TOPIC MODELS IN INFORMATION RETRIEVAL

A Dissertation Presented

by

XING WEI

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

August 2007

Computer Science

TOPIC MODELS IN INFORMATION RETRIEVAL

A Dissertation Presented

by

XING WEI

Approved as to style and content by:

____

W. Bruce Croft, Chair

____

James Allan, Member

____

Andrew K. McCallum, Member

____

John Staudenmayer, Member

____

Andrew G. Barto, Department Chair
Computer Science

DEDICATION

To my parents and my husband Xuerui

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor, Prof. Bruce Croft, for his continuous support and guidance over the six enjoyable, memorable, wonderful years. I will remember all that I have learned from him, both inside and outside this field. I would also like to thank the other members of my dissertation committee, Prof. James Allan, Prof. Andrew McCallum, and Prof. John Staudenmayer for taking the time to evaluate my dissertation, and whose valuable feedback and comments dramatically improved this dissertation.

I have been fortunate to receive a great deal of encouragement and assistance in this dissertation from all CIIR faculty, students and staff members. I especially appreciate Xiaoyong Liu and Don Metzler's efforts on helping me reproduce their experimental results, and I thank Yun Zhou and Wei Li for providing me with their code or results and all the insightful discussions I had with them. Many other colleagues have also given me tremendous helps at some stage during my Ph.D. studies: Anca Sailer, David Pinto, Hema Raghavan, Vanessa Murdock, Ruchi Mahindru, Gautam Kar, Jimeng Sun, Gang Luo, Chunqiang Tang, Hao Yang, David Mimno, Ramesh Nallapati, Jiwoon Jeon, Mark Smucker, Ron Bekkerman, Fernando Diaz, Giridhar Kumaran, Trevor Strohman, Ben Carterette, Xing Yi, and Niranjan Balasubramanian. I have learned a great deal from all these people, and part of what I have learned is reflected in this dissertation.

Last, but definitely not least, I am grateful to my family. My father, who taught me how to play with a diode as a six years old, ignited my interests in science that I hope would last throughout my lifetime. My brother and my husband, who are both Ph.D. students in computer science, have had many valuable discussions with me during my thesis work. My mother and my parents-in-law always encourage me from the bottom of their hearts and greatly support my work as much as they can.

This work was supported in part by the Center for Intelligent Information Retrieval, in part by NSF grant #CNS-0454018 , in part by The Central Intelligence Agency and the National Science Foundation under NSF grant #EIA-9983215, in part by PAWARSYSCEN-SD grant number N66001-02-1-8903, in part by NSF Grant #IIS-0527159, and in part by Advanced Research and Development Activity and NSF grant #CCF-0205575. Any opinions, findings and conclusions or recommendations expressed in this material are the author's and do not necessarily reflect those of the sponsor.

ABSTRACT

TOPIC MODELS IN INFORMATION RETRIEVAL

AUGUST 2007

XING WEI, B.A., SOUTHEAST UNIVERSITY, CHINA

M.A., SOUTHEAST UNIVERSITY, CHINA

M.A., UNIVERSITY OF MASSACHUSETTS AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor W. Bruce Croft

Topic modeling demonstrates the semantic relations among words, which should be helpful for information retrieval tasks. We present probability mixture modeling and term modeling methods to integrate topic models into language modeling framework for information retrieval. A variety of topic modeling techniques, including manually-built query models, term similarity measures and latent mixture models, especially Latent Dirichlet Allocation (LDA), a formal generative latent mixture model of documents, have been proposed or introduced into IR tasks. We investigated and evaluated them on several TREC collections within presented frameworks, and show that significant improvements over previous work can be obtained. Practical problems such as efficiency and scaling considerations are discussed and compared for different topic models. Other recent topic modeling techniques are also discussed.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

1.1 Motivation

The goal of Information Retrieval (IR) systems is to retrieve relevant information by comparing query and document texts. From the computer's point of view, these texts are often represented simply as "bags" of words. Documents are retrieved using ranking algorithms that involve measuring word overlap. When human beings retrieve information, they use background knowledge to interpret and understand the text and effectively "add in" words that may be missing. Ranking algorithms solely based on matching the literal words that are present will fail to retrieve much relevant information.

There has been much research in IR to address the problem of "vocabulary mismatch". Manual techniques such as using hand-crafted thesauri and automatic techniques such as query expansion and clustering all attempt to provide a solution, with varying degrees of success. All of these techniques can be described as methods for identifying the "topic" or topics being discussed in a query or document text, and then using this knowledge of topics to include semantically related words. There are many possible definitions of a topic, but in this thesis we view a topic as a probability distribution over words, where the distribution implies semantic coherence. In other words, high probabilities in the topic probability distribution mean that the words are semantically related. For example, a topic related to fruit would have high probabilities for the words "fruit", "apple", "orange", and even "juicy". Note that a topic model does not specify precisely what the semantic relationship is between the words, but simply that they are related. The relationships of words can be as simple as the natural connection between “fruit” and “apple”; they can also be some complicated associations such as being related in a certain context of other words. It is our hypothesis that an IR system should be able to automatically build and use topic models to more reliably improve retrieval effectiveness than has been possible with previous techniques. In this thesis, we develop and test generative retrieval models (also called language models) that incorporate topic modeling concepts, including both manually-built topics and topics built automatically using recent work in machine learning.

1.2 Existing Topic Models for Improving Retrieval

A number of topic modeling techniques have been studied in previous IR research:

Manual Thesauri. The earliest method of incorporating topic models in IR was by using terms from hand-crafted thesauri, which are typical manually-built topic models. Manual indexing has often been viewed as a gold standard and a thesaurus as a “correct” way of incorporating new words or phrases, but building a thesaurus is very labor-intensive and it is very difficult to get people to agree on the semantic classifications involved. Inconsistencies and ambiguity in the use of these thesauri have produced poor results when they are used for retrieval experiments. In a manually built thesaurus, two words are usually regarded as related when their meanings have something in common; however, two words are actually also related if they are brought together in the context of a topic or subject of discourse: they are related by their shared reference (Sparck Jones, 1971).

Term Clustering. Given the difficulties of constructing thesauri manually, people hoped to obtain topic models more easily and effectively by automatic data-driven techniques. Many word similarity measures have been developed, including vector-based similarity coefficients (Qiu and Frei, 1993; Sparck Jones, 1971), linguistic-based analysis such as using head-modifier relationships to determine similarity (Grefenstette, 1992), and probabilistic co-occurrence models (van Rijsbergen, 1977; Cao et al., 2005). Post-processing based on the original similarity measure, such as Markov Chain (Cao et al., 2007) and Generalized Latent Semantic Analysis (GLSA, Matveeva, 2005) has also been applied to further analyze the semantic associations between words. These techniques can be used to find "close" terms based on their metrics and group the terms into clusters/topics, or build a word distribution for each term based on the similarity between the term and other words. Other one-step techniques have also been developed to cluster terms such as in a way that the perplexity is minimized (Brown et al., 1992). Thus, topic models can be easily generated (e.g., by replacing each term with all the words occurring in the cluster/topic to which it belongs) to conduct IR tasks. Quite a few interesting retrieval results have been achieved, but due to inconsistent results and the experimental settings, further research is still necessary to clearly conclude how these techniques can be used to improve retrieval performance and how much benefit they can provide, especially on test collections of realistic size.

Document Clustering. Grouping terms is a straightforward approach to finding related words for topic models; grouping documents, at the same time, has also been effectively used to build topic models by either constructing term clusters based on document clusters (Crouch, 1990) or viewing a document cluster as a topic, and then all documents in the cluster having the identical topic model (Croft, 1980; Liu and Croft, 2004). Both term similarity and document similarity reflect semantic connections of words, though they do provide different information. In document clustering, term similarity is not taken into account; in term clustering, although some algorithms are grounded on document-based vectors, each document is treated as an independent element even if some documents are obviously closer than others.

Latent Semantic Analysis. Latent Semantic Analysis or LSA (Deerwester et al., 1990) is an approach that combines both term- and document clustering. LSA usually takes a term-document matrix in the vector space representation (Salton and Mcgill, 1983) as input, and applies singular value decomposition (SVD)-based dimensionality reduction techniques to the matrix. Thus documents and terms are mapped to a representation in the latent semantic space, which is based on topics rather than individual terms and thus much smaller than the original representation space. As an important and novel topic modeling technique, LSA has been heavily cited in many areas including IR and inspired many new research directions. It has been applied into a range of applications and interesting retrieval results on small collections have been achieved with automatic indexing with LSA (Latent Semantic Indexing, LSI) (Deerwester et al., 1990; Dumais, 1995). The technique does, however, have some problems mainly due to its unsatisfactory statistical foundation and computational complexity. Retrieval effectiveness has never been demonstrated on large collections. One problem with the model is that word observations are not real-valued as in the SVD process; on the contrary, natural text is a fundamentally discrete phenomenon.

Probabilistic LSA. The probabilistic Latent Semantic Indexing (pLSI) model introduced by Hoffman (1999) was designed as a discrete counterpart of LSI to provide a better fit to text data and to overcome some deficiencies of LSI. pLSI is a latent variable model that models each document as a mixture of topics. Although there are some problems with the generative semantics of pLSI, Hoffman has shown that pLSI outperformed both a standard term matching method (based on raw term frequencies) and Latent Semantic Indexing (LSI) in the Vector Space Model retrieval framework (Hoffman, 1999). However, the data sets used were very small and not representative of modern IR environments. Specifically, the collections in those experiments only contained a few thousand document abstracts.

Relevance Feedback. In addition to the above topic modeling techniques that construct topics off-line, there are some online techniques based on relevance or pseudo-relevance feedback (Lavrenko and Croft, 2001), which can also address semantic match to some extent. They can be viewed as topic models in general by treating each query as a topic, and topics will be built by analyzing retrieved documents or user feedback, but their motivation is relevance, not semantic relationships of words. Efficiency is a problem for those online models due to the extra round of retrieval to acquire the relevance judgments in the use of relevance feedback model. In our thesis we focus on off-line topic models, i.e., topic models built before hand according to the collection and independent of specific queries. We will compare off-line topic models with relevance feedback models in our study.

1.3 Integrating Topic Models

With the topic models derived from previous methods, texts are reformulated (i.e. usually expanded) to improve the retrieval effectiveness. Both query and document reformulation processes have been investigated. Query reformulation has been extensively studied for its generally good retrieval results, but it has to be done online, and the expanded queries which result in submitting more terms to the system also negatively affects query response time. Document reformulation is transparent to users and more efficient in terms of query response time, although offline processing of the entire collection can be time-consuming and memory-expensive. Through the development of hardware, document expansion has become popular in recent years (Liu and Croft, 2004; Cao et al., 2005). We are more interested in document reformulation for its online efficiency and the space of improvements.

In 1998, Ponte and Croft presented a statistically-principled approach based on a generative model for IR - the language model for IR. It has been confirmed by a number of groups to be a theoretically attractive and potentially effective probabilistic framework for studying information retrieval problems, and then has quickly become one of the most popular frameworks for IR (Croft and Lafferty, 2003; Ponte and Croft, 1998). The language modeling framework has opened up new ways of thinking about the retrieval process, as well as new conceptual views of topic models in information retrieval. This thesis will focus on exploring the usage and effectiveness of topic models, especially probabilistic topic models, in this new probabilistic framework.

Recently, some simple topic models have been examined to do document reformulation in the language modeling framework and their effectiveness has preliminarily been shown. Liu and Croft (2004) demonstrated that the mixture of unigrams model, also known as the cluster model (McCallum, 1999), can achieve significant and consistent improvements over document-based retrieval models across several TREC collections. Cao et al. (2005) utilized the probabilistic co-occurrence model that catches the co-occurrence of two words within a fixed distance, in conjunction with a predefined thesaurus, to build topic models. Significant improvements on a couple of TREC collections were reported. As simple topic models, the mixture of unigrams model generates a whole document from one topic under the assumption that each document is related to exactly one topic, and the probabilistic co-occurrence model always associates all observations of a distinct term with only one topic assuming that all the identical term tokens belong to only one topic. These assumptions may, however, be too simple to effectively model a large collection of documents.

1.4 Research Summary

According to the construction processes, topic models can be manually built or automatic ones which are data driven; according to the unit in the text associating processes, topic models can be term-term associating models or term group associating models:

- Term-term associating models have been developed to model associations between two single terms. In term-term associating models each term, which is recognized by its spelling, is a unit (in the works that phrases are considered, such as (Jing and Croft, 1994) we view a phrase as one term). The assumption behind term-term associating models is that the term is the basic unit of language and one term has only one meaning. This is not a perfect assumption for natural language but it catches the character of language that people tend to use one term to indicate same/similar/related meanings, and simplifies the modeling process.
- Term group associating models are to model associations between two groups of terms, such as passages of text or documents. This avoids the assumption that one term has only one meaning. The spelling of the term is not the basic unit here; instead, many occurrences of the same term may have different meanings, thus different associations. Although the restricted assumption for term-term associating models has been relaxed, new assumptions are usually added to the modeling process of term group associating models, such as in document clustering all occurrences of term tokens in one document are assumed to belong to one unit.

Thus we have four types of topic models: manual term-term association, manual term group association, automatic term-term association, and automatic term group association. Because there are not many variations for manual methods and there has already been much research, such as the manual term-term associations that were investigated in Cao et al. (2005) within the language modeling framework, in this thesis we put manual term-term association and manual term group association in one category and will study manual term group association. Automatic term-term association will be studied from the perspective of the term similarity measure, and automatic term group association will be studied with the latent mixture model, which is the dominant method to model term group associations offline. Now we have three types of topic models and we carry out our study based on this categorization.

(I) Manually-built topic models

Manually-built topic models are constructed by human understanding of language, which is based on pre-defined knowledge and rules. We investigate retrieval performance with topic models constructed manually based on a hand-crafted directory resource. The original query is smoothed on the manually selected topic model, which can also be viewed as an "ideal" user context model. Because the manually-built topic models produce better retrieval performance on a subset of the queries, selective query expansion is applied to improve the overall performance. This work was also published in (Wei and Croft, 2007-RIAO).

Manual processing can usually provide precise and useful information with relatively less noise, but an automatic method is expected to be more effective for many problems related with manual processing (Sparck Jones, 1971), such as incomplete topics due to the labor-intensiveness and lack of experts. Therefore, many automatic topic modeling methods have been developed.

(II) Term similarity measure – automatic term-term association.

Modeling term similarity, also called "term relationships" or "word associations" in the literature, is to obtain the closeness of one term to another.

It is an automatic version of hand-crafted thesauri to catch the association implied in the basic unit of language – the word.

In this thesis we study how to utilize term association measures to do document modeling, and what types of measures are effective in document language models. We also present a probabilistic term association measure, compare it to some traditional methods, such as the similarity co-efficient and window-based methods, in the language modeling (LM) framework. This work was also published in (Wei and Croft, 2007-ECIR).

(III) Latent mixture model – automatic term group association.

Because of the success of statistical approaches to representing text, Information Retrieval has the potential of benefiting from recent advances in the fields of statistical modeling and machine learning. Research in these fields has led to new mathematical models that effectively discover latent "topics" in large text collections. Associations of text are not only dependent on the term itself as the term-term association describes, but also related with its context; thus latent mixture models have been used to model term group association by representing text as a mixture of latent topics, such as in the cluster model, where document (instead of term) associations are considered. One of these models, Latent Dirichlet Allocation or LDA (Blei et al, 2003), has quickly become one of the most popular probabilistic text modeling techniques in machine learning and has inspired a series of research papers (e.g., Girolami and Kaban, 2005; Teh et al., 2004). LDA has been shown to be effective in some text-related tasks such as document classification, but the feasibility and effectiveness of using LDA in IR tasks remains unknown. Possessing fully generative semantics, LDA overcomes the drawbacks of previous topic models such as probabilistic Latent Semantic Indexing (pLSI). Language modeling, which is one of the most popular statistically principled approaches to IR, is also a generative model for IR (Ponte Croft and Lafferty, 2003; Ponte and Croft, 1998), motivating us to examine LDA-style topic models in the language modeling framework. This work was also published in (Wei and Croft, 2006)

Given the encouraging results with topic models in previous work and the potential advantages of new topic models, we want to examine how the new topic models of the three types, especially the latent mixture models such as LDA can be used for information retrieval.

Compared to other models like the popular unigram-based language model approaches which are defined on individual terms, topic models offer a new and interesting means to model documents. However, in most topic models such as LDA, a topic represents a combination of words; and it may not be as precise a representation as words in other models. Therefore the topic model itself (commonly used with a relatively limited number of topics) may be too coarse to be used as the sole representation for IR. Indeed, our preliminary experiments show that directly employing the LDA model or some other topic models hurts retrieval performance. We propose two frameworks to incorporate topic models into the language modeling framework.

(a) Probability mixture model

A probability mixture model is a probability distribution that is a convex combination of other probability distributions. It can be understood as a linear smoothing with topic background and has been widely used in previous IR research.

(b) Term models

Since most topic models associate each word token with a topic, it is natural to connect a topic "feature" with each token. Even if a topic model does not have an explicit connection for each token (for instance, in type (I) topic models we build a manual topic model for each query based on all the query terms), the mapping performed by text modification implicitly defines connections. Topic models then give each word in a document a new feature, such as that, in term clustering, a word belongs to a cluster or a combination of clusters (soft clustering), and in the LDA model, each word is generated from a topic *z*. Based on the fact that the "topic" feature is connected with word sequences word by word, the models to integrate topics are also designed to work in this fashion of "word by word", which are term models associated with each term token. We present a term model with back-off smoothing (TBS) as an example of term models to incorporate topics, which works as a back-off smoothing.

Within the above two frameworks of using topic models for IR, we study the three types of topic models, introduce or propose new ones for each type, investigate the query and document reformulation processes, evaluate their retrieval effectiveness, and discuss efficiency issues.

1.5 Research Contributions

The contributions that this thesis makes to the field of information retrieval are as follows:

- The first study of generative topic models used for representation in information retrieval. We investigate a range of topic models, especially generative topic models, in different manners of text representation in the language modeling framework. Retrieval effectiveness is evaluated and compared.
- The first evaluation of LDA-style topic models with very large text collections. We evaluate LDA and other state-of-the-art LDA-style topic models on several representative TREC collections of reasonable size.
- The first study of the computational efficiency issues with using LDA-style models for retrieval on very large text collections. Efficiency is a problem for many automatic topic models due to the expensive computation related with large text collections. We study the computation complexity of LDA-style topic models, and control the complexity with approximate parameter settings in the LDA training process.
- The first synthesis study and evaluation of older topic modeling techniques such as manually-built thesauri and term association on large scale collections. We propose a term associating method and compare its effectiveness with traditional similarity measures on TREC collections.
- A cost-benefit comparison of simpler topic-modeling techniques like term-term association with LDA-based techniques. Effectiveness and

computation complexity are discussed and compared for different styles of topic models.

1.6 Outline of Thesis

The remainder of this thesis is organized in the following manner. In Chapter 2, we discuss related work of topic models used in Information Retrieval, including manually-built topic models, term-term associations and latent mixture models such as pLSI and the cluster model. We present the probability mixture model and the term model with back-off smoothing to integrate topics in language modeling framework for IR in Chapter 3, and evaluate them with each type of topic models in the following Chapters. In Chapter 4 we study retrieval effectiveness with manually-built topic models, and propose to do selective query expansion after result analysis. In Chapter 5 we present a term association method based on joint probability and test its effectiveness together with other term-term association measures. We investigate LDA on retrieval tasks in Chapter 6, with an analysis of its feasibility and comparison with term association methods. In Chapter 7 several recent topic models are evaluated such as the special words with background model (SWB, Chemudugunta et al., 2006), pachinko allocation model (PAM, Li & McCallum, 2006) and the topical *n*-gram model (TNG, Wang and McCallum, 2005). In Chapter 8 we conclude our work and present future directions.

CHAPTER 2

RELATED WORK

In this chapter we review the research related to this thesis. We discuss manually-built topic models in Section 2.1, term-term associations in Section 2.2, term group associations, including LSA, latent mixture modeling and relevance modeling in Section 2.3, and integration of topic models in Section 2.4.

2.1 Manually-Built Topic Models

Manual processing is one of the earliest topic modeling techniques used in IR. Since the beginning of IR research, people have been trying to manually add in related words to expand the matching of literal terms.

There are various types of manually-built topic models: hand-crafted thesauri are manually-built term clusters, which are term-term associating models; the directory service offered by many web sites is a term group associating model that manually categorizes documents; user feedback can also be viewed as manually-built topic models which categorizes documents by users. The manual approach still attracts considerable interest from the IR community, and open resources like WordNet and the Open Directory project[i] (ODP) have been studied extensively (Cao et al., 2005; Liu et al., 2004; Mandala et al., 1998). Most research, however, has focused on query expansion, and manually-built topic models have not consistently improved retrieval performance.

[i] http://www.dmoz.com/

In our study, we are more interested in document reformulation with off-line topic models. Within the language modeling framework, Cao et al. (2005) reformulate document models using term associations extracted both from a manually built thesaurus (WordNet) and from a co-occurrence based automatic technique, which considers term co-occurrence in a non-overlapping window. They achieve significant improvements over a baseline query likelihood system on some TREC collections. The improvements with WordNet only are not significant.

Manually-built topic models benefit from the precision of manual processing, but require a lot of human labor from linguists or experts. In addition to the problem of labor intensiveness, it is very difficult to get people to agree on the semantic classifications involved. Inconsistencies and ambiguity in the use of these thesauri have produced poor results when they are used for retrieval experiments. Also, it is a fact that human beings tend to stick to obvious principles of classification (Sparck Jones, 1971). They are likely to group words by their direct connections in meaning when working on semantic associations. Thus, in manually built term clusters, two words are usually regarded as related when their meanings have something in common; however, two words are actually also related if they are brought together in the context of a topic or subject of discourse: they are related by their shared reference (Sparck Jones, 1971). In (Sparck Jone, 1971) a nice example was given: “boundary”, “layer” and “flow” look dissimilar, but they are related in the context of aerodynamics.

Therefore, an automatic, instead of the manual mode, is expected to be more effective for topic modeling.

2.2 Term Association

Most automatic approaches to modeling semantic associations of text are based on term co-occurrence or grammatical analysis. Grammatical analysis provides very specific knowledge about term relationships, but it is not as robust as using term co-occurrence (Manning et al., 2007). Accurate but limited knowledge that provides few related terms is unlikely to substantially improve the retrieval output. Term co-occurrence has been widely used in semantic association studies based on the intuition that co-occurring words are more likely to be similar. In term associating models term similarity is measured with the co-efficient of two term-document vectors, which was widely used in earlier work such as term clustering (Sparck Jones, 1971; van Rijsbergen, 1979; Qiu and Frei, 1993). In term group associating models document similarity is measured with the co-efficient of two document-term vectors for document clustering (Liu and Croft, 2004). In this thesis we focus on co-occurrence based techniques, and in this section term associations will be discussed.

