Semantics-based Language Models for Information Retrieval and Text Mining

A Thesis

Submitted to the Faculty

of

Drexel University

by

Xiaohua Zhou

in partial fulfillment of the

requirements for the degree

of

Doctor of Philosophy

November 2008

ACKNOWLEDGEMENTS

I would like to express my greatest thanks to my advisor, Xiaohua (Tony) Hu. Without his excellent and effective guidance, I could not have completed this thesis. Tony gave me the freedom to explore the research areas I was interested in at the initial stage, which finally led to my current thesis topic. After I decided on my research topic, he proposed many great ideas to improve my work. Tony would send me a couple of recently published papers related to my work almost every week, which broadened my mind very much. Tony sets very strict academic requirement for his students. But he is also a very generous friend in life. I enjoyed working with Tony closely over the past three years.

I appreciate the help I and support I received from many on the faculty at the iSchool at Drexel. Dr. Il-Yeol Song was my initial advisor at iSchool and also the coauthor of my first paper during my PhD studies. It is safe to say that he ignited my enthusiasm for research. In addition, he was a wonderful mentor for life. We had countless conversations regarding career development, job interviews, and even presentation skills, which will continue to benefit me for countless years, without a doubt. I even worked with Dr. Hyoil Han for about a year and a half, and I thank her for bringing me to the research area of natural language processing and text mining. And finally, I would like to express gratitude to Dr. Xia Lin, one of my committee members, for his many useful comments on my thesis proposal and other research.

I am grateful to all my friends and to my fellow PhD students with whom I spent five great years at Drexel. Muk has been my officemate for five years. Nan gave me a lot of

help in life, especially in the first year I moved to Philadelphia. George organized many parties to enrich our lives. Aek coauthored a couple of papers with me on question classification and question answering. Shanshan helped me process paperwork after I left Philadelphia. Deima and I had many interesting discussions on research projects. I would like to especially thank Xiaodan and his wife Yaya for being my best friends in Philadelphia. Xiaodan and I have worked very closely and coauthored over ten papers. Yaya cooked countless lunches and dinners for us when we were working on our papers.

The research related to this thesis was supported in part by NSF Career Grant IIS 0448023, NSF CCF 0514679, Pennsylvania Department of Health Tobacco Settlement Formula Grant (No. 240205 and No. 240196), and Pennsylvania Department of Health Grant (No. 239667). My travels to many academic conferences, including SIGIR'06, ECIR'06, IJCAI'07, CIKM'07, and SDM'08, were generously sponsored by the College of Information Science and Technology.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Semantics-based Language Models for Information Retrieval and Text Mining
Xiaohua Zhou
Xiaohua Hu

The language modeling approach centers on the issue of estimating an accurate model by choosing appropriate language models as well as smoothing techniques. In the thesis, we propose a novel context-sensitive semantic smoothing method referred to as a topic signature language model. It extracts explicit topic signatures from a document and then statistically maps them into individual words in the vocabulary. In order to support the new language model, we developed two automated algorithms to extract multiword phrases and ontological concepts, respectively, and an EM-based algorithm to learn semantic mapping knowledge from co-occurrence data. The topic signature language model is applied to three applications: information retrieval, text classification, and text clustering. The evaluations on news collection and biomedical literature prove the effectiveness of the topic signature language model.

In the experiment of information retrieval, the topic signature language model consistently outperforms the baseline two-stage language model as well as the context-insensitive semantic smoothing method in all configurations. It also beats the state-of-the-art Okapi models in all configurations. In the experiment of text classification, when the size of training documents is small, the Bayesian classifier with semantic smoothing not only outperforms the classifiers with background smoothing and Laplace smoothing, but it also beats the active learning classifiers and SVM classifiers. On the task of clustering, whether or not the dataset to cluster is small, the model-based k-means with semantic smoothing performs significantly better than both the model-based k-means with background smoothing and Laplace smoothing. It is also superior to the spherical k-means in terms of effectiveness.

In addition, we empirically prove that, within the framework of topic signature language models, the semantic knowledge learned from one collection could be effectively applied to other collections. In the thesis, we also compare three types of topic signatures (i.e., words, multiword phrases, and ontological concepts), with respect to their effectiveness and efficiency for semantic smoothing. In general, it is more expensive to extract multiword phrases and ontological concepts than individual words, but semantic mapping based on multiword phrases and ontological concepts are more effective in handling data sparsity than on individual words.

