ABSTRACT

INFORMATION FILTERING BY MULTIPLE EXAMPLES

by
Mingzhu Zhu

A key to successfully satisfy an information need lies in how users express it using keywords as queries. However, for many users, expressing their information needs using keywords is difficult, especially when the information need is complex. Search By *Multiple* Examples (SBME), a promising method for overcoming this problem, allows users to specify their information needs as a set of *relevant documents* rather than as a set of keywords.

Most of the studies on SBME adopt the Positive Unlabeled learning (PU learning) techniques by treating the user's provided examples (denoted as query examples) as positive set and the entire data collection in the database as unlabeled set. User's information need is then represented as a query vector, which is obtained from the query examples or further augmented with unlabeled data as negative examples, in which the documents are ranked according to their degree of similarity to the query vector. The query examples are treated as being relevant to a single topic to build the query vector, but it is often the case that they belong to multiple topics. New methods are needed to deal with such a topic diversity issue.

Furthermore, there are many PU learning algorithms available, but it is still unknown which methods perform most effectively for SBME, as the experiments conducted in the previous studies have not taken into account the user search situation, where the size of the query examples varies and is much smaller than the size of the unlabeled data. When the query examples are much fewer than the unlabeled data, the system effectiveness may downgrade dramatically because of the class imbalance problem. Thus, it is important to identify the most effective PU learning algorithms for SBME and explore how to improve the system effectiveness further.

In the previous studies on SBME, a document is usually treated as a vector, of which the features are terms in the collections. Such a term-vector based document representation brings high dimensionality problems when the collection is large; or even worse, some noisy features seriously degrade the performance of the learning algorithms. Feature selection is necessary for solving the high dimensionality problem.

This research proposes a framework named Information Filtering by Multiple Examples (IFME) to explore how to improve SBME by: (1) solving the topic diversity issue by adopting probabilistic topic models to predict user's information need from the query examples; (2) tackling the class imbalance problem by adopting machine learning techniques; (3) identifying the most effective PU learning algorithms for SBME, (4) adopting ensemble learning techniques to improve the effectiveness of the PU learning algorithms for SBME further; and (5) adopting topic model for feature dimension reduction. The experimental results show that the proposed framework addressed the research questions successfully.

INFORMATION FILTERING BY MULTIPLE EXAMPLES

by
Mingzhu Zhu

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

Department of Information Systems

May 2015

APPROVAL PAGE

INFORMATION FILTERING BY MULTIPLE EXAMPLES

Mingzhu Zhu

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

Dr. Lian Duan, Committee Member Date
Assistant Professor of Information Systems, NJIT

Dr. Vincent Oria, Committee Member Date
Associate Professor of Computer Science, NJIT

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

Dr. Yihong Zhao, External Committee Member Date
Assistant Professor, Division of Biostatistics, New York Univesity

BIOGRAPHICAL SKETCH

Author: Mingzhu Zhu

Degree: Doctor of Philosophy

Date: May 2015

Undergraduate and Graduate Education:

- Doctor of Philosophy in Information Systems,
  New Jersey Institute of Technology, Newark, NJ, 2015

- Master of Science in Electrical Commence,
  Wuhan University, Wuhan, Hubei, P. R. China, 2010

- Bachelor of Science in Electrical Commence,
  Wuhan University, Wuhan, Hubei, P. R. China, 2008

Major: Information Systems

Presentations and Publications:

Mingzhu Zhu, Wei Xiong and Y.F. Brook Wu. Learning to Rank with only Positive Examples, *Proceedings of the13th International Conference on Machine Learning and Applications*, 2014: 87-92.

Mingzhu Zhu, Chao Xu and Y.F. Brook Wu. Positive Unlabeled Learning to Discover Relevant Documents Using Topic Models for Feature Selection. *Proceedings of the 10th International Conference on Data Mining*, 21-24 July 2014: 169-175.

Mingzhu Zhu, Chao Xu and Y.F. Brook Wu. Topic Model based Query Intent Prediction for Search by Multiple Examples. *Proceedings of the 16th International Conference on Artificial Intelligence*, 21-24 July 2014: 365-371.

Mingzhu Zhu and Y.F. Brook Wu. Search by multiple examples. *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, 24-28 Feb. 2014: 667-672.

Chao Xu, Mingzhu Zhu, Y. Liu, and Y.F. Brook Wu. User Profiling for Query Refinement. *Proceedings of 20th Americas Conference on Information Systems*, 2014.

Chao Xu, Mingzhu Zhu, and Y.F. Brook Wu. Personalizing Query Refinement Based on Latent Tasks. *Proceedings of the 10th International Conference on Data Mining*, 2014.

Mingzhu Zhu, Y.F. Brook Wu, M. S. Vasavada, and Jason Wang. Learning to Rank Biomedical Documents with only Positive and Unlabeled Examples: A Case Study. *In Biological Data Mining and its Applications in Healthcare*, Chapter 13, World Scientific Publishing Company, 2013: 373-392.

Mingzhu Zhu, Chao Xu, and Y.F. Brook Wu. IFME: Information Filtering by Multiple Examples with Under-Sampling in a Digital Library Environment. *Proceedings of the 13rd International ACM/IEEE Joint Conference on Digital Libraries*, 22-26 July 2013: 107-110.

This dissertation is dedicated to my beloved family

To my parents, parents-in-law, brother and sister,
My beloved wife, Kwan Chiu and my son, Alvin

本博士论文特此献给我的家人

我的父母，岳父母，弟弟和妹妹，
我亲爱的妻子胡昀炤与儿子朱彦旭

ACKNOWLEDGMENT

First, I would like to express my deepest appreciation to my research advisor, Dr. Yi-fang Brook Wu, for her insightful guidance, support, encouragement, kindness and enthusiasm during the course of this work and my entire graduate study at NJIT.

I would also like to thank the rest of my thesis committee members, Dr. Lian Duan, Dr. Vincent Oria, Dr. Songhua Xu, and Dr. Yihong Zhao, for their valuable comments, insightful questions and actively participating in my committee. It is my great honor to have the committee's guidance and help to complete this dissertation.

In addition, I would like to thank my fellow graduate students such as Chao, Ye, Yanchi, Chris, and Regina at Earth Lab and CoLab for the interesting discussions.

Finally, I am grateful to all the colleagues such as Dr. Michael Bieber, Dr. Richard Egan, Dr. Lin Lin, Dr. Quentin Jones, and Dr. Toll Will in the Information Systems department at NJIT for their support and encouragement.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

Chapter **Page**

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

Chapter **Page**

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

LIST OF FIGURES

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

1.1 Background

Information overload is a common problem with the volume of digital information being created increasing exponentially. This is especially true for researchers who need to keep up with the relevant literature in their research domain. For instance, the growth of biology literature makes it almost impossible for a researcher to obtain all the publications of interests manually, even on specialized topics (Jensen, Saric, & Bork, 2006).

Traditionally, most researchers keep themselves up to date with the literature by using keyword-based Information Retrieval (IR) systems such as Google Scholar, ACM digital libraries and PubMed. A key to the success of using these modern keyword-based IR systems to satisfy an information need lies in how users express it using keywords as queries. However, expressing an information need using a simple string of keywords is often difficult, especially when the information need is complex. There are also many cases where the keyword-based search is unsatisfactory, as most of the knowledge in the modern era is a combination of concepts, topics, methods, and ideas. For instance, with the growing popularity of interdisciplinary studies, researchers often need to search articles related to multiple domains (e.g., searching articles on using machine learning techniques for entity extraction in bioinformatics), it would be difficult for them to express the information need using a simple string of keywords. To make matters worse, this task can be even more difficult to accomplish when users are not familiar with the domain of the articles of interest, thus it is hard to compose the appropriate keyword-based queries.

Theoretically, a search task can be accomplished by using a keyword-based search engine, but depending on the complexity of the information need, the information seeking process might require a few iterations of searching and browsing of the search results.

To overcome the shortcomings of the keyword-based search, a new search paradigm named "Search By Multiple Examples" (SBME) (El-Arini & Guestrin, 2011; Zhang & Lee, 2008, 2009; Zhu, Xu, & Wu, 2013) has been proposed to enable users to express their information needs using multiple relevant examples (denoted as query examples). For instance, in the article search scenario, having already read some papers on a specific topic, the researchers can use these examples as queries to retrieve more relevant articles.

SBME can also be applied in other scenarios where there are some relevant documents available and more relevant documents are needed. For instance, by considering user's clicked links as relevant information, a search engine could recommend the user with the most relevant ads from the ad collection (Radlinski et al., 2008).

1.2 Motivation

The previous studies on SBME can be classified into three categories, according to the methodologies of adopting the query examples for document ranking. All of these methods have limitations, and thus motivate this research.

El-Arini and Guestrin (2011) propose concept networks that are built using citation based graph analysis techniques, for article recommendation. A directed, acyclic graph is defined for each concept in the document collection to model how ideas travel between documents. Then the influence between any two articles in the document collection is defined in a probabilistic, concept-specific notion. The main limitation of the graph analysis based method is that some important and relevant documents for an information need may never be cited as they might be new articles or their importance might not be identified by the community.

Zhang and Lee (2008, 2009) adopt Positive Unlabeled learning (PU learning) models by considering the query examples as positive data and the documents in the database as unlabeled data. The PU learning based method can be used in a broader area, especially, when the query documents are formed by documents without citations (e.g., abstracts or paragraphs), but it suffers several limitations, which motivate this study to explore new methodologies to improve the performance of SBME further.

One major problem of the PU learning based studies on SBME is that all the documents in the online database are considered as unlabeled data for model training. Not only do such methods suffer efficiency issues as the high volume of the unlabeled data can reduce the speed of a PU learning algorithm for document ranking, but also that the system effectiveness may downgrade dramatically because of the class imbalance problem: the size of the positive examples P (denoted as |P|) is much smaller than the size of unlabeled data U (denoted as |U|) (Ling & Sheng, 2010).

Another method for SBME is the term selection method (i.e., the centroid method) which selects important terms from the query examples to build query vectors for conducting traditional keyword-based search (Rocchio, 1971). Although this method is efficient as the terms can be used directly to conduct standard keyword-based search using modern IR systems, it may be biased as the query examples are considered as belonging to a single topic, which may actually belong to multiple topics. For instance, a set of documents related to “Information Retrieval” may contain some other topics such as “business” or “biology” as the background information. It is biased to build query vectors from the query examples without considering the topical diversity issue.

Figure 1.1 provides an example to show the topical diversity issue of query examples. For illustrative purpose, the vectors of the query documents are represented in a three dimensional space, where Ta, Tb, and Tc denote the terms or features of the vectors.

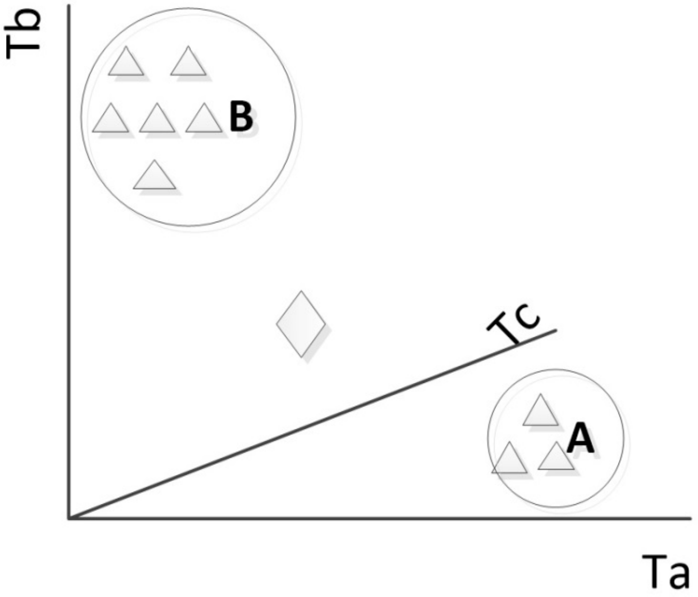


Figure 1.1 Topic diversity of query examples.

A triangle denotes the term vector of a query example, and the diamond denotes the centroid of the vectors of the query examples, which is used as the query vector in the centroid method. Each axis denotes the *tfidf* value of the corresponding term. Based on the topics covered in the query documents, they can be classified into two categories (topics): group A and group B. This indicates that the documents of interests are relevant to the two topics, respectively. However, the centroid method does not consider the topic diversity issue by simply using the centroid of the query vectors to build the query vector. As there are more examples in group B than in group A, it is more likely that the terms in group B are more representative of user's true information needs than the terms in group A. This research attempts to solve the topic diversity issue by adopting topic modeling techniques to predict user's information needs from the query examples.

In previous studies of PU learning, P and U are usually balanced. In SBME setting, it is often the case that the query examples from a user are much fewer than the documents in the unlabeled data, i.e., all the documents in the database. So the data for training a PU learning model is imbalanced: the positive examples are much fewer than the unlabeled data. The system performance can downgrade dramatically because of this class imbalance problem (Ling & Sheng, 2010). This research proposes an under-sampling based approach to solve the class imbalance problem in a PU learning based SBME framework.

Furthermore, there are many PU learning algorithms available (Calvo, Larrañaga, & Lozano, 2007; Denis, Gilleron, & Tommasi, 2002; Elkan & Noto, 2008), but it is still unknown which PU learning algorithms have the best performance for SBME in terms of effectiveness and efficiency. For instance, some PU learning algorithms such as bias-SVM (Liu, Dai, Li, Lee, & Yu, 2003) have the state-of-the-art performance in terms of effectiveness, but they are too inefficient to be applied for online search. This research tries to bridge this gap by proposing a two-stage based framework to improve the efficiency of PU learning based SBME, and studying which PU learning algorithm performs most effective in the proposed framework. Then ensemble learning is incorporated to improve the system effectiveness further.

Most PU learning algorithms are based on the Vector Space Model, where a document is treated as a vector, of which the features are terms in the collections. Such a term-vector based document representation brings high dimensionality problems; or even worse, some noisy features seriously degrade the performance of the learning algorithms. Therefore, it is important to conduct feature selection to improve the efficiency as well as the effectiveness of PU learning algorithms. This research proposes to adopt a topic model to transfer the documents into topic vectors and examines whether this method can outperform the traditional feature selection methods such as CHI in terms of effectiveness and efficiency.

In summary, the aim of this research is to assist users to search documents using multiple examples to express their information needs. A series of studies were conducted to explore how to improve the performance of SBME by 1) using topic analysis techniques to predict user's information need, 2) studying how to solve the class imbalance problem in the PU learning based SBME system to rank the positive documents in the potentially relevant documents from the database as high as possible, especially when |P| (the size of query examples P) is small, 3) identifying the state-of-the-art PU learning algorithms for SBME, 4) adopting ensemble learning techniques to improve the system effectiveness further, and 5) examining whether topic model based feature selection method can improve the performance of SBME in terms of effectiveness and efficiency.

This section described the motivation of this research. The following section will present the proposed solution and the research questions.

1.3 The Proposed Framework

1.3.1 Research Framework

This research intends to explore how to improve the performance of SBME through the design and development of a framework, named Information Filtering by Multiple Examples (IFME), which incorporates topic modeling, PU learning, under-sampling, and ensemble learning in information filtering on document collections. The proposed IFME framework aims to assist users to filter out irrelevant documents from document collections using multiple documents to represent an information need, which may be difficult to express using a simple string of keywords. Different from the previous studies on SBME, where no topic analysis has been conducted on the query examples, the proposed study adopts a topic model to analyze the topic distribution of the query examples to solve the topic diversity issue. Previous studies on SBME are conducted following the inductive supervised learning scenario, where it is assumed there is a large number of positive examples and the size of the positive data and the unlabeled data are similar. The proposed IFME framework is following the transductive learning paradigm in information retrieval, where the size of the positive data (i.e., user's input query examples) is much smaller than the unlabeled data (i.e., all the potential relevant documents in the database). This research also seeks to investigate how the proposed methods perform in a simulated user search situation, where |P| is varying and is much smaller than |U|.

The research framework, illustrated in Figure 1.2, consists of four parts: 1) topic model based information need modeling, 2) data transformation, 3) adopting under-sampling to solve the class imbalance problem, 4) PU Learning algorithms improvement for SBME, and 5) performance evaluation.

1.3.1.1 Topic Model Based Information Need Modeling. To address the topic diversity issue, this study tries to utilize topic analysis techniques to better model user's information need from the query examples. This study has two assumptions: 1) the true information need consists of some aspects of the query examples, and 2) each document can be related to more than one topic. Based on these two assumptions, a topic model is used to predict the probabilities that the query examples belong to a certain topic. Then the query vector is built using the latent topic distribution information for document ranking. To accomplish this research goal, a method using topic model for query intent prediction from multiple examples is proposed. The proposed method was compared with the baselines, i.e., centroid method and the *tfidf* method, which do not take into account the topic diversity issue for modeling the information need from the query examples. The experimental results show that the topic based method can solve the topic diversity issue effectively. The proposed method was also compared with another two baselines: Skyline based method and FANN, which have been proposed in the database domain to identify the top-k most similar objects to a group of query objects.

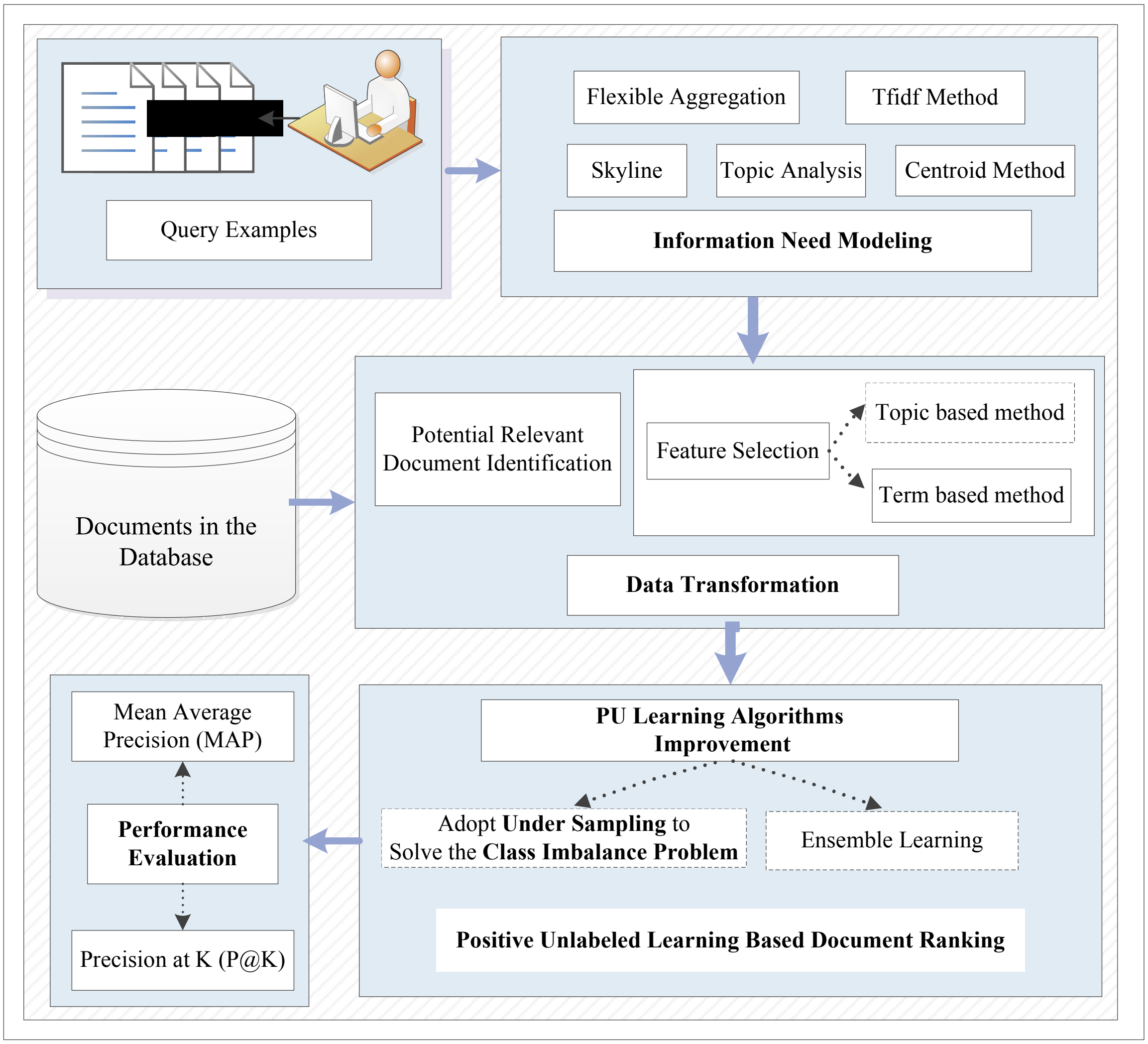

[redacted]
Query Examples
Flexible Aggregation
Tfidf Method
Skyline
Topic Analysis
Centroid Method
Information Need Modeling
Documents in the Database
Potential Relevant Document Identification
Feature Selection
Topic based method
Term based method
Data Transformation
Mean Average Precision (MAP)
Performance Evaluation
Precision at K (P@K)
PU Learning Algorithms Improvement
Adopt Under Sampling to Solve the Class Imbalance Problem
Ensemble Learning
Positive Unlabeled Learning Based Document Ranking

Figure 1.2 The research framework.

1.3.1.2 Data Transformation. To rank documents using PU Learning algorithms, potential relevant documents need to be identified from the database. Although all the documents in the database can be used as unlabeled data, it is inefficient as the size of the documents in a commercial database can be huge.

There are two stages involved in the PU learning based ranking methods: 1) document preprocessing, 2) using PU Learning algorithms to rank the unlabeled data. In the first stage, documents are usually transformed into term vectors after feature selection and feature weight determination. In the second stage, PU Learning algorithms are applied on the prepared data (i.e. term vectors) for learning classifiers and making predictions on the unlabeled data. Thus, the key for a given PU learning algorithm to achieve good performance is to select a set of good features and use appropriate feature weighting methods.

Tfidf method, one of the most widely used feature selection methods in Text Mining and Information Retrieval, is a standard method to compare the performance of different ranking algorithms. It is based on the Vector Space Model, where a document is treated as a vector, of which the features are terms in the collections. Such a term-vector based document representation brings high dimensionality problems; or even worse, some noisy features seriously degrade the performance of the learning algorithms. So it is important to select a small number of features to improve the system performance in terms of efficiency and effectiveness. This study examines how the standard feature selection methods (e.g., Chi square test) perform in the IFME framework; and tests whether using topic model for feature selection can achieve better performance than the standard feature selection method in terms of effectiveness and efficiency.

1.3.1.3 Adopting Under-sampling to Solve Class Imbalance Problem. The previous studies on PU learning based SBME assume that the dataset for the model training is balanced, which is usually not the case. It is high likely that the size of the query examples is much smaller than the size of the documents in the database, thus the system performance can downgrade dramatically because of the class imbalance problem (Ling & Sheng, 2010).

This research proposes a novel method that combines under-sampling and ensemble learning to solve this issue, thus the effectiveness of the PU learning based SBME system can be improved. Specifically, K percent of the unlabeled data is sampled N times for training with the positive set (query examples) to build N models, which is used to rank the documents in the unlabeled data. The final rank of an instance in U is generated using the rank information from the N runs. Experiments were conducted on three benchmark datasets to evaluate the effectiveness of the proposed method using different sizes of query examples.

1.3.1.4 PU Learning Algorithms Improvement for IFME. The previous studies using PU learning algorithms for SBME consider the entire data collection as unlabeled data. This is inefficient as the size of the entire data collection is too large. This research proposes a two-stage based framework by first reducing the search space through identifying potentially relevant documents using a standard keyword-based search system. Then a PU learning model is trained by considering the query examples as positive data and the potentially relevant documents as unlabeled data, which are in turn ranked by the model.

There are many PU learning algorithms available, but it is still unknown which is most effective in the IFME framework. This research tries to investigate the effectiveness of two state-of-the-art PU learning algorithms for IFME in the two-stage based framework. They have been shown effective in text classification in terms of F measure (Liu, 2007). However, it is still unknown how they perform for document ranking in the IFME setting, where the size of P is small (i.e., |P|=2). In addition, the classification algorithms that are shown effective in terms of F measure, which is a popular evaluation measure in text classification, may perform poorly for document ranking in terms of Mean Average Precision (MAP) and precision at k (*p@k*), which are the standard methods to evaluate ranking systems (Manning, Raghavan, & Schütze, 2008). This study tries to bridge this gap by conducting extensive experiments to identify the most effective PU learning algorithms for IFME using the standard methods for evaluating ranking systems.

Ensemble learning has been shown effective by combing different learning algorithms in traditional text mining area. This research tries to explore whether it is effective to adopt ensemble learning to improve IFME by taking advantage of different PU learning algorithms.

Using topic models, each document can be transferred into a topic distribution. Since the number of topics needed in topic modeling is much smaller than the size of terms in the collection, converting documents into topic vectors can potentially improve system's efficiency. This research tries to explore whether using topic models for feature selection can improve the system performance in terms of efficiency and effectiveness.

1.3.1.5 Performance Evaluation. The experiments conducted in the previous studies have not taken into account the real user search situations, which are very different from the controlled experiments scenario. For instance, in online search, different users may provide different numbers of query examples for the same information need. Even for the same user, the number of query examples can be different for different information needs. To evaluate the performance of IFME framework, most of the previous studies simply use a large number of relevant examples to simulate user's queries. This is biased, as the performance of certain models, which perform well when they are built on a large size of training data, may perform poorly when they are built on a small size of training data. In order to adopt PU learning for IFME, it is important to identify which PU learning algorithms perform well in online settings, where it is often the case that only a few query examples are available. In this study, different sizes of query examples were generated to simulate user's search behaviors to better evaluate the proposed methods.

1.3.2 Research Questions

The primary goal of this research is to address the following research questions:

1) How can topic analysis techniques be used to predict user's true information need from the query examples?
2) How can the class imbalance problem of adopting PU learning in IFME be solved using machine learning techniques?
3) Which PU learning algorithms perform most effective for SBME in terms of Mean Average Precision (MAP) and precision at k (p@k)?
4) Whether ensemble learning can be adopted to improve the effectiveness of the PU learning algorithms in the proposed IFME framework?
5) Whether topic model based feature selection method can improve the performance of SBME in terms of effectiveness and efficiency?
6) How does the size of query examples affect the system's effectiveness?

Experiments have been conducted on three benchmark datasets, and the experimental results show that the proposed approaches outperform baselines across all studies. More details can be found in subsequent chapters.

1.4 Dissertation Outline

The remainder of this dissertation is organized as follows. Chapter 2 presents an overview of the previous work on SBME and related studies. Chapter 3 describes the proposed method of adopting topic modeling for solving the topic diversity issue. Chapter 4 presents combining under-sampling and ensemble learning to solve the class imbalance problem in a PU learning based IFME framework. Chapter 5 describes the evaluation of adopting the state-of-the-art PU learning algorithms for IFME and presents using ensemble learning to improve the system effectiveness. Chapter 6 describes the topic model based feature dimension reduction method. Chapter 7 provides summaries, limitations, contributions and future directions of this study.

CHAPTER 2

LITERATURE REVIEW

The main goal of this study is to improve the effectiveness of SBME through adopting topical modeling, Positive Unlabeled learning (PU learning), under-sampling and ensemble learning. As an alternative to the keyword-based search, SBME aims to assist users to express their information needs using multiple examples. Section 2.1 presents an overview of keyword-based search, in which the classic techniques for keyword-based search are described. Section 2.2 presents an overview of SBME. The remaining part of the chapter describes the related machine learning techniques adopted in this research.

2.1 Keyword-based Search

Information Retrieval (IR) concerns the obtaining of information resources to an information need from a data collection (Manning et al., 2008). In the era of information overload, automated information retrieval systems such as search engines become the most widely used tools for people to satisfy their information needs. In order to use an IR system, an information need is usually represented as one or a series of queries to match the data collection, which can be structured data or unstructured text documents.

With the development of Word Wide Web, information retrieval involves several tasks and applications, which includes text search (Barbic & Choy, 1989; Eiron & McCurley, 2003; Salton & Buckley, 1988; Zobel & Moffat, 2006), multimedia search (Maybury, 1997; Rui, Huang, & Chang, 1999; Rui, Huang, & Mehrotra, 1997; Schmid & Mohr, 1997) and other media search (Eyal & Aposporos, 2004; Foote, 1999; Wang, 2003).

Text search is one of the most popular tasks with the success of the commercial search engines such as Google, Yahoo and Bing. In text search, the usual search scenario is that a user submits a query that is a formal statement of an information need, to a search engine, which will return a list of documents in a ranked order.

In traditional IR, a query to conduct a search is usually in one of two forms: a sentence or a list of terms. Such a search paradigm is called as keyword-based search as the queries are expressed as a set of keywords (Wei & Croft, 2006; Xu & Papakonstantinou, 2005).

The core of keyword-based search is to model how people compare texts and design computer algorithms to accurately perform this comparison. The documents are ranked based on the extent to which they are relevant to the query. Simply speaking, a desirable relevant document contains the information that the user, who submits the query to a search engine, is looking for. As the same concept can be expressed in different words, simply comparing the text of a query with the text of a document to conduct exact match usually produces very poor results. To address this issue, many retrieval models have been proposed and tested their effectiveness (Fuhr, 1992; Raghavan & Wong, 1986; Song & Croft, 1999).