Similarity coefficient. One of the traditional types of automatic term associating methods is based on similarity coefficients between two terms. Starting with a basic term-document matrix, similarity coefficients can be obtained between pairs of distinct terms based on co-occurrence of terms in the documents of the collection. Letting d_{ik} represent the weight or value of term t_i in document D_k and d_{jk} represent the weight or value of term t_j in document D_k, a typical similarity measure between term t_i and t_j is given by

$$SIM(t_i, t_j) = \sum_{k=1}^{n} d_{ik} \cdot d_{jk} \qquad (2\text{-}1)$$

where *n* documents are taken into account (Salton, 1989). Sparck Jones (1971) described a few coefficients as similarity measures in term clustering and reported a large number of experiments using automatically built term clusters. She found that, in general, one obtains a better retrieval performance with the aid of term clusters than with the terms alone. Unfortunately, the evidence has not been conclusive (van Rijsbergen, 1979). The work by Minker et al. (1972) did not confirm the findings of Sparck Jones, and in fact they have shown that in some cases keyword clusters can be detrimental to retrieval effectiveness. Salton (1973), in a review of the work of Minker et al. (1972), has questioned their experimental design which leaves the question of the effectiveness of term clusters still to be resolved by further research (van Rijsbergen, 1979). In 1993, Qiu and Frei computed similarity coefficients in the VSM retrieval framework but they did not form strict clusters; instead, they directly used the coefficient in their computation to expand queries. A notable improvement in retrieval effectiveness was reported in their experiments.

Conditional probability. Another important group of word similarity measures is based on estimating the conditional probability of a term given another term. Van Rijsbergen (1979) and Gao et al. (2005) compute the conditional probability by samples of co-occurrence. A non-overlapping window is applied to measure the co-occurrence in (Gao et al., 2005) and a sliding-window method (Hyperspace Analogue to Language, HAL) is described in (Burgess et al., 1998). A typical computation of the probability (the strength of term relationship/similarity) is as follows:

$$P(t_i, t_j) = f(t_i, t_j) / \sum_k f(t_i, t_k) \qquad (2\text{-}2)$$

where $f(t_i, t_j)$ is the frequency of co-occurrences of t_i and t_j, such as in a window. This group of statistically principled methods may fit the language modeling framework better than the vector-based methods, since the language modeling approach is also a statistically principled method. Cao et al. (2005) reformulate documents within the language modeling framework using term associations extracted both from a manually built thesaurus (WordNet) and from an automatic technique based on term co-occurrence in a non-overlapping window. They achieve significant improvements over a baseline query likelihood system on some TREC collections, and obtain better results by further processing the original term-term associations with Markov chains (Cao et al., 2007). The window-based approach, however, always requires an appropriate setting for the window size, and the improvements using only the automatic model are not as impressive.

After term associations are constructed by these methods, some post-processing techniques can be applied to further improve the associations such as in (Cao et al., 2007) and in GLSA (Matveeva, 2005), or to make the results compatible with systems by clustering such as in (Sparck Jones, 1971).

Simple term-term association has significant advantages over term group association considering the offline efficiency of document reformulation. Although a number of studies of the use of term associations and clusters to improve retrieval performance have been conducted, further research is still necessary due to mixed previous results and recent advances in the fields of statistical modeling and information retrieval. The lately developed language modeling approach with a solid theoretical setting is an effective framework for studying IR problems, and has been widely used in many recent studies in IR. We reexamine term associating techniques in the new framework and compare them with more complicated topic modeling techniques such as LDA.

2.3 Term Group Association

2.3.1 Latent Semantic Analysis

Latent Semantic Analysis or LSA (Deerwester et al., 1990) makes use of dimensionality reduction techniques to capture semantic relations among words and documents. LSA usually takes a term-document matrix *X* in the vector space representation (Salton and Mcgill, 1983) as input, and applies singular value decomposition (SVD)-based dimensionality reduction techniques to the matrix, *X=TSD*, where *T* and *D* have orthogonal columns and *S* is diagonal. The small diagonal elements in the *S* will be ignored as "noise", and the new matrix *X'=T'S'D'* is after truncating the corresponding "noise" factors. *X'* can be used to replace *X* as an alternative perhaps better representation for retrieval. Documents and terms are mapped to a representation (*X'*) in the latent semantic space, which is based on topics rather than individual terms and thus much smaller than the original representation space.

As an important and novel topic modeling technique, LSA has been heavily cited in many areas including IR and inspired many new research directions. It has been applied into a range of applications and interesting retrieval results on small collections have been achieved with automatic indexing with LSA (Latent Semantic Indexing, LSI) (Deerwester et al., 1990; Dumais, 1995). The technique does, however, have some problems mainly due to its unsatisfactory statistical foundation and computational complexity. Word observations in text modeling are not real-valued as in the SVD process; on the contrary, natural text is a fundamentally discrete phenomenon. Retrieval effectiveness is not conclusively better and has never been demonstrated on large collections. Dumais (1995) points out that the lack of specificity is a problem of using LSA in retrieval.

2.3.2 Latent Mixture Model

2.3.2.1 Cluster-Based Retrieval

The cluster model, also known as the mixture of unigrams model, has been well examined in IR research. In the cluster model, it is assumed that all documents fall into a finite set of *K* clusters (topics). Documents in each cluster discuss a particular topic *z*, and each topic *z* is associated with a multinomial distribution *P*(*w*|*z*) over the vocabulary. The process of generating a document *d* ($w_1 ... w_{N_d}$) in the cluster model is as follows:

1) Pick a topic *z* from a multinomial distribution with parameter θ_z
2) For $i = 1 ... N_d$, pick word w_i from topic *z* with probability *P(* w_i *|z)*.

The overall likelihood of observing the document *d* from the cluster model is:

$$P(w_1 ... w_{N_d}) = \sum_{z=1}^{K} P(z) \prod_{i=1}^{N_d} P(w_i \mid z) \qquad (2\text{-}3)$$

One of the parameter estimation methods for the mixture of unigrams model is to cluster documents in the collection into *K* groups and then use a maximum likelihood estimate a topic model *P(w|z)* for each cluster. Liu and Croft (2004) adopted this method with a K-means clustering algorithm. They incorporated the cluster information into language models as smoothing. With the new document model they conducted experiments on several TREC collections, finding that cluster-based retrieval performs consistently better across collections. Significant improvements over document-based retrieval were obtained.

The cluster model possesses fully generative semantics, but the assumption that each string (document) is generated from a single topic is limiting and may become problematic for long documents and large collections.

2.3.2.2 Probabilistic Latent Semantic Indexing (pLSI)

The probabilistic Latent Semantic Indexing model, which was introduced by Hoffman (2003) quickly gained acceptance in a number of text modeling applications. pLSI, also called an aspect model, is a latent variable model for general co-occurrence data which associates an unobserved class (topic) variable with each observation (i.e., with each occurrence of a word). The roots of pLSI go back to Latent Semantic Indexing/Analysis (Deerwester et al, 1990). pLSI was designed as a discrete counterpart of LSI to provide a better fit to text data. It can also be regarded as an attempt to relax the assumption made in the mixture of unigrams model that each document is generated from a single topic. pLSI models each document as a mixture of topics. The following process generates documents in the pLSI model:

1) Pick a topic mixture distribution $P(.|d)$ for each document d,
2) Pick a latent topic z with probability $P(z|d)$ for each word token,
3) Generate the word token w with probability $P(w|z)$.

The probability of generating a document *d*, as a bag of words $w_1 ... w_{N_d}$ (N_d is the number of words of document *d*), is:

$$P(w_1 ... w_{N_d}) = \prod_{i=1}^{N_d} \sum_{z=1}^{K} P(w_i \mid z) P(z \mid d) \tag{2-4}$$

Hoffman (1999) applied pLSI to retrieval tasks in the Vector Space Model framework, albeit on small collections. He exploited pLSI both as a unigram model to smooth the empirical word distributions and as a latent space model to provide a low-dimensional document/query representation. Significantly better retrieval performance over the standard term matching method based on the raw term frequencies and Latent Semantic Indexing (LSI) was reported on all four collections, which contained 1033, 1400, 3204, and 1460 document abstracts respectively. The smoothing parameter was optimized by hand for each collection.

Although large improvements were reported, the collection sizes and the document lengths in the collections are far from representative of realistic IR environments, making the effectiveness of the mixture-of-topics model on IR tasks still unclear. In addition, the baseline retrieval model was far from state-of-the-art. The pLSI model itself has a problem in that its generative semantics are not well-defined (Blei et al, 2003); thus there is no natural way to predict a previously unseen document, and the number of parameters of pLSI grows linearly with the number of training documents, which makes the model susceptible to overfitting.

2.3.3 Relevance Model.

As we described in Section 2.1.1, user feedback can be viewed as manually-built topic models, but methods based on pseudo-relevance feedback are automatic term group association techniques. The relevance model presented by Lavrenko and Croft (2001) is a representative technique and has excellent performance.

The key to relevance model retrieval is estimating the relevance model. Each document is then scored for retrieval by the distance of its model to the relevance model. Conceptually, the relevance model is a description of an information need or, alternatively, a description of the topic area associated with the information need. From the query modification point of view, the relevance model is the modified query that has a probability (weight) for every term in the vocabulary (Lavrenko, 2001). It is estimated from the query alone, with no training data, as a weighted average of document models, with the estimates of $P(D|Q)$ serving as mixing weights:

$$P(w|Q) = \sum_{D} P(w|D)P(D|Q) \qquad (2\text{-}5)$$

where *P(D|Q)* is estimated by Bayes Rule:

$$P(D|Q) \propto P(Q|D)P(D) \qquad (2\text{-}6)$$

Since $P(Q)$ does not depend on D, the above proportionality holds. With uniform priors, *P(D)*, the posterior probability *P(D|Q)* amounts to a normalization since we require *P(D|Q)* to sum to 1 over all documents. *P(w|D)* and *P(Q|D)* are from language model and query likelihood retrieval. Then, each document is scored by the KL-divergence of its model to the relevance model.

Relevance modeling provides a formal method for incorporating query modification into the language modeling framework, and this approach has achieved good performance in previous experiments (Lavrenko, 2001). It is an online technique based on pseudo-feedback and can also address semantic match to some extent. It can be viewed as topic models in general by treating each query as a topic, and topics will be built by analyzing retrieved documents, but the motivation here is relevance, not semantic relationships of words. Efficiency is a problem to online models for the extra round of processing. In this thesis we focus on off-line topic models, i.e., topic models built before hand according to the collection and independent of specific queries, but we will compare the off-line topic models with relevance models in our study.

2.4 Integrating Topic Models

2.4.1 Query Reformulation vs. Document Reformulation

Topic models have been used to improve retrieval effectiveness by reformulating queries or documents. Usually the original text is replaced or expanded with its corresponding topics. Some reformulations do not have the clear process of replacing or expansion, but instead the reformulation process is implicit, such as in the spreading activation techniques (Salton and Buckley, 1988; Croft et al., 1989; Croft and Thompson, 1987), in which the expansion is actually acquired during the process of following links between nodes that represent terms or documents.

Query reformulation has been extensively studied with many topic models in various IR frameworks (Fang and Zhai, 2006; Qiu and Frei, 1993; Jing and Croft, 1994; Xu and Croft, 1996; Lavrenko and Croft, 2001). The well-known pseudo-relevance feedback process, which expands the initial query vocabulary by adding terms contained in previously retrieved documents, is one of the best query expansion techniques in terms of retrieval performance (Lavrenko and Croft, 2001). Most query reformulation models do term group association to find terms related to the entire query, which contains more information than individual words and thus may produce better results (Qiu and Frei, 1993; Jing and Croft, 1994). Some query reformulation techniques based on term-term associations such as (Bai et al., 2005) do post-processing to generate associations with the entire query. These query-based expansion processes have to be done online, in that they require an extra processing or even a search in the whole collection (for relevance feedback) for each query, which negatively affects query response time. Also, the efficiency of an IR system depends heavily on the number of terms of the query submitted to the system; query expansion therefore has its disadvantages in spite of the generally good retrieval results.

Document reformulation can be done offline without query inputs, thus being transparent to users and more efficient in terms of query response time. Offline processing, however, can be time-consuming and memory-expensive because it needs to process the associations of every term in every document of the entire collection, which is one of the reasons that document expansion was not popular until recent years. In this thesis, we are more interested in document reformulation for its online efficiency and space of improvements.

2.4.2 Combination

Combining the original text with topic models derived from it is a popular method used to reformulate document models for IR since the topic models themselves are usually not as precise as the original words for retrieval tasks. There are several possible frameworks to do the combination:

Hidden Markov Model. Miller et al. (1999) presented a Hidden Markov Model (HMM) Information Retrieval system. They take the observed data to be the query Q, and posit a separate state for each of several mechanisms of query word generation, for example, state s_1 for choosing a word from the original document and state s_2 for generating a word from the topics of the document. There is a process for each individual document that generates the query words one by one. Under the assumption that the transition probabilities are independent of the previous state for this framework, the probability of a query being produced by a document in an example system with two states will be

$$P(Q \mid D_k \ is \ R) = \sum_{q \in Q} (\alpha_1 P(q \mid s_1) + \alpha_2 P(q \mid s_2)) \qquad (2\text{-}7)$$

To estimate parameters, they assume that the transition probabilities are the same for all documents, and they use maximum likelihood estimation for the output distributions. Then the transition probabilities α_1 and α_2 will be estimated by Expectation Maximization (EM) algorithm with some training examples. This framework can be simplified to a linear combination, which has been widely used to combine several generative mechanisms for IR, and parameter/weight estimation is the key problem in the combination process.

Parametric mixture model. Zhai and Lafferty (2002) applied a parametric mixture model to do the combination, which is also a linear combination and is inherited by many other works such as Cao et al. (2005). They used EM to maximize the probability of generating a query, but for each query a new estimation is needed, which affects the online efficiency of the system without significant performance gain.

Smoothing. Liu and Croft (2004) integrate topic models in another mechanism as a smoothing background.

$$P(w \mid D) = \frac{N_d}{N_d + \mu} P_{ML}(w \mid D) + (1 - \frac{N_d}{N_d + \mu})[\lambda P_{ML}(w \mid cluster) \\ + (1 - \lambda) P_{ML}(w \mid coll)] \qquad (2\text{-}8)$$

where P_{ML} represents the model estimated by maximum likelihood estimation. N_d is the number of word tokens in document *d*. The cluster model is first smoothed with the collection model by a linear smoothing with weight λ, and the document model is then smoothed using the smoothed cluster model by a Dirichlet smoothing with prior μ. Parameters are estimated by maximizing retrieval effectiveness, which is measured by mean average precision (MAP), on one training collection, and applied to all other collections.

All of the above integrating frameworks can be understood as linear combination with different methods to estimate the combination weights. The parameter estimation in the HMM framework is simplified with strict assumptions and the metric to be maximized during the training process is not straightforward to retrieval effectiveness; the EM algorithm for the parametric mixture model has more flexibility by including more parameters, but also has the efficiency problem at the same time; the smoothing integration maximize retrieval effectiveness directly, but how to formulate the smoothing model is totally by experience, and in our experiments Equation (2-8) does not achieve the best performance.

CHAPTER 3

INTEGRATING TOPIC MODELS INTO RETRIEVAL FRAMEWORK

3.1 Introduction

In this chapter we describe the methods of using topic models in IR framework to improve retrieval effectiveness. The language modeling approach (Croft and Lafferty, 2003; Ponte and Croft, 1998; Song and Croft, 1999) was adopted as the IR framework for the following reasons: (1) It is a statistically-principled framework based on a generative model; many of the topic models we study in this thesis, especially the state-of-the-art ones, are also generative models. (2) It has been confirmed by a number of groups to be a theoretically attractive and potentially effective probabilistic framework for studying information retrieval problem (Ponte and Croft, 1998; Berger and Lafferty, 1999). The language modeling framework has opened up new ways of thinking about the retrieval process, as well as new conceptual views of topic models in information retrieval. Its solid theoretical setting and promising experimental results provide and motivate new directions of the construction and integration process of new concepts. (3) It is one of the most popular frameworks for IR, so it is easier to compare with results from the same framework. (4) We have done some preliminary experiments within other frameworks, such as the Vector Space modeling framework, and no improvements over the language modeling framework have been shown. (5) It is very effective so provides a state-of-the-art starting point.

The basic approach for IR in the language modeling framework is the query-likelihood method with a multinomial unigram document model, which is usually estimated by maximum likelihood estimation and smoothed on the entire collection. The multinomial document model is defined on individual terms. Topic models, which represent texts with topics, offer a new and interesting means to model documents. However, since in topic models documents are represented with topics, which is usually a probabilistic combination of words, it may not be as precise a representation as words in other models. Therefore the topic model itself (commonly used with a relatively limited number of topics) may be too coarse to be used as the sole representation for IR. Indeed, our preliminary experiments show that directly employing the LDA model or some other topic models to represent documents hurts retrieval performance.

A probability mixture model and a term model with back-off smoothing are presented to integrate topic models in this Chapter. We are more interested in document modeling, so these two frameworks will be used to reformulate document models with topics. But for manually-built topic models, it is infeasible to build topic models manually for each documents, thus we also explore query reformulation with a probability mixture model to combine the original query and topic models in Chapter 4.

3.2 Document Modeling: Probability Mixture Model (PMM)

A probability mixture model is a probability distribution that is a convex combination of other probability distributions. The combination format has been widely used in IR to integrate various probabilistic models for query representation or document representation. Suppose that P is a mixture of n probability distributions P_i, then

$$P(x) = \sum_{i=1}^{n} \lambda_i P_i(x) \qquad (3\text{-}1)$$

where 0< λ_i < 1 and $\sum_{i=1}^{n} \lambda_i = 1$.

To estimate the mixture weights λ_i, as we described in Chapter 2, in previous works of mixture models for IR Miller et al. (1999) applied a Hidden Markov Model (HMM) framework; Zhai and Lafferty (2002) and Cao et al. (2005) used Expectation Maximization (EM) on the mixture model; Liu and Croft (2004) integrate topic models as background smoothing.

With EM, the parameter estimation process is online, i.e., for each query an EM estimation will be run, which makes retrieval less efficient. In the HMM framework, a reasonable amount of relevant documents are needed to estimate the parameters, which may not be available for some realistic tasks. Also, maximizing the likelihood of observation is not as straightforward as maximizing retrieval effectiveness directly. Considering both efficiency and effectiveness based on previous experience, we maximize Mean Average Precision (MAP), instead of probabilities, on one collection for training, and use the parameters on all other collections. MAP is used as the optimization criterion here because it is our final evaluation metric. This procedure is similar as the parameter estimation process in Liu and Croft (2004) and Metzler and Croft (2005). It is simple, straightforward, efficient, and as effectiveness as those more complicated ones.

3.3 Term Model with Back-off Smoothing (TBS)

3.3.1 Term Models and Document Models

The basic approach to using language models for IR is the query likelihood method where each document is scored by the likelihood of its model generating a query *Q*,

$$P(Q \mid D) = \prod_{q \in Q} P(q \mid D) \qquad (3\text{-}2)$$

where *D* is a document model, *Q* is the query and *q* is a query term in *Q*. *P*(*Q*|*D*) is the likelihood of the document model generating the query terms under the "bag-of-words" assumption that terms are independent given the documents. $P(q_i \mid D)$ is specified by the document model with Dirichlet smoothing (Zhai and Lafferty, 2001),

$$P(w \mid D) = \frac{N_d}{N_d + \mu} P_{ML}(w \mid D) + (1 - \frac{N_d}{N_d + \mu}) P_{ML}(w \mid coll) \qquad (3\text{-}3)$$

where $P_{ML}(w|D)$ is the maximum likelihood estimate of word *w* in the document *D*, $P_{ML}(w|coll)$ is the maximum likelihood estimate of word w in the entire collection, μ is the Dirichlet prior, and N_d is the number of word tokens in document *d*.

Document modeling (estimating *P*(*w*|*D*)) is crucial to retrieval. Traditional language modeling techniques for document-retrieval usually regard a document as a whole, since the entire document is one unit in processing, i.e., retrieval. For instance, in the inference of Dirichlet smoothing, a prior is given to the whole document model, and the observation is the full word sequence of the document. However, with topic models each term token in a document will have a new feature, which is the topic associated with the token. In term clustering, a term belongs to one or more clusters/topics; in the LDA model, each term occurrence is generated from a topic *z*. Based on the fact that the semantic "topic" feature is connected with term sequences word by word (token by token), the models to integrate topics are also constructed to work in this fashion of "word by word", as follows:

For each term token *t* in a document *d*, we define a term model M_t. The word distribution of this model, $P(w \mid M_t)$, represents the probability of generating an arbitrary word *w* from the term model. And the word distribution of the document model will be

$$P(w \mid d) = \sum\nolimits_{t=1}^{N_d} P(w \mid M_t, d) P(M_t \mid d) = \sum\nolimits_{t=1}^{N_d} P(w \mid M_t) P(M_t \mid d) \qquad (3\text{-}4)$$

For computation convenience we choose to use uniform distribution for $P(M_t \mid d)$, then Equation (3-4) will be

$$P(w \mid d) = \frac{1}{N_d} \sum\nolimits_{t=1}^{N_d} P(w \mid M_t) \qquad (3\text{-}5)$$

With term modeling, the semantic "topic" feature is connected with word sequences one token by one token, which provides much flexibility of integrating topic features. Potentially, tokens of the same word in one document can have different meanings, such as in the same document two "apple"s can refer to different meanings – one may be a fruit and the other may be a computer. The term modeling framework is able to handle the difference between word tokens by building a model for each token, which also provides many possibilities of including a variety of other new features and thus makes term modeling a promising framework. In the future, semantic and syntactic features can both be integrated in this term modeling framework.

3.3.2 Term Model with Back-off Smoothing (TBS)

To integrate topics into term models, we apply back-off smoothing, which is often used in *n*-gram language models (Katz, 1987). The probability of word *w* in the term model of token *t* will be,

$$P(w \mid M_t) \propto \begin{cases} 1, & if \;\; w = t \\ Z_t(w), & if \;\; w \neq t \end{cases} \tag{3-6}$$

With Equation(3-5) and (3-6),

$$P(w \mid D) = \frac{1}{N_d} \sum_{t=1}^{N_d} P(w \mid M_t) \propto \frac{D_w \times 1.0 + \sum_{t \neq w} Z_t(w)}{N_d} = \frac{D_w}{N_d} + \frac{1}{N_d} \sum_{t \neq w} Z_t(w) \tag{3-7}$$

The back-off smoothing in Equation (3-6) is to integrate topics. We also need to smooth the document model on general English as most document language models do. In both of PMM and TBS, the duty of smoothing on general English can also be taken by topic models, but instead of constructing and tuning the smoothing parameters in topic models, we apply Dirichlet smoothing on Equation (3-7) and have,

$$P(w \mid D) = \frac{N_d}{N_d + \mu} \times \frac{D_w}{N_d} + \frac{\mu}{N_d + \mu} \times \frac{C_w}{N_C} + \frac{1}{N_d} \sum_{t \neq w} Z_t(w) \tag{3-8}$$

Eq. (3-7) is a non-parameter model. Eq. (3-8) introduces the Dirichlet smoothing parameter μ for convenience. But μ has been shown in many experiments to be a relatively insensitive parameter, which is usually fixed to be 1000 and the best results are often obtained with this setting. In document models (Equation (3-8)) constructed from TBS we found that the results would be slightly better if we lower the value of μ to 500. We fix μ=500 for TBS and μ=1000 for all other cases in this thesis without especially tuning it as a parameter.

3.4 PMM vs. TBS

PMM is a traditional framework in IR to integrate different factors. It can be understood as a linear smoothing from IR point of view. TBS is also a promising framework based on back-off smoothing.

We will illustrate how these two models work and the difference of them with an example: there is a document “Apple pie, cookie”, and this document is associated with 33% *fruit* topic and 67% *baked food* topic from some topic model. The probability of the term “fruit” will be very low in the original document model (it is not exact 0 only because of smoothing), and the probability of the term “pie” will be close to 1/3.