CHAPTER 1: INTRODUCTION

Statistical language modeling has a solid theoretical foundation and is usually effective for a variety of applications such as information retrieval (Ponte and Croft, 1998), text classification (McCallum and Nigam, 1998), text clustering (Zhong and Ghosh, 2005), and topic analysis (Blei et al., 2003; Hofmann, 1999). Statistical models often require a large number of training data to estimate an accurate and robust model. However, many applications in information retrieval and text mining can not provide a large number of training data. For example, in text retrieval, the system has to estimate a model for each single document; in text classification, it is very expensive to get labeled training documents. The lack of training data often degrades the performance of statistical language models.

Unlike statistical language models, human knowledge works well in the case of small training data. For example, the statistical language modeling approach will probably categorize a document that does not contain the word "car" but contains word "auto" as irrelevant to the query "car". However, human knowledge would find the document relevant to the query, because "auto" and "car" are synonyms. For this reason, we propose to incorporate semantic knowledge into the traditional language models in information retrieval and text mining. A statistical language model with the extension of semantic knowledge is referred to as a semantics-based language model in this thesis.

1.1 Statistical Language Model

A *statistical language model*, or more simply a *language model*, is a probabilistic mechanism for generating text. Such a definition is general enough to include an endless variety of schemes. To use language modeling, we usually need to estimate a model from training data and then compute the generative probability of a given text according to the estimated model. Model estimation is based on the assumption of word distributions made on the data. Multinomial distribution is frequently used in text retrieval, text clustering, text classification, and topic detection with statistical language models (Lafferty and Zhai, 2001; McCallum and Nigam, 1998; Zhong and Ghosh, 2005). In applications such as text retrieval, text clustering, and text classification, word orders are often not considered; that is, all words are independently distributed. The language model based on independence assumption is referred to as a unigram language model. Conversely, if word orders are modeled in applications such as speech recognition (i.e., the distribution of words depends on previous n-words) the model is called n-gram language model. Bigram language model (in which the present word depends on the previous word only) is the most frequently used in the family of n-gram language models.

The effectiveness of language modeling often depends on two factors. One is the choice of underlying word distributions. In general, if the model chosen reflects the distribution of the real data, it will work effectively. The other factor is the smoothing of the language model. Because of the lack of large numbers of training data, many words appearing in testing texts may not appear in training texts. Therefore, we should smooth the language model, assigning a nonzero probability to those “unseen” words, because zero probability is not allowed in probabilistic framework.

The purpose of language model smoothing, however, is much more than avoiding zero probability. For some applications such as text retrieval, suitable model smoothing can significantly improve the performance of information retrieval. For example, a document containing word “car” could be returned for the query “auto” if we can incorporate text semantics into the model smoothing. Lafferty and Zhai (2001) referred to such a smoothing method that incorporates context and synonym information into the model as *semantic smoothing*. A text often contains many general words, which usually have similar distributions over different topics. If the effect of those general words (i.e., noise) can be relieved, the performance of some applications such as text classification and clustering will definitely be improved. In short, the language model smoothing is a task to (1) assign reasonable probability values to those unseen words, and (2) adjust the probability values of those seen words.

Language modeling was initially used in speech recognition (Bahl et al., 1983). In recent years it has been widely applied into information retrieval because of its solid mathematical foundation and empirical effectiveness. In this thesis, we will not only study its use in information retrieval but also explore its effectiveness in text classification and clustering. Moreover, we will study new semantic smoothing methods for further improving the effectiveness of language models in information retrieval and text mining.

1.2 Information Retrieval and Text Mining

Information retrieval (IR) is a task of searching documents that satisfy particular user information needs. This task is often represented by a query. With the rapid growth of digital data such as web pages and scientific literature, information retrieval becomes more and more important. Instead of searching documents as a whole, text mining tries to find nontrivial and new patterns or knowledge from enormous textual data. Frequently used text mining applications include information extraction, text classification, text clustering, topic analysis, and so on.

Information retrieval and text mining have been extensively studied, and numerous methods and systems have been developed for these tasks. In recent years, language models have received more and more attention from researchers in this community. Language models not only have solid theoretical foundations but also achieve very good empirical results. In addition, the idea of this method is extremely simple. For whatever application, the use of language models can be decomposed into two problems: (1) estimating a language model from training texts and (2) computing the probability of generating text. However, statistical language models can be improved in many aspects. The following four issues are challenging language models when applied to information retrieval and text mining.

