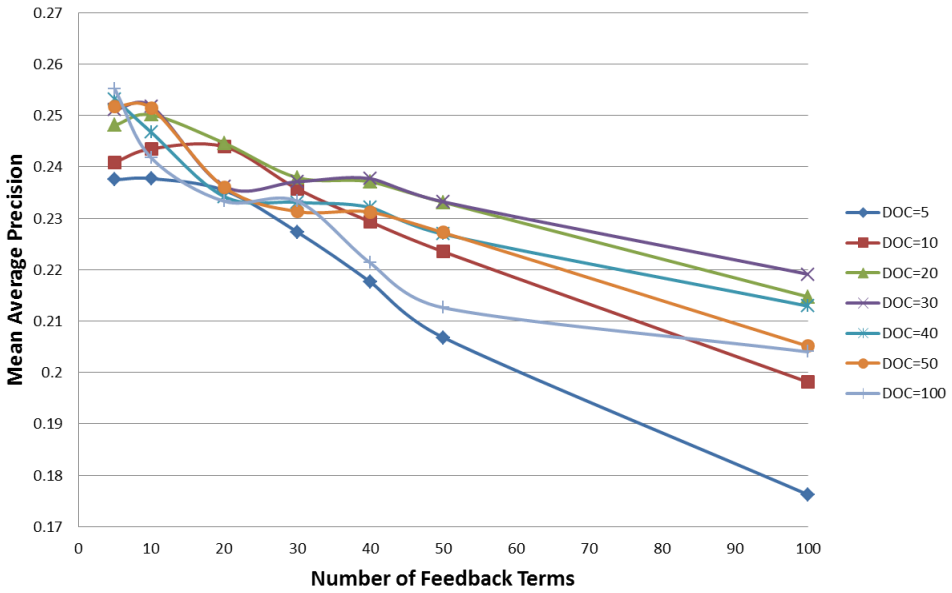

Figure 3.6: Results for QEE for WikipediaMM collection using fixed number of feedback document.

Although QEE does not produce better results than QE, it is interesting to test the impact of using queries expanded from external resources to expand from the target corpus again for retrieval (QEE+QE). Also runs of using queries expanded from target corpus to expand from external resources for retrieval (QE+QEE) can be tested. For our experiment, we selected the best run obtained using the QEE method where $R = 100$ and $K = 5$. This selection ensures that we have the best queries from QEE method. After QEE, the new queries are sent to the target corpus for QE. The results using different parameter settings are shown in Figure 3.7 and Figure 3.8.

The QEE+QE method produces similar curves to the QE method with better results. Figure 3.7 shows that more feedback documents for QE hurts the final retrieval effectiveness when using a fixed number of expansion terms. Figure 3.8 shows that adding more expansion terms does not change the final

retrieval effectiveness when using a fixed number of feedback documents. These are similiar conclusions to those we found in the QE experiments. These results also indicate that QEE provides useful feedback information to the original queries since QEE+QE gives better results than QE.

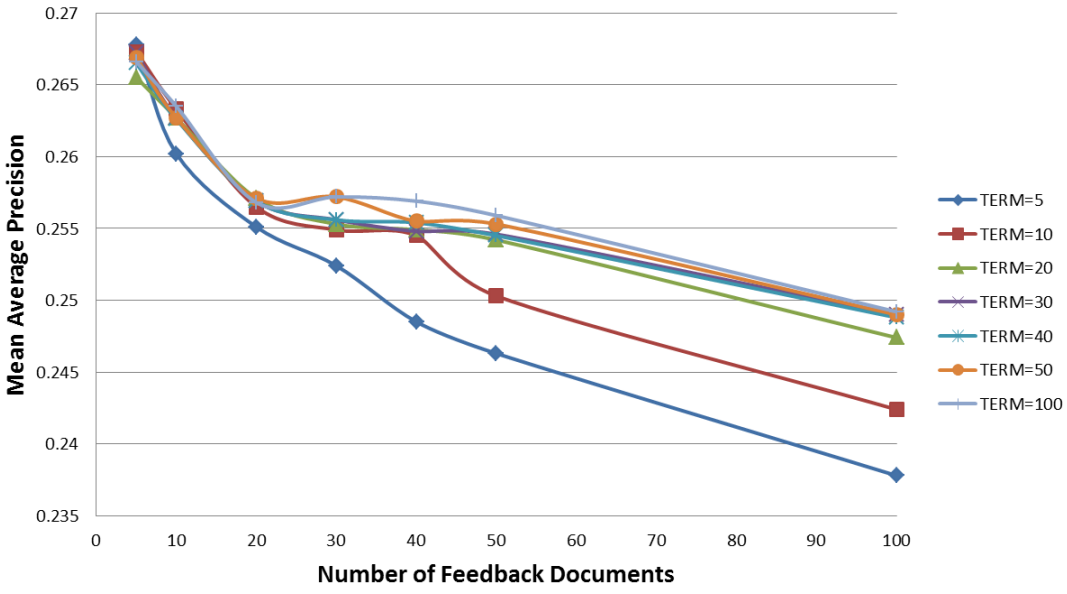

Figure 3.7: Results for QEE+QE for the WikipediaMM collection.

Furthermore, we test the results of QE+QEE. For QE, we select the parameter which produced the best result among our QE Runs, where $R = 5$ and $k = 5$. After QE from the target corpus, the expanded queries were applied to the Wikipedia collection for QEE. The results of QE+QEE Runs are shown in Table 3.3. The best result of QE+QEE outperforms the best result of QE method, but it does not outperform the best result of the QEE+QE method.

To compare the different query expansion methods, we show detailed results in Table 3.3. For each method, the best result after parameter tuning was selected. In Table 3.3, *Okapi* is the baseline run using only the Okapi retrieval

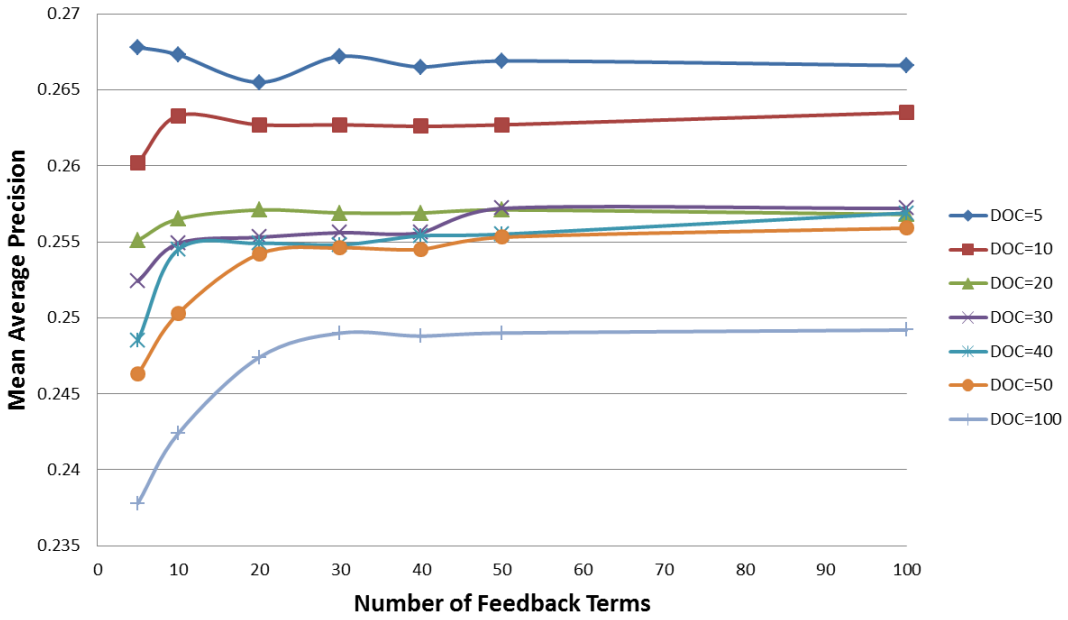

Figure 3.8: Results for QEE+QE for the WikipediaMM collection.

model without QE process from any collection. For the results, our analysis is based on MAP values with NDCG, R-Prec, P@10 also included in Table 3.3. The results show that:

- QE is an effective method compared to run without QE.
- QEE achieves comparable results to QE, and the difference between the QEE and QE is not significant by the MAP values.
- $QE + QEE$ does not achieve signifcantly better result compared to the QE method.
- $QEE + QE$ outperforms the Okapi, QE, QEE for four different evaluation metrics including MAP, NDCG, R-Prec and P@10. The result of $QEE + QE$ is significantly better than the Okapi, QE and QEE by the MAP

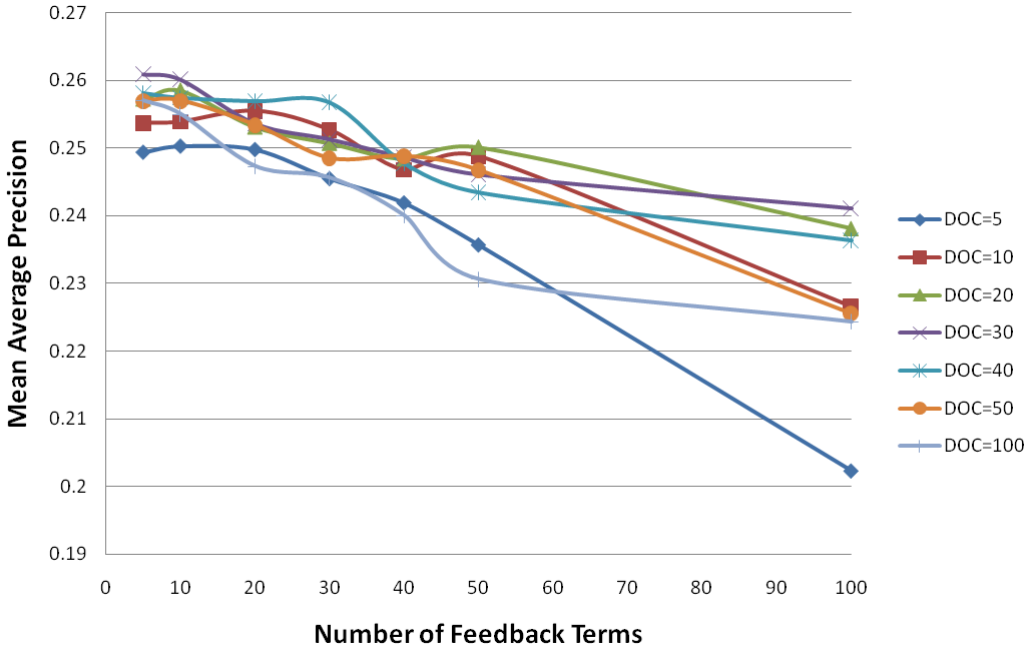

Figure 3.9: Results for QE+QEE for the WikipediaMM collection.

values.

Table 3.3: Results of different query expansion methods. '+' means the improvements over the baseline are statistically significant for the MAP scores.

Runs	MAP		NDCG	R-Prec	P@10
Okapi	0.2338		0.4931	0.2805	0.3453
QE	0.2588^{+}	+10.69%	0.5014	0.3035	**0.3720**
QEE	0.2551^{+}	+9.11%	0.5253	0.3011	0.3427
QEE + QE	$\mathbf{0.2678}^{+}$	+14.54%	**0.5268**	**0.3071**	**0.3720**
QE + QEE	0.2609^{+}	+11.59%	0.5255	0.2969	0.3693

In the next section, we propose a Definition-based Relevance Feedback (DRF) method using external resources for the text-based image retrieval task. In this method, we hypothesis that the definition documents (the documents from external resources whose title contains the query terms) are good feedback documents to provide feedback information for a user query, and that

these defintion documents can help to focus on the useful feedback documents.

3.3 Definition Document based Relevance Feedback

In this section, we introduce our DRF (Definition document based Relevance Feedback) method. DRF uses the query to get the feedback documents from the external resources before the retrieval process. It assumes the documents whose titles contain most of the query terms are relevant to the user query. These pre-found Definition Documents (DDs) are used to find more useful feedback documents in the external corpus. The DDs can be used to weight feedback documents in the process of selecting feedback terms from the external corpus. The framework of the DRF algorithm is shown in Figure 3.10:

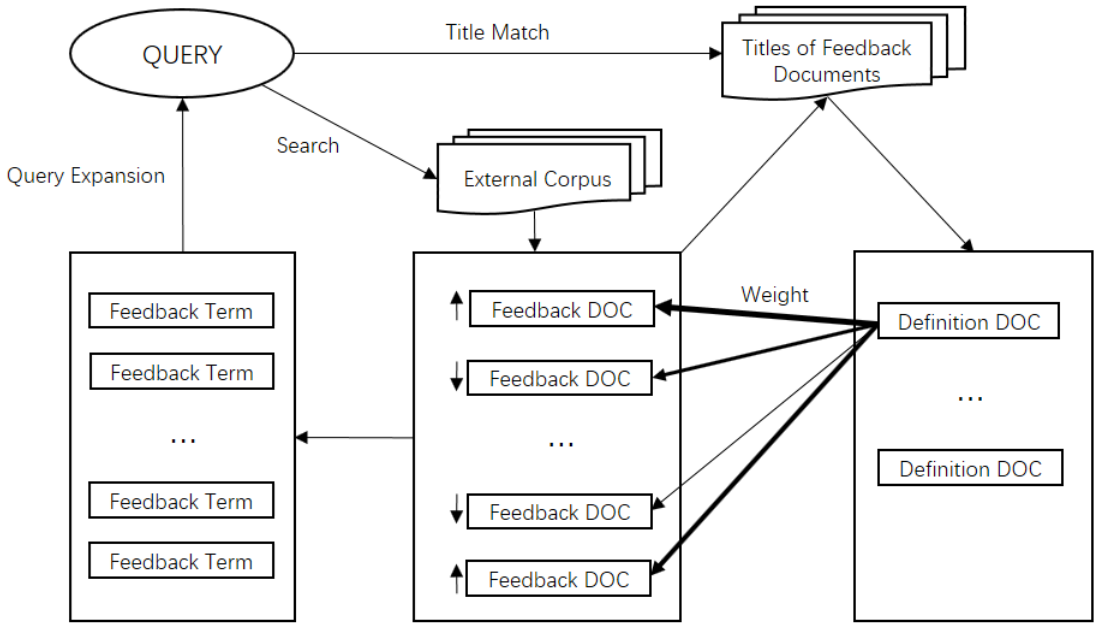

Figure 3.10: Flowchart of the DRF algorithm.

The details of this proposed method are as follows:

1. The user query is applied to an index of external resource (English DBpedia documents in our research) [1] to conduct an initial retrieval run to produce a ranked list. The top ranked documents from this retreival run is called external feedback documents.

2. The user query is applied to the top-ranked external feedback documents retrieved in stage 1 to conduct key-term title matching (see details in Section 3.3.1) to find the DDs for this query.

3. The DDs identified in the second stage are used to compare with the top ranked external feedback documents from the initial retrieval run in stage 1 using the Jaccard coefficient (refer to this as the "rating").

4. Similarity scores between the DDs and the external feedback documents are used to form a new weight for each external feedback document. A higher weight means that the external feedback document is more similar to the DDs.

5. Expansion terms are selected from the external feedback documents from stage 2 with new associated weights (described in Section 3.3.2). Feedback documents with higher weight add more feedback information.

6. The new expanded query is applied to the target search collection to carry out the final retrieval run.

[1] A DBpedia document is the first paragraph of the normal Wikipedia document.

As shown in Figure 3.10, the main difference between the DRF method and the standard Okapi feedback method is the way in which the feedback documents are weighted. For the Okapi feedback method, all the feedback documents are given the same weight. For the DRF method, greater feedback documents that are more similar to the definition documents are given greater weight. In Figure 3.10, the different width of the lines between the feedback documents and the definition documents indicates the different similarity levels between them. The more similar, the more weight the feedback documents are given. In Figure 3.10, the arrow on the left of the feedback documents indicates documents given more or less weight.

3.3.1 Identifying DDs by Keyterm Title Matching

In this section, we address the problem of finding the DDs for a query in the external Wikipedia corpus. For most queries, external documents strongly related to concepts expressed in the query can be found in Wikipedia. We refer to these external documents as "relevant" to the query in the sense that they essentially describe one or more concepts contained in the query.

Given a query such as "Ferrari", a Wikipedia DD appears among the top-ranked document list after an initial retrieval run. This DD can be found by searching the titles of all the Wikipedia documents. The document will also usually appear at the top of the ranked list when using the user query to search the entire Wikipedia abstract documents. In our implementation, we use the user query to find the DDs in the top ranked external feedback documents for the simple implementation. Figure 3.11 illustrates an example

user query with a DD in Wikipedia.

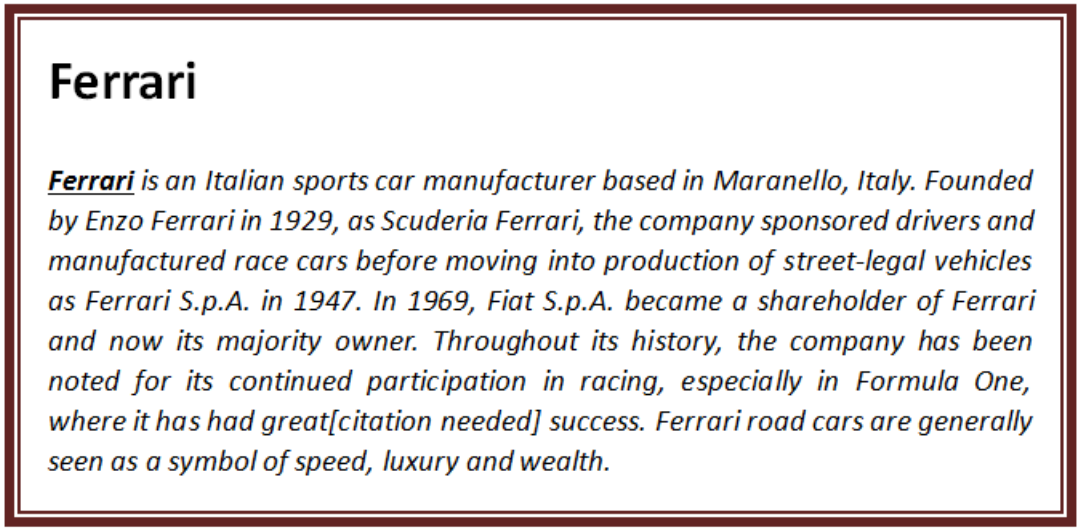

Figure 3.11: Definition document example.

Since a Wikipedia document whose title is exactly same as the user's query is not be found for all queries, our DRF method also allows a more relaxed matching approach for DDs. We use a partial matching approach to find Wikipedia documents whose title contains the key term of the user query as the DDs for the current user query. Given a query Q: $\{q_1,q_2,...,q_m\}$ and a document D with title T, the key term q_t $(1 \leq t \leq m)$ of the Q is the term with highest idf score given by Equation 3.1.

$$idf(t) = log\frac{N}{n} \tag{3.1}$$

In Equation 3.1, N is the total number of Wikipedia documents and n is the number of Wikiepdia documents containing the term t. We use the term's idf score trained from Wikipedia corpus to select key term in the current user query. This score is used rather than $tf \cdot idf$ since tf usually is 1 for

query terms. We use the idf score trained from the Wikipedia corpus rather than a corpus consisting of only the user queries since this method has been shown to be effective for query term weighting method in IR research [Salton & Buckley, 1997]. A Wikipedia abstract document D whose title contains q_t is called the DD of query Q.

Our approach is based on the following observations:

- The key term (term with high importance) in the user query indicates the user's main focus of the information need.
- The title indicates what the Wikipedia abstract document is about and distinguishes it from other documents.
- Wikipedia abstract document provides a direct description of the concepts of its title and includes background information which further elaborates on this topic.

Based on these facts, we assume that Wikipedia documents whose title contains the key term of the user query can potentially provide a richer vocabulary to describe the user information need than the terms in the original user query itself. Thus, we use the information in the retrieved Wikipedia collection to update the user query using a relevance feedback method.

3.3.2 Feedback Term Weighting

Our QE method ranks potential expansion terms using Equation 3.2 as described in [Robertson, 1991; Robertson *et al.*, 1994; Robertson & Spärck Jones,

1994].

$$Weight(t) = r * rw(t) \tag{3.2}$$

$$rw(t) = log\frac{(r+0.5)(N-n-R+0.5)}{(n-r+0.5)(R-r+0.5)} \tag{3.3}$$

In Equation 3.2, r is the number of known retrieved documents containing term t and rw is defined as Equation 3.3. In Equation 3.3, r is the number of known relevant documents term t occurs in; R is the number of known relevant document for a request; N is the number of documents in the collection; n is the number of documents term t occurs in. Given R feedback documents, this method assigns all feedback documents from the initial retrieval the same weight.

Issuing a query to the external resource results in the retrieval of x feedback documents and t DDs by title matching. Since the top ranked documents are not all relevant to the user query, it is not appropriate to assign the same weight to all of these documents. In our research, we seek to utilize information from an external documents to find documents are more likely to be relevant to the query in the process of relevance feedback.

Before the final retrieval process, we can find the DDs for the user query from the Wikipedia collection. These DDs are more likely to be relevant to the user query since the titles of the DDs are exactly or partially the same with the user query. Then we assume that the external feedback documents similar to the DDs are more likely to relevant to the user query. In this way, we can give the external feedback documents different weight when selecting

feedback terms in the process of external query expansion.

As an extension to the standard Okapi feedback method, we give different weights to the feedback documents, thus a new score replaces r in Equation 3.2. To compute our revised weighting score, we introduce two additional scores:

1. The similarity of the external feedback document fd_i $(1 \leq i \leq x)$ and the DD dd_j $(1 \leq j \leq t)$: $S(fd_i, dd_j)$

2. The average similarity score for DD j $(1 \leq j \leq t)$ with all external feedback documents: $sim_{avg}(dd_j)$

The similarity score of an external feedback document and DD $S(fd_i, dd_j)$ is computed using the Jaccard coefficient as shown in Equation 3.4, where V_{fd_i}, V_{dd_j} are the vocabulary sets of documents fd_i and fd_j [Jaccard, 1901]. Jaccard coefficient has been successfuly utilized in finding similar documents in previous research [Haveliwala *et al.*, 2002].

$$S(fd_i, dd_j) = S(dd_j, fd_i) = \frac{V_{fd_i} \bigcap V_{dd_j}}{V_{fd_i} \bigcup V_{dd_j}} \tag{3.4}$$

$S(fd_i, dd_j)$ is normalized into $[0, 1]$ using Equation 3.5, where $S_{min}(fd_i)$ is the minimum similarity score between feedback document i and one of the DDs j and $S_{max}(fd_i)$ is the maximum similarity score between feedback document i and one of the DDs.

$$S^{'}(fd_i, dd_j) = \frac{S(fd_i, dd_j) - S_{min}(fd_i)}{S_{max}(fd_i) - S_{min}(fd_i)} \tag{3.5}$$

The average similarity score of DD dd_j with all feedback documents ($sim_{avg}(dd_j)$) is given by Equation 3.6.

$$sim_{avg}(dd_j) = \frac{\sum_{i=1}^{x} S(fd_i, dd_j)}{x} \tag{3.6}$$

Based on Equation 3.5 and 3.6, the similarity score for feedback document i and all DDs is given by Equation 3.7.

$$G(fd_i) = \frac{\sum_{j=1}^{t}(S(fd_i, dd_j) - sim_{avg}(dd_j))S^{'}(fd_i, dd_j)}{\sum_{j=1}^{t} S^{'}(fd_i, dd_j)} \tag{3.7}$$

In Equation 3.7, we use a score $S(fd_i, dd_j) - sim_{avg}(dd_j)$ to divide the DDs into two groups:

- A group where $S(fd_i, dd_j) - sim_{avg}(dd_j) > 0$ contributes positive influence to the weight of the feedback document
- A group where $S(fd_i, dd_j) - sim_{avg}(dd_j) < 0$ contributes negative influence

If the similarity between feedback document and DD is higher than this DD's average similarity score with all feedback documents, the DD makes a positive influence in the weighting process, and vice versa.

With the new weights for all external feedback documents from the initial retrieval, the top expansion terms are selected using Equation 3.8, where r is the set of feedback documents which contain term t.

$$WT(t) = rw(t) \cdot \sum_{t \in r} G(fd_i) \tag{3.8}$$

Table 3.4: Overview on the definition documents.

No. of topics	75
No. of overall definition documents	262
Average No. of DDs per topic	3.5
DDs with total match	77
Topics with total match DDs	26

3.3.3 Evaluation of the DRF Method

In this section, we describe our evaluation of our DRF method. An overview of the DDs for the topics is shown in Table 3.4. 26 of the 75 queries have complete match DDs from Wikipedia, with a partial match with DDs being found for all the other queries.

3.3.4 Comparing DRF with PRF

The main purpose of the DRF method seeks to to utilize the information of external resources more effectively for QE. In this section, we compare our proposed DRF method with the standard Okapi feedback method (QEE) using the Wikipedia abstracts as an external resource.

First we examine results for different numbers of feedback documents and expansion terms. The results of these experiments are shown in Figure 3.12 and Figure 3.13. From these results, we can see that addition of more feedback documents from external resources does not change the retrieval effectiveness very much. These are similar results to those found for the QEE method using PRF as Figure 3.4. Similar curves are also seen in Figure 3.13 and Figure 3.6 when using different numbers of expansion terms.

Comparing the DRF and QEE method, the best results for these two meth-

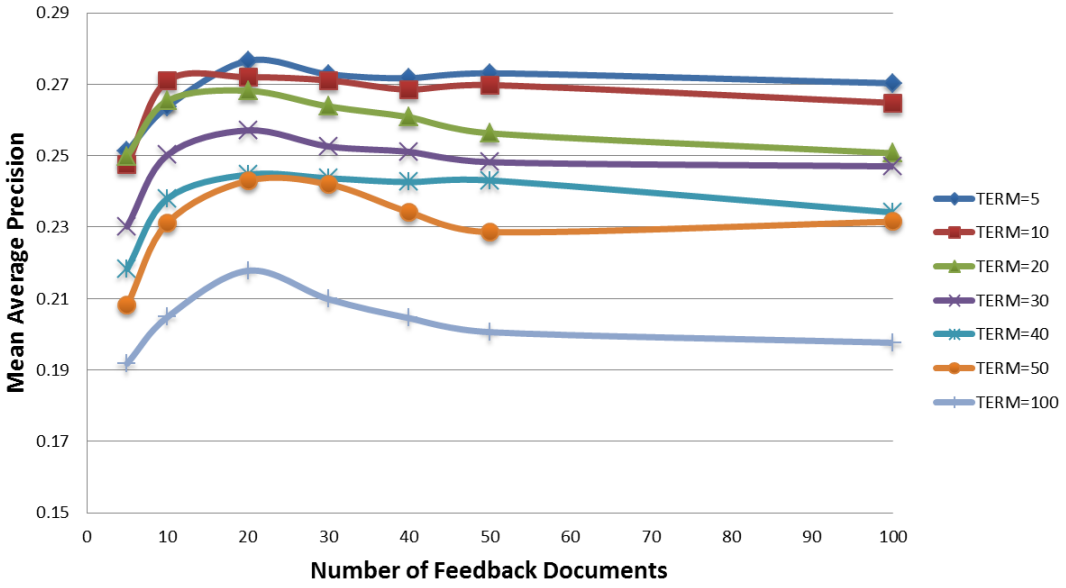

Figure 3.12: Results for DRF for the WikipediaMM collection using a fixed number of feedback terms.

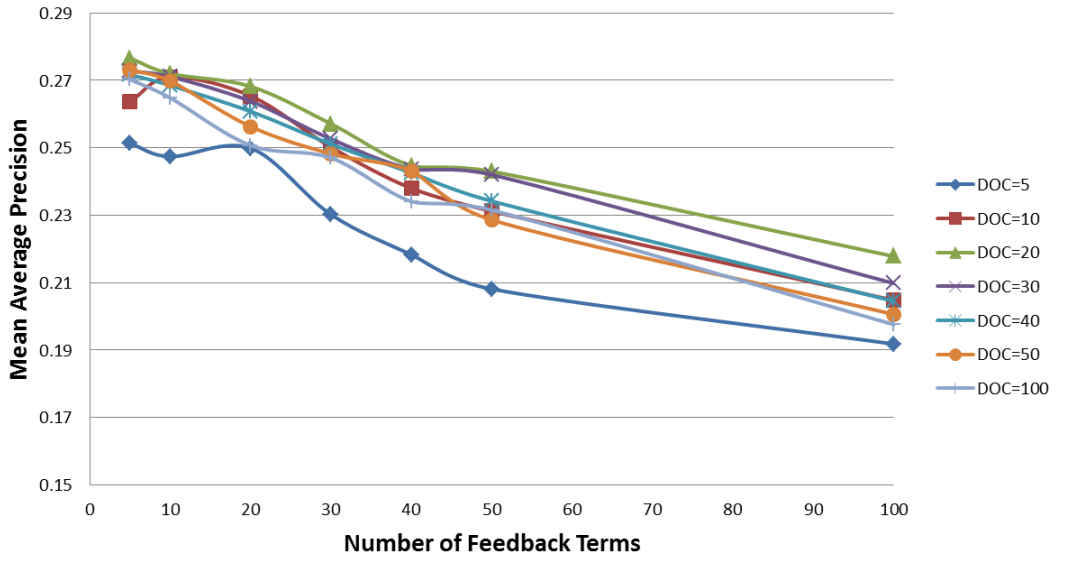

Figure 3.13: Results for DRF for the WikipediaMM collection using a fixed number of feedback documents.

ods are all observed when the number of expansion terms is 5. However the best results are achieved under a different number of feedback documents. When we set the number of expansion terms as 5, the results of the two methods are shown in Figure 3.14. These results show that the DRF method achieves its best result with a number of feedback documents ($R = 20$), while the PRF method needs more feedback documents to identify good expansion terms ($R = 100$). These results indicate that our proposed DRF method is useful for utilizing feedback documents to select good feedback terms in the process of query expansion. The results of DRF also indicate that adding too many feedback documents does not help improve the selection of good expansion terms. While for the QEE method, the top ranked feedback documents may not suit the selection of good expansion terms, adding more feedback documents helps to identify good expansion terms from the low-ranked feedback documents. In the end, the two methods achieve similar results when the number of feedback documents is large ($R = 100$). The change in the results with the increase in the number of feedback documents for these two methods can be seen in Figure 3.14.