In the PMM document model, the probability of the term “fruit” will be calculated as shown in Figure 3.1.

$$P(w \mid D) = \lambda P(w \mid D_{orig}) + (1-\lambda)P(w \mid D_{topic})$$

$$P(fruit \mid D) = \lambda \underbrace{P(fruit \mid \text{"Apple pie, cookie"})}_{0} + (1-\lambda)\underbrace{P(fruit \mid 33\% \text{ "Fruit" topic} + 67\% \text{ "Baked food" topic})}_{33\% * P(fruit \mid \text{"Fruit" topic}) + 67\% * P(fruit \mid \text{"Baked food" topic})}$$

Figure 3.1: The probability mixture document model.

P(*fruit* | “Fruit” topic) in Figure 3.1 has a high value and thus *P*(*fruit*|*D*) will be adaptively smoothed by the above convex combination. This effect of combination can be controlled by the mixture weight *λ*. The probability of the term “pie” *P*(*pie*|*D*) will be computed in the same way with corresponding contribution from *P*(*pie* | “Fruit” topic) and *P*(*pie* | “Baked food” topic). However, different from the computation shown in the above figure, *P*(*pie* | “Apple pie, cookie”) is not 0.

In the TBS document model the probability of the term "fruit" will be proportional to the average of *P*(*fruit* | apple's topic), *P*(*fruit* | pie's topic) and *P*(*fruit* | cookie's topic); the probability of the term "pie" is illustrated as Figure 3.2 shows, which is proportional to the average of *P*(*pie* | apple's topic), 1 and *P*(*pie* | cookie's topic):

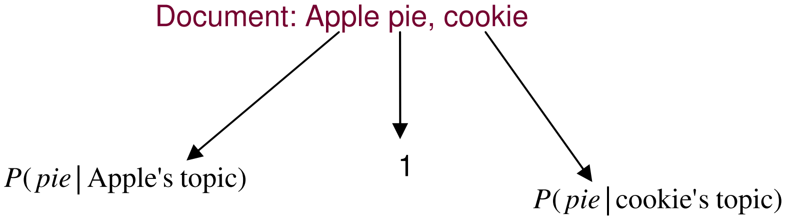

Document: Apple pie, cookie
$P(pie\,|\,\text{Apple's topic})$
1
$P(pie\,|\,\text{cookie's topic})$

Figure 3.2: The TBS document model.

Both of the probability mixture model and the term model with back-off smoothing are frameworks to integrate topics, but their ways to use topics are very different. TBS is an easy-tuned model, which does not introduce any new parameters. PMM, however, has a new parameter λ that needs to be finely tuned but also provides corresponding flexibility.

These two frameworks are used to integrate topic models in the following chapters of this thesis, i.e., Chapter 4, 5, 6 and 7. In Chapter 5 and 6, especially, we run TBS and PTM with the two most popular types of topic models on same data sets to compare their effectiveness.

CHAPTER 4

MANUALLY-BUILT TOPIC MODELS

4.1 Introduction

4.1.1 Topic Models and User Context

In topic models, the semantic properties of text are expressed in terms of topics (Steyvers and Griffiths, 2007), which are represented by probability distributions over words in our study and the distribution implies semantic coherence; topics can thus be used as knowledge background which provides semantically related words to expand the literal matching of words that are present in text. The expanded retrieval algorithms can be applied in various IR applications to compensate for literal word-matching algorithms in two ways:

(1) providing general information to address the "vocabulary mismatch" problem as we described in Chapter 1. The users of IR systems often use different words to describe the concepts in their queries than the authors use to describe the same or relevant concepts in their documents (Xu, 1997), such as a user may use "apple" as a query and a relevant document may contain "McIntosh" only. Both of manual and automatic topic models have been built to make up this gap through query expansion and/or document expansion methods (Sparck Jones, 1971; Qiu and Frei, 1993; Xu and Croft, 1996; Jing and Croft, 1994; Cao et al., 2005; Deerwester et al., 1990; Hoffman, 1999; Lavrenko & Croft, 2001).

Hand-crafted thesauri are early examples of manually built topic models; directory services, which are based on documents, can also provide profiles of the general connections among words. Given the difficulties of constructing topic models manually, people hoped to obtain topic models more easily and effectively by automatic data-driven techniques. The effectiveness of automatic topic models in IR, especially the ones which are not only based on term-term relationships (e.g. document clustering), makes is very interesting to investigate the retrieval performance with manually-built topic models other than hand-crafted thesauri.

(2) providing user specific information to integrate user context. The goal of Information Retrieval is to retrieve documents relevant to a user's information need, and the aim of contextual retrieval is to "combine search technologies and knowledge about query and user context into a single framework in order to provide the most 'appropriate' answer" (Allan, et al., 2002). In a typical retrieval environment, we are given a query and a large collection of documents. The basic IR problem is to retrieve documents relevant to the query. A query is all the information that we have to understand a user's information need and to determine relevance. Typically, a query contains only a few keywords, which are not always good descriptors of content. Given this absence of adequate query information, it is important to consider what other information sources can be exploited to understand the information need, such as context. User context, which includes user related information that reflects topical interests, is an important information source in addition to queries to help in understanding a user's information need and to determine relevance. The query "apple" that was input by a user who has a computer science background may be different from the query "apple" that was input by a user who has a food science background, and topical context can help differentiating these two queries. Contextual retrieval is based

on the hypothesis that context information will help describe a user's needs and consequently improve retrieval performance.

There is a variety of context information, such as query features, user background, user interests, etc. We focus on user related information that reflects topical interests, and we refer to this as user context, which is often simply described as "context" or "user profiles" in other papers. The corresponding research field has been called various names such as "personalized IR", "user modeling", "user orientation", "contextual retrieval", etc. In some cases, context is used to refer to short term user interests with respect to specific queries. User profiles, however, can also be used for longer-term, broad topical interests. In this chapter, we focus on user models representing longer-term topical interests that can be used to improve specific queries.

User context information has received considerable attention recently, especially in commercial search engines. User-oriented analytical studies emerged as early as the 1970's (Belkin and Robertson, 1976; Pejtersen, 1979; Ingwersen, 1992), but it wasn't until the mid-80's that practical "real world" systems were studied (Belkin and Croft, 1987). User oriented approaches and user context information have received more attention recently, including in commercial search engines. For example, Watson (Budzik et al., 2001; Leake et al., 1999) predicts user needs and offers relevant information by monitoring the user's actions and capturing content from different applications, such as Internet Explorer and Microsoft Word. The "Stuff I've Seen" system (Dumais et al., 2003) indexes the content seen by a user and provides contextual information for web searches. Google also featured personal history features in its "My Search History" service Beta version.

Despite the recent focus on this problem, it is still not clear what the benefits of user context are, especially with test collections of realistic size.

4.1.2 Manually-Built Topic Models

Topic models can help the retrieval process by providing additional information to present words, which can be either general knowledge like word meaning/common sense (as the connection between "apple" and "McIntosh") or user oriented information. Although a number of studies have been conducted on these two aspects for topic models as we described in Chapter 2 and Section 4.1.1, the effectiveness of manually-built topic models is still not clear, especially on collections of realistic size. Most recent research on topic models has focused on automatic techniques. To give a broader picture of the potential effectiveness of these approaches, in this chapter we investigate the use of manually-built topic models. In real-world IR applications building topic models by hand is often infeasible due to its prohibitive price. Even the simplistic manual topic representation – hand-crafted thesauri are limited by the construction and maintenance price. However, through the popularization of Internet in recent years, topicalized information like the directory service offered by many web sites, has become a significant information resource with reasonable quality, which makes it easier to build topic models manually and also makes it interesting to see how much improvements we can get from this information. Also, manual processing is flexible and capable of generating appropriate topic models including both general knowledge and user context. So the results can benefit both research directions:

- From the point of view of general topic models, the success of automatically-built topic models (usually built on the experimental collections) makes it interesting to see the performance gain with manual methods. Some semi-manual methods have been applied in previous research based on hand-crafted thesauri (e.g., Kwon et al.,1994; Cao et al., 2005), which can be viewed as a simplistic topic representation. In this chapter we will use manually-constructed directory service, which is a popular topic representation and assign topics/directories to text by hand. So the process can show the effectiveness of fully manual methods, which reflects the potential improvement from using available hand-crafted topic resources.
- From the point of view of user specific topic models, the manually-built topic models can be viewed as "ideal" context models. Compared to the type of user models built by observing user behavior, these models should be more focused and less "noisy". Also, considering that the available resource may contain insufficient information for some topics, in our experiments we discard the queries for which the resource does not contain sufficient data in order to generate "ideal" context models and thus produce an empirical upper bound for retrieval performance gain with user modeling.

In other words, we focus on the potential improvement from using some well-organized and pre-available resource to form topic models for retrieval. In our first experiment we choose the "best" topic model for each query in a set of TREC queries and use this topic model to modify the query using language modeling techniques (Croft and Lafferty, 2003). The topic model provides background information for the query and, in effect, expands the query with related terms. The use of general topic models or context information to expand queries has been used in a number of studies (e.g., Bai et al., 2005; Shen and Zhai, 2003). Topic models are based on categories from the Open Directory project[ii] (ODP). We compare these “ideal” topic models with the performance of relevance models (RMs), which are non-user based topic models constructed automatically for each query using the pseudo-relevance feedback approach.

We then examine differences between these two approaches, and whether they can be combined to give better performance. We also examine techniques for automatically selecting a topic model from the Open Directory categories and compare this to the manual selection and relevance model approaches.

4.2 Effectiveness of Manually-Built Topic Models

To show the potential improvements of the available topic resource and demonstrate an empirical upper bound of using user context in IR, we simulate an “ideal” topic model for each query by selecting the “best” topics for it from the Open Directory project categories. Then we incorporate the model into a language modeling framework as a smoothing or background model for the query. We compare the results with two other techniques in the language modeling framework, which do not use other

[ii] http://www.dmoz.com/

resources or context information, to estimate the potential performance improvement using the context topic models.

In Section 4.3, we examine the combination of the topic model with the relevance model at both model level and query level.

4.2.1 Constructing Topic Models from the Open Directory

To construct the topic model for each query, we manually select the “closest” categories from the Open Directory project, according to some rules to approximate an “ideal” user model.

4.2.1.1 Open Directory Project

The Open Directory project (ODP), also known as DMoz (for Directory.Mozilla, the domain name of ODP), is an open content directory of Web links that is constructed and maintained by a community of volunteer editors. It is the largest, most comprehensive human-edited directory of the Web.

An ontology is a specification of concepts and relations between them. ODP uses a hierarchical ontology scheme for organizing site listings. Listings on a similar topic are grouped into categories, which can then include smaller categories. This ontology has been used as the basis of user profiles for personalized search (Trajkova and Gauch, 2004).

The Open Directory Project homepage claims that their directory contains more than 500,000 categories, some of which are very specific and small. Trajkova and Gauch (2004) use only the top few levels of the concept hierarchy, and further restrict them to only those concepts that have sufficient data (the Web links) associated with them, in their user profile building. In order to build the “best” topic model, we use the whole concept/topic hierarchy, but we ignore the categories that contain insufficent data (less than 5 Web links in our experiments). We currently only retrieve the first-level Web pages mentioned in a category without considering further links, to avoid including irrelevant information, and to make the topic model more focused.

4.2.1.2 Choosing Categories

We want to choose the “closest” categories for a query. “Closest” can be interpreted here as “deepest”, that is, there is no applicable category of the query that is deeper (in the hierarchy structure) than the currently selected one. In Figure 4.1, for example, “Energy” is closer than “Technology” to the query of “hydrogen fuel automobiles” (Topic 382 in the TREC7 ad hoc retrieval task) and “Transportation” is closer than “Energy”, and there is no sub category in “Transportation” that can cover the query. In this example, “Top/Science/Technology/Energy/Transportation/” is selected as one of the “closest” categories. For two categories that do not have direct hierarchical relations, their distances to the query are not comparable and both can be selected. For example, both “Transportation” and “Hydrogen” in Figure 4.1 may be selected.

The above category selection process can be described by two rules:

1) The category should cover the query content.
2) The category should be the closest (deepest in the hierarchical structure) to the query. This provides the most specific/best information in the Open Directory for this query.

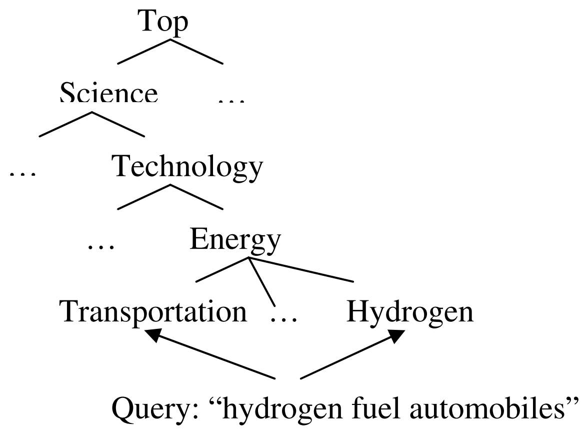


Figure 4.1: An example of hierarchical categories.

4.2.1.3 Constructing Topic Models

After we select the categories for the queries, we download the Web links in the categories we chose. As we said in Section 4.2.1.1, we download only the first-level pages in the Web links. Then we have a topic collection for each query and we build the topic model *U* where $P(w|U)$ is estimated by maximum likelihood estimation to be the number of occurrences of *w* in the topic collection divided by the total number of term occurrences in the topic collection.

To incorporate this topic model into the retrieval framework, we applied the probability mixture model we presented in Chapter 3 to combine the original multinomial query model with the topic model to build a modified query.

$$P(w|Q) = \lambda P_{ML}(w|Q) + (1-\lambda)P_{ML}(w|U) \qquad (4\text{-}1)$$

where $P_{ML}(w|Q)$ is the maximum likelihood estimate of the word *w* in the in the query *Q*, which is estimated by the number of occurrences of *w* in *Q* divided by the number of total term occurrences in *Q*. $P_{\mathrm{ML}}(w|U)$ is the maximum likelihood estimate of the word in the topic model, which is estimated by the number of occurrences of *w* in the topic model *U*. With the co-efficient for Dirichlet smoothing (Zhai and Lafferty, 2001),

$$P(w \mid Q) = \frac{\|Q\|}{\|Q\| + \mu} P_{ML}(w \mid Q) + (1 - \frac{\|Q\|}{\|Q\| + \mu}) P_{ML}(w \mid U) \quad (4\text{-}2)$$

where ||Q|| is the length of the query.

We tried both constant value and Dirichlet co-efficient for *λ*, and chose Dirichlet co-efficient with *μ*=8 based on empirical evidence. Constant value performs better on some of the experiments but its overall performance is not as consistent as Dirichlet co-efficient in our experiments.

After the new query model is built, documents are ranked by the KL divergence between the query model and the document model (Croft and Lafferty, 2003).

In our experiments there are some queries (9 in TREC6, 8 in TREC7 and 15 in TREC8) for which we are unable to find appropriate categories in the Open Directory project, and some queries for which there is insufficient data (too few web links) in the categories we find. We ignore the topic models for these queries to best estimate the potential performance improvement of user context.

4.2.2 Baseline Algorithms

We chose two baseline retrieval models: query likelihood and relevance models. Query likelihood (QL) is a simple retrieval technique and common baseline. Relevance modeling (RM) is an effective query modification technique that fits cleanly into the language modeling framework (Croft and Lafferty, 2003). We chose relevance modeling as a baseline because it is a non-context based query modification approach.

Relevance models modify the queries using the pseudo-feedback approach which relies only on an initial ranking of the documents.

1) Baseline 1: query likelihood model

We use the query likelihood model where each document is scored by the likelihood of its model generating a query *Q*. As we have described in Chapter 3,

$$P(Q \mid D) = \prod_{q \in Q} P(q \mid D) \qquad (4\text{-}3)$$

where *D* is a document model, *Q* is the query and *q* is a query term in *Q*. *P(Q|D)* is the likelihood of a document's model generating the query terms under the assumption that terms are independent given the documents. We construct the document model with Dirichlet smoothing,

$$P(w \mid D) = \frac{\|D\|}{\|D\| + \mu} P_{ML}(w \mid D) + (1 - \frac{\|D\|}{\|D\| + \mu}) P_{ML}(w \mid coll) \qquad (4\text{-}4)$$

where $P_{ML}(w|D)$ is the maximum likelihood estimate of word *w* in the in document *D*, and $P_{ML}(w|coll)$ is maximum likelihood estimate of word *w* in the collection. In our experiments, we used a fixed Dirichlet prior with μ=1000.

2) Baseline 2: relevance model retrieval

The key to relevance model retrieval is estimating the relevance model. Each document is then scored for retrieval by the distance of its model to the relevance model.

Conceptually, the relevance model is a description of an information need or, alternatively, a description of the topic area associated with the information need. From the query modification point of view, the relevance model is the modified query that has a probability (weight) for every term in the vocabulary (Lavrenko, 2001). It is estimated from the query alone, with no training data, as a weighted average of document models, with the estimates of *P(Q|D)* serving as mixing weights:

$$P(w\,|\,Q) = \sum_{D} P(w\,|\,D)P(D\,|\,Q) \qquad (4\text{-}5)$$

Models of the top 50 documents are mixed with Equation (4-5). It is actually a pseudo-relevance feedback process. The document models are linearly smoothed with a constant value λ=0.9,

$$P(w\,|\,D) = \lambda P_{ML}(w\,|\,D) + (1-\lambda)P_{ML}(w\,|\,coll) \qquad (4\text{-}6)$$

where *P(D|Q)* is estimated by Bayes Rule:

$$P(D\,|\,Q) \propto P(Q\,|\,D)P(D) \qquad (4\text{-}7)$$

Since *P*(*Q*) does not depend on *D*, the above proportionality holds. With uniform priors, *P(D)*, the posterior probability *P(D|Q)* amounts to a normalization since we require *P(D|Q)* to sum to 1 over all documents. *P(Q|D)* here is from Equation (4-3).

Then, each document is scored by the KL-divergence of its model to the relevance model. Here the document models are estimated using linear smoothing with a constant λ=0.9 as in Equation (4-6). All the choices of soothing types and parameters are based on experimental evidence.

Relevance modeling provides a formal method for incorporating query modification into the language model framework, and this approach has achieved excellent performance in previous experiments (Lavrenko, 2001).

4.2.3 Experiments

4.2.3.1 System Details

Our experiments were based on TREC ad-hoc retrieval tasks. The data sets include three TREC title query sets: TREC6 (301-350), TREC7 (351-400) and TREC8 (401-450). We indexed the TREC document collections for these data sets using Lemur[iii] – a language modeling and information retrieval toolkit. In all experiments, we used the Krovetz (Krovetz, 1993) stemmer and the default stop word list in Lemur. Retrieval runs are evaluated using trec_eval[iv] provided as part of the TREC ad hoc task.

4.2.3.2 Results

The retrieval performance of manually selected topic models is shown in Table 4.1 with the baseline results. From the table, we can see that, compared to the query likelihood baseline, the manually-built topic model shows some improvement for each query set. Compared to the relevance model baseline, however, the retrieval results with manually-built topic models are not consistent. On the TREC6 collection, there is some improvement, but results are significantly worse on TREC7 and only the same on TREC8. This demonstrates that even under ideal conditions where the topic model is manually chosen, topic models based on the directory service do not perform better than an automatic method that is user independent. Although this result is limited in that the directory service could be improved or these are not real user models, it certainly casts

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

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

doubt on the approach of improving queries through pre-defined topic resources or context-based background.

Table 4.1(a): Comparison of retrieval with the manually-built topic model(MT) with the query likelihood (QL) model and the relevance model (RM) on TREC 6. The evaluation measure is Mean Average Precision. %chg(QL) denotes the percent change in performance over QL, and %chg(RM) denotes the change over RM.

TREC6 queries 301-350 (title)					
	QL	RM	MT	%chg (QL)	%chg (RM)
Rel	4611	4611	4611		
Rret	2358	2171	2423	+2.8	+11.6
0.00	0.6768	0.6184	0.7131	+5.4	+15.3
0.10	0.4648	0.4662	0.5	+7.6	+7.3
0.20	0.3683	0.3662	0.3832	+4.1	+4.6
0.30	0.2821	0.2904	0.3305	+17.2	+13.8
0.40	0.2385	0.2495	0.2716	+13.9	+8.9
0.50	0.1906	0.2101	0.2109	+10.7	+0.38
0.60	0.1528	0.1541	0.1693	+10.8	+9.9
0.70	0.1324	0.1088	0.1161	-12	+6.7
0.80	0.0708	0.0597	0.0643	-9.2	+7.7
0.90	0.0423	0.026	0.0412	-2.6	+58.5
1.00	0.0221	0.0108	0.0221	0	+104.6
Avg	0.2193	0.2133	0.2344	+6.89	+9.9

Table 4.2(b): Comparison of retrieval with the manually-built topic model(MT) with the query likelihood (QL) model and the relevance model (RM) on TREC 7.

TREC7 queries 351-400 (title)					
	QL	RM	MT	%chg (QL)	%chg (RM)
Rel	4674	4674	4674		
Rret	2290	2939	2429	+6.1	-17.4
0.00	0.7221	0.6407	0.7376	+2.2	+15.1
0.10	0.429	0.4861	0.4989	+16.3	+2.6
0.20	0.33	0.3849	0.3613	+9.5	-6.1
0.30	0.2795	0.3316	0.3109	+11.2	-6.2
0.40	0.2177	0.2879	0.2295	+5.4	-20.3
0.50	0.1566	0.2462	0.1681	+7.4	-31.7
0.60	0.1028	0.1949	0.1125	+9.4	-42.3
0.70	0.0683	0.1518	0.081	+8.6	-46.6
0.80	0.0489	0.1099	0.0507	+3.7	-53.9
0.90	0.0384	0.0608	0.0371	-3.4	-39.0
1.00	0.0126	0.0181	0.0131	+4.0	-27.6
Avg	0.1944	0.2515	0.2127	+9.4	-15.4

Table 4.2(c): Comparison of retrieval with the manually-built topic model(MT) with the query likelihood (QL) model and the relevance model (RM) on TREC 8.

TREC8 queries 401-450 (title)					
	QL	RM	MT	%chg (QL)	%chg (RM)
Rel	4728	4728	4728		
Rret	2764	3085	2835	+2.6	-8.1
0.00	0.7552	0.7097	0.7744	+2.5	+9.1
0.10	0.4979	0.5041	0.5321	+6.9	+5.6
0.20	0.3786	0.411	0.3988	+5.3	-3.0
0.30	0.3235	0.3571	0.3285	+1.6	-8.0
0.40	0.2574	0.304	0.2588	+0.5	-14.9
0.50	0.2246	0.2525	0.2182	-2.8	-13.6
0.60	0.1752	0.191	0.1737	-0.9	-9.1
0.70	0.1397	0.1409	0.1227	-11.5	-12.9
0.80	0.1043	0.0925	0.0983	-5.8	+6.3
0.90	0.0897	0.054	0.0841	-6.2	+55.7
1.00	0.0567	0.0247	0.0465	-18.0	+88.26
Avg	0.2497	0.2546	0.2529	+1.28	-0.67

4.2.3.3 Result Analysis

A more in-depth analysis of the results gives some indication why the manually-built topic model does not perform as well overall as the relevance model. We find that the manually-built topic model performs somewhat better on some queries, and much worse on others. Table 4.2 shows the number of queries that benefit (or suffer) from manually-built topic models. Generally, manually-built topic models work better on queries that do not have a clear topic, especially those containing words that have several meanings. On the other hand, relevance models work better on queries that are very specific and clear. For example, the query of “mainstreaming” (Topic 379 in the TREC7 ad hoc retrieval task) refers to a special education field, but after stemming this word has multiple meanings not related to education, which results in the system retrieving many irrelevant documents. In this situation, the relevance model technique for modifying the query does not help since there is too much incorrect information. In contrast to this, the manually selected topic model is based on a human interpretation of the query and therefore is focused on the correct meaning.