A retrieval model defines the matching process between a query and a document, which is the basis of the ranking algorithm that is used by a search engine for ranking search results. In modern information retrieval, retrieval models are usually based on the statistical properties of text rather than the linguistic structure (Croft, Metzler, & Strohman, 2010). For example, ranking algorithms are typically designed based on the frequency of word occurrences rather than whether the word is a noun or a verb (Salton & Buckley, 1988). Although some models do incorporate linguistic features, they are proved less important.

The two most popular models are Boolean Model (BM) and Vector Space Model (VSM). This section briefly introduces Boolean Model and describes the VSM in detail, which is the foundation of this research.

BM is one of the simplest and earliest retrieval models. Because of its simplicity and neat formalism, BM is one of the most popular methods adopted in the early commercial bibliographic systems. Based on the set theory and Boolean algebra, it adopts AND, OR or NOT Boolean operators to join the query terms (Turtle & Croft, 1992). One drawback of BM is that it is often difficult to translate an information need into a Boolean expression. In fact, ordinary users, who are not well trained in Boolean algebra, often find it hard and awkward to express their queries in Boolean expressions.

In BM, the index terms are considered either present or absent in documents, thus the term weights are assumed to be all binary. As a result, each document is predicted as either relevant or non-relevant. This exact matching often excludes the documents that are relevant to user's interests but use terms different from those in the query. To overcome the exact matching issue, term weighting has been proposed and shown substantial improvement in retrieval performance. This leads to the popularity and success of the vector space model.

2.1.1 Vector Space Model

Different from BM, Vector Space Model (VSM) proposes a framework for term representation and similarity measure between documents thus partial match is possible

(Salton, 1971; Salton & Buckley, 1988). In VSM, the terms are assigned non-binary weights to the index terms in queries and documents to measure their degree of importance. The term weights are ultimately adopted to calculate the degree of similarity between the user's query and a document in the system. The retrieved documents are then sorted in decreasing order of the similarity values. In this case, it is feasible for VSM to identify documents that match the query terms only partially.

In VSM, each document and a user's query are represented as k-dimensional vectors. Each item of the vector represents a term in the document or the query. The term can be a single word or a phrase. For instance, a document vector d_j can be expressed as:

$$d_j=<w_{1j},\ w_{2j},\ w_{kj}>$$

where w_{ij} $(1 \leq i \leq k)$ is the weight associated with the *term* t_i in the j_{th} document, which is a non-negative value measuring the degree of importance of term t_i in this document, and K is the number of terms in the system.

2.1.2 Term Weighting

There are many approaches for assigning weights to the terms of a document vector. The most popular one is the *tf-idf* method. Here, the term frequency *tf*(*t, d*) is defined as the number of times the term *t* occurs in the document *d*. The higher the value of *tf*(*t, d*), the more important *t* is in *d*. The problem of simply using term frequencies for term weighting is that it does not take into account the document collection size and how the terms are distributed in the collection. In other words, for a particular document, if two terms have the same term frequency, the term that appears in a smaller number of documents in the collection is more important than the other. As the more documents in the collection a term appears, the less likely it is able to distinguish the documents. Thus it is necessary to consider the number of documents a term appears when calculating its weight. The inverse document frequency *idf*(*t*) is proposed to indicate how important a term *t* is in distinguishing the documents in a collection of documents. The inverse document frequency is calculated using the following formula:

$$idf(t) = \log (N/df(t))$$

where N is the total number of documents in the collection, and $df(t)$ is the document frequency of t, which is defined as the number of documents containing t.

The *tf-idf* value of the term *t* is defined as the product of its *tf* and *idf* values, i.e.

$$tf\text{-}idf(t,d) = tf(t,d) \times idf(t)$$

This formula means the importance of a term in a document increases when its frequency in the document increases, and decreases when its document frequency increases.

Different weight schemes have been investigated in the previous studies (Manning et al., 2008; Zobel & Moffat, 1998), and the best results are obtained by using the tf-idf method based on precision and recall. In this research, tf-idf method is adopted for term weighting.

2.1.3 Similarity Calculation

To calculate the degree of similarity between a query and a document in the system, two types of similarity measures are used most: the distance-based method and the angular-based method. The former assumes that documents close to each other in the vector space are likely to be similar; the latter assumes that document vectors in the same direction are likely to be similar. Other similarity measures (Zobel & Moffat, 1998) such as Jaccard coefficient, Overlap formulation, Dice formulation have also been proposed, but they are used less often and their performance is not as good as the distance-based measure or angular-based measure.

2.1.4 Evaluation

Once the similarity values between a query and each of the documents are calculated, the documents will be ranked in the decreasing order of the similarity values. Since different ranking algorithms may provide different retrieved results for the same query, it is necessary to judge the effectiveness of a ranking algorithm. The most widely used measures are precision and recall (Buckland & Gey, 1994). Precision is the proportion of retrieved documents that are relevant, and recall is the proportion of relevant documents that are retrieved. As most of the retrieval models produce ranked output, to summarize the effectiveness of a ranking, precision and recall values are calculated as every rank position or at fixed recall levels from 0.0 to 1.0 in increments of 0.1.

The most popular method to summarize the effectiveness of a ranking is averaging the precision values from the raking positions where the retrieved document is relevant. For a ranking algorithm, as many queries will be conducted, its effectiveness is summarized as the Mean Average Precision (MAP). In modern IR, as the volume of documents can be huge, users may only be able to screen the top ranked documents. *P@k* (precision at position K) is more and more used for IR systems evaluation (Manning et al., 2008).

Keyword-based search is one of the most widely adopted search paradigms because of the success of commercial search engines, but it is often difficult for users to express their information needs simply using a single string of keywords. It is necessary to provide users with other methods for information need expression. SBME has been proposed recently to enable users to express their information needs using multiple relevant documents. The following sections will present SBME and other studies related to this research.

2.2 Search by Multiple Examples

Different from the traditional keyword-based search, SBME allows users to express their information needs using a set of relevant examples rather than a set of keywords. Although it is a valuable research topic, there are only very few studies in the literature.

2.2.1 Citation Graph Analysis based Method

In the study by El-Arini and Guestrin (2011), the entire list of references of a group of papers is adopted to find related publications. The query is defined as a small set of papers that are relevant to the research task at hand. More relevant articles are selected by "optimizing an object function based on a fine-grained notion of influence between documents." The authors define a directed, acyclic graph for each concept in the document collection to capture how ideas travel between documents. In the graph, nodes represent articles that contain the concept and the edges represent citations and common authorship. The degree of influence is captured by calculating a weight for each edge between two nodes. The weight of the edge represents the probability of direct influence from the head to the tail of the edge with respect to the concept that is contained by the articles (nodes) in the graph. Given these edge weights, they define a probabilistic, concept-specific notion of influence between any two articles in the document collection. One concern of this method is that not all documents are cited; because newer articles' citations will not be known until the articles citing them are published. Thus, this approach is ill-equipped to find newly released relevant documents, and it does not work with documents without links or citations.

2.2.2 PU Learning based Method

Another direction for SBME is based on PU learning. Employing the same idea of using multiple examples to express user's information needs, Zhang and Lee (2008, 2009) conduct studies on SBME from the perspective of PU learning in two scenarios: online queries and offline queries. For the online queries, they propose a one-class SVM learning from the query set; for offline queries, they propose to learn an SVM model from the query set as well as the entire corpus. Not only does this method have efficiency issues because of the high volume of the unlabeled data, but also that the system effectiveness may downgrade dramatically because of the class imbalance problem.

The term selection methods (e.g., centroid method) are more efficient than the PU learning based methods as the terms can be used directly to conduct standard keyword-based search using modern IR systems; however, these methods have the topic diversity issue as the query examples are considered only belonging to a single topic, which may actually belong to multiple topics.

One goal of this study is to solve these issues through generating high quality queries from the query examples by employing topic modeling techniques to represent user's true information needs. Then query vectors are built using the topic distribution information to rank documents in the online database directly.

Besides, the performance of one-class SVM is extremely poor (X. Li & Liu, 2003), especially when |P| is small, as it does not take advantage of the information in the unlabeled instances, which is helpful for improving classifier training. Thus the one-class SVM is actually not a good approach for online search. This research proposes a two-step approach by first reducing the scope of the unlabeled data from the entire data collection to a subset of the data collection, then training a more effective PU learning model on the smaller subset. This study also tries to identify more robust PU learning algorithms for IFME and adopt ensemble learning to improve the system performance further.

Duh and Kirchhoff (2008) present a transductive learning framework to explore how to improve ranking performance using partially labeled data. They adopt KernelPCA (Schölkopf, Smola, & Müller, 1998) to generate better features from the unlabeled data and use the features via Boosting for learning different ranking functions adapted to the individual test queries. However, as this transductive learning framework needs both positive and negative training data, it is not applicable in the setting where users only have positive examples.

In addition, the previous studies on SBME take no consideration of the change of |P| for evaluation. The PU learning algorithms perform well with a large positive set may perform poorly when |P| is small. Different from the experiments conducted by other researchers (Duh & Kirchhoff, 2008; Zhang & Lee, 2009), which use large balanced training data, this study intends to better evaluate the proposed methods by manipulating the size of P to simulate user's search situation in online search.

2.2.3 Aggregation Similar Search

SBME has also been investigated in the database domain (Borzsony, Kossmann, & Stocker, 2001; Y. Li, Li, Yi, Yao, & Wang, 2011). Specifically, SBME is related to aggregate similarity search, in which a group of query instances (examples) Q is used to retrieve the most (top-k) similar object from the underline database P, where an aggregation function is used to calculate the similarity between each object in the database and the query objects. Since a user's information need is also represented as multiple examples in aggregate similarity search, this section briefly overviews two popular methods in this domain: Skyline operation based method and Flexible Aggregate Similarity Search.

The Skyline based method tries to identify the top ranked instances from the database using the Skyline operation (Borzsony et al., 2001). Skyline analysis has been shown useful in many applications such as multi-criteria decision making, top-K queries and nearest neighbor search. Given a set of N points P_1, P_2, P_3, … , P_N in a d-dimensional space, the Skyline operator returns the points P_i (1<=i<=N) such that P_i is not dominated by any other point. The "dominate" relationship between a point *p* and another point *q* is defined as follows: *p* dominates *q* if *p* is as good as or better than *q* in all dimensions and better in at least one dimension.

In this study, the effectiveness of adopting Skyline operation for Search By Multiple Examples is studied. For each document q_i in the query examples with M documents, a document p_j in the database can be assigned a score S_{ji}, (e.g., the distance between q_i and p_j), so p_j can be represented as an M-dimensional vector: p_j=<S_{j1},S_{j2},…,S_{jM}>. Then the skyline operation can be conducted on the points p_i (1<=i<=N) in the M-dimensional space. By applying the Skyline operation, a set of points Sk(P) will be returned, which can be ranked using the traditional ranking methods such as the centroid method. Then the points Sk(P) are removed from the set P, and the same process is conducted on the remaining points in P iteratively. Let Sk_i(P) denote the points returned by the i_{th} Skyline operation. By ranking the points in SK_i(P) higher than the points in SK_j(P) when i<j, all the points in P can be ranked. Since this process is time consuming, the process ends when the top K (e.g., K=30) documents are identified and ranked.

Flexible Aggregate Similarity Search (FANN)(Y. Li et al., 2011) generalizes Aggregation Nearest Neighbor (ANN)(Papadias, Tao, Mouratidis, & Hui, 2005; Razente, Barioni, Traina, Faloutsos, & Traina Jr, 2008; Yiu, Mamoulis, & Papadias, 2005), which aims to retrieve the top-K similar instances to a query Q with M instances from the database P. In ANN search, the similarity is an aggregation (e.g., sum or max) of the distances between an instance p in the database and all the instances in Q. FANN generalizes ANN by defining the similarity as an aggregation over the distances between p and any subset of ø|Q| instances in Q for some support 0< ø <=1. In ANN search, all objects in the query objects are used to define the optimal query answer, which requires an object in P must be similar to all objects in Q, which could be too restrictive in practice. FANN allows the user to specify a support 0< ø <=1, and aims to identify the top-k objects in P that are the most similar to any ø$|Q|$ objects in Q. So, when ø =1, the FANN problem reduces to the ANN search problem.

The FANN problem is normally defined as follows: given a set of points P in the database, and a set of query points Q, where $|P|=N$ and $|Q|=M$. Let $\delta(p,q)$ define the distance for any two points p and q. Let g be the aggregation function (e.g., sum), and ø be the support value in (0,1]. Then g(p,S), the aggregation distance between a point p and a group of points S can be defined as:

$$g(p, S)=g(\delta(p,q_1), \ldots, \delta(p,q_{|S|})), q_i \epsilon S \text{ for } i = 1,\ldots, |S|.$$

Given, P, Q, δ, g, and ∅, a FANN query returns:

$$(p^*, Q_\emptyset^*) = \underset{p\epsilon P, Q_\emptyset \subseteq Q}{argmin}\ g(p, Q_\emptyset), \text{ where } |Q_\emptyset| = \lceil \emptyset M \rceil$$

Let $r^* = g(P^*, Q_\emptyset^*)$ denote the optimal aggregate distance, and $Q_\emptyset^p$ be the $|Q_\emptyset|$ points in Q that are closest to a point p in P, the FANN problem can be stated as finding

$$p^*= \underset{p\epsilon P}{argmin}\ r_p, \text{ where } r_p = g(p, Q_\emptyset^p).$$

The most straightforward method for answering a FANN query is to scan all points in P. For each point p in P, the points $Q_\emptyset^p$ (the $|Q_\emptyset|$ nearest neighbors of p in Q) are identified to calculate r_p. Then the p* is the point with the smallest r_p. Since each point p is associated with an r_p value, the points P can be ranked based on r_p in ascending order.

When the size of P is large, it is too expensive to do a linear scan of all points in P. The Threshold Algorithm (TA) (Fagin, Lotem, & Naor, 2003) is a more efficient method to identify the top-k objects from P. For each point q_i in Q, a list L_i is built, in which all the points in P are ranked based on their distances to q_i. Let $\delta(p(L_i),q_i)$ denote the distance between p and q_i in the *i*th list L_i, p can be viewed as an object with M (the size of Q) attributes with its *i*th attribute taking value $\delta(p(L_i),q_i)$. Then the aggregated score of $r_p = g(p, Q_\emptyset^p)$ can be calculated using the $|Q_\emptyset|$ smallest attribute values of p.

The previous studies mainly focus on the efficiency issue of FANN, but it is still unknown how FANN performs for document search in terms of effectiveness (i.e., p@k). In this study, experiments have been conducted to compare the performance of FANN with other methods in terms of p@k.

2.3 Positive Unlabeled Learning

Text classification is an important problem in machine learning and information retrieval. The classic approach for text classification is to first manually label a set of documents, which are called training data. Then a classification model is trained on the labeled data to build a classifier for assigning the predefined labels (categories) to future instances. This approach is called supervised learning as all of the training documents have been labeled before the training process. The main issue of supervised learning is that a large number of manually labeled training instances are needed for training an accurate model. On one hand, the labeling process is labor intensive and time consuming. On the other hand, the labeled data may not represent the data that need to be classified; especially the manually collected negative instances most likely do not represent all negative instances in the entire data collection (Liu, 2007).

In recent years, researchers have studied using unlabeled data to improve the performance of the supervised learning algorithms. The specific approach of using a small size of labeled set and a large size of unlabeled set is called semi-supervised learning, where positive instances, negative instances and unlabeled data are needed. As a special case of semi-supervised learning, the PU learning works on only a positive set P and an unlabeled set U (Altun, Belkin, & Mcallester, 2005; Calvo et al., 2007; Liu, Lee, Yu, & Li, 2002; Nigam, McCallum, Thrun, & Mitchell, 1998; Yu, Zhai, & Han, 2003). It aims to identify the hidden positive documents or negative documents from the unlabeled data (denoted as NU).

Nigam et al. (1998) use a small set of labeled instances and a large set of unlabeled instances to build a classifier. When both positive and unlabeled data are available, the positive training data can be used to estimate the positive class' conditional probability, $p(x|+)$, and the unlabeled data can be used to estimate $p(x)$. With the prior $p(+)$, which is known or can be estimated using data from other sources, the negative class' conditional probability can be obtained as follows:

$$p(x \mid -)=(p(x)-p(+)p(x \mid +))/(1-p(+))$$

Denis et al. (2002) adopt the conditional probability $p(x|-)$ to perform text classification with Naive Bayes methods. Liu et al. (2002) adopt an Expectation Maximization (EM) algorithm and Naive Bayesian classification method to separate positive and negative instances.

Yu, Han, and Chang (2004) propose a mapping-convergence algorithm for PU learning. There are two stages in their algorithm: mapping stage and convergence stage. In the mapping stage, they perform initial approximation of reliable negative instances (denoted as RN). In the convergence stage, they iteratively run an internal classifier that maximizes margins to progressively achieve the true boundary of the positive class in the feature space.

Most PU learning algorithms belong to such a two-stage strategy category. For instance, Liu et al. (2003) propose to identify RN using Rocchio classifier in the first step.

In the second step, a set of SVM classifiers run iteratively to identify the best classifier. They show that this method (denoted as RcSVM) outperforms the previous PU learning algorithms significantly.

Another direction of using PU learning is identifying positive instances from unlabeled data directly. Liu et al. (2003) propose the biased SVM (Support Vector Machine) based on the observation that minimizing the error of misclassifying the unlabeled instances while constraining the accuracy of classifying the positive instances will lead to a good classifier (Liu et al., 2002). They set higher weight to the positive instances and lower weight to the unlabeled instances, which are assumed to be negative, to train the biased SVM (Liu et al., 2003). Elkan and Noto (2008) propose the Transforming Prediction Model (TPM) based on the random sampling assumption that the positive training data is randomly sampled from the positive population. A function is learned to rank the instances in the unlabeled data based on the probability that they are positive. Based on the "selected completed at random" assumption that the labeled positive instances are chosen completely randomly from the positive population, the authors found that a traditional supervised learner can be trained by treating the unlabeled data U as negative, then U is ranked by the learner. Elkan and Noto show that this method outperforms the biased SVM in terms of both effectiveness and efficiency significantly. They also show that it is effective to adopt this method to rank biomedical documents (Noto, Saier Jr, & Elkan, 2008).

2.4 Modeling Information Need

The quality of a query result is mainly determined by two factors: 1) the quality of user's queries that represent the information needs; and 2) the effectiveness of the ranking module of an IR system. Ranking models have been significantly improved over the years. However, there is still a gap between the user information need and the query that is often represented as text, which brings a big challenge to modern IR systems.

One approach for improving the query quality is using handcrafted controlled vocabularies for both indexing and composing queries (Fidel, 1991). However, the effectiveness of such methods is largely depending on the quality of the controlled vocabularies, which may need extensive human input and oversight for creation and maintenance.

Another popular method for improving query quality is relevance feedback (Agichtein, Brill, & Dumais, 2006; Rocchio, 1971; Salton & Buckley, 1997), which aims to improve the quality of the original query by adopting the information from the feedback documents to reduce the distance between the input text query and user's true information need in the vector space. For instance, the famous relevance feedback method, Rocchio algorithm (Rocchio, 1971), computes the query vector by taking into account both the documents in the positive set and the negative set using the following formula:

$$\vec{q} = \frac{1}{|P|}\sum_{i=1}^{|P|} X_i - \frac{1}{|N|}\sum_{j=1}^{|N|} X_j, \text{ where } X_i \in P \; and \; X_j \in N \, .$$

where P denotes the relevant document set, N denotes the irrelevant document set.

Relevance feedback has been shown effective in practice, but users are often reluctant to provide explicit feedback (Joachims, 2002). Pseudo feedback is an alternative method that assumes a certain number of top-ranked documents from the returned results of the original query to be relevant documents and uses them to update the original query

(Shen & Zhai, 2005; Tao & Zhai, 2004, 2006). However, some of the top ranked documents may be actually irrelevant in practice. In addition, the precision of the ranking system may be downgraded because of the polysemy effect: the added terms may have different meanings from user's intended meaning.

Query log based studies have been proposed to model the information needs associated with a query by analyzing the large sets of query log data. Wen, Nie, and Zhang (2001) compute query similarity using query distance and click through data. Fonseca, Golgher, de Moura, and Ziviani (2003) adopt association rule methods for mining query logs and query sessions to discover the correlation between queries. Zhao et al. (2006) use query session to compute query similarities by calculating their popularity over time. Different from these methods, which rely on large sets of log data, this research intends to predict query intents by utilizing the topic distributions from the query examples using topic modeling techniques.

A main difference between this research and relevance feedback is that IFME belongs to a different search paradigm from relevance feedback.

IFME is easier to use than relevance feedback in the scenario where users have difficulty to express their information needs using simple keywords. In relevance feedback, a user has to conduct keyword-based searches using manually constructed keywords to express an information need; in SBME, an information need, which is usually too complicated to be expressed in keywords, is expressed using multiple relevant documents.

Besides, the feedback documents in relevance feedback are different from the query examples in SBME. The system performance of a relevance feedback system is highly dependent on the quality of the original query that is constructed manually. The feedback documents are usually the top ranked documents in the returned search result list.

If the original query is poorly constructed, most of the top ranked documents may be irrelevant, thus the top ranked documents that are used as feedback documents may not represent user's true information needs.

If the original query is well constructed, and most of the top ranked documents are relevant, but they are highly correlated as all of the feedback documents contain the terms in the original query. However, in IFME, the query examples are not obtained by using an initial keyword-based query, but are provided by the user. Thus, it is possible that the query examples in IFME may contain only few common terms, either because different words of the same concept are used in the sample documents or because they represent different aspect of the user's information need. In this case, the previous methods in relevance feedback (e.g., the term selection method) may perform poorly because of the topic diversity issue.

The traditional relevance feedback system's feedback function can be integrated into the SBME system. For instance, the users can add more relevant documents into the query examples to run the SBME system iteratively until they are satisfied with the results.

2.5 Inductive and Transductive Inference

This research also follows the transductive learning paradigm. Transductive learning is an extension of standard supervised learning in the setting of semi-supervised learning. It adopts the test data (unlabeled data) to train a model, which is in turn applied on the test data.

Most previous studies in machine learning and text mining fields are based on inductive inference (Angluin & Smith, 1983), in which the learner tries to induce a model from the training data aiming to have a low error rate on the test data that is unseen when the model is built.

Different from inductive inference, transductive learning focuses on a given set of instances (i.e., a test set) and tries to classify them with least errors (Vapnik & Vapnik, 1998).

Let $D_{train}=\{(x_1,y_1),\ldots.(x_n,y_n)\}$ denote an i.i.d (Independent and Identically Distributed) sample of n training instances, an inductive learner L seeks a function $h:X(D_{train})\rightarrow Y$, where $X(D_{train})$ is the input space and Y is the output space, by minimizing the structure risk using a penalty function to control the bias/variance tradeoff. The function is used for making predictions on a test data D_{test} which is also an i.i.d sample from the population, from which the training data D_{train} is sampled. In inductive learning, the test dataset is different from the training dataset, which means the testing data set is not used to build the function or classifier. In transductive learning setting, the learner is built using both the training data D_{train} and the test set $D_{test} =\{(x_1,?,),\ldots.(x_m,?)\}$, where the question mark denotes that the class or label for the instance x_i is unknown in the training process, such that the erroneous predictions on the test instances are minimized.

Since both the training and test sets are available in the training process, the function can be learned to adapt to the test data, thus transductive inference has the potential to outperform inductive learning. In recent years, transducitve learning has attracted booming interests from researchers in machine learning and text mining fields (Duh & Kirchhoff, 2008; Joachims, 1999).

Transductive Support Vector Machines (TSVM) (Joachims, 1999) is one of the most widely used tranductive learning method. Different from Support Vector Machines (SVM) (Hearst, Dumais, Osman, Platt, & Scholkopf, 1998), which tries to induce a general decision boundary for a learning task, TSVM incorporates the test set into the training process and aims to minimize prediction errors to just those particular instances in the test set.

This study adopts the same idea of transductive inference that the test data is included in the training process, but negative training data is not needed. As TSVM must be trained using both positive and negative data, it is not applicable for SBME, where there is no negative training data. This study adopts PU learning in the transductive inference framework to identify the most effective PU learning algorithms and investigates the feasibility of ensemble learning to improve the effectiveness of SBME.

2.6 Topic Modeling

In machine learning and nature language processing, topic models (Blei, Ng, & Jordan, 2003; Griffiths & Steyvers, 2002; Hofmann, 2001) are statistical models that uncover the hidden abstract "topics" in document collections. The basic idea behind topic models is that documents are mixtures of topics, where a topic is a probability distribution over words. Using these models, new methods can be developed to organize, search, and browse the large size of document collections.

Applications of statistical topic models in text analysis have received much attention in recent years, especially in information retrieval and text mining fields (Blei & Lafferty, 2006; Griffiths & Steyvers, 2004; Steyvers, Smyth, Rosen-Zvi, & Griffiths, 2004; Wei & Croft, 2006). For instance, in the studies by Griffiths and Steyvers (2004) and Blei and Lafferty (2006), topic models are adopted to extract scientific research topics; in the study conducted by Wei and Croft (2006), an LDA-based document model is proposed for ad-hoc retrieval in the language modeling framework.

An early topic model is the probabilistic Latent Semantic Analysis (pLSA) proposed by Hofmann (2001). The assumption behind pLSA is that the interdependence between words in a document can be explained by the latent topics to which the document belongs. The word occurrences in a document are conditionally independent on an assigned topic.

The most popular topic model is the Latent Dirichlet Allocation (LDA) (Blei et al., 2003), which is extended from the pLSA. The basic assumption behind LDA is that documents are associated with latent topics, and the corpus is modeled as a Dirichlet distribution of the topics, where each topic is characterized by a distribution over words. Based on this assumption, each document is represented as a probability distribution over some topics, and each topic is represented as a probability distribution over a number of words.

Using LDA, the topic distributions of each document can be inferred from the data collection. The results are represented in two matrices:

1. Document-Topic matrix, denoted as $M=T_d \times T_p$, where T_d is the number of unique documents in the system, and T_p is the number of topics. M_{ij} is the probability that a

document T_{di} has been assigned to topic T_{pj}.

2. Word-Topic frequency matrix, denoted as $WT=W\times T_p$, where W is the number of unique words in the dataset, and T_p is the number of topics. WT_{ij} is the probability that the word W_i has been determined into the topic T_j by the LDA model.

The matrix M can be used to get the most likely topics to which the query examples belong. Then the set of words that are associated with these topics can be obtained. They will be selected for representing user's true information needs to overcome the topic diversity issue.

2.7 Summary

In this chapter, keyword-based search and the classic techniques for information retrieval is reviewed. Then previous work on SBME and related machine learning techniques adopted in this research are presented. Based on the literature review, the proposed IFME framework follows the PU learning based SBME approach in the transductive inference framework. More details are discussed in the subsequent chapters.

CHAPTER 3

TOPIC MODEL BASED QUERY INTENT PREDICTION FOR SEARCH BY MULTIPLE EXAMPLES

In this chapter, a topic model based query intent prediction framework is presented. Topic distributions from the query examples are generated to identify the most likely topics that can be used to represent user's true information need.

3.1 Overview

When a user's search intent is represented using multiple examples, which are most likely belonging to multiple topics, it is biased to use the traditional methods such as the centroid method to build a query vector simply using the centroid of the vectors of the query examples.

This study proposes a topic model based method to predict user's true information need from the query examples. Specifically, a topic model is adopted to conduct topic analysis on the query examples to obtain topic distributions, which are used to predict the most likely topics that the query examples may belong to. Then for each topic, the terms are ranked based on the probability that they are predicted as belonging to the topic. These terms for each topic along with the corresponding probability values are used to build a query vector to conduct standard keyword-based queries using a traditional keyword-based IR system. Several queries are generated in this manner when the query examples are predicted as belonging to multiple topics. The result fusion module is then used to rank all the documents that are returned from the keyword-based search system using the built queries. Figure 3.1 shows an overview of the proposed framework, which consists of four main components: topic modeling analysis, term selection, keyword-based information retrieval and results fusion.

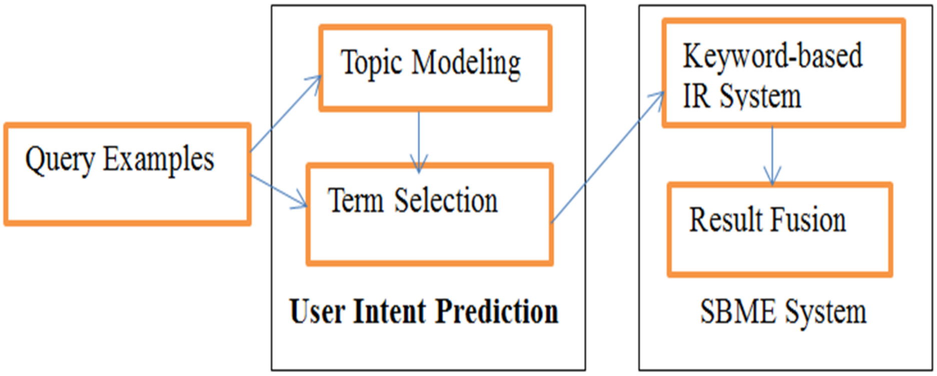


Figure 3.1 A framework for topic model based query intent prediction.