We compare the best runs for the DRF and QEE methods in Figure 3.15. We also show detailed results of the best runs for DRF and QEE in Table 3.5.

Table 3.5: Comparison of Results for QEE and DRF.

Runs	MAP	NDCG	R-Prec	P@10
QEE	0.2551	0.5253	0.3011	0.3427
DRF	**0.2766**	**0.5393**	**0.3195**	**0.3573**

To further analyze the performance of the DRF method, we compare the run $DRF + QE$ which uses the expanded queries from DRF method to con-

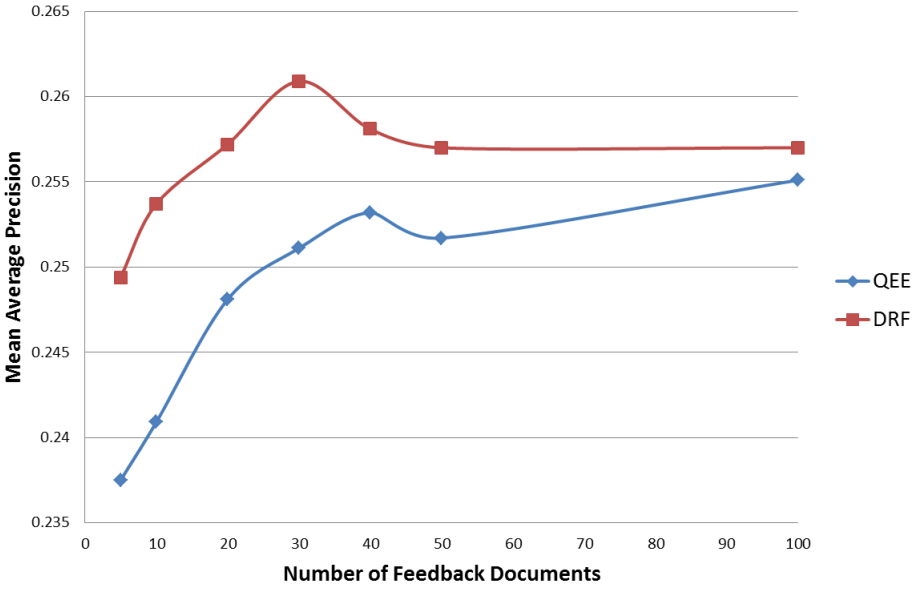

Figure 3.14: Comparison of DRF and PRF using the same number of expansion terms (5).

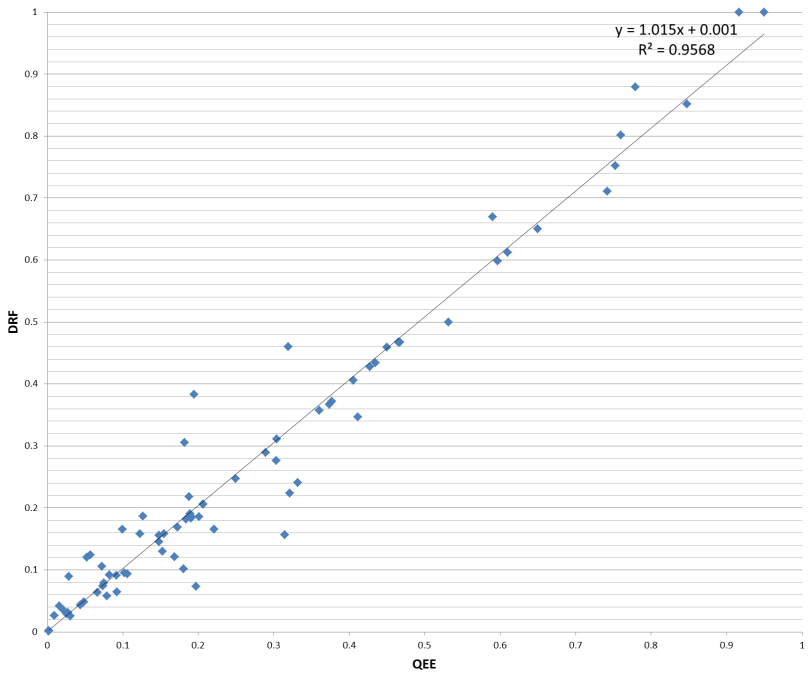

Figure 3.15: Comparison of DRF and PRF using the same number of expansion terms (5).

Table 3.6: DRF+QE performance under different parameter settings.

Number of Feedback Documents	Number of Expansion Terms						
	5	10	20	30	40	50	100
5	0.2482	0.2493	0.2593	0.2400	0.2327	0.2261	0.2128
10	0.2696	0.2752	0.2756	0.2645	0.2506	0.2465	0.2248
20	0.2796	0.2789	0.2785	0.2658	0.2552	0.2580	0.2368
30	0.2712	0.2808	0.2727	0.2597	0.2561	0.2577	0.2329
40	0.2744	**0.2813**	0.2679	0.2623	0.2531	0.2497	0.2221
50	0.2775	0.2803	0.2648	0.2623	0.2578	0.2443	0.2220
100	0.2751	0.2793	0.2595	0.2570	0.2511	0.2479	0.2153

duct query expansion on the target corpus, and then uses the new expanded queries for retrieval on the target corpus. We use the query produced from the best DRF run ($R = 20$ and $k = 5$) as shown in Figure 3.12 and 3.13. Detailed results for the $DRF + QE$ method are shown in Table 3.6. The results are also illustrated in Figrue 3.16. These results show that 10 feedback terms for QE after the DRF method gives the best result in our experiments. Adding more feedback terms from the target corpus reduces the overall results.

All the experimental runs can be associated with a query expansion process on the target collection. The results of comparing these runs are shown in Table 3.7. The results in Table 3.7 are the best results for these methods after parameter tuning. The results show that DRF outperforms both Okapi feedback on the target corpus and Okapi feedback on the external resource methods when they are associated with a process of query expansion on the target collection.

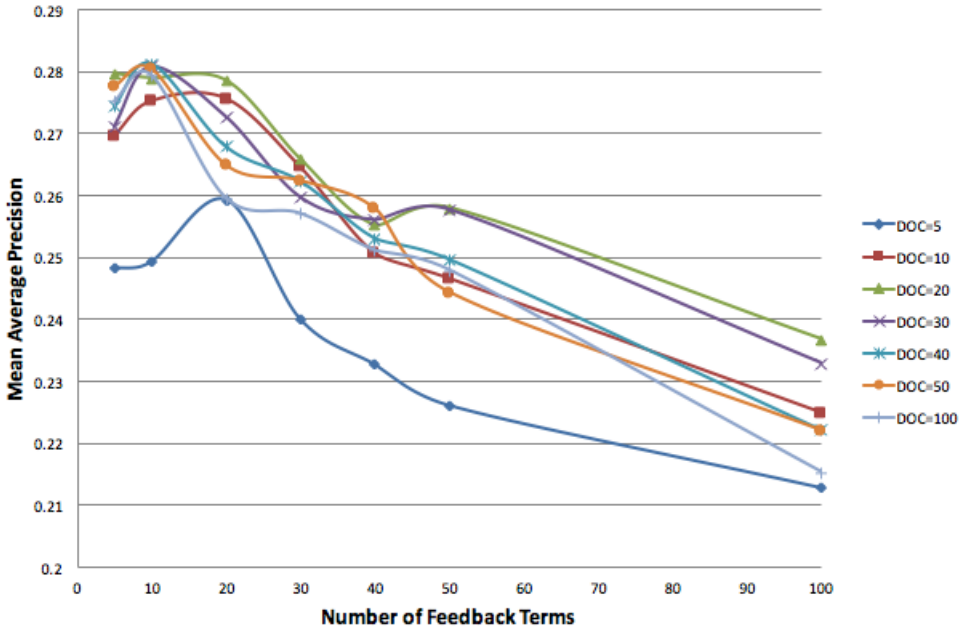

Figure 3.16: Results of DRF+QE method.

Table 3.7: Results Comparison for QEE and DRF. '+' means the improvements over the QE method are statistically significant for the MAP scores.

Runs	MAP		NDCG	R-Prec	P@10
QE	0.2588		0.5014	0.3035	0.3720
QEE+QE	0.2678	+3.48%	0.5268	0.3071	0.3720
DRF+QE	**0.2813**$^{+}$	+8.69%	**0.5482**	**0.3173**	**0.3760**

3.4 More Experiments on Second Query Set

To further investigate our DRF method, we examine results of QEE and DRF methods using a second query set. This query set contains 45 queries on the same Wikipedia image collection. The query set and relevance judgements are taken from the WikipediaMM 2009 task [Tsikrika & Kludas, 2010]. In Table 3.8, we compare four runs: Okapi retrieval model (Run: Okapi), query expansion on the target corpus (Run: QE), DRF on the external resource (Run:

DRF), and DRF on the external resource with subsequent query expansion on target corpus (Run: DRF+QE). To obtain these results, all the parameters were set to the same as those used to obtain the best results on the query set of WikipediaMM 2008 in our earlier experiments. These results verify the effectiveness of our proposed methods of external query expansion. Similiar conclusion can be summarized as:

- QEE gets similar results with the QE method, and both methods are significantly better than the Okapi baseline method.
- QEE combined with QE method gets the better result than the QE method.
- Our proposed DRF method combined with QE gets the best result in all runs. The improvement is significant compared to all other runs by the MAP values.

Table 3.8: Results On a Second Query Set. '+' means the improvements over the baseline method are statistically significant for the MAP scores.

Runs	MAP		NDCG	R-Prec	P@10
Okapi	0.1447		0.4533	0.1873	0.2444
QE	0.1549^{+}	+7.05%	0.4584	0.1984	**0.2556**
QEE	0.1581^{+}	+9.26%	0.4693	0.1803	0.2089
QEE+QE	0.1628^{+}	+12.51%	0.4689	0.1799	0.2111
DRF	0.1595^{+}	+10.23%	0.4744	0.1940	0.2378
DRF+QE	$\mathbf{0.1810}^{+}$	+25.09%	**0.4967**	**0.2061**	**0.2556**

3.5 Discussion

The key issue in blind QE is selecting useful expansion terms from pseudo relevant documents from the prior retrieval run. One precondition for QE is that

the target corpus contains sufficient information to enrich the original query. In our retrieval tasks, the sparse data problem in the target corpus breaks the requirement since the target documents usually are very short and do not contain sufficient term to produce stable and effective QE. Although applying a standard QE method achieves improvement compared to the baseline method, the results show that adding more feedback documents decreases the results. This can be explained since the target corpus may not provide sufficient good feedback information.

Our QEE method seeks to resolve the problem of sparse data by enriching the query from external resources. Since our chosen external resource is general and informative enough to provide useful information to the original query, the results from QEE work well and address the problem observed for the standard QE method. Also the results of the QEE method suggest that only the top expansion terms from the external resource are useful for enrichment of the original query (The results in Figure 3.6 show that top 5 feedback terms produces the best result for QEE method).

Furthermore, we find that the weights of the feedback documents for selecting expansion terms in the prior retrieval run from external resource are important. It can be explained that if we make good feedback documents contribute more, the QEE method will be more effective. In our proposed DRF method, we utilize the definition documents of the query to re-weight the feedback documents in the prior retrieval. Our results show that the new weighting scheme produces better feedback terms and leads to better retrieval results. Results on two query sets show that our proposed DRF method helps to improve the retrieval effectiveness.

3.6 Summary

In this chapter, we introduced external QE in the context of a text-based image retrieval task which has a typical sparse data problem in target corpus. As a solution, an external knowledge resource was introduced into the relevance feedback process. Our experiments show that the sparse target documents can not provide enough feedback information for user queries. The external corpus overcomes the insufficient information of the target corpus and enable useful expansion terms to be selected. Combining the external QE and QE from target corpus gives us a better result than using QE from target corpus only.

Furthermore, we presented a DRF method for QE from external resources. The method utilizes information from the external corpus by matching definition documents in the external corpus before the retrieval process. We assume that feedback documents which are similar to the definition documents of user queries provide more useful feedback information than those are not. Experimental results shows that DRF combined with QE method achieves significant improvement compared to any other methods in our text-based image retrieval task.

Based on the proposed research questions we propose in the beginning of this chapter, we get the answers as:

- How does the classical query expansion perform for retrieval tasks with sparse information? Based on our experimental results by the QE method, we find that QE method still can get a reasonably good result compared to the method without QE. But we also found that the target documents

cannot provide enough good feedback documents and it gave the opportunity to introduce an external collection in the feedback process.

- Which is better to compare query expansion from the target collection with query expansion from an external collection? In our experiments, query expansion from the target collection and external collection gets similar results, and the combination of these two methods gets a better result.

- Is the classical query expansion algorithm the best for query expansion using external resources? Our proposed DRF method gets a better result than the classical QE method when utilizing the external resources.

In the next chapter, we investigate document expansion utilizing external resources.

Chapter 4

Investigating the Utilization of External Resources in Document Expansion

In Chapter 3, we examined the potential for Query Expansion (QE) to improve Information Retrieval (IR). Our investigation explored standard QE methods using the target documents with external resources. Our experiments demonstrated that queries expanded using external resources can help to resolve the query-document matching problem to enable the retrieval of additional useful documents from the target retrieval collection.

In a typical IR process, the opposite side from user queries is the target document collection. In this chapter, we investigate the hypothesis that enriching document information for short documents can provide us with a further mechanism for improving IR effectiveness. We again explore the use of a Wikipedia abstract collection as an external resource to be applied in a

Document Expansion (DE) process. In this process, our study again examines the potential for external resources to augment the target collection by acting as a source of feedback information. In effect, DE seeks to expand the description of the topics in the documents by adding additional related terms in an attempt to address the query-document term mismatch problem from the document side.

DE has received much less attention than QE in previous research [Singhal & Pereira, 1999a; Billerbeck & Zobel, December 2005]. The limited amount of earlier research in DE has not reported conclusions regarding DE to improve retrieve effectiveness for IR tasks in general. In our research, we seek to answer the following questions by exploring a typical IR task for short documents, since short document retrieval task is more obviously affected by the sparse data problem in IR. This chapter seeks to answer the following research questions:

- Is DE using external resources useful for short document retrieval? Since the query-document mismatch problem may be more severe in short document retrieval than for long document retrieval, DE from external resources may have a better chance to resolve the query-document mismatch in this scenario.

- What is the best way to utilize a DE technique in short document retrieval? There are various ways in which can be used in a DE retrieval task. We aim to find an effective way to utilize DE for short document retrieval.

- Is DE a better method than QE for short document retrieval? Is IR

effectiveness improved if QE and DE are used in combination for the same task?

The structure of the chapter is as follows. We first introduce background and related work exploring the utilization of DE in IR in Section 4.1. We then describe our proposed method for DE using the Wikipedia abstract collection as an external resource and evaluate the method in Section 4.2. We discuss and summarize our findings on DE from external resources for this task in Section 4.3 and Section 4.4.

4.1 Background and Related Work

Document expansion (DE) is a technique for enriching target documents by adding topically related terms for IR. DE was first introduced in the field of speech retrieval where automatic transcriptions are noisy leading to query-document mismatch problems [Singhal & Pereira, 1999b]. In this work, documents were used as queries to retrieve items from an external collection of documents which were then used as the sources of expansion terms. The steps involved in this DE process were:

- Select a collection of documents that will serve as the source of related documents. In this work, the external collection was a collection topically similiar to the target retrieval collection. In this case, a test newswire collection was used as the external collection for a spoken news retrieval task.

- Find documents related to each speech document by using the document as a query to retrieve the 10 most similar documents from the external collection using a $tf \cdot idf$ method.

- Modify the speech transcriptions for each document using Rocchio's QE formula as shown in Equation 4.1 using the retrieved documents. In Equation 4.1, $\overrightarrow{D}_{old}$ is the initial document vector, $\overrightarrow{D}_i$ is the vector of the i-th related document, and $\overrightarrow{D}_{new}$ is the modified document vector, and α is the coefficient to adjust the relative weighting of the original document and related documents.

$$\overrightarrow{D}_{new} = \alpha \overrightarrow{D}_{old} + \frac{\sum_{i=1}^{10} \overrightarrow{D}_i}{10} \tag{4.1}$$

This DE method was applied on the TREC-7 Spoken Document Retrieval (SDR) track. This included 23 queries and recordings of 2,866 broadcast news stories. The external collection utilized in the experiments was a newswire data set from the same period as the target collection. The experimental results showed that enriching the documents via DE with the external collection yielded retrieval effectiveness, which improved not only over the original erroneous transcription, but also over a perfect manual transcription, since not only misrecognized words were added to the transcript, but also topically related words which had not actually been spoken. Using DE, the loss of retrieval effectiveness due to automatic transcription errors was reduced from $15-27\%$ relative to retrieval from human transcriptions to only about $7-13\%$ on alternative transcripts, even for automatic transcripts with word error rates as high as 65%. This work demonstrated that DE could be uti-

lized to enrich noisy text by an Automatic Speech Recognition(ASR) system for later text-based retrieval in speech retrieval.

Similar to the speech retrieval task, [Levow & Oard, 2002] reported work exploring post-translation DE for Mandarin news stories in an effort to partially recover terms that may have been mistranscribed, mis-segmented or mis-translated. This work was done in the context of a cross-language topic tracking task where English news stories were used to find Mandarin news stories on the same topic. The Mandarin news stories were translated into English, and then these English news stories were used as queries to search an external collection to find related documents. Selected terms were extracted from the top-ranked external documents (an external large newswire collection) to expand the translated English news stories. Then the expanded English news stories were indexed for retrieval using English queries. The results showed DE improved topic tracking effectiveness in the TDT-3 topic tracking task [1].

In these studies, speech recognition and machine translation produce text with noise which can impact on later text-based IR effectiveness. This research demonstrated that DE can be an effective method for addressing the problem of low effectiveness in noisy text such as transcribed documents and translated documents.

DE is not only useful for retrieval tasks with noisy data, it can also be utilized when the target data itself is not sufficient to build language models of the documents. In research in language model IR, a method to expand each document with a probabilistic neighborhood was proposed in [Tao *et al.*,

[1] http://www.itl.nist.gov/iad/mig/tests/tdt/1999/

2006]. This work was motivated by the insufficient sampling of documents in language modeling. DE in this work was essentially smoothing the document model by adding information from the neighbourhood of the documents. Cosine similarity was used to compute the neighbourhood relations of the documents. A document $d^{'}$ is generated using the pseudo term count shown in Equation 4.2.

$$c(w,d^{'}) = \alpha c(w,d) + (1-\alpha) \times \sum_{b \in C-\{d\}} (r_d(b) \times c(w,b)) \tag{4.2}$$

The parameter α is used to control the balance of the original document model and the expanded document model. $r_d(b)$ is computed as shown in Equation 4.3.

$$r_d(b) = \frac{sim(d,b)}{\sum_{b' \in C-\{d\}} sim(d,b')} \tag{4.3}$$

Here, d is the document for expansion, and $d^{'}$ is the expanded document, and b is the neighbourhood document, and $r_d(b)$ is the normalized cosine similarity, and C is the whole collection. Evaluation was carried out on six TREC data sets: AP (Associated Press news 1988-90), LA (LA Times), WSJ (Wall Street Journal 1987-92), SJMN (San Jose Mercury News 1991), DOE (Department of Energy), and TREC-8 (the ad-hoc data used in TREC8). The experimental results showed that DE method outperformed both no expansion baseline and the cluster-based model [Liu & Croft, 2004]. Compared to the no-expansion methods, DE-based language model achieved an improvement in MAP of between 4.4% to 15.5% in various collections. With respect to the cluster-based language model, the improvement in MAP of DE

method ranged from 2.5% to 7.7% for different test collections. The work also suggested that short-length documents obtain more information from their neighbourhood in the framework of language model retrieval. This arises since building useful language models for short documents is more difficult than for longer documents. This research shows that expanding the language models of documents helps to produce a better match between the document models and query models by relevance than method without expanding. Thus, the final retrieval effectiveness can be improved by the DE method.

Past research on DE for IR has also reported negative results. An attempt to employ DE in image retrieval for the ImageCLEF photo task 2007 [Grubinger *et al.*, 2008] degraded the performance by 28.24% in MAP when using the web pages as the reference corpus [Chang & Chen, 2007]. In this work, documents were expanded from the top-ranked snippets retrieved by a web search engine, with only the document title used as the query to search for relevant documents. The proposed DE method limited the expansion terms to those terms in the retrieved snippets near to the terms of original document within a $5 - term$ window. In this same task, QE achieved an improvement of 16.11% improvement in performance in terms of MAP compared to the run without QE or DE.

A study reported by [Billerbeck & Zobel, December 2005] showed that DE only has a limited effect and concluded that the technique was unpromising on several TREC newswire retrieval tasks. They examined three term-weighting methods for DE: Okapi BM25, Term Selection Value (TSV) [**?**] and Kullback-Leibler Divergence (KLD) [Croft, 2000]. In document centric DE,

each complete document was used as a query and the top expansion terms determined through local analysis (relevance feedback from the top ranked documents from an initial retrieval from the target document collection) were appended to the document. In term centric DE, each term was used as a query to find relevant documents by QE. Then the term was added to those top-ranked documents which did not contain this term. The expansion resource in these DE experiments was the target collecton itself.

The evaluation was based on six TREC newswire collections: WSJ2 (Wall Street Journal 1990-92), AP, NW (newswire collection from TREC-7 and TREC-8), FBIS (Foreign Broadcast), FT (Financial Times 1991-94), LA. Results on several TREC newswire data collections showed KLD worked better for document centric DE, while TSV was better than Term centric DE. For almost all these collections, the DE methods did not outperform a QE baseline based on evaluation using MAP. The inconclusive results of these existing studies encourage our research to better understand DE in IR tasks, especially for IR tasks with sparse data.

DE has also been investigated in various other areas of IR such as conversation retrieval [Wang & Oard, 2009], concept-based IR [Baziz *et al.*, 2007] and novelty detection [Zhang *et al.*, 2002]. A DE method was proposed for conversation text retrieval task [Wang & Oard, 2009]. The research exploited contextual properties (both explicit and hidden) to probabilistically expand each message to provide a more accurate representation of the message. This work targeted disentanglement, which seperated individual conversations from online discussions. In this work, each message in a conversation was expanded within its temporal and social context. In the process of expansion, messages

which were close to the original message in time were given higher weights. Evaluation carried out on a collection consisting of real text streams produced in Internet Relay Chat showed that the proposed method outperformed a non-expanded based baseline by 24% according to the criterion of F-measure.

Concept-based IR incorporating DE was investigated in [Baziz *et al.*, 2007]. Concept-based IR aims at retrieving relevant documents on the basis of their meaning rather than their keywords. In this work, the authors proposed to expand the documents to add concepts that were closely related to those expressed in the documents. This was done on a relatively small test collection which contained 25 topics and 7,823 medical paper abstracts. The proposed DE method outperformed a vector-based model [Bordogna & Pasi, 1995] which utilized BM25 as the term weighting method. Other research attempted to apply a structured lexical database such as WordNet to expand documents in IR tasks. Each noun in a document was used to find the hypernyms in WordNet [Zhang *et al.*, 2002]. If the hypernyms appeared in the query, the noun was replaced by the hypernym. This method achieved better results than QE in a TREC Novelty task, which aimed to find key factors in documents.

In summary, previous investigations of DE have met with mixed results in various IR tasks. However, DE has been relatively neglected as an area of research compared to QE. The positive findings of some investigations, particularly for noisy documents, indicate that DE is a potentially promising approach for improving retrieval performance for some IR tasks, with suitable attributes such as for text with noise produced by speech recognition or machine translation. In the framework of our thesis, DE forms one aspect of

RF via QE using external resources.

In this chapter, we re-visit DE in the context of retrieval of images annotated with brief textual labels. This task is challenging for IR since such annotations are generally short, often with no redundancy of description, and typically do not follow any particular standard in terms of vocabulary selection or level of detail, leading to a high likelihood of a mismatch with user queries. Thus, if we can build an improved connection between image annotations and user queries, it has the potential to greatly benefit retrieval effectiveness. In this context, DE becomes an attractive option if it can be shown to work reliably. Furthermore, we utilize a large external collection to enrich the documents in the DE process, which has not been examined in detail in previous research. This large volume of external documents could bring useful information, but also noisy information into the original document collection. Finding the right way to utilize large external collections in DE makes this a challenging research topic. In the next section, we introduce our proposed DE method using external resources.

4.2 Document Expansion using External Resources

In this section, we describe our investigation into the use of DE for text-based image retrieval. This task is chosen since the documents are generally very short and thus frequently fail to adequately describe the annotated image leading to significant query/document mismatch problems in retrieval. Our research seeks to explore the utilization of a large external collection for DE by directly adding index terms to the target documents in a short document

retrieval task.

Our initial method of DE is similar to a typical QE process. Pseudo Relevance Feedback (PRF) is used as a DE method with the Okapi algorithm [Robertson, 1991; Robertson *et al.*, 1994; Robertson & Spärck Jones, 1994]. PRF reformulates the query from two parts: the original query terms and expansion terms from the top ranked documents from the feedback source. In our research on DE, a Wikipedia abstract collection is again utilized as the external resource for feedback. The reason for choosing this Wikipedia abstract collection is the same as our QE research since Wikipedia abstracts are informative enough to enrich the sparse data of the target collection.

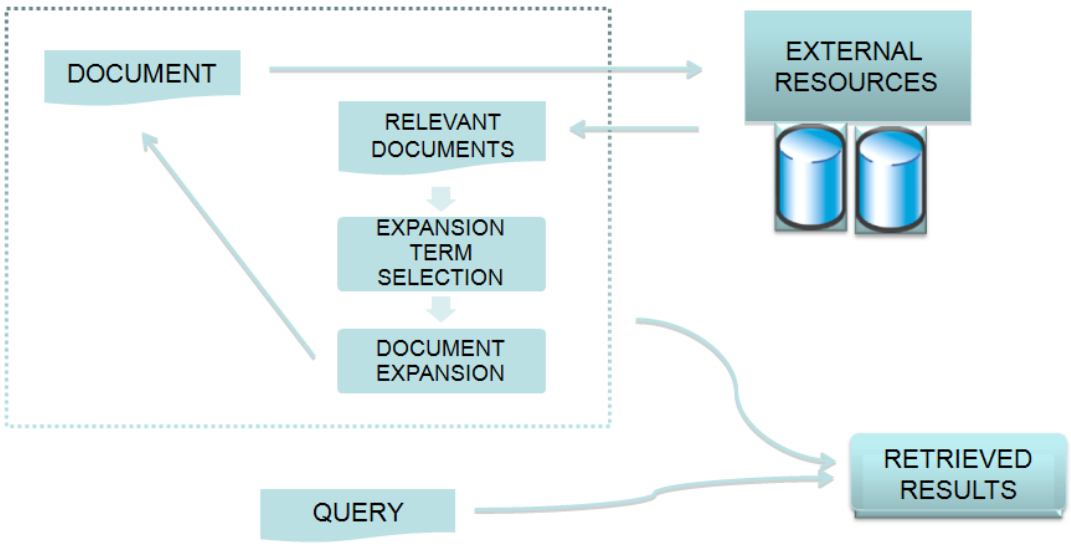

Figure 4.1: System overview for retrieval using document expansion using an external text collection.

Figure 4.1 presents an overview of the system for DE using the Wikipedia abstract collection. Each document in the original target collection is used as a query to retrieve items from the external collection. Expansion terms are extracted from the top ranked documents retrieved from the Wikipedia abstract

collection. The selected expansion terms are then added to the original document to form the expanded document. All expanded documents are then re-indexed for retrieval.

The Robertson Offer Weight (OW) is used to select the terms appearing in the top ranked documents from the Wikipedia abstract collection [Robertson, 1991], as shown in Equation 4.5. While the OW has been used effectively in QE, here we investigate its use for DE.

In computing the OW, r is the number of documents which contain term t_i in the top ranked documents and $RW(t_i)$ (Relevance Weight) is shown in Equation 4.4. In Equation 4.4, N is the total number of documents in this collection; n is the number of documents which contain term t_i. The terms with the highest OW are selected as the expansion terms to be added to the original documents for indexing.

$$RW(t_i) = log\frac{(r+0.5)(N-n-R+r+0.5)}{(n-r+0.5)(R-r+0.5)} \tag{4.4}$$

$$OW(t_i) = r * RW(t_i) \tag{4.5}$$

4.2.1 Evaluation of a Simple Document Expansion Method

In this section, we evaluate our proposed DE method for our text-based image retrieval task. This is the same task we evaluated for our proposed QE methods in Chapter 3. Since we showed our QE method to be effective for this task, we compare our DE method against the results for our QE method for this task.

In our experiments, the Okapi BM25 retrieval model in the Lemur toolkit [1] was used for retrieval. For the setting of the parameters in Okapi BM25 model, k_1 was set to 2.0 and b to 0.75. The setting is suggested as a good starting point for Okapi BM25 model in [Robertson & Spärck Jones, 1994]. For all documents in our experiments (queries and documents in the target collection and the Wikipedia abstract collection), 571 stop words (the list of stop words in the SMART system [Salton, 1971]) were removed. All the terms were stemmed using the Porter stemmer (implementation in Lemur toolkit). We again used the WikipediaMM 2008 collection for our experiments. This includes 75 queries and 151,520 documents with relevance judegements.