Table 4.2: Numbers of queries that MT or RM performs better respectively. MT refers to the queries MT performs better and RM refers the ones that RM is better. EQ refers to same performance. The last column is the difference between column "MT" and column "RM".

	MT	EQ	RM	Diff
TREC6	32	1	17	+15
TREC7	25	0	25	0
TREC8	22	0	28	-6

In the above example, the "ideal" topic model works better. However, there are other queries in which relevance models work better. One such query, "poaching, wildlife preserves" (Topic 407 in the TREC8 ad hoc retrieval task), is very clearly about poaching in wildlife preserves. The initial ranking produces good documents and relevance modeling modifies the query appropriately. Manually-built topic models also have the potential to work well on these types of queries if there are specific categories in the ODP. In this example, the granularity of the category is much broader than documents. The category closest to this example is "wildlife preserves", which misses the important "poaching" part, and the results are worse than relevance models. Even if we have a specific category related to the query, relevance models can still perform better. The content of the specific category in the Open Directory project can be much less than the relevant documents in the whole collection, and the information for query modification that it provides is not as good as the information the collection provides. This is also one of the drawbacks of real user models – usually a user's background is not better than the whole collection, and pseudo-feedback techniques often provide more information than user models.

4.3 Combination

Based on the results and the above analysis, we tried to improve on the relevance model baseline. The manually-built topic models are built in an “ideal” simulation, which theoretically, leaves no room for improvement. But from the analysis in Section 4.2.3.3, we find that the manually-built topic model and the relevance model work well on different kinds of queries, which naturally leads to studying some way of combining the advantages of both models. The most straightforward way is to combine these two models at the model level. Another possibility is to employ a technique that selects different models for different queries.

4.3.1 Model-level Combination

As described in Section 4.2.2, to compute the relevance models we need $P(Q|D)$ from Equation (4-3). This is a basic step for relevance model computation. Since we have the manually-built topic model, which achieves better performance than the query likelihood model, we replace the query likelihood model with the manually-built topic model retrieval in Equation (4-7) and complete the other steps as usual. This is a model-level combination, which is denoted by MCOM in Table 4.3. The average precision is presented in Table 4.3 and the numbers of queries for which the combination model improves over relevance model are shown in Table 4.4.

4.3.2 Query-level Combination: Clarity Score Selection

Query modification showing improvement for only some of the queries is a common problem in information retrieval. When examining the results of any query expansion method over a large number of queries, one always finds that nearly equal numbers of queries are helped and hurt by the given technique (Cronen-Townsend et al., 2004). Cronen-Townsend et al. developed the clarity metric for choosing which queries benefit most from query expansion techniques (Cronen-Townsend and Croft, 2002 ; Cronen-Townsend et al., 2002; Cronen-Townsend et al., 2004). The weighted clarity score is defined by:

$$clarity = \sum_{w \in V} \frac{u(w)P(w|Q)}{\sum_{w' \in V} u(w')P(w'|Q)} \log_2 \frac{P(w|Q)}{P(w|coll)} \qquad (4\text{-}8)$$

where $u(w)$ are the term weights and V is the vocabulary of the collection.

A low clarity score means the query is not very effective and may need modification. In Cronen-Townsend et al.'s original application, the clarity score was used to predict when to use relevance model retrieval to do query modification. According to the analysis in Section 4.2.3.3, "clear" queries achieve better performance with relevance models and "unclear" queries achieve better performance with manually-built topic models. Thus the clarity score is a reasonable selection method to predict when to use the topic model to do query modification.

This is a query-level combination, which is represented by QCOM in Table 4.3. Clarity score selection leads to improvements over relevance models on all three tasks. The improvement is more significant particularly at the top of the ranked list. This is a good sign since a user often goes through only the documents that are provided first and the documents near to the end plays a less significant role when there are a large number of documents retrieved.

Table 4.3: Comparison of the manually-built topic model with two combinations. %chg denotes the percent change in performance over RM (measured in average precision).

TREC6 queries 301-350 (title)				
RM	MCOM	%chg	QCOM	%chg
0.2133	0.1817	-14.8	0.2172	+1.8%
TREC7 queries 351-400 (title)				
RM	MCOM	%chg	QCOM	%chg
0.2515	0.2596	+3.2%	0.2673	+6.3%
TREC8 queries 401-450 (title)				
RM	MCOM	%chg	QCOM	%chg
0.2546	0.2700	+6.0%	0.2573	+1.1%

The numbers of queries that are improved (or not improved) by a combination at the query-level, as compared to relevance models, is reported in Table 4.4. With clarity score selection, more queries benefit from the query-level combination than relevance models on all the three TREC tasks.

4.4 An Automated Categorization Algorithm

Given that manually-built topic models based on ODP categories showed some promise in our previous results, we also investigated an algorithm for automatically selecting a category for a query. In this case, rather than simulating “ideal” topic models or user context models, we are viewing the ODP categories as an alternative to relevance modeling for automatically smoothing the query (i.e. providing topical context).

Table 4.4: Numbers of queries that MCOM or RM performs better. MCOM/QCOM refers to the queries MCOM/QCOM performs better and RM refers the ones that RM is better. EQ refers to same performance. The last column is the difference between column "MCOM" and column "RM".

	MCOM	EQ	RM	Diff
TREC6	19	11	20	-1
TREC7	24	8	18	+6
TREC8	24	14	12	+12
	QCOM	EQ	RM	Diff
TREC6	13	30	7	+6
TREC7	10	35	5	+5
TREC8	9	33	8	+1

4.4.1 Algorithm

The following is the automated categorization algorithm we used for experiments:

1) Treat the whole open directory as a collection and each category as a document. There are descriptions of the sites in each category, which we treat as the document content (the queries are the original title queries as we used in previous experiments). We retrieved the top 5 categories by query likelihood, and only select the categories from these five.
2) Try to find the categories that are close to the query according to the following rules:

(a) All the query terms show up in the category name, which is a directory with the category names at each level, e.g., “Top/Computers/Artificial_Intelligence/Applications”.

(b) The most detailed category name, which is “Applications” in the above example, contains only query terms.

3) If we are unable to find the complete categories covering all query terms in the second step, we will use the categories that either have a query likelihood score, computed in 1), larger than a certain threshold, or contain more than half of the query terms.

All the comparisons are made after stemming and stopping. We built topic models as for the hand-selected categories in Section 4.2, and repeated the experiments on the relevance model baseline with the two combination algorithms.

4.4.2 Results

The retrieval performance with automated categorization is shown in Table 4.5 as AC, and the two combination methods are also employed and included for comparison. The numbers of queries that each model works better on are reported in Table 4.6.

We found there were slight improvements compared to relevance models. We note that the average precision of AC on TREC8 was better than the manual selection model. Automatic selection of topic models is clearly a viable technique for query reformulation and is complementary to the technique of document reformulation such as in the cluster-based document model in (Liu and Croft, 2004).

Table 4.5: Retrieval performance with automated query categorization and two combination algorithms. The evaluation measure is Mean Average Precision. %chg denotes the percent change in performance over RM.

TREC6 queries 301-350 (title)					
RM	AC	MCOM	%chg	QCOM	%chg
0.2133	0.2267	0.1820	-14.7%	0.2162	+1.4%
TREC7 queries 351-400 (title)					
RM	AC	MCOM	%chg	QCOM	%chg
0.2515	0.1959	0.2435	-3.2%	0.2534	+0.8%
TREC8 queries 401-450 (title)					
RM	AC	MCOM	%chg	QCOM	%chg
0.2546	0.2545	0.2661	+4.5%	0.2580	+1.3%

An important result is that the clarity score selection again shows good performance again in Table 4.6, as in Table 4.4. There are always more queries on which QCOM performs better than relevance models on all the three TREC tasks.

4.5 Discussion

As described earlier, this chapter aims at the effectiveness of manually-built topic models, which can be viewed as an "ideal" usage of available topic resources and also an "ideal" user context model. So we are interested in the following two questions: 1) can these topic models, which represent hand-crafted topic resource and user context, improve retrieval performance, and 2) how much performance gain can we get from them. Our experimental results provide some indications of the answers.

Table 4.6: Numbers of queries that AC or RM performs better, with the comparisons after MCOM and QCOM.

	AC	EQ	RM	Diff
TREC6	31	0	19	+12
TREC7	23	0	27	-4
TREC8	22	0	28	-4
	MCOM	EQ	RM	Diff
TREC6	14	12	24	+10
TREC7	19	8	23	+4
TREC8	18	19	13	-5
	QCOM	EQ	RM	Diff
TREC6	13	27	10	+3
TREC7	11	32	7	+4
TREC8	10	32	6	+4

4.5.1 Can Topic Resource/User Context Improve IR?

In our experiments, the manually-built topic model showed some improvement over the query likelihood baseline, but the model itself does not show a consistent or significant improvement over the relevance model baseline. As an “ideal” manual topic/user context model, the topic model estimates an empirical upper bound on the benefits of hand-crafted topic resource/user context modeling when it is used to modify a query. Besides, the ideal user models are much more focused than real user models would be. Even given this advantage, this model is inconsistent and is not better overall, compared to relevance modeling, which does not need additional user information. This reflects the difficulty in improving retrieval with user context.

There is some improvement in the results after combination for the manually selected models, and the advantage of combination was evident even in a simple automatically selected topic model. In Table 4.4 and Table 4.6, clarity scores did some useful prediction since the combination approach performs better for the majority of queries.

So, the answer to the first question is that topic resource/user context in the form of topic models is unlikely to have significant benefits based on our experiments with the ODP categories.

4.5.2 How Much Gain Can We Get?

From our experiments, the empirical upper bounds we estimated are not dramatically higher than the relevance model retrieval. Some queries perform well, but many suffer in the user context approaches. In the results after query-level combination, which are relatively consistent, less than 7% improvement is found on average precision, dependent on the TREC tasks. This shows the room for improvement is very limited. The individual upper bound for each query varies a lot. For some queries, the manually-built topic model performs very well. The performance improvement of the example query "mainstreaming" we mentioned in Section 4.2.3.3 is shown in Table 4.7.

Table 4.7: Comparison of QL, RM and MT performance on query "mainstreaming".

	QL	RM	MT
Rel Rret	16 6	16 5	16 14
0.00	0.2	0.0625	1
0.10	0.0292	0.0211	1
0.20	0.0292	0.008	0.5556
0.30	0.0116	0.008	0.5556
0.40	0	0	0.1633
0.50	0	0	0.1633
0.60	0	0	0.0694
0.70	0	0	0.0205
0.80	0	0	0.0186
0.90	0	0	0
1.00	0	0	0
Avg	0.0186	0.0066	0.2756

4.6 Summary

We built topic models manually based on a topic resource, which is also hand-crafted, to estimate the potential improvements those hand-crafted topic resources could bring to IR in the language modeling framework, and the result also reflects the potential improvement of user context by viewing the topic models as simulated "ideal" user context models. After experimenting with queries from several TREC ad-hoc retrieval tasks, we found that the manually-built topic models provided little benefit for the overall document retrieval performance compared to relevance models, an automatic non-extra resource based query modification model. In some cases, the topic model improves the results, but in other cases relevance models are more effective, and the overall results did not show that manually-built topic models perform better on these tasks.

Based on the observation that manually-built topic models and relevance models benefit different queries, we investigated a combination approach. Our experiments confirmed that an automatic selection algorithm using the clarity score improves retrieval results.

We also established that topic models based on the ODP categories can be a potentially useful source of information for retrieval. In particular, we showed that query-level combination with the automatically selected categories by PMM improves retrieval performance.

CHAPTER 5

TERM ASSOCIATING MODELS

5.1 Introduction

Modeling term associations automatically is important to Information Retrieval (IR) systems. As we described in Chapter 1, ranking algorithms solely based on matching the literal words that are present in queries and documents will fail to retrieve much relevant information. For this reason, term associations, which are also called "term relationships" or "word similarity" in the literature, have been introduced to add new terms to the query/document representations that are related to the original terms. Besides term associations, which usually refers to associations between two single terms (term-term association), there can also be associations between two groups of terms (term group association). In this chapter we discuss term-term association and in Chapter 6 we will discuss term group association.

As we discussed in Chapter 2, term co-occurrence is more often used in IR than grammatical analysis to capture semantic associations, and simple term-term association has significant advantages over term group association considering the offline efficiency of document reformulation. Cao et al (2005)'s work sheds light on the effectiveness of integrating term associations into the language modeling framework. On the other hand, the term independence assumption ("bag of words") of the unigram language model is well known to be inappropriate for natural language. This has led many language model researchers to study term associations. The window-based approach used by Cao et al. (2005), however, always requires an appropriate setting for the window size, and the improvements using only the automatic model are not as impressive.

As a summary, we are interested in an automatic term associating method based on term co-occurrence in the language modeling framework, especially for dealing with document reformulation. Term associating models have been studied for decades. Some integration processes of term associations are carried out with language models and some associating processes like the window-based co-occurrence model are probabilistic methods. But none of the associating processes have been performed within the language modeling framework. In this chapter we study the traditional term co-occurrence based automatic term associating methods in the document reformulation task, and propose a new and simple method, which is based on the language modeling approach and thus fits within this framework naturally, to model term associations for retrieval operations.

5.2 Traditional Term Associating Methods

The history of traditional term associating methods has been briefly discussed in Chapter 2. In this section we describe the details of the term associating methods that we will experiment within our framework.

5.2.1 Similarity Coefficient

A variety of similarity coefficients have been developed and applied to measure term associations in IR environments, such as the cosine similarity, weighted and unweighted Tamimoto (Sparck Jones, 1971), etc. The coefficient used in Qiu & Frei

(1993)'s concept-based query expansion is one example. They built a term-document matrix and computed the similarity between any two terms as follows;

$$SIM(t_i, t_j) = \sum_{k=1}^{n} d_{ik} \cdot d_{jk} \quad , \qquad \mathbf{1})$$

$$d_{ik} = \frac{(0.5 + 0.5 \times \frac{ff(d_k, t_i)}{\max ff(t_i)} iif(d_k)}{\sqrt{\sum_{j=1}^{n} ((0.5 + 0.5 \times \frac{ff(d_j, t_i)}{\max ff(t_i)} iif(d_j))^2}} \quad , \qquad 2)$$

where $ff(d_k, t_i)$ is the frequency of term t_i in document d_k, $iff(d_k)$=log(m/| d_k|), m is the number of terms in the collections and |d_k| is the number of different terms in document d_k. max $ff(t_i)$ is the maximum frequency of term t_i in all documents. The d_{ik}'s and d_{jk}'s signify feature weights of the indexing features (documents). Then, the similarity between a term and a query is defined as the weighted sum of the similarity values between the term and individual terms in the query. To expand a query, terms with the highest similarity to the query are added and the weight of each added term takes its similarity value with the original query. Significant improvements in retrieval effectiveness were reported in their paper (Qiu and Frei, 1993).

Although many techniques in this area have been tested and some interesting results were obtained, most of the techniques have been used to do query expansion. Few studies on document modeling with term similarity coefficients have been conducted.

5.2.2 Co-occurrence in Windows

Another important group of term association measures estimates the conditional probability of a term given another term. Van Rijsbergen (1979) and Cao et al. (2005)

compute the conditional probability using co-occurrence samples. To compute the conditional probability of two terms by their co-occurrence in a window is a practical method for both its simplicity and effectiveness. A non-overlapping window is applied to measure the co-occurrence in (Cao et al., 2005) and a sliding-window method (Hyperspace Analogue to Language, HAL) is described in (Burgess et al., 1998). A typical computation of the co-occurrence probability (the strength of term association) is as follows:

$$P(t_j \mid t_i) = f(t_i, t_j) / \sum_k f(t_i, t_k) \qquad (5\text{-}1)$$

where $f(t_i, t_j)$ is the frequency of co-occurrences of t_i and t_j.

5.2.2.1 Non-overlapping window

A non-overlapping window is often used to measure the co-occurrence of two terms. In this window-based method, two words are considered as co-occurring once when the distance between them is less than the window size. For instance, Xu and Croft (1996) developed a metric used for query expansion based on the non-overlapping window method and achieved excellent performance (Xu, 1997; Xu and Croft, 1996); Cao et al. applied non-overlapping windows in document modeling in combination with WordNet and obtained significant improvements on two TREC collections(Cao et al., 2005).

5.2.2.2 Sliding window

In addition to setting a threshold to judge the co-occurrence of terms as in the non-overlapping window method, the distance between two words are also taken into account in some term-association models, such as in (Burgess et al., 1998; Gao et al., 2001; Lund and Burgess, 1996; Bai et al., 2005). Sliding window method is one of the examples, which is also called HAL Space (Hyperspace Analogue to Language) (Burgess et al., 1998; Lund and Burgess, 1996). By moving a window across the text, an accumulated co-occurrence matrix for all terms is produced. Compared to the non-overlapping window method, the sliding window method takes accumulated co-occurrence in all possible non-overlapping windows and in this way, the strength of association between two words is inversely proportional to their distance. Some interesting results with the sliding window method are obtained in previous works, including query expansion tasks in the language modeling framework (Bai et al., 2005; Burgess et al., 1998; Lund and Burgess, 1996). However, its effectiveness on document modeling tasks is still unknown.

In both the non-overlapping window and the sliding window methods, the size of the window is a parameter that needs to be determined.

5.3 Modeling Term Associations by Joint Probability

5.3.1 Term Associating Models

Previous research has shown the effectiveness of modeling and integrating term associations into information retrieval processes. Especially, constructing term-term associations and integrating them into document models is an attractive way considering both its online efficiency and large-collection feasibility. Also, the language modeling framework provides and motivates new directions of the construction and integration process of term associations. In this section, we present an approach in the language modeling framework to estimating the conditional probability of terms by joint probability through Bayesian rule, and the joint probability will be computed by unigram document models.

To get a sense of the association or closeness between two terms, *w* and *t*, we consider *P*(*w*|*t*), which is the probability of observing *w* when *t* is given. By Bayesian rule, we have

$$P(w \mid t) = P(wt) / P(t) \qquad (5\text{-}2)$$

To estimate the join probability of observing the word *w* and the term *t*, instead of counting co-occurrence samples in windows, we assume that *w* and *t* are identical and independent samples from a unigram document model *D*. Then the total probability of observing *w* together with *t* is:

$$P(wt) = \sum_{D \in \prod} P(D_{orig}) P(wt \mid D_{orig}) = \sum_{D \in \prod} P(D_{orig}) P(w \mid D_{orig}) P(t \mid D_{orig}) \qquad (5\text{-}3)$$

where ∏ represents some finite universe of unigram document models, and D_{orig} represents the original unigram document model which was estimated with maximum likelihood estimation. We choose to use uniform priors $P(D_{orig})$ and limit the universe ∏ to the collection we test on. Then, with Equation (5-2) and Equation (5-3),

$$P(w \mid t) = \frac{\sum_{D} P(w \mid D_{orig}) P(t \mid D_{orig})}{\sum_{w} \sum_{D} P(w \mid D_{orig}) P(t \mid D_{orig})} \qquad (5\text{-}4)$$

Thus, for each term *t*, there is a list of words *w* with the probability *P*(*w*|*t*) representing the association of *w* and *t*. We can view this probability as the association/closeness between *w* and *t*.

5.3.2 Document Language Models with Term Associations

To integrate the association information into document models, we apply the PMM framework and TBS framework we presented in Chapter 3. With PMM, the computation will be the same as computing the word distribution in documents through the probabilistic association measure (Equation (5-5)), and then combining it with the original term model by linear combination (Equation (5-7)).

$$P(w \mid D_{\exp}) = \sum_{t \in D_{orig}} P(w \mid t) P(t \mid D_{\exp}) \qquad (5\text{-}5)$$

where D_{exp} represents the document model for expansion, which is the topical document representation, and we assume $P(t|D_{exp})=P(t|D_{orig})$. Equation (5-5) is similar to the retrieval methodology using translation models proposed by Berger and Lafferty to incorporate term associations into document language models (Berger and Lafferty, 1999). With the translation model, the document model becomes

$$P_{TR}(w \mid D) = \sum_{t} tr(w \mid t) P(t \mid D) \qquad (5\text{-}6)$$

where $tr(w|t)$ is the translation model for mapping a document term t to an arbitrary term w. The translation probability $tr(w|t)$ describes the degree of link between a term w and the document term t. If we set $tr(w|t)$ to be $P(w|t)$, then Eqn (5-5) and Eqn (5-6) will be same.

The final PMM document model would be

$$\begin{aligned} P(w \mid D) &= \lambda P(w \mid D_{orig}) + (1-\lambda) P(w \mid D_{\exp}) \\ &= \lambda \left(\frac{N_d}{N_d+\mu} P_{ML}(w \mid D) + (1 - \frac{N_d}{N_d+\mu}) P_{ML}(w \mid coll)\right) \\ &\quad + (1-\lambda) \sum_{t \in D_{orig}} P(w \mid t) P(t \mid D_{orig}) \end{aligned} \qquad (5\text{-}7)$$

where λ is the integration co-efficient. This is the only parameter to our model, and is also one of the parameters to the other models we compare to in Section 5.4.

In this paper we try several association measures to model $P(w|t)$ in Equation (5-7), including the similarity co-efficient, the non-overlapping window method, the sliding window method, and the joint probability method we propose. In the similarity co-efficient method, we normalize its co-efficient to be consistent with the probabilistic application as following:

$$P(t_j \mid t_i) = SIM(t_i, t_j) / \sum_k SIM(t_i, t_k) \qquad (5\text{-}8)$$

5.4 Experiments and Results

5.4.1 Data

We conduct experiments on five data sets taken from TREC: the Associated Press Newswire (AP) 1988-90 with queries 51-150, Wall Street Journal (WSJ) 1987-92 with queries 51-100 and 151-200, Financial Times (FT) 1991-94 with queries 301-400, San Jose Mercury News (SJMN) 1991 with queries 51-150, and LA Times (LA) with queries 301-400. Queries are taken from the “title” field of TREC topics. Queries that have no relevant documents in the judged pool for a specific collection have been removed from the query set for that collection. Statistics of the collections and query sets are given in Table 5.1.

These five collections, including the query sets and relevance judgments, were the same as used by the experiments in the following chapters (Chapter 6 and 7) so that we can compare the results later.

Table 5.1: Statistics of data sets.

Collection	Contents	# of dos	Size	Queries	# of Queries with Relevant Docs
AP	Associated Press newswire 1988-90	242,918	0.73Gb	TREC topics 51-150 (title only)	99
FT	Financial Times 1991-94	210,158	0.56Gb	TREC topics 301-400 (title only)	95
SJMN	San Jose Mercury News 1991	90,257	0,29Gb	TREC topics 51-150 (title only)	94
LA	LA Times	131,896	0.48Gb	TREC topics 301-400 (title only)	98
WSJ	Wall Street Journal 1987-92	173,252	0.51Gb	TREC topics 51-100 & 151-200 (title only)	100

5.4.2 Parameters

There are several parameters that need to be decided in our experiments. For the retrieval experiments, the proportion of the term association part in the PMM framework must be specified (λ in Equation (5-7)). For the similarity measure, the window sizes need to be determined. We use the AP collection as our training collection to estimate the parameters. The WSJ, FT, SJMN, and LA collections are used for testing whether the parameters optimized on AP can be used consistently on other collections. At the current stage of our work, the parameters are selected through exhaustive search or manual hill-climbing search. All parameter values are tuned based on mean average precision (MAP).

The retrieval results by tuning the window sizes in the non-overlapping window and the sliding window methods we have are shown as follows.

Table 5.2: Retrieval results on AP with different non-overlapping window size (W).