3.2 Topic Model Based Query Vector Construction

Based on the hypothesis that user's query examples belong to multiple topics, and the topics with higher probabilities are more likely to represent user's true information need, the most widely used topic model LDA is adopted to predict the topic distributions of the query examples. Using LDA, two probability distributions can be obtained: 1) p(t|d), the probability that document d is predicted as belonging to topic t; and 2) p(w|t), the probability of a term w under topic t. Table 3.1 shows an example of the four topic distributions for a set of query examples with 10 documents. It can be seen that only the 5th document is assigned to topic 2 (T_2) with probability 0.127, and none of other documents is assigned to this topic. This suggests that the terms of topic 2 may not represent the true information need. On the other hand, four documents are assigned to topic 1 (T_1) with probability higher than 0.5 (the scores in bold), and three documents are assigned to topic 4 (T_4) with a probability higher than 0.7 (the scores underlined). This suggests that it is more likely that the terms under topic T_1 and T_4 should be used to represent the information need expressed using the 10 query examples.

Table 3.1 An Example of Document Topic Distributions

Doc#	1	2	3	4	5	6	7	8	9	10
T_1	**0.79**	**0.981**	0.353	**0.515**	0.468	0	0	**0.566**	0	0.282
T_2	0	0	0	0	0.127	0	0	0	0	0
T_3	0.129	0.019	0.451	0.117	0.165	0.897	0	0.158	0	0
T_4	0.081	0	0.196	0.369	0.241	0.103	1	0.276	1	0.718

The following part presents the formalization of the idea of using topic modeling for information need prediction.

Let D={D_1,D_2,…D_n} denotes the set of query examples, let K denotes the number of topics to train an LDA model. The likelihood that the query document D belonging to topic T_i is calculated as follows:

$$L(D,T_i) = \sum_{j=1}^{j=n} p(T_i|D_j)$$

Given a topic T_i, the probability that a term w should be selected to represent the information need under this topic is denoted as $p(T_i|w)$, which is calculated using the following formula:

$$p(T_i|w) = \frac{p(w|T_i) * p(T_i)}{p(w)}$$

where p(w) denotes the global distribution of term w, and P(T_i) is calculated using the following formula:

$$p(T_i) = \frac{1}{n}\mathrm{L}(\mathrm{D}, \mathrm{T}_i)$$

Let S_i={w_{i1}, w_{i2},…., w_{im}} denote the terms that are associated with topic T_i, where w_{ij} denotes the j_{th} term under topic T_i. The corresponding query vector with respect to T_i can be represented as $\vec{S_\iota} =< p(T_i|w_{ij}), \dots p(T_i|w_{im}) >$, where $p(T_i|w_{ij})$ is the weight of term $w_{ij.}$

For each topic, a query vector is built using the topic distribution information generated from the LDA model. Then each vector is used to conduct a standard keyword-based search using a modern IR system. The next section presents how the documents in the system are ranked.

3.3 Result Fusion

For any given document d in the database, the similarity score between d and query vector under topic T_i (i.e., $\vec{S_\iota}$) can be computed using the cosine method:

$$sim(\vec{s_\iota}, \vec{d}) = \frac{\vec{s_\iota} \cdot \vec{d}}{\|\vec{s_\iota}\| \|\vec{d}\|} = \frac{\sum_{j=1}^{j=m} p(T_i|w_{ij}) * w_{dj}}{\sqrt{\sum_{j=1}^{j=m} p(T_i|w_{ij})^2 * \sum_{j=1}^{j=m} w_{dj}^{\;2}}}$$

where $\vec{d}$ represents the vector of document d, and w_{dj} denotes the weight of the feature w_{ij} in the vector $\vec{d}$. The ranking score of document d can be determined using the maximum value of the scores between d and the query vector of each topic (i.e., $\vec{s_\iota}, i\epsilon[1, k]$):

$$Rank(d) = \max\left(sim\left(\vec{s_i}, \vec{d}\right)\right), where\ i\epsilon[1, K] and\ i\epsilon N$$

3.4 Experiments and Results

This section reports the results of the experiments conducted on three benchmark datasets to evaluate the performance of the proposed method (called topic method) and the four baselines. The first baseline is the tf-idf based term selection method, which adopts the top terms that have the highest tf-idf scores to build the query vector. The second baseline is the centroid method that builds the query vector using the centroid of the query examples. The third baseline is the Skyline based method (denoted as Skyline), which adopts the Skyline operation to rank the documents in the database using the query examples. The fourth baseline is the Flexible Aggregate Similarity Search method (denoted as FANN), in which the similarity between the query examples and a document p in the database is an aggregation over the distances between the document p and a subset of documents in the query examples with a support value. Stanford core NLP[1] was adopted for stop words removal, stemming and lemmatization, and the IR system for this research was developed based on Lucene[2].

3.4.1 Data Collections

The experimental studies were conducted on three benchmark datasets.

The first dataset is the Reuters-21578 dataset, which is a commonly used benchmark news article collection in text classification. Each article has topic labels such

[1] http://nlp.stanford.edu/software/corenlp.shtml

[2] http://lucene.apach.org

as “acq”, “crude” and “money-fx”. There are 135 potential topic categories. Same as previous studies (Liu et al., 2003), only the documents in the most frequent 10 categories were used as the source of user’ query examples, while all other documents were used to form the unlabeled dataset.

The second dataset is the WebKB collection (Craven, McCallum, PiPasquo, Mitchell, & Freitag, 1998), which contains web pages gathered from university computer science departments. The pages are grouped into seven categories. This study used the four classes: course, faculty, project, and student, which contain most frequent instances. After removing those documents that are not in one of these categories, there are 4,168 instances left. The resulting vocabulary has 7,770 words.

The third dataset is the 20 Newsgroup (20NG) dataset (Joachims, 1996), which is a collection of 20,000 messages collected from twenty different Usenet groups. The dataset is classified by newsgroup names. After stop words removal and stopping, the resulting vocabulary has 70,216 words.

3.4.2 Evaluation Methods

Two standard IR evaluation measures, Mean Average Precision (MAP) (Manning et al., 2008) and p@k, were adopted for evaluation. A good ranking means all the relevant (positive) results are in the top ranked positions. In ranking the results of a query, MAP represents the mean of the average precision (AP) scores. Let L be a ranked list of documents, and R be the relevant documents being retrieved. The AP score is calculated using the following formula:

$$\mathrm{AP(L)} = \frac{1}{|R|}\sum_{k=1}^{n}\left(p(k) \times rel(k)\right)$$

where $|R|$ is the number of retrieved relevant documents; k is the rank in the sequence of the retrieved documents; n is the number of retrieved documents; p(k) is the precision at cut-off k in the list; rel(k) is an indicator function equaling 1 if the item at rank k is a relevant document, and zero otherwise The MAP score is the mean arithmetical value of the AP scores.

In IR, users are often interested in the precision of the top returned documents (Manning et al., 2008), which is denoted as p@k. Compared with MAP, p@k is a more user-oriented measure, as users hope to find relevant documents by only scanning the top few (e.g., 30) documents in the returned search results. This study chose k=10, 20, 30, respectively, for the system performance evaluation.

3.4.3 Experimental Design

To simulate user's search behavior in IFME, for each dataset, this study randomly generated query examples from each topic of the dataset. For each topic of interest, the documents that belonged to it were considered as positive examples to form the positive pool, and other documents in the same data collection formed the negative pool. For instance, 10 positive and 10 corresponding negative pools were obtained from the Reuters data collection. From a positive pool, a subset of |P| examples was randomly sampled to simulate user's query examples. The documents in the negative pool and the remaining documents in the positive pool formed the searching space.

For each rank, Average Precision, and p@k (k=10, 20, 30) values were calculated.

The MAP and average p@k scores for each run of the experiments were calculated using the mean of Average Precisions, and the mean of p@k for all the selected topics. For example, for the Reuters data collection, each experiment was conducted 10 runs, and the final MAP and average p@k scores were calculated by averaging over the MAP and average p@k scores from the 10 runs.

3.4.4 Experimental Results

In this section, the results of topic determination will be first presented. Then the results of performance comparison between the proposed method and the baselines will be described.

3.4.4.1 Topic Determination. To train an LDA model, the size of the latent topics (denoted as K) must be predefined. This study randomly sampled a subset of documents from each of the datasets to conduct experiments trying different numbers of K to see how the system performance changed when the topic number varied. The experimental results are shown in Figure 3.2, 3.3 and 3.4, where |P| denotes the size of the query examples. Observations include:

1) When |P| was small (e.g. |P|<=5), the increase of K led to the decrease of MAP,

2) When |P| increased from 2 to 10, the improvement of MAP was significant.

3) When |P| was larger than 10 (e.g. |P|=20), with the increase of K, the performance of the system first increased then decreased. In average, for the sampled Reuters dataset, WebKB dataset and 20NG dataset, the maximum system performance achieved when K=20, 10 and 30, respectively. Therefore, K was set as 20, 10 and 30, respectively for the Reuters, WebKB and 20NG dataset to conduct the following experiments.

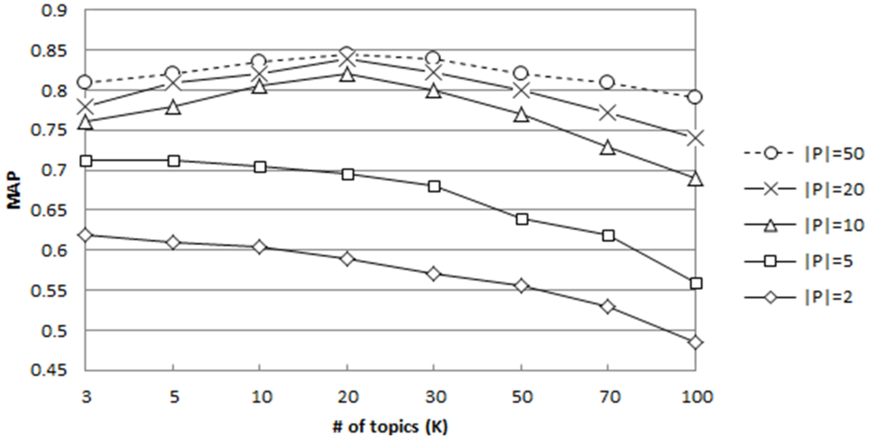


Figure 3.2 *M*AP on Reuters dataset under different numbers of topics.

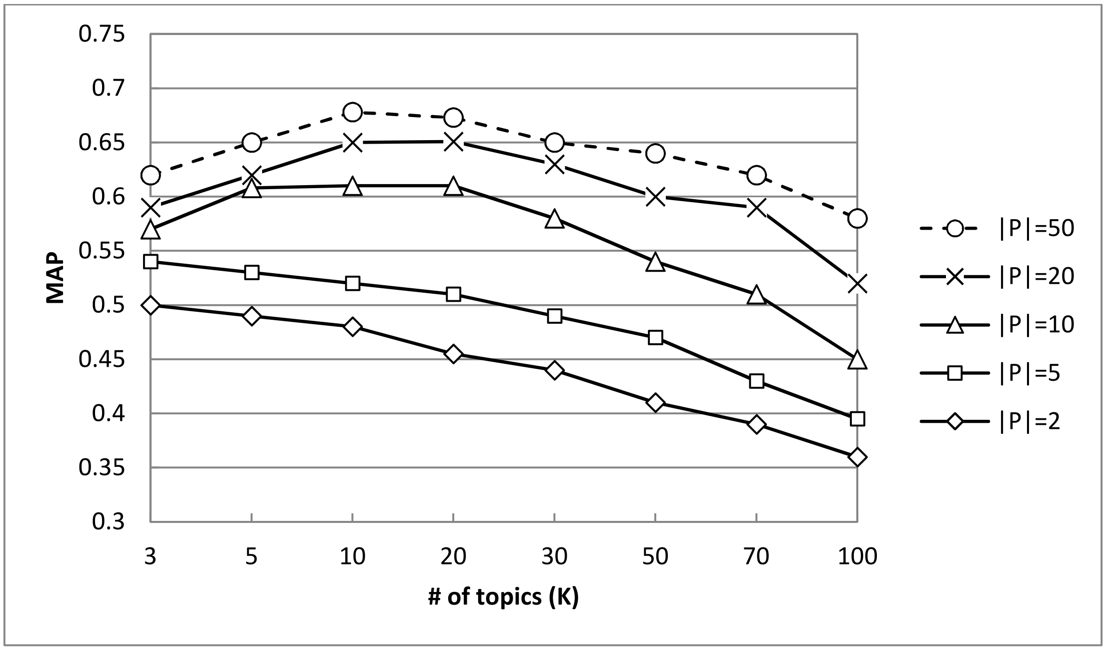


Figure 3.3 MAP on WebKB dataset under different numbers of topics.

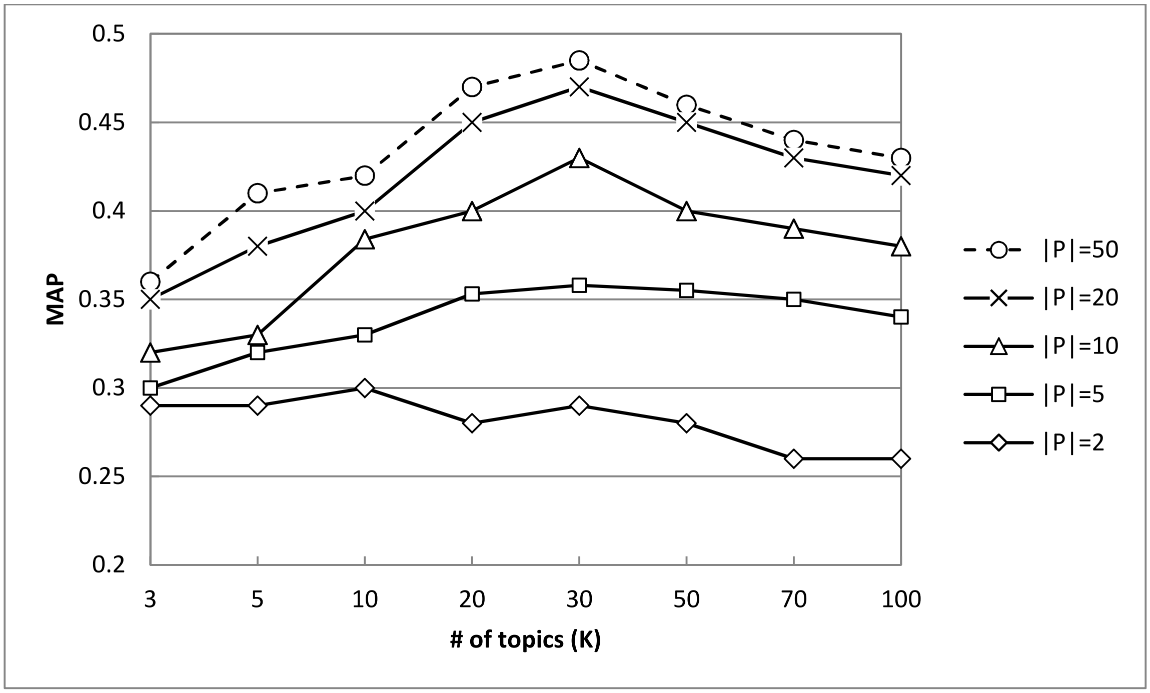


Figure 3.4 MAP on 20NG dataset under different numbers of topics.

3.4.4.2 Performance Comparison. The performance of the proposed method (denoted as Topic) was first compared with two baselines: tf-idf based method (denoted as Tfidf), and the centroid method (denoted as Centroid) in terms of MAP, and p@k (k=10, 20 and 30). The results are shown in Table 3.2, Table 3.3. and Table 3.4. The best results for each experiment are presented in bold.

It can be observed that: 1) when |P|<=20, when |P| increased, the performance of all the methods tended to increase; 2) when |P|>20, when |P| increased, the system performance tended to decrease. In all cases, the topic based method performed better than the two baselines significantly in terms of MAP.

For the experimental results on Reuters dataset, when |P| was small (e.g., |P|=2), the topic based method outperformed the baselines slightly in terms of p@10, but the performance improvement was significant in terms of MAP. However, when |P| was larger (e.g., |P|=10, |P|=20), the topic based method outperformed the baselines significantly in terms of both MAP and p@10. For instance, when |P|=20, the p@10 value for the topic based method was about 10% higher than the centroid method, and 6% higher than the Tfidf method. Similar observations can also be obtained from the experimental results on WebKB and 20NG dataset.

These observations from the study indicate that the proposed method can deal with the topic diversity issue by adopting the topic distribution information in the query examples.

Table 3.2 Experimental Results on Reuters Dataset, Where the Bold Numbers Indicate the Best Performance

|P|	Algorithm	MAP	p@10	p@20	p@30
2	Topic	**0.604**	**0.812**	**0.805**	**0.791**
	Centroid	0.551	0.788	0.759	0.745
	Tfidf	0.442	0.789	0.732	0.703
5	Topic	**0.702**	**0.925**	**0.913**	**0.893**
	Centroid	0.684	0.883	0.868	0.825
	Tfidf	0.452	0.857	0.783	0.742
10	Topic	**0.825**	**0.935**	**0.928**	**0.873**
	Centroid	0.773	0.887	0.894	0.826
	Tfidf	0.520	0.915	0.838	0.815
20	Topic	**0.842**	**0.943**	**0.925**	**0.914**
	Centroid	0.781	0.847	0.886	0.838
	Tfidf	0.56	0.886	0.856	0.845
30	Topic	**0.850**	**0.939**	**0.917**	**0.912**
	Centroid	0.792	0.853	0.872	0.844
	Tfidf	0.568	0.883	0.862	0.836
50	Topic	**0.837**	**0.875**	**0.87**	**0.892**
	Centroid	0.813	0.825	0.831	0.84
	Tfidf	0.573	0.903	0.865	0.821

Table 3.3 Experimental Results on WebKB Dataset, Where the Bold Numbers Indicate the Best Performance

|P|	Algorithms	MAP	p@10	p@20	p@30
2	Topic	**0.482**	**0.697**	**0.648**	**0.651**
	Centroid	0.457	0.652	0.613	0.621
	Tfidf	0.435	0.577	0.607	0.625
5	Topic	**0.522**	**0.697**	**0.688**	**0.672**
	Centroid	0.411	0.632	0.605	0.613
	Tfidf	0.423	0.615	0.622	0.604
10	Topic	**0.599**	**0.765**	**0.750**	**0.739**
	Centroid	0.555	0.713	0.707	0.686
	Tfidf	0.459	0.732	0.665	0.675
20	Topic	**0.655**	**0.808**	**0.793**	**0.836**
	Centroid	0.612	0.633	0.624	0.66
	Tfidf	0.509	0.73	0.71	0.68
30	Topic	**0.663**	**0.784**	**0.788**	**0.812**
	Centroid	0.604	0.657	0.672	0.691
	Tfidf	0.501	0.71	0.715	0.705
50	Topic	**0.679**	**0.755**	**0.776**	**0.770**
	Centroid	0.61	0.713	0.705	0.716
	Tfidf	0.516	0.74	0.727	0.713

As it is too inefficient for the Skyline based method (Skyline) and the Flexible Aggregate Similarity Search method (denoted as FANN) to rank all the documents in the database, only top 30 documents were identified and ranked. So p@k (k=10, 20 and 30) were used to compare the performance of the Topic and Centroid methods with Skyline and FANN methods. For the FANN method, a parameter selection study was conducted to identify the best ø. The results are shown in Figures 3.5, 3.6, and 3.7. It can be observed that the change of ø affects the system performance significantly. On average, when ø is around 0.2, FANN has optimal performance. It can be also observed that the optimal ø depends on |P|. For example, for Reuters dataset, when |P|=5, the system effectiveness is optimal when ø=0.6, but when |P|=10, the system effectiveness is optimal when ø=0.4. When |P|=30, when ø=0.2, 0.2, and 0.15 for Reuters, WebKB and 20NG dataset, respectively, the system effectiveness is optimal.

Table 3.4 Experimental Results on 20NG Dataset, Where the Bold Numbers Indicate the Best Performance

|P|	**Algorithms**	**MAP**	**p@10**	**p@20**	**p@30**
2	Topic	**0.291**	**0.798**	**0.693**	**0.67**
	Centroid	0.288	0.735	0.632	0.59
	Tfidf	0.264	0.711	0.615	0.562
5	Topic	**0.359**	**0.801**	**0.776**	**0.722**
	Centroid	0.314	0.741	0.713	0.682
	Tfidf	0.289	0.763	0.721	0.689
10	Topic	**0.431**	**0.834**	**0.738**	**0.731**
	Centroid	0.349	0.752	0.685	0.687
	Tfidf	0.298	0.769	0.692	0.693
20	Topic	**0.474**	**0.831**	**0.772**	**0.763**
	Centroid	0.412	0.740	0.688	0.665
	Tfidf	0.321	0.681	0.660	0.634
30	Topic	**0.496**	**0.849**	**0.761**	**0.715**
	Centroid	0.393	0.755	0.698	0.641
	Tfidf	0.312	0.706	0.644	0.623
50	Topic	**0.485**	**0.758**	**0.695**	**0.674**
	Centroid	0.342	0.685	0.643	0.611
	Tfidf	0.304	0.723	0.652	0.613

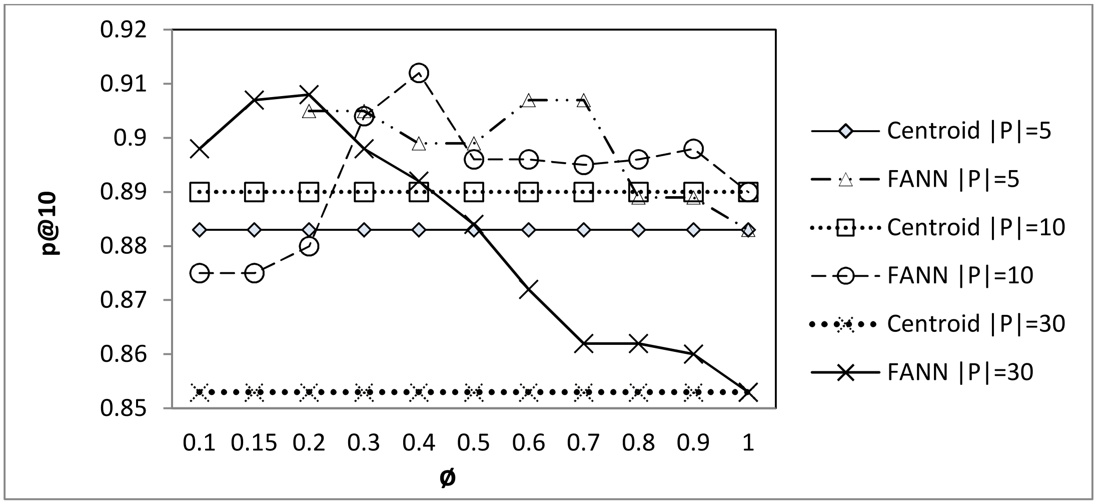


Figure 3.5 P@10 on Reuters dataset under different values of *ø*.

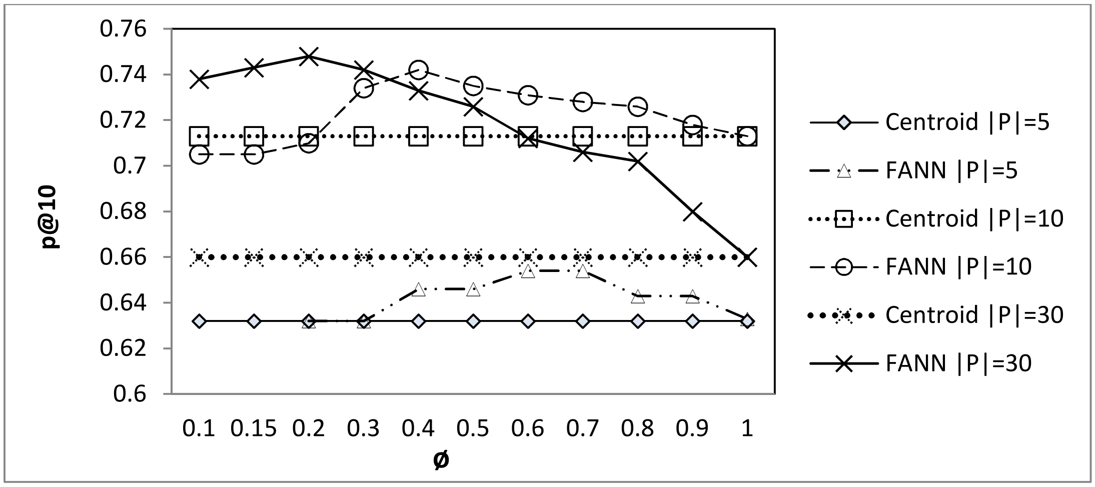


Figure 3.6 P@10 on WebKB dataset under different values of *ø*.

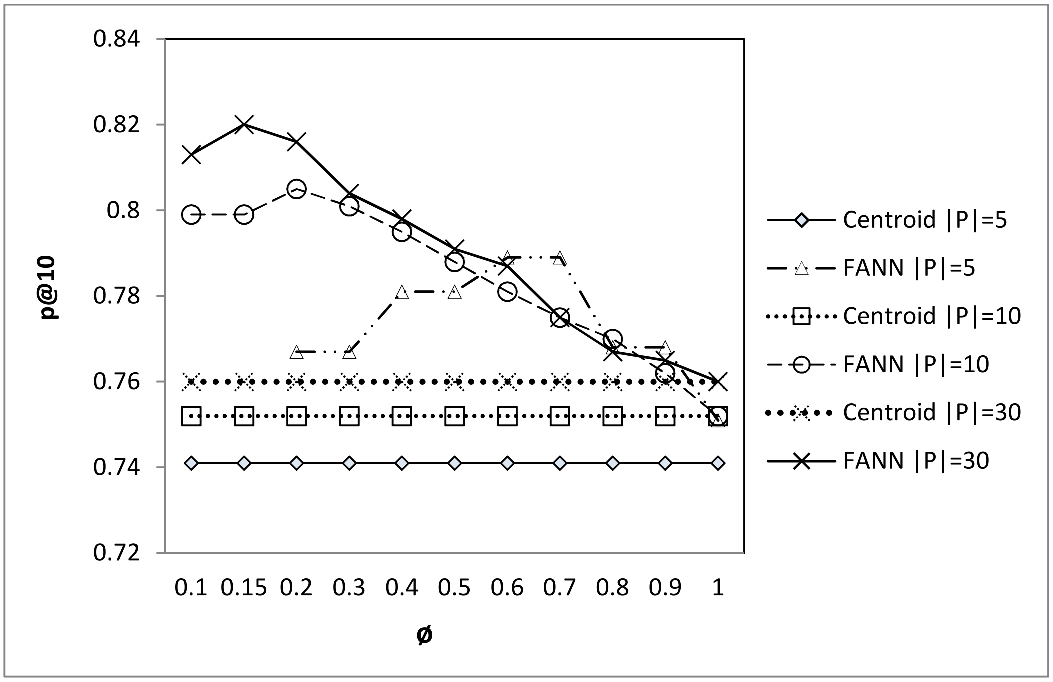


Figure 3.7 P@10 on 20NG dataset under different values of *ø*.

The performance comparison results are shown in Figures 3.8, 3.9 and 3.10 for Reuters, WebKB and 20NG dataset, respectively, where |P|=30, and for the FANN method, ø was set as 0.2, 0.2 and 0.15 for Reuters, WebKB and 20NG dataset, respectively. It can be observed that: The Skyline method has comparative performance with the Centroid based method. The FANN method outperforms the Centroid method in all the cases, which suggests that it is effective to select only a subset of the documents in the query examples to represent users' information needs. The topic based method outperforms all the baselines, which indicate the effectiveness of adopting topic analysis to solve the topic diversity issue.

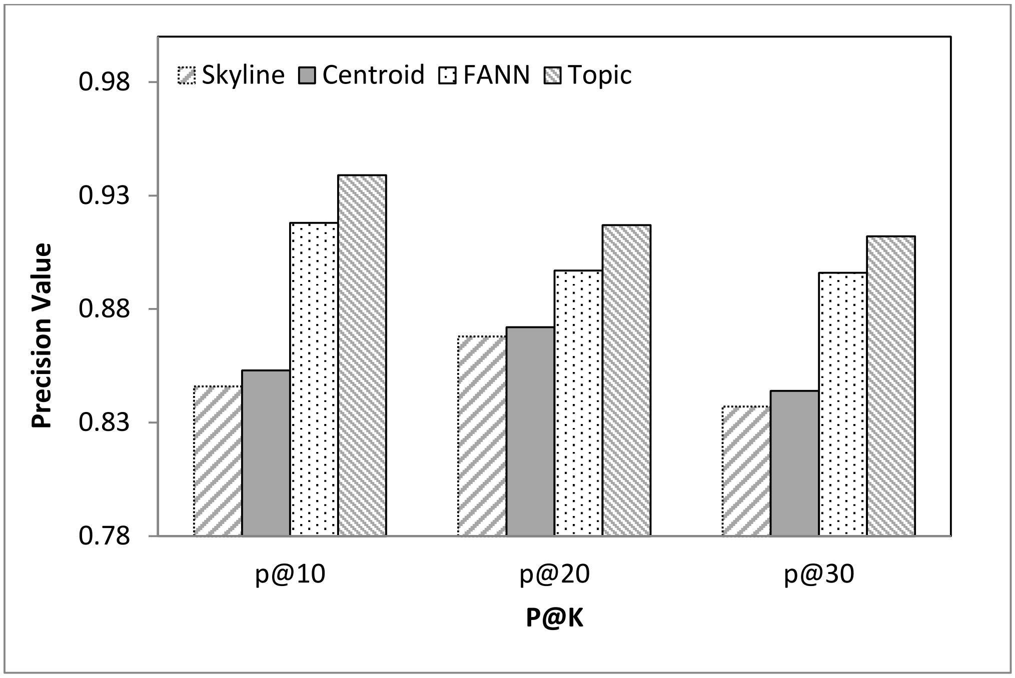


Figure 3.8 Performance comparison between the four methods on Reuters dataset.