In a typical DE process, three parameters affect the experimental results: the number of feedback documents for each original document, the number of feedback terms for each original document, and the coefficent used to combine the terms from the original document and the feedback terms. In Table 4.1, we show the results in MAP for various combinations of feedback documents and feedback terms.

Table 4.1: Results of different parameter settings for DE methods in MAP.

DOC \ TERM	10	20	40	60	80	100
10	0.2377	0.2363	0.2418	0.2398	0.2325	0.2274
20	0.2358	0.2376	0.2510	0.2499	0.2482	0.2464
40	0.2359	0.2412	0.2516	**0.2552**	0.2535	0.2533
60	0.2310	0.2421	0.2464	0.2523	0.2507	0.2500
80	0.2303	0.2413	0.2528	0.2514	0.2550	0.2510
100	0.2271	0.2390	0.2503	0.2509	0.2523	0.2533

The results in Table 4.1 show that the best results come from using a rela-

[1] http://lemur.org/

tively high number of feedback documents and terms ($DOC = 40$, $TERM = 60$). These numbers are considerably higher than those used to achieve optimal results for our QE experiments in Chapter 3 ($DOC = 5$, $TERM = 5$).

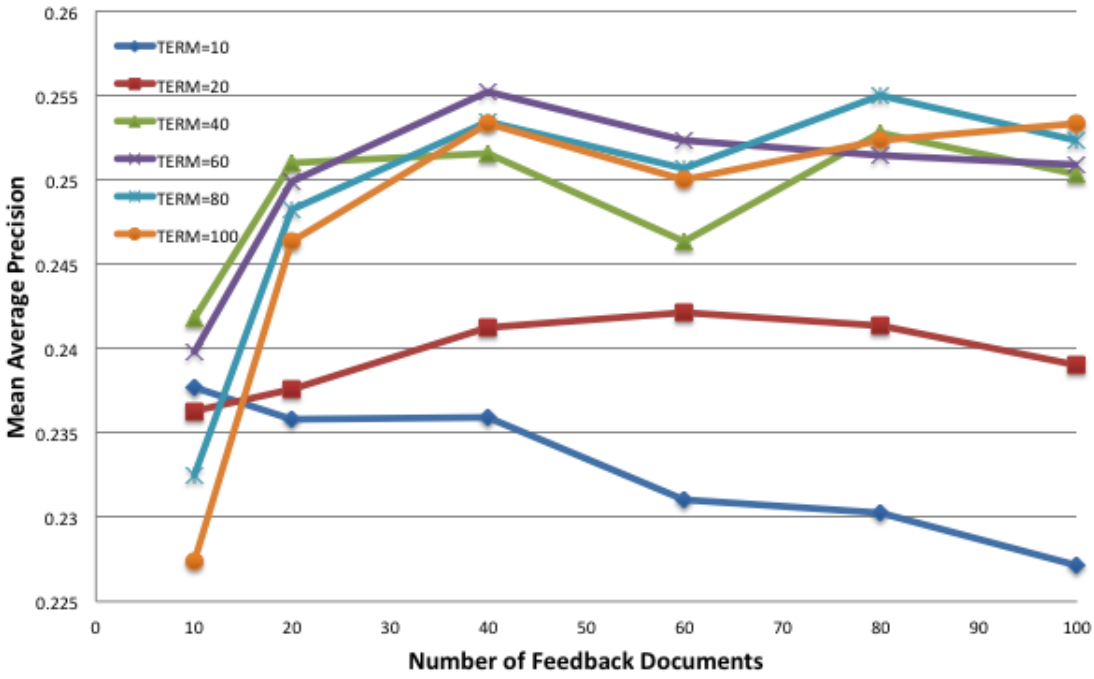

Figure 4.2: Results of the simple DE method with a fixed number of feedback documents.

To investigate the effect of differing numbers of feedback documents using the simple DE method, we fix the number of feedback terms and show the results with different numbers of feedback documents in Figure 4.2. From these results of using a fixed feedback terms, it can be seen that adding 40 feedback documents is a good choice. Adding more feedback terms does not change the retrieval effectiveness. When adding more than 80 feedback documents, all the results begin to drop. This suggests adding too many feedback documents brings noise into the original documents.

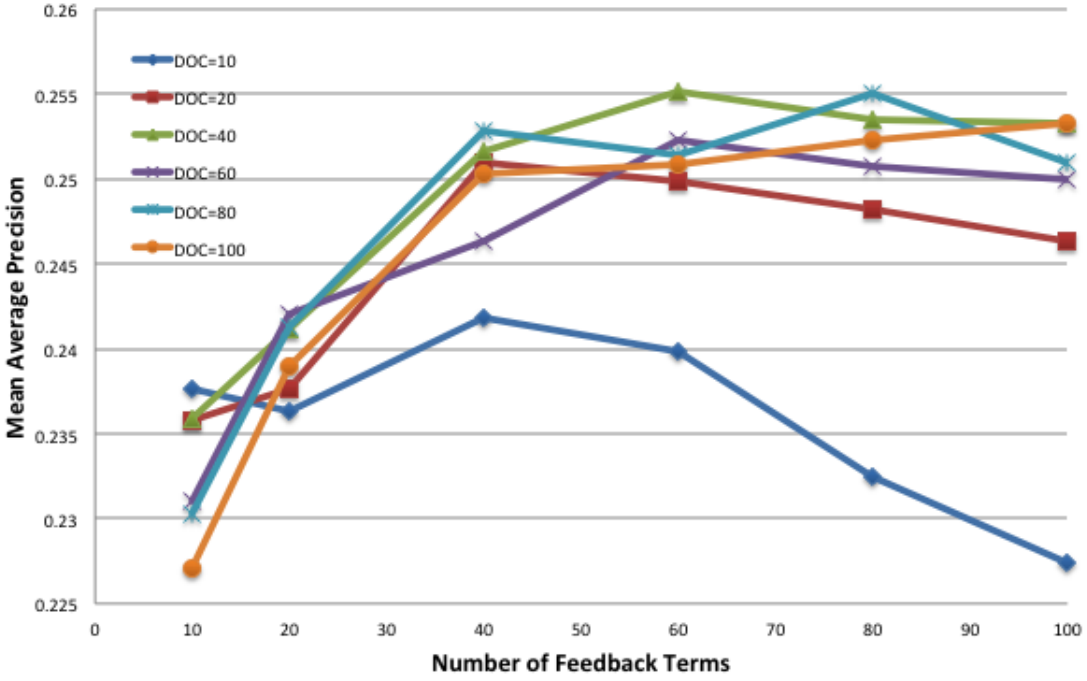

Figure 4.3: Results of simple DE method with a fixed number of feedback terms.

To investigate the effect of a different number of feedback terms using the simple DE method, we fixed the number of feedback documents and show the results with different numbers of feedback terms in Figure 4.3. From the results of the fixed number of feedback documents, when adding terms from 10 to 40, the results improve. This suggests that adding more terms is useful for our simple DE method. The results reach their highest level when 60 to 80 terms are added. The number is already larger than the average number of terms in the original documents. One conclusion that can be made when the number of feedback documents or feedback terms is less than 10, the results are lower than other Runs. This suggests that the simple DE method needs a high number of feedback terms and feedback documents.

In the initial experiments, the coefficient was set to 1 in all cases meaning that the feedback terms are given the same contribution as the terms in the original documents for indexing. Alternative values of the coefficient are examined later in this section. The experimental results of different Runs are listed in In Table 4.2 to compare the DE method with the no-expansion baseline and the QE method from the target collection.

Table 4.2: Comparing the simple DE with other methods. '+' means the improvements over the baseline are statistically significant for the MAP scores.

Runs	MAP		NDCG	R-Prec	P@10
no expansion	0.2338		0.4931	0.2805	0.3453
QE	**0.2588**$^{+}$	+10.69%	0.5014	0.3035	**0.3720**
DE	0.2552^{+}	+9.15%	**0.5326**	**0.3106**	0.3627

Our results show that the DE method achieves similiar performance to the QE method, but that the DE result is slightly lower than that for QE according to the criterion of MAP. Compared to the QE method, one advantage of the DE method is that DE does not require two retrieval passes at retrieval time. All the expanded terms are added into the index before retrieval, which saves retrieval processing time at search time.

Next we test alternative coefficient values in the DE process. Three typical coefficients (0.5, 1, 2) are tested in our experiments for the runs in Table 4.3. In our experiments, the coefficient is the weight for the feedback terms to be added in the original documents. If the coefficient is 0.5, the weight for the original terms will be twice that of the expanded terms. If the coefficient is 1, the feedback terms will be added with the same significance as the original documents. If the coefficient is 2, all the feedback terms will be added with twice the weight of the original documents. The results for these runs are

shown in Table 4.3. From the results in Table 4.3, it can be seen that the best choice of coefficient is below 1, since assigning too much influence to the expansion terms will impact on the meanings of the original documents. In the experimental Runs of Table 4.3, the number of feedback documents is set as 40 and the number of feedback terms is set as 60. This setting was shown to be the most effective Run for a coefficient value of 1 in Table 4.1.

Table 4.3: Results for different coefficient values for simple DE methods.

Runs	MAP	NDCG	R-Prec	P@10
coefficient=0.5	**0.2705**	**0.5506**	**0.3284**	**0.3680**
coefficient=1.0	0.2552	0.5326	0.3106	0.3627
coefficient=2.0	0.2355	0.5044	0.2931	0.3347

Since DE adds more terms into the target documents, it is interesting to examine changes in the vocabulary in the target collection. When adding 40 external terms to each target document (in our experiments, 40 feedback terms produces best DE result), the size of the vocabulary of the original target collection and the new expanded target collection is shown in Table 4.4.

Table 4.4: Comparison of the vocabulary size for original collection and the expanded collection.

Collection	Vocabulary Size
Original Collection	193,417
Expanded Collection	202,052

As shown in the Table 4.4, the difference in the size of vocabulary for these two collections is less than 5%. This indicates that although DE brings terms into the original target documents, these terms are mostly same as in the vocabulary of the original unexpanded target collection.

4.2.2 Document Expansion with Document Reduction

Using the whole original document as the query to find relevant documents in the external resource is a straightforward approach to DE. This method has been explored in previous DE work [Singhal & Pereira, 1999b]. In this approach, all the terms in the document are treated with the same weight as the terms in a query to find "relevant" documents. This simple approach may not be an optimal method for DE since the full documents contain much information which may not be useful for enriching the original document. We propose a method to extract the main topic from the original document and to use only the resulting key terms selected from the original document as the query to find relevant documents for DE in the external collection.

In our study, given an image metadata document "blue flower shot by user" for example, an obvious problem can be identified. In this document, the phrase "blue flower" is the main content of the document. Leaving the noise words "shot by user" in the query does not help to find useful "relevant" documents in the external resource. Similar observations can be made for many other documents since often a document contains many terms not associated with its main topic. This observation motivates us to use only the important terms in target document as a query to retrieve relevant documents in external resources. We propose to reduce the terms in each document by ranking its terms using the significance weights in decreasing order and removing all terms below a given cut-off value (taken as a percentage). We refer to this process as *Document Reduction* (DR) in our research. We utilize the Okapi BM25 function as the term weighting scheme to rank the terms

contained in each document since the Okapi BM25 function has been proven to be a good method to estimate term weights in IR research [Robertson *et al.*, 2000].

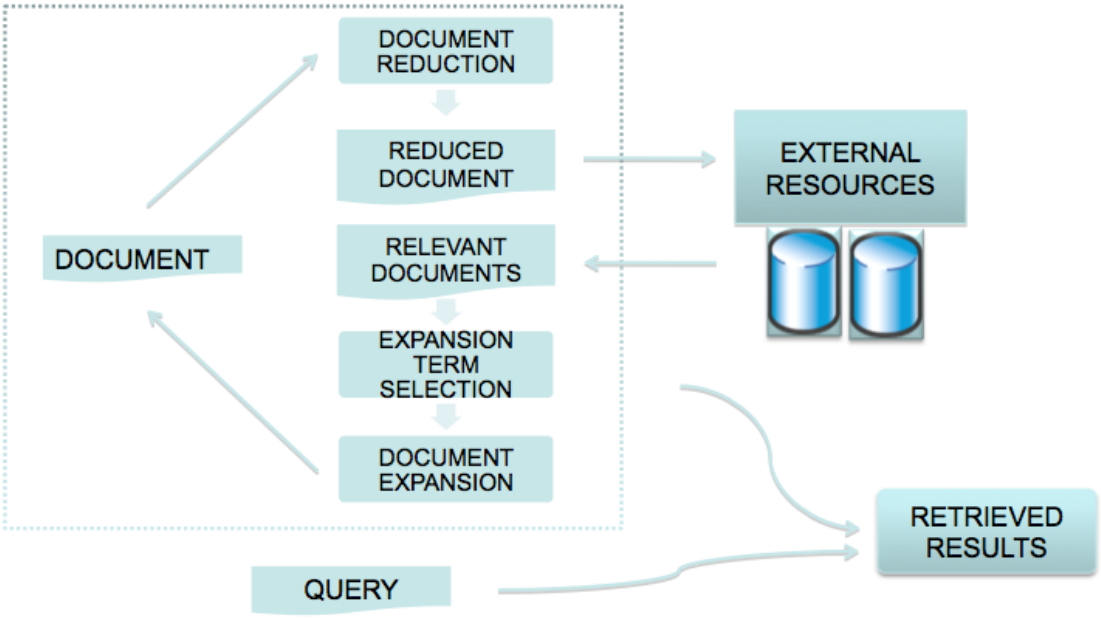

Figure 4.4: System overview for document expansion incorporating document reduction.

Figure 4.4 shows an overview of the system which utilizes the new DE method incorporating DR. In this system, DR is first applied to each document in the target collection to obtain a reduced document. Each reduced document is then applied to the external resource to retrieve relevant documents for DE. Expansion terms are extracted from the top ranked retrieved documents from external resource as in the simple DE method. The expansion terms are then added to the original document. The expanded documents are then indexed for retrieval.

As an example of the DE process, consider the following document from the WikipediaMM collection. After standard IR preprocessing, we have "bill-

cratty2 summary old publicity portrait dancer choreographer bill cratty. photo jack mitchell. licensing promotional". If the important words are selected manually from the document, a new reduced document might be "old publicity portrait dancer choreographer bill cratty". Using this reduced document as the query document is potentially better than the original one in terms of locating potentially useful DE terms from documents in the external collections, since it is more focused.

For automatic reduction of the document, we first compute all the term *idf* scores of the collection vocabulary as defined in Equation 4.6. Then for each word t_i in document D, we compute its BM25 weight using Equation 4.7.

$$idf(t_i) = log\frac{N}{n} \tag{4.6}$$

$$BM25(t_i, D) = idf(t_i) * \frac{tf(t_i, D) * (k_1 + 1)}{tf(t_i, D) + k_1(1 - b + b * \frac{|D|}{avgdl})} \tag{4.7}$$

Here $tf(t_i, D)$ is the frequency of word t_i in document D; k_1 and b are parameters ($k_1 = 2.0$, $b = 0.75$, starting parameters suggested by Robertson & Spärck Jones [1994]); $|D|$ is the length of the document D; and $avgdl$ is the average length of documents in the collection. For the above example, the BM25 score of each term for this document is shown in Table 4.5 after removing the stop words.

If we choose 50% as the percentage by which to reduce the document length applying DR, we obtain the new document "billcratty2 cratty choreographer dancer mitchell bill" for the above example. The automatically formed document is almost the same as the manually formed document shown above.

Table 4.5: Example of document *BM25* term weights

Term	Score	Term	Score
billcratty2	13.316	publicity	6.238
cratty	12.725	portrait	5.515
choreographer	12.046	promotional	4.389
dancer	10.186	photo	2.696
mitchell	8.850	summary	2.297
bill	7.273	licensing	2.106
jack	7.174		

We call the cut-off value for DR *document reduction rate*, which can be defined as: for *document reduction rate* p%, we keep p% of the original length of the document as the query for retrieving external documents for DE. The length of a document is defined as the number of terms in the document after stop words are removed and repeated terms are counted as different terms. Using the reduced document as the query to retrieve external documents will generate different top-ranked documents compared to the DE method without DR process. Thus different expansion terms are selected from these different top ranked documents, which then means that different expanded documents are created for the final retrieval. We evaluate the results of these two different DE methods in the next section.

4.2.3 Evaluation of Document Expansion with Document Reduction Method

In this subsection, we evaluate our proposed DE method incorporating DR. In this method, the document reduction rate is an important parameter to form the query for retrieval from the external collections. MAP results for a

range of DR rates are shown in Figure 4.5. For the runs shown in Table 4.6, the number of feedback terms and feedback documents for DE are all set to 10, and the coefficient is set as 1.0. These parameter settings give reasonable results for our simple version of DE experiments in Section 4.2.1.

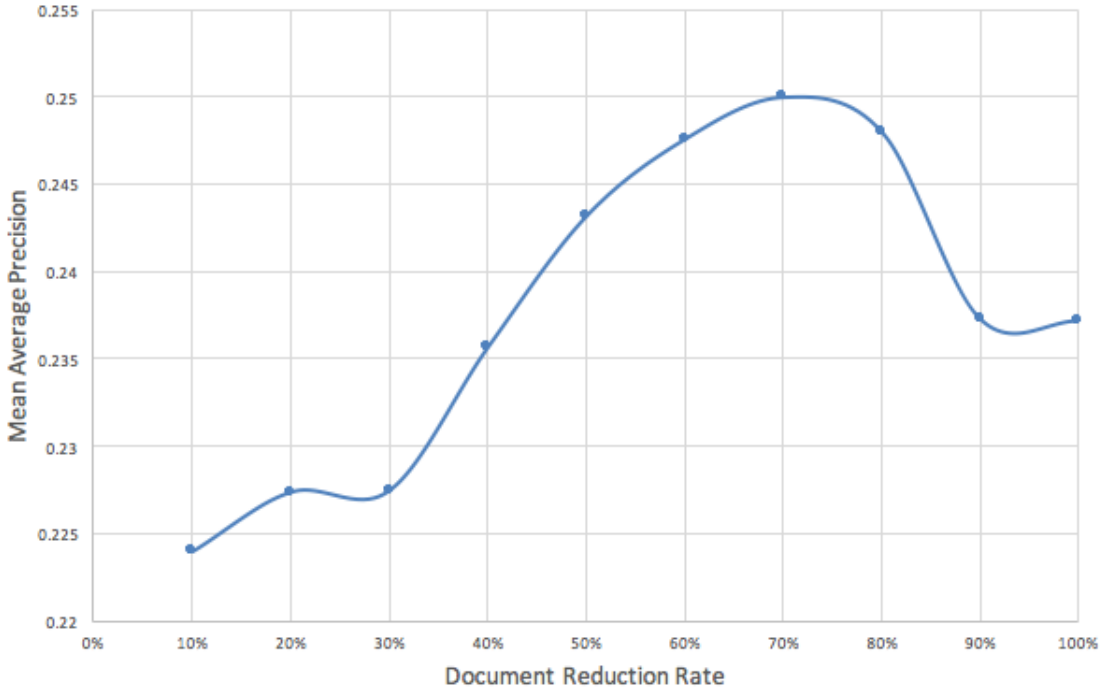

Figure 4.5: Performance of DE with different DR Rate.

The results show that a DR rate of 70% gives the best retrieval performance in terms of MAP. The full results of these runs are shown in Table 4.6 for reference. The results indicate that for DR, keeping the majority of terms but not all in the original document as the query for retrieving from the external resource is helpful for document expansion.

To explore the performance of the DE+DR methods, we examine different parameters using different numbers of feedback documents and feedback

Table 4.6: Document Reduction Rate.

DR Rate	MAP	NDCG	P@10	R-Prec
10%	0.2240	0.4841	0.3160	0.2746
20%	0.2274	0.4841	0.3427	0.2878
30%	0.2275	0.4914	0.3493	0.2765
40%	0.2357	0.4840	0.3373	0.2788
50%	0.2432	0.4942	**0.3627**	**0.2930**
60%	0.2476	0.5042	**0.3627**	0.2928
70%	**0.2500**	**0.5242**	0.3613	0.2895
80%	0.2480	0.4971	0.3600	0.2888
90%	0.2373	0.4821	0.3578	0.2777
100%	0.2343	0.5068	0.3200	0.2733

terms in Table 4.7. In Table 4.7, the document reduction rates are all set as 70%.

Table 4.7: Results of different parameter settings for DE+DR methods.

DOC \ TERM	10	20	40	60	80	100
10	**0.2500**	0.2420	0.2383	0.2292	0.2228	0.2156
20	0.2467	0.2429	0.2366	0.2289	0.2286	0.2237
40	0.2462	0.2396	0.2375	2276	0.2254	0.2208
60	0.2469	0.2444	0.2343	0.2298	0.2282	0.2256
80	0.2473	0.244	0.2373	0.2295	0.2254	0.2227
100	0.2459	0.2415	0.2339	0.2281	0.2269	0.2243

We show the results when the number of feedback documents or feedback terms are fixed in Figure 4.6 and Figure 4.7. These results suggest that adding too many terms harms the retrieval effectiveness. This suggests that the reduced documents can be good queries to find the relevant terms without needing to add unrelated "noise" terms. This is different from the simple version of the DE method where the whole document is used as a query. The same conclusion can be found that adding more feedback documents does not

change the results too much. This suggests that a small number of feedback documents is enough for DE with DR method.

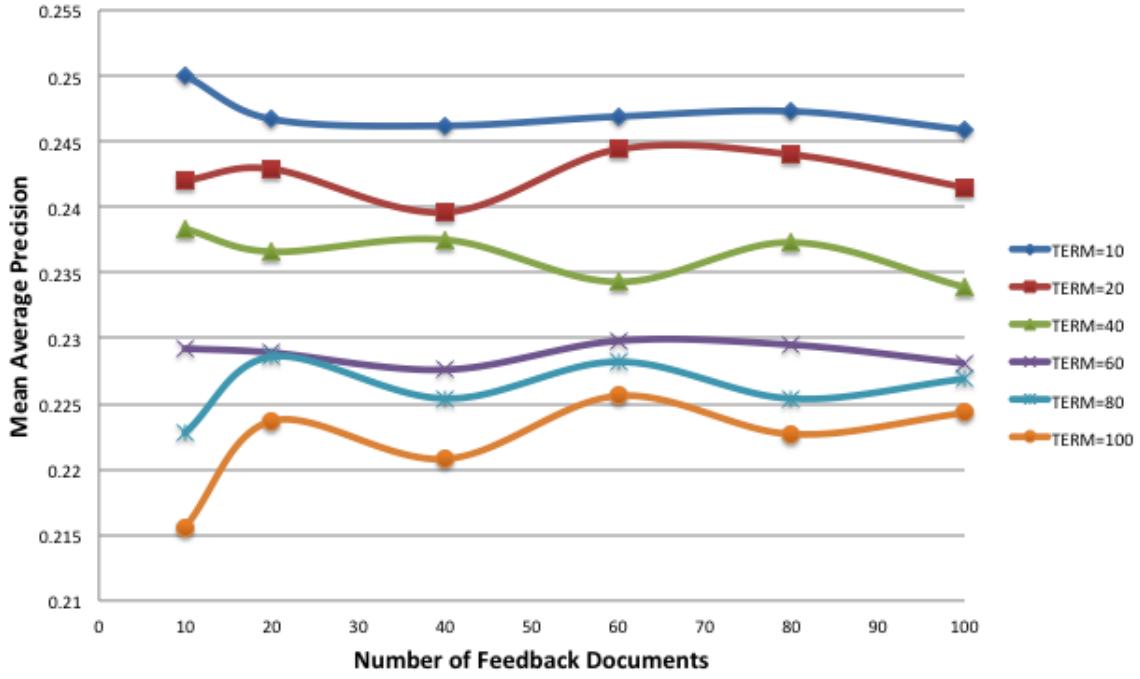

Figure 4.6: Results of the DR+DE method with a fixed number of feedback documents.

To further examine the effectiveness of the techniques explored in our work, we combine the DE with DR method with the QE method. The results for this combination are shown in Table 4.8. The results show that the DE method is improved by the combination. The combination of DR, DE and QE produces the best result in our experiments.

We show the best results of these different methods in Table 4.9. All the runs use the parameter settings which produce the best results in our experiments.

Compared to the standard Okapi method, we get a 17.75% improvement

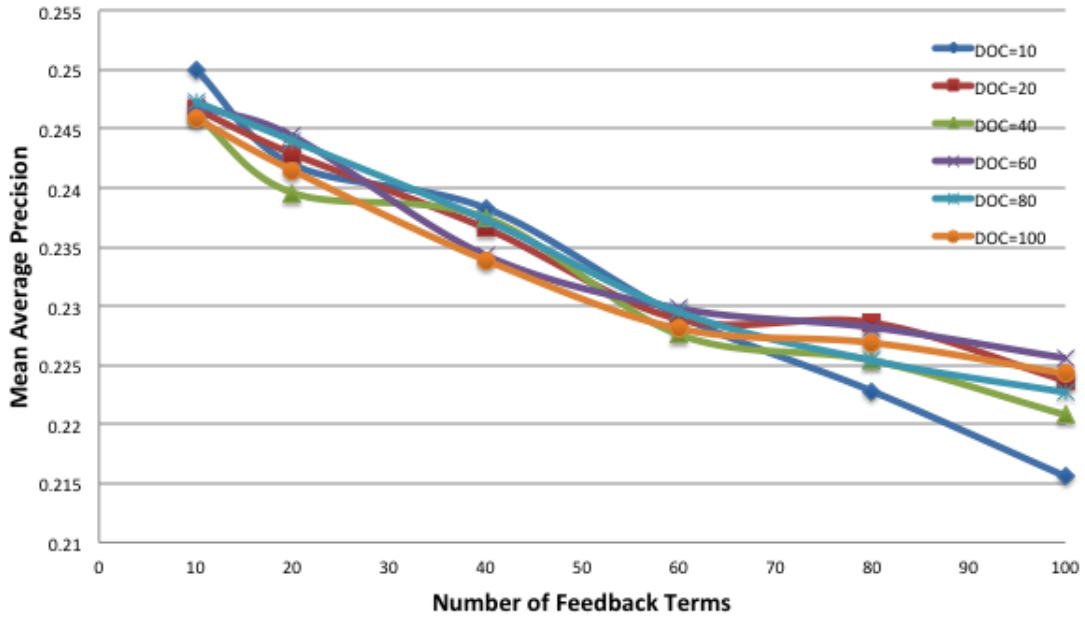

Figure 4.7: Results of DR+DE method with a fixed number of feedback terms.

Table 4.8: Results of different parameter settings for DE+DR+QE methods.

DOC \ TERM	10	20	40	60	80	100
10	0.2679	0.2681	0.2689	0.2707	**0.2753**	0.2723
20	0.2618	0.2627	0.2612	0.2615	0.2626	0.2633
40	0.2586	0.2553	0.2514	0.2513	0.2520	0.2520
60	0.2535	0.2530	0.2520	0.2504	0.2512	0.2500
80	0.2477	0.2474	0.2475	0.2476	0.2473	0.2468
100	0.2466	0.2469	0.2486	0.2484	0.2492	0.2489

Table 4.9: Comparison of results of different expansion methods. '+' means that the improvements over the baseline are statistically significant for the MAP scores.

Runs	MAP		NDCG	R-Prec	P@10
Okapi	0.2338		0.4931	0.2805	0.3453
QE	0.2588^{+}	+10.69%	0.5014	0.3035	**0.3720**
DE	0.2552^{+}	+9.15%	0.5326	0.3106	0.3627
DE+DR	0.2500^{+}	+6.93%	0.5242	0.2895	0.3613
DE+DR+QE	$\mathbf{0.2753}^{+}$	+17.75%	**0.5543**	**0.3078**	0.3600

in MAP when combining document reduction with a rate of 70% with DE and QE by incorporating document reduction.

Performing significance tests for our results, there are 75 topics for the WikipediaMM 2008 task. We compare the results from the baseline experiment without QE (Baseline) with the combination of document reduction, DE and QE (DR + DE + QE). For t-test the two-tailed P value is 0.0003. So by conventional criteria, this difference is considered to be extremely statistically significant. The increase in MAP of the results from DE+QE to DR+DE+QE is also significant ($p = 0.0326$).

For the results of the proposed DE methods, it can be explained that the documents retrieved at top ranks from the DE methods are more useful for QE than the unexpanded documents retrieved from the target collection.

4.2.3.1 Efficiency Issues

Since DE makes the index size bigger than the original one, we tested the index time for the unexpanded collection and expanded collection in Table 4.10. The computing environment for this experiment was a PC with a $Core2@2.0GHZ$ CPU, $4GB$ memory in an $Ubuntu/linux$ operation system.

Table 4.10: Index Statistics.

Runs	Baseline	Document Expansion	
Index Time (s)	17.005	20.780	+22.20%
Index Size (Mb)	51.6m	69.5m	+34.69%
Document Length	24	40	+66.67%

In addition, we tested the querying time for 75 queries for several different runs in Table 4.11. We do not find significant change in the query time for

our DE methods. The results show that QE increases query time, while the longer documents resulting from DE do not affect query time significantly.

Table 4.11: Average Query Time.