W	10	30	50
MAP	0.2310	0.2381	0.2376

Table 5.3: Retrieval results on AP with different sliding window size (W).

W	10	30	50	70
MAP	0.2295	0.2361	0.2374	0.2372

5.4.3 Complexity

The complexity of the term associating model based on joint probability is $O(\sum_{d}(N_{w\_d})^2)$, where N_{w_d} is the number of unique words in document *d*. The complexity of window-based methods is linear with the window size *W* and the number of word tokens N_t. If we compare these two numbers only, then we can consider $W{*}N_{t\_d}$ and $(N_{w\_d})^2$ for each document, where N_{t_d} is a number of tokens in document *d*, which is actually the document length. With a reasonable setting of *W* and a typical TREC collection as AP, $W{*}N_{t\_d}$ is smaller than $(N_{w\_d})^2$. But these two complexity numbers are based on different data structures: for the joint probability computation, we only need word index; but for the window-based computation, we also need token sequence. In our implementation, the time complexity for window-based method are much more than $W{*}\ N_t$ due to the limitation of memory space.

5.4.3 Experimental Results

In all experiments, both the queries and documents are stemmed, and stopwords are removed.

5.4.3.1 Other Term-Associating Methods

We test the effectiveness of some traditional term-term associating methods that we discussed in Section 5.2 with PMM document models, and present the retrieval results in Table 5.4.

Similarity co-efficient: With the parameter setting λ=0.8, which was obtained by training on the AP collection, we run experiments with the similarity co-efficient based document models (SCDM) on other collections. Some improvements, including significant improvements on one of the five collections, are achieved over query likelihood retrieval by integrating the similarity co-efficient into document models.

Non-overlapping window: With λ=0.7 and window size W=30, which were obtained by training on the AP collection, we run experiments with the non-overlapping window based document models (NWDM) on other collections. Significant improvements on two of the five collections are obtained over query likelihood retrieval.

Sliding window: Retrieval results of the document models based on the sliding window method, with λ=0.6 and W=50, are shown in Table 5.4. Significant improvements on two of the five collections over the query likelihood retrieval are achieved. Table 5.4 also shows that the sliding window performs better than the non-overlapping window, which was adopted in (Cao et al., 2005; Cao et al., 2007) as an automatic term associating method to be integrated into language document models.

Table 5.4: Comparison of query likelihood retrieval (QL) and retrieval with document models based on similarity coefficient (SCDM), non-overlapping window method (NWDM), or sliding window method (SWDM). The evaluation measure is MAP. %chg denotes the percentage change in average precision. Stars indicate statistically significant differences with a 95% confidence according to the Wilcoxon test.

Collection	QL	SCDM	%chg over QL	NWDM	%chg over QL	SWDM	%chg over QL	%chg over NWDM
AP	0.2161	0.232	+7.62*	0.2381	+10.15*	0.2375	+9.88*	-0.25
FT	0.2558	0.2652	+3.68	0.2640	+3.22	0.2690	+5.14	+1.86*
SJMN	0.1985	0.2068	+4.18	0.2118	+6.67*	0.2142	+7.86*	+1.12
LA	0.2290	0.2305	+0.62	0.2362	+3.12	0.2485	+8.48	+5.20*
WSJ	0.2908	0.2866	-1.44*	0.2827	-2.79	0.2905	-0.10	+2.76*

5.4.3.2 Term Associations by joint probability

We test PMM document models based on the term associating method by joint probability (JPDM) that we present, and show the retrieval results in Table 5.5. λ=0.6 for these experiments, and we process only the top 400 related terms of each term. On four of the five collections JPDM retrieval achieves significant improvements over query likelihood retrieval. On the WSJ collection, no improvements are achieved with λ=0.6, and then we especially tuned λ for it and obtained improvement with λ=0.2 as shown in the last line of Table 5.5.

Table 5.5: Comparison of query likelihood retrieval (QL) and retrievals with JPDM and JPDM-ap.

Collection	QL	JPDM	%chg over QL	JPDM-ap	%chg over QL	%chg over JPDM	JPDM-all
AP	0.2161	0.2400	+11.03*	0.2400	+11.03*	0	0.2422
FT	0.2558	0.2754	+7.66*	0.2636	+3.05	-4.28	0.2842
SJMN	0.1985	0.2180	+9.80*	0.2139	+7.74*	-1.88	0.2186
LA	0.2290	0.2516	+9.85*	0.2426	+5.91	-3.59	0.2547
WSJ	0.2908	0.2870	-1.32	0.2884	-0.83	+0.49	0.2910
WSJ (λ=0.2)	0.2908	0.2971	+2.15	N/A	N/A	N/A	N/A

In previous experiments, we build term associations for each collection respectively. To test the easy applicability of the term associating method we present, we also run experiments with the term associations constructed only from the AP collection (JPDM-ap), or all of the five collections (JPDM-all). Results of JPDM-ap and JPDM-all are presented in Table 5.5.

JPDM-all achieves the best performance among JPDM, JPDM-all and JPDM-ap. This shows that more training data lead to higher performance, because more data can imply more knowledge about the term associations. At the same time, term associations trained only on the AP collection are also effective on other collections. So, the term associations built by joint probability do not have to be trained on the specific collection of experiments.

5.4.3.3 PMM Vs. TBS

Table 5.6 compares retrieval results of PMM document model and TBS document model with the term associating method we presented based on joint probability. Further comparison and analysis of these two frameworks will be done in Section 6.3.4.3.

Table 5.6: Comparison of retrieval with PMM and TBS document model based on term similarity measure trained on the AP collection.

Collection	ORIG	JPAP-PMM	JPAP-TBS
AP	0.2161	0.2400	0.2377
FT	0.2558	0.2754	0.2758
SJMN	0.1985	0.2180	0.2185
LA	0.2290	0.2586	0.2508
WSJ	0.2908	0.2870	0.2923

5.5 Summary

We have proposed a probabilistic term associating model in the language modeling framework, which measures term associations through their joint probability, and a document retrieval model that integrates term associations into document models through PMM or TBS. We did experiments and compared the model we proposed with other popular term associating methods on ad-hoc retrieval tasks.

The experimental results showed that modeling term associations through joint probability was effective in the language modeling framework. Document models that include term associations outperformed the query likelihood model, and term associations constructed by joint probability achieved better performance than other term associating models, such as window co-occurrence methods, in the language modeling framework. Comparing the two window co-occurrence methods, the sliding window method performs better than the non-overlapping window method on the retrieval tasks. We also showed that term associations trained on other collections were effective in our model, and more training data leads to better performance.

CHAPTER 6

LATENT MIXTURE TOPIC MODELING

6.1 Introduction

Representing the content of text documents is a critical part of any approach to information retrieval (IR) and many other research fields. Typically, documents are represented as a “bag of words”, meaning that the words are assumed to occur independently. To capture important relationships between words, researchers have proposed approaches that represent documents as mixtures of latent “topics” in large text collections. As we discussed in Chapter 1, the difference of these latent mixture models and term associating models is the type of data that they define associations on. Term associating models model associations between one single term and another. Associations are only dependent on the vocabulary entry of the term. With latent mixture models, associations of text are not only dependent on the term itself as the term associating model describes, but also related with its context; thus latent mixture models have been used to model term group association by representing text as a mixture of latent topics (such as in the cluster model, where document, instead of term, associations are considered).

The well-known Latent Semantic Indexing (LSI) technique, which was introduced in 1990 (Deerwester et al, 1990), is a term group associating method. More recently, Hoffman (1999) described the probabilistic Latent Semantic Indexing (pLSI) technique (for the details of LSI and pLSI, please refer to Chapter 2). This approach uses a latent variable model that represents documents as mixtures of topics. Although

Hoffman showed that pLSI outperformed LSI in a vector space model framework, the data sets used were small and not representative of modern IR environments. Specifically, the collections in these experiments only contained a few thousand document abstracts.

As we mentioned in Chapter 1, the new latent mixture topic model, Latent Dirichlet Allocation (LDA, Blei et al, 2003), has recently become one of the most popular probabilistic text modeling techniques in machine learning and has inspired a series of research papers (e.g., Girolami and Kaban, 2005; Teh et al, 2004). LDA has been shown to be effective in some text-related tasks such as document classification, but the feasibility and effectiveness of using LDA in IR tasks remains mostly unknown. Possessing fully generative semantics, LDA potentially overcomes the drawbacks of previous topic models such as pLSI (Hoffman, 1999). Language modeling (Croft and Lafferty, 2003; Ponte and Croft, 1998) is also a generative model, motivating us to examine LDA-based document representations in the language modeling framework.

The LDA approach will be compared with an approach that builds topic models using document clusters, known in the machine learning literature as the mixture of unigrams model (McCallum, 1999). As detailed in Section 2.3.2.1, Liu and Croft (2004) showed that document clustering can improve retrieval effectiveness in the language modeling framework. Retrieval based on cluster models (referred to here as cluster-based retrieval) performed consistently well across several TREC collections, and significant improvements over document-based retrieval models were reported. In the language modeling framework, the cluster-based topic models were used to smooth the probabilities in the document model (Liu and Croft, 2004). As a much simpler topic model, the mixture of unigrams model generates a whole document from one topic under the assumption that each document is related to exactly one topic. This assumption may, however, be too simple to effectively model a large collection of documents. In contrast, LDA models a document as a mixture of multiple topics.

Given the potential advantages of LDA as a generative model of documents, and the encouraging results with topic models in previous work, we carried out a detailed evaluation of the effectiveness of LDA-based retrieval in large collections. Azzopardi et al. (2004) also discussed the applications of LDA models and reported inconclusive results on several small collections. In this chapter, we integrate LDA into our probability mixture modeling and term modeling frameworks to build new document representation for IR, evaluate it on TREC collections, and discuss efficiency issues. We also compare its retrieval performance with the term associating model we presented in Chapter 5 as a comparison between term group associations and term-term associations.

6.2 Latent Dirichlet Allocation

As we described in Chapter 2, the pLSI model has a problem with inappropriate generative semantics. Blei et al. (2003) introduced a new, semantically consistent topic model, Latent Dirichlet Allocation (LDA), which immediately attracted a considerable interest from the statistical machine learning and natural language processing communities. The basic generative process of LDA closely resembles pLSI. In pLSI, the topic mixture is conditioned on each document. In LDA, the topic mixture is drawn from a conjugate Dirichlet prior that remains the same for all documents. The process of generating a corpus is as follows (we consider the smoothed LDA here):

1) Pick a multinomial distribution ϕ_z for each topic *z* from a Dirichlet distribution with parameter β ;
2) For each document *d*, pick a multinomial distribution θ_d from a Dirichlet distribution with parameter α ,
3) Pick a topic $z \in \{1 \ldots K\}$ from a multinomial distribution with parameter θ_d ,
4) Pick a word *w* from a multinomial distribution with parameter ϕ_z .

Thus, the likelihood of generating a corpus is:

$$\begin{aligned} &P(Doc_1,...,Doc_N \mid \alpha,\beta) \\ &= \iint \prod_{z=1}^{K} P(\phi_z \mid \beta) \prod_{d=1}^{N} P(\theta_d \mid \alpha) (\prod_{i=1}^{N_d} \sum_{z_i=1}^{K} P(z_i \mid \theta) P(w_i \mid z,\phi)) d\theta d\phi \end{aligned} \tag{6-1}$$

The LDA model is represented as a probabilistic graphical model in Figure 6.1.

Compared to the pLSI model, LDA possesses fully consistent generative semantics by treating the topic mixture distribution as a *k*-parameter hidden random variable rather than a large set of individual parameters which are explicitly linked to the training set; thus LDA overcomes the overfitting problem and the problem of generating new documents in pLSI.

Compared to the cluster model, LDA allows a document to contain a mixture of topics, relaxing the assumption made in the cluster model that each document is generated from only one topic. This assumption may be too limited to effectively model a large collection of documents; in contrast, the LDA model allows a document to exhibit multiple topics to different degrees, thus being more flexible.

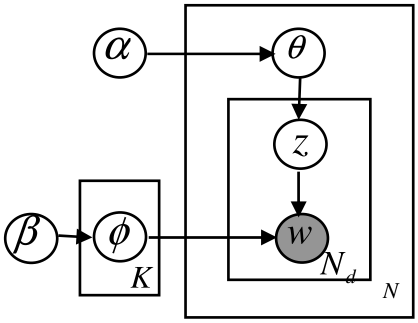


Figure 6.1: Graphical model representation of LDA. *T* is the number of topics; *N* is the number of documents; and N_d is the word tokens in document.

The LDA model is very complex and cannot be solved by exact inference. There are a few approximate inference techniques available in the literature: variational methods (Blei et al, 2003), expectation propagation (Griffiths and Steyvers, 2004) and Gibbs sampling (Geman and Geman, 1984; Griffiths and Steyvers, 2004). We use Gibbs sampling and draw the topic assignment z_i iteratively for each token *i* according to the following conditional probability distribution:

$$P(z_i = j \mid z_{-i}, \alpha, \beta, Doc_1, \ldots, Doc_N) \propto \frac{n_{-i,j}^{(w_i)} + \beta_{w_i}}{\sum_{v=1}^{V} (n_{-i,j}^{(v)} + \beta_v)} \times \frac{n_{-i,j}^{(d_i)} + \alpha_{z_i}}{\sum_{t=1}^{T} (n_{-i,t}^{(d_i)} + \alpha_t)} \qquad (6\text{-}2)$$

where $n_{-i,j}^{(w_i)}$ is the number of instances of word w_i assigned to topic z=j, not including the current token, α and β are hyper-parameters that determine how heavily this empirical distribution is smoothed, and can be chosen to give the desired resolution in the resulting distribution, $n_{-i,j}^{(d_i)}$ is the number of words in document d_i (the document that token *i* belongs to) assigned to topic z=j, not including the current token. Thus $\sum_{v=1}^{V} n_{-i,j}^{(v)}$ is the total number of words assigned to topic z=j; and $\sum_{t=1}^{T} n_{-i,t}^{(d_i)}$ is the total number of words in document d, not including the current one (Griffiths and Steyvers, 2004).

6.3 Experiments

6.3.1 Data

We conducted experiments on the same TREC data sets that we have described in Chapter 5: the Associated Press Newswire (AP) 1988-90 with queries 51-150, Wall Street Journal (WSJ) 1987-92 with queries 51-100 and 151-200, Financial Times (FT) 1991-94 with queries 301-400, San Jose Mercury News (SJMN) 1991 with queries 51-150, and LA Times (LA) with queries 301-400. Statistics of the collections and query sets have been presented in Table 5.1.

These five collections, including the query sets and relevance judgments, are the same as used by Liu and Croft (2004) in order to compare retrieval effectiveness based on different topic models. The only difference between the two experimental settings is that we left out the Federal Register (FR) collection for two reasons: (1) The query set of this collection contains only 21 queries with relevant documents, (the query sets of other collections contain at least 94 valid queries); (2) In these 21 valid queries there are six that have only one relevant document in the collection and thus may cause biased results.

6.3.2 Parameters

There are several parameters that need to be determined in our experiments. For the retrieval experiments of the probability mixture model (PMM), the mixture weight λ must be specified. For the LDA estimation, the number of topics must be specified; the number of iterations and the number of Markov chains also need to be carefully tuned due to its influence on performance and running time. We use the AP collection as our training collection to estimate the parameters. The WSJ, FT, SJMN, and LA collections are used for testing whether the parameters optimized on AP can be used consistently on other collections. At the current stage of our work, the parameters are selected through exhaustive search or manual hill-climbing search. All parameter values are tuned based on average precision since retrieval is our final task. The parameter selection process, including the training set selection, also follows Liu and Croft (2004) to make the results comparable.

We use symmetric Dirichlet priors in the LDA estimation with $\alpha = 50 / K$ (K is the number of topics) and β =0.01, which are common settings in the literature. Our experience shows that retrieval results are not very sensitive to the values of these parameters.

6.3.2.1 Parameters in LDA Estimation

Document models consisting of mixtures of topics, like pLSI and LDA, have previously been tested mostly on small collections due to their relatively long running time. It will be shown in Section 6.3.3 that the iteration number in LDA estimation plays an important role in its complexity. Generally, more iterations means that the Markov chain reaches equilibrium with higher probability, and after a certain number of iterations (burn-in period) the invariant distribution of the Markov chain is equivalent to the true distribution. So it would be ideal if we could take samples right after the Markov chain reach equilibrium. However, in practice, convergence detection of Markov chains is still an open research question. That is, no realistic method can be applied on the large IR collections to determine the convergence of the chain.

Researchers in the area of topic modeling tend to use a large number of iterations to guarantee convergence. However, in IR tasks it is almost impossible to run a very large number of iterations due to the size of the data set. Besides, a finely tuned topic model does not naturally mean good retrieval performance. Instead, a less accurate distribution of topics may be good enough for IR purposes. Furthermore, we have λ and μ in our model to adjust the influence of the LDA model. For example, if the LDA estimation is coarse, we may reduce the smoothing weight and let the LDA estimation share a part of smoothing.

In order to get a good iteration number that is effective for IR applications, we use the AP collection for training and maximizing the average precision score as the optimization criterion since it is our final evaluation metric. We try different iteration numbers, and also do experiments with different numbers of Markov chains, each of which is initialized with a different random number, to see how many chains are needed for our purposes. The results are presented in Figure 6.2 and Figure 6.3, respectively. After 50 iterations and with more than 3 Markov chains, performance is quite stable, so we use these values in the final retrieval experiments. The running time of each iteration with large topic numbers can be expensive; 30 iterations and 2 chains are a good trade off between accuracy and running time, and these values are used in the parameter-selecting experiments, especially when selecting a suitable number of topics.

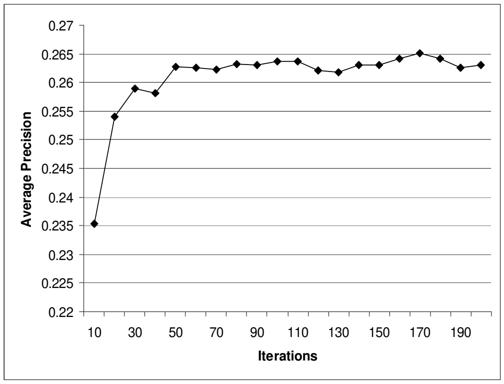


Figure 6.2: Retrieval results (in average precision) on AP with different number of iterations. *K*=400; λ=0.7; 1 Markov chain.

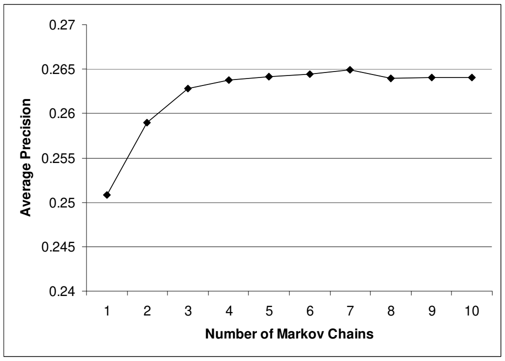


Figure 6.3: Retrieval results (in average precision) on AP with different number of Markov chains. *K*=400; λ=0.7; 30 iterations.

Selecting the right number of topics is also an important problem in topic modeling. Nonparametric models like the Chinese Restaurant Process (Blei et al, 2004; Teh et al, 2004) are not practical to use for large data sets to automatically decide the number of topics. A range of 50 to 300 topics is typically used in the topic modeling literature. 50 topics are often used for small collections and 300 for relatively large collections, which are still much smaller than the IR collections we use. It is well known that larger data sets may need more topics in general, and it is confirmed here by our experiments with different values of K (100, 200, …) on the AP collection. K=800 gives the best average precision, as shown in Table 6.1. This number is much less than the corresponding optimal *K* value (2000) in the cluster model (Liu and Croft, 2004). As we explained in Chapter 2, in the cluster model, one document can be based on one topic, and in the LDA model, the mixture of topics for each document is more powerful and expressive; thus a smaller number of topics is used. Empirically, even with more parsimonious parameter settings like *K*=400, 30 iterations, 2 Markov chains, statistically significant improvements can also be achieved on most of the collections.

Table 6.1: Retrieval results (in MAP) on AP with different number of topics (K).

K	50	100	200	300	400	500
Average precision	0.2397	0.2431	0.2520	0.2579	0.2590	0.2557
K	600	700	800	900	1000	1500
Average precision	0.2578	0.2609	0.2621	0.2613	0.2585	0.2579

6.3.2.2 Parameters in Retrieval Model

For the probability mixture model (PMM), in order to select a suitable value of λ, we use a similar procedure as above on the AP collection and find 0.7 to be the best value in our search. From the experiments on the testing collections, we also find that λ=0.7 is the best value or almost the best value for other collections. We set the Dirichlet smoothing parameter μ =1000 since the best results are consistently obtained with this setting.

For the TBS document model, there is no other parameter than the Dirichlet smoothing prior μ, which is fixed to be 500 for TBS as we described in Chapter 3.

6.3.3 Complexity

Complexity is often a big concern for topic models. Even the simple cluster model suffers from potentially high computational costs. Liu and Croft (2004) used a three-pass K-means algorithm primarily motivated by its efficiency. They showed that the running time for each pass/iteration grows linearly with the number of documents (*N*) and the number of classes (*K*), i.e., *O(KN).* We adopt Gibbs sampling (Geman and Geman, 1984) to estimate the LDA model. Roughly speaking, the complexity of each iteration of the Gibbs sampling for LDA is also linear with the number of topics/clusters and the number of documents, which is also *O(KN).* Due to the large sizes of document collections, we give a more detailed analysis.

The time-consuming part of the Gibbs sampling in the LDA model is linear with *I*, *K* and $N * \overline{N}_t$, where *I* is the number of iterations, *K* is the number of topics, *N* is the number of documents and $\overline{N}_t$ is the average number of tokens in one document. In K-means clustering algorithm, the computation is linear with *I*, *N*, and $K * \overline{N}_w$, where *I* is the number of passes/iterations, and $\overline{N}_w$ is the average number of unique terms in one cluster. (We use the average numbers, $\overline{N}_t$ and $\overline{N}_w$, instead of the corresponding sums to make the following comparison easier.)

To compare the running time of these two algorithms we compare realistic values of these items.

(1) *K*: The selected number of topics (*K*) in the LDA model is generally less than the selected number of topics/clusters in the cluster model because in the LDA model topics can be mixed to represent one document, but in the cluster model one document can based on only one topic.

(2) *I*: The number of iterations (*I*) will probably have a larger value in the LDA algorithm. In Liu and Croft (2004), the number of iterations for K-means is 3. Such a small *I* does not work well for Gibbs sampling in the LDA model. The selection of *I* is very important to make sure that the Markov chains reach equilibrium. In Section 6.3, we will show that *I* = 30 ~ 50 is reasonable in our experiments.

(3) $\overline{N}_t$ vs. $\overline{N}_w$: It is hard to make an assertion about the relationship of these two items, especially since $\overline{N}_w$ is highly related to the selection of *K*. While in our experiments and settings, the number of unique terms in a cluster is often larger than $\overline{N}_t$ since one cluster often contains quite many documents.

The above comparison shows that the efficiency of the two algorithms is similar. In experiments, we also find that the difference in running times between LDA and K-means is trivial. Based on our experience based on using several IR collections, these two algorithms are comparable in computational costs and there is no clear evidence showing that one algorithm is obviously more efficient.

6.3.4 Experimental Results

In all experiments, both the queries and documents are stemmed, and stopwords are removed.