Handle Different Representations

The unigram language model is the model most often used. It captures individual word distributions with the assumption that all words are independent of each other. This assumption does not hold up very well in the real word. Moreover, it is very difficult to interpret the result with the unigram language model. For example, when one uses the unigram language model to summarize a set of biomedical articles, all multiword biological terms will break down into individual words, which gives final results that make no sense at all. To solve this problem, the language should be able to capture collocations (phrases) according to the context. An n-gram language model (Wang and McCallum, 2005) can capture the context of words.

The n-gram language model assumes that the distribution of the current word depends on the previous n words. To some sense, this idea is still motivated from the point of view of syntax because it limits the word dependency to only adjacent words. More semantically, a word in a sentence could link to other words that are not syntactically next to the target. Gao et al. (2004) has applied this idea into the language modeling approach to IR, and it has significantly improved the IR performance over the unigram language model. I argue that such a kind of semantic-based representation can capture more truth of the word structure.

Solve Data Sparse Problem

Language modeling approaches often need to first train a language model according to the training data and then apply the trained model to testing data. Thus, when training data is insufficient, some testing words will not appear in the training data. Because in probabilistic framework one can not assign zero probability to a word, one should take some strategy to smooth the trained language models, that is, assign a reasonable probability to unseen words. Laplace smoothing will simply assign a fixed probability to all unseen words. Some background smoothing methods (Zhai and Lafferty, 2001b) will interpolate the trained language model with a background collection (corpus) model. However, these smoothing approaches do not capture the semantic relationship between training words and testing words. For example, we have one document containing the word “auto” and a query for “car”. If we use Laplace smoothing or background smoothing, the document will not return for the query. But if the smoothing method takes into account the semantic relationship between the testing words and training words, the document will be retrieved for the query. In this sense, smoothing of language models does more than avoid zero probability. A suitable smoothing method can significantly improve the performance of the model.

Incorporate Contextual and Sense Information

Fundamentally, statistical language modeling is based on word co-occurrence. However, word polysemy is a wide phenomenon. If a language model assumes that the same word in different context has the same meaning, the model may fail or be compromised in some cases. Take the example of semantic smoothing discussed in preceding sections. To smooth the language model in a semantic fashion, one has to quantify the semantic relationship between words according to some training dataset. The word “mouse” may be mapped to both “keyboard” and “cat” with high probabilities without any contextual constraint because of the polysemy of “mouse”. But if the context is considered, say “computer” modifies “mouse”, “mouse” may still be mapped to “keyboard” with high probability, but to “cat” with low probability. The incorporation of contextual and sense information makes very specific semantic mapping. In general, if contextual and sense information is considered when developing language models, the model will be more accurate and effective.

Eliminate Data Noise

Word distributions in real texts are always the mixture of a series of models from the same family but with different parameters. For a particular application, it is always the case that only one or several models are of interest, and the remaining models will be excluded as “noise”. This idea is very useful for many applications, especially topic detection and summarization. For example, when one attempts to obtain a set of topical words from a given document set, if all words are treated as a sampling from one model, all stop words will be recognized as topical words, which actually do not make sense at all. But if an extra background is introduced, this problem could be solved. However, eliminating noise is not always as simple as introducing a background model.

The aforementioned four issues are especially severe in some domains such as biomedical literature. Texts in biomedical literature present several unique characteristics in comparison with texts in other domains. First of all, biological and medical terms often use multiple words to represent a unit meaning. For example, if one breaks down the terminology “high blood pressure” into three common words “high”, “blood”, and “pressure”, it does not make sense at all. Second, word synonymy and polysemy is a serious problem in biomedical literature. A biological term such as proteins and genes often has many synonyms. Meanwhile, the wide use of short names and abbreviations deteriorates the polysemy problem. Third, biomedical literature is full of various biological relationships. These relationships are very important to biologists. However, the sentence structure in biomedical articles is often quite complex and hence difficult to automatically extract these relationships. Fourth, this domain has a very large term space. For example, UMLS (Unified Medical Language System) has a collection of more than one million concepts in the domain of biomedicine. From the point of view of statistical language modeling, these characteristics violate the underlying assumptions of many simple language models. Thus, the direct adoption of statistical language models into biomedical literature may end with bad performance.