Runs	Query Time (s)	
Okapi	1.714	
QE	2.596	+51.46%
DE	1.852	+8.05%
DE + QE	2.734	+59.51%

4.2.3.2 Per-topic Analysis

In this subsection, we examine the per-topic difference between the Okapi run and DE+DR+QE run. There are 75 topics in the WikipediaMM 2008 task. Comparing the Okapi and DE+DR+QE method, for 47 topics the MAP improves and for 27 topics it decreases, while for 1 topic the MAP is unchanged as shown in Figure 4.8. We select an example document to observe the actual result of DE method.

For topic 23, the query terms are "british trains". Before DE, the document IDs for the top 10 results are: **1980516**[1], 222020, 316360, **228342**, **1032854**, **1475020**, **1192327**, **1487499**, **1125229**, **2227472**. Before DE, the P@10 is 0.8. While after DE, we observe P@10 as 1.0. All the top ten documents are relevant document: **1487499**, **1125229**, **1423946**, **1032854**, **1475020**, **1192327**, **1185704**, **1109791**, **2329048**, **1239902**. We select document *1423946* as an example shown in Figure 4.9 to observe the effectiveness of DE, since its rank for topic 23 improves from 116 in the Okapi run to 48 in QE run and 3 in

[1]Bold font means it is relevant with the topic

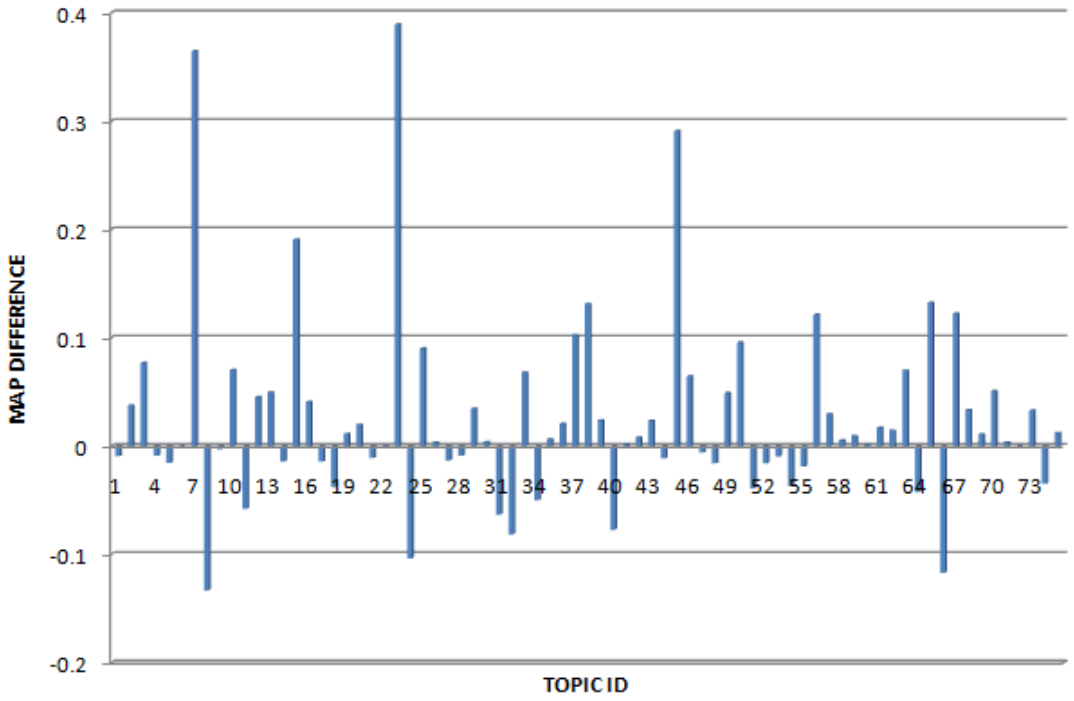

Figure 4.8: Average precision difference for DE.

DR+DE+QE run. In this example, we can find the term "train", which does not appear in the original document but is present after DE.

```
<DOC>
<DOCNO>1423946</DOCNO>
<TEXT>
<ORIGINAL>
norwich british rail class 960 class on 31st january 2004 at
the time this unit was painted in railtrack blue green livery
it has since been reclassified as british rail and repainted
in network rail yellow livery image by phil scott
</ORIGINAL>
<EXPANSION>
rail units multiple unit diesel blue electric locomotives
green train livery services type locomotive introduced freight
car passenger vehicles theotokos steam
</EXPANSION>
</TEXT>
</DOC>
```

Figure 4.9: Document expansion example.

4.2.4 Additional Experiments with Second Query Set

We also evaluated our DE method on a second query set. The results of these runs are shown in Table 4.12. This query set includes 45 topics and related relevance judgements. The same query set was used in Chapter 3 for the evaluation of our QE methods from external resources. For the experiments in Table 4.12, we use the same parameter settings as in Table 4.9 for the same methods.

From the results of the second query set in Table 4.12, similiar findings from the first query set can be observed:

Table 4.12: Results On a Second Query Set. '+' means the improvements over the baseline are statistically significant for the MAP scores.

Runs	MAP		NDCG	R-Prec	P@10
Okapi	0.1205		0.3840	0.1698	0.1978
QE	0.1251	+3.82%	0.3759	0.1667	0.1800
DE	0.1452^{+}	+20.50%	0.4564	0.1956	0.2400
DE+QE	0.1612^{+}	+33.78%	0.4910	0.1914	0.2400
DE+DR	0.1623^{+}	+34.69%	0.4668	0.2003	**0.2556**
DE+DR+QE	$\mathbf{0.1771}^{+}$	+46.97%	**0.4953**	**0.2077**	**0.2556**

- DE gets better results than the QE method.
- DE incorporating the QE method gets a better result than the DE methed only.
- The combination of the DE, DR and QE method gives the best result for this query set.

4.3 Discussion

Why does DE improve the text-based image retrieval effectiveness? From our observations, the image metadata text has very similar characteristics to a typical query text. It consists of few words and focuses on a single topic. In standard ad-hoc retrieval tasks (such as those at TREC newswire tasks and elsewhere) for text retrieval, documents are typically news articles which are longer and may cover more than one topic. Expanding a long document covering more than one topic using a DE algorithm may not be effective for subsequent retrieval, since it is hard to add additonal terms to these documents which will actually improve their retrievability since they will generally con-

tain a much richer description of these topics without adding words which dilute its focus. In our experiments, a metadata document is usually very short, which is an intrinsic advantage for the metadata document to make use a DE algorithm. Using the metadata document as the query, it has a better chance of locating relevant documents within the related external resources. Selecting the top expansion terms and adding them into the metadata document enriches the metadata document vocabulary, but does not weaken its meaning. Thus the expanded metadata document will have more opportunities to be searched effectively by users with an improved chance of query document match. Overall the effects are similar to those of QE. Another aspect in our experiments is the related external resource. The retrieval task is conducted on image metadata so we selected the Wikipedia abstract collection as the document expansion resource. The Wikipedia abstract collection covers the overall topics of general information. This selected external resource has been shown to be an appropriate resource for the DE process in our experiments.

We believe that an important difference between DE and QE is that the former can be improved by the process of document reduction since using the whole document as the query to find relevant documents is not the best way for DE. Document reduction can help to remove noise from the query document and lead to a better relevant documents ranked list. Another difference is that DE is conducted before indexing, and is thus an offline technique while QE is conducted at retrieval time and is an online technique having a significant computational cost at retrieval time. Thus DE has the advantage of having lower cost impact at retrieval time, while improving retrieval effectiveness.

4.4 Summary

DE from external resources can improve retrieval performance for a text-based image retrieval task. We have demonstrated this for metadata of image documents which can be viewed as short-length documents which usually contain few words to describe the content of the image. Less terms in documents cause higher change of query-document mismatch in the IR process. Expanding the metadata from related external resources can help to solve the query-document mismatch problem in this task. QE is the classical way to resolve the query-document mismatch, and our findings show DE to be a better method which outperforms the QE method for this task. This shows that DE is a very effective way to resolve the query/document mismatch. QE has higher cost at retrieval time since it is an online algorithm in a real search environment. DE is an offline method, and thus consumes offline computing time and this has great potential usability in the real search applications.

Since our external resources are also short-length documents, our experiment results show that a high number of the assumed relevant documents and assumed relevant terms in the pseudo relevant feedback process is a good choice. We find that using the whole document as the query to do DE can introduce too much noise, and we reduce the document by selecting important words, then use the reduced document as the query to get the relevant documents. This process can help to achieve higher retrieval performance. Finally, we find DE's main impact will take effect in the final QE process. Combining DR, DE and QE produces the best results in text-based image retrieval.

To answer the questions in the beginning of this chapter, our main findings

in this research are as follows:

- Is DE using external resources useful for short document retrieval? From our experimental results, we can find that DE can get a similar result with the classical QE method. Thus, DE can be seen as a useful method for IR tasks with the sparse data problem.

- What is the best way to utilize a DE technique in short document retrieval? Rather than using the whole document as the query for DE, we found that combination of the DE method with document reduction gets better results in our experiments.

- Is DE a better method than QE for short document retrieval? Is IR effectiveness improved if QE and DE are used in combination for the same task? Our best results show that the combination of DE and QE methods gets the best results. It indicates that these two methods should be used together in IR tasks with the sparse data problem.

In the next chapter, we examine the utilization of external resources for enriching user data for a personalized data retrieval task.

Chapter 5

Exploring External Resources in Personalized Modelling

In Chapter 3 and Chapter 4, we concluded that Query Expansion (QE) and Document Expansion (DE) using external information resources can be effective for improving text-based image retrieval. These investigations demonstrated that external resources can help to alleviate the sparse data problem from both the query and document sides. In this chapter, we extend our study on the utilization of external resources to enrich user data in IR tasks. In this investigation we explore the use of external resources for user modelling. A typical IR application incorporating user modelling is personalized search. Personalized search seeks to provide individualised search results for each specific user. In user modelling for personalized search, the user's historical data are used to build a user search interest model. The user data for this model often suffers from the sparse data problem since it is difficult to collect sufficient user data to build a suitable user model. This leads to inef-

fective or unreliable personalized search outputs. In this chapter, we propose a Wikipedia-based personalized modelling method for a personalized search task to resolve this problem.

Since using external resources for personalized search task is a new topic, we need to answer the following research questions to test whether the external resources can be helpful for improving the retrieval effectiveness for personalized search task:

- How to utilize widely available external resources for user modeling in personalization?
- Is there a simple and effective solution to utilize external resources for general personalized web data search task?

This chapter is structured as follows. Section 5.1 discusses related work on user modelling in personalized search. Section 5.2 introduces the framework of our proposed personalized retrieval system. Sections 5.2.1 and 5.2.2 describe the details of our work on personalized modelling including user modelling and document modelling using Wikipedia. Section 5.3 then presents the experimental set-up and results for a personalized search task. Section 5.4 concludes by discussing the implications of our findings for the use of external resources in personalized search.

5.1 Background and Related Work

User modelling for personalisation in search has attracted increased attention within the IR research community in recent years [Pretschner & Gauch, 1999;

Shen *et al.*, 2005; Ferragina & Gulli, 2005; Qiu & Cho, 2006; Dou *et al.*, 2007; Xu *et al.*, 2008b]. To build a user search interest model, usually some form of knowledge base is used to record a user's search interests. This knowledge base typically consists of many search categories such as the ODP web category. The user model (or user profile) can be created based on these categories with various weights to indicate the search interests of each specific user. The user model is then utilized to direct the search system to produce personalized search results for the user.

There are many studies describing the construction of user models for personalized search. [Pitkow *et al.*, 2002] describes two general approaches to personalising search results for individual users. One method extends the user's original query with the user's specific interests (query augmentation); the other re-ranks the results individually for different users (result re-ranking). In query augmentation, the similarity between the query and the user model is computed and the query is augmented by terms seen in previous searches. In result re-ranking, the user model re-ranks search results based upon the similarity of the content of the target documents in the retrieved results and the user profile.

[Liu *et al.*, 2002] matches search results with categories that the user is interested in. [Gauch *et al.*, 2003] automatically creates user profiles by classifying user data using a web directory category. These profiles are found to significantly improve search results. [Jeh & Widom, 2003] utilises the user's profile to compute the importance of web pages which are of interest to a specific user. [Ramanathan & Kapoor, 2009] creates user profiles using Wikipedia. In this work, documents are mapped to a set of Wikipedia concepts, then a hi-

erarchical profile was constructed from these concepts. Finally, these concepts are annotated with information that may be helpful in information filtering or advertising.

User modelling has been widely applied in web search. [Kritikopoulos & Sideri, 2005] proposes an approach of search engine personalisation based on Web communities. [Liu *et al.*, 2004] suggests a novel technique to learn user profiles from the user's search histories. A user profile and a general profile are learned from the user's search history and a category hierarchy, respectively. User profiles are then used to improve retrieval effectiveness in web search. [Micarelli *et al.*, 2007] illustrates several important user personalisation approaches and techniques developed for the web search domain, along with examples of real systems currently being used on the Internet. [Micarelli *et al.*, 2007] categorises several important types of personalized search approaches:

- Current context: the current context of the user, such as the browsed pages, emails or edited documents, are exploited to recognise the user's needs and used to retrieve documents related to the user's activities.

- Search history: user historical data, such as search results, documents selected by the user, anchor text, topics in the web pages, data such as click through rate, browsing pattern and number of page visits, are used to build user models and personalized the search results.

- Rich representations of user needs: user data such as user feedback on results is included in the user query to obtain the personalised search results.

- Collaborative approaches: collaborative approaches deliver relevant resources based on previous ratings by users with similar tastes and preferences.

- Result clustering: result clustering groups the query results into several clusters for easy reading of the results by topics.

- Hyper-textual data: hyper-textual data approaches rank the search results that match the user-selected topics higher, providing tailored output for each user.

In this chapter, our personalized search method utilises the user's search history data for user modelling, since search historical data is easy to collect, and it costs the least effort on the part of the user and has been widely used in many real personalized systems. In previous research, two methods have typically been used to model user search interests from the user historical data. One is to model user search interests using pre-defined key terms [Keenoy & Levene, 2005]; the other is to model user search interests using a category system such as DMOZ [1] [Chirita *et al.*, 2005]. Table 5.1 shows a brief comparison of these used user modelling methods from user historical data.

Table 5.1: Overview of User Modelling Method

Modelling Method	Advantage	Drawback
Key Terms	easy to utilise	difficult to maintain; difficult to update
DMOZ	models web users well	DMOZ is not updated; categories are too broad

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

Although work exists exploring the personalized search task, no previous work has characterised the personalized search as a sparse data problem. In this setting, the query lacks an explicit statement of the user's search interests, the user's historical data is not sufficient for user modelling, and the documents do not explicitly reveal the underlying topical interests. It is impossible to tag manually every user query and web document with key terms to describe the missing knowledge. And this brings a problem that needs to be solved by automatic methods.

Our previous research in chapter 3 and 4 utilized external resources to alleviate the sparse data problem at the query and document sides. In this chapter, we focus on the sparse data problem in the data used in the process of user modelling. We propose to utilise the external resources in the process of the user modelling to enrich the user models using this additional information. We hypothesise that these richer user models have a better chance to effectively personalise the search results compared to the non-enriched method.

Previous research has usually utilized web categories to build the user models where the actual textual information in the user data is ignored. Our method utilises the textual information in the user historical data to build the user models from the external resources. We propose to utilise Wikipedia documents as our user modelling resource for personalized search, since it has broad coverage of human knowledge and is updated frequently.

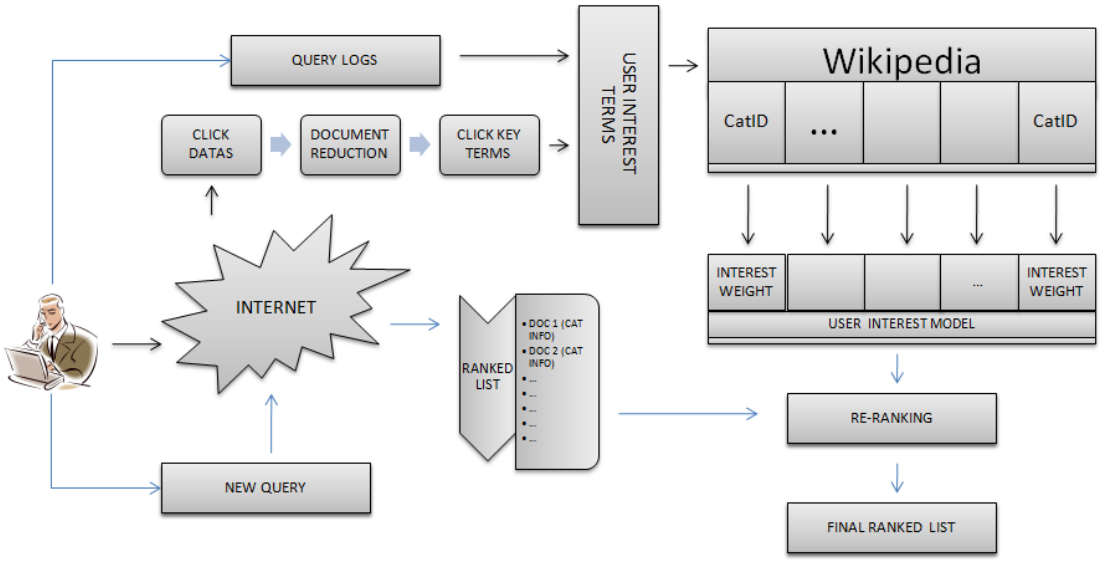

Figure 5.1: Wikipedia for user modelling.

5.2 External Resources in Personalized Modelling

Our research aims to utilise external resources to build a general user modelling method for personalized search. In this section, we first describe the system framework for our experiments in Figure 5.1. The basic process of the user modelling method is based on the previous research described in [Pretschner & Gauch, 1999]. We replace the web category based user modelling used in this previous work with our Wikipedia solution. The system consists of three parts:

- User modelling from the user's past search data using Wikipedia (associating the user data with various weights of Wikipedia categories).
- Document modelling using Wikipedia (associating the target documents

with various weights of Wikipedia categories).

- Re-ranking the search results for a new query from the current user by using the user model and document models of the top ranked documents.

Personalisation research aims to bridge the mismatch between the user's query and the underlying topics of target documents by incorporating knowledge of the user's existing search interests. To do so, a well-structured knowledge system is needed. In our research, Wikipedia (Wikipedia abstract collection) is selected for use as this knowledge resource. Wikipedia is used as the external resource because it can provide good coverage for the topics of the user context. Before items can be used to build user models and document models, Wikipedia documents are clustered into categories using an unsupervised algorithm. In our research, each Wikipedia document is assigned to a single category for simplification. These categories can be labelled with a unique *category id* automatically. After clustering, each Wikipedia document is associated with a unique *category id*.

To build the user model, our system starts from the user's historical queries. Each historical query is used to search the Wikipedia collection using a standard text retrieval algorithm. This process produces a ranked list containing the top N retrieved Wikipedia documents. These documents are assumed to be relevant to the historical user query. Since each Wikipedia document is associated with a *category id*, all these *category id*s can be combined to create a vector for this historical user query. The length of the vector is N, and it may include duplicate *category id*s since the top N Wikipedia documents

may include documents in same Wikipedia category. If a user has k historical queries, a total of k N-length vectors is generated. These search interest vectors are then merged to create a user model. The details of the merging algorithm are described in Section 5.2.1.

Another resource used to build the user interest model is their click-through documents. These click-through documents can also be assumed to indicate the user's search interests. The click-through documents are also used as queries to search Wikipedia. Thus similar to the user historical queries, user models can be produced using the click-through data.

To compute the similarities between the user models and the target documents, the target documents are also required to be associated with document models. For the target documents, each document is used as a query to search Wikipedia to produce a ranked list. The *category id* of the top ranked Wikipedia documents are recorded as the underlying topics of this web document. These *category id*s can form a vector for this document. The vector is assumed to describe the underling topics of this document. This process is called *document modelling* which creates document models for target web documents.

When a new query from the same user arrives, the user model and document models are used to address the sparse data problem between the user search interests as expressed by the query and the underlying topics of the target documents. To utilise the user models and document models, new queries are applied to the target document collection to obtain an initial ranked list using text retrieval algorithm. Each document in the ranked list is associated with a document model. The similarity between the document model and

user model is used to predict the possibility that this document satisfies the user's information need, based on the assumption that the user's search interests are consistent. Combining the text retrieval score and the similarity score between the user model and document model produces a new ranking score for the top ranked documents in the initial retrieval. Re-ranking using this new score is used to adjust the ranked position for every top ranked document. The details of this algorithm are described in remainder of this chapter.

5.2.1 Application of Wikipedia for User Modelling and Document Modelling

To model the user search interests, we choose Wikipedia as the knowledge category system. Wikipedia contains a large amount of category information for each document. In the official category system of Wikipedia, the Wikipedia documents are divided into twelve broad categories: reference, culture, geography, health, history, mathematics, nature, people, philosophy, religion, society, technology. However, a user's search interests will be more specific than these broad categories. Thus, these high level categories are not sufficient to model a user's search interests. In this work, we propose using a clustering algorithm to group the Wikipedia documents into categories. We do this using one of most popular methods as k-means clustering algorithm to group Wikipedia documents [Steinbach *et al.*, 2000]. The k-means clustering procedure is as follows:

1. The first document processed is placed in the first cluster.

2. Each document in the collection is compared to each existing cluster and assigned to the highest scoring cluster that exceeds the specified threshold score.

3. If no cluster score exceeds the threshold, the document is placed in a new cluster.

4. Repeat steps 2 and 3 until all documents have been assigned to clusters.

The similarity of documents in step 2 is computed using a cosine similarity shown in Equation 5.1. In Equation 5.1, the documents A and B are described by vectors including the term frequency of n terms. n is the total number of individual terms in the collection vocabulary. In this study, the threshold to assign a document to a cluster is 0.1 as suggested in the Lemur toolkit [1].

$$similarity = cos(\Theta) = \frac{A \cdot B}{\|A\| \, \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}} \tag{5.1}$$

Our Wikipedia clustering results are shown in as Table 5.2, the distribution of document numbers into categories is shown in Figure 5.2. As shown in Figure 5.2, about half of the Wikipedia categories have less than 10 documents. This indicates that the clustering algorithm not only groups documents into popular categories, but also places them into categories with only a few documents. This provides the opportunity to model user search interests at a more specific level. By using the k-means unsupervised clustering algorithm, each Wikipedia document is marked with a *category id* from 1 to 4,785.

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

Table 5.2: Results of Wikipedia Clustering

Number of Clusters	4785
Average No. of Documents of a Cluster	70
Largest No. of Documents in a Cluster	15803
Smallest No. of Documents in a Cluster	1

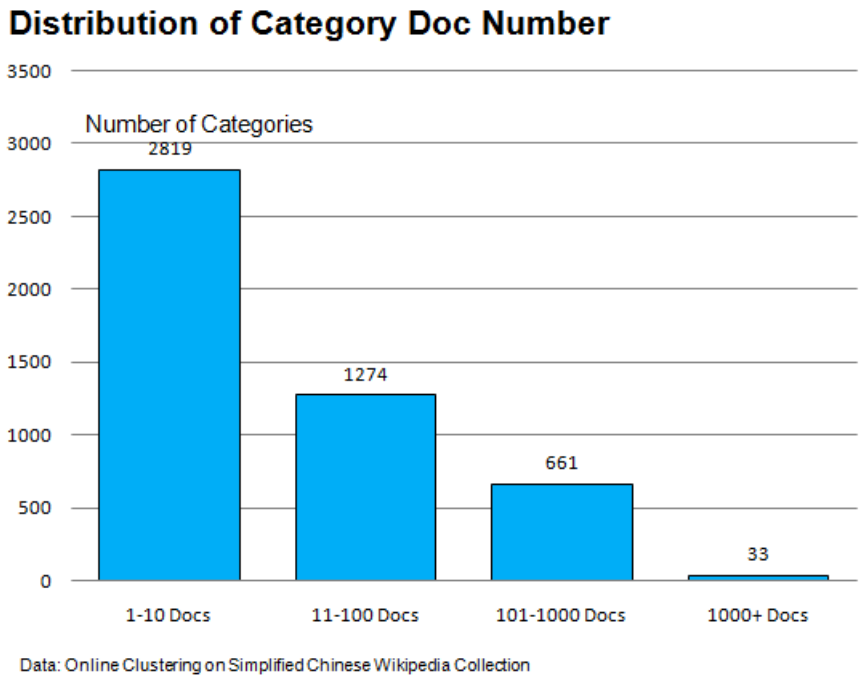

Figure 5.2: Distribution of number of documents in a cluster.

To model a user's search interests, the most useful resource is the user query logs. Query logs consist of the user's historical search queries and corresponding click-through documents. Our user modelling methods use these two types of resources separately. For the historical queries, we use every query to search the Wikipedia collection by Okapi BM25 retrieval model. The top N documents retrieved for each query are assumed to be relevant. Since each Wikipedia document is associated with a *category id*, we use these *category id*s to form a 4,785 dimensional vector: (cat_1, cat_2,...,cat_{4785}). For the

element in the vector, if the top N documents contain the corresponding category, the value of the element is set as 1, and all the other elements are set as 0. For each user historical query, there exists a $4,785$ dimensional vector.

If the user has k historical queries, we merge all these vectors into one. The new vector still contains $4,785$ dimensions and the value of the element is the sum of the corresponding elements in the k vectors which the new vector is merged from. We refer to this as the historical queries based *user interests vector*.

Using the same method, the click-through documents from a specific user can be used as queries to search the Wikipedia collection. This produces a click-through documents based user search interest model for the specific user.

From an alternative perspective, each web document is written by an author. This author also has their specific interest, so this document can also be associated with an interest vector. To compute this association, we use the document as a query to search Wikipedia. The top ranked returned Wikipedia documents are assumed to be relevant to this web document. The *category id*s of these documents form a vector for this web document. This can be recorded as (cat_1, cat_2, ..., cat_{4785}) where $4,785$ is the number of Wikipedia categories. If the top ranked Wikipedia documents contain the specific category, the corresponding element of the vector is set to 1, and all the other elements are set to 0. The vector is taken as the document model of the web document.

5.2.2 Re-ranking Retrieval Results

For a new query from a user, a standard text retrieval method is used to compute a ranked list from the target collection. This ranked list will be the same for every user for this query. To provide a personalized result to user, we re-rank the general result using the user model and the document model.

In the ranked list, each top ranked web document is associated with a category vector (cat_1, cat_2, ..., cat_{4785}) as described in Section 5.2.1. For this user, there exists an user model (vector (cat_1, cat_2, ..., cat_{4785})) built from the user's search interests as indicated from their historical queries or click-through documents. The $Score_{interest}$ is used to describe the relationship between a web document and a user search interests model defined in Equation 5.2.

$$Score_{interest} = \frac{U \cdot D}{||U||||D||} = \frac{\sum_{i=1}^{n} U_i \times D_i}{\sqrt{\sum_{i=1}^{n}(U_i)^2} \times \sqrt{\sum_{i=1}^{n}(D_i)^2}} \tag{5.2}$$

The score is the cosine similarity between the user model U and the document model D. The value of $Score_{interest}$ is between 0 and 1. To re-rank the search results, we define $Score_{re-rank}$ as the similarity score between the target document and the user query based on the initial rank and $Score_{interest}$ in Equation 5.3.

$$Score_{re-rank} = \frac{1001 - Rank_{initial}}{1000} + \lambda * Score_{interest} \tag{5.3}$$

Here, 1000 is the number of the documents in the ranked list for re-ranking, $Rank_{initial}$ is the ranked position of the document in the initial run. In the Equation 5.3, any initial retrieval method can be utilized and the ini-

tial ranked list be re-ranked. The 1000 documents are then re-ranked by the $Score_{re-rank}$. The ranked position is used as the initial score for re-ranking and $Score_{interest}$ is used to adjust the documents rank in the initial ranked list. For re-ranking, the documents are re-ranked by descending order of $Score_{re-rank}$. The $Score_{re-rank}$ ensures that a document with a high $Score_{interest}$ for the user ranks high in the re-ranked result.

5.3 Evaluation

In this section, we describe our experimental investigation to evaluate our proposed method. In order to do this, we use the following external resources: Wikipedia collection, user logs from a search system, and the corresponding target web collection. We use data from a Chinese commercial search engine - SOGOU.COM (NASDAQ: SOHU). The data includes one month's user query logs and a target Chinese Web collection. The format of each line in the user logs can be described as: **UserId, UserQuery, RankedPosition, RankOfUserClick, ClickThroughUrl**. In the log, each line describes one search activity from one user.

- *UserId* is the unique id for this search engine user; *UserQuery* is a search query input by this user.
- *RankedPosition* is the ranked position for the click-through URL in the ranked list.
- *RankOfUserClick* is a sequence number of the user clicks for this URL.
- *ClickThroughUrl* is the URL of the click-through document for *userQuery*.

In this search log, only the clicked documents are recorded and the unclicked documents from the same query are not recorded. Useful data entries in the user logs for our research in this chapter are *UserId*, *UserQuery*, and *ClickThroughUrl*.

Table 5.3: Overview of Experiment Data

Data	Number
Users	80
Test Queries	80
Training Queries	734
Training Click-Through Links	2,311

Table 5.3 shows an overview of the experimental log data. The 80 users with the most number of search queries during the month were selected from the SOGOU query logs. Each user is associated with one testing query. To obtain the user models for these users, 734 historical search queries and 2,311 click-through documents were used. The target collection is a subset of the SOGOU Chinese Web collection. This subset contains the documents visited in the month's query log. The reason to remove the other web documents from the target collection is:

- It is difficult to process the original web data (5TB) on a typical PC environment.
- Search of a large collection is not our research interest in these experiments.

Our external resource for user modelling is the simplified Chinese Wikipedia document set. Information about the Chinese Wikipedia Collection (dumped

in Jan. 2011) [1] is shown in Table 5.4. Information about the target Chinese Web collection is shown in Table 5.5.