6.3.4.1 Retrieval Experiments with PMM

The LDA model has a new representation for a document based on topics. After we get the posterior estimates of θ and ϕ, we can calculate the probability of a word in a document as following,

$$P_{lda}(w \mid d, \hat{\theta}, \hat{\phi}) = \sum_{z=1}^{K} P(w \mid z, \hat{\phi}) P(z \mid \hat{\theta}, d)$$

(6-3)

where $\hat{\theta}$ and $\hat{\phi}$ are the posterior estimates of θ and ϕ respectively. We use Gibbs sampling and the approximation of $\hat{\theta}$ and $\hat{\phi}$ can be obtained directly. From a Gibbs sample, we use we use $(n_j^{(w)} + \beta_w) / \sum_{v=1}^{V} (n_j^{(v)} + \beta_v)$ to approximate $\hat{\phi}$ and $(n_j^{(d)} + \alpha_z) / \sum_{t=1}^{T} (n_t^{(d)} + \alpha_t)$ to approximate $\hat{\theta}$ after a certain number of iterations (burn-in period) being accomplished, where $n_j^{(w)}$ is the number of instances of word *w* assigned to topic *z=j*, $n_j^{(d)}$ is the number of words in document *d* assigned to topic *z=j* (Griffiths and Steyvers, 2004).

Thus the LDA-based PMM document model will be

$$\begin{aligned} P(w|D) = \lambda(&\frac{N_d}{N_d+\mu} P_{ML}(w|D) + (1-\frac{N_d}{N_d+\mu}) P_{ML}(w|coll)) \\ &+ (1-\lambda)(\sum_{t=1}^{K} \frac{n_j^{(w)} + \beta_w}{\sum_{v=1}^{V} (n_j^{(v)} + \beta_v)} \times \frac{n_j^{(d)} + \alpha_z}{\sum_{t=1}^{T} (n_t^{(d_i)} + \alpha_t)}) \end{aligned} \qquad (6\text{-}4)$$

The retrieval results on the AP collection are presented in Table 6.2, with comparisons to the result of query likelihood retrieval (QL) and cluster-based retrieval (CBDM). Statistically significant improvements of PMM with LDA topics (LDA-PMM) over both QL and CBDM are observed at many recall levels, with 21.64% and 13.97% improvement in mean average precision respectively.

Table 6.2: Comparison of query likelihood retrieval (QL), cluster-based retrieval (CBDM) and retrieval with the LDA-based probability mixture model (LDA-PMM). The evaluation measure is average precision. AP data set. Stars indicate statistically significant differences in performance with a 95% confidence according to the Wilcoxon test.

	QL	CBDM	LDA-PMM	%chg over QL	%chg over CBDM
Rel.	21819	21819	21819		
Rel. Retr.	10130	10751	12064	+10.09*	+12.21*
0.00	0.6422	0.6485	0.6795	+5.8*	+4.8*
0.10	0.4339	0.4517	0.4844	+11.6*	+7.2*
0.20	0.3477	0.3713	0.4131	+18.8*	+11.2*
0.30	0.2977	0.317	0.3661	+23.0*	+15.5*
0.40	0.2454	0.2668	0.311	+26.8*	+16.6*
0.50	0.2081	0.2274	0.2666	+28.1*	+17.2*
0.60	0.1696	0.1794	0.2245	+32.4*	+25.1*
0.70	0.1298	0.1444	0.1665	+28.3*	+15.3*
0.80	0.0865	0.1002	0.118	+36.5*	+17.8*
0.90	0.0480	0.0571	0.0694	+44.7	+21.6
1.00	0.0220	0.0201	0.0187	-15.1	-6.8
Avg	0.2179	0.2326	0.2651	+21.64*	+13.97*

With the parameter setting λ=0.7, 50 iterations and 3 Markov chains, we run experiments on other collections and present results in Table 6.3. We compare the results with CBDM, and the results of the query likelihood model are also listed as a reference. On all five collections, retrieval with LDA-based PMM achieves improvements over both of query likelihood retrieval and cluster-based retrieval, and four of the improvements are significant (over CBDM). Considering that CBDM has already obtained significant improvements over the query likelihood model (and Okapistyle weighting, Liu and Croft, 2006) on all of these collections, and is therefore a high baseline, the significant performance improvements from LBDM are very encouraging.

Table 6.3: Comparison of cluster-based retrieval (CBDM) and retrieval with the LDA-based probability mixture model (LDA-PMM). The evaluation measure is average precision. %chg denotes the percentage change in performance (measured in average precision) of LDA-PMM over QL and CBDM. Stars indicate statistically significant differences in performance between LDA-PMM and QL/CBDM with a 95% confidence according to the Wilcoxon test.

Collection	QL	CBDM	LDA-PMM	%chg over QL	%chg over CBDM
AP	0.2179	0.2326	0.2651	+21.64*	+13.97*
FT	0.2589	0.2713	0.2807	+7.54*	+3.46*
SJMN	0.2032	0.2171	0.2307	+13.57*	+6.26*
LA	0.2468	0.2590	0.2666	+8.02[v]	+2.93
WSJ	0.2958	0.2984	0.3253	+9.97*	+9.01*

Unlike the basic document representation, the LDA-based document model is not limited to only the literal words in a document, but instead describes a document with many other related highly probable words from the topics of this document. Like the query expansion technique, this reformulated representation of document improves the retrieval performance as well. For example, for the query “buyout leverage”, the document “AP900403-0219”, which talks about “Farley Unit Defaults On Pepperell Buyout Loan”, is a relevant document. However, this document focuses on the “buyout” part, and does not contain the exact query term “leverage”, which makes this

[v] This improvement is significant according to t-test, and almost significant (with a 93% confidence) according to the Wilcoxon test.

document rank very low. By the LDA-based document model, this document is closely related with two topics that have strong connections with the term “leverage”: the *economic* topic is strongly represented with this document because the document contains quite many representative terms of this topic, such as “million”, “company”, “bankruptcy”; the *money market* topic is closely connected to “bond”, which is also a very frequent word in this document. By these words and their strongly associated topics, the connection between the document and the term “leverage” is built up. In this way, the document is ranked higher with the LDA-based document model. The multiple topics in one document help to represent clearer association between the topics and the terms than a single topic, as one topic is very limiting to model long documents that indeed talk about a variety of issues.

Table 6.4 shows an example of the topics associated with a document. The document is actually “AP900403-0219” that we discussed above. We list the top 5 topics for this document and the top 10 words in each topic with corresponding probabilities.

6.3.4.2 Comparison and Combination with Relevance Models

In Table 6.5 we compare the retrieval results of the LDA-PMM with the relevance model (RM), which incorporates pseudo-feedback information and is known for excellent performance (Lavrenko and Croft, 2001). On some collections, the results of the two models are quite close. RM uses pseudo-feedback information and thus needs *online* processing, i.e., it effectively does an extra search for each query, which makes it less efficient in reacting to users’ inputs. As an *offline*-processing model that does not do any extra processing on queries, the LDA-based PMM retrieval model performance is quite impressive. In another words, we estimate the LDA model offline only once, and then the probability mixture model can process real-time queries much more efficient than RM with similar performance.

Table 6.4: An example of topical document model by LDA.

Topic 1:	Topic 2:	Topic 3:
company 0.072359	s 0.076526	bond 0.121074
share 0.048106	steven 0.060858	junk 0.035927
stock 0.045680	mill 0.057960	market 0.030698
million 0.022542	great 0.049431	investor 0.030509
shareholder 0.019582	georgia 0.028320	invest 0.028007
percent 0.019255	pacific 0.028085	high 0.024533
offer 0.018970	textile 0.025799	issue 0.021888
corp 0.014770	paper 0.024221	debt 0.020750
takeover 0.014506	farley 0.023396	finance 0.017940
buy 0.013682	point 0.022312	secure 0.017833
Topic 4:	Topic 5:	Document "AP900403-0219"
debt 0.103697	bankruptcy 0.083795	Topic 1 0.225
loan 0.099891	file 0.048261	Topic 2 0.202
bank 0.087656	creditor 0.043383	Topic 3 0.111
pay 0.034809	company 0.043100	Topic 4 0.088
billion 0.034053	million 0.041746	Topic 5 0.046
interest 0.029302	reorganize 0.033888	
borrow 0.027971	chapter 0.032615	
lend 0.025020	plan 0.031464	
finance 0.022393	court 0.029343	
credit 0.020054	protect 0.028221	

Table 6.5: Comparison of the relevance models (RM) and the LDA-based probability mixture models (LDA-PMM). The evaluation measure is average precision. %diff indicates the percentage change of LDA-PMM over RM.

Collection	QL	LDA-PMM	RM	%diff
AP	0.2179	0.2651	0.2745	-3.42
FT	0.2589	0.2807	0.2835	-0.99
SJMN	0.2032	0.2307	0.2633	-12.38
LA	0.2468	0.2666	0.2614	+0.20
WSJ	0.2958	0.3253	0.3422	-4.94

Figure 6.4 compare LDA-PMM with RM at query level. Each point in the figure represents the percentage of improvements of RM over LDA-PMM on one query. There are 54, 42, 44, 51, 52 queries that RM performs better than LDA-PMM on the AP, FT, LA, SJMN, and WSJ collections respectively. Figure 6.4 shows that these two models benefit different queries.

We also combine the relevance model and LDA-PMM to do retrieval. In this case, the retrieval results using LDA-PMM are used as the pseudo-feedback for the relevance model. Results are shown in Table 6.6, and results of the query likelihood model are also listed as a reference. Moderate improvements are obtained, which is much better than the very small improvements reported in Liu and Croft (2004) for the combination of RM and CBDM.

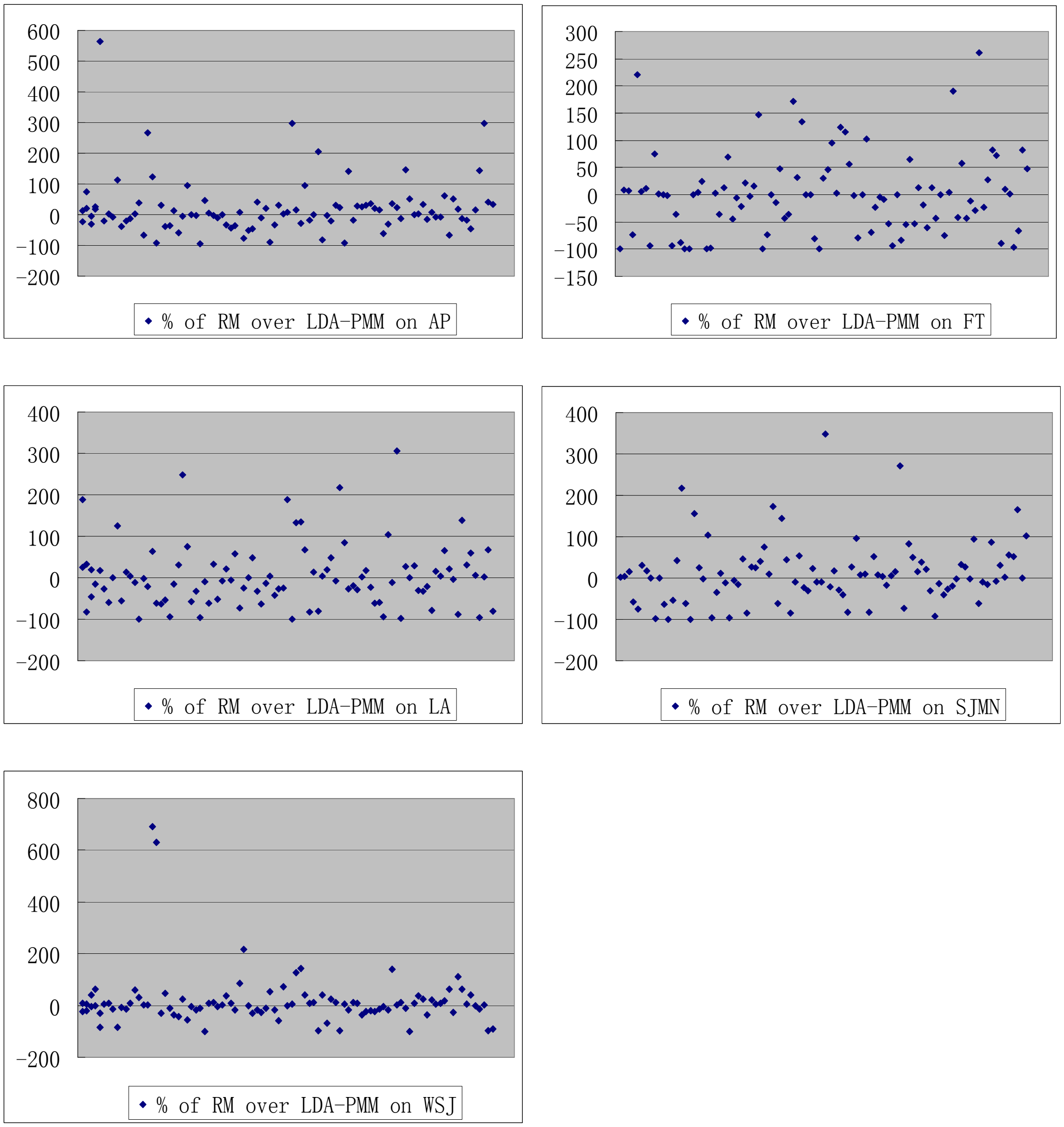


Figure 6.4: Comparison of RM and LDA–PMM at query level.

We also combine the relevance model and LDA-PMM to do retrieval. In this case, the retrieval results using LDA-PMM are used as the pseudo-feedback for the relevance model. Results are shown in Table 6.6, and results of the query likelihood model are also listed as a reference. Moderate improvements are obtained, which is much better than the very small improvements reported in Liu and Croft (2004) for the combination of RM and CBDM.

Table 6.6: Comparison of the relevance model (RM) and the combination of RM and the LDA-based probability mixture model (RM+LDA-PMM). The evaluation measure is average precision. %chg denotes the percentage change in performance (measured in average precision) of RM+LDA-PMM over RM. Stars indicate statistically significant differences in performance between RM+LDA-PMM and RM with a 95% confidence according to the Wilcoxon test.

Collection	QL[vi]	RM	RM+LDA-PMM	%chg
AP	0.2161	0.2758	0.2869	+4.00
FT	0.2558	0.2889	0.2907	+0.62
SJMN	0.1985	0.2547	0.2603	+2.22
LA	0.2290	0.2509	0.2715	+8.21
WSJ	0.2908	0.3405	0.3606	+5.91*

6.3.4.3 PMM Vs. TBS

In TBS, we use $\sum_{t=1}^{T}[(n_j^{(w)}+\beta_w)/\sum_{v=1}^{V}(n_j^{(v)}+\beta_v)]\times[(n_j^{(d)}+\alpha_z-P(z=t\,|\,w_d)D_w)/\sum_{t=1}^{T}(n_t^{(d)}+\alpha_t)]$ to approximate $\sum_{t\neq w}Z_t(w)/N_d$ in Equation (3-9) after a certain number of iterations (burn-in period) being accomplished, where $P(z=t\,|\,w_d)$ is estimated by $[(n_j^{(w)}+\beta_w)/\sum_{v=1}^{V}(n_j^{(v)}+\beta_v)]\ \times[(n_j^{(d)}+\alpha_z)/\sum_{t=1}^{T}(n_t^{(d)}+\alpha_t)]$ and D_w represents the frequency of word *w* in document *D*.

[vi] The QL&RM baseline in Table 6.5 is slightly different with Table 6.4 because in the experiments of Table 6.4, in order to compare with the results in Liu and Croft (2004), we directly load their index into our system and then run the experiments on their index to get nearly identical results.

Table 6.7 shows the retrieval results of PMM and TBS with LDA. In Table 5.6 at Section 5.4.3.3, we compare results of these two frameworks with the term-term associating model we presented in Chapter 5. From the results in these two tables, we can see that the LDA-based retrieval performance of the term model with back-off smoothing (TBS) is quite close to the probability mixture model (PMM), although TBS does not introduce any new parameters. Also, TBS performs consistently over collections and topic models.

Table 6.7: Comparison of query likelihood (QL), cluster-based retrieval (CBDM), and retrieval with the probability mixture model (PMM) and the term model with back-off smoothing (TBS). The evaluation measure is average precision.

Collection	QL	CBDM	LDA-PMM	LDA-TBS
AP	0.2179	0.2326	0.2651	0.2655
FT	0.2589	0.2713	0.2807	0.2739
SJMN	0.2032	0.2171	0.2307	0.2317
LA	0.2468	0.2590	0.2666	0.2668
WSJ	0.2958	0.2984	0.3253	0.3218

The improvement on the AP collection in Table 6.3 and Table 6.7 is relatively larger than on the other collections. Although we tune parameters on the AP collection, further parameter adjustment does not improve the performance on the other collections. Compared to the relevance model results in Table 6.5, we conjecture that it is due to the property of the documents and the queries that the improvement on the AP collection is larger than on the other collections.

6.4 Comparison with Term-term Association

We have categorized topic models into three types and discussed the difference of manually-built topic models (type I) with automatic methods (type II & III) in previous chapters. In this section we will compare the two types of automatic topic modeling methods: type II (term-term association) and type III (term group association).

6.4.1 Efficiency

We use the term associating model based on joint probability as a representative of term-term associating models for both of its performance and effectiveness on IR. Its complexity is $O(\sum_d (N_{w\_d})^2)$, where N_{w_d} is the number of unique words in document *d*. As we explained in Section 6.3.3, the time-consuming part of the Gibbs sampling in the LDA model is linear with *I*, *K* and $N * \overline{N}_t$, where *I* is the number of iterations, *K* is the number of topics, *N* is the number of documents and $\overline{N}_t$ is the average number of tokens in one document.

To compare these two complexities, we decompose the comparison to be based on one document. For the term associating model on one document *d*, the computation time is linear with $(N_{w\_d})^2$; for the LDA estimation, it is linear with *K*, *I* and N_{t_d}, where N_{t_d} is the number of tokens in the document d. N_{t_d} is larger or equal than N_{w_d}. *K* is 800 in our setting, and we know that the average of N_{t_d} is much smaller than 800. So the term associating model is much more efficient than the LDA model.

6.4.2 Effectiveness

Table 6.8 shows the comparison of JPDM-all and LDA-based PMM document models (LDA-PMM). LDA-PMM achieves better performance than the term association model.

Table 6.8: Comparison of query likelihood retrieval (QL) and retrievals with LBDM, JPDM, and JPDM-all.

Collection	QL	LDA-PMM	JPDM-all	%chg over QL	%chg over JPDM	%chg over LBDM
AP	0.2161	0.2629	0.2422	+12.05*	+0.92*	-7.91*
FT	0.2558	0.2795	0.2842	+11.10	+3.20	+1.68
SJMN	0.1985	0.2279	0.2186	+10.10*	+0.27*	-4.06*
LA	0.2290	0.2563	0.2547	+11.21*	+1.24	-0.63
WSJ	0.2908	0.3244	0.2910	+0.07	+1.41*	-10.30*

6.4.3 Discussion

Term group association is more effective on IR tasks, which shows the advantages of group association. However, based on our analysis and confirmed by our experiments, the term association modeling is much faster than the LDA model estimation. Also, we have shown in Chapter 5 that it is very easy and effective to apply the term associations trained on other collections, which is impossible for the LDA model training.

The assumption behind term-term associating models is that the single term is the basic unit of language and one term has only one meaning. This is not a perfect assumption for natural language. In term group associations the context of the term is also involved. But the assumption based on single terms catches the character of language that people tend to use one term to indicate same/similar/related meanings, and simplifies the modeling process.

Therefore, term association and term group association are two topic modeling methods to meet different application requests: for collections in reasonable size, we would suggest LDA based document models for document representation; for collections too small or too large that are hard to run topic models, it may be better to apply term associations that have already been learned from other data sets.

CHAPTER 7

OTHER TOPIC MODELS

Latent mixture modeling has been shown to be an effective topic modeling technique and many recent topic models have been developed based on it. Motivated by the success of LDA-based document models in IR, we studied several new topic models within IR framework in this chapter and present the preliminary IR results of these models.

7.1 N-gram Topic Model

7.3.1 Introduction to TNG

There are mainly two types of word dependencies being studied and shown to be effective to IR: 1) topical (semantic) dependency, which is also called long-distance dependency. Two words are considered dependent when their meanings are related and they co-occur often, such as “fruit” and “apple”; 2) phrase dependency, also called short-distance dependency. As reported in literature, retrieval performance can be boosted if the similarity between a user query and a document is calculated by common phrases instead of common words (Fagan, 1989; Evans et al., 1991; Strzalkowski, 1995; Mitra et al. 1997). Most research on phrases in information retrieval has employed an independent collocation discovery module. In this way, a phrase can be indexed exactly as an ordinary word.

Topic models in our study target semantic dependencies, but phrase dependencies are also critical to capturing the meaning of text. Word order and phrases are not only important for syntax, but also important for lexical meaning. A collocation is a phrase with meaning beyond the individual words. For example, the word “pie” in “apple pie” may be generated from a *fruit* topic with a high probability, but the word “pie” in “pie chart” is probably not. We studied several topic models with short-distance dependency in the IR framework, including the bigram topic model (BTM, Wallach, 2006), the LDA collocation model (LDACOL, Steyvers and Griffiths, 2005) and the topical *n*-gram model (TNG, Wang and McCallum, 2005). The graphical model representation of these three models is in Figure 7.1 and notation is listed in Table 7.1. Their retrieval effectiveness is evaluated and compared.

Table 7.1: Notation for Figure 7.1.

Symbol	Description
T	number of topics
D	number of documents
W	number of unique words
N_d	number of word tokens in document *d*
$z_i^{(d)}$	the topic associated with the i^{th} token in the document *d*
$x_i^{(d)}$	The bigram status between the $(i\text{-}1)^{th}$ token and i^{th} token in the document *d*
$w_i^{(d)}$	the i^{th} token in document *d*
θ^d	The multinomial (discrete) distribution of topics w.r.t. the document *d*
ϕ_z	The multinomial (discrete) unigram distribution of words w.r.t. topic *z*
ψ_v	In Figure 7.1(b), the binomial (Bernoulli) distribution of status variables w.r.t. previous word *v*
ψ_{zv}	In Figure 7.1(c), the binomial (Bernoulli) distribution of status variables w.r.t. previous topic *z*/word *v*

σ_{zv} In Figure 7.1(a) and (c), the multinomial (discrete) bigram distribution of words w.r.t. topic *z*/word *v*

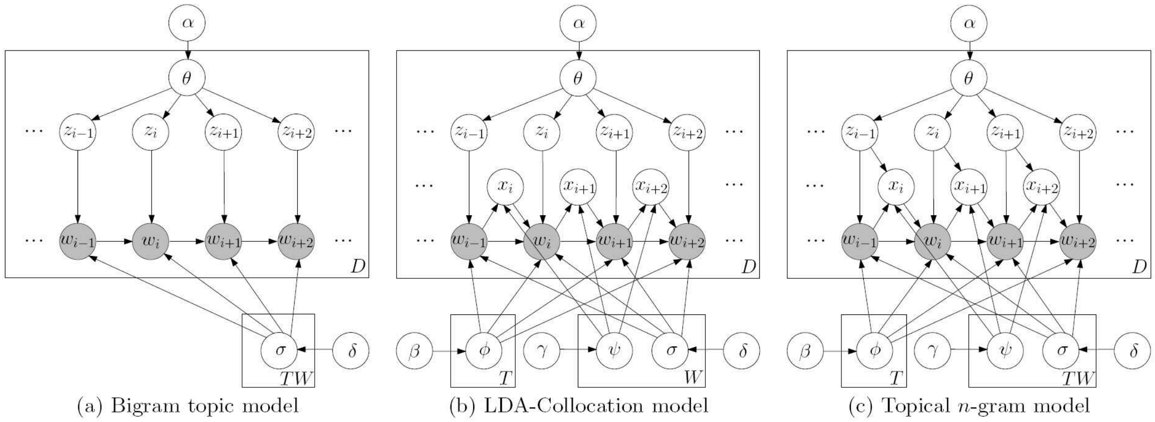
σ_v In Figure 7.1(b), the multinomial (discrete) bigram distribution of words w.r.t. previous word *v*

α Dirichlet prior of θ

β Dirichlet prior of ϕ

γ Dirichlet prior of ψ

δ Dirichlet prior of σ



(a) Bigram topic model (b) LDA-Collocation model (c) Topical *n*-gram model

Figure 7.1: Three *n*-gram based topic models.