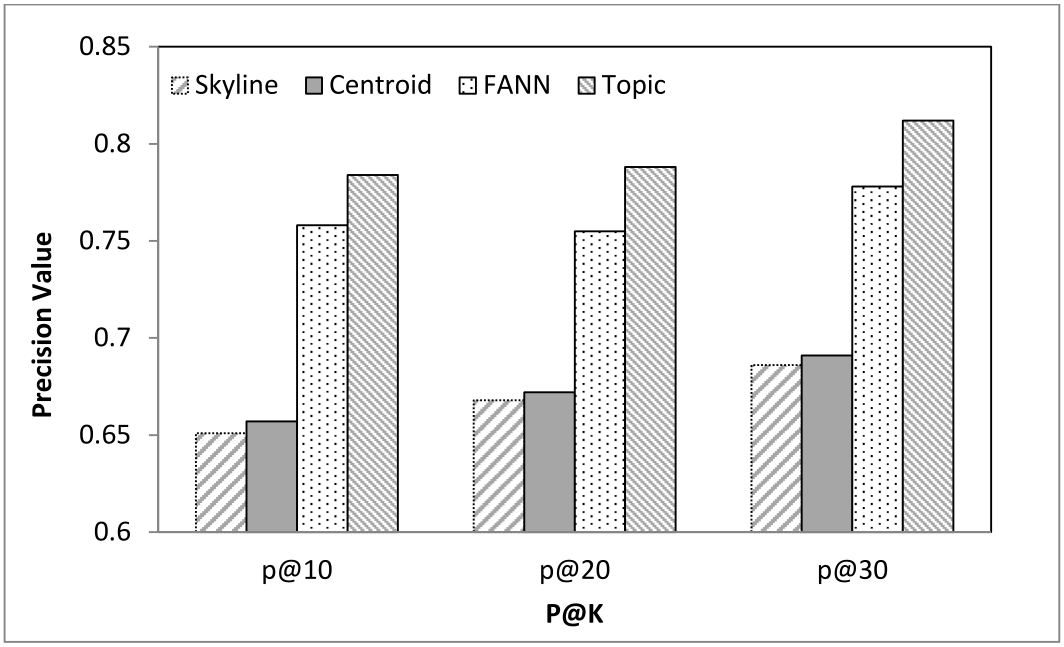


Figure 3.9 Performance comparison between the four methods on WebKB dataset.

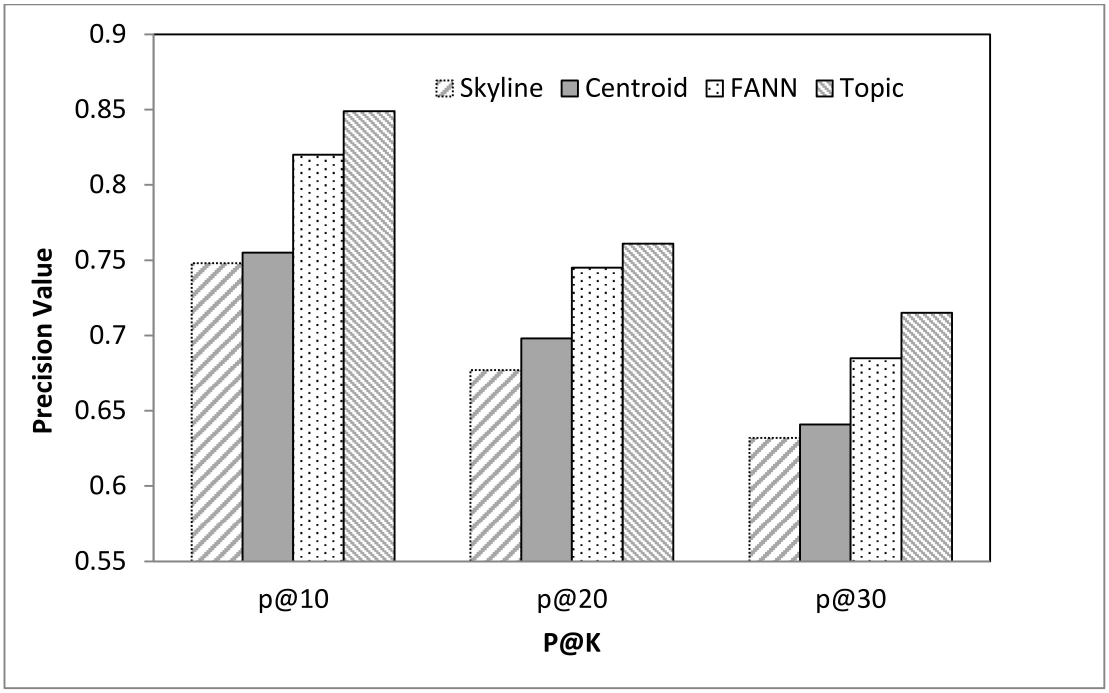


Figure 3.10 Performance comparison between the four methods on 20NG dataset.

3.5 Discussion

The success of the keyword-based search is highly dependent on the quality of the constructed keywords. However, it is often the case that it is difficult for users to select the appropriate keywords to represent an information need. SBME is a new search paradigm that allows users to express their information needs using a couple of relevant documents rather than a set of keywords. However, previous studies on SBME assume that query examples belong to one topic; in reality, they would most likely belong to multiple topics. The ignorance of the topic diversity issue of the query examples motivates this study to adopt topic analysis techniques to predict user's true information need from the query examples.

Based on the assumption that the most likely topics to which the query examples belonging are more likely representing user's true information need, topic modeling is adopted for information need prediction. The experimental results on three benchmark datasets show that this method works well.

To conduct topic model training and inference, a topic model should be trained using a large dataset that is selected based on the following two criteria: 1) it must be large enough to contain as many topics as possible; 2) it should be able to represent the documents in the collection of the search system. In this study, a subset of documents was randomly sampled from a dataset for topic model training. In practice, once the system is deployed into an online system, a topic model can be trained using a larger dataset, such as a subset of Wikipedia. Once the model is trained, it can be used by all the users of the system.

One concern of the proposed method is that the number of topic K should be predefined. A study was conducted to investigate how the system performance changed when K varied. For example, the experimental results on Reuters dataset indicate that: 1) the change of K indeed affects the ranking performance; for instance, the system performance downgrades when K is less than 20 or larger than 30, and 2) on average, setting K=20 on the sampled Reuters dataset leads to the highest MAP of the proposed method. In practice, a large dataset that is representative for the domain of the documents in the database can be sampled for topic model training.

The system performance also depends on the size of query examples |P|. For example, the experimental results on Reuters dataset show that: When |P| is smaller than 20, the system performance tended to increase with more query examples provided, but when |P| is larger than 20, the system performance tended to decrease. One possible reason is that when the size of query examples is too large, the query examples may belong to too many topics thus the system effectiveness starts to decrease. A potential solution for this problem can be applying cluster analysis on the positive examples and separate search results will be provided for each clustered group. However, in both cases, the experimental results show that the topic based method performs better than the baselines, when more relevant examples used as the query, in terms of MAP. This is because the more the query examples, the more likely that they are belonging to different topics. The experimental results from the study suggest that the proposed method can deal with the topic diversity problem by adopting the topic distribution information from the query examples learned from the LDA model.

3.6 Summary

This chapter presents a topic model based query intent prediction framework. The process for adopting topic model analysis techniques to solve the topic diversity issue of the query examples is presented. The results from the experiments conducted on three benchmark datasets indicate the effectiveness of the proposed method. A study also conducted to compare the FANN method and the centroid method, and the experimental results show the effectiveness of selecting a small subset of the query examples to represent users' information need.

CHAPTER 4

ADOPTING UNDER-SAMPLING TO SOLVE THE CLASS IMBALANCE PROBLEM IN THE PU LEARNING BASED IFME FRAMEWORK

Chapter 3 has described how to adopt topic analysis techniques to solve the topic diversity issue to better model user's true information needs. This chapter presents the proposed method for solving another research question: resolving the class imbalance problem in the PU learning based IFME framework. This study proposes a Transductive Positive Unlabeled Learning (TPULearning) based framework to solve the class imbalance problem using under- sampling method. Experiments have been conducted on three benchmark datasets, and the results show that the proposed method outperforms the baseline method significantly.

4.1 Overview

Most of the previous studies on SBME are based on PU learning. The idea is that considering a few relevant documents (provided by a user) as positive data and the documents in an online database as unlabeled data (called U), the documents in U are ranked by a PU learning algorithm. A major limitation of employing the PU learning based SBME to online search is that the system performance can downgrade dramatically because of the class imbalance problem: the size of a user's query examples is much smaller than the size of the unlabeled documents in the online database. However, the previous studies on SBME assume that the positive set and the unlabeled set are balanced, which is often not the case in practice. In online search, it is most likely that a user provides only a few relevant documents as a query. In this case, the training data is extremely imbalanced.

In this research, a method combines under-sampling and ensemble learning is proposed to solve the class imbalance problem in the PU learning based IFME framework.

The PU learning algorithm adopted in this study is the Transforming Prediction Model (TPM) (Elkan & Noto, 2008), which is the most effective PU learning algorithm that has been shown effective in identifying biomedical documents (Noto et al., 2008). Different from the research by Noto et al. (2008), the focus of this research is not to build a classifier for generalization. Instead, this study follows a transductive learning paradigm aiming at ranking the positive examples in the unlabeled data as high as possible to help users identify more documents similar to the provided examples. The method is denoted as Tranducitve Positive Unlabled Learning (TPULearning). It simply assumes the unlabeled data (called U) as negative to train a traditional classifier with the positive examples (called P). The classifier is in turn applied on the same unlabeled data to predict the probability that an instance in U is positive. Then the unlabeled data is ranked based on the predicted probability values.

4.2 The Transductive PU Learning Based Framework

If a user has a long-term information need, it should be relatively easy for him/her to select relevant documents from search results or provide relevant documents gathered from previous searches. Under that assumption, a framework is proposed to assist users to explore relevant articles from an online database. In this framework, users can express their information needs using a set of relevant documents. Then the documents in the database are ranked based on the relevant documents provided by the user.

The proposed framework is shown in Figure 4.1. The iterative document re-ranking using PU learning is the core of the framework.

Through the interface, a user can initialize the scope of the text collection to form the searching space. This is helpful for the users who are only interested in the articles containing some specific keywords, or published in some specific journals. By default, all of the documents in the system form the searching space.

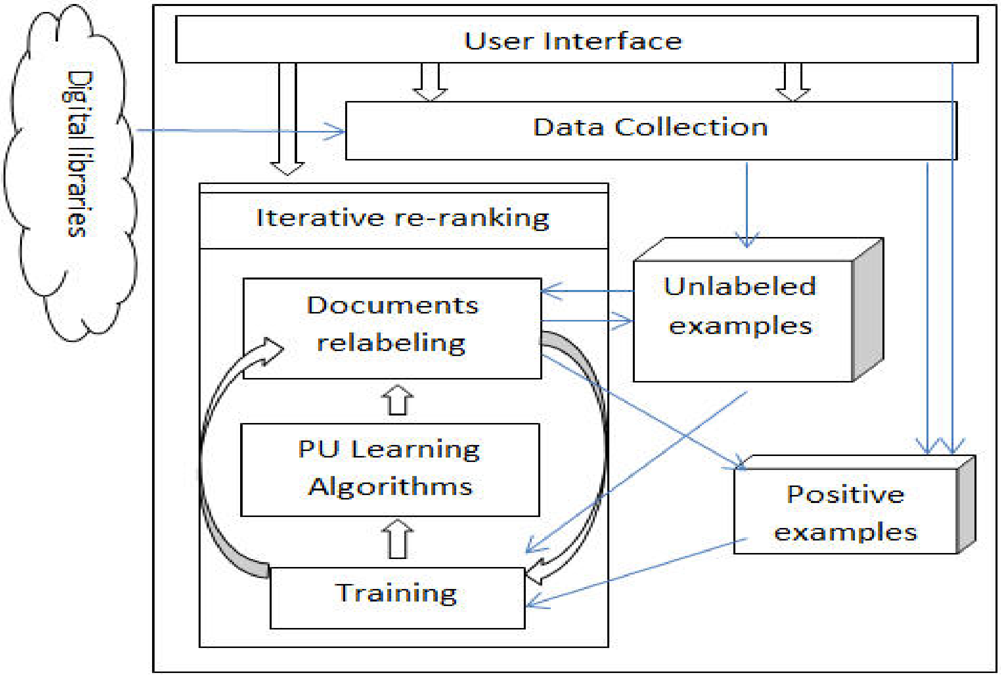


Figure 4.1 The TPULearning based framework.

After the initialization, the user can provide more relevant documents to express their information needs. These documents can be the documents in the system similar to the binder function in ACM digital library, or documents that are from other sources such as in the user's local computer. Such documents are called query documents. They form the positive data P and the rest of the documents in the searching space form the unlabeled data U. After the system conducts feature selection, all the documents are transferred into vectors. Then a standard PU learning model is trained from P and U. The model is then applied on U to rank the documents based on the probability that the documents in U are predicted as being positive. The ranked list is returned to the user for screening. After more positive examples are identified, a re-ranking procedure can be conducted iteratively. This study focuses on the first run of the document ranking process. The other iterative runs of the document ranking process are the same as the first run, except with more positive examples as inputs. As more positive documents are identified, it is expected that performance of the re-ranking will improve after each run.

4.3 Adopting Under-Sampling to Solve the Class Imbalance Problem

In traditional supervised learning, when a dataset is imbalanced, the performance of the learning model can decrease dramatically. In the PU learning based system, when |P| is much smaller than |U|, the dataset for learning becomes seriously imbalanced, which may degrade the performance of the classifier seriously. In an online search environment, it is often the case that the training data for the PU learning algorithm is imbalanced, so it is necessary to investigate how to overcome the class imbalance problem in the PU learning based system. An efficient strategy for dealing with class imbalance is under-sampling (Kotsiantis, Kanellopoulos, & Pintelas, 2006). A novel method is proposed that combines under-sampling with ensemble learning to solve the class imbalance problem for the PU learning based SBME system.

Specifically, M percent of the unlabeled data is sampled N times during the same procedure of the PU learning based ranking, and the final rank of an instance in U is the average rank of them in the N runs. In order to balance the class distribution, M should be approximately $(|P|/|U|)*100$. If $|U|$ is far larger than $|P|$, which is usually the case in online search, M will be very small. In this case, some instances may never be sampled. To overcome this problem, N should be large enough to ensure that the expected number of times an instance in U is sampled is not too small. The pseudo code for the proposed method is listed in algorithm 4.1.

Algorithm 4.1: Adopting Under-Sampling for Solving the Class Imbalance Problem

```
Input: positive set $P=\{X_1,X_2,\dots,X_{|P|}\}$, and
       unlabeled set $U=\{X_1,X_2,\dots X_{|U|}\}$.

For i=1 to N
1. Under-Sampling: randomly sample a subset $SU_i$ from U.
2. Feature selection on P and $SU_i$.
3. Learning a Classifier $C_i$ using P and $SU_i$;
4. Prediction: $R_i$=Rank ($SU_i$, $C_i$).
End for

For i=1 to U
$R_i$= Combine(Rank($X_i$), i=1 to N).
End for

Output: a rank list for U based on $R_i$
```

"Rank (SU_i, C_i)" uses the classifier C_i to predict the dataset SU_i, which will be ranked based on the probability that an instance is predicted as being positive. The basic idea of the proposed method is that by under-sampling a small set from U, the size of P and SU_i is more equal, thus the training set is more balanced. The final ranking for each of the instances in U is combined using all the information from each of the prediction from the classifier trained on P and SU_i. Here, the new rank for an instance in U is simply calculated using the weighted mean rank of X_i from each of the under-sampling stage. Let R_{ik} denote the rank of X_i in the k_{th} run of the under-sampling stage, and let P_{ik} denote the corresponding probability value of X_i being predicted as being positive. The final rank for X_i is calculated using the following formula:

$$R_i = \frac{\sum_{k=1}^{k=N} R_{ik} \times P_{ik} \times S_k}{\sum_{k=1}^{k=N} P_{ik} \times S_k}$$

where S_k=1, if X_i is selected in the k_{th} run of the under-sampling, otherwise S_k=0.

4.4 Experiments and Results

4.4.1 Data Collections

The experiments were also conducted using the three benchmark datasets presented in Chapter 3. In real word applications of information retrieval, the size of the unlabeled dataset is usually much larger than the size of the positive set that represents a user's information need. Therefore, unbalanced class distributions were randomly generated to simulate the real user search situations.

The first dataset is the Reuters dataset. Only the categories where the number of documents exceeded 250 were chosen to form the experimental datasets. There were six topics satisfying such a requirement; altogether there were 7,272 documents. For each of the six topics, the documents that belonged to it were considered as positive examples, and other documents formed the negative examples. Altogether, 6 positive, and 6 negative pools were generated, respectively, which were used to generate different datasets for the experiments. The description of the 6 subsets is in Table 4.1.

Table 4.1 Description of the Reuters Dataset

Topic	**acq**	**crude**	**Inter-est**	**Money-fx**	**earn**	**trade**
# of docs	2246	377	258	281	3762	348

The second dataset is the WebKB dataset, which contains web pages gathered from university computer science departments. The pages are grouped into seven categories. The top four classes: course, faculty, project, and student, which contain most frequent instances, were adopted in this study. The third dataset is the 20NG dataset, which is a collection of 20,000 messages collected from twenty different Usenet groups. The dataset is classified by newsgroup names.

4.4.2 Experimental Design

Each run of an experiment was conducted using the following procedure. The positive and negative example pools were formed using the method described in chapter 3. From the positive pool, a subset of |P| examples was randomly sampled to form the positive training data, and the same was performed on |PU| examples to form the unlabeled training data, with the constraint that there was no overlap between P and PU. Then a subset of |NU| examples was randomly sampled from the negative pool for the unlabeled training data. Then an SVM classifier was trained on the training data (P+U), and it was in turn applied on U to predict the probability that a document in it was positive. The documents in U were ranked based on the probability values, and an AP score was calculated for the rank. The final MAP score for each run of the experiment was calculated using the mean AP for all the selected topics. For each dataset, an experiment was conducted 10 runs, and the final MAP score was calculated by averaging over the MAP scores from the 10 runs.

4.4.3 Experimental Results

This section presents the results of the study for comparing the performance between the under-sampling based method (USTPULearning) and the original TPULearning method. In all experiments, the under-sampling method was carried out 100 times (i.e., N=100). In each run of the sampling, 20% of the documents in U were randomly sampled to form the training data.

P was changed with different combinations of PU and NU to conduct the study (Figure 4.2 to Figure 4.7). Observations include:

1) When more positive examples were provided, the effectiveness of the TPULearning method increased;

2) The under-sampling based method improved the effectiveness of TPULearning method significantly;

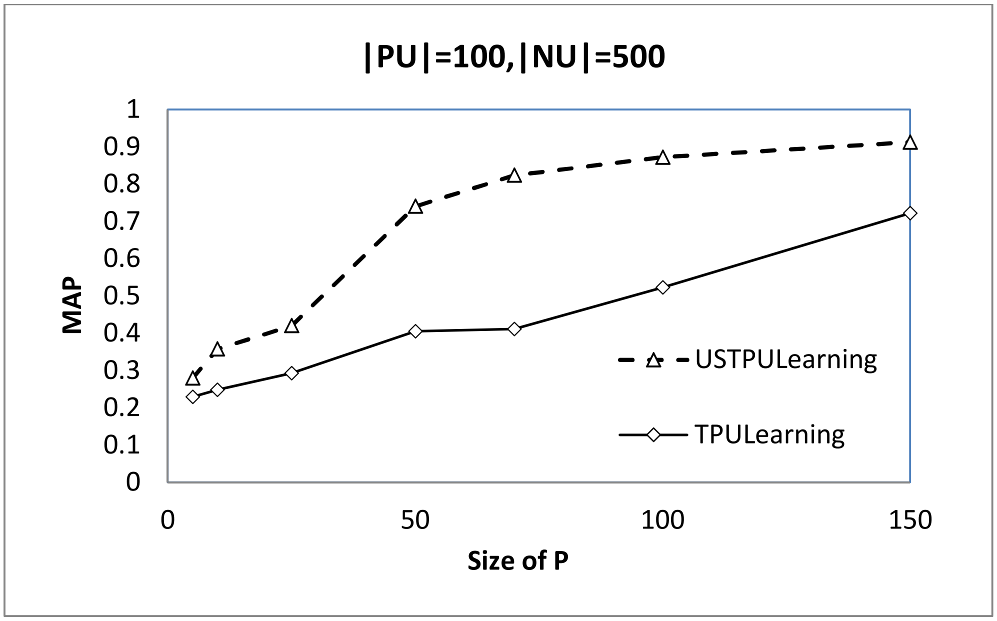


Figure 4.2 Performance comparison for the two methods under different numbers of |P|, when |PU|=100, |NU|=500 on Reuters dataset.

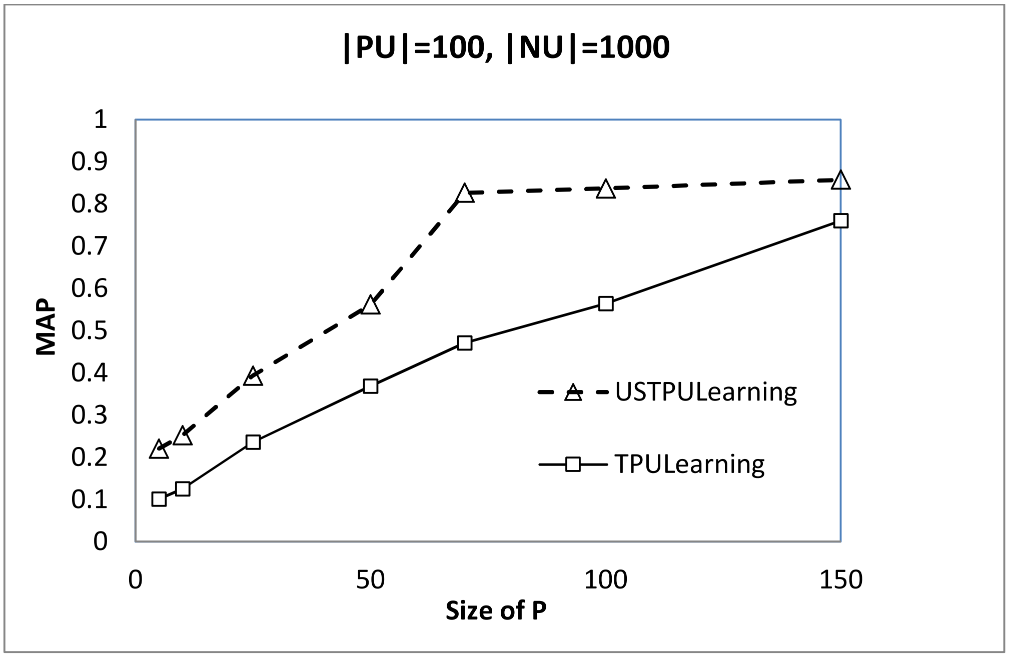


Figure 4.3 Performance comparison for the two methods under different numbers of |P|, when |PU|=100, |NU|=1000 on Reuters dataset.

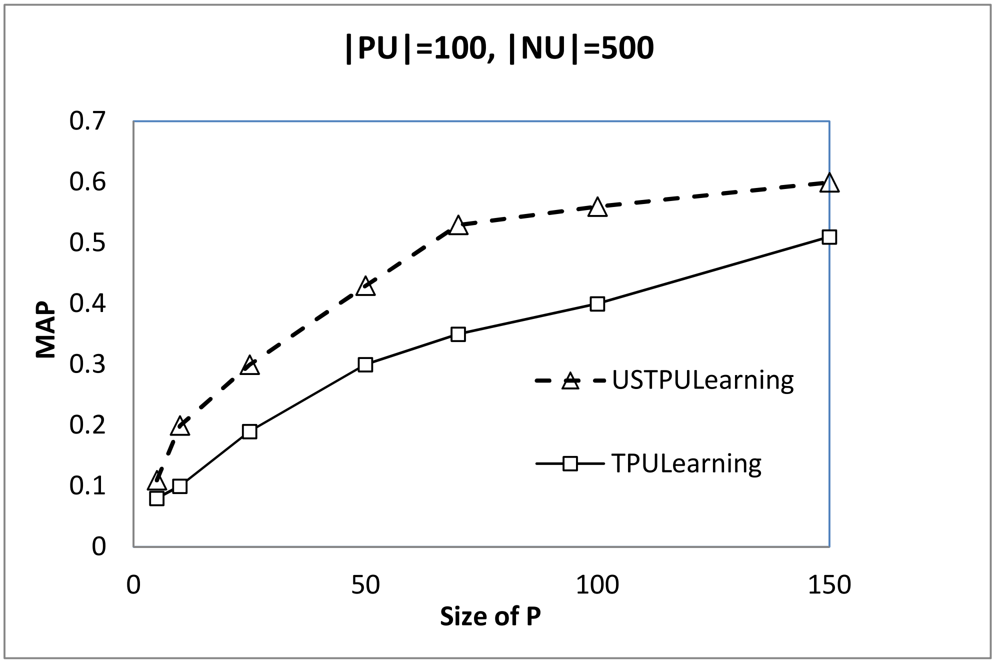


Figure 4.4 Performance comparison for the two methods under different numbers of |P|, when |PU|=100, |NU|=500 on WebKB dataset.

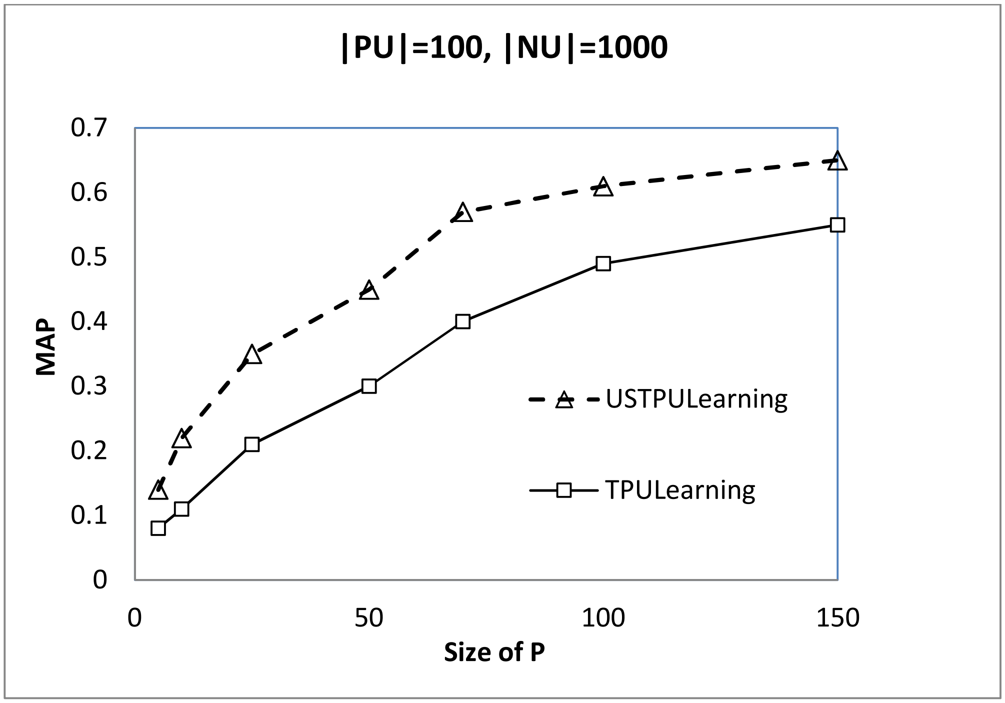


Figure 4.5 Performance comparison for the two methods under different numbers of |P|, when |PU|=100, |NU|=1000 on WebKB dataset.