Table 5.4: Overview of Chinese Wikipedia Collection

Number of Documents	332,900
Number of Terms	10,959,403
Number of Unique Terms	232,858
Average Document Length	32

Table 5.5: Overview of Chinese Web Collection

Number of Documents	507,262
Number of Terms	425,885,526
Number of Unique Terms	3,747,439
Average Document Length	839

From the overview of the Wikipedia collection and the target collection, we can see that:

- Compared to the Wikipedia documents, the target web pages have longer average length.
- The Wikipedia collection has less individual terms, while the web data has 16 times the number of individual terms in Wikipedia data.
- This difference arises even through there are a similar number of documents in Wikipedia data and web page data collections.

These findings suggest that the web page data is more complex than the Wikipedia data. It can be explained that the Wikipedia data are well controlled with a relatively small vocabulary and that the web data are more

[1] http://dumps.wikimedia.org/

diverse for the vocabulary usage. Thus it may be easier for the Wikipedia documents to be grouped to a specific category compared to the web page documents due to the vocabulary usage. In our research, this also motivates us to utilise the Wikipedia collection to model the web page documents into the Wikipedia categories rather than directly clustering the web page collection.

For the relevance judgement, we assume the user's clicks as the relevance for this user query to the target document. Using the clickthrough data to improve the web search has been applied in the previous reserach [Joachims, 2002; Dupret *et al.*, 2007]. In this way, our relevance judgement is biased to the top-ranked results from the original search engine (sogou.com). The search engine has implemented algorithms to rank documents from text similarity, page rank and many other factors. Thus those documents from important websites have high chance to be ranked in the top results in the search engine. Our algorithm aims to improve the ranks of these clicked documents from the top ranked results from the search engine.

Our experiments compare Wikipedia-based personalized method with the Okapi BM25 text retrieval method. Experimental results are shown in Table 5.6. Our experiments include three runs: Okapi BM25 as the Baseline Run, historical query model Run (user modelling using the user historical queries), and historical click-through document model Run (user modelling using the user historical click-through documents). The baseline Run utilises the standard Okapi BM25 retrieval on the target collection; the historical query model Run first conducts the Okapi BM25 retrieval and then re-ranks the top 1000 results using the historical query based user model and the

Wikipedia based document models for these top-ranked documents; the historical click-through model Run re-ranks the same Okapi BM25 result using the click-through documents based user model and document models. From the results, the click-through document model improves the retrieval effectiveness, but the historical query model fails compared to the Baseline Run.

Table 5.6: Compare Wikipedia based personalized retrieval with Okapi BM25.

Runs	MAP		P@10	R-Prec	NDCG
Okapi BM25	0.0578	-	0.0639	0.0499	0.2124
Query Model	0.0272	-52.94%	0.0506	0.0336	0.1648
Click-Through Model	**0.1180**	+41.52%	**0.0699**	**0.1141**	**0.2568**

The results show that user modelling based on user queries is not effective for the personalized search task. This is because that the queries contain less information, which is not sufficient to record the user search interests. Using the queries to search the Wikipedia retrieves Wikipedia documents but these documents may not be relevant to the user's search interests. The click-through documents are exactly the web documents which have been clicked by the users. User modelling based on the click-through documents proves to be effective in our experiments.

For the result of the Okapi BM25 run in our experiments, the retrieval effectiveness is relatively low since our judgement results are built on the user's click results from the search engine. Usually, users only click the top ranked documents produced by the search engine. So the user's click documents have a high preference for the top ranked documents from the search engine. General search engine ranks documents not only by text similarities but also by other factors such as the importance of the documents in the web

collection. Thus, the reason why Okapi BM25 method gives lower ranks to the user's clicked documents in our experiments can be explained since this method does not consider other factors into account.

Based on the Okapi BM25 results, our click-through model re-ranks the Okapi results by user modeling. In this run, the relevant results (the user's clicked documents) get better positions compared to the Okapi BM25 method. It can be explained that the relevant documents which are more similar to the user search interests in user models get the better positions. And those documents with high text similarities but not similar to the user search interests get the lower position in our re-ranking method. Then it indicates that our proposed user modeling method can be helpful to improve the ranks for those documents similar to the user's search interests.

In the click-through documents based method, we adjust the coefficient λ in Equation 5.3 to obtain the results shown in Figure 5.3. These results show that setting the coefficient to 2 achieves the best retrieval effectiveness. Further increasing the coefficient gives no additional improvement. The results show that the user modelling component plays a more significant role in Equation 5.3.

5.3.1 Per-topic Analysis

Figure 5.4 shows the difference in MAP for the experimental results. This compares the baseline Okapi model run with the click-through model based run. The results show that for the 80 queries, 25 queries get worse results after using the click-through user model and for 7 queries the results do not

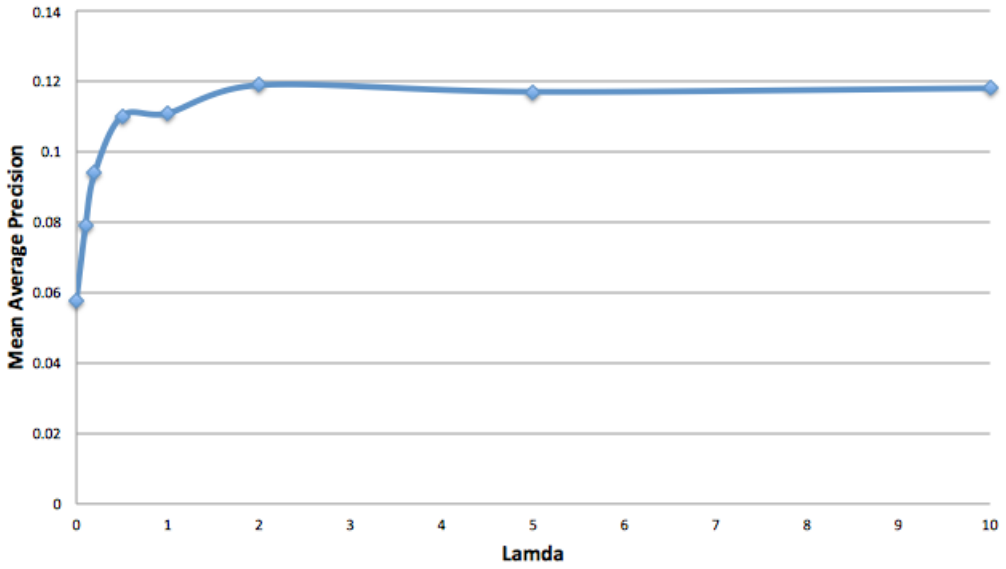

Figure 5.3: The results in MAP of different λ settings for personalized search.

change, for 48 queries the results get improvement (60%). We also compare the click-through based method and query model based method, the MAP difference for these runs is shown in Figure 5.5. The results show that for the 80 queries, 26 queries get worse results in click-through model than in query model and for 14 queries the results do not change, for 40 queries the results get better improvement (50%).

5.3.2 Discussion

Using external resources for personalized search offers a potential way of addressing the sparse data problem. To provide personalized search results for users, essential resources such as user logs are needed to model the user search interests. The historical data in the log is usually not sufficient to record all the user information. The assumption in this process is that if the user has

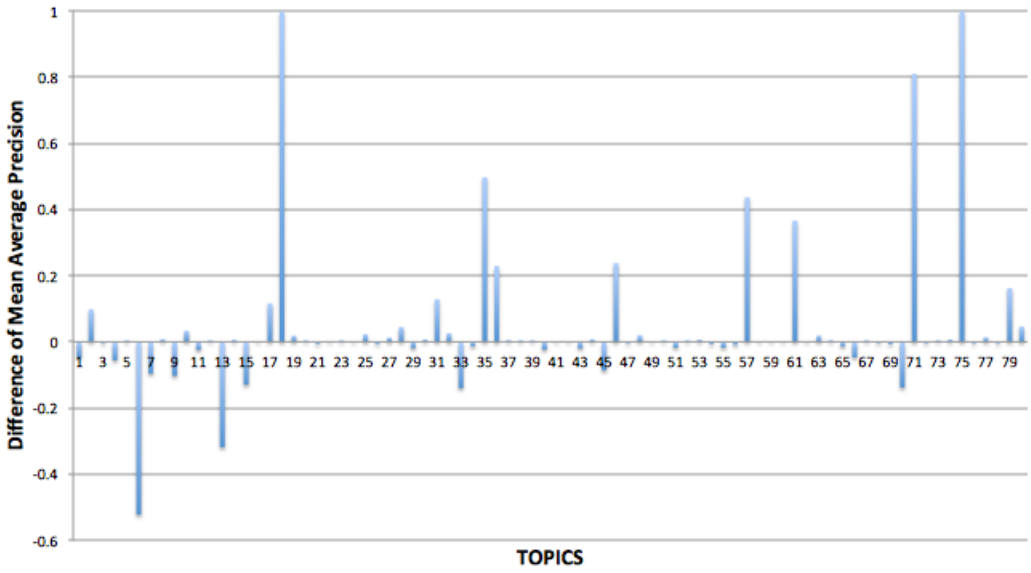

Figure 5.4: Comparison in MAP between click-through model and Okapi method.

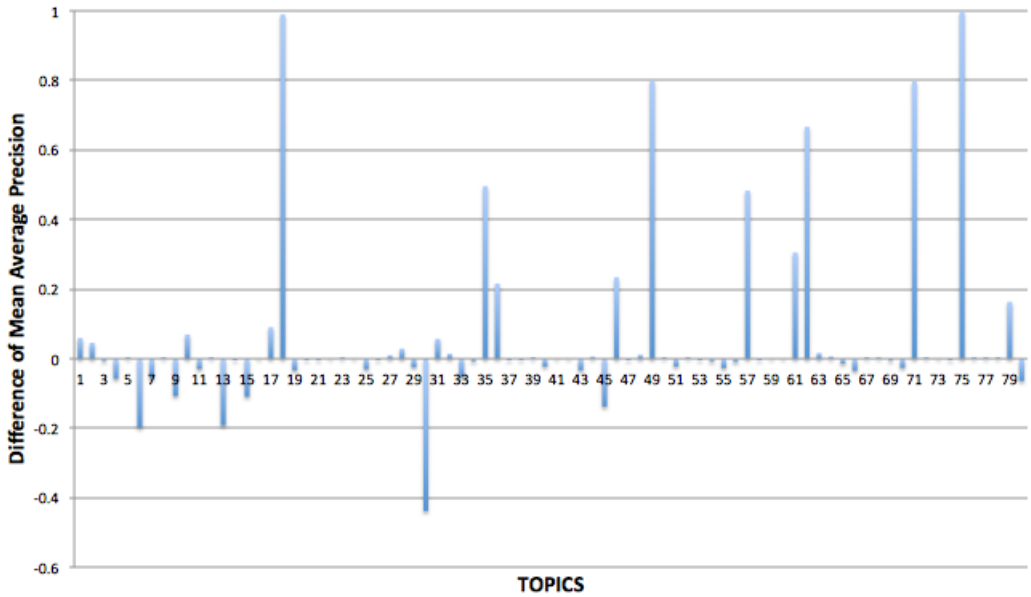

Figure 5.5: Comparison in MAP between click-through model and query model.

issued a query or clicked a document in a category, there is a good chance that they will issue a query or click a document of the same category in their later search activities. In this process, how to model user search interests into categories is a challenge. In the research of this chapter, external resources are utilized to model the user's historical data into categories. This is essentially enriching the user information using external knowledge. The experimental results show improvement in user modelling when using external resources method compared to the baseline run without personalized modelling.

In the experimental results, we also find that the historical queries based user modelling method does not perform as well as the click-through document model. This is usually because the historical queries are typically short and ambiguous. It is difficult to map these queries into categories. Thus the user models based on these queries do not help to provide personalized search results. The click-through documents are usually long-length documents which contain more information than the user queries. These documents can be mapped into categories which are used to build user models. This forms a good basis for later personalized re-ranking experiments and our experimental results show its effectiveness. In the parameter setting experiments for the click-through based model, the results show that the retrieval effectiveness improves with increased importance of user modelling. This demonstrates the effectiveness of user modelling from external resources for this personalized search task.

In our experiments, we use the click-through documents as the substitute for the relevant documents to the user query. This is due to that in the web search task, usually it is difficult to collect relevant judgment for user query,

especially for the personalized search task. Improving the user clicks can still help to improve the user satisfaction for the web search since the click-through rate is also an important metric to evaluate the quality of the search engine [1]. Further experiments using human judgement can be useful to validate our method based on external resources. Also in this chapter, our baseline is based on the Okapi method which is different with the original ranked list from the search engine. In the next chapter, we evaluate our method in learning to rank framework which simulates the original ranking function in a more similar way.

5.4 Summary

In this chapter, we have described a study of utilizing a Wikipedia-based user modelling to re-rank text retrieval results. The Wikipedia-based user modelling method consists of user models for user search interests and document models for the underlying topics of the target document set. This method is essentially enriching the missing information about the user and document sides by external resources in a personalized search task. The experimental results show that the user's click-through documents have the potential to model their search interests well and that it can help the user to get better retrieval results in the future search activities. This shows that Wikipedia-based user modelling is a promising direction to explore for personalized retrieval. In our experiments, the click-through based model outperforms the historical queries based model due to the click-through documents contain more

[1] https://googleblog.blogspot.jp/2008/09/search-evaluation-at-google.html

information about the user than the short and ambiguous user queries.

To answer the research questions in the beginning of the chapter, we get the conclusions as follows:

- How to utilize widely available external resources for user modeling in personalization? In this chapter, we propose a clustering based method to classify the external resources into categories, thus the user historical data can be mapped into these categories to build user models.

- Is there a simple and effective solution to utilize external resources for general personalized web data search task? The external resources based user models are used to compare with the target documents by topical relevance in our experiments, and then the topical relevance is used to rank the target documents for different users.

Chapter 6

Exploring External Resources in Learning to Rank

In our earlier research, we applied external resources to classical information retrieval (IR) techniques, such as query expansion and document expansion. Our results showed the effectiveness of using external resources in short document retrieval tasks. In our later research, we investigated the utilization of external resources for user modelling in personalized search task. Our results show the external resources can be applied in the process of building user models. The combination of user modelling using external resources from user clickthrough data with a text based similarity method shows improvement compared to use of only the text based similarity method for the personalized search task. In this chapter, we further investigate methods of user modelling from external resources in a learning to rank framework. The purpose of this research is to investigate the effectiveness of utilizing external resources in a learning to rank framework since it represents a state of the art

of ranking method in the modern industrial search applications.

Previous research in IR has shown the potential of personalisation for improving retrieval effectiveness in current search system [Pretschner & Gauch, 1999; Jeh & Widom, 2003; Liu *et al.*, 2004; Speretta & Gauch, 2005; Dou *et al.*, 2007]. Personalized search research focuses on the utilization of user data for building the user search interests model. The user data may include user historical clickthrough data, user historical queries, a user profile consisting of user-defined interest topics, and search interests gathered from the user's friends in a social network. Based our research in chapter 5, we utilise the user's historical clickthrough documents for building user models in a learning to rank retrieval system.

Historical search logs generally contain information relating to the interests of the user based on their previous search activity. This search history typically contains the user's unique id, their search queries, and the identities of the documents they clicked after issuing each query. This is useful information to track the user's search activity, but most importantly it is valuable for capturing the user's search interests. User search interests can be inferred from the user's search log data using various algorithms to construct models of their interests. One less considered problem in this process is the sparse data problem in user data which leads the constructed user models can not capture the full user search interests. From our experience in the previous research, the external resources can be a good supplement for the sparse data problem in IR research. Our method to build user models from Wikipedia categories has been shown to be effectiveness in a personalized search task. In this chapter, we utilise external resources in the process of building user

models by using knowledge contained in them.

In a typical personalized retrieval method, the user models are constructed from historical log data. The user model is then incorporated into the retrieval process to generate personalized search results for the individual user. In this process, the user historical data is the most important information resource to capture the user search interests. Possible circumstances where the problems arise include: commercial search systems that do not store the full historical data from their users due to privacy constraints; a search user is beginning to be interested in a new topic which is not in their historical search log; and new topics that emerge online which are not be captured by the historical search logs. These scenarios illustrate situations where search logs are not sufficient to represent the full extent of a user's topics of search interest. We refer to topics that are of interest to a user, but are not covered by search logs as *hidden topics*. In this chapter, we analyse the problem of insufficient data in user historical data and propose a method of improving search for hidden topics by enriching the user logs data from external resources - in our case, a web data collection.

We utilise Latent Dirichlet Allocation (LDA) [Blei *et al.*, 2003] to build user models from the user's historical data in this work. LDA is a widely used topic modelling method for text analysis. LDA models a document by viewing it as a mixture of topics. These topics can be used as the search interests of the search users. The purpose of personalized search is to achieve personal relevance (the relevance to describe the degree of the content of the target document to satisfy the specific search user's interest in a topics) between user historical data and the target web documents. In simple terms, personal

relevance can be computed as the similarity between the topics of the user data and the target documents. LDA can be applied to build the topic models for user historical data and target web documents. Personal relevance can then be calculated by finding the similarity between the topic models derived from user historical data and target web documents. However, as pointed out above, the log data may not provide sufficient information to cover the user's search interests. This motivates us to propose a method to expand the user data from external resources in the process of user modelling to resolve this problem. Thus the expanded user models can cover more topics of potential interest to this user, and make the personal relevance between the user and the target documents better.

In this chapter, we focus on several aspects of the personalized search task: identifying the problem of insufficient data in user logs; building user models for existing user topics in log data; expanding user models to cover potential hidden topics for users; and utilizing the expanded user models to personalise search results for user queries in a learning to rank framework. Our experiments investigate the insufficient data problem of user log data using a commercial search log archive.

The research questions we are addressing in this chapter are listed as:

- How to model the user and document in topic modeling framework for personalized search task?
- How to utilize external resources in user modeling and document modeling?
- How to utilize external resources based user models to rank documents

in personalized search task in learning to rank framework?

This chapter is structured as follows: Section 6.1 overviews the background and related work to our investigation, Section 6.2 introduces the insufficient data problem in user log data, Section 6.3 describes our user modeling method from external resources, Section 6.4 evaluates our proposed method, and Section 6.5 summaries our research work on personalized search by expanding user models from external resources.

6.1 Background and Related Work

Personalized search aims to provide a different personalized ranking of retrieved items to each individual user of a search system. The motivation for this approach is that different users, even when using the same query, relevant material may be individual to the specific user. Nowadays, many Internet services, such as Google, Bing, Yahoo, Netflix, Amazon provides personalized search functions to users. It has been shown that personalization can increase the user's satisfaction, and bring benefits both for users and on-line services. A successful example of industrial application of personalized search can be seen in [Das *et al.*, 2007]. The benefits of personalization in a search system can be described as:

- Users get more relevant information to one's search interests in the refined top ranked results from personalized search.
- Users spend less time locating relevant information within personalized search results.

- Online services get more clicks by providing personalized search results since the users prefer to click only the top ranked results.
- Online services save computing cost by providing more relevant results to users using the same computing resources.
- Better search experience helps to retain the users to continue to use the online service when they are provided with the personalized search results.

Our research aims to resolve the insufficent data problem in the user's historical log data by building new user search models. We utilize a topic modeling method to build user models. A basic way to acquire personal relevance information is by using the similarities between the topic models of the user's historical data and those within the target documents. In our research, different methods of computing the personal relevance are used as features in a learning to rank framework. Learning to rank framework is state-of-the-art approach in the current search system online. Testing these new methods in a learning to rank framework demonstrates approach in more generalized real industrial search applications. Learning to rank is typically useful in online search systems since they typically contain many factors which affect the ranking of the target documents. Learning to rank methods have been proven to be an effective way in combining many factors for online search [Liu, 2009].

In the following parts of this section, we introduce the LDA method for topic modeling and Ranking SVM method for learning to rank which are utilized in our research.

6.1.1 LDA for Topic Modeling

Topic modeling has gained significant attention in the machine learning and IR research community [Blei *et al.*, 2003; Wei & Croft, 2006]. LDA is one of the most widely used topic modeling algorithms [Blei *et al.*, 2003]. LDA is a three-level hierarchical Bayesian model, in which each item of a collection is modeled as a finite mixture over an underlying set of topics. Each topic is modeled as an infinite mixture over an underlying set of topic probabilities. In text modeling, the topic probabilities provide an explicit representation of a document. LDA assumes the following generative process for each document i in a corpus D:

1. Choose $\theta_i \sim \text{Dir}(\alpha)$, where $i \in \{1, ..., M\}$ and $\text{Dir}(\alpha)$ is the Dirichlet distribution for parameter α
2. Choose $\phi_k \sim \text{Dir}(\beta)$, where $k \in \{1, ..., K\}$.
3. For each of the words w_{ij}, where $j \in \{1, ..., N_i\}$
   (a) Choose a topic $z_{i,j} \sim \text{Multinomial}(\theta_i)$.
   (b) Choose a word $w_{i,j} \sim \text{Multinomial}(\phi_{z_{i,j}})$.

In the above generative process, α is the parameter of the Dirichlet prior on the per-document topic distributions, β is the parameter of Dirichlet prior on the per-topic word distribution, θ_i is the topic distribution for document i, ϕ_k is the word distribution for topic k, z_{ij} is the topic for the jth word in document i, and w_{ij} is the specific word. In LDA, the dimensionality k of the Dirichlet distribution (and thus the dimensionality of the topic variable z) is

assumed known and fixed. The word probabilities are parameterized by a $k \times V$ matrix β where $\beta_{ij} = p(w^j = 1|z^i = 1)$, which was treated as a fixed quantity that is to be estimated in the training process.

The graph model of LDA is shown in Figure 6.1. To utilize LDA on a specific corpus it is necessary to estimate the parameters of the model. In this research, we utilize Gibbs sampling introduced in [Griffiths & Steyvers, 2004] for parameter estimation. Given a document d in a web corpus and topics z_i in LDA, LDA provides the probability of topics of a given document as $P(z_i|d)$, $1 \leq i \leq K$. The explicit representation of topic probabilities can be used to represent the document using LDA. Thus using LDA modeling each document can be presented as a k dimension vector, where k is the number of topics when training the LDA model and the value of the vector element k is the probability that the document belongs to topic k.

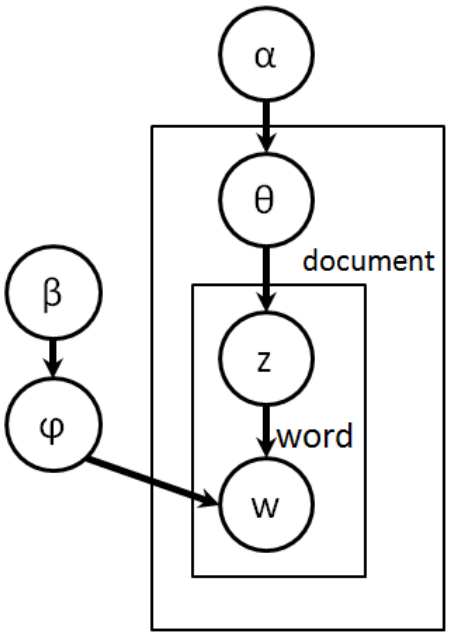

Figure 6.1: The LDA topic model.

Beyond standard LDA, several variations have been proposed in the topic modelling literature, including the Correlated Topic Model (CTM) [Blei & Lafferty, 2006a] and the Dynamic Topic Model (DTM) [Blei & Lafferty, 2006b]. The CTM differs from LDA by replacing the Dirichlet distribution of topics from the corpus by a logistical normal distribution. Logistical normal distribution performs a logistic transformation on the multinomial normal distribution to relax the independence constraints of a Dirichlet distribution. Thus, the CTM can produce a generative process under the assumption that topics are correlated with each other. The DTM focuses on the modelling of the temporal information in the documents. The purpose of DTM is to model the time evolution of topics. It uses state space models on the natural parameters of the multinomial distributions that represent the topics.

Apart from theoretic research on topic modelling, topic modelling has been widely applied in natural language processing applications such as classification and clustering algorithms [Lacoste-Julien *et al.*, 2008]. LDA is also used in document ranking in IR [Wei & Croft, 2006]. In this work, LDA-based similarity score between the query and document is linearly combined with a language model based similarity score. Reported results show that this LDA-based document retrieval model outperforms the relevance language model [Lavrenko & Croft, 2001].

The application of topic modelling to personalized search is a new research area in recent years. A new topic model including the user and log information into the generative process was proposed in [Carman *et al.*, 2010] for personal web search task using the AOL search log data. However, their results do not achieve an improvement compared to an LDA baseline without

user and log information. [Song *et al.*, 2010] focuses on sorting the relevant and irrelevant parts of user logs to optimize search personalization for a personalized search task in a self-built search system. The idea is to build a topic model using the user search logs, and to update the current query model with a topic close to the query in a KL-divergence retrieval model. Similar to our work, this work models the user historical data by topic modelling. However their work updates the user query model with the historical user data, while our proposed method works at another level to update the user topic model from external resources. Additionally, they are more focused on query classification relating to historical user topics, while our research concentrates on the matching of hidden topics in personalized search. [David *et al.*, 2012] proposes a generative model which includes the user, query, and document information. The findings of this work demonstrate gains in retrieval performance for queries with high ambiguity, and show particularly large improvements for acronym queries. The evaluation of the proposed method is based on a web search task in the log data from a major search engine.

6.1.2 Ranking SVM for Learning to Rank

Learning to rank is an application of machine learning techniques in IR. It has been successfully applied in commercial search engine systems such as Bing and the search platform of many other Internet services. The main idea of learning to rank is that since in many search applications, many different factors affect the potential relevance of the target web documents such as the personal relevance between the query user and the target web documents, the

text similarity between the queries and documents, the significance of the target documents themselves, etc. Learning to rank methods aim to combine all these factors as features in a ranking method. Like any supervised machine learning task, learning to rank methods need training data to learn the optimal ranking model for a specific task. A typical training data set for web data search task contains the user queries, the target documents, and the value of each feature for the query and the target documents. An example of learning to rank research dataset is the Microsoft learning to rank dataset [1]. This dataset contains 136 features for the queries and the target documents. For each query/document pair, some example features in a web data search task are as:

- the number of query terms that the document contains
- the ratio of the number of query terms contained in the document compared to the number of all query terms
- the length of the document
- the sum of IDF scores of all document terms in the document
- Okapi BM25 score between the query and the document
- language model IR score using an absolute smoothing method between the query and the document
- PageRank score of the document

[1] http://research.microsoft.com/en-us/projects/mslr/

With all these feature scores and the matching score between query and document, the training data for all query document pairs in learning to rank can be represented as:

0 qid:1 1:3 2:0 3:2 4:2 ... 135:0 136:0

2 qid:1 1:3 2:3 3:0 4:0 ... 135:0 136:0

...

In the above training dataset, the first column is the relevance score of the query/document pair. The second column is the query id. The following columns are the feature scores of all these 136 features for this query/document pair. A typical learning to rank system is shown in Figure 6.2.

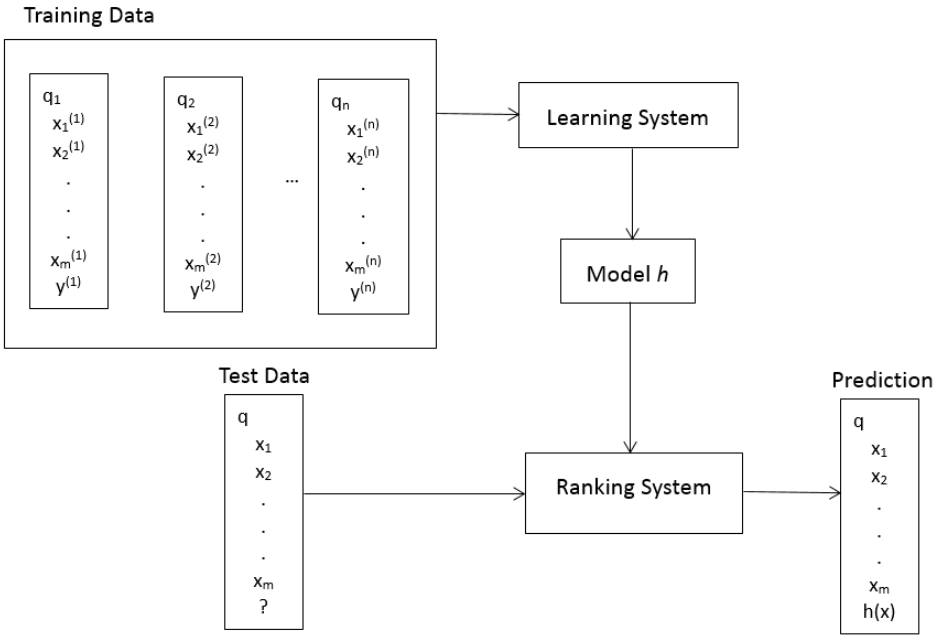

Figure 6.2: Example learning to rank framework.

As shown in Figure 6.2, the training data contains n query/document pairs. Each query/document pair contains m features, with the value of the

feature i ($1 \leq i \leq m$) given as x_i. For this query/document pair, the relevance score is given as y. In a learning to rank system, the retrieval model h is developed from the available training data in a training process. This retrieval model h is then utilized in the ranking system. For a new query/document pair with all the values of the m features without the relevance score y, the ranking system utilizes the ranking model h to produce the final relevance scores for the new query/document pairs. Thus for a candidate document set recalled by a query, a retrieval model $h(x)$ produces different scores for different query/document pairs which can be used to produce a ranked list. In a working search engine, the historical user logs are used to produce the ranking model h and h is utilized to rank the new documents recalled by the new user query.