For simplicity, all the models discussed in this section make the 1st order Markov assumption, that is, they are actually bigram models. However, all the models have the ability to “model” higher order *n*-grams ($n > 2$) by concatenating consecutive bigrams.

In the bigram topic model, a word will be generated from a multinomial distribution specific to the previous word and the current topic; in the LDA Collocation model, bigram status ($x_i^{(d)}$) is introduced to denote if a bigram can be formed with the previous token, but it does not have topic as the second term of a bigram is generated from a distribution conditioned on the previous word only; in the topical n-gram model, bigram status is also dependent on the topic. One of the key contributions of TNG is to make it possible to decide whether to form a bigram for the same two consecutive word tokens depending on their nearby context (i.e., co-occurrences). For example, the phrase "white house" carries a special meaning in a document about politics, but in the context of a document about real estate, it may not be a collocation.

The topical *n*-gram model automatically and simultaneously takes cares of both semantic co-occurrences and phrases. Also, it does not need a separate module for phrase discovery, and everything can be seamlessly integrated into the language modeling framework. In this section, we illustrate the difference in IR experiments of applying the TNG and LDA models, and compare the IR performance of all three models with short-distance dependency on a TREC collection. This work was also published in (Wang et al., 2007).

7.3.2 IR Experiments

The SJMN dataset, taken from TREC with standard queries 51-150 that are taken from the "title" field of TREC topics, covers materials from San Jose Mercury News in 1991. All text is downcased and only alphabetic characters are kept. Stop words in both the queries and documents are removed. If any two consecutive tokens were originally separated by a stopword, no bigram is allowed to be formed. In total, the SJMN dataset we use contains 90,257 documents. 6 queries that have no relevant documents have been removed from the query set.

The number of topics are set to be 100 for all models, and symmetric priors *α*=1, *β*= 0.01, *γ*=0.1, and *δ*=0.01. Here, we aim to compare the models instead of the results.

7.3.2.1 Topical N-gram Models in IR

To calculate the query likelihood from the TNG model within the language modeling framework, we need to sum over the topic variable and bigram status variable for each token in the query token sequence. Given the posterior estimates $\widehat{\theta}, \widehat{\phi}, \widehat{\psi}$ and $\widehat{\sigma}$, the query likelihood of query *Q* given document *d* can be calculated as (a dummy *q₀* is assumed at the beginning of every query, for the convenience of mathematical presentation)

$$P_{TNG}(Q \mid d) = \prod_{i=1}^{\|Q\|} P_{TNG}(q_i \mid q_{i-1}, d) \qquad (7\text{-}1)$$

where

$$P_{TNG}(q_i \mid q_{i-1}, d) = \sum_{z_i=1}^{T} (P(x_i = 0 \mid \hat{\psi}_{q_{i-1}}) P(q_i \mid \widehat{\phi}_{z_i}) + P(x_i = 1 \mid \hat{\psi}_{q_{i-1}}) P(q_i \mid \widehat{\sigma}_{z_i q_{i-1}})) P(z_i \mid \widehat{\theta}^{(d)}) \qquad (7\text{-}2)$$

and

$$P(x_i \mid \hat{\psi}_{q_{i-1}}) = \sum_{z_{i-1}=1}^{T} (P(x_i \mid \hat{\psi}_{z_{i-1} q_{i-1}}) P(z_{i-1} \mid \widehat{\theta}^{(d)}) \qquad (7\text{-}3)$$

Due to stopping and punctuation removal, we may simply set $P(x_i = 0 \mid \hat{\psi}_{q_{i-1}}) = 1$ and $P(x_i = 1 \mid \hat{\psi}_{q_{i-1}}) = 0$ at corresponding positions in a query. Under first order Markov assumption, $P(Q \mid d) = P(q_1 \mid d) \prod_{i=2}^{\|Q\|} P(q_i \mid q_{i-1}, d)$, and with the probability mixture model

$$P(q_i \mid q_{i-1}, d) = \lambda P_{orig}(q_i \mid d) + (1 - \lambda) P_{TNG}(q_i \mid q_{i-1}, d) \qquad (7\text{-}4)$$

The results from our experiments with TNG did not show significantly better performance than LDA. But TNG achieves better results on some queries. To illustrate the difference of TNG and LDA in IR applications, we select a few of the 100 queries that clearly contain phrase(s), and another few of them that do not contain phrase due to stopping and punctuation removal, on which we compare the IR performance (MAP) as shown in Table 7.2.[vii]. These preliminary results show the possibility of further improvements with query-level model selection of TNG.

Table 7.2: Comparison of LDA and TNG on TREC retrieval performance (MAP) of eight queries. The top four queries obviously contain phrase(s), and thus TNG achieves much better performance. On the other hand, the bottom four queries do not contain common phrase(s) after preprocessing (stopping and punctuation removal). Surprisingly, TNG still outperforms LDA on some of these queries.

No.	Query	LDA	TNG	Change
053	Leveraged Buyouts	0.2141	0.3665	71.20%
097	Fiber Optics Applications	0.1376	0.2321	68.64%
108	Japanese Protectionist Measures	0.1163	0.1686	44.94%
111	Nuclear Proliferation	0.2353	0.4952	110.48%
064	Hostage-Taking	0.4265	0.4458	4.52%
125	Anti-smoking Actions by Government	0.3118	0.4535	45.47%
145	Influence of the ``Pro-Israel Lobby"	0.2900	0.2753	-5.07%
148	Conflict in the Horn of Africa	0.1990	0.2788	40.12%
	All queries	0.1789	0.1752	-2.06

[vii] The results in Table 7.2 and Table 7.3 are preliminary results for model comparison. They are not globally comparable, such as to the results reported in Chapter 6, because of different experimental settings.

7.3.2.2 Comparison of BTM, LDACOL and TNG on TREC Ad-hoc Retrieval

In this section, we compare the IR effectiveness of the three *n*-gram based topic models on the SJMN dataset, as shown in Table 7.3. For a fair comparison, the weighting factor λ are independently chosen to get the best performance from each model. The Dirichlet priors are reset to adjust the proportion of *n*-gram part. This is consistent with our experience on applying bigram language models in IR, which also requires to be tuned to be a very small proportion in order to improve retrieval results.

Table 7.3: Comparison of the bigram topic model (λ =0.7), LDA collocation model (λ=0.9) and the topical n-gram Model (λ=0.8) on TREC retrieval performance (MAP). * indicates statistically significant differences in performance with 95% confidence according to the Wilcoxon test. TNG performs significantly better than other two models overall.

No.	Query	TNG	BTM	Change	LDACOL	Change
061	Israeli Role in Iran-Contra Affair	0.1635	0.1104	-32.47%	0.1316	-19.49%
110	Black Resistance Against the South African Government	0.4940	0.3948	-20.08%	0.4883	-1.16%
117	Capacity of the U.S. Cellular Telephone Network	0.2801	0.3059	9.21%	0.1999	-28.65%
130	Jewish Emigration and U.S.-USSR Relations	0.2087	0.1746	-16.33%	0.1765	-15.45%
138	Iranian Support for Lebanese Hostage-takers	0.4398	0.4429	0.69%	0.3528	-19.80%
150	U.S. Political Campaign Financing	0.2672	0.2323	-13.08%	0.2688	0.59%
	All Queries	0.2122	0.1996	-5.94%*	0.2107	-0.73%*

Under the Wilcoxon test with 95% confidence, TNG significantly outperforms BTM and LDACOL on this standard retrieval task.

It is interesting to see that different models are good at quite different queries. For some queries (such as No. 117 and No. 138), TNG and BTM perform similarly, and better than LDACOL, and for some other queries (such as No. 110 and No. 150), TNG and LDACOL perform similarly, and better than BTM. There are also queries (such as No. 061 and No. 130) for which TNG performs better than both BTM and LDACOL. We believe that they are clear empirical evidence that the TNG model is more effective on IR tasks than BTM and LDACOL.

We analyze the performance of the TNG model for query No. 061, as an example. As we inspect the phrase ``Iran-Contra" contained in the query, we find that it has been primarily assigned to two topics (politics and economy) in TNG. This has increased the bigram likelihood of some documents emphasizing the relevant topic (such as "SJMN91-06263203"), thus helps promote these documents to higher ranks. As a special case of TNG, LDACOL is unable to capture this and leads to inferior performance.

It is true that for certain queries (such as No. 069 and No. 146), TNG performs worse than BTM and LDACOL, but we notice that all models perform badly on these queries and the behaviors are more possibly due to randomness.

Table 7.4 shows an example of the topics associated with a document. The document is "SJMN91-06005068". We list the top 4 topics for this document, with the top 10 words and the top 10 phrases in each topic.

Table 7.4: An example of the topics associated with a document by TNG.

Topic 1:	Topic 2:	Topic 3:	Topic 4:
years	news	vice	people
died	newspaper	president	time
born	magazine	sold	years
wife	mercury	director	good
worked	media	fiscal	back
retired	editor	named	make
home	york	chief	lot
son	paper	executive	year
served	article	vp	day
age	page	officer	things
san jose	mercury news	vice president	san jose
heart attack	robert maxwell	net income	years ago
war ii	quiz answers	san jose	united states
heart failure	canary islands	chief executive	los angeles
york city	executive editor	chief operating	san francisco
golden weddinganniversary	national enquirer	general manager	bay area
long illness	larry jinks	executive officer	santa clara
cremated remains	kevin maxwell	chief financial	palo alto
golden wedding	rupert murdoch	executive vice	mercury news
san carlos	bob ingle	op income	high school

7.2 Pachinko Allocation Model (PAM)

7.2.1 Introduction to PAM

LDA captures correlations among words by forming topics, but it does not explicitly model correlations among topics (Li and McCallum, 2006). However, topic correlations are common in real-world text data, e.g., the *fruit* topic is more likely to co-occur with the *baking food* topic than the *money market* topic. To address this problem, Li and McCallum (2006) presented the pachinko allocation model (PAM), in which the concept of topics are extended to be distributions not only over words, but also over other topics. They described a four-level PAM as Figure 7.2 shows.

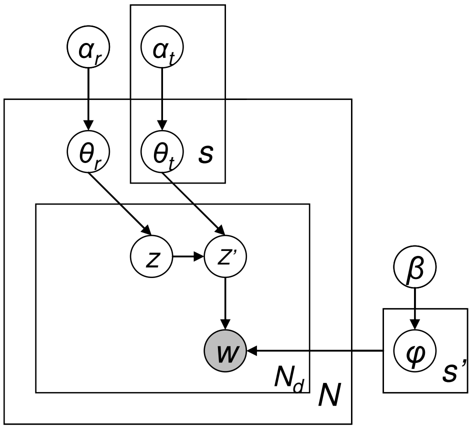


Figure 7.2: Graphical model representation of PAM. *N* is the number of documents; *N_d* is the word tokens in document; *s* is the number of topics in the second level and *s'* is the number of topics in the third level.

7.2.2 IR Experiments

We compute the document model with the leaf topics in PAM and construct new document models with probability mixture modeling as we have done for LDA. Because PAM is more expensive than LDA, we experiment on two subsets of the AP collection. One contains 1,913 documents and the other contains 20,000 documents (the AP collection contains 242,918 documents in total). For the small data set, we set 50 topics for LDA, 50 sub-topics and 10 super-topics for PAM. For the large data set, we set 200 topics for LDA, 200 sub-topics and 10 super-topics for PAM. Other parameter settings are the same as described in Chapter 6.

The retrieval results by number of iterations are shown in Figure 7.3, with 3 Markov chains.

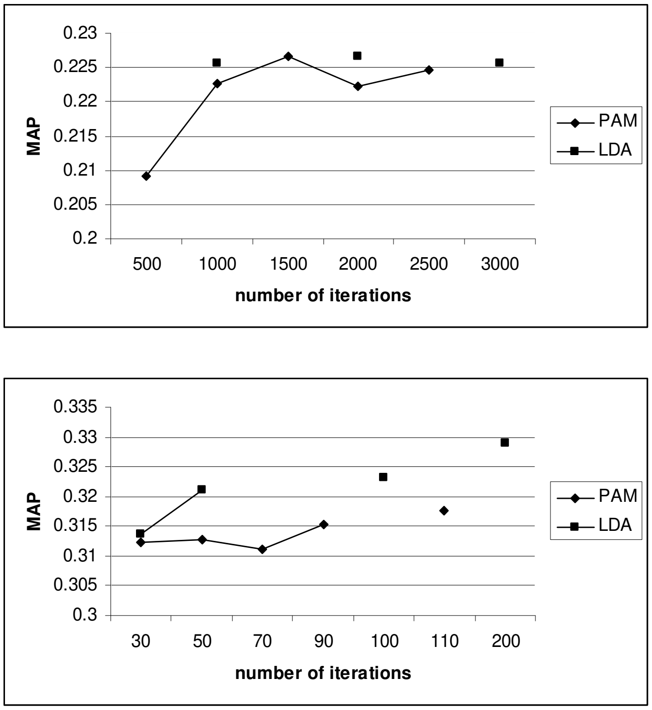


Figure 7.3 Retrieval results with PAM/LDA-based document models.

Figure 7.3 (top) is for the small dataset and the other is for the large data set. From these preliminary results we can see that the PAM-based document model has not achieved better results than LDA-based document model and thus we did not pursue large-scale experiments. In the retrieval results on the large data set, we pick up two queries on which one model is significantly better than the other. On query 068-"Health Hazards from Fine-Diameter Fibers", LDA-based document model performs better; on query 104-"Catastrophic Health Insurance" PAM-based document model performs better. For most of other queries, the retrieval results of these two models are quite close.

7.3 Special Words Topic Model

7.3.1 Introduction to SWB

Topic models are usually used together with the traditional word-based document models in the retrieval process because topics may be too coarse to be used as the only representation. This indicates that specific aspects in each document are not well captured by the topic model. The special words with background (SWB) model was recently proposed by Chemudugunta et al. (2006) as one of the state-of-the-art topic models. It extends the LDA model by representing each document as a combination of a mixture distribution over general topics, a background distribution over common words, and a distribution over words that are treated as being specific to that document.

Compared to the LDA model, the difference is that in SWB each word token is associated with a latent random variable x, taking value x=0 if the word w is generated from a topic as in the LDA model, x=1 if the word is generated from the special word distribution ψ for that document (with a Dicirchlet prior parameterized by β_1) and x=2 if the word is generated from a background distribution Ω for the collection (with a Dicirchlet prior parameterized by β_2). x is sampled from a document-specific multinomial π with a Dirichlet prior γ. The graphical representation of SWB is shown in Figure 7.4.

The SWB model has been shown to outperform LDA and TF*IDF on small and dense (the ratio of relevant documents are much higher than the usual TREC collections) collections when applied by itself without combination with the original document model: it captures more special words, and achieve better performance in retrieval experiments.

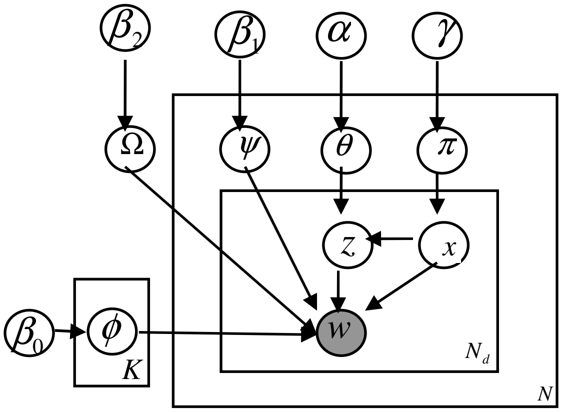


Figure 7.4: Graphical model representation of SWB.

This model targets modeling the general and specific aspects of documents and thus provides a generative framework for document modeling in retrieval, however, in retrieval tasks the two aspects are not separatable. Topic models are used in IR to address the problem of "vocabulary mismatch" by effectively adding in words that may be missing. The general aspects such as topics are to help interpret and understand the specific aspects such as exact words contained in a document. So these two aspects are not independent. Especially, in the early topic models such as term clustering based on word similarity, there is no generative framework and no separation of general and special aspect. Our experimental results show that retrieval with only SWB performs miserably, even much worse than the basic query likelihood (QL) model.

Although SWB does not gain good retrieval performance by itself, modeling general and special aspects may improve the topic distribution since it generates the words that are "most likely" in a topical word distribution from a topic. This may lead to more meaningful topics and thus better performance. SWB is a complicated model containing a huge number of parameters. We pursued a variant of SWB by having the multinomial *π*, which is the distribution of *x,* fixed for a corpus. That is, it will not be document-specific; instead, it will be collection-specific. We compare this simplified special words with background (SSWB) model with the original SWB and show that there is no performance loses. SSWB is just a variant of SWB. We developed this model not for better results than SWB, but to make the whole process of and the model less complicated and thus hopefully more efficient.

The conditional probability of a word *w* given a document *d* is:

$$\begin{aligned} p(w \mid d) = p(x=0 \mid d)\sum_{t=1}^{T} p(w \mid z=t)p(z=t \mid d) + p(x=1 \mid d)p^{'}(w) \\ + p(x=2 \mid d)p^{''}(w) \end{aligned} \qquad (7\text{-}5)$$

where *p*'(*w*) is the special word distribution and *p*''(*w*) is the background word distribution. We combine this document model with the original document model by the parameter mixture model.

7.2.2 IR Experiments

We conducted experiments on the same TREC data sets as in Chapter 5 and 6: the Associated Press Newswire (AP) 1988-90 with queries 51-150, Wall Street Journal (WSJ) 1987-92 with queries 51-100 and 151-200, Financial Times (FT) 1991-94 with queries 301-400, San Jose Mercury News (SJMN) 1991 with queries 51-150, and LA Times (LA) with queries 301-400. Queries are taken from the “title” field of TREC topics. These five collections, including the query sets and relevance judgments, are the same as used in previous chapters in order to compare the effectiveness of different topic models for retrieval. In all experiments, both the queries and documents are stemmed, and stopwords are removed.

The AP collection is used as our training collection to estimate the parameters. We first tuned the parameters based on the retrieval experiments with the standard LDA model. Then we applied similar setting on other models without especially tuning them. The only parameter that we changed for other models is the number of iterations. SWB/SSWB are much more complicated models, so we especially tuned the number of iterations on the AP collections for them. For both tuning processes we consider the efficiency and choose only the number of iterations after which there will not be big performance gain.

We use symmetric Dirichlet priors in the LDA estimation with $\alpha = 50/K$ and β_0=β_2 =0.01, β_1=0.0001, γ=0.3, which are common settings in the literature. Our experience shows that retrieval results are not very sensitive to the values of these parameters. In order to select a suitable value of λ, we experiment on the AP collection with the standard LDA model and find λ=0.7 to be the best value in our search. From the experiments on the testing collections, we also find that λ=0.7 is the best value or almost the best value for other collections. We set the Dirichlet prior μ=1000 since the best results are consistently obtained with this setting.

7.2.2.1 Comparison of SWB-based Retrieval and SSWB-based Retrieval

The retrieval results on the five collections are presented in Figure 7.5, with comparisons to the result of query likelihood retrieval (QL).

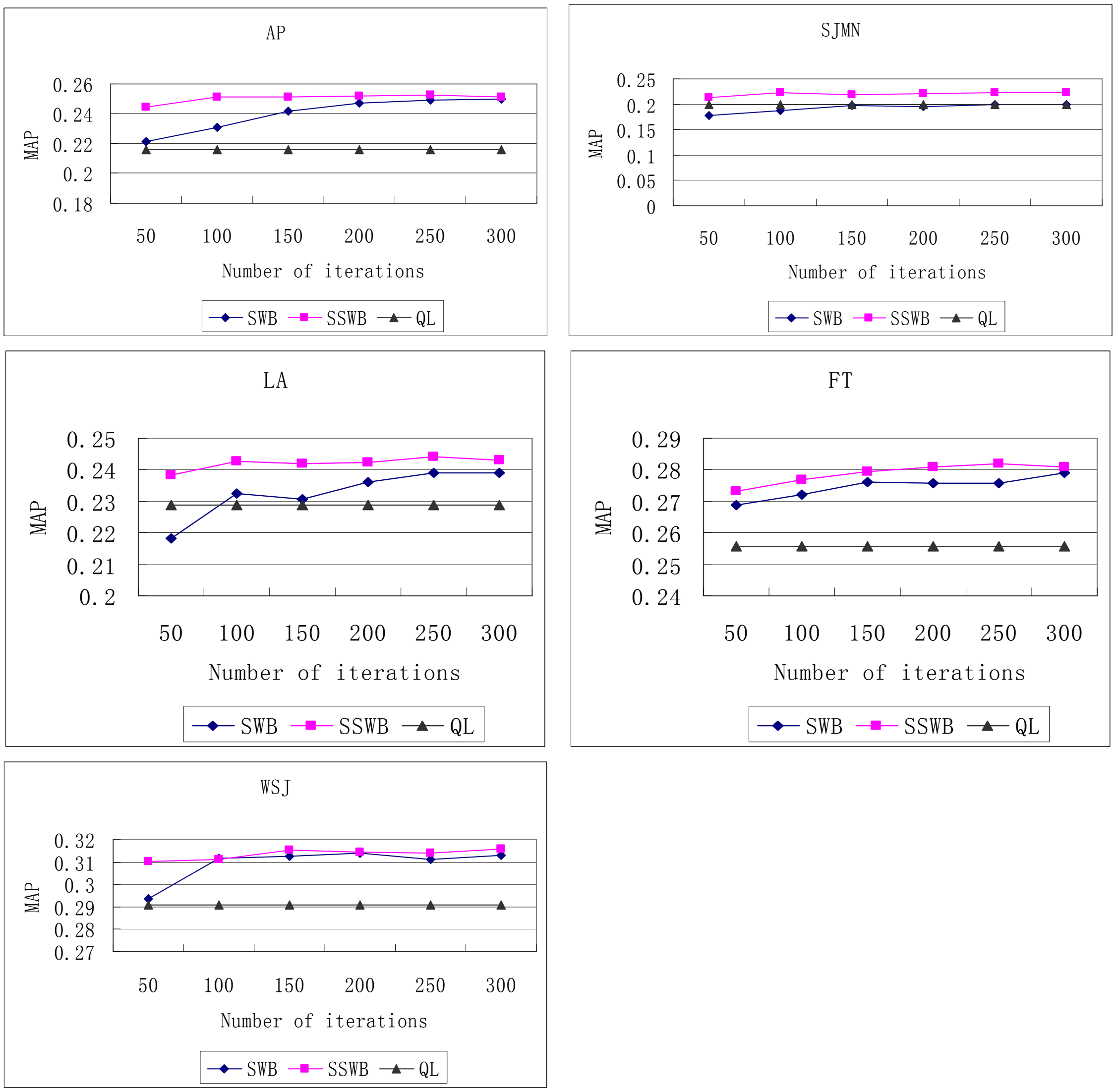


Figure 7.5: Comparison of SWB-based Retrieval and SSWB-based Retrieval in Mean Average Precision (MAP), with QL as a baseline. Number of iterations varies from 50 to 300. *K*=400 topics, 1 Markov chain.