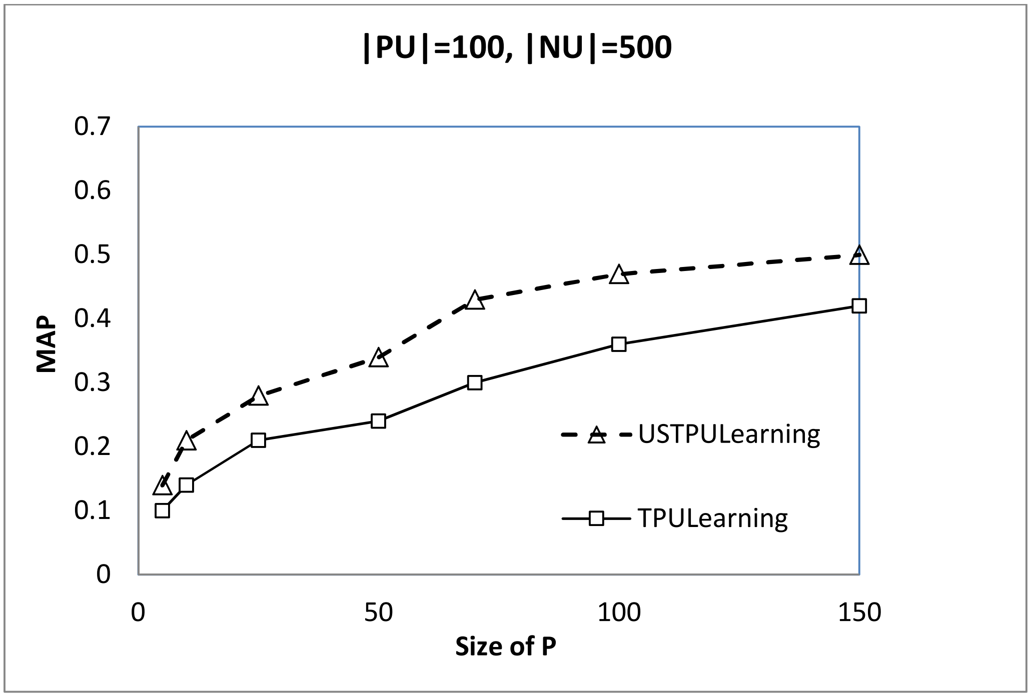


Figure 4.6 Performance comparison for the two methods under different numbers of |P|,

when |PU|=100, |NU|=500 on 20NG dataset.

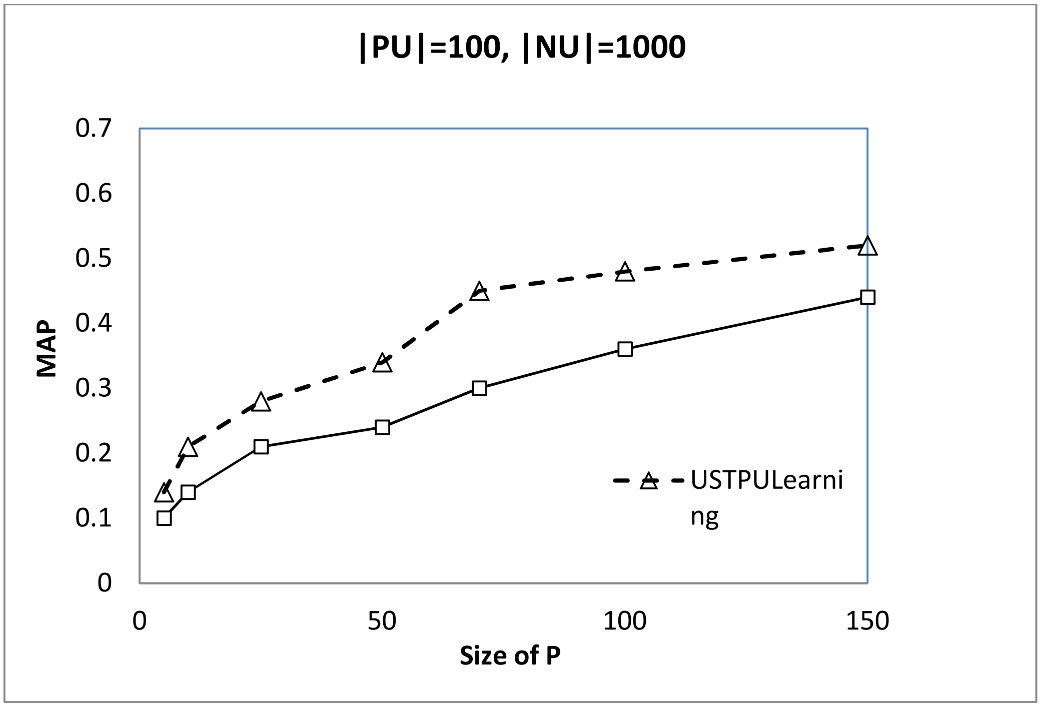


Figure 4.7 Performance comparison for the two methods under different numbers of |P|, when |PU|=100, |NU|=1000 on 20NG dataset.

3) Given the same size of P, with the increase of the proportion of PU in U, the performance of the two methods gradually decreased. The reason is that with more positive examples in the unlabeled data, which is considered as negative data for PU learning, the unlabeled data becomes much noisier thus the system performance tends to decrease.

4) When the size of P was large, say |P|>70, the performance of the under-sampling method appeared to become stable with the increase of |P|. However, the performance of the original method appeared to increase consistently with the increase of |P|.

4.5 Discussion

The experimental results from the study show that TPULearning is very promising, when |P| is large. This suggests that when a large set of positive examples are available, the TPULearning method can be used to identify more relevant documents. This is consistent with the findings from (Noto et al., 2008).

However, in an online search environment, it is often the case that |P| is usually very small. When |P| is small, the performance of the TPULearning method is poor. There are two possible reasons. First, when the number of features is large, the performance tends to decrease, as most features are from the unlabeled data. Second, when |P| is small, the dataset becomes seriously imbalanced. Under-sampling method is incorporated to reduce the imbalance by sampling a small set from U in each run, thus the performance of the method is improved. Experiments on the study show that the performance was improved by as much as 40% on Reuters dataset. For instance, the improvement was almost 40%, when |P| =70, |PU|=100 and |NU|=500 on Reuters dataset.

When |U| is large, which is often the case in an online search environment, N should be large enough to make sure that the expected number of times an instance in U is sampled is not too small. One concern is that, when N is large, it will bring efficiency problems. In the future, distributed computing methodologies can be adopted to perform the random sampling based learning in a parallel manner. For instance, a controller can be used to generate N training datasets, which will be assigned to other nodes of the clusters for training and prediction. The predicted results will be sent back to the controller such that the final ranking scores of the documents in the unlabeled data can be calculated and the unlabeled documents can be ranked accordingly.

4.6 Summary

In this chapter, a Transductive PU Learning based framework is proposed to help users conduct document retrieval using multiple examples to express their information needs. An under-sampling based algorithm is proposed to solve the class imbalance problem in the PU learning based framework. The experimental results suggest that the proposed approach can solve the class imbalance problem effectively.

CHAPTER 5

ADOPTING ENSEMBLE LEARNING TO IMPROVE THE EFFECTIVENESS OF THE PU LEARNING BASED IFME SYSTEM

5.1 Overview

In the previous studies on SBME, all the documents in an online database are considered as unlabeled data. This is inefficient as the size of the documents in a modern online database can be huge. This study proposes a PU learning based framework for SBME with a two-stage based approach: 1) potential relevant documents identification to form the searching space; and 2) adopting PU learning techniques to rank the documents in the searching space by treating the query examples as positive training set P and the potentially relevant documents as unlabeled training set U. The first step aims to reduce the size of the unlabeled set thus efficiency can be improved. The new searching space is a subset of the entire collection, which should contain as many relevant documents as possible. In the second step, a transductive learning approach is followed to train a PU learning (Liu, 2007) model from P and U. The model is in turn applied to rank the instances in U according to the likelihood that they are predicted belonging to the positive class.

There are many PU learning algorithms available. They have been shown effective in text classification in terms of F measure. However, it is still unknown which PU learning algorithm has the best performance for document ranking in terms of MAP or p@k in the SBME setting, where the size of P can be small (e.g., |P|=2). On the other hand, the classification algorithms that are shown effective in term of F measure, which is a popular evaluation measure in text classification, may perform poorly for document ranking in terms of Mean Average Precision (MAP), and precision at k (*p@k*), which are the standard methods for evaluating the performance of ranking systems.

This study aims to investigate the effectiveness of the state-of-the-art PU learning algorithms for SBME to bridge this gap by conducting extensive experiments to identify the most effective PU learning algorithms for document ranking. Specifically, using MAP and *p@k* (*k*=10, 20, 30) as the evaluation methods, experiments were conducted on three benchmark datasets to compare the performance of two state-of-the-art PU learning algorithms (RcSVM and TPM) and the Rocchio classifier (Rc) in the proposed framework, with the change of |P| to simulate user's online search activities. RcSVM and TPM are selected for comparison because they are efficient and have the state-of-the-art performance. Rc is chosen as it is a widely used text classification method that has decent performance. This research also studies the feasibility of adopting ensemble learning to improve the system effectiveness by taking advantage of different PU learning algorithms.

5.2 The Two-stage Based Approach

The proposed framework, which aims to assist users to rank documents using multiple relevant examples as queries, consists of two steps: 1) potential relevant documents identification; and 2) transductive PU learning based re-ranking.

5.2.1 Stage 1: Potential Relevant Document Identification

In this step, the query examples are used to identify potential relevant documents from the entire data collection. These documents form the new searching space for the PU learning based ranking in the second step. Most previous studies treat the entire corpus in the database as the searching space, but the huge volume of the documents in the data collection makes this method inefficient. This step is also helpful for users who are only interested in the articles containing some specific keywords, or being published in some specific journals. One main goal of this step is to include as many relevant documents in the searching space as possible. This means recall is more important in this step. So any technique that can improve recall can be adopted in this step. This study adopts the “Boolean OR query” method (Salton, Fox, & Wu, 1983) for this purpose.

Specifically, an IR system is developed based on Lucene[3]. After stop words removal and stemming, each document is transferred into a term list. Each single term is then used as an index term to build the inverted index file. Given a set of query examples P, a set of important terms is extracted to represent the documents in P. The potentially relevant documents from the whole data collection are retrieved by using the extracted terms to conduct a “Boolean OR search” to achieve a high recall. In this study, the query terms are extracted from P using the following formula:

$$Score(t_i, P) = \frac{\sum_{j=0}^{|P|} tfidf(t_i, p_j) \times L_j}{\sum_{j=0}^{j=|P|} L_j}$$

where |P| is the size of the query examples P, p_j is the j_{th} document in P, L_j is the length of p_j, $tfidf(T_i,p_j)$ is the *tf-idf* score for term t_i in p_j. The terms are ranked based on the scores, and a maximum of X terms are selected for retrieving potentially relevant documents from the whole data collection. In the Lucene based system, X is set as 1024, which is the maximum number allowed.

[3] http://lucene.apach.org

5.2.2 Stage 2: Transductive PU Learning Based Re-ranking

The output of the first step forms the search space. Then a PU learning model is trained by treating the query examples as a positive set, and the documents in the searching space as an unlabeled set. The model is in turn used to re-rank the documents in the searching space.

This study investigates three PU learning algorithms: Rc, RcSVM, and TPM. The idea of ensemble learning is adopted to combine Rc with RcSVM and TPM to examine whether there is any improvement for document ranking. The next section will describe the PU learning algorithms, and the proposed ensemble learning methods.

5.3 PU Learning Algorithms

Many PU learning algorithms are available, but it is still unknown which PU learning algorithms should be chosen for SBME. This section will first demonstrates why it is necessary to re-evaluate the PU learning algorithms in the document ranking setting. Then the state of art PU learning algorithms that could be adopted for building SBME systems will be described.

5.3.1 The Necessity to Re-evaluate the PU Learning Algorithms in the SBME Setting

The performance of adopting PU learning for ranking cannot be obtained based on the previous results in adopting PU learning for text classification, which are evaluated in terms of F measure. As a PU learning algorithm that is effective in terms of F measure may perform poorly when it is adopted for text ranking. For instance, suppose there is a set of test data that contains four positive (denoted as 1) and 6 negative (denoted as -1) instances, which is shown below:

Class **1 1 1 1** -1 -1 -1 -1 -1 -1

Suppose there are two learning algorithms that could be used to rank the data. Using Learner A (denoted as LA), the data is ranked as follows:

Rank	1	2	3	4	5	6	7	8	9	10
True class	1	1	-1	-1	-1	-1	-1	-1	1	1
Predicted class	**1**	**1**	-1	-1	-1	-1	-1	-1	-1	-1

Suppose a threshold is chosen such that the top two instances are predicted as positive, while the others are predicted as negative, which means all the negative instances are classified correctly, the F score for LA is 0.667.

Using Learner B (denoted as LB), the data is ranked as follows:

Rank	1	2	3	4	5	6	7	8	9	10
True class	1	1	-1	-1	1	1	-1	-1	-1	-1
Predicted class	**1**	**1**	**1**	**1**	-1	-1	-1	-1	-1	-1

If a threshold is chosen such that the top four instances are predicted as positive and the others are predicted as negative, the F score for LB is 0.5.

LA is superior to LB in terms of F measure. However, when using the standard IR evaluation measures such as Mean Average Precision (MAP), LB is much better than LA, as the MAP score for LA is 0.683, and the MAP score for LB is 0.817.

The above example shows that the evaluation results on adopting PU learning algorithms in text classification should not be used to predict how they will perform in text ranking. To identify the most effective PU learning algorithms in the SBME setting, it is necessary to re-evaluate the performance of PU learning algorithms in terms of standard IR evaluation measures such as MAP or p@k.

5.3.2 Identifying the Candidate PU learning Algorithms for Comparison

In the inductive learning setting, a model can be trained beforehand before it is applied to make predictions on the unseen data. However, in the transductive learning based framework, a model is trained for each query (i.e., the query examples) and the corresponding unlabeled data (i.e., the documents in the searching space), so it is crucial that the selected model is efficient. On the other hand, the model also should be simple enough and has few parameters to tune. Since a model with too many parameters must be tuned for different domains. The documents in an online database usually belong to multiple domains, so it is unlikely to choose the best parameters for a query that may belong to any domain.

This study investigates three kinds of PU learning algorithms that are efficient and have only few parameters to tune. The first one is the Rocchio classifier (Rc), which has been widely used in information filtering and text classification. The second one is the transforming prediction method, which is the state-of-the-art PU learning algorithm (Elkan & Noto, 2008). It has been shown effective for biological documents identifying without negative training data (Noto et al., 2008). The third one is Rocchio based Support Vector Machines (RcSVM) (Liu et al., 2003), which is also an efficient PU learning algorithm that has the state-of-the-art performance.

5.3.2.1 The Rocchio Classifier. Rocchio is a widely used method in relevance feedback and text classification (Joachims, 1996; Liu et al., 2003). It is based on the classic

Vector Space Model, where each document is represented as a K-dimensional vector where each dimension corresponds to a feature (i.e., a term). Let D denote the training dataset, and C_i denote the instances in class c_i. A Rocchio classifier for class c_i is built by constructing a prototype vector $\vec{\boldsymbol{c}}_{\boldsymbol{i}}$ using the following formula:

$$\vec{c_i} = \frac{\alpha}{|C_i|}\sum_{\vec{d}\epsilon C_i}\vec{d} - \frac{\beta}{|D - C_i|}\sum_{\vec{d}\epsilon D-C_i}\vec{d}$$

In text classification, each instance in the test collection is also represented as a vector $\vec{d}$. The similarity score between vector $\vec{d}$ and each of the prototype vector can be calculated using the cosine similarity measure. Then d is classified to the class whose prototype vector is more similar to the vector $\vec{d}$.

The pseudo code for adopting Rc in the proposed framework is shown in Algorithm 5.1. The difference between this algorithm and the traditional Rocchio classifier is that there is no negative training data; and instead, U is treated as negative set. This design is based on the transforming prediction theory (Elkan & Noto, 2008) which states that if the "selected completed at random" assumption is hold, a learner trained based on P and U which is treated as negative can be used instead of a learner trained based on P and N

Algorithm 5.1: Using Rocchio Classifier for SBME

```
Input: Query examples P, and potentially relevant documents U.
Assign each document in P the class label 1;
Assign each document in U the class label -1;
Building a Rocchio prototype vector $\vec{p}$ for class P using P and U;
for each document $u_i$ in U
    Calculating the similarity value $sim(\overrightarrow{u_i}, \vec{p})$ between $u_i$ and the prototype vector.
Ranking $u_i$ higher than $u_j$ if $sim(\overrightarrow{u_i}, \vec{p}) > sim(\overrightarrow{u_j}, \vec{p})$.
```

(true negative data), if the learner is used for ranking.

This algorithm can also be used for identifying some reliable negative data (denoted as RN), such that some other classifiers such as Support Vector Machines (SVM) can be trained based on P and RN. Actually this method named RcSVM is proposed by Liu et al. (2003). They show that RcSVM works effectively in text classification. However, this method has not been adopted for document ranking. It is still unknown how it performs in the SBME setting in terms of standard IR evaluation measures.

5.3.2.2 Transforming Prediction Model. The Transforming Prediction Model (TPM) (Elkan & Noto, 2008) is based on the random sampling assumption that the positive training data is randomly sampled from the positive population. Let x be an instance, and let s be a random variable such that s=1 denotes x is labeled, and s=0 denotes x is unlabeled. Let y=1 denote x is positive. In the PU learning scenario, only positive examples are labeled, so when s=1 (x is labeled), then y=1 (x is positive) is certain. However, when s=0 (x is unlabeled), y is unknown (x can be positive or negative).

To rank the instances in U based on the probability that they are positive, the goal is to learn a function such that $f(x) = p(y=1|x)$ as closely as possible. Based on the "selected completed at random" assumption that the labeled positive examples are chosen completely randomly from the positive population, $p(s=1|y=1, x)$ is a constant, which leads to the following lemma:

Lemma 1: Suppose $p(s=1|y=1, x)$ is a constant $c\epsilon(0,1]$, then $p(y=1|x) = p(s=1|x)/c$;

The proof of Lemma 1 is based on the assumption that $p(s=1|y=1, x) = p(s=1|y=1)$. Let $g(x) = p(s=1|x)$, based on lemma 1, it can be obtained that $f(x) = g(x)/c$. This means if the function f is only used to rank the instance x according to the probability that x is positive (y=1), then the function g can be used instead of f, since sorting the instances by $p(s=1|x)$ is the same as sorting them by $p(y=1|x)$. So a traditional supervised learner can be trained by treating the unlabeled data U as negative, then U is ranked by the learner.

Elkan and Noto (2008) show that this method outperforms the biased SVM (Liu et al., 2003) in terms of both effectiveness and efficiency, significantly. They also show that it's effective to adopt this method to rank biomedical documents. This study investigates this method in the proposed framework, and compares its performance with other PU learning algorithms.

5.3.2.3 Rocchio based Support Vector Machine. The Rocchio based Support Vector Machine (RcSVM) consists of two steps: 1) extracting some reliable negative instances from the unlabeled set, 2) then building SVM classifier iteratively.

The first step is accomplished by using the Rocchio classifier. Given a set of positive data P and unlabeled data U, a Rocchio classifier is built by treating P as positive training data and U as negative treating data. Then the classifier is used to classify U. Those instances in U that are classified as negative form the reliable negative set RN.

In the second step, P, RN and Q=U-RN are used to run SVM iteratively. After each run, it is expected that more possible negative data will be identified to add to RN. The algorithm for adopting RcSVM for SBME is shown in Algorithm 5.2.

It should be noted that even though SVM is built iteratively, it does not guarantee that Slast is the best one, as some positive instances in Q could be classified as negative, which may downgrade the performance of the last classifier. So θ is used to check whether

Slast has gone wrong. However, this method will not work if the first and the last classifiers are poor. And the success of the Slast is highly dependent on S1. If S1 is weak, then it is highly possible that Slast is also weak. The advantage of using Rc is that the identified negative instances is often very pure, so S1 is often quite strong. As it is highly possible that NQ contains some positive instances, the possibility that Slast may perform worse than S1 is high. In addition, the iteratively running of SVM classifier is time and resources consuming. For these reasons, S1 is used as the final classifier for SBME.

Algorithm 5.2: Using RcSVM for SBME

```
Input: Query examples P, and unlabeled set U
    RN= Reliable negative instances identified by Rc.
    Q=U-RN
    Assign each document in P the class label 1;
    Assign each document in RN the class label -1;
    i=1;
    while(true)
        Train a SVM classifier $S_i$ using P and RN;
        Use $S_i$ to classify Q;
        NQ={$x$| $x$ is the instance in Q that is classified as negative by $S_i$ };
        if NQ={} then break;
        else Q=Q-NQ;
        RN=RN∪NQ;
        i=i+1;
    Use $S_i$, denoted as $S_{last}$, to classify P;
    if $\theta$ percent of P are classified as negative then use $S_1$ as the final classifier;
    else use $S_{last}$ as the final classifier;
```

5.3.2.4 The Proposed Ensemble Learning Methods. In machine learning, ensemble methods combine multiple models to obtain better predictive performance than could be obtained from any of the constituent models (Rokach, 2010). Previous studies on ensemble learning show that the combination of weak classifiers can generate more effective classifiers. As discussed above, a good learning algorithm evaluated using F measure may perform poorly when it is used for ranking. Since there is no research on combining multiple PU learning algorithms for text ranking, this study proposes two ensemble learners by combining Rc with RcSVM and TPM, respectively, and adopt the new methods for SBME. This study investigates whether the combination of Rc with RcSVm and TPM can improve the performance of each of them when they are used in the proposed framework.

The algorithm of the combination of Rc and RcSVM (denoted as Rc-RcSVM) is shown in Algorithm 5.3. $R(Rc,u_i)$ and $R(S_1, u_i)$ denote the rank values of the instance u_i when it is ranked by the classifier Rc and S_1, respectively. To use $R(u_i)$ for ranking, for two instances u_i and u_j, if $R(u_i)< R(u_j)$, then u_i is ranked higher than $u_{j.}$

The algorithm for Rc-TPM is similar to Rc-RcSVM except that RcSVM is replaced with the TPM technique for ranking U.

Algorithm 5.3: Using Rc-RcSVM for SBME

```
Input: Query examples P and Unlabeled data U
    Train a Rc classifier using Algorithm 1;
    Use the Rc classifier to rank the instances in U;
    Use the Rc classifier to identify reliable negative instances from
    U;
    Train a RcSVM classifier using algorithm 2;
    Use $S_1$ to rank the unlabeled data U;
     for each instance $u_i$ in U
        $R(u_i)=R(Rc,u_i)+R(S_1, u_i)$
    Rank U based on $R(u_i)$;
```

5.4 Experiments and Results

This section reports the results of the experiments conducted on three benchmark datasets to evaluate the five learning algorithms (Rc, RcSVM, TPM, Rc-RcSVM, and Rc-TPM) in the proposed transductive PU learning based framework. The performance of TPU learning based method was compared with the baseline, where extracted terms from the query examples were used as queries for conducting a keyword search. Stanford core NLP[4] was used for stop words removal, stemming and lemmatization. Lib-SVM (Chang & Lin, 2011) was used to implement the RcSVM and TPM. As suggested by (Noto et al. (2008)), all SVMs were trained using a quadratic kernel and the default parameter setting.

5.4.1 Data Collections

The study was also conducted on the three benchmark datasets. The first dataset is the Reuters-21578 dataset, in which each article has topic labels such as “acq”, “crude” and “money-fx”. There were 135 potential topic categories, of which only the most frequent 10 were used as the source of user’s query examples, while all the documents formed the unlabeled dataset. After stop words removal and stemming, the resulting vocabulary set had 19,241 words.

The second dataset is the WebKB collection (Craven et al., 1998), which contains web pages gathered from university computer science departments. The pages are grouped into seven categories. This study used the four classes: course, faculty, project, and student, which contain most frequent instances. After removing those documents that are not in one of these categories, there are 4,168 instances left. The resulting vocabulary has

[4] Stanford University. Stanford Core NLP Software. http://nlp.stanford.edu/software/corenlp.shtml

7,770 words.

The third dataset is the 20 Newsgroup (20NG) dataset (Joachims, 1996), which is a collection of 20,000 messages collected from twenty different Usenet groups. The dataset is classified by newsgroup names. After stop words removal and stopping, the resulting vocabulary has 70,216 words.

5.4.2 Experimental Design

Each run of an experiment was conducted using the procedure described in chapter 3. For each topic of interest in a data collection, the documents that belonged to it were considered as positive examples to form the positive pool, and all other documents in the same data collection formed the negative pool. From a positive pool, a subset of |P| examples was randomly sampled to simulate user's query examples. In default, the documents in the negative pool and the remaining documents in the positive pool formed the searching space.

In this study, the searching space was reduced through conducting a keyword-based search from the default searching space using the terms extracted from the query examples as the query. The top N documents, which were potentially relevant, in the search result list formed the new searching space. N should be chosen for making a trade-off between the efficiency and effectiveness of the system. When N was too small, only a small fraction of the positive documents were included in the unlabeled data, when N was large, the efficiency of the system sacrifices.

A transductive learning model was trained using the query examples and the unlabeled data, which was ranked by the model. An AP, and *p@k* (*k*=10, 20, 30) values were calculated for the rank. The MAP and average *p@k* scores for each run of the experiments were calculated using the mean AP, and mean *p@k* for all the selected topics. For each data collection, 10 runs for an experiment were carried out, and the final MAP and average *p@k* scores were calculated by averaging over the MAP and average *p@k* scores from the 10 runs.

5.4.3 Experimental Results

5.4.3.1 Potential Relevant Documents Identification. The aim of this step is to retrieve a small set of documents from the data collection, which should contain as many relevant documents as possible. This means recall is crucial in this step. The extracted terms from the query examples are used to conduct a Boolean OR search to achieve a high recall. An IR system based on Lucene, which is a widely adopted open source IR toolkit, was developed. The average recall at k (*r@k*) values on the three datasets given in Tables 5.1, 5.2 and 5.3 show that the first step of the proposed method works well. A large proportion of the relevant documents in the data collection can be included in the unlabeled data, which is much smaller than the entire data collection. For instance, for Reuters dataset, when k=3000, the recall is around 0.8, even when only 2 documents are used as queries.

How many of the top ranked document should be selected to form the searching space? It is hoped that at a certain position N in the ranked list, the recall is no less than 80%. This means, at least 80% of the relevant documents in the entire data collection are included in the new searching space. As long as the efficiency of the system does not deteriorate too much, N could be large to have recall achieve more than 90%. In this study, N is set to 4000, 3000 and 8000 for the Reuters, WebKB and 20NG dataset, respectively to conduct the following experiments.

Table 5.1 Average r@k Scores with Different Sizes of P (|P|) on Reuters Dataset

	|P|	***r@ 1000***	***r@ 2000***	***r@ 2500***	***r@ 3000***	***r@ 3500***	***r@ 4000***
Reuters	2	0.51	0.68	0.74	0.80	0.84	0.87
	3	0.48	0.67	0.73	0.78	0.83	0.86
	5	0.47	0.66	0.73	0.78	0.83	0.86
	30	0.52	0.69	0.75	0.79	0.83	0.86
	50	0.52	0.69	0.74	0.79	0.83	0.87
	100	0.53	0.70	0.75	0.79	0.83	0.87

Table 5.2 Average r@k Scores with Different Sizes of P (|P|) on WebKB Dataset

	|P|	***r@ 1000***	***r@ 2000***	***r@ 2500***	***r@ 3000***	***r@ 3500***	***r@ 4000***
WebKB	2	0.48	0.68	0.76	0.86	0.92	0.98
	3	0.47	0.70	0.76	0.86	0.92	0.98
	5	0.49	0.72	0.77	0.87	0.93	0.98
	30	0.49	0.72	0.77	0.87	0.93	0.98
	50	0.59	0.73	0.77	0.87	0.93	0.98
	100	0.59	0.73	0.78	0.88	0.93	0.98

Table 5.3 Average r@k Scores with Different Sizes of P (|P|) on 20NG Dataset

	|P|	***r@ 3000***	***r@ 4000***	***r@ 5000***	***r@ 6000***	***r@ 7000***	***r@ 8000***
20NG	2	0.51	0.58	0.62	0.70	0.77	0.81
	3	0.56	0.62	0.68	0.73	0.77	0.82
	5	0.70	0.76	0.81	0.84	0.86	0.88
	30	0.75	0.81	0.85	0.88	0.90	0.92
	50	0.76	0.82	0.87	0.90	0.92	0.94
	100	0.77	0.83	0.87	0.91	0.92	0.94

5.4.3.2 Performance Comparison Between Different PU Learning Algorithms.

The performance of the five PU learning algorithms was compared in the proposed framework. The experimental results on Reuters, WebKB, 20NG dataset are shown in Tables 5.4, 5.5, and 5.6, respectively, where Rc denotes the Rocchio classifier, RcSVM denotes the Rocchio based SVM, TPM denotes the method based on transforming prediction, and Rc-RcSVM and Rc-TPM are the new methods proposed here. Rc-RcSVM denotes the combination of Rc with RcSVM. Rc-TPM denotes the combination of Rc with TPM. This section reports the results when |P|=2, 5, 10, 20, 30, and 50. Experiments using other values of |P| were also conducted, and the results were consistent.

The experimental results show that:

On average, when |P| increased from 2 to 20, the performances of all the algorithms tended to increase; when |P| increased from 20 to 50, the system's performance tended to be stable or to decrease slightly.

In all the cases, Rocchio classifier performed better than RcSVM and the TPM model significantly. This is interesting, as RcSVM and TPM are the state-of-the-art PU learning algorithms that perform better than Rc in text classification. However when they are used for documents ranking in the proposed framework, Rc is superior.