In the framework of learning to rank, the core algorithm is the learning method used to produce the ranking model. From previous research, a number of learning methods have been explored for producing ranking models. Typical learning to rank methods can be categorized as pairwise [Joachims, 2002; Freund *et al.*, 2003; Burges *et al.*, 2005; Zheng *et al.*, 2007a; Cao *et al.*, 2006; feng Tsai *et al.*, 2006; Zheng *et al.*, 2007a,b; Jin *et al.*, 2008; Chen *et al.*, 2010], point-wise[Fuhr, 1989; Cooper *et al.*, 1992; Crammer & Singer, 2001; Li *et al.*, 2008; Sculley, 2010] and list-wise[Xu & Li, 2007; Cao *et al.*, 2007a,b; Yue *et al.*, 2007; Qin *et al.*, 2008; Xia *et al.*, 2008; Xu *et al.*, 2008a; Taylor *et al.*, 2008]. A detailed introduction describing the various learning to rank methods can be found in [Liu, 2009]. The three approaches are categorized by how many documents are used to calculate the loss each time in the training process. The pointwise approach uses each document to calculate the loss (the differ-

ence between the predicted relevance score for target document and the true ground score for this document in the training collection) and the overall loss is summed from all the target documents in the training data; the pairwise approach uses a pair of documents to calculate the loss and the overall loss is summed from all pairs of documents of the same ranked list in the training data; the listwise approach calculate the loss from the list of the documents and the overall loss is summed from all the ranked lists in the training data. Currently there is no theoretical proof to show which approach is better, and our selection of a ranking method is based on the experimental results in previous search tasks [Qin *et al.*, 2010].

In our research, we utilized a learning to rank method called Ranking SVM. This is widely used for research purpose and shows very effective results in many learning to rank applications [Joachims, 2002; Cao *et al.*, 2006]. Ranking SVM has been proven to produce state-of-the-art results for standard learning to rank datasets such as LETOR [Qin *et al.*, 2010].

Ranking SVM utilizes the user's clicks to indicate that these clicked documents are more likely to be relevant to the user than the non-clicked documents. Thus in our research, while web documents clicked by users are marked by a relevance score 1 while the non-clicked documents are marked by a relevance score 0. This setting is assumed to reflect the user's judgments of document relevance for a specific query. Of course, a relevance set formed in this way will be noisy or incomplete. The user may click on a document in error based on a misleading document snippet summary in the SERP which suggests that a document is relevant when it is not, or the user may cease clicking items once their information need is satisfied without clicking all of

the visible relevant items or relevant items may appear below the rank of documents checked by the user. However, the clicked documents are clearly of interest to the user, even if they are not ultimately found to be relevant or do not represent the full relevance set, and the impact on their ranks when retrieved is taken in our work to correlate to user satisfaction when exploring the contents of a search engine SERP produced in response to their query.

For a user query, the Ranking SVM method utilizes Kendall's τ to compare the ranking sequences and the true ground sequences, where the relevant documents are all ranked before the non-relevant documents. Kendall's τ is defined as shown in Equation 6.1, where n is the number of documents in a ranked list. If the ranking sequence is exactly same as the true ground sequence, Kendall's τ gets its highest value of 1.

$$\tau = \frac{(number of concordant pairs) - (number of discordant pairs)}{n(n-1)/2} \tag{6.1}$$

It has been proved that Kendall's τ is related to the average precision, and it has been demonstrated that maximizing Kendall's τ is connected to improve retrieval quality. The proof of this conclusion can be seen in [Joachims, 2002]. Thus the goal of the ranking function is to maximize the expected value of Kendall's τ. Given a training sample S of size n containing queries q with their target ranking r: $(q_1, r_1), (q_2, r_2), ...(q_n, r_n)$

The learner L selects a ranking function f from a family of ranking func-

tions F that maximizes the empirical τ on the training sample:

$$\tau_S(f) = \frac{1}{n}\sum_{i=1}^{n}\tau(r_{f(q_i)}, r_i^*) \tag{6.2}$$

r_i^* is the true ground sequence for query q_i.

The target is to design an algorithm and a family of ranking functions F so that finding the function f maximizing is efficient, and that this function generalizes well beyond the training data. Consider the class of linear ranking functions shown in Equation 6.3.

$$(d_i, d_j) \in f_{\vec{w}(q)} \Leftrightarrow \vec{w}\Phi(q, d_i) > \vec{w}\Phi(q, d_j) \tag{6.3}$$

where $\vec{w}$ is a weight vector that is adjusted by learning, and $\Phi(q, d)$ is a mapping onto features that describe the match between query q and document d. Thus the task of making the ranking of document pairs the same of the true ground sequence is changed into the task of finding the right parameters to satisfy the right side of Equation 6.3 .

Figure 6.3 illustrates how the weight vector w determines the ordering of four points in a two dimensional example. For any weight vector w, the points are ordered by their projection onto w. This means that for w_1 the points are ordered $(1, 2, 3, 4)$ while w_2 implies the ordering $(2, 3, 1, 4)$.

For the class of linear ranking functions, this is equivalent to finding the weight vector so that the maximum number of the following inequalities is fulfilled:

$$\forall (d_i, d_j) \in r_1^* : \vec{w}\Phi(q_1, d_i) > \vec{w}\Phi(q_1, d_j)$$

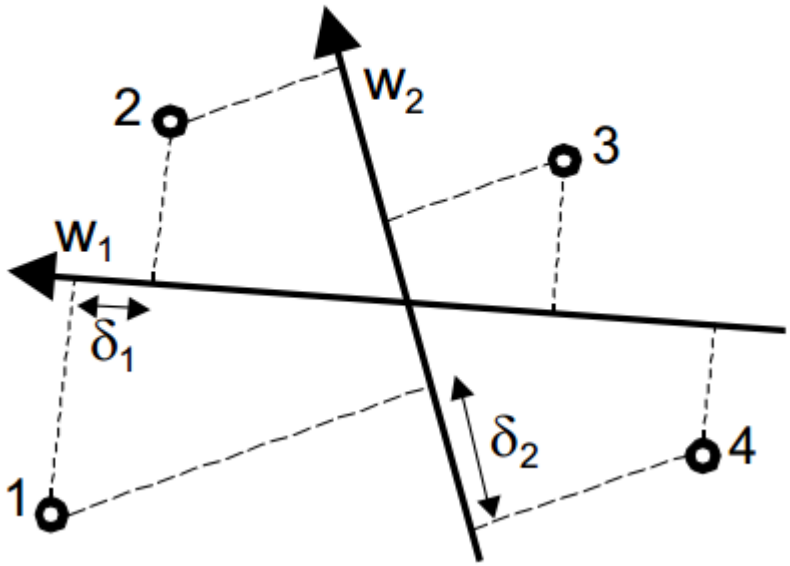

Figure 6.3: Example of how two weight vector rank four points.

...

$$\forall(d_i, d_j) \in r_n^* : \vec{w}\Phi(q_n, d_i) > \vec{w}\Phi(q_n, d_j)$$

A analysis of this result shows that the problem is NP-hard. However, just like in classification SVMs, it is possible to approximate solution by introducing (non-negative) slack variables ξ and minimizing the upper bound. Adding SVM regularization for margin maximization to the objective leads to the following optimization problem. Ranking SVM transforms the ranking problem into an optimisation problem shown in Equation 6.4.

$$minimize : V(\vec{\omega}, \vec{\xi}) = \frac{1}{2}\vec{\omega} \cdot \vec{\omega} + C \sum \xi_{i,j,k} \tag{6.4}$$

subject to:

$$\forall(d_i, d_j) \in r_1^* : \vec{w}\Phi(q_1, d_i) \geq \vec{w}\Phi(q_1, d_j) + 1 - \xi_{i,j,1}$$

...

$$\forall(d_i, d_j) \in r_n^* : \vec{w}\Phi(q_n, d_i) \geq \vec{w}\Phi(q_n, d_j) + 1 - \xi_{i,j,n}$$

$$\forall i \forall j \forall k : \xi_{i,j,k} \geq 0$$

r^* is the target rankings, $\vec{w}$ is a weight vector that is adjusted by learning. $\Phi(q, d)$ is a mapping onto features that describe the match between query q and document d like score from Okapi BM25, and C is a parameter that allows trading-off margin size against training error, and $\xi_{i,j,k}$ are non-negative slack variables.

In our research, Ranking SVM is utilized as the framework to combine the ranking factors in the personalized search task. In the following sections, we analyse the insufficient data problem in user data and introduce our solutions.

6.2 Topic Modelling on Web Corpus

To analyze the sparse data problem in user historical data, we apply topic modelling on the web documents including the user historical click-through documents. In our research, the user historical data is from a month's user logs in a Chinese web search engine. This collection is the aggregation of

all the user's click-through documents in a month. The collecion is used as the target corpus in our personalized search task and all the user's historical click-through documents in a month are included in this dataset. The documents are extracted from an 130 million web pages by the user's clicked urls in the log data. These web pages are crawled from the simplified Chinese Internet websites by a commercial search engine. In our experiments, only the web pages clicked by the users of the search engine in the month of the logging time are kept for the retrieval task. In this way, our experiments can be controlled in a reasonable scale for research purposes.

LDA models a document into topics where each topic consistes of terms with probabilities. Sample topics with significant keywords belonging to the topics are shown as Table 6.1[1]. These results are generated with a $K = 10$ LDA topic model. In parameter settings of LDA, smaller values of K produce broader topics, while larger values of K gives the narrower topics of the corpus. In Table 6.1, the labels of the topics are manually created, since LDA does not produce the text labels for topics. In the LDA model, each term belongs to multiple topics with different probabilities $(0 < p < 1)$. Thus in Table 6.1, the top terms do not belong to the corresponding topic only. The results in Table 6.1 mean these words have a high probability of belonging to these topics and the documents with these words have a high probability of belonging to these topics.

From the results of topic modeling for the Chinese web corpus, we find that the LDA models classifies the simplified Chinese web documents into

[1]For reader's convenience, the terms are translated from simplified Chinese into English manually.

Table 6.1: Sample Top Words from Topics in LDA.

Topic	Label	Top 5 Keywords
1	Entertainment	movie music free download tv
2	Computer	game software download play system
3	Education	university major department school exam
4	Location	beijing shanghai china guangdong nanjing
5	Geomancy	predict constellation lottery divination character
6	Economic	management work enterprise build fund
7	Company	company product enginer technology design
8	Name	li zhang wang liu chen
9	Number	one month day time year
10	Food	fish vegetable soup health medicine

meaningful topics. This demonstrates that topic modeling can be used to describe the user's historical data, and that this user data can be explained as belonging to meaningful topics using topic modeling.

6.2.1 Topic Change in Search Log

In this section, we illustrate the topics present in our user log data. By illustrating the topics contained in the user's historical data, we observe the change of user's search interests in the framework of topic modeling. If the user's search topics change with time, it demonstrates that modeling the current user's historical data may not be good to cover the user's search topics.

We train an LDA model with 10 topics on the target corpus. We set the parameter as 10 for easy observation of the topic change in our experiments since less topics are too broad to model the topics and more topics are hard to observe the results. Using LDA, each document is associated with a topic distribution and we call this distribution as document topic model M_d.

We track the overall change in the user topic distributions in the user log

data. Each day the user log contains t click-through documents <d_1, ..., d_t>. Each document d_i ($1 \leq i \leq t$) has a document topic model M_{d_i}. We track each topic z_j ($1 \leq j \leq 20$) using Equation 6.5. In Equation 6.5, $p(z_j|d_i)$ can be estimated from LDA inference as introduced in section 6.1. The $topicScore$ is used as a quantity indicator of how often this topic is visited on an individual day. These topic counts change with time, and we plan to investigate how these topics change over one-month of log data.

$$topicScore_j = \sum_{i=1}^{t} p(z_j|d_i) \tag{6.5}$$

We show the topic change during one month's time in Figure 6.4. We use 10 topics as examples to show the change of the topic scores. In Figure 6.4, each line represents one topic in the web corpus. There are 10 lines in the figure with each representing one topic. The Figure gives the user's topic change during one month. From Figure 6.4, we can observe the following findings:

- The user's search interests change with the time.
- Different topics have different topic scores which indicate that some topics get more attention from the users and some do not.
- The trend of the changing of topics are different: some topics get more attention than other topics over time while others get less.

We further investigate the behavior of individual users in the search log. Given a user in the search log as an example, we show a sample topic distribution for one in 10 days period as Figure 6.5. The topic score is defined by

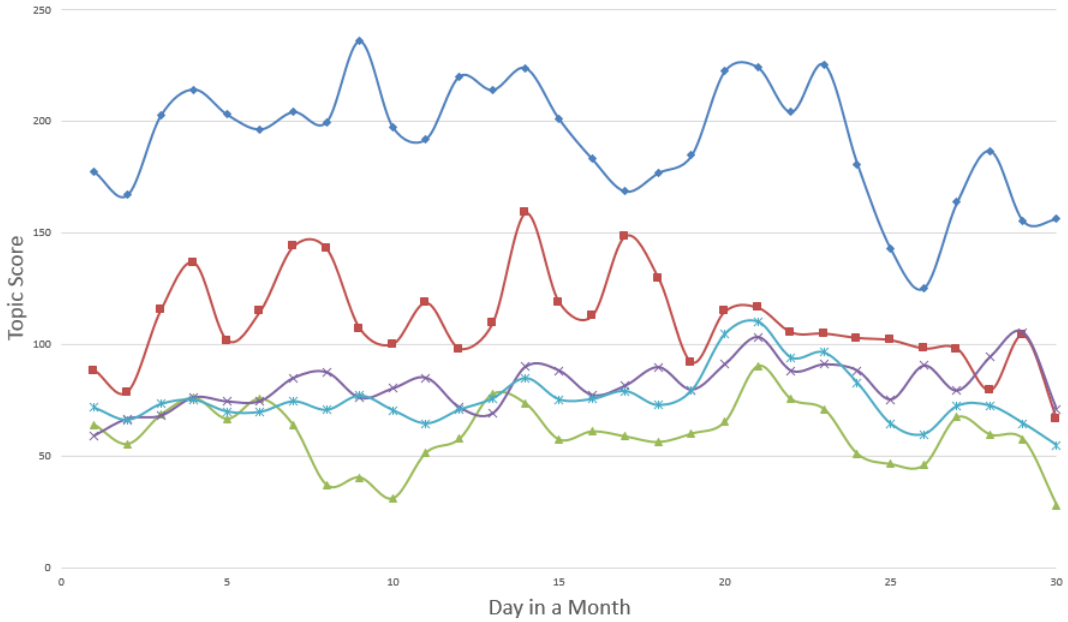

Figure 6.4: Topic change during one month for all users.

Equation 6.5 and the results in Figure 6.5 are based on the log data for this user. We observe the user's query log: in day one the user is interested in the topics of *entertainment* and *software*; after that, the user begins to be more interested in *education*, *location*, and *geomancy*.

In Figure 6.5, the data is extracted from the LDA modeling results (10 topics) and we remove 5 topics with a low score for this user (topic score is below than 0.1) and keep 5 topics. We remove the topics with low scores since they are not the main interested topics for this user. The challenge in this scenario for this particular user is that if the personalized algorithm records the user's search interests in day one such as *entertainment* and *software*, the system may not match the personalized results for *education* and *location* in the following days. If the system still considers the user's search interests as topics in day one only, it could harm the user's personalized search experience

in other topics. By analysing this particular user, we illustrate an example that the historical log data may not be sufficient to cover the potential user search topics. This can happen to any search user. This phenomenon motivates us to propose a new personalized user modelling method to cover potential topics which are not represented in the historical user log data.

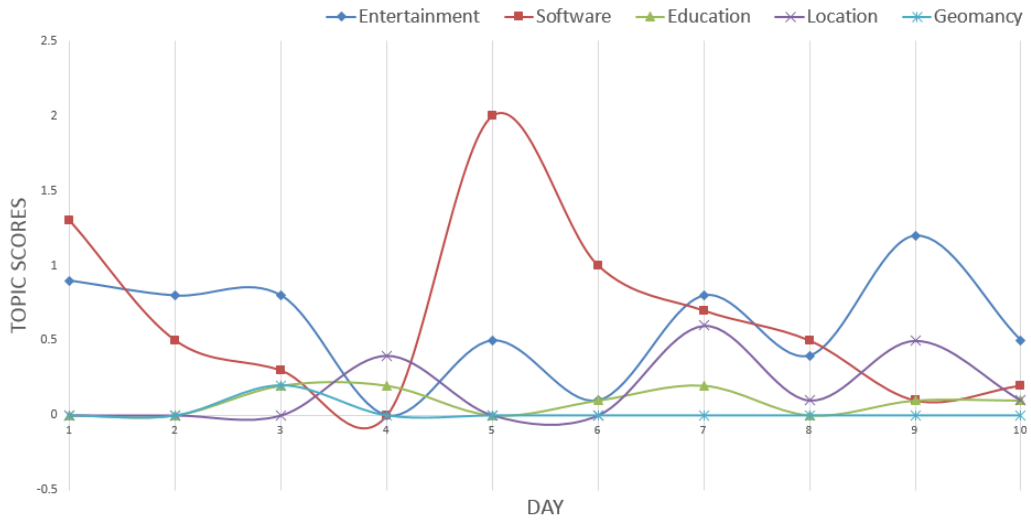

Figure 6.5: Topic change sample for one user.

In the following parts of this chapter, we describe a method to model the correlation of the topics and include potential topics which are not included in the historical user log in the personalized search framework. We hypothesis that the topics in external resources can be helpful to enrich the topics in the user logs, and that this can help the IR system to provide better personalized search results for new topics from the users.

6.3 External Resources for Personal Relevance

In this section, we introduce our method for the construction of user topic models and their use to compute personal relevance between the user and target web documents. Furthermore, we introduce a method of updating the user topic model with external resources for matching the hidden topics. Our approach relies on the user historical search log to capture the user's search interests. To build user topic models, we utilize the user's clickthrough documents, since their use in user modelling was shown to be effective in Chapter 5.

In our research, we utilize the topic distribution of the user's historical clickthrough documents by topic modelling using LDA to model their historical search interests. Given a historical clickthrough document from a user u, there exists a topic model on this document M_d including all $P(z_i|d)$ ($1 \leq i \leq K$, K is the number of topics in LDA). We combine all the topic models of a user's click-through documents into one topic model for this user. The sum of the probabilities of each topic in different clickthrough documents is divided by the number of clickthrough documents in the user logs to obtain the probability of this topic in the user model as shown Equation 6.6. In Equation 6.6, l is the number of click-through documents for this user. If one document is clicked by the same user twice, it is counted as two documents in l. Since in each user document d, the sum of $P(z_i|d)$ is 1, the sum of $P(z_i|u)$ is still 1 in Equation 6.6. We refer to this process of building user topic models

as user modelling in personalized search.

$$P(z_i|M_u) = \frac{\sum_{j=1}^{l} P(z_i|d_j)}{l} \tag{6.6}$$

In our research, we do not apply the previous user modelling methods such as using manually defined terms or website categories to record the user's search interests. We utilize the LDA to model the user's clickthrough documents and use the topic models of the user's clickthrough documents to model the user search interests. In the user models of search interests, the high score of probability of a topic $(P(z_i|M_u))$ in the models can be explained that the user has high interests to a topic in his past search activities. This method in user modelling has the following advantages compared to previous methods:

- Topic modelling is a useful tool of catching the semantic structure of the user's clickthrough documents into topics. Thus it can be utilized to record the user interest topics from the user historical data and promote the forthcoming search results in the same topics. In personalizing the user's serach results, giving the search results which belongs to the same topic as the past user's search interests is more reasonable than giving the same text content as before to the user.

- Topic modelling based method does not need to maintain the knowledge system such as the the web directory to record the user's search interests in our topic modelling based method. For building such a knowledge system to record the user's search interests, it takes human efforts to construct, maintain and update which is necessary for using it

effectively.

- For the topic modelling method, it is easy to include external resources in the process of user modeling rather than using only the user's historical clickthrough data. In our research, we utilize the external resources by updating the user search topics.

In the following subsections, we introduce our LDA based personal relevance in personalized search task.

6.3.1 Building User Models

As described in Section 6.3, we combine the document models of the user's click-through documents as the user model. To give an pratical example, we give some examples of clickthrough documents by urls for a user in our experimental log data as (user id: 008781065409879385) in Figure 6.6.

```
http://2004.sina.com.cn/star/liu_xiang/
http://2004.sina.com.cn/zt/liuxiang_tf/index.shtml
http://bbs.phoenixtv.com/fhbbs/viewtopic.php?t=1939549
http://book.sina.com.cn/excerpt/livlivsz/2006-03-27/1122198369.shtml
http://business.sohu.com/20040823/n221670684.shtml
http://eladies.sina.com.cn/s/2006/0613/1036256631.html
http://health.sohu.com/20060112/n241381385.shtml
http://index.sports.sohu.com/person/plist-1166.html
http://liuxiang.sports.cn/
http://liuxiangcn.com/
```

Figure 6.6: Example of a user's clicked urls.

For a user, the historical clickthrough documents are extracted from these urls. The user model is constructed from the topic models of these clickthrough documents. In our experiments, the user model are generated from topic modeling with 100 topics. The selection of number of topics as 100 has been successfully utlized in previous IR tasks[Wei & Croft, 2006]. The document model is from the topic modelling and the topic scores are generated from the Equation 6.6. We select several typical user models as examples shown in Figures 6.7, 6.8, 6.9, 6.10. From these examples, we can observe that:

- Users are generally interested in more than one topic as shown in Figure 6.7, 6.8, 6.9 and 6.10.
- Some users have diversified search interests, with their search interests focused on more than 3 topics as shown in Figure 6.7.
- Some users are particularly interested in one topic as shown in Figure 6.9.

The differing search interest patterns of these users indicates the potential of personalized search. These user models are utilized in our algorithm to achieve the personal relevance between users and documents.

6.3.2 LDA-based Personal Relevance

Before we compute the personal relevance between user historical data and the target search documents, an initial search run is carried out using a standard retireval model - Okapi BM25. For each top-ranked document in the initial ranked list, there exists a topic model M_d. For each user, one has a

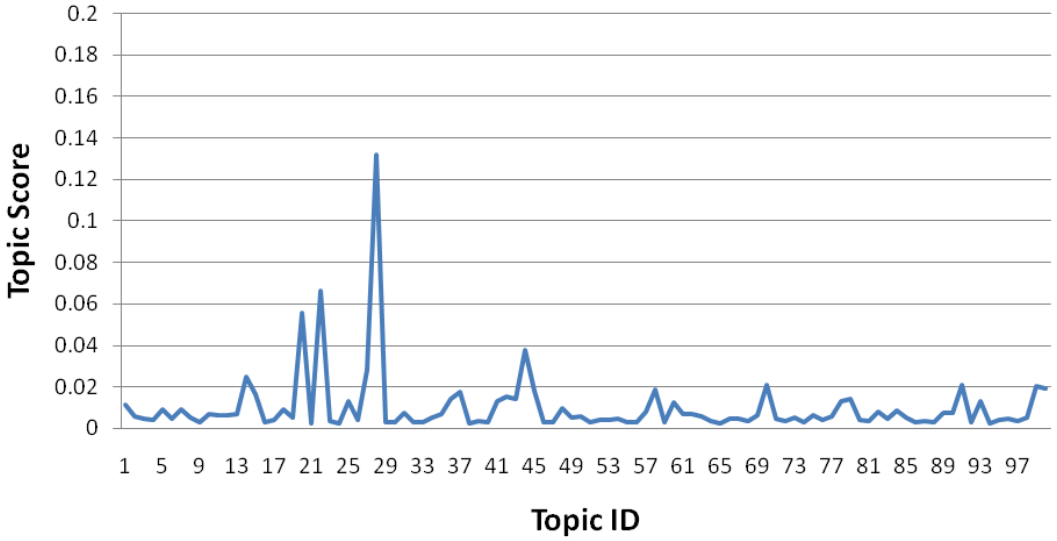

Figure 6.7: Example of user model (1).

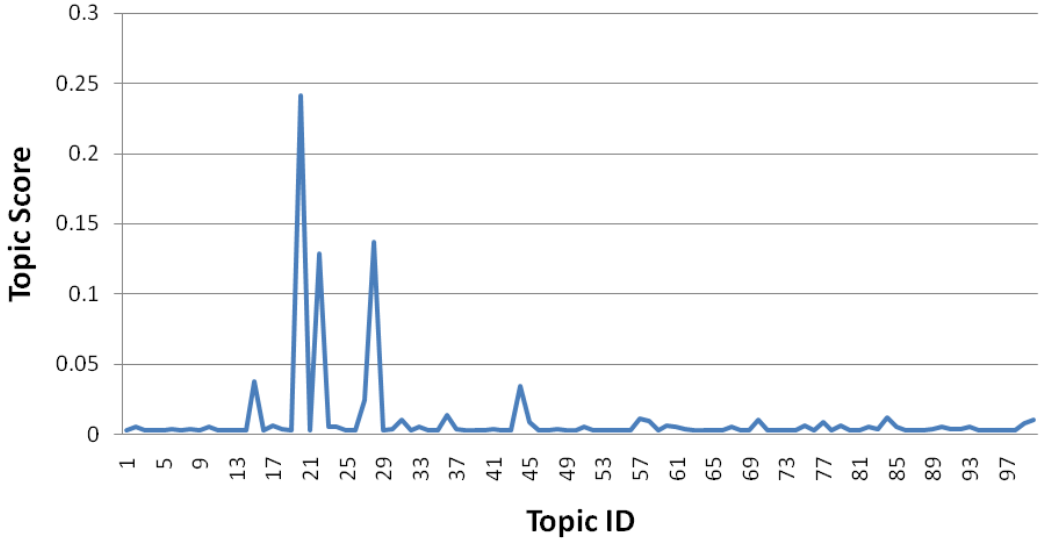

Figure 6.8: Example of user model (2).

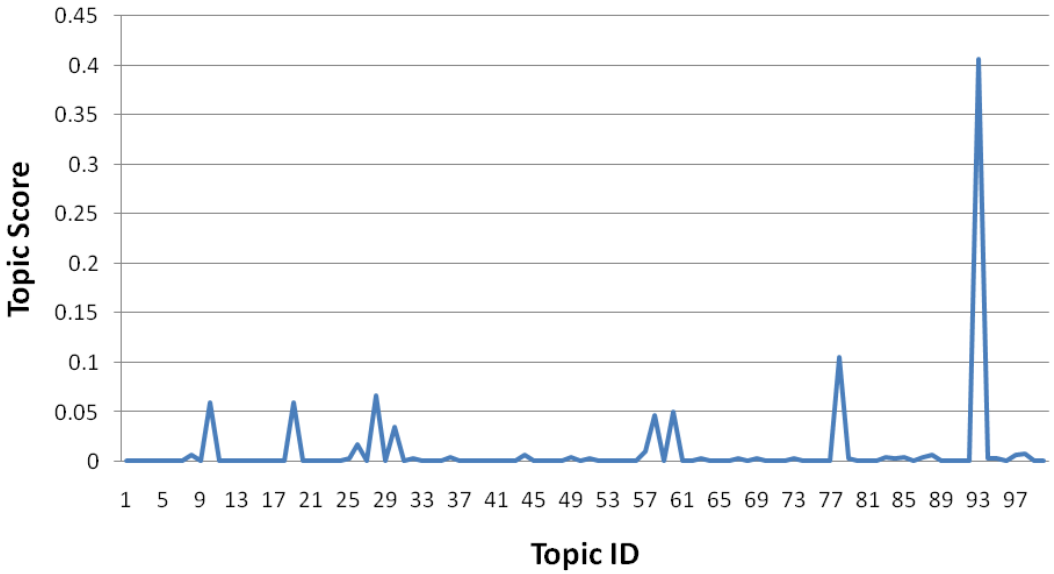

Figure 6.9: Example of user model (3).

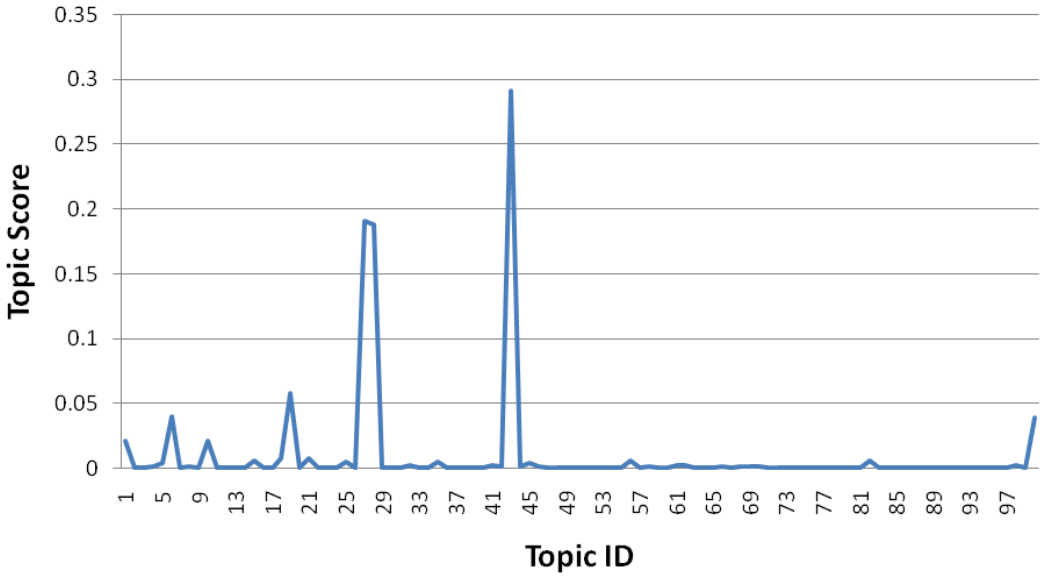

Figure 6.10: Example of user model (4).

user topic model M_u as defined in Equation 6.6. We utilize the similarity between these two models to measure the personal relevance between user and web documents. To compute the similarity between the user topic model and the document topic model, we use the Hellinger distance between topic models as shown in Equation 6.7. This has been utilized in correlated topic model [Blei & Lafferty, 2006a]. The Hellinger distance is designed to compare two probability distributions for similarity. Other popular similarity score as cosine simiarity is designed to compute the simiartiy between vectors with values. One advantage of Hellinger disantce utilized in computing document similaritiries represented by LDA topics is that it is a symmetric score for two documents while the other popular similarity score as KL-divergence is a non-symmetric score. The Hellinger distance of two topic M_1, M_2 is defined as shown in Equation 6.7. In Equation 6.7, the two document models contain k topics.

$$H(M_1, M_2) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{k} (\sqrt{P(z_i|M_1)} - \sqrt{P(z_i|M_2)})^2} \tag{6.7}$$

For each query and the documents retrieved from the initial retrieval model, we compute the Hellinger distance between the user topic model and document model of each retrieved document to measure the personal relevance for this query for this user and this target document. If the Helinger distance is low, it means the target document has very similar topics to the topics in the user's historical clickthrough documents; if it is high, it means that the target document has very different topics with the user's historical clickthrough documents.