We can see that retrieval with SSWB performs better than SWB within 300 iterations on all the five collections especially for small numbers of iterations. So in the following section we use SSWB to compare with the standard LDA model on retrieval tasks.

7.2.2.2 Comparison of SSWB-based Retrieval and LDA-based Retrieval

We inherit the parameters trained on the AP collection for the LDA-based retrieval, except the number of iterations. With the parameter setting λ=0.7, 100 iterations and 2 Markov chains, we run experiments with SBDM and present results in Table 7.5. We compare the results with the results of LBDM, and the results of the query likelihood model are also listed as a reference. On four collections, SSWB-based retrieval achieves improvements over both of query likelihood retrieval and LDA-based retrieval, and two of the improvements are significant (over LDA-based retrieval). Considering that LDA-based retrieval has already obtained excellent retrieval performance, and the parameters are turned for it, the performance improvements from SBDM are interesting.

Table 7.5: Comparison of query likelihood retrieval (QL), retrieval with LDA-based document models (LBDM), and retrieval with the SSWB-based document models (SBDM). The evaluation measure is average precision. %chg denotes the percentage change in performance (measured in average precision) of SBDM over QL and LBDM. "*" or "+" indicate statistically significant differences in performance between SBDM and QL/LBDM with a 95% confidence according to the Wilcoxon or Sign test.

Collection	QL	LBDM	SBDM	%chg over QL	%chg over LBDM
AP	0.2161	0.2567	0.2509	+16.1*$^{+}$	-2.26
SJMN	0.1985	0.2181	0.2204	+11.01*$^{+}$	+1.05
FT	0.2558	0.2750	0.2801	+9.47*$^{+}$	+1.82
LA	0.2290	0.2424	0.2493	+8.85*$^{+}$	+2.86*
WSJ	0.2908	0.3157	0.3206	+10.26*$^{+}$	+1.57^{+}

Table 7.6 shows an example of the topics associated with a document. The document is “AP900403-0219” that we discussed in Section 6.3.4.1. We list the top 5 topics for this document, with the top 10 words in each topic.

Table 7.6: An example of the topics associated with a document by SSWB.

Topic 1:	Topic 2:	Topic 3:	Topic 4:	Topic 5:
company	bank	company	yen	million
stock	debt	million	stock	1
share	loan	operate	point	5
shareholder	billion	business	close	2
takeover	pay	sale	trade	3
buyout	credit	corp	dollar	4
corp	interest	announce	market	7
billion	finance	own	exchange	6
co	lend	base	tokyo	8
percent	financial	sell	deal	estimate

We show the retrieval performance with SSWB. The improvements are not much, however, the topics are based only on part of the documents. We have also done retrieval experiments with the topics only as in the LDA-based retrieval by just ignoring the document-specific words and background words, and the retrieval performance remains similar (almost the same). On the five collections – SJMN, FT, LA, WSJ, AP, the fractions of words assigned to special words distribution with the SSWB model are 14%, 9%, 15%, 13% and 11% respectively. This is an interesting observation: by ignoring some of the text, the topics trained with probabilistic mixture models are comparable, or even more effective, to IR than the topics trained on the entire collection.

With this observation, further improvements on both of performance and efficiency on topic modeling for IR are possible.

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

The goal of the research in this thesis was to investigate topic models in IR framework and improve retrieval effectiveness. We studied all three types of topic models, developed or introduced a new topic model for each type, and presented two frameworks to integrate them. In Chapter 4, 5 and 6, we discussed one type of topic models in each chapter, explored their application on IR tasks, and compared the models within and across types regarding the efficiency and retrieval effectiveness on TREC data sets. Manually-built topic models improve retrieval effectiveness, but the overall results are not better than automatic methods and a method to selectively apply manually-built topic models has been shown to be promising. Automatic topic models, especially term group associating models as latent mixture models, have been shown to be effective to IR with reasonable efficiency. LDA-based document models achieve significantly better results over previous work, and term-term association based on joint probability also performs well with cost-benefit consideration.

In this chapter we will summarize the research contributions of this thesis, and describe future directions.

8.1 Contributions

The contributions of the research in this thesis are as follows:

- The first study of generative topic models used for representation in information retrieval. We investigate a range of topic models, especially generative topic models, in different manners of text representation in the language modeling framework. Retrieval effectiveness is evaluated and compared.

- The first evaluation of LDA-style topic models with very large text collections. We evaluate LDA and other recent topic models on several representative TREC collections of reasonable size.
- The first study of the computational efficiency issues with using LDA-style models for retrieval on very large text collections. Efficiency is a problem for many automatic topic models due to the expensive computation related with large text collections. We study the computation complexity of LDA-style topic models, and control the complexity with approximate parameter setting in the LDA training process.
- The first synthesis evaluation of older topic modeling techniques such as manually-built thesauri and term association on large scale collections. We propose a term associating method and compare its effectiveness with traditional similarity measures on TREC collections.
- A cost-benefit comparison of simpler topic-modeling techniques like term-term association with LDA-based techniques. Effectiveness and computation complexity are discussed and compared for different styles of topic models.

8.2 Future Work

There are a number of directions in which further research can be pursued:

1. Combining short-distance dependency and long-distance dependency of words. Long-distance (semantic) dependency and short-distance (phrase) dependency

have both been shown to be effective in IR experiments. These two types of dependencies are the two main factors that have been used to improve ad-hoc retrieval performance, but rarely combined. We have investigated an *n*-gram topic models which contain both factors and obtained some interesting preliminary results. Also, distance itself as a feature plays a crucial role in topic models: it is shown in Chapter 5 that the distance between words can be used to improve the word associations for IR, which can be regarded as a simple topic model. So, it will be interesting and promising to study the distance feature, considering both short-distance and long-distance dependency, incorporate them into state-of-the-art topic models and explore the effectiveness.

2. Faster or more effective topic modeling/approximation techniques for IR. The massive amount of data available today makes it often impossible to apply very complicated topic modeling techniques to large (even web-scale) data sets, and thus makes it extremely important to improve the efficiency of topic models. On the contrary, studies with SWB in Chapter 7 show that more complicated techniques can potentially further improve retrieval performance and/or effectiveness. Mimno and McCallum (2007) present DCM-LDA which is feasible on large-scale data, but it requires the corpus to be structured, and how to structure IR data, including Web data, is an open question; other faster topic modeling algorithms have also been developed, such as (Li, 2007), and it would be interesting to test their effectiveness on IR tasks. In addition, studies in Chapter 7 shows that training on part of the data may be as effective as the full. Topic models that are more efficient and effective are expected to further improve their application in IR.

3. Performance prediction and evaluation with latent mixture models. Performance prediction is a new task that IR researchers are just starting to pay more attention to. Accurate prediction has the potential of a crucial impact both on the users and the retrieval system. Evaluation has been a fundamental area in IR research for a long time. The recently developed latent mixture models can find out topics more accurate than previous techniques in retrieval experiments, and their applications in prediction and evaluation tasks can be beneficial to IR research.

4. Integrating or combining other features with topic modeling for IR. Most current automatic topic modeling techniques capture semantic associations by analyzing text co-occurrence, but data resource for IR often contains many other features, such as the date and author of the document, and the parts of speech of the terms. Especially, Web data contains hyperlinks, which has been confirmed to be an effective feature for retrieval. How to combine these features together is an interesting and promising direction. LSI and LDA-style topic models project the data into a latent semantic space; SVD-based hyperlink analysis also performs the similar projection. Cohn and Hofmann (2001) combine these two types of analysis; the effectiveness of these types of models in IR or whether it can be integrated into topic modeling itself would be very interesting.

Other directions for further research can be applying topic models selectively, e.g., different topic modeling approaches for different queries/documents, post-processing methods with obtained topics, or better combination strategies. In addition, the data and model we used for manually-built topic models have limitations. It would be interesting to examine other data sets and different situations for hand-crafted topic models.

BIBLIOGRAPHY

Allan, J. et al., Challenges in Information Retrieval and Language Modeling. Report of a Workshop held at the Center for Intelligent Information Retrieval. SIGIR Forum Spring 2003, 37 (1).

Azzopardi, L., Girolami, M and van Rijsbergen, C.J., Topic Based Language Models for Ad Hoc Information Retrieval. In Proceedings of the International Joint Conference on Neural Networks, Budapest, Hungary, 2004.

Bai, J., Song, D., Bruza, P., Nie, J., and Cao, G., Query Expansion Using Term Relationships in Language Models for Information Retrieval. In Proceedings of ACM 14th Conference on Information and Knowledge Management (CIKM), 2005, 688-695, 2005.

Berger, A. and Lafferty, J., Information Retrieval as Statistical Translation. In Proceedings of the 22nd ACM SIGIR Conference on Research & Development on Information Retrieval, 222-229, August 15-19, 1999, Berkeley, California, United States.

Belkin, N. and Croft, W.B., Retrieval Techniques. Annual review of information science and technology, Vol.22, 109-145, 1987.

Belkin, N. and Robertson, S.E., Information Science and The Phenomena of Information. Journal of the American Society for Information Science, 27, 4, 197-204, 1976.

Blei, D. M., Ng, A. Y., and Jordan, M. J., Latent Dirichlet Allocation. Journal of Machine Learning Research, 3, 993-1022, 2003.

Blei, D., Griffiths, T., Jordan, M., and Tenenbaum, J., Hierarchical Topic Models and the Nested Chinese Restaurant Process. In Proceedings of Advances in Neural Information Processing Systems, Cambridge, MA, MIT Press, 2004.

Brown, P.F., Della Pietra, V.J., deSouza, P.V., Lai, J.C. and Mercer, R.L., Class-based n-gram models of natural language. Comp. Linguistics, 18(4), 467—479, 1992.

Budzik, J., Hammond K., and Birnbaum, L., Information Access in Context. Knowledge Based Systems 14 (1-2), Elsevier Science. 37-53, 2001.

Burgess, C., Livesay, K., and Lund, K., Explorations in Context Space: Words, Sentences, Discourse. Discourse Processes, 25(2&3), 211—257, 1998.

Cao, G., Nie, J., and Bai, J., Integrating Word Relationships into Language Models. In Proceedings of the 28th ACM SIGIR Conference on Research & Development on Information Retrieval, 298-305, 2005.

Cao, G., Nie, J., and Bai, J., Using Markov Chains to Exploit Word Relationships in Information Retrieval. In Proceedings of the 8th Conference on Large-Scale Semantic Access to Content (RIAO'07), May 2007, Pittsburgh, US.

Chemudugunta, C., Smyth, P. and Steyvers, M., Modeling General and Specific Aspects of Documents with a Probabilistic Topic Model, In Proceedings of Advances in Neural Information Processing Systems, 2006.

Cohn, D. and Hofmann, T., The Missing Link: A Probabilistic Model of Document Content and Hypertext Connectivity, In Proceedings of Advances in Neural Information Processing Systems, 2001

Croft, W. B., A Model of Cluster Searching Based on Classification. Information Systems, Vol. 5, 189-195, 1980.

Croft, W.B. and Lafferty, J., Language Modeling for Information Retrieval. Kluwer International Series on Information Retrieval, 13, Kluwer Academic Publishers, 2003. http://www.wkap.nl/prod/b/1-4020-1216-0.

Croft, W.B., Lucia, T.J., Cringean, J., and Willett, P., Retrieving Documents By Plausible Inference: An Experimental Study. Information Processing and Management, 25, 599-614, 1989.

Croft, W.B. and Thompson, R., I3R : A New Approach to the Design of Document Retrieval Systems. Journal of the American Society for Information Science, 38(6), 389-404, 1987.

Cronen-Townsend, S. and Croft, W.B., Quantifying Query Ambiguity. In Proceedings of Human Language Technology 2002, 94-98, 2002.

Cronen-Townsend, S., Zhou, Y., and Croft, W.B., Predicting Query Performance. In Proceedings of the 25th ACM SIGIR Conference on Research & Development on Information Retrieval, 299-306, 2002.

Cronen-Townsend, S., Zhou, Y., and Croft, W.B., A Language Modeling Framework for Selective Query Expansion. CIIR Technical Report, IR-338, University of Massachusetts, Amherst, MA, 2004.

Crouch, C. J., An Approach to The Automatic Construction of Global Thesauri. Information Processing & Management, 26(5), 629-640, 1990.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R., Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science, 41(6), 391-407, 1990.

Dumais, S. T., Latent Semantic Indexing (lsi): Trec-3 report. In Proceedings of the Text Retrieval Conference (TREC-3). 219-230, 1995.

Dumais, S.T., Cutrell, E., Cadiz, J.J., Jancke, G.G., Sarin, R. and Robbins D.C., Stuff I've Seen: A system for personal information retrieval and re-use. In Proceedings of the 25th ACM SIGIR Conference on Research & Development on Information Retrieval, 2003.

Evans, D.A., Ginther-Webster, K., Hart, M., Lefferts, R.G. and Monarch, I.A., Automatic Indexing Using Selective NLP and First-order Thesauri. In Proceedings of RIAO'91, 624–643, 1991.

Fagan, J., The Effectiveness of A Nonsyntactic Approach to Automatic Phrase Indexing for Document Retrieval. Journal of the American Society for Information Science, 40(2):115–139, 1989.

Fang, H. and Zhai, C., Semantic Term Matching in Axiomatic Approaches to Information Retrieval. In Proceedings of the 29th ACM SIGIR Conference on Research & Development on Information Retrieval, 115-122, 2006.

Gao, J.F., Nie, J.-Y., Zhang, J., Xun, E., Zhou, M. and Huang, C.: Improving Query Translation for CLIR using Statistical Models. In Proceedings of the 24th ACM SIGIR Conference on Research & Development on Information Retrieval, 96-104, 2001.

Gauch S., Chaffee J., and Pretschner A., Ontology-Based Personalized Search and Browsing. Web Intelligence and Agent Systems, Vol. 1 No. 3-4, 219-234, 2004.

Geman, S., and Geman, D., Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. In IEEE Transactions on Pattern Analysis and Machine Intelligence, 6, 721-741, 1984.

Girolami, M. and Kaban, A., Sequential Activity Profiling: latent Dirichlet allocation of Markov chains. Data Mining and Knowledge Discovery, 10, 175–196, 2005.

Girolami, M. and Kaban, A., On an Equivalence between PLSI and LDA. In Proceedings of the 26th ACM SIGIR Conference on Research & Development on Information Retrieval, 433-434, 2003.

Grefenstette, G., Use of Syntactic Context to Produce Term Association Lists for Text Retrieval. In Proceedings of the 15th ACM SIGIR Conference on Research & Development on Information Retrieval, 89-97, 1992.

Griffiths, T. L., and Steyvers, M., Finding Scientific Topics. In Proceeding of the National Academy of Sciences, 5228-5235, 2004.

Griffiths, T. L., Steyvers, M., Blei, D. and Tenenbaum, J., Integrating Topics and Syntax. In Proceedings of Advances in Neural Information Processing Systems 17, 2005.

Hofmann, T. Probabilistic Latent Semantic Indexing. In Proceedings of the 22nd ACM SIGIR Conference on Research & Development on Information Retrieval, Berkeley, CA, USA, 1999.

Ingwersen, P., Information Retrieval Interaction. London: Taylor Graham. 1992.X, 246p, 1992.

Jing, Y. and Croft, W.B.: An Association Thesaurus for Information Retrieval, In Proceedings RIAO'94, 146-160, 1994.

Sparck Jones, K., Automatic Keyword Classification for Information Retrieval. London: Butterworths, 1971.

Katz, S. M., Estimation of Probabilities from Sparse Data for The Language Model Component of A Speech Recognizer. IEEE Trans. Acoustics, Speech and Signal Processing (ASSP) 35 400—401, 1987.

Krovetz, R., Viewing Morphology as An Inference Process. In Proceedings of the 16th ACM SIGIR Conference on Research & Development on Information Retrieval, New York, 191-202, 1993.

Kwon, O.W., Kim, M.C., and Choi K. S., Query Expansion Using Domain-Adapted, Weighted Thesaurus in An Extended Boolean Model. In Proceedings of ACM 3rd Conference on Information and Knowledge Management (CIKM), 140-146, 1994.

Leake, D., Scherle, R., Budzik, J., and Hammond, K. J., Selecting Task-Relevant Sources for Just-in-Time Retrieval. In Proceedings of The AAAI-99 Workshop on Intelligent Information Systems, AAAI Press, 1999.

Li, W., Pachinko Allocation: DAG-Structured Mixture Models of Topic Correlations. Ph.D. Thesis, University of Massachusetts, 2007.

Li, W. and McCallum, A., DAG-Structured Mixture Models of Topic Correlations. In Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, Pennsylvania, USA, 2006.

Lavrenko, V. and Croft, W. B., Relevance-Based Language Models. In the 24th ACM SIGIR Conference on Research & Development on Information Retrieval, 120-127, 2001.

Lin., D., An Information-Theoretic Definition of Similarity. In Proceedings of the 15th International Conference on Machine Learning, Madison, WI, 1998.

Liu, S., Liu, F., Yu, C., and Meng, W., An Effective Approach to Document Retrieval via Utilizing WordNet and Recognizing Phrases. . In Proceedings of the 27th ACM SIGIR Conference on Research & Development on Information Retrieval, 266-272, 2004.

Liu, X. and Croft, W. B., Cluster-Based Retrieval Using Language Models. In Proceedings of the 27th ACM SIGIR Conference on Research & Development on Information Retrieval, 186-193, 2004.

Lund, K. and Burgess, C., Producing High-dimensional Semantic Spaces from Lexical Co-occurrence. Behavior Research Methods, Instruments, & Computers, 28(2), 203—208, 1996.

Mandala, R., Tokunaga, T., and Tanaka, H., Ad Hoc Retrieval Experiments Using WordNet and Automatically Constructed Theasuri. In Proceedings of the seventh Text REtreival Conference, pages 475-481, 2998.

Manning, C.D., Raghavan, P., and Schütze, H., Introduction to Information Retrieval, Cambridge University Press, 2007.

Matveeva, I., Levow, G., Farahat, A., and Royer, C., Generalized Latent Semantic Analysis for Term Representation. In Proceedings of RANLP 2005.

McCallum, A. Multi-Label Text Classification with A Mixture Model Trained by EM. In AAAI workshop on Text Learning, 1999.

Mimno, D. and McCallum, A., Organizing the OCA: Learning faceted subjects from a library of digital books. In the Proceedings of the ACM IEEE Joint Conference on Digital Libraries, Vancouver, BC, Canada, June 2007

Mitra, M., Buckley, C., Singhal, A. and Cardie, C., An analysis of statistical and syntactic phrases. In Proceedings of RIAO'97, 200–214, Montreal, CA, 1997.

Ogilvie, P. and Callan, J., Experiments Using the Lemur Toolkit. In Proceedings of the 2001 Text Retrieval Conference, 103-108, 2001. http://www.lemurproject.org/

Pejtersen, M., Investigation of Search Strategies in Fiction Bases on An Analysis of 134 User-Librarian Conversations. In IRFIS 3 Proc. Oslo: Statens Biblioteksskole, 107-131, 1979.

Ponte, J. and Croft, W. B., A Language Modeling Approach to Information Retrieval. In Proceedings of the 21st ACM SIGIR Conference on Research & Development on Information Retrieval, 275-281, 1998.

Qiu, Y. and Frei, H., Concept Based Query Expansion, In Proceedings of the 16th ACM SIGIR Conference on Research & Development on Information Retrieval, 160-169, 1993.

Rosen-Zvi, M., Griffiths, T., Steyvers, M., and Smyth, P. The Author-Topic Model for Authors and Documents. In Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence. Banff, Alberta, Canada, 2004.

Salton, G., Automatic Text Processing. Addison-Wesley, 1989.

Salton G. and Buckley, C.: On the Use of Spreading Activation Methods in Automatic Information Retrieval. In Proceedings of the 21st ACM SIGIR Conference on Research & Development on Information Retrieval, 275-281, 1998.

Salton, G. and Lesk, M., Computer Evaluation of Indexing and Text Processing. Prentice-Hall, 143-180, 1971.

Salton, G., and Mcgill, M. J., Introduction to Modern Information Retrieval. McGraw-Hill, 1983.

Shen, X., and Zhai, C., Exploiting Query History for Document Ranking in Interactive Information Retrieval. In Proceedings of the 26th ACM SIGIR Conference on Research & Development on Information Retrieval, 2003.

Song F. and Croft W.B., A General Language Model for Information Retrieval. In the Proceedings of Eighth International Conference on Information and Knowledge Management, Kansas City, MO, November 2-6, 1999.

Steyvers, M. and Griffiths, T., Matlab Topic Modeling Toolbox 1.3. http://psiexp.ss.uci.edu/research/programs data/toolbox.htm, 2005

Steyvers, M. and Griffiths, T., Probabilistic Topic Models. In T. Landauer, D. S. McNamara, S. Dennis, & W. Kintsch (Eds.), handbook of Latent Semantic Analysis. Hillsdale, NJ: Erlbaum, 2007.

Strzalkowski, T., Natural Language Information Retrieval. Information Processing and Management, 31(3):397–417, 1995.

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M., Hierarchical Dirichlet processes. Technical Report, Department of Statistics, UC Berkeley, 2004.

Trajkova, J. and Gauch S., Improving Ontology-Based User Profiles. In Proceedings of RIAO'04, Vaucluse, France, 380-389, 2004.

van Rijsbergen, C.J., A Theoretical Basis for The Use of Co-occurrence Data in Information Retrieval, Journal of Documentation, 33, 106-119, 1977.

van Rijsbergen, C.J., Information Retrieval. 2nd edn. London: Butterworths, 1979. http://citeseer.ist.psu.edu/vanrijsbergen79 information.html

Voorhees, E. M., Query Expansion Using Lexical-semantic Relations. In Proceedings of the 17th ACM SIGIR Conference on Research & Development on Information Retrieval, 61-69, 1994.

Wallach, H., Topic Modeling: beyond bag-of-words. In Proceedings of the 23rd International Conference on Machine Learning, 2006.

Wang, X. and McCallum, A., A Note on Topical N-grams. Technical Report at University of Massachusetts (UM-CS-2005-071), 2005.

Wang, X., McCallum, A. and Wei, X., Topical N-grams: Phrase and Topic Discovery, with an Application to Information Retrieval. To appear in Proceedings of the 7th IEEE International Conference on Data Mining (ICDM), Oct 28-31, 2007.

Wei, X. and Croft, W. B., LDA-Based Document Models for Ad-hoc Retrieval. In Proceedings of the 29th Annual International ACM SIGIR Conference on Research & Development on Information Retrieval (SIGIR), pp. 178-185, 2006.

Wei, X. and Croft, W. B., Modeling Term Associations for Ad-hoc Retrieval Performance within Language Modeling Framework. In Proceedings of the 29th European Conference on Information Retrieval (ECIR), pp. 52-63, 2007.

Wei, X. and Croft, W. B., Investigating Retrieval Performance with Manually-Built Topic Models. In Proceedings of RIAO 2007 - 8th Conference - Large-Scale Semantic Access to Content (Text, Image, Video and Sound), paper number 12, 2007.

Xu, J., Solving the Word Mismatch Problem Through Automatic Text Analysis. Ph.D. Dissertation. Department of Computer Science, University of Massachusetts, 1997.

Xu, J. and Croft, W.B.: Query Expansion Using Local and Global Document Analysis. In Proceedings of the 19th ACM SIGIR Conference on Research & Development on Information Retrieval, 1996.

Zhai, C. and Lafferty, J., A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval. In Proceedings of the 24th ACM SIGIR, 334-342, 2001.