The four issues are all related to text semantics. We believe language models with the augmentation of semantic knowledge will achieve better performance. The availability of more and more general or domain-specific online dictionaries, thesauruses, and ontologies makes it possible to enhance traditional language models with new power. However, how to adopt these resources to the traditional text mining approaches becomes a challenging problem. Many empirical studies reported positive results after incorporating human-created knowledge into information retrieval and text mining. But the majority of these approaches are somehow ad hoc. It is then urgent to motivate a formal framework to adopt various semantic resources.

1.3 Research Questions

In general, the thesis will answer four research questions. First, *where and how can one learn the semantic mapping knowledge*? Semantic mapping is one of the most frequently used semantic knowledge. Semantic mapping in this thesis is equivalent to finding a weighted vector for each topic signature. The element of the vector corresponds to a word in a vocabulary. The weight for each element indicates the semantic association between the topic signature and the corresponding word. The summation of all weights is equal to one. We want to find a method which can learn semantic mapping knowledge efficiently and effectively. Moreover, the training data the method requires should be easy to collect. Ideally, the semantic mapping knowledge produced is reusable.

Second, *how can we effectively integrate language models with semantic mapping knowledge*? The utilization of semantic mapping knowledge for applications such as information retrieval, text classification, and clustering is not trivial. There are hundreds of ways to integrate semantic knowledge in literature. Some of them are ad hoc, and some of them are theory-oriented. Some of them work well only on particular applications or datasets, and some of them are effective in very general sense. We want to find a generic approach to combine language models and semantic mapping knowledge. The new approach not only fits information retrieval, but also suits text clustering and classification. It not only works on general domains such as news stories and web pages, but also brings significant improvement on very specific domains such as biomedical literature. Besides, the new approach does not require extensive tuning when applying to new applications or new domains.

Third, *does semantic mapping improve the language modeling approach to text retrieval and mining*? This question will test if the language models with semantic mapping knowledge outperform the ones without semantic mapping knowledge. To answer this question, we will evaluate the new semantics-based language models on three applications: information retrieval, text classification, and text clustering. For each application, we test the new model on several datasets in different domains. In particular, we are interested in the biomedical domain and news domain. This research question also includes two sub-questions. Is the semantics-based language model sensitive to parameters? For this reason, we also monitor the performance variance of the semantics-based language model when its parameter changes. Do all types of semantic mapping knowledge improve the language model? In the experiment, we evaluate three types of semantic mapping knowledge. We are especially interested in the interaction effect between semantic mapping knowledge and data domain.

Last, *does the semantics-based language model outperform other state-of-the-art approaches to text retrieval and mining*? In literature, there are many approaches to information retrieval, text classification, and text clustering. Language modeling is only one of the most effective approaches. For this reason, we compare semantics-based language models to other state-of-the-art nonlanguage model approaches. For example, in an information retrieval experiment, we evaluate famous Okapi models; in text classification, we compare with the SVM method; and in text clustering, we evaluate spherical k-means.