Similar to other similarity scores between a user query and target web documents, the Hellinger distance between the user topic model and the topic model of the target web document can be used as a feature in a learning to rank framework to represent the personal relevance between the user and the target document.

6.3.3 LDA-Based Personal Relevance from External Resources

As we explained in Section 6.2, recording the user's historical log data is not sufficient to capture the user's future search interests. Thus using this recorded data to build user models is insufficient to compute the personal relevance when the user issues a query expressing a new search interests. To address this problem, we propose to expand the user topic model using external information resource. We chose to use a web corpus as the external resource since it covers a very wide range of topics. Our experience tells that one of the necessary conditions for the external resource is that it should contain the terms or topics relevant to the user queries. We construct a web corpus from the Simplified Chinese web corpus by randomly selecting documents from a large corpus. For a large web corpus, each document is given an index number from 1 to N (the number of documents in the large web corpus). When a random number (the range of this number is between 1 and N) is produced, the web document with this random number is selected as the document in the new collection. The process is stopped until the new collection contains $10,000$ documents. The selection of documents is for controlling the collection into a reasonable size for experiments.

We expand the historical user topic model with the topic model from the documents in the external web corpus. The new personal relevance is determined by caculating the similarity between the updated user model and topic model of target web documents.

For the external web documents, we compare the user model with the topic model of each document using the Hellinger distance as shown in Equation 6.7. We set the top t documents with lowest Hellinger distance with user model as the topical relevance feedback documents. All these top feedback documents form the external corpus with a feedback topic model M_{fd} as the method we build the user topic model M_u. We update the user model with the feedback topic model as shown in Equation 6.8. In our experiments, we set t as 10 since we do not want to include too much external information into the user models to change the user search interests. And also 10 has been shown to be a reasonable number for including external documents for query expansion in relevance feedback in our previous research.

$$p^{'}(z_i|M_u) = p(z_i|M_u) + \lambda * \frac{\sum_{j=0}^{t} p(z_i|M_{fd_j})}{t} \tag{6.8}$$

Since the sum of the $p^{'}(z_i|M_u)$ needs to be 1, $p^{'}(z_i|M_u)$ is normalized by the Equation 6.9. Thus $p^{''}(z_i|M_u)$ is used as the topic score in the user's updated search interest model.

$$p^{''}(z_i|M_u) = \frac{p^{'}(z_i|M_u)}{\sum_{i=0}^{k} p^{'}(z_i|M_u)} \tag{6.9}$$

With the new updated user topic model, the LDA-based personal relevance score is computed again to get the new relevance score between user

topic model and the target web documents using Equation 6.7. We refer to the new score as the LDA-based personal relevance score.

Furthermore, inspired by research in document expansion [Tao *et al.*, 2006] and length normalization [Robertson & Spärck Jones, 1994; Singhal *et al.*, 1998], we take into account the size of user data in the process of user model expansion. In previous document expansion work, the results show that short-length documents need more feedback information from external corpus to enrich themselves [Tao *et al.*, 2006]. In our work in Chapter 4, the document expansion method works well on short-length documents. Based on the previous investigation, we assume that the user data with little number of clickthrough documents needs more expansion information from external corpus, and the user data with large number of clickthrough documents need less expansion information from external corpus. To include the number of the user historical clickthrough documents in the log data into our method, we propose a size-based user model expansion method. We adapt the length normalization method from [Robertson & Spärck Jones, 1994; Singhal *et al.*, 1998] into our research. We modify Equation 6.8 of our user model expansion method into Equation 6.11.

$$p(z_i|M_{fd}) = \frac{\sum_{j=0}^{t} p(z_i|M_{fd_j})}{t} \tag{6.10}$$

$$p^{'}(z_i|M_u) = p(z_i|M_u) + \lambda * p(z_i|M_{fd}) * \frac{AvgSize}{Size} \tag{6.11}$$

$p^{'}(z_i|M_u)$ is then normalized using Equation 6.9 to satisfy the sum of the probabilities of user topics to be 1.

In Equation 6.11, $\frac{AvgSize}{Size}$ is used to normalize the size of user data which is the number of user click-through documents for the individual user in the historical log data; and $AvgSize$ is the average number of user click-through documents for the individual user. If the size of the user data is larger than the average size of the user data, the feedback part ($p(z_i|M_{fd})$) plays a lesser important role in the overall score; if the size of the user data is smaller than the average size of the user data, the feedback part ($p(z_i|M_{fd})$) plays a more important role in the overall socre.

6.3.4 Learning-based Retrieval Model

A standard method to achieve effective web search is to include a range of relevance clues or signals as features in a Learning to Rank framework [Liu, 2009]. In our experiments, we utilize the Ranking SVM toolkit [Joachims, 2006]. SVM-Rank is an efficient implementation software package for Support Vector Machine (SVM) ranking in IR [1]. Our LDA-based personal relevance scores can be used as features in the RankSVM framework.

For the Sogou data, we have the whole target web corpus and the user's search logs. An example of query log data can be shown in Talbe 6.2 [2]:

The features we are using in the experiments are the score of classical text retrieval model, initial ranked position from search engine, score of LDA-based personal relevance, score of LDA-based hidden topic match, score of LDA-based hidden topic match by size normalization. These features include

[1] https://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

[2] The query in Chinese has been translated into English manually for the reader's convience.

Table 6.2: An Example of Sogou Search Log.

00:00:00	time to record the log
2982199073774412	the user's unique id
360 safe guard	user query
8	the rank of the doucment in the rank list
3	the rank of the user has clicked on this document for the same query
http://download.it.com.cn	the url of the clicked document

the important factors for a typical web search task as text similarity between query and document (Okapi score), the importance of the web pages (the original search engine rank), personal relevance between the user and the target documents (our proposed LDA based methods). The details of features are as:

Okapi Score (OS) Okapi BM25 is a classic text retrieval algorithm and it produces a similarity score between the query and document.

Okapi Score Position (OSP) the position of the document in the initial ranked list from the Okapi BM25 ranking model.

Minimum of Search Engine Ranked Position (Min-SERP) The minimum ranked position of the document in the historical user click-through log. If a document has appeared twice in all ranked lists of the overall user logs, and in one time its ranked position is 1 and the other position is 5. Then the minimum of Search Engine Ranked Position (SERP) for this document is 1, and the maximum of SERP is 5, and the average SERP is 3.

Maximum of Search Engine Ranked Position (Max-SERP) The maximum ranked

position of the document in the historical user click-through log. The description can be found in feature Min-SERP.

Average of Search Engine Ranked Position (Avg-SERP) The average ranked position of the document in the historical user click-through log. The description can be found in feature Min-SERP.

Minimum of User Click Position (Min-UCP) The minimum rank of user click of the document for queries in the historical user click-through log. If a document has been clicked by users twice in all the ranked lists of the user logs and the first click sequence is the first click in a ranked list and the other click is the fifth click in a ranked list, then the minimum rank of user click for this document is 1, the maximum of user click position is 5 and the average user click position is 3. The feature is calculated from the all ranking lists in the user logs.

Maximum of User Click Position (Max-UCP) The maximum rank of user click of the document for queries in the historical user click-through log. The description can be found in feature Min-UCP.

Average of User Click Position (Avg-UCP) The average rank of user click of the document for queries in the historical user click-through log. The description can be found in feature Min-UCP.

LDA-based Personal Relevance (LPR) We use the defined personal relevance described in subsection 6.3.2 to indicate the personal relevance between target document and user.

LDA-based Hidden Topic Match (LHTM) We use the defined score of hidden topic match described in subsection 6.3.3 to indicate the personal relevance between hidden user topics and target web documents.

LDA-based Hidden Topic Match by Size Normalization (LHTMSN) We use the defined score of hidden topic match by size normalization described in subsection 6.3.3 to indicate the personal relevance between hidden user topics and target web documents.

The reason why we include the SERP and User Click Position (UCP) as the features in our experiments is we want to build a simulation of ranking methods for a web search task. In perfect practice, we should implement many features used in working search engine system. These features can be PageRank scores of the documents, the relevance of the anchor text and the user query, the click-through rate of the documents and etc. But the implementation work to build features for a web search system could consume too much time and this is not necessary for research purpose in our work. We propose to use the information in the user logs to simulate these important features usually used in a commercial search engine. For our proposed feature, the SERP scores for a document in the user log can be viewed as the importance of this document in the overall documents which play a similar role as the PageRank in the overall web documents. The UCP scores for a document can be viewed as metrics to how much the user wants to click to this document which play a similar role to the click-through rate for a document.

Overall, we use the text similarities, the original SERP, the user click information to produce a ranking method to simulate the original web search.

For each feature such as the original SERP, we use the minimum, maximum and average SERP as features to include more information in our feature engineering and this is a typical technique for building features in web search task and similar technique can be seen in the Microsoft Letor dataset [Qin *et al.*, 2010]. Based on these features we extract from the user logs, we add our proposed LDA based features to test whether the external resources can help to improve the effectiveness of the ranking method compared to our simulated web search ranking method.

In the following section, we evaluate the performance of these features in the Learning to Rank framework for a personalized search task. We test the hypothesis that these new features utilizing external resources can improve the overall retrieval effectiveness in the personalized task.

6.4 Evaluation

For our experimental setup, we again use the data from the Chinese commercial search engine - SOGOU.COM (NASDAQ:SOHU). This dataset was introduced in Section 5.3 of Chapter 5. The data includes one month's user query logs and a target Chinese Web collection. In the Learning to Rank method, we transfer all the data into features for training and testing. We use the same format as used in the SVMRank toolkit, as shown in Figure 6.11.

Figure 6.11 shows some examples of the feature tables. The 11 features used in our experiments are described as in Table 6.3, and the detailed description of these features can be found in section 6.3.4 of this chapter. Each line of the data in Figure 6.11 describes a query/document pair. The column

```
0 qid:1 1:14.7322 2:1 3:8 4:8 5:8 6:1 7:1 8:1 9:3.4924
        10:2.75332 11:3.01564
0 qid:1 1:14.7215 2:2 3:1 4:1 5:1 6:1 7:1 8:1 9:3.67499
        10:3.64854 11:3.61235
1 qid:1 1:14.6706 2:3 3:9 4:52 5:23 6:1 7:12 8:6 9:1.33252
        10:1.18035 11:1.19382
1 qid:1 1:14.6706 2:4 3:2 4:40 5:23 6:1 7:6 8:3 9:2.84597
        10:2.75658 11:2.74741
0 qid:1 1:14.5385 2:5 3:12 4:12 5:12 6:1 7:4 8:2 9:2.06866
        10:1.39491 11:1.62316
0 qid:1 1:14.4555 2:6 3:2 4:2 5:2 6:2 7:2 8:2 9:3.58439
        10:2.83131 11:3.0973
```

Figure 6.11: Example of feature table.

one is the judgement score for the query and the document(1 means the document has been clicked when this document is returned from the search engine using the query while 0 means not clicked), and column two is the query id, and the remaining columns are the value for the 11 features between the query and the document.

Table 6.3: The Description of the Feature Table Data.

Feature ID	Description of Feature
1	Okapi Score (OS)
2	Okapi Score Position (OSP)
3	Minimum of Search Engine Ranked Position (Min-SERP)
4	Maximum of Search Engine Ranked Position (Max-SERP)
5	Average of Search Engine Ranked Position (Avg-SERP)
6	Minimum of User Click Position (Min-UCP)
7	Maximum of User Click Position (Max-UCP)
8	Average of User Click Position (Avg-UCP)
9	LDA-based Personal Relevance (LPR)
10	LDA-based Hidden Topic Match (LHTM)
11	LDA-based Hidden Topic Match by Size Normalization (LHTMSN)

For the match score between the document and the query, all the user's

click-through documents are labeled as 1 and non-clicked documents as 0. To Ranking SVM, these labels mean that the clicked documents should rank before the non-clicked documents. We use 80 queries with 1000 returned web documents using the Okapi model. For each query and document pair, we produce the necessary feature scores. Thus overall, we have $80,000$ instances for training and testing in our experiments. We split these instances into $40,000$ of 40 queries as the training data and $40,000$ as the testing instances for the other 40 queries (The first 40 queries as the training data and the remaining 40 queries as the testing data).

6.4.1 Comparison with baselines

To test the effectiveness of our proposed methods, we investigate its effectiveness by combining different combination of features. The main investigation is the comparison with the features combination with and without our proposed personalized features. The difference of these Runs is listed as:

- The Run *Okapi* uses the features: OS and OSP.
- The Run $Okapi + rank + clickrank$ uses the features: OS, OSP, rank features, and clickrank features.
- The Run $Okapi + rank + clickrank + LDA1$ uses the features: OS, OSP, rank features, clickrank features and LDA1.
- The Run $Okapi + rank + clickrank + LDA12$ uses the features: OS, OSP, rank features, clickrank features, LDA1 and LDA2.

- The Run $Okapi + rank + clickrank + LDA123$ uses the features: OS, OSP, rank features, clickrank features, LDA1, LDA2 and LDA3.
- The Run $Okapi + rank + clickrank + LDA2$ uses the features: OS, OSP, rank features, clickrank features and LDA2.
- The Run $Okapi + rank + clickrank + LDA3$ uses the features: OS, OSP, rank features, clickrank features and LDA3.

We show our experimental results in Table 6.4. We show seven sets of results. For our previously listed features, we refer to the features related to the ranks of the documents as rank features including Min-SERP, Max-SERP and Avg-SERP; we refer to the features related to the user click ranks of the documents as clickrank features including Min-UCP, Max-UCP and Avg-UCP, and the three LDA related features as LDA1(LPR), LDA2(LHTM), and LDA3(LHTMSN).

Table 6.4: Comparison of search effectiveness with different features combination.

Runs	MAP	NDCG@10	P@10	ERR@10
Okapi	0.0589	0.0909	0.0467	0.0977
Okapi+rank+clickrank	0.3732	0.4119	0.2533	0.3418
Okapi+rank+clickrank+LDA1	0.4113	0.5346	0.2933	0.338
Okapi+rank+clickrank+LDA12	**0.4389**	0.5346	0.2967	0.3698
Okapi+rank+clickrank+LDA123	0.4339	**0.5659**	**0.3133**	**0.417**
Okapi+rank+clickrank+LDA2	0.3995	0.4919	0.2667	0.4068
Okapi+rank+clickrank+LDA3	0.4178	0.5448	0.2667	0.34

The results in Table 6.4 show that using of the Okapi score is not suitable for this task. This means that it is not effective to rank documents based on the query-document matching for this tasks. However, this result can be

anticipated for several reasons. It is worth noting that the assumption that all relevant documents have previously been clicked by the user when using a commercial search engine may impact negatively on this results. Users of web search engines typically click on high ranked documents, this possible relevant documents retrieved at lower ranks will not have been considered by the user. Thus, they are assumed not to be relevant and if ranked higher in the Okapi lists than their original rank in the commercial search engine will actually impact negatively on the results. Also, we cannot know the features used to create the ranked lists for the commercial search engine, but it is likely that their ranking is at least partially based on web link structure in the form of a PageRank type score, and general popularity of the content as measured by clicks on the documents by users of the search engine. Thus, the ranking of these clicked documents will have been based significantly on factors unrelated to content, and hence we can anticipate that content only based ranking based on an Okapi type function will produce poor retrieval effectiveness.

Incorporating rank based signals into the ranking function based on the initial ranking from the commercial search engine produces a large improvement in all retrieval metrics, as could be expected since they are based on the behavior of the commercial search engine, and gives us a more useful point of comparison for out investigations.

Looking further at Table 6.4 , we can see that each of the LDA methods individually improve on the okapi+rank+clickrank results. None of the separate LDA methods are clearly superior with different methods preferred as measured by alternative retrieval metrics. The most interesting result however

is that the best overall result is achieved by combination of all three methods in the learning-to-rank framework. This indicates that each of these signals provides difficult useful information related to the ranking of documents of interest to the user which can be used effectively in combination.

To further illustrate the effectiveness of our proposed methods, we also show the difference of MAP for individual query between Run $Okapi + rank + clickrank + LDA1$ and $Okapi + rank + clickrank + LDA12$ as Figure 6.12. The results show that MAP is improved for most queries, and that the p-value to compare the MAP scores of the two Runs by the t-test is $p = 0.04884$.

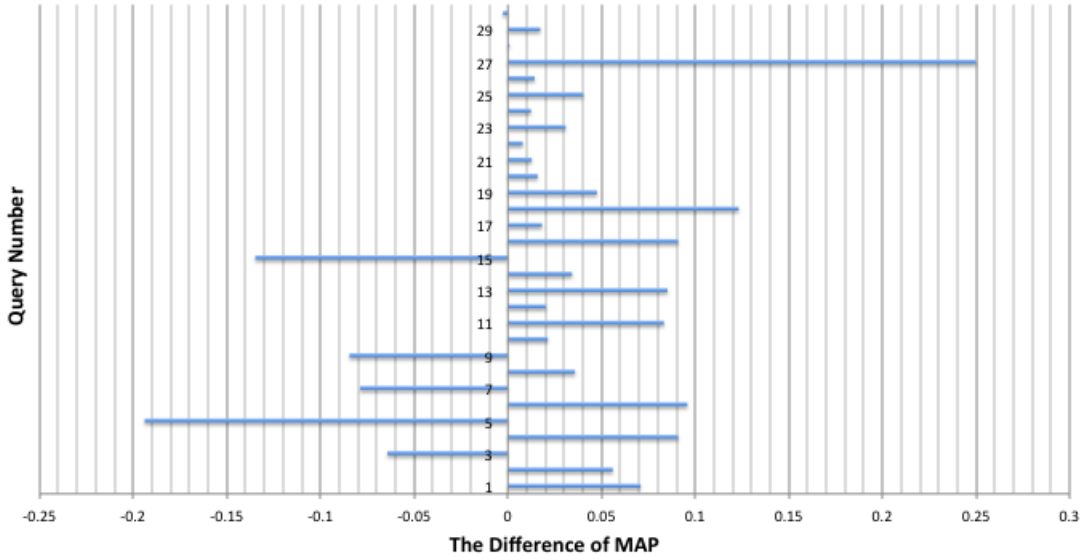

Figure 6.12: Differece of MAP for Runs Okapi+rank+clickrank+LDA1 and Okapi+rank+clickrank+LDA12.

6.4.2 Discussion

Classical IR algorithms focus on the textual relevance without considering whether the search results satisfy an individual user's search interests. Personal relevance aims to reveal the user's query intent and return relevant documents describing the topic from the perspective of the user's search interest in the topics. In the situation that the user's history log does not provide information about the user's current interest topic which differs from their past typical interests, it is difficult for IR algorithms to provide the personalized results for the user query.

In this chapter, we describe a method to enrich the user search interest model from external resources - a web corpus. This method utilizes the correlation between the historical user topics and the hidden topics among the web corpus to enrich the user search topic model. Providing personalized results for the new search topics of users is very important for the user search experience. How to model the user's new interests before they are present in the search log is a challenging problem for a search system. We define the LDA-based topic relevance as a solution to model the relation between the historical topics and hidden topics for the user. Furthermore, the Learning to Rank framework is used to combine several features concerning topical relevance between target corpus and user logs. The results of our experiments show improvement on a Chinese web data search task.

In our experiments, we aim to improve the rank based on the user's clicks. We use the features include Okapi, rank and clickrank to simulate the ranked list from the search engine. These features include the text similarity between

the query and the documents, the ranked position of the documents in the user logs, the rank that the user chooses to click for this documents in the user logs. Our experimental results show that the combination with the LDA features improve the overall retrieval effectiveness.

6.5 Summary

In this chapter, we have described a study of a topic model based personalized method for a web data search task. We conducted the corpus analysis with the web corpus and user logs. Our findings showed that users are not always consistent in the search topics in their own search logs. We proposed to utilize external resources to extend the model of user search interests based on the user's historical log data. This updating is based on the correlation between the potential user topics and the past search topics. We define an LDA-based topic relevance score and an LDA-based hidden topic match score to describe the correlation between the potential user topics and past search topics. These scores are used as features in a Learning to Rank framework. Our results show significant improvement compared to the baseline system produced by combination of standard features without using the external resources.

To answer the research questions we propose in the beginning of this chapter, we get the conclusions as follows:

- How to model the user and document in topic modeling framework for personalized search task? In our research, the user historical data and target documents are modeled by LDA and then each user and document can be represented by topic weights from LDA.

- How to utilize external resources in user modeling and document modeling? Topic models from user historical data can be enriched from topics models of external documents in our research. This method aims to cover the potential topics which are missed in the current user data.

- How to utilize external resources based user models to rank documents in personalized search task in learning to rank framework? The similarity between user models and documents models of the target documents can be used to describe the personal relevance between the user and the target documents. Then these similarity scores can be used as features in learning to rank framework to rank target documents for different users.

Chapter 7

Conclusions and Future Work

This thesis focuses on the utilization of external resources to improve IR effectiveness. Many IR applications are objected to a sparse data problem. This problem of sparse data can greatly affect retrieval effectiveness. In past research of IR, relevance feedback (RF) has been shown to be a key effective method to address this problem. In typical RF algorithms, a key assumption is that the target corpus contains enough information to suitably enrich the user query. When this assumption is not valid, a sparse data problem occurs, it can potentially cause the failure of typical RF methods. On the other hand, as we have demonstrated that widely available external resources such as Wikipedia and other Internet web documents can potentially play a role in resolving the sparse data problem in the RF process. Our research begins with the sparse data problem in queries and documents, and then goes on addressing the user which are three most important parts of modern IR tasks. This chapter summarizes the findings of this thesis and gives a suggestion for potential future directions for IR research following these topics.

7.1 Contributions of the Thesis

The previous research topics are relevant to this thesis are primarily relevance feedback and personalized search where much work already exists. In the state-of-art work, mainstream research has focused on how to effectively utilize the information from the target corpus or the user logs to enrich the original user query. Progress has been made by utilizing many techniques such as query expansion from the target corpus including various feedback term weighting methods, user modeling using the categories of the ODP system, learning-based methods to choose feedback terms, clustering methods to find a key topic in the feedback information, and building user models for the user search logs. There has been less focus part on using the external resources to resolve the sparse data problem existing in the target corpus or the user logs. Our overall idea is using external resources to resolve the sparse data problem in the IR tasks. We view IR as a task which should include the user, the query, and the target documents. A complete IR system should model the search interests of the user and provide personalized search results to the user retrieved from the target corpus for a specific user query. In this thesis, we attack the sparse data problem from all these three aspects: user, query, and documents.

After the survey work on relevance feedback and personalized search of Chapter 2, we began the research on the use of external resources for query expansion in Chapter 3. Query expansion (QE) is one of the most widely used methods to improve retrieval effectiveness in many IR tasks. We explored QE using external resources. To demonstrate our research hypothesis, we tested

our external QE method on a text-image retrieval task, which is a typical retrieval task where sparse data problem happens. Our results demonstrate that external QE works better than the state-of-art QE on the target corpus. To further investigate the external QE method, our proposed definition-based relevance feedback algorithm showed further improvement compared to the utilization of standard RF method on the external resources. This is because our algorithm utilizes external resources more thoroughly compared to the indiscriminate imitation of the standard RF method from the target corpus to the external resources. It shows that the utilization of external resource to enrich user queries can play a significant role in improving retrieval effectiveness. Utilizing the definition documents found in the external resources for the user query in the process of relevance feedback can further improve the final results. This is the first step we introduce the external resources in the IR process from the classical methodology of QE. Our contribution to the knowledge in this work can be summarized as:

- QE from external resources can bring more useful feedback information to the process of relevance feedback in IR tasks with the sparse data problem.
- The combination of QE from external resources and target collection produces the best result in IR tasks with the sparse data problem.
- We propose a new DRF method to utilize the external resources in IR tasks with the sparse data problem.

In the fourth chapter, we introduced our work on document expansion using external resources. DE is a less used method on retrieval tasks due to

past research reports various results by applying DE on IR tasks including negative results. Regarding IR tasks with the sparse data problem, usually the target document is short and needs more terms to describe the content of it which is different with the newswire retrieval tasks widely explored on the classical IR research. We hypothesized that the DE method can help to resolve the sparse data problem. We demonstrate that DE works well and our results outperform the best result in the official runs on the same evaluation task. This is due to the sparse data problem on the document sides being the key factor influencing retrieval effectiveness in short-length documents retrieval. Resolving the sparse data problem on target documents is the most effective way to improve the retrieval effectiveness and our results demonstrate this conclusion. Our proposed method uses short documents as the query to search external resources to get the relevant documents. The classical Okapi relevance feedback method can play an effective role to get the relevant information from the external resources. This feedback information provides an effective supplement to enrich the original short documents with sparse data problem. In our research, our proposed document reduction method also improves the standard document expansion which has been used before. This methodology helps to improve the final retrieval results further. One notable question in this research is that our good retrieval results may be due to the content of used external resources covering the topics of the target collection well. This is an interesting topic to explore in the future to examine the relationship between the target corpus and the external resources when utilizing external resources for retrieval tasks. Our contribution to the knowledge in this work can be summarized as:

- DE from external resources can get similar result with the QE method from target corpus in IR tasks with the sparse data problem.
- The combination of the DE method for document reduction gets better results in IR tasks with the sparse data problem.
- The combination of DE and QE methods get the best results. It indicates that these two methods should be utilized together in IR tasks with the sparse data problem.

Chapter 5 and 6 describes our research moving from the query and document to the user side. One ultimate goal of IR research is to provide personalized search results for each user. In this process, user historical data is an important resource to capture the search interests of users. Personalized search includes user historical information in the retrieval process to adapt retrieval for the individual user. Evidence shows that it is usually difficult to collect the complete historical data of users in many search tasks. Thus, the sparse data problem exists when modeling the user search interests using user historical data. In our research on personalized search, we introduce external resources as a knowledge base to build a user model for later retrieval processes. Modelling the user's search interests is the key step to conduct the later personalized retrieval. Our clustering method divides the external resources into categories. This is a necessary step to build a knowledge base for modeling user search interests and underlying topics of target documents. In this research, the user historical data and the target corpus are all mapped into this knowledge base to build the topic distribution of the user and the target documents. The similarity between these two distributions can produce

the topical relevance between user historical data and target documents. This is the first step we try to utilize external resources on personalized search task. Our results show significant improvement compared to the standard retrieval method without considering topical relevance between user search interests and topics of target documents. In chapter 6, the topical relevance between the user data and target documents are used as features in a learning to rank framework and it shows the improvement in the retrievlal effectiveness. Our contribution to the knowledge in this work can be summarized as:

- We propose a clustering based method to classify the external resources into categories. Thus the user historical data can be mapped into these categories to build user models. The external resources based user models are used to compare with the target documents by topical relevance in our experiments, and then the topical relevance is used to rank the target documents for different users. This work demonstrates that the external resources can be utilized into building user models for personalized search task.

- In our research, the user historical data and target documents are modeled by LDA and then each user and document can be represented by topic weights from LDA. Topic models from user historical data can be enriched from topics models of external documents in our research. The similarity between user models and documents models of the target documents can be used to describe the personal relevance between the user and the target documents. Then these similarity scores can be used as features in learning to rank framework to rank target documents for

diverse users. This work demonstrates that the external resources based features help to improve the retrieval effectiveness in learning to rank framework.

In this thesis, we demonstrate the effectiveness of enriching IR on three aspects of the IR process: user, query, and document. Our results demonstrate several important conclusions: The first conclusion is a direct utilization of the classical algorithms from the target corpus to external corpus can help to resolve the sparse data problem and it demonstrates the robustness of the classical algorithms on different situation. This is concluded from our work on external query expansion, external document expansion and personalized search using Wikipedia. The second conclusion is that when utilizing the external resources finely designed algorithms can help to further improve the effectiveness of utilization of external resources, and this is concluded from our work on definition-based relevance feedback and document reduction work. The third conclusion is that modern machine learning techniques such as ranking SVM and topic modelling are effective methods to utilize external resources on personalized search tasks, and this validated in our work on learning-based hidden topic matching.

7.2 Revisiting the Hypotheses of the Thesis

As we described at the beginning of this thesis, we hypothesized that external resources can be helpful in the relevance feedback process and to improve the personalized search tasks for sparse data problem in IR tasks. From our research on the relevance feedback using external resources - query expansion

and document expansion, our experiments show that the significant retrieval improvement can be achieved compared to the classical methods using only the target corpus as the feedback sources. It demonstrates that the external resources can be helpful to improve the retrieval effectiveness in the process of relevance feedback for IR tasks with the sparse data problem.