The combination of Rc with RcSVM and TPM can improve the performance of each of the algorithms. Since Rc performed better than RcSVM and TPM, t-test was conducted to examine whether the new methods outperformed Rc significantly. The results show that the ensemble learning based methods achieve much better performance than Rc. For instance, in Table 5.4 when |p|=50, the difference between Rc-TPM and Rc is around 13% in terms of *p@10*. Overall, Rc-TPM and Rc-RcSVM outperformed other methods significantly in terms of *p@30*. This means they performed well in retrieving top ranked documents.

TPM performed poorly when |P| is small, although it had the state-of-the-art performance in text classification when a large set of positive training instances were available. Rc-TPM had lower performance than Rc in terms of MAP, but it performed better than Rc in terms of *p@k*.

The experimental results from the study suggest that, when users express their information needs using multiple examples, it is not ideal to simply extract terms from the query examples to conduct a traditional keyword-based search. Using Rc or the ensemble learning based PU learning algorithm for text ranking, the proposed PU learning based system performs much better than the baselines even when users only use very few relevant documents as queries.

Table 5.4 Experimental Results on Reuters Dataset, Where the Bold Numbers Indicate the Best Performance, * and ** Denote the Proposed Method Outperforms Rc at 0.10 and 0.05 Significance Level, Respectively

|P|		MAP	p@10	p@20	p@30
2	Rc	0.58	0.788	0.769	0.765
	RcSVM	0.542	0.775	0.728	0.704
	TPM	0.292	0.633	0.56	0.527
	Rc-RcSVM	**0.606**	0.82*	**0.812****	0.799*
	Rc-TPM	0.448	**0.83****	0.795	**0.807****
	TermBaseline	0.442	0.789	0.732	0.703
5	Rc	0.712	0.923	0.903	0.868
	RcSVM	0.639	0.775	0.793	0.794
	TPM	0.323	0.663	0.614	0.589
	Rc-RcSVM	**0.724**	**0.94**	**0.915**	**0.9****
	Rc-TPM	0.537	0.91	0.875	0.897**
	TermBaseline	0.452	0.857	0.783	0.742
10	Rc	**0.812**	0.945	0.924	0.883
	RcSVM	0.721	0.865	0.848	0.842
	TPM	0.379	0.76	0.699	0.675
	Rc-RcSVM	0.806	**0.972***	**0.955***	**0.938****
	Rc-TPM	0.708	0.957*	0.946*	0.924**
	TermBaseline	0.524	0.935	0.869	0.822
20	Rc	**0.849**	0.95	0.936	0.911
	RcSVM	0.762	0.868	0.857	0.857
	TPM	0.441	0.835	0.775	0.732
	Rc-RcSVM	0.843	**0.972**	**0.953***	**0.939***
	Rc-TPM	0.683	0.92	0.945*	0.93*
	TermBaseline	0.56	0.927	0.876	0.845
30	Rc	**0.858**	0.93	0.928	0.91
	RcSVM	0.757	0.85	0.851	0.841
	TPM	0.459	0.783	0.739	0.734
	Rc-RcSVM	0.842	0.932	0.924	**0.935***
	Rc-TPM	0.709	**0.99****	**0.935**	0.923
	TermBaseline	0.57	0.89	0.854	0.834
50	Rc	0.836	0.865	0.86	0.88
	RcSVM	0.741	0.785	0.818	0.856
	TPM	0.454	0.773	0.719	0.748
	Rc-RcSVM	0.**838**	0.927**	0.912**	0.936**
	Rc-TPM	0.751	**0.996****	**0.97****	**0.97****
	TermBaseline	0.583	0.913	0.875	0.85

Table 5.5 Experimental Results on WebKB Dataset, Where the Bold Numbers Indicate the Best Performance, * and ** Denote the Proposed Method Outperforms Rc at 0.10 and 0.05 Significance Level, Respectively

|P|	Algorithms	MAP	p@10	p@20	p@30
2	Rc	0.468	**0.685**	0.63	0.638
	RcSVM	0.462	0.581	0.603	0.59
	TPM	0.315	0.525	0.5	0.487
	Rc-RcSVM	**0.485**	0.675	**0.644**	**0.642**
	Rc-TPM	0.381	0.619	0.591	0.602
	TermBaseline	0.442	0.586	0.617	0.631
5	Rc	0.501	0.69	0.678	0.652
	RcSVM	0.513	0.581	0.606	0.608
	TPM	0.339	0.525	0.534	0.513
	Rc-RcSVM	**0.539***	0.65	**0.691**	**0.681***
	Rc-TPM	0.434	**0.694**	0.662	0.656
	TermBaseline	0.453	0.625	0.633	0.635
10	Rc	0.595	0.753	0.737	0.726
	RcSVM	0.591	0.669	0.688	0.702
	TPM	0.364	0.638	0.584	0.602
	Rc-RcSVM	**0.647***	**0.813****	**0.82***	**0.816****
	Rc-TPM	0.49	0.744	0.753	0.746
	TermBaseline	0.489	0.74	0.715	0.705
20	Rc	0.623	0.625	0.685	0.7
	RcSVM	0.647	0.731	0.722	0.735
	TPM	0.392	0.669	0.65	0.629
	Rc-RcSVM	**0.68****	**0.831****	0.816**	**0.825****
	Rc-TPM	0.555	0.825**	**0.825****	0.81**
	TermBaseline	0.519	0.79	0.74	0.722
30	Rc	0.655	0.795	0.795	0.798
	RcSVM	0.675	0.712	0.691	0.8
	TPM	0.4	0.644	0.663	0.656
	Rc-RcSVM	**0.706****	0.862**	0.866**	0.85**
	Rc-TPM	0.585	**0.938****	**0.891****	**0.877****
	TermBaseline	0.539	0.82	0.772	0.75
50	Rc	0.64	0.735	0.75	0.757
	RcSVM	0.618	0.675	0.691	0.692
	TPM	0.423	0.838	0.75	0.713
	Rc-RcSVM	**0.672***	0.781**	0.8**	0.815**
	Rc-TPM	0.606	**0.869****	**0.878****	**0.875****
	TermBaseline	0.518	0.77	0.747	0.733

Table 5.6 Experimental Results on 20NG Dataset, Where the Bold Numbers Indicate the Best Performance, * and ** Denote the Proposed Method Outperforms Rc at 0.10 and 0.05 Significance Level, Respectively

|P|	Algorithms	MAP	p@10	p@20	p@30
2	Rc	0.283	0.744	0.651	0.632
	RcSVM	0.275	0.726	0.631	0.611
	TPM	0.168	0.546	0.438	0.397
	Rc-RcSVM	**0.295**	**0.765**	0.682	**0.673***
	Rc-TPM	0.291	0.727	**0.694***	0.657
	TermBaseline	0.261	0.711	0.628	0.612
5	Rc	0.336	0.754	0.725	0.701
	RcSVM	0.316	0.731	0.701	0.687
	TPM	0.197	0.728	0.616	0.601
	Rc-RcSVM	**0.381***	0.786	**0.764***	**0.747***
	Rc-TPM	0.372	**0.791***	0.717	0.706
	TermBaseline	0.329	0.726	0.715	0.694
10	Rc	0.361	0.773	0.742	0.724
	RcSVM	0.316	0.702	0.714	0.689
	TPM	0.214	0.641	0.612	0.633
	Rc-RcSVM	**0.383**	**0.812***	**0.799****	**0.782****
	Rc-TPM	0.356	0.785	0.754	0.736
	TermBaseline	0.302	0.725	0.712	0.677
20	Rc	**0.423**	0.776	0.754	0.734
	RcSVM	0.398	0.733	0.713	0.693
	TPM	0.265	0.632	0.624	0.582
	Rc-RcSVM	0.421	**0.793**	**0.799***	**0.775***
	Rc-TPM	0.417	0.752	0.776	0.724
	TermBaseline	0.314	0.761	0.733	0.681
30	Rc	0.415	0.732	0.679	0.654
	RcSVM	0.388	0.652	0.613	0.608
	TPM	0.298	0.644	0.587	0.573
	Rc-RcSVM	**0.446**	**0.762**	**0.721***	**0.713***
	Rc-TPM	0.432	0.754	0.711	0.689
	TermBaseline	0.387	0.622	0.601	0.621
50	Rc	0.364	0.713	0.659	0.637
	RcSVM	0.321	0.635	0.635	0.622
	TPM	0.287	0.615	0.621	0.583
	Rc-RcSVM	**0.371**	**0.764***	0.679	**0.684***
	Rc-TPM	0.359	0.751	**0.698***	0.671
	TermBaseline	0.298	0.612	0.553	0.542

5.5 Discussion

Although both RcSVM and TPM are the state-of-art PU learning methods in text classification, the experimental results show that they perform worse than Rc when they are used for document ranking. In the ensemble learning based approach designed in this study, Rc is combined with RcSVM and TPM, respectively. The experimental results show that the combination of a weak classifier with two state-of-the-art PU learning algorithms, respectively can achieve better ranking performance. For instance, Rc-RcSVM has a higher p@30 score than other methods when |P|=50 on all the three datasets. This suggests that the proposed ensemble learner could help users who would like to scan the top ranked documents. Since the proposed ensemble learner has high precision of the top ranked documents, it is expected that they can be adopted to improve the performance of a pseudo-relevance feedback system, which assumes the top ranked documents to be relevant.

5.6 Summary

In this chapter, a PU learning based SBME framework is proposed using a two stage approach to improve the system efficiency. Experiments were conducted to re-evaluate the effectiveness of the state-of-the art PU learning algorithms performed in SBME setting using three benchmark datasets. Ensemble learning based PU learning algorithms were proposed to improve the effectiveness of the SBME system.

CHAPTER 6

POSITIVE UNLABELED LEARNING TO DISCOVER RELEVANT DOCUMENTS USING TOPIC MODELS FOR FEATURE SELECTION

In traditional studies on SBME, documents are treated as vectors, of which the features are keywords in the collections. Such a term-vector based document representation brings high dimensionality problems when the collection is large. This research proposes a framework using PU learning for SBME using latent topics identified by a topic model for feature dimension reduction. Specifically, Latent Dirichlet Allocation (LDA) is adopted to reduce the feature dimension of document vectors to a lower dimension of topic vectors. Then the procedure of discovering relevant documents using a PU learning method is conducted in the topic space.

6.1 Overview

Most of the previous studies on SBME adopt the transductive Positive Unlabeled learning (PU learning) techniques by treating the query examples as positive training data P and the documents in the entire data collection as unlabeled data U. There are two stages involved in these methods: 1) document preprocessing, 2) using PU Learning algorithms to rank the unlabeled data. In the first stage, documents are usually transformed into term vectors after feature selection and feature weight determination. In the second stage, PU Learning algorithms are applied on the prepared data (i.e., term vectors) for learning classifiers and making prediction on the unlabeled data. Thus, the key for a given PU learning algorithm to achieve good performance is to select a set of good features and use appropriate feature weighting methods.

From a machine learning perspective, feature selection is one of the basic problems of documents representation (Yang & Pedersen, 1997), which aims to extract a small subset of features from the problem domain to retain the fundamental information of the documents while getting rid of the redundant, irrelevant or even ambiguous features. As the main goal of transductive learning is not about learning a model for generalization; instead, it is about learning a model for each dataset of interest. Feature selection is of vital importance in the SBME scenario using transductive PU learning. As a result, the learning and prediction process must be very efficient. So it is crucial to identify a small number of features to represent the documents, as a high number of features will bring the dimensionality problem.

In a comparative study of feature selection methods in statistical learning of text classification, Yang and Pedersen (1997) evaluate document frequency (DF), information gain (IG), mutual information (MI), $\chi 2$ (CHI) and term strength (TS); and find IG and CHI to be the most effective term-based feature selection methods. In this research, CHI is chosen as the baseline feature selection method because of its simplicity and effectiveness.

As the number of terms selected from CHI is still large, a topic model based method is proposed to reduce the dimension size further. As a new method to represent a document as a topic distribution, topic model (e.g., LDA) has received substantial attention from the machine learning and text mining community. This research explores the possibility of using topic model for feature selection as a means to achieve dimension reduction while maintaining comparable search effectiveness.

Latent Dirichlet Allocation (LDA) is one of the most popular topic models that allow documents to have a mixture of topics (Blei et al., 2003). It allows sets of documents to be explained by latent topics, which can explain why some terms which are related to a special topic are similar. Using LDA, the topic distribution of a document, i.e., the probability that the document belonging to each of the latent topics, can be obtained. Then a document can be represented as a topic vector by using each of the LDA discovered topics as a feature and the probability as the corresponding feature weight. The resulted topic vectors can be used as the input to a PU Learning system.

To accomplish this goal, this research proposes a framework of using PU learning for SBME using topic models to perform feature dimension reduction by transforming the document representation from a term vector into a topic vector. The purpose of this research is to explore whether the latent topics discovered by LDA are effective in calculating the similarity between two documents in a topic level, and whether such topic based similarity calculation can improve the performance of PU learning algorithms. Specifically, experiments haven been conducted on three benchmark datasets to compare the performance of the PU Learning based SBME system between two feature selection methods: 1) using LDA to represent documents as topic vectors, and 2) using CHI method for feature selection to represent documents as tf-idf based term vectors. The experimental results show that the topic model based method has comparable performance with the term based method in terms of effectiveness, but outperforms the term based method significantly in terms of efficiency.

6.2 The Proposed Method

6.2.1 Overview of the Proposed Framework

This section describes the proposed framework of applying PU learning for SBME using latent topics identified by a topic model. The framework, illustrated in Figure 6.1, consists of the following modules and steps.

1) Data collection that is used for training a topic model

2) Training module produces a topic model

3) User's query examples form the positive data

4) Other search results form the unlabeled data

5) Converting P and U into topic vectors

6) Running a PU Learning algorithm on the topic vectors of P and U

7) Ranking the instances in U using the PU learning algorithm.

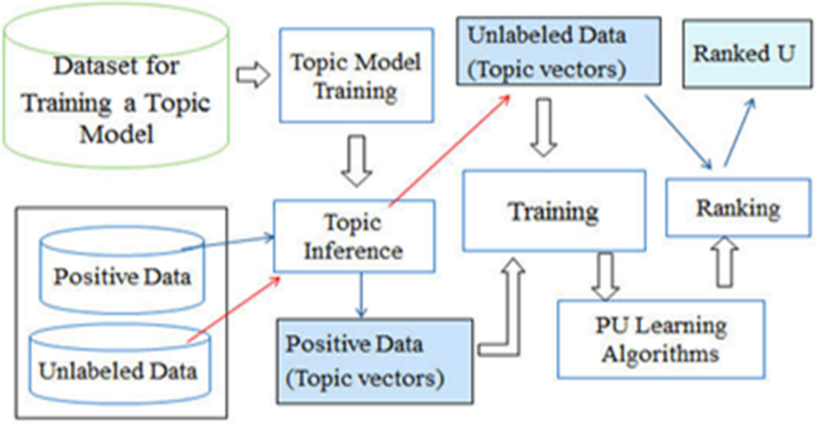


Figure 6.1 The Framework of PU Learning based SBME system using topic model for feature selection.

The proposed framework begins in the top left corner where a database stores the document collection. The training module produces a topic model to represent the document collection. The query examples or the positive documents from the user form the positive data P and the rest of the search results form the unlabeled data U. All the documents in P and U are transferred into topic vectors using the trained topic model. Then a standard PU learning algorithm is applied on the topic vectors such that the documents in U will be re-ranked.

Since the topic vectors are ultimately used for PU learning, the dataset for training a topic model should be carefully selected such that the topic inference module can make predictions that reflect the true nature of the documents in P and U. The criteria to select the topic model training data include: 1) it must be large enough to contain as many topics as possible; 2) it should be able to represent the documents in the collection of the search system. The selection of the training data is usually done by experts based on the collection in the system. For a search system that contains a large set of documents that are from different domains, the training data should be collected from as many domains as possible. Another strategy is to use a dataset that contains almost all of the domains. For example, for a normal search system, a subset of Wikipedia can be used for the topic training. If no such collection is available, an alternative is to use a sample of the documents in the retrieval system to train the topic model, which is adopted by this research.

In the module of topic model training, a pSLA or LDA model can be adopted. LDA is chosen in this research, as it has a more complete document generation assumption, and it has been shown as more effective than pSLA.

Once an LDA model is trained, it is used for topic inference for each document in the retrieval system. Actually, topic model training and inference is taking place offline. Therefore, it is unknown whether a document to be processed belongs to P or U. In Figure 6.1, P and U are used to illustrate how the proposed framework works.

Once a PU learning algorithm is selected, the process of training and ranking the unlabeled data is similar to the traditional methods, where the documents are represented as term vectors. In this research, Rocchio classifier is chosen as the PU learning algorithm because of its effectiveness.

6.2.2 Using CHI for Feature Selection

To distinguish the positive examples from the negative examples in the unlabeled set, it is important to use feature selection to identify features of negative examples and positive examples. Feature selection is a process that a subset of the terms in the training set is selected and used as features in text classification (Forman, 2003). Feature selection is based on such an algorithm that a utility measure for each of the terms to a class is computed and the K terms that have the highest values of the measure will be selected. Other terms that have lower values will not be used in the classification.

Yang et al. (1997) show that CHI is one of the most effective feature selection methods in text categorization. In this research, CHI (Liu, 2007) is selected as the baseline feature selection method for getting a set of features for the term based PU learning algorithm. As a popular utility measure for feature selection, CHI is applied to test the independence of two random variables in statistics. In feature selection, the two random variables represent the occurrence of the term and the occurrence of the class. The utility measure is calculated by using the following formula:

$$\mathrm{X}^2(t,c) = \frac{N \times (AD - CB)^2}{(A + C)(B + D)(A + B)(C + D)}$$

where A is the frequency of *t* and *c* co-occur, B is the frequency of *t* occurs without *c*, C is the frequency of *c* occurs without *t*, D is the frequency of neither *t* nor *c* occurs, and N is the total number of documents.

In this research, a set of experiments were conducted using the top K features identified by CHI. With different K being used, the performance of the methods will vary accordingly. When K is too small, the features may underrepresent the documents, thus the performance of the algorithms degrades. When K is too large, not only too much noisy information is included, but also consequently the high dimensionality of the document vectors. Both are detrimental to the performance. The goal is to select the best K for conducting the following experiments.

6.2.3 Using LDA for Feature Selection

Using statistical topic models to perform text analysis has received much attention in recent years, especially in information retrieval and text mining fields (Blei & Lafferty, 2006; Griffiths & Steyvers, 2004; Wei & Croft, 2006). For instance, Griffiths et al (2004) and Blei et al. (2006) adopt topic models to extract scientific research topics.

In this study, a topic model is adopted to extract topics from documents and convert each document into a topic vector. One advantage of using topic model for feature selection is that they reduce the dimensionality of feature space significantly. This is important as high dimensionality causes several problems for text mining algorithm

(Kriegel, Kröger, & Zimek, 2009). Latent Dirichlet Allocation (LDA) (Blei et al., 2003) is chosen as the specific topic model. The basic assumption behind LDA is that documents are associated with latent topics, and the corpus is modeled as a Dirichlet distribution of the topics, where each topic is characterized by a distribution over words. Based on this assumption, each document is represented as a probability distribution over some topics, and each topic is represented as a probability distribution over a number of words.

In the preprocessing stage, the LDA model can be used to get the topic distributions of each document in the data collection. The result can be represented in two matrices:

1. Document-Topic matrix, denoted as $M=T_d \times T_p$, where T_d is the number of unique documents in the system, and T_p is the number of topics. M_{ij} is the probability that a document T_{di} has been assigned to topic T_{pj}.

2. Word-Topic frequency matrix, denoted as $WT=W\times T_p$, where W is the number of unique words in the dataset, and T_p is the number of topics. WT_{ij} is the probability that the word W_i has been determined into the topic T_j by the LDA model.

The matrix M can be used to represent a document as a topic vector, where the topics are the attributes and the probability is the weight for the corresponding feature.

6.3 Experiments and Results

6.3.1 Data Collections

Similar to Chapter 3, the experiments were conducted on three benchmark datasets. The first dataset is a subset from the Reuters-21578 dataset, which is a collection of news stories. Each of the news stories is assigned a topic label. The category where the number of document exceeds 100 was selected to build the experimental datasets. There were 10 topics satisfying such a requirement.

The second dataset is the WebKB dataset, which contains web pages gathered from university computer science departments. The pages are grouped into seven categories. The top four classes: course, faculty, project, and student, which contain most frequent instances, were adopted in this study.

The third dataset is the 20NG dataset, which is a collection of 20,000 messages collected from twenty different Usenet groups. The dataset is classified by newsgroup names.

6.3.2 Experimental Design

For each topic in a data collection, the documents belonging to it were used to form the positive pool, the remaining documents formed the negative pool. |P| documents were randomly sampled from the positive pool as the query examples. Then the unlabeled dataset was constructed by randomly sampling |PU| positive examples and |NU| negative examples with the constraints that there is no overlap between PU and P.

A topic with a specific number of |P|, |PU| and |NU| (i.e. |P|=1, |PU|=60, |NU|=1000) forms a unit of the experiment, which results in an AP and a P@10 value. Each unit of the experiments was carried out 10 times, and the average AP and P@10 value was calculated for each topic for a specific number of |P|, |PU| and |NU|. The MAP and P@10 for a specific |P|, |PU| and |NU| were the mean AP and P@10 over the 10 topics.

To reflect the real situation of information retrieval, |PU| was kept much smaller than |NU|; and |P| changed from 1 to 30. In both of the experiments of feature selection using CHI and the topic determination for LDA based method, the unlabeled data was kept unchanged, and |P| changed from 1 to 30.

After the best number of features (K) and the number of topics (N) were identified, additional experiments were conducted to see whether the LDA based method outperforms CHI based method using the best K and N.

6.3.3 Experimental Results

6.3.3.1 Feature Selection. Feature selection experiments were first conducted to select a set of features using CHI for the term based Rocchio classifier. The unlabeled data was kept unchanged, while the size of the positive example changed from 1 to 30 to simulate the real situations of online search. The experimental results on Reuters, WebKB and 20NG dataset are shown in Figures 6.2, 6.3 and 6.4, respectively, where |P| denotes the size of positive examples.

Observations include:

1) when the number of features (denoted as K) increased, the performance of the ranking system tended to increase, but when K is larger than a certain value, the increase in K led to little increase in the MAP value. For example, there is little increase in the MAP value when K is larger than 4000, 3000 and 5000 for the Reuters, WebKB and 20NG dataset, respectively. This indicates that CHI method is useful in selecting a subset of features for the Rocchio classifier. Therefore, K was set to 4000, 3000, and 5000 for the Reuters, WebKB and 20NG dataset, respectively to conduct the following experiments.

2) when |P| increased from 1 to 10, the performance of the system increased significantly. Afterwards, the increase in |P| led to very small increase in the MAP value.

This is because with more documents in P, the topics in the query examples become more diversified thus more query examples provided less effect in improving the system performance.

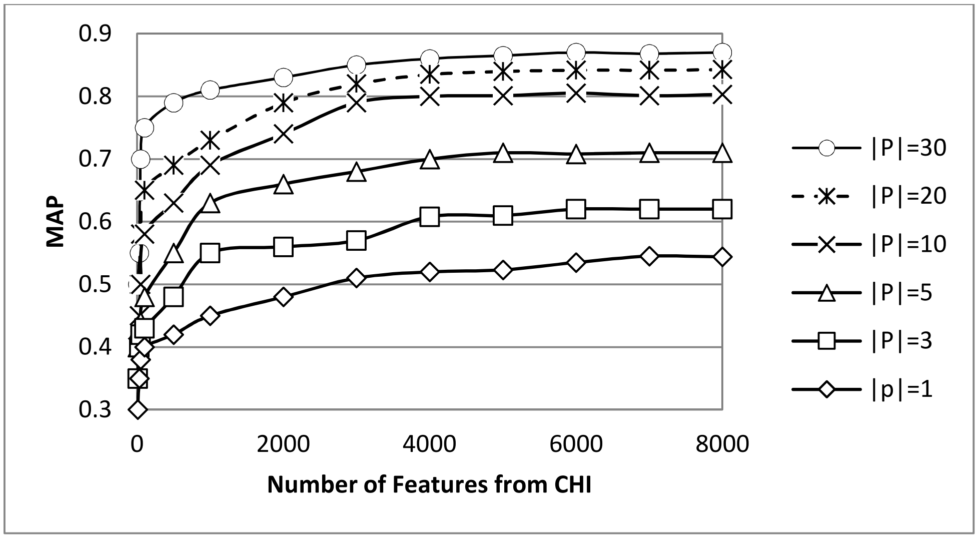


Figure 6.2 MAP under different numbers of features on Reuters dataset (|P| is the size of query examples).

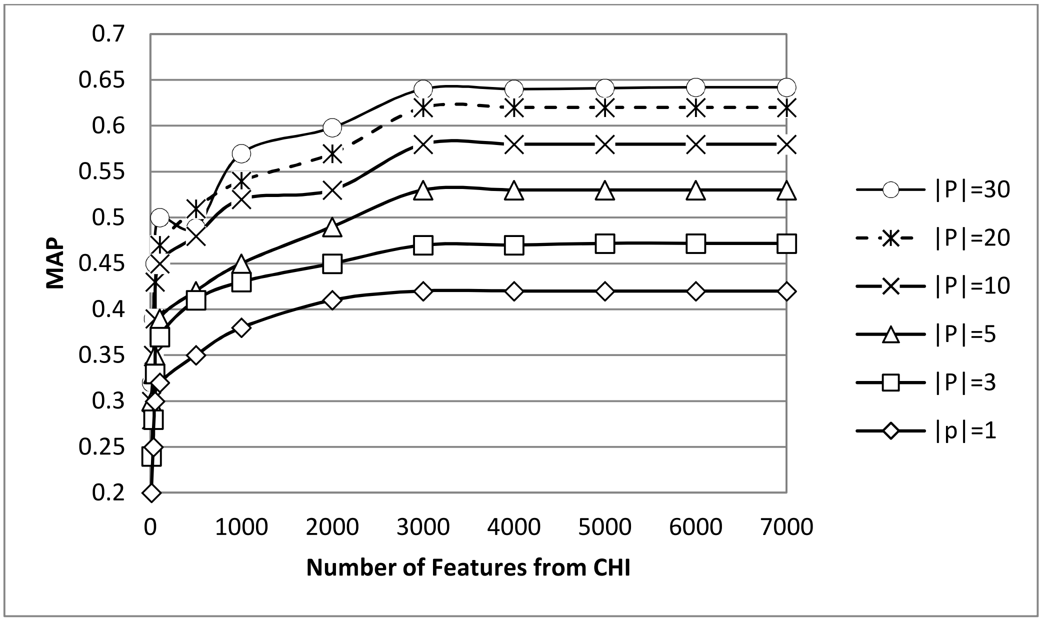


Figure 6.3 MAP under different numbers of features on WebKB dataset (|P| is the size of query examples).

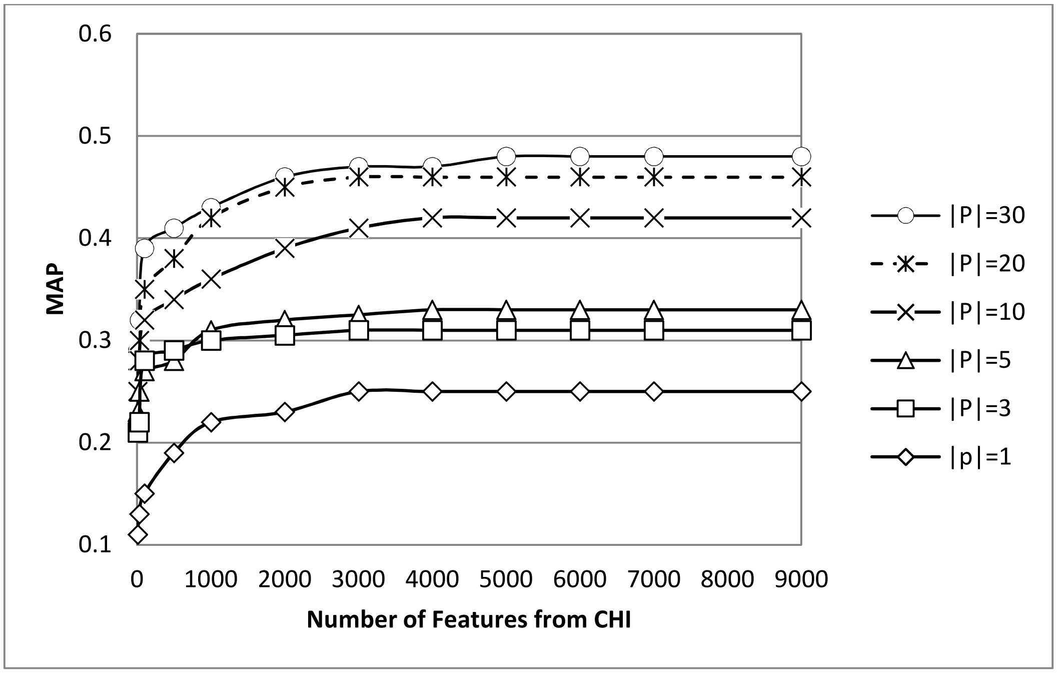


Figure 6.4 MAP under different numbers of features on 20NG dataset (|P| is the size of query examples).