Furthermore, in the personalized search task where sparse data problem happens on the user historical data, the enrichment of the user data from external resources also demonstrates its effectiveness compared to the text similarity based method and simulated search engine baseline. All this evidence demonstrates that external resources can help resolve the sparse data problem with all aspects of the IR process - user, query, and document.

7.3 Future Directions

In this thesis, we proposed using external resources to resolve sparse data problem in IR research. This could be an important direction to explore for improving the overall retrieval effectiveness since the main problem in many IR tasks where there is a lack of data from one to all components in the IR process. Without enough information, it is difficult to improve the retrieval effectiveness. From this thesis's research, many possible future directions can be proposed.

- Query expansion is a classical method to enrich the user query from the target corpus. Our research improves the typical QE method by utilizing the external resources. Although our research has demonstrated external query expansion can help to improve the retrieval effective-

ness, there is a chance to fully utilize more information of the external resources such as the link graph relation of the Wikipedia pages and web pages. Those pages linked to the definition documents can provide more information to help finding more relevant information to enrich the user query.

- External resources have been shown to be helpful for improving retrieval tasks with sparse data problem in our experiments. An interesting topic is how to select appropriate external resources for different target corpus. The purpose of IR is to satisfy the user's information need by the user query from the target corpus. If there is a better method to select external resources for the retrieval task, it may be better to resolve the sparse data problem within the retrieval process. The research question is how to select an external resource for different retrieval tasks. Similarity or topic coverage between the target corpus, the user query and the external resource could be useful methods to carry on this research.

- Current document expansion methods can handle short documents very well since usually short documents usually only focus on a single topic. Enriching short document from external resources typically uses the short document as a query to find more relevant information from external resources to enrich itself. This is a straightforward method to apply document expansion. But for those documents which contain more than one main topic, the situation is more complex. Current topic modeling method can help to identify the main topics in documents. Thus, the top terms belong to different topics can be used as different queries to find

different relevant information about different topics. This relevant information belonging to different topics can be used to enrich the original document. It may potentially help to resolve the sparse data problem of complex documents which include multiple topics.

- In query expansion research, state-of-art algorithms can process different queries with different parameters for acquiring feedback information. The hypothesis is that different queries should be treated differently rather than use a unified coefficient for all the queries. The assumption is that more ambiguous queries should be given more feedback information to make their focus clearer. This method can be also adapted to the DE process. Different documents have different levels of sparse data problem, which are the levels that they are needed to be enriched from external resources. A learning-based method may help to decide what is the best parameter to acquire feedback information in the DE process. This is an interesting topic to explore when conducting DE from external resources.

- Our experiments with the use of external resources with an LDA model of the user's topical interests for personalized search have illustrated the potential for unstructured web-based knowledge sources to successfully augment information gathered from the user's click-through data, when small amounts of data are collected on topics of interest user leading to a sparse data problem. Our investigation was carried out using a small randomly selected external collection of web-documents. Further investigation is needed to explore the potential impact of using a larger

external document collection with out LDA methods. In addition, we could explore considering the selection of the contents for this collection. Potentially different types of content may be found to be more useful, e.g. documents from more reliably sources or containing more detailed descriptions of topics may be more useful. If this is the case, then suitably filtering external collections may improve effectiveness of our LDA methods. Our earlier work on document expansion for image retrieval showed improved effectiveness using an enhanced term selection method, and it may be beneficial to consider a more sophisticated interaction with the external resources for search personalization.

In this thesis, we present the sparse data problem in two retrieval tasks: text-based image retrieval and personalized search. Our research focuses on utilizing external resources to enrich the three aspects of the typical retrieval process: user, query, and document. Our research concludes that the external resources can help to resolve the sparse data problem in all these three aspects. In the past research, less attention has been paid to the sparse data problem which greatly harms retrieval effectiveness. Our research provides a perspective on utilizing external resources in resolving sparse data problem for IR tasks. Deeper questions regarding the sparse data problem may be answered in future work on this topic, since there is still a long way to go before IR systems achieve best possible performance.

Glossary

Information Retrieval Information retrieval (IR) is concerned with satisfying a user's information need by retrieving documents or data collection in other formats such as image, video and speech.

Relevance Feedback Relevance feedback is a feature of some systems to take the results from an initial retrieval operation for a given query and to use this to create a revised query and revise the sytem parameters to improve retrieval effectinvess in a later retrival run for this query.

Query Expansion Query expansion (QE) is the process of reformulating a seed query to improve retrieval performance in information retrieval operations. In the context of Web search engines, query expansion involves evaluating a user's input (what words were typed into the search query area, and sometimes other types of data) and expanding the search query to match additional documents.

Document Expansion Document expansion (DE) is the process of expanding target documents by terms from target collection or external collection before the indexing of target collection. It aims to bring more information into the target documents to improve retrieval performance in

information retrieval. There is still no conclusion about whether DE is useful for general IR tasks.

Personalization Personalization technology enables the dynamic insertion, customization or suggestion of content in any format that is relevant to the individual user, based on the user's implicit behaviour and preferences, and explicitly given details.

User Profile A user profile is a collection of personal data associated with a specific user. A profile refers therefore to the explicit digital representation of a person's identity. A user profile can also be considered as the computer representation of a user model.

User Modelling User modelling is a subdivision of human-computer interaction and describes the process of building up and modifying a user model. The main goal of user modelling is customization and adaptation of systems to the user's specific needs.

Topic Model In machine learning and natural language processing, a topic model is a type of statistical model for discovering the abstract "topics" that occur in a collection of documents.

Publications List

- J.Min, P.Wilkins, J.Leveling and G.J.F.Jones. DCU at WikipediaMM 2009: Document Expansion from Wikipedia Abstracts. In Proceedings of the CLEF 2009: Workshop on Cross-Language Information Retrieval and Evaluation, Corfu, Greece, 2009.

- J.Min, J.Leveling, D.Zhou and G.J.F.Jones. Document Expansion for Image Retrieval. In Proceedings of 9th international conference on Adaptivity, Personalization and Fusion of Heterogeneous Information (RIAO 2010), Paris, France, April 2010.

- W.Magdy, J.Min, J.Leveling, and G.J.F.Jones. Building a Domain-Specific Document Collection for Evaluating Metadata Effects on Information Retrieval. In Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010), Malta, May 2010.

- J.Min, J.Jiang, J.Leveling, G.J.F.Jones and A.Way. DCU's Experiments for the NTCIR-8 IR4QA Task. In Proceedings of the Eighth NTCIR Workshop on Research in Information Access Technologies, Tokyo, Japan, June 2010.

- J.Min, J.Leveling and G.J.F.Jones. Document Expansion for Text-based

Image Retrieval at WikipediaMM 2010. CLEF (Notebook Papers/LABs/ Workshops), Padua, Italy, 2010.

- W.Li, J.Min and G.J.F.Jones. A Text-Based Approach to the ImageCLEF 2010 Photo Annotation Task. CLEF (Notebook Papers/LABs/Workshops), Padua, Italy, 2010

- J.Min and G.J.F.Jones. Building User Interest Profiles from Wikipedia Clusters. In Proceedings of the Workshop on Enriching Information Retrieval (ENIR 2011) at SIGIR 2011, Beijing, China, July 2011.

- J.Min and G.J.F.Jones. External Query Reformulation for Text-based Image Retrieval . In Proceedings of the 18th Symposium on String Processing and Information Retrieval (SPIRE 2011), Pisa, Italy, October 2011.

- J.Min, C.Lopes, J.Leveling, D.Schmidtke and G.J.F.Jones. Multi-Platform Image Search using Tag Enrichment. In Proceedings of the Thirty-Fifth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2012), Portland, OR, U.S.A., pp1018, August 2012.

Bibliography

Baziz, M., Boughanem, M., Pasi, G. & Prade, H. (2007). An information retrieval driven by ontology from query to document expansion. In *Large Scale Semantic Access to Content (Text, Image, Video, and Sound)*, RIAO '07, pages 301–313, Pittsburgh, Pennsylvania. 94, 95

Bendersky, M., Metzler, D. & Croft, W.B. (2012). Effective query formulation with multiple information sources. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, pages 443–452, Seattle, Washington, USA. 47

Billerbeck, B. & Zobel, J. (December 2005). Document expansion versus query expansion for ad-hoc retrieval. In *The Tenth Australasian Document Computing Symposium*, pages 34–41, Sydney, Australia. 10, 88, 93

Blei, D. & Lafferty, J. (2006a). Correlated topic models. In Y. Weiss, B. Schölkopf & J. Platt, eds., *Advances in Neural Information Processing Systems 18*, pages 147–154, MIT Press, Cambridge, MA. 153, 174

Blei, D.M. & Lafferty, J.D. (2006b). Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pages 113–120, Pittsburgh, Pennsylvania. 153

Blei, D.M., Ng, A.Y. & Jordan, M.I. (2003). Latent dirichlet allocation. *J. Mach. Learn. Res.*, **3**, pages 993–1022. 147, 151

Bordogna, G. & Pasi, G. (1995). Linguistic aggregation operators of selection criteria in fuzzy information retrieval. *International Journal of Intelligent Systems*, **10**, pages 233–248. 95

Bouchoucha, A., He, J. & Nie, J.Y. (2013). Diversified query expansion using conceptnet. In *Proceedings of the 22Nd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '13, pages 1861–1864, San Francisco, California, USA. 47

Braun, M., Dellschaft, K., Franz, T., Hering, D., Jungen, P., Metzler, H., Müller, E., Rostilov, A. & Saathoff, C. (2008). Personalized search and exploration with mytag. In *Proceeding of the 17th international conference on World Wide Web*, WWW '08, pages 1031–1032, Beijing, China. 40

Buckley, C., Salton, G. & Allan, J. (1994a). The effect of adding relevance information in a relevance feedback environment. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '94, pages 292–300, Dublin, Ireland. 23, 62

Buckley, C., Salton, G., Allan, J. & Singhal, A. (1994b). Automatic query expansion using smart: Trec 3. In *TREC*, pages 69–80. 9

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N. & Hullender, G. (2005). Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, ICML '05, pages 89–96, Bonn, Germany. 157

Cao, G., Nie, J.Y., Gao, J. & Robertson, S. (2008). Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 243–250, Singapore, Singapore. 25

Cao, Y., Xu, J., Liu, T.Y., Li, H., Huang, Y. & Hon, H.W. (2006). Adapting ranking svm to document retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 186–193, Seattle, Washington, USA. 157, 158

Cao, Z., Qin, T., Liu, T.Y., Tsai, M.F. & Li, H. (2007a). Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pages 129–136, Corvalis, Oregon. 157

Cao, Z., Qin, T., Liu, T.Y., Tsai, M.F. & Li, H. (2007b). Learning to rank: From pairwise approach to listwise approach. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 129–136, Corvalis, Oregon, USA. 157

Carman, M.J., Crestani, F., Harvey, M. & Baillie, M. (2010). Towards query log based personalization using topic models. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 1849–1852, Toronto, ON, Canada. 153

Carmel, D., Zwerdling, N., Guy, I., Ofek-Koifman, S., Har'el, N., Ronen, I., Uziel, E., Yogev, S. & Chernov, S. (2009). Personalized social search based on the user's social network. In *Proceedings of the 18th ACM conference*

on Information and knowledge management, CIKM '09, pages 1227–1236, Hong Kong, China. 42, 43

Chang, Y.C. & Chen, H.H. (2007). Experiment for using Web Information to do Query and Document Expansion. In *Working Notes for the CLEF 2007 Workshop*. 93

Chen, J., Chu, W., Kou, Z. & Zheng, Z. (2010). Learning to blend by relevance. *CoRR - Computing Research Repository*, **abs/1001.4597**. 157

Chirita, P.A., Nejdl, W., Paiu, R. & Kohlschütter, C. (2005). 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, Salvador, Brazil. 124

Chirita, P.A., Firan, C.S. & Nejdl, W. (2007). 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, Amsterdam, The Netherlands. 42

Collins-Thompson, K. & Callan, J. (2005). Query expansion using random walk models. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, CIKM '05, pages 704–711, Bremen, Germany. 55

Collins-Thompson, K. & Callan, J. (2007). Estimation and use of uncertainty in pseudo-relevance feedback. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 303–310, Amsterdam, The Netherlands. 25, 26

Cooper, W.S., Gey, F.C. & Dabney, D.P. (1992). Probabilistic retrieval based on staged logistic regression. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '92, pages 198–210, Copenhagen, Denmark. 157

Crammer, K. & Singer, Y. (2001). Pranking with ranking. In *Advances in Neural Information Processing Systems 14*, pages 641–647. 157

Croft, B.W., ed. (2000). *Advances in Information Retrieval: Recent Research from the Center for Intelligent Information Retrieval*, The Kluwer International Series in Information Retrieval, Kluwer Academic Publishers, Boston. 93

Cui, H., Wen, J.R., Nie, J.Y. & Ma, W.Y. (2003). Query expansion by mining user logs. *IEEE Trans. on Knowl. and Data Eng.*, **15**, pages 829–839. 41

Das, A.S., Datar, M., Garg, A. & Rajaram, S. (2007). Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 271–280, Banff, Alberta, Canada. 149

David, S., Kevyn, C.T., Paul, N.B., Ryen, W.W., Susan, D. & Bodo, B. (2012). Probabilistic models for personalizing web search. In *Proceedings of the international conference on Web search and web data mining*, WSDM '12, Seattle, Washington, USA. 154

Diaz, F. & Metzler, D. (2006). Improving the estimation of relevance models using large external corpora. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 154–161, Seattle, Washington, USA. 15, 45, 48

Dou, Z., Song, R. & Wen, J.R. (2007). 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, Banff, Alberta, Canada. 122, 146

Dupret, G., Murdock, V. & Piwowarski, B. (2007). Web search engine evaluation using click-through data and a user model. In *In Proceedings of the Workshop on Query Log Analysis (WWW), 2007*. 137

feng Tsai, M., Liu, T.Y., Qin, T., Chen, H.H. & Ma, W.Y. (2006). Frank: A ranking method with fidelity loss. Tech. Rep. MSR-TR-2006-155, Microsoft Research. 157

Ferragina, P. & Gulli, A. (2005). A personalized search engine based on web-snippet hierarchical clustering. In *Special interest tracks and posters of the 14th international conference on World Wide Web*, WWW '05, pages 801–810, Chiba, Japan. 122

Freund, Y., Iyer, R., Schapire, R.E. & Singer, Y. (2003). An efficient boosting algorithm for combining preferences. *J. Mach. Learn. Res.*, **4**, pages 933–969. 157

Fuhr, N. (1989). Optimum polynomial retrieval functions based on the probability ranking principle. *ACM Trans. Inf. Syst.*, **7**, pages 183–204. 157

Gauch, S., Chaffee, J. & Pretschner, A. (2003). Ontology-based personalized search and browsing. In *Web Intelligence and Agent Systems*, vol. 1, pages 219–234, IOS Press, Amsterdam, The Netherlands, The Netherlands. 122

Griffiths, T.L. & Steyvers, M. (2004). Finding scientific topics. *PNAS*, **101**, pages 5228–5235. 152

Grubinger, M., Clough, P., Hanbury, A. & Muller, H. (2008). Overview of the imageclef photo 2007 photographic retrieval task. In C. Peters, V. Jijkoun, T. Mandl, H. Muller, D. Oard, A. Penas, V. Petras & D. Santos, eds., *Advances in Multilingual and Multimodal Information Retrieval*, vol. 5152 of *Lecture Notes in Computer Science*, pages 433–444, Springer Berlin Heidelberg. 93

Harman, D. (1992). Relevance feedback revisited. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '92, pages 1–10, Copenhagen, Denmark. 23

Haveliwala, T.H. (2002). Topic-sensitive pagerank. In *Eleventh International World Wide Web Conference (WWW 2002)*. 36, 40

Haveliwala, T.H., Gionis, A., Klein, D. & Indyk, P. (2002). Evaluating strategies for similarity search on the web. In *Proceedings of the 11th International Conference on World Wide Web*, WWW '02, pages 432–442, Honolulu, Hawaii, USA. 75

He, B. & Ounis, I. (2007). Combining fields for query expansion and adaptive query expansion. *Inf. Process. Manage.*, **43**, pages 1294–1307. 55

He, J., Hollink, V. & de Vries, A. (2012). Combining implicit and explicit topic representations for result diversification. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 851–860, Portland, Oregon, USA. 46, 48

Ide, E. (1968). New experiments in relevance feedback. In *Scientific Report ISR-14, Cornell University*. 15, 19

Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, **37**, pages 547–579. 75

Jeh, G. & Widom, J. (2003). Scaling personalized web search. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 271–279, Budapest, Hungary. 122, 146

Jin, R., Valizadegan, H. & Li, H. (2008). Ranking refinement and its application to information retrieval. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 397–406, Beijing, China. 157

Joachims, T. (2002). Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, pages 133–142, Edmonton, Alberta, Canada. 137, 157, 158, 159

Joachims, T. (2006). Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 217–226, Philadelphia, PA, USA. 178

Jones, K.S., Walker, S. & Robertson, S. (2000). A probabilistic model of information retrieval: development and comparative experiments - Part 1. *Information Processing and Management*, **36**, pages 779–808. 22

Keenoy, K. & Levene, M. (2005). Personalisation of web search. In B. Mobasher & S. Anand, eds., *Intelligent Techniques for Web Personalization*, vol. 3169 of *Lecture Notes in Computer Science*, pages 201–228, Springer Berlin / Heidelberg. 124

Kritikopoulos, A. & Sideri, M. (2005). The compass filter: Search engine result personalization using web communities. In B. Mobasher & S. Anand, eds., *Intelligent Techniques for Web Personalization*, vol. 3169 of *Lecture Notes in Computer Science*, pages 229–240, Springer Berlin / Heidelberg. 123

Kwok, K.L. (2000). Improving English and Chinese ad-hoc retrieval: A Tipster text phase 3 project report. *Information Retrieval*, **3**, pages 313–338. 55

Lacoste-Julien, S., Sha, F. & Jordan, M.I. (2008). Disclda: Discriminative learning for dimensionality reduction and classification. In *Advances in Neural Information Processing Systems 21*, pages 897–904. 153

Lam-Adesina, A.M. & Jones, G.J.F. (2001). Applying summarization techniques for term selection in relevance feedback. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, pages 1–9, New Orleans, Louisiana, United States. 23

Lavrenko, V. & Croft, W.B. (2001). Relevance based language models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, pages 120–127, New Orleans, Louisiana, United States. 45, 153

Levow, G.A. & Oard, D.W. (2002). *Signal boosting for translingual topic tracking: document expansion and n-best translation*, pages 175–195. Kluwer Academic Publishers, Norwell, MA, USA. 91

Li, P., Burges, C. & Wu, Q. (2008). Learning to rank using classification and gradient boosting. In *Advances in Neural Information Processing Systems 20*, pages 0–7. 157

Liu, F., Yu, C. & Meng, W. (2002). Personalized web search by mapping user queries to categories. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, pages 558–565, McLean, Virginia, USA. 15, 36, 122

Liu, F., Yu, C. & Meng, W. (2004). Personalized web search for improving retrieval effectiveness. *IEEE Transactions on Knowledge and Data Engineering*, **16**, pages 28–40. 123, 146

Liu, T.Y. (2009). Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, **3**, pages 225–331. 150, 157, 178

Liu, X. & Croft, W.B. (2004). Cluster-based retrieval using language models. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '04, pages 186–193, Sheffield, United Kingdom. 92

Lv, Y. & Zhai, C. (2009). Adaptive relevance feedback in information retrieval. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 255–264, Hong Kong, China. 24, 25

Lv, Y. & Zhai, C. (2010). Positional relevance model for pseudo-relevance feedback. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pages 579–586, Geneva, Switzerland. 25

Micarelli, A., Gasparetti, F., Sciarrone, F. & Gauch, S. (2007). Personalized search on the world wide web. In *The adaptive web: methods and strategies of web personalization*, pages 195–230, Springer-Verlag, Berlin, Heidelberg. 123

Page, L., Brin, S., Motwani, R. & Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab. 40

Pitkow, J., Schütze, H., Cass, T., Cooley, R., Turnbull, D., Edmonds, A., Adar, E. & Breuel, T. (2002). Personalized search. *Communications of the ACM*, **45**, pages 50–55. 122

Pretschner, A. & Gauch, S. (1999). Ontology based personalized search. In *Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence*, ICTAI '99, pages 391–398. 15, 33, 38, 44, 121, 126, 146

Qin, T., Zhang, X.D., Tsai, M.F., Wang, D.S., Liu, T.Y. & Li, H. (2008). Query-level loss functions for information retrieval. *Inf. Process. Manage.*, **44**, pages 838–855. 157

Qin, T., Liu, T.Y., Xu, J. & Li, H. (2010). Letor: A benchmark collection for research on learning to rank for information retrieval. *Inf. Retr.*, **13**, pages 346–374. 158, 182

Qiu, F. & Cho, J. (2006). Automatic identification of user interest for personalized search. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 727–736, Edinburgh, Scotland. 40, 122

Ramanathan, K. & Kapoor, K. (2009). Creating user profiles using wikipedia. In A. Laender, S. Castano, U. Dayal, F. Casati & J. de Oliveira, eds., *Conceptual Modeling - ER 2009*, vol. 5829 of *Lecture Notes in Computer Science*, pages 415–427, Springer Berlin / Heidelberg. 122

Rijsbergen, C.J.V. (1979). *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edn. 21

Robertson, S. & Spärck Jones, K. (1994). Simple, proven approaches to text retrieval. Tech. Rep. UCAM-CL-TR-356, University of Cambridge, Computer Laboratory. 16, 73, 97, 99, 106, 177

Robertson, S., Walker, S., Beaulieu, M. & Willett, P. (1999). Okapi at trec-7: Automatic ad hoc, filtering, vlc and interactive track. *In*, **21**, pages 253–264. 54

Robertson, S.E. (1991). On term selection for query expansion. *J. Doc.*, **46**, pages 359–364. 15, 21, 22, 56, 73, 97, 98

Robertson, S.E. & Walker, S. (2000). Okapi/Keenbow at TREC-8. In *The Eighth Text REtrieval Conference (TREC 8)*, pages 151–161. 54

Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M. & Gatford, M. (1994). Okapi at TREC-3. In *Proceedings of the Third Text REtrieval Conference*, pages 109–126. 56, 62, 73, 97

Robertson, S.E., Walker, S. & Hancock-beaulieu, M. (2000). Experimentation as a way of life: Okapi at TREC. *Information Processing and Management*, **36**, pages 95–108. 54, 105

Rocchio, J. (1971). Relevance feedback in information retrieval. In *In Gerard Salton, editor, The SMART Retrieval System-Experiments in Automatic Document Processing*, pages 313–323. 9, 15, 19

Salton, G. (1971). *The SMART Retrieval System - Experiments in Automatic Document Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA. 19, 59, 99

Salton, G., ed. (1988). *Automatic Text Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA. 33

Salton, G. & Buckley, C. (1997). Readings in information retrieval. In K. Sparck Jones & P. Willett, eds., *Readings in information retrieval*, chap. Term-weighting approaches in automatic text retrieval, pages 323–328, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. 73

Salton, G., Wong, A. & Yang, C.S. (1975). A vector space model for automatic indexing. *Commun. ACM*, **18**, pages 613–620. 19

Sculley, D. (2010). Combined regression and ranking. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 979–988, Washington, DC, USA. 157

Shen, X., Tan, B. & Zhai, C. (2005). Implicit user modeling for personalized search. In *Proceedings of the 14th ACM international conference on Information*

and knowledge management, CIKM '05, pages 824–831, Bremen, Germany. 42, 122

Singhal, A. & Pereira, F. (1999a). Document Expansion for Speech Retrieval. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 34–41. 11, 88

Singhal, A. & Pereira, F. (1999b). Document expansion for speech retrieval. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 34–41, Berkeley, California, USA. 89, 104

Singhal, A., Choi, J., Hindle, D., Lewis, D.D. & Pereira, F.C.N. (1998). Att at TREC-7. In *Text REtrieval Conference*, pages 186–198. 177

Song, W., Zhang, Y., Liu, T. & Li, S. (2010). Bridging topic modeling and personalized search. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 1167–1175, Beijing, China. 154

Speretta, M. & Gauch, S. (2005). 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. 41, 146

Steinbach, M., Karypis, G. & Kumar, V. (2000). A comparison of document clustering techniques. In *Proceedings of Workshop on Text Mining, at The Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD 2000)*, Boston, MA, USA. 129

Sugiyama, K., Hatano, K. & Yoshikawa, M. (2004). 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, New York, NY, USA. 41

Tao, T. & Zhai, C. (2006). Regularized estimation of mixture models for robust pseudo-relevance feedback. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 162–169, Seattle, Washington, USA. 24

Tao, T., Wang, X., Mei, Q. & Zhai, C. (2006). Language model information retrieval with document expansion. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 407–414, New York, USA. 91, 177

Taylor, M., Guiver, J., Robertson, S. & Minka, T. (2008). Softrank: Optimizing non-smooth rank metrics. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, WSDM '08, pages 77–86, Palo Alto, California, USA. 157

Teevan, J., Morris, M.R. & Bush, S. (2009). 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, Barcelona, Spain. 36, 42, 43, 44

Theodora Tsikrika, J.K. (2008). Overview of the wikipediamm task at imageclef 2008. In *Working Notes for the CLEF 2007 Workshop*, Aarhus, Denmark. 56

Tsikrika, T. & Kludas, J. (2010). Overview of the wikipediamm task at imageclef 2009. In C. Peters, B. Caputo, J. Gonzalo, G. Jones, J. Kalpathy-Cramer, H. Muller & T. Tsikrika, eds., *Multilingual Information Access Evaluation II. Multimedia Experiments*, vol. 6242 of *Lecture Notes in Computer Science*, pages 60–71, Springer Berlin Heidelberg. 82

van Rijsbergen, C., Robertson, S. & Porter, M. (1980). *New models in probabilistic information retrieval*. London: British Library. (British Library Research and Development Report, no. 5587). 59

Vargas, S., Santos, R.L.T., Macdonald, C. & Ounis, I. (2013). Selecting effective expansion terms for diversity. In *Proceedings of the 10th Conference on Open Research Areas in Information Retrieval*, OAIR '13, pages 69–76, Lisbon, Portugal. 48

Walker, S., Robertson, S., Boughanem, M., Jones, G.J.F. & Jones, K.S. (1998). Okapi at trecâĂŞ6: Automatic adhoc, vlc, routing, filtering and qsdr. In *The Sixth Text REtrieval Conference (TRECâĂŞ6)*, 125âĂŞ136. 54

Wang, L. & Oard, D.W. (2009). Context-based message expansion for disentanglement of interleaved text conversations. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 200–208, Boulder, Colorado. 94

Wei, X. & Croft, W.B. (2006). Lda-based document models for ad-hoc retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 178–185, Seattle, Washington, USA. 12, 151, 153, 171

Wen, J.R., Dou, Z. & Song, R. (2009). *Personalized Web Search*, pages 2099–2103. Springer US, Boston, MA. 41

Westerveld, T. & van Zwol, R. (2007). The INEX 2006 multimedia track. In *N. Fuhr, M. Lalmas, and A. Trotman, editors, Advances in XML Information Retrieval:Fifth International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2006, Lecture Notes in Computer Science/Lecture Notes in Artificial Intelligence (LNCS/LNAI)*, Springer-Verlag. 57

White, R., Ruthven, I. & Jose, J.M. (2002). The use of implicit evidence for relevance feedback in web retrieval. In *Proceedings of the 24th BCS-IRSG European Colloquium on IR Research: Advances in Information Retrieval*, pages 93–109. 19, 24

Xia, F., Liu, T.Y., Wang, J., Zhang, W. & Li, H. (2008). Listwise approach to learning to rank: Theory and algorithm. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 1192–1199, Helsinki, Finland. 157

Xu, J. & Croft, W.B. (1996). 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*, SIGIR '96, pages 4–11, Zurich, Switzerland. 9

Xu, J. & Li, H. (2007). Adarank: A boosting algorithm for information retrieval. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 391–398, Amsterdam, The Netherlands. 157

Xu, J., Liu, T.Y., Lu, M., Li, H. & Ma, W.Y. (2008a). Directly optimizing evaluation measures in learning to rank. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 107–114, Singapore, Singapore. 157

Xu, S., Bao, S., Fei, B., Su, Z. & Yu, Y. (2008b). Exploring folksonomy for personalized search. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 155–162, Singapore, Singapore. 36, 39, 44, 122

Xu, Y., Jones, G.J. & Wang, B. (2009). Query dependent pseudo-relevance feedback based on wikipedia. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 59–66, Boston, MA, USA. 55

Yue, Y., Finley, T., Radlinski, F. & Joachims, T. (2007). A support vector method for optimizing average precision. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 271–278, Amsterdam, The Netherlands. 157

Zhang, M., Song, R., Lin, C., Ma, S., Jiang, Z., Jin, Y., Liu, Y. & Zhao, L. (2002). Expansion-Based technologies in finding relevant and new informa-

tion: THU TREC2002 novelty track experiments. In *the Proceedings of the Eleventh Text Retrieval Conference (TREC)*. 94, 95

Zheng, Z., Chen, K., Sun, G. & Zha, H. (2007a). A regression framework for learning ranking functions using relative relevance judgments. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 287–294, Amsterdam, The Netherlands. 157

Zheng, Z., Zha, H., Zhang, T., Chapelle, O., Chen, K. & Sun, G. (2007b). A general boosting method and its application to learning ranking functions for web search. In *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, pages 1697–1704. 157

Zhu, Y., Xiong, L. & Verdery, C. (2010). Anonymizing user profiles for personalized web search. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 1225–1226, Raleigh, North Carolina, USA. 41