6.3.3.2 Topic Determination. To train an LDA model, the size of the latent topics (denoted as N) must be predefined. Different numbers of topics were tried to see how the performance of the system changed when N varied. The experimental results on the three datasets are shown in Figures 6.5, 6.6 and 6.7, respectively, where |P| denotes the size of the positive examples. Observations include:

1) When |P|=1, the system performance first increased then decreased. For instance, when |P|=1, the system performance achieved maximum when N=50 on Reuters dataset.

2) When |P| increased from 1 to 10, the increase of MAP was significant. This observation was consistent with that in the last section.

3) When |P| was larger than 3, and when N increased, the performance of the system first increased then decreased. On average, when N=50, 30, and 70, the performance was

the best for Reuters, WebKB and 20NG dataset, respectively. Based on such an observation, N was set to 50, 30 and 70, respectively for the three datasets to conduct the following experiments.

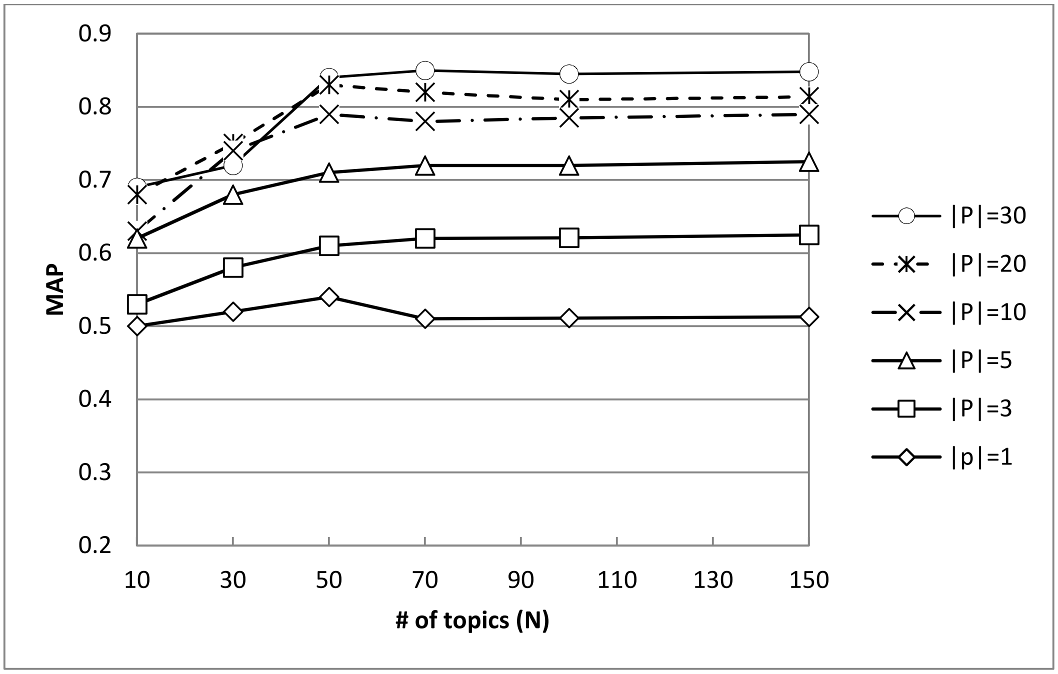


Figure 6.5 MAP under different numbers of topics on Reuters dataset.

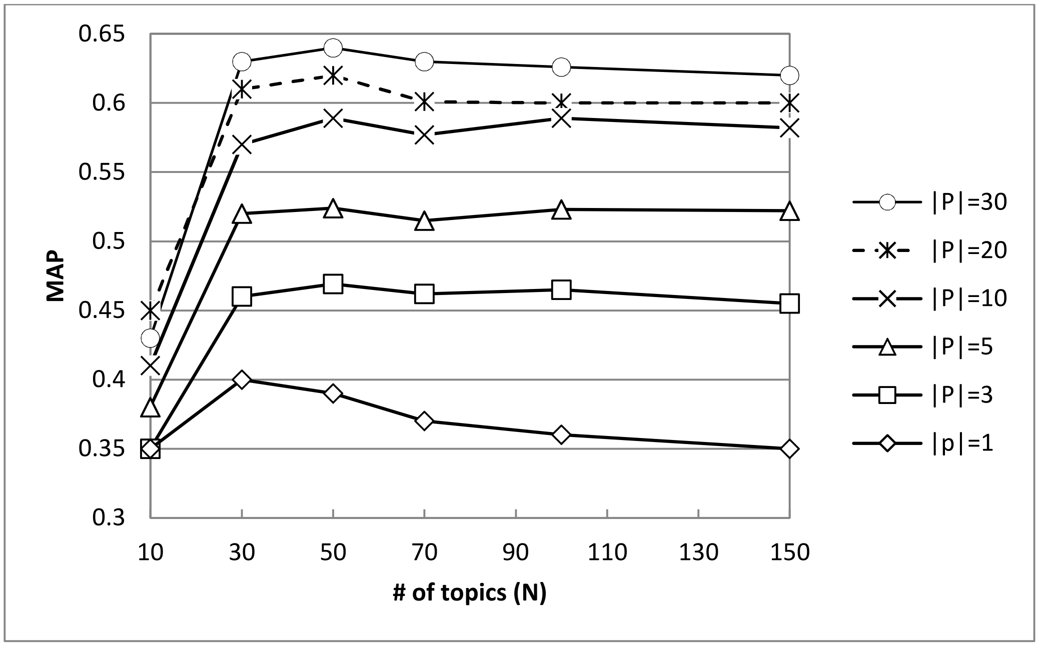


Figure 6.6 MAP under different numbers of topics on WebKB dataset

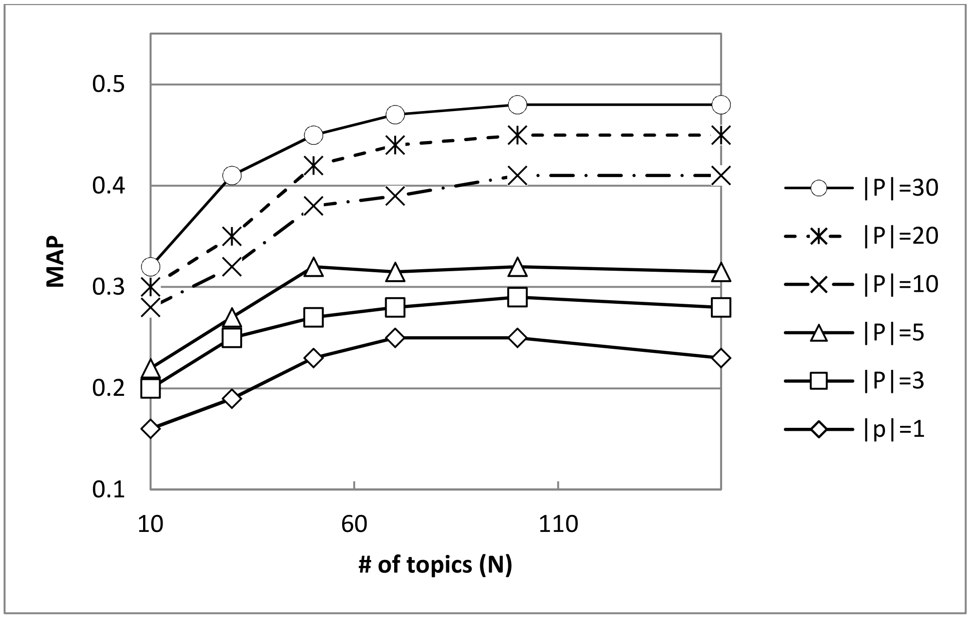


Figure 6.7 MAP under different numbers of topics on 20NG dataset.

6.3.3.3 Performance Comparison. To compare the performance of the topic based Rocchio classifier (TopicRoc) with the term based one (TermRoc), experiments were conducted with α=|PU|/|NU| =20% and 10% to simulate the real situations of online search, where the proportion of positive examples in the unlabeled set was small.

For a given unlabeled dataset with a specific α (i.e., α=20%), the size of the positive examples changed from 1 to 30. Both MAP and P@10 were recorded. The experimental results are shown in Figure 6.8 to Figure 6.19, where TopicRoc(N=X) denotes the topic based Rocchio method using X as the topic size.

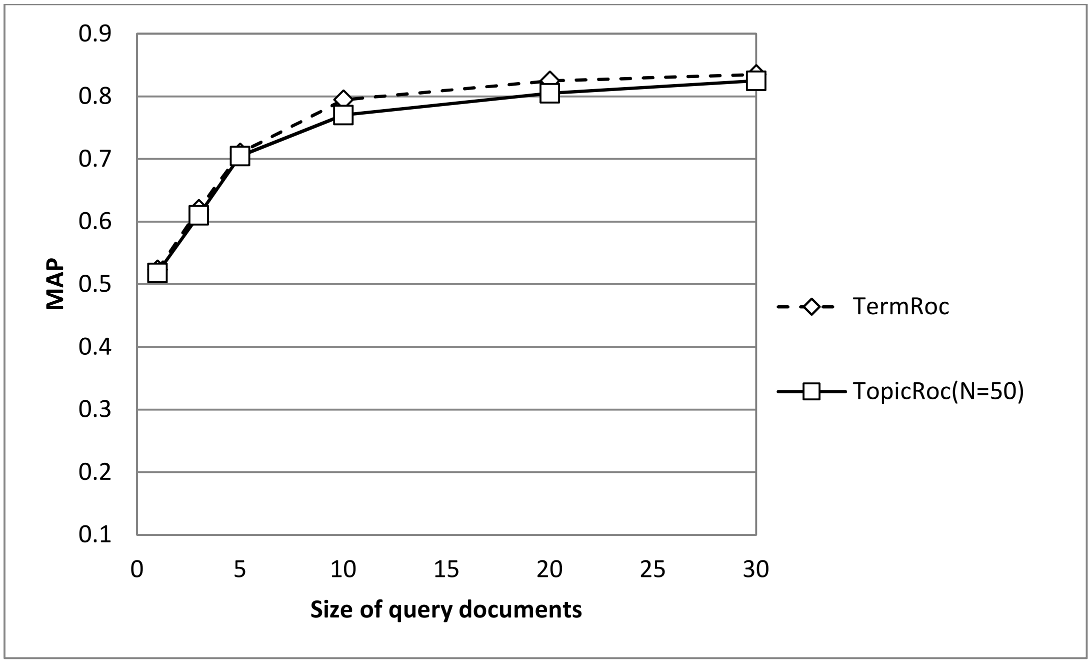
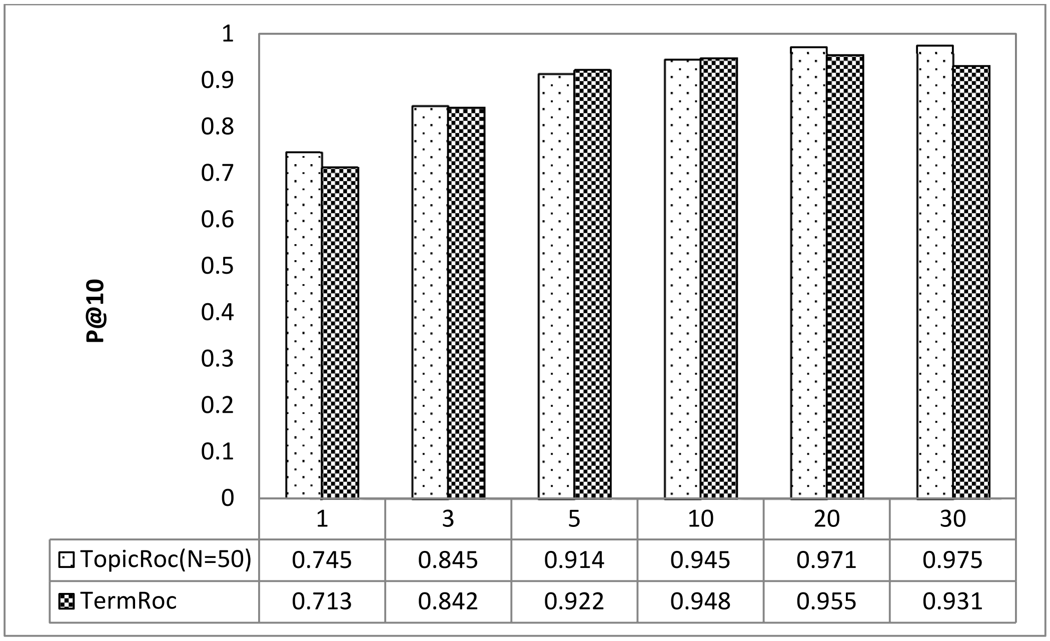


Figure 6.8 MAP under different numbers of query documents on Reuters dataset. (α=20%).



	1	3	5	10	20	30
□ TopicRoc(N=50)	0.745	0.845	0.914	0.945	0.971	0.975
▩ TermRoc	0.713	0.842	0.922	0.948	0.955	0.931

Figure 6.9 P@10 under different numbers of query documents on Reuters dataset. (α=20%).

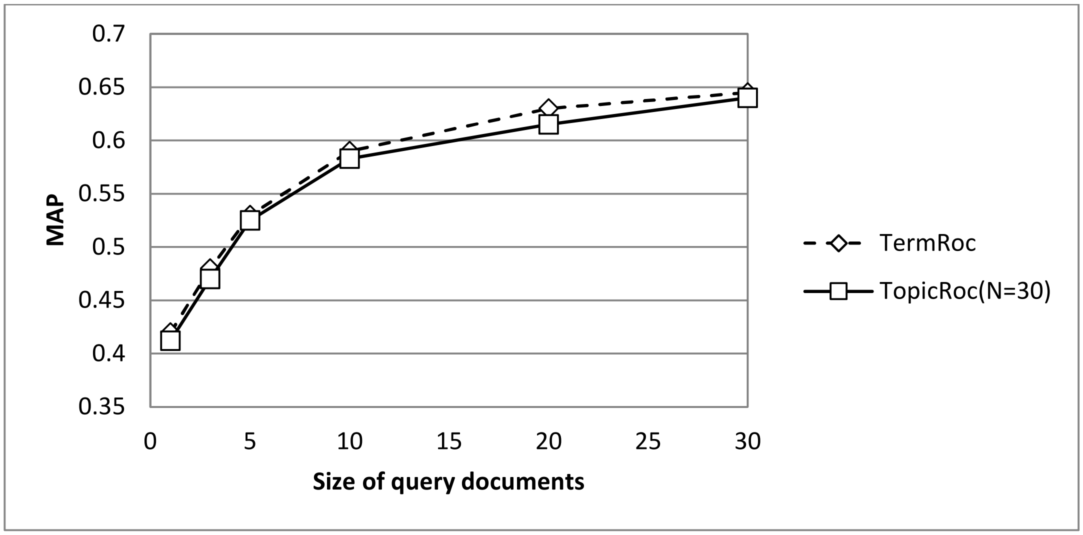
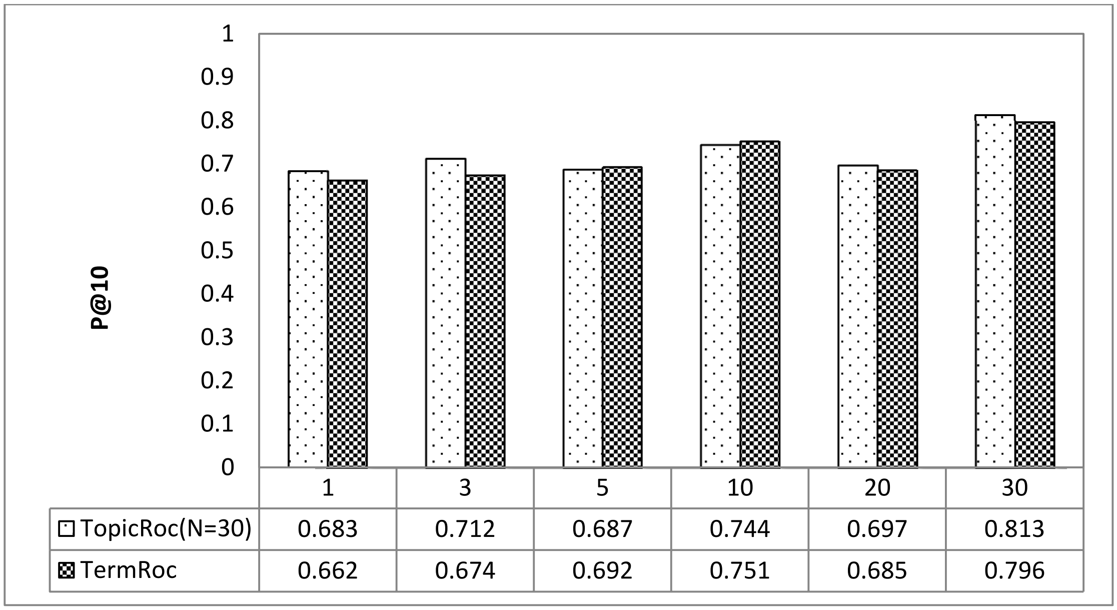


Figure 6.10 MAP under different numbers of query documents on WebKB dataset (α=20%).



	1	3	5	10	20	30
⊡ TopicRoc(N=30)	0.683	0.712	0.687	0.744	0.697	0.813
▩ TermRoc	0.662	0.674	0.692	0.751	0.685	0.796

Figure 6.11 P@10 under different numbers of query documents on WebKB dataset (α=20%)

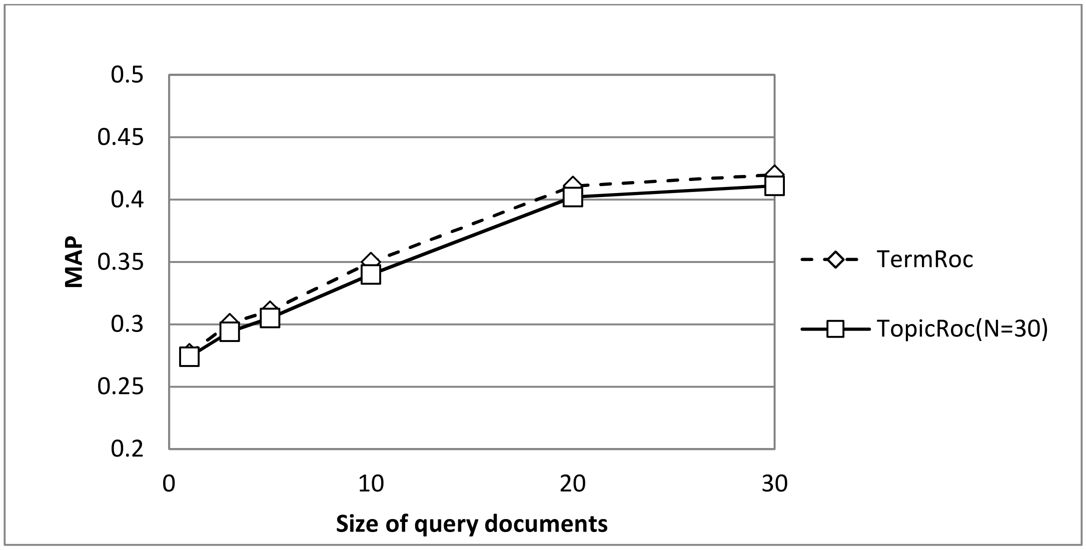
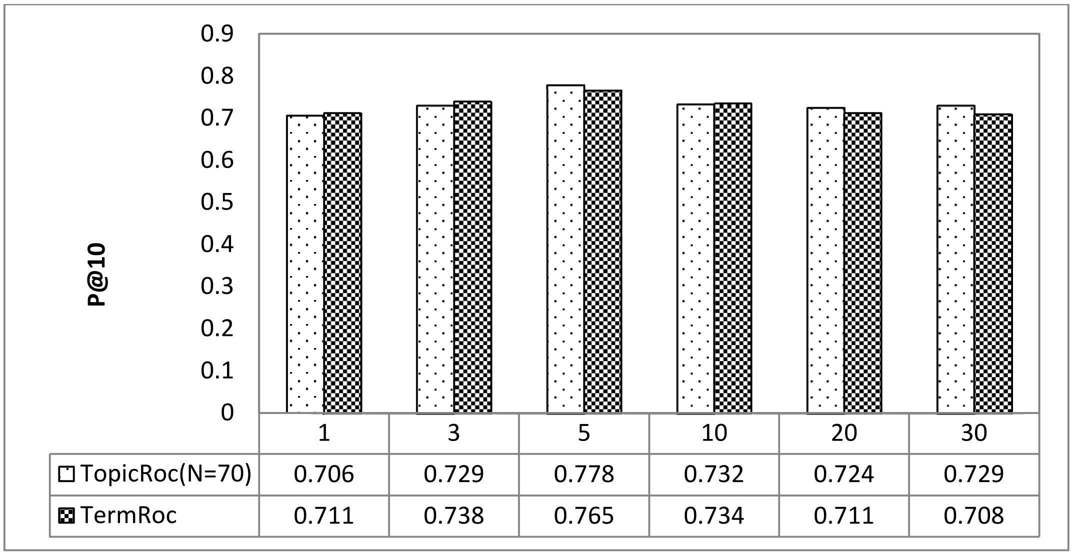


Figure 6.12 MAP under different numbers of query documents on 20NG dataset (α=20%).



	1	3	5	10	20	30
□ TopicRoc(N=70)	0.706	0.729	0.778	0.732	0.724	0.729
▩ TermRoc	0.711	0.738	0.765	0.734	0.711	0.708

Figure 6.13 P@10 under different numbers of query documents on 20NG dataset (α=20%)

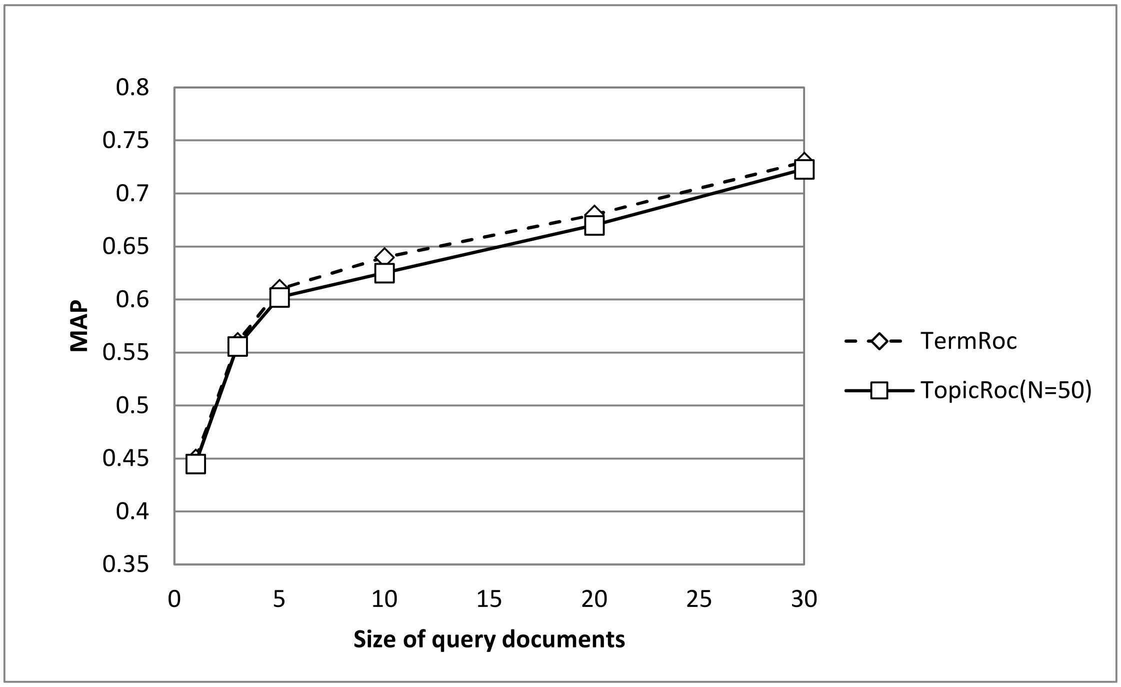
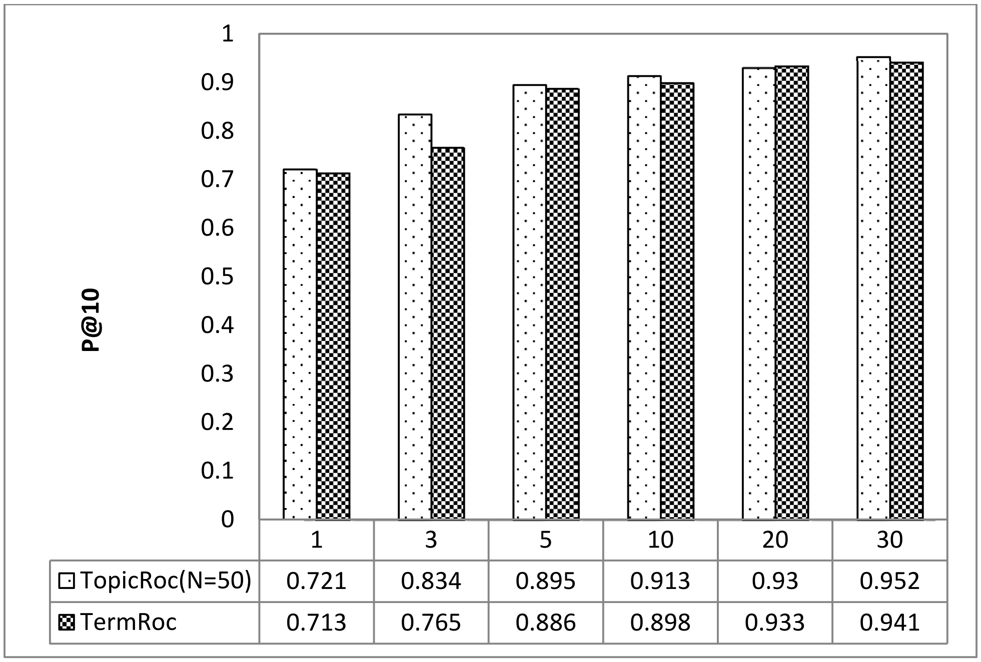


Figure 6.14 MAP under different numbers of query documents on Reuters dataset (α=10%).



	1	3	5	10	20	30
⊡ TopicRoc(N=50)	0.721	0.834	0.895	0.913	0.93	0.952
▩ TermRoc	0.713	0.765	0.886	0.898	0.933	0.941

Figure 6.15 P@10 under different numbers of query documents on Reuters dataset (α=10%).

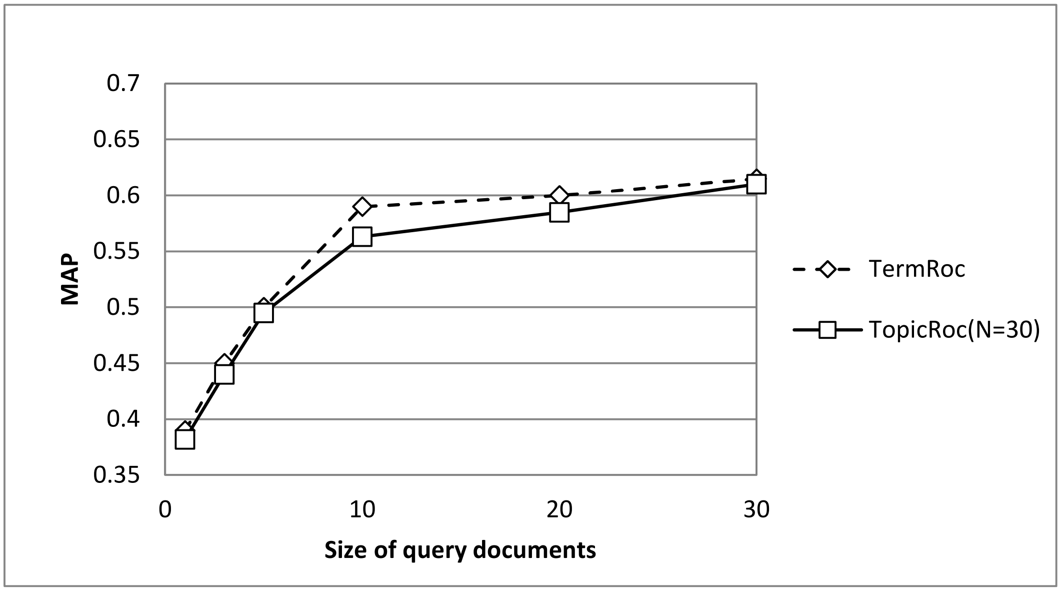
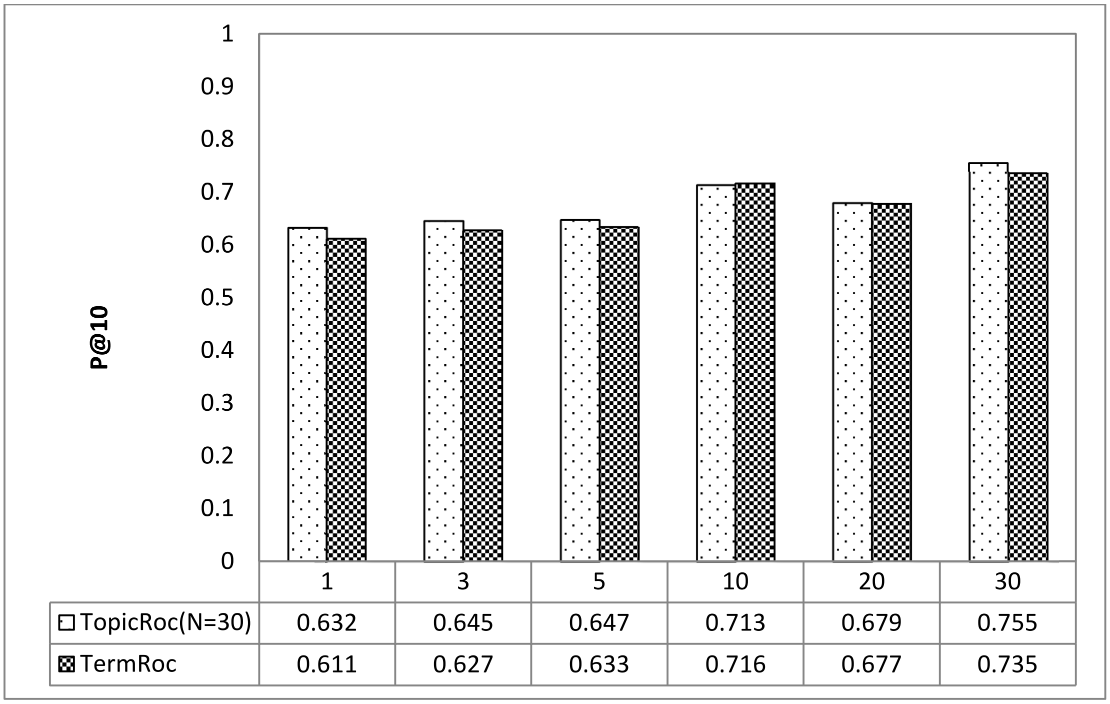


Figure 6.16 MAP under different numbers of query documents on WebKB datasets (α=10%).



	1	3	5	10	20	30
□ TopicRoc(N=30)	0.632	0.645	0.647	0.713	0.679	0.755
▩ TermRoc	0.611	0.627	0.633	0.716	0.677	0.735

Figure 6.17 P@10 under different numbers of query documents on WebKB dataset (α=10%).

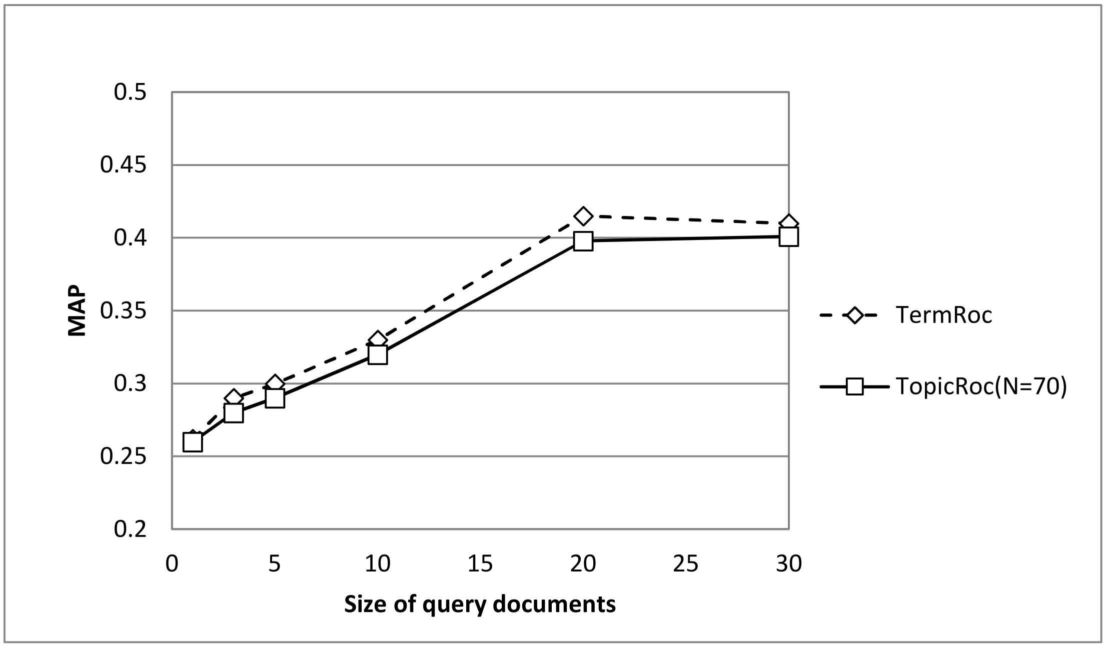
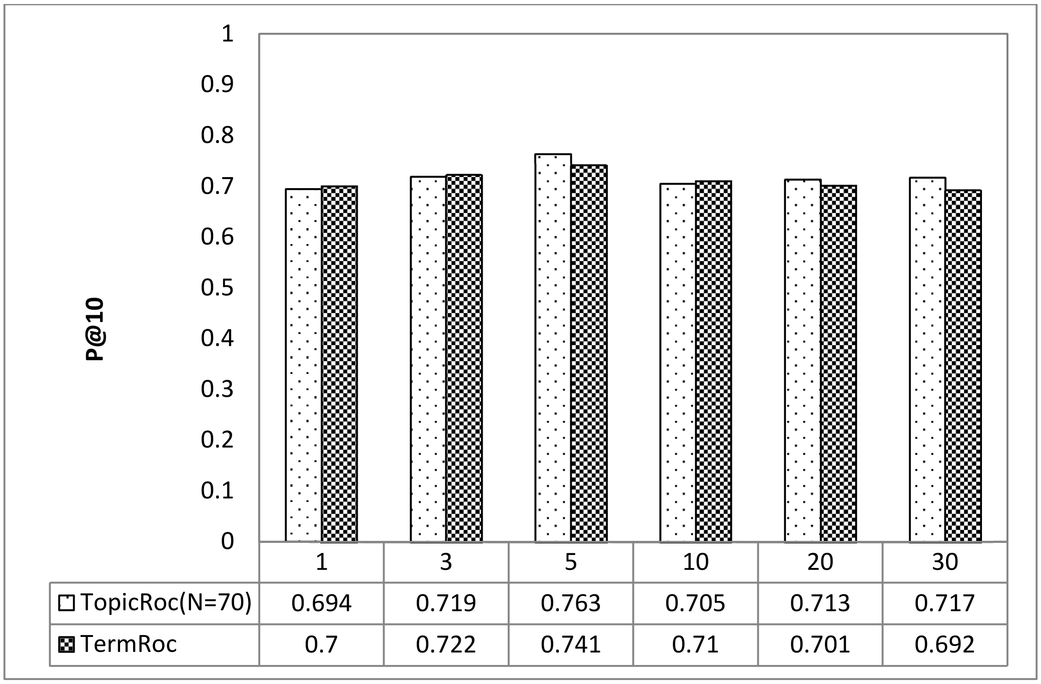


Figure 6.18 MAP under different numbers of query documents on 20NG datasets (α=10%).



	1	3	5	10	20	30
□ TopicRoc(N=70)	0.694	0.719	0.763	0.705	0.713	0.717
▦ TermRoc	0.7	0.722	0.741	0.71	0.701	0.692

Figure 6.19 P@10 under different numbers of query documents on 20NG dataset (α=10%).

When MAP was used as the evaluation measure, it can be observed that the two methods had comparable performance in all the cases. For example, when |P| was smaller than 10, the two methods had almost the same performance. However, when p@10 was used as the evaluation measure, it can be observed that the topic based method performed as well as or even better than the term based method. For instance, from Figure 6.9, it can be observed that the topic based method performed better than the term based method when |P|=1, 20 and 30. In Figure 6.19, it can be observed that the topic based method performed better than the term based method when |P|=5 and 30. Since users usually pay more attention on the top ranked results, the P@10 measure is more useful in evaluating the effectiveness of a ranking system when the size of the potential relevant documents is large. The experimental results using the p@10 measure indicates that the topic based method is effective.

6.3.3.4 Efficiency Comparison Since the size of features used in the topic based method is much smaller than the size of the features used in the term based method, the TopicRoc should be more efficient than the TermRoc. Another experiment was conducted to show the difference of the efficiency between the two methods. Let |PU|=60 and |NU|=1000, and the features used for the term based method 4000, 3000 and 5000, respectively for the Reuters, WebKB and 20NG dataset, the experimental results are listed in Table 6.1.

It can be observed that the topic based method is more efficient than the term based method. For instance, on Reuters dataset, the topic based method took less than 40% the time used by the term based method. In practice, the number of terms in a real IR system such as PubMed is much larger than 4000, which indicates that the topic based method is more efficient than the term based method in reality.

Table 6.1 Computation Efficiency Comparisons Between the Topic Based Method and the Term Based Method, the Time Unit is Millisecond (ms)

|P|	Dataset	TopicRoc	TermRoc
|P|=5	Reuters	28.3	74.6
	WebKb	25.2	65.3
	20NG	41.2	95.5
|P|=20	Reuters	29.6	79.1
	WebKb	27.5	67.4
	20NG	44.7	99.6
|P|=30	Reuters	29.9	84.5
	WebKb	27.9	72.8
	20NG	47.2	103.8

6.4 Discussion

The experimental results from the study indicate that the proposed method using topic model for feature dimension reduction outperforms the term based method using p@10 for evaluation. Using MAP measure, the proposed method has comparable performance with the term based method. Such results indicate the effectiveness of using topic models for document representation in the PU learning based SBME system.

One advantage of the proposed method is that the number of features used for representing the document vector is much smaller than the number of terms required in the term based method. For instance, on Reuters dataset, the proposed method only requires 50 topic features, but the term based method requires 4000 terms as features. From the perspective of computation efficiency, the proposed method is superior to the term based method. In real IR systems, where the size of terms is much larger, the topic based method will be far more efficient than the term based method.

One concern of the proposed method is that it needs topic model training and topic inference of transferring all the documents in the IR system into topic vectors, which requires lots of computing resources. However, the topic model training and inference can take place offline, so it has no effect on the online part of the system.

It should be noted that the dataset for training the topic model is of particular importance of discovering appropriate latent topics for the documents in the IR system. If the dataset is not representative to the documents in the IR system, the derived topic vectors may be misleading. As a result, the topic based PU learning algorithms will have poor results. In this research, a subset of data from the IR system was sampled for topic model training, and it turned out that this method worked well.

Using topic model for documents analysis has attracted lots of interests before. These approaches are different in how they use the topics from topic models. This work is the first of using topic model for document representation in a PU learning based IFME framework. This research indicates the potential of using topic models to represent documents for other tasks such as clustering analysis.

6.5 Summary

In this chapter, a PU learning based method is proposed using topic models for feature selection. LDA is adopted to transfer documents into topic vectors such that the similarity calculation between any two documents is conducted in the topic level. Experimental results indicate that the proposed method has comparable performance with the baseline method in terms of effectiveness but outperforms the baseline method significantly in terms of efficiency.

CHAPTER 7

SUMMARY, LIMITATIONS, DISCUSSION AND CONTRIBUTIONS

This chapter first summarizes the major findings of the study, and outlines the limitations of this research. Then the discussions and contributions of this study are presented.

7.1 Summary of the Study Results

The aim of this research is to explore how to improve the effectiveness of SBME through the adoption of topic modeling, positive unlabeled learning, under-sampling and ensemble learning.

In Chapter 3, this study proposes to conduct topic analysis on query examples to solve the topic diversity issue. Topic distribution information is adopted to predict user's true information needs from the query examples. The experimental results on three benchmark datasets show that the proposed topic model based information need prediction method works well in identifying user's true information needs from the query examples.

In Chapter 4, an under-sampling and ensemble learning based method is proposed to solve the class imbalance problem in the transductive PU learning based document ranking system. The experimental results on three benchmark datasets show that the proposed approach outperforms the baseline method significantly.

In Chapter 5, many Positive Unlabeled learning algorithms for document classification were discussed; and their performance in document ranking in IFME was compared, where they were evaluated using MAP and p@k. This study also proposes a new ensemble learning based approach. The experimental results on three benchmark datasets suggest that the state-of-the-art PU learning algorithms do not have the best performance in the proposed IFME framework and the proposed ensemble learning based algorithms outperform the baseline methods significantly.

In Chapter 6, previous studies on vector space model based SBME were discussed. Such a term based method brings high dimensionality problems. This research adopts topic model to transfer documents into topic vectors to reduce feature dimension. The experimental results on three benchmark datasets indicate that the proposed method performs as well as the baseline method in terms of MAP and outperforms the baseline method significantly in terms of efficiency

7.2 Limitations

This section presents the main limitations of this research.

7.2.1 Limitation Regarding Dataset Size

In this study, three benchmark datasets in ***text classification*** are adopted to evaluate the effectiveness of the proposed methods. Ideally, a much larger dataset should have been used for evaluation in this study. However, the existing large datasets in IR are built for keyword-based search, where user's information need is represented as a string of keywords. There is a lack of information about the topics of the documents in the IR datasets. As a result, this research adopted the widely used datasets in text classification, which are well labeled but smaller in size. Since the evaluation datasets were small, the efficiency issue was not formally evaluated. One resulted concern of the study is that the proposed method may not be efficient when it is applied on a large dataset. However, a searching space reduction method in IFME framework is proposed in Chapter 5, which can solve the efficiency issue when a larger dataset is adopted.

7.2.2 Using Simulated Query Examples in the Evaluation

In the IFME framework, users provide query examples to represent their information need. One goal of this research is to adopt topic analysis techniques to solve the topic diversity issue for better modeling user's information needs. To systematically evaluate the system effectiveness under many different experimental conditions and across three experimental datasets, query examples were built by randomly sampling a set of documents under a topic in a benchmark dataset. The randomly sampled query examples, as opposed to real user-generated query examples, thus were used to represent an (potentially complicated) information need out of necessity. More discussion on the evaluation is made in Section 7.3.1.

7.2.3 Cold Start Problem

IFME requires a user to provide query examples as input. When a user starts to investigate a new information need, s/he might not have positive samples to serve as query examples. This problem is called the cold start problem which is prevalent in recommender systems research. One possible solution is that when a user starts to use the system, the search activities can be recorded in the search log, with clicked documents considered query examples automatically, at least initially. The system can also ask users to provide explicit feedback: when a user starts a search, explicit relevant feedback documents can be used to form the query examples, or the top ranked documents can be used as the implicit feedback to form the query examples.

7.3 Discussion

7.3.1 Evaluation and Performance

In this study, a single topic relevance judgment based method is adopted to evaluate the system performance, because the documents in the datasets are labeled as belonging to only one topic. However, it is unknown which subtopics the simulated query examples belong to or even the number and coverage of the subtopics. It is possible that the sampled query examples may contain too many different subtopics (too diversified) than if the query examples were generated by real users. An observation from the experimental results is: when the size of the query examples was large, the system effectiveness tended to decrease. A possible reason is that, although the top ranked documents might belong to the subtopics of the query examples and be relevant, with single topic based evaluation datasets, they could not fall into the relevant category. It is possible that, had experimental subjects and real user-generated query examples were used in the evaluation, the performance might have been better than what has been reported.

7.3.2 Topic Model Training

One goal of the study is to explore how to use topic information from the query examples to better model user's information needs. To use topic models for document analysis, the number of topic K should be predefined. This process is called topic model training, which should be performed on a large dataset that is selected based on the following two criteria: 1) it must be large enough to contain as many topics as possible; 2) it should be able to represent the documents in the collection of the search system. In this study, parameter selection studies were conducted by randomly sampling a subset of documents from a data collection to investigate how the system performance changed when K varied. The experimental results show that the change of K indeed affects the effectiveness of the ranking system. For example, the experimental results on Reuters dataset indicate that: 1) the system performance downgrades when K is less than 20 or larger than 30, and 2) on average, setting K=20 on the sampled Reuters dataset leads to the highest MAP of the proposed method. In practice, once the system is deployed into an online system, a topic model can be trained using a comprehensive dataset, such as Wikipedia. However, if the document collection is domain specific, only the subset of Wikipedia pertaining to the subject areas of the document collection should be used to train the topic model.

7.3.3 Efficiency Issue for the Under-Sampling based Method

In the under-sampling based method, when |U| (the size of unlabeled data) is large, which is often the case in an online search environment, N (the number of times to perform under sampling) should be large enough to make sure that the expected number of times an instance in U is sampled is not too small. However, the larger N is, the less efficient the system performance will become. In the future, distributed computing methodologies can be adopted to do the random sampling based learning in a parallel manner. For instance, a controller can be used to generate N training datasets, which will be assigned to other nodes of the clusters for training and prediction. The predicted results will be sent back to the controller such that the final ranking scores of the documents in the unlabeled data can be calculated and the unlabeled documents can be ranked accordingly.

7.4 Contributions

In document search, it is often difficult for users to express their information needs as a set of keywords. The proposed work tries to overcome this problem through the design and development of a framework that facilitates users to search by using multiple examples. The research involves: 1) studying whether topic analysis can be used to solve the topic diversity issue by predicting use's true information needs that are expressed as multiple examples, 2) identifying the state-of-the-art PU learning algorithms for IFME and investigating the feasibility of adopting ensemble learning to improve the system performance further, 3) better evaluating IFME with the size of query examples varying, and 4) adopting under sampling to solve the class imbalance problem.

The resulted system can be deployed in online digital libraries such as Google Scholar and PubMed. This will bring significant convenience for researchers to keep up to date their knowledge of their research interests. For instance, it will be helpful for researchers who need to collect documents for a literature review.

The proposed work also offers novel methodologies that are helpful for conducting text analysis. For instance, it provides insights to domain experts on constructing text datasets for a special purpose. In supervised learning, in order to make sure the learned model has a good performance, domain experts often need to build a large set of training data, which is often time consuming and tedious. The resulted system from this research can be helpful for them to get more relevant documents using the documents at hand.

Furthermore, the utilization of under-sampling for solving the class imbalance problem and the investigation of how to use topic modeling for users' information needs prediction can also provide inspirations to the researchers in the community.

REFERENCES

Agichtein, E., Brill, E., & Dumais, S. (2006). Improving web search ranking by incorporating user behavior information. *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 19-26.

Altun, Y., Belkin, M., & Mcallester, D. A. (2005). Maximum margin semi-supervised learning for structured variables. *Advances in neural information processing systems*, pp. 33-40.

Angluin, D., & Smith, C. H. (1983). Inductive inference: Theory and methods. *ACM computing surveys (CSUR), 15*(3), 237-269.

Barbic, F., & Choy, D. M. (1989). Text search system: *Google Patents, U.S. Patent 4,823,306.*

Blei, D. M., & Lafferty, J. (2006). Correlated topic models. *Advances in neural information processing systems, 18*, 147.

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

Borzsony, S., Kossmann, D., & Stocker, K. (2001). The skyline operator. *Proceedings of the 17th International Conference on Data Engineering,* pp. 421-430.

Buckland, M. K., & Gey, F. C. (1994). The relationship between recall and precision. *Journal of the American Society for Information Science (JASIS), 45*(1), 12-19.

Calvo, B., Larrañaga, P., & Lozano, J. A. (2007). Learning Bayesian classifiers from positive and unlabeled examples. *Pattern Recognition Letters, 28*(16), 2375-2384.

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST), 2*(3), 27.

Craven, M., McCallum, A., PiPasquo, D., Mitchell, T., & Freitag, D. (1998). Learning to extract symbolic knowledge from the World Wide Web. *Proceedings of the 5th National Conference on Artifical Intelligence.*

Croft, W. B., Metzler, D., & Strohman, T. (2010). Search engines: Information retrieval in practice. *Addison-Wesley Reading,* pp. 351-358.

Denis, F., Gilleron, R., & Tommasi, M. (2002). Text classification from positive and unlabeled examples. *Proceedings of the 9th International Conference on*

Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU'02.

Duh, K., & Kirchhoff, K. (2008). Learning to rank with partially-labeled data. *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval,* pp. 251-258.

Eiron, N., & McCurley, K. S. (2003). Analysis of anchor text for web search. *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval,* pp. 459-460.

El-Arini, K., & Guestrin, C. (2011). Beyond keyword search: discovering relevant scientific literature. *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining,* pp. 439-447.

Elkan, C., & Noto, K. (2008). Learning classifiers from only positive and unlabeled data. *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining,* pp.213-220.

Eyal, A., & Aposporos, G. (2004). Streaming media search system: Google Patents.

Fagin, R., Lotem, A., & Naor, M. (2003). Optimal aggregation algorithms for middleware. *Journal of Computer and System Sciences, 66*(4), 614-656.

Fidel, R. (1991). Searchers' selection of search keys: II. Controlled vocabulary or free-text searching. *Journal of the American Society for Information Science (JASIS), 42*(7), 501-514.

Fonseca, B. M., Golgher, P. B., de Moura, E. S., & Ziviani, N. (2003). Using association rules to discover search engines related queries. P*roceedings of the First Conference on Latin American Web Congress*, pp. 66-71.

Foote, J. (1999). An overview of audio information retrieval. *Multimedia Systems, 7*(1), 2-10.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *the Journal of machine Learning research, 3*, 1289-1305.

Fuhr, N. (1992). Probabilistic models in information retrieval. *The Computer Journal, 35*(3), 243-255.

Griffiths, T. L., & Steyvers, M. (2002). A probabilistic approach to semantic representation. *Proceedings of the 24th annual conference of the cognitive science society,* pp. 381-386.

Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National academy of Sciences of the United States of America, 101*(Suppl 1), 5228-5235.

Hearst, M. A., Dumais, S., Osman, E., Platt, J., & Scholkopf, B. (1998). Support vector machines. *Intelligent Systems and their Applications, IEEE, 13*(4), 18-28.

Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. *Machine learning, 42*(1-2), 177-196.

Jensen, L. J., Saric, J., & Bork, P. (2006). Literature mining for the biologist: from information retrieval to biological discovery. *Nat Rev Genet, 7*(2), 119-129.

Joachims, T. (1996). A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. *Proceedings of the 14th International Conference on Machine Learni*ng, pp. 143-151.

Joachims, T. (1999). Transductive inference for text classification using support vector machines. *Proceedings of the 16th International Conference on Machine Learning,* pp. 200-209.

Joachims, T. (2002). Optimizing search engines using clickthrough data. Joachims, *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 133-142.

Kotsiantis, S., Kanellopoulos, D., & Pintelas, P. (2006). Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering, 30*(1), 25-36.

Kriegel, H.-P., Kröger, P., & Zimek, A. (2009). Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data (TKDD), 3*(1), 1.

Li, X., & Liu, B. (2003). Learning to classify texts using positive and unlabeled data. *Proceedings of the 18th international joint conference on Artificial intelligence,* pp.587-592.

Li, Y., Li, F., Yi, K., Yao, B., & Wang, M. (2011). Flexible aggregate similarity search. *Proceedings of the 2011 ACM SIGMOD international conference on management of data*, pp. 1009-1020.

Ling, C. X., & Sheng, V. S. (2010). Class Imbalance Problem *Encyclopedia of Machine Learning* (pp. 171-171): Springer.

Liu, B. (2007). Web data mining: exploring hyperlinks, contents, and usage data. *Springer Science & Business Media*.

Liu, B., Dai, Y., Li, X., Lee, W. S., & Yu, P. S. (2003). Building text classifiers using positive and unlabeled examples. *Proceedings of the thrid IEEE International Conference on Data Mining*, pp. 179-186.

Liu, B., Lee, W. S., Yu, P. S., & Li, X. (2002). Partially supervised classification of text documents. *Proceedings of the Nineteenth International Conference on Machine Learning,* pp. 387-394

Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to information retrieval (Vol. 1): *Cambridge university press Cambridge*.

Maybury, M. T. (1997). Intelligent multimedia information retrieval: *MIT Press*.

Nigam, K., McCallum, A., Thrun, S., & Mitchell, T. (1998). Learning to classify text from labeled and unlabeled documents. *Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*, pp. 792-799.

Noto, K., Saier Jr, M. H., & Elkan, C. (2008). Learning to find relevant biological articles without negative training examples *AI 2008: Advances in Artificial Intelligence* (pp. 202-213): Springer.

Papadias, D., Tao, Y., Mouratidis, K., & Hui, C. K. (2005). Aggregate nearest neighbor queries in spatial databases. *ACM Transactions on Database Systems (TODS), 30*(2), 529-576.

Radlinski, F., Broder, A., Ciccolo, P., Gabrilovich, E., Josifovski, V., & Riedel, L. (2008). Optimizing relevance and revenue in ad search: a query substitution approach. *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval,* pp. 403-410.

Raghavan, V. V., & Wong, S. M. (1986). A critical analysis of vector space model for information retrieval. *Journal of the American Society for information Science, 37*(5), 279-287.

Razente, H. L., Barioni, M. C. N., Traina, A. J. M., Faloutsos, C., & Traina Jr, C. (2008). A novel optimization approach to efficiently process aggregate similarity queries in metric access methods. *Proceedings of the 17th ACM conference on Information and knowledge management,* pp. 193-202.

Rocchio, J. J. (1971). Relevance feedback in information retrieval. *The SMART retrieval system: Experiments in automatic document processing*.

Rokach, L. (2010). Ensemble-based classifiers. *Artificial Intelligence Review, 33*(1-2), 1-39.

Rui, Y., Huang, T. S., & Chang, S.-F. (1999). Image retrieval: Current techniques, promising directions, and open issues. *Journal of visual communication and image representation, 10*(1), 39-62.

Rui, Y., Huang, T. S., & Mehrotra, S. (1997). Content-based image retrieval with relevance feedback in MARS. *Proceedings of the IEEE International Conference on Image Processing (ICIP).*

Salton, G. (1971). The SMART retrieval system—experiments in automatic document processing, *Prentice-Hall, Inc., Upper Saddle River*.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management, 24*(5), 513-523.

Salton, G., & Buckley, C. (1997). Improving retrieval performance by relevance feedback. *Readings in information retrieval, 24*(5).

Salton, G., Fox, E. A., & Wu, H. (1983). Extended Boolean information retrieval. *Communications of the ACM, 26*(11), 1022-1036.

Schmid, C., & Mohr, R. (1997). Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*(5), 530-534.

Schölkopf, B., Smola, A., & Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation, 10*(5), 1299-1319.

Shen, X., & Zhai, C. (2005). *Active feedback in ad hoc information retrieval.* Shen, X., & Zhai, C. (2005, August). Active feedback in ad hoc information retrieval. *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 59-66.

Song, F., & Croft, W. B. (1999). *A general language model for information retrieval.* Song, F., & Croft, W. B. (1999, November). A general language model for information retrieval. *Proceedings of the eighth international conference on Information and knowledge management*, pp. 316-321.

Steyvers, M., Smyth, P., Rosen-Zvi, M., & Griffiths, T. (2004). Probabilistic author-topic models for information discovery. *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 306-315.*

Tao, T., & Zhai, C. (2004). A two-stage mixture model for pseudo feedback. *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval,* pp. 486-487.

Tao, T., & Zhai, C. (2006). Regularized estimation of mixture models for robust pseudo-relevance feedback. *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval,* pp. 162-169.

Turtle, H. R., & Croft, W. B. (1992). A comparison of text retrieval models. *The Computer Journal, 35*(3), 279-290.

Vapnik, V. N., & Vapnik, V. (1998). *Statistical learning theory* (Vol. 2): Wiley New York.

Wang, A. (2003). An Industrial Strength Audio Search Algorithm. *Proceedings of the International Conference on Music Information Retrieval*.

Wei, X., & Croft, W. B. (2006). LDA-based document models for ad-hoc retrieval. *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 178-185.

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

Xu, Y., & Papakonstantinou, Y. (2005). Efficient keyword search for smallest LCAs in XML databases. *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pp. 527-538.

Yang, Y., & Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. *Proceedings of the Fourteenth International Conference on Machine Learning*, pp.412-420.

Yiu, M. L., Mamoulis, N., & Papadias, D. (2005). Aggregate nearest neighbor queries in road networks. *Knowledge and Data Engineering, IEEE Transactions on, 17*(6), 820-833.

Yu, H., Han, J., & Chang, K.-C. (2004). PEBL: Web page classification without negative examples. *Knowledge and Data Engineering, IEEE Transactions on, 16*(1), 70-81.

Yu, H., Zhai, C., & Han, J. (2003). Text classification from positive and unlabeled documents. *Proceedings of the twelfth international conference on Information and knowledge management*, pp. 232-239.

Zhang, D., & Lee, W. S. (2008). Learning with support vector machines for query-by-multiple-examples. *Proceedings of the 31st annual international ACM*

SIGIR conference on Research and development in information retrieval, pp. 835-836.

Zhang, D., & Lee, W. S. (2009). Query-By-Multiple-Examples using Support Vector Machines. *JDIM, 7*(4), 202-210.

Zhao, Q., Hoi, S. C., Liu, T.-Y., Bhowmick, S. S., Lyu, M. R., & Ma, W.-Y. (2006). Time-dependent semantic similarity measure of queries using historical click-through data. *Proceedings of the 15th international conference on World Wide Web*, pp. 543-552.

Zhu, M., Xu, C., & Wu, Y.-F. B. (2013). IFME: information filtering by multiple examples with under-sampling in a digital library environment. *Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries*, pp. 107-110.

Zobel, J., & Moffat, A. (1998). Exploring the similarity space. *ACM SIGIR Forum*, 32(1):18-34.

Zobel, J., & Moffat, A. (2006). Inverted files for text search engines. *ACM computing surveys (CSUR), 38*(2), 6.