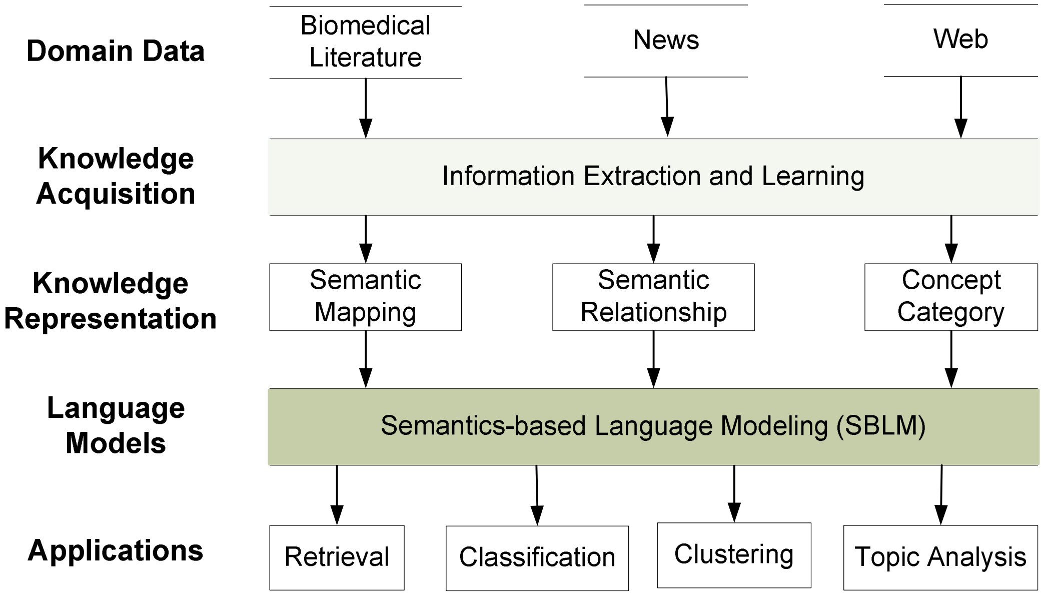


Figure 1.1: The research framework of the semantics-based language modeling.

In summary, our research framework for the semantics-based language models is shown in Figure 1.1. We first extract useful information from various domains and then transform that information into semantic knowledge through a machine-learning process. The learned semantic knowledge is then utilized by semantics-based language models that will be applied to various text applications. Because of time and data source constraints, we focus on two domains (news and biomedical literature), one knowledge representation (semantic mapping knowledge), and three text applications (information retrieval, text classification, and text clustering) in this thesis.

1.4 The Organization of the Thesis

Chapter 2 reviews the related work including traditional language model smoothing approaches, existing semantic smoothing approaches, and other approaches to information retrieval, text classification, and text clustering.

In Chapter 3, we propose the topic signature language model that incorporates semantic mapping knowledge into language models. This chapter first introduces the notion of topic signatures and the general mechanism of utilizing topic signatures for language model smoothing. Then three types of topic signatures (individual words, multiple-word phrases, and ontological concepts), as well as their extraction approaches, are described. After that, we present an EM-based algorithm to learn semantic mapping knowledge, that is, mapping topic signatures to a set of individual words. Last, we propose the topic signature language model. Chapter 3 answers our first and second research questions.

Chapters 4, 5, and 6 present three applications of the topic signature language model. They are information retrieval, text classification, and text clustering, respectively. In these applications, we evaluate the following tasks:

(1) Does the topic signature language model outperform baseline language models?

(2) Which type of topic signature is most effective for topic signature language models: words, multiword phrases, or ontological concepts?

(3) How robust is the topic signature language model for those applications?

(4) Does the topic signature language model outperform other state-of-the-art approaches to text retrieval and mining?

To avoid dataset bias, we evaluate each application on multiple datasets. In particular, we test news collections and biomedical literature.

In Chapter 7, we summarize the thesis and discuss the future work.

1.5 Notations

We would like to introduce the notations used in the thesis before we describe the details of our method. The frequently used notations are listed in table 1.1.

Table 1.1: Notations used in the thesis.

Notation	Description
C	A collection of documents
d, d_i	A document
c, c_i	A cluster or class
Q	A query
q, q_i	A query term
W	A sequence of words
w	A word
$c(w,d)$	The count of word w in document d
V	The vocabulary
t, t_k	A topic signature
*	A change is significant according to the paired-sample t-test at the level of $p<0.05$
**	A change is significant according to the paired-sample t-test at the level of $p<0.01$

CHAPTER 2: LITERATURE REVIEW

2.1 Language Models

Language models were initially used to improve the performance of speech recognition systems (Bahl et al., 1983). The task of speech recognition could be framed as to predict a series of words given the acoustic signals. As we know, it is extremely challenging to build an accurate acoustic model due to accent and physical difference of individuals at different conditions. Even if one could develop a perfect acoustic model for each word, the model would still be unable to handle the problem when the pronunciations of two words are identical, for example, "I" and "eye". With the help of language models, the search space and error can be dramatically reduced. For example, if one knows the previous word is "his", the current word is more likely to be "eye" rather than "I".

Formally, the task of speech recognition is to predict a series of words *W* which maximizes the joint probability of the language model *P(W)* and the acoustic model *P(W|S)* given the acoustic signal *S*.

$$W = \arg\max_{W} P(W)P(W \mid S) \qquad (2.1)$$

If one assumes the next word only depends on the preceding word, the language model will be a bigram model as described in the equation (2.2).

$$P(W) = P(w_0)\prod_{\mathrm{i=1}}^{\mathrm{n}} P(w_i \mid w_{i-1}) \qquad (2.2)$$

The model parameter $P(w_i \mid w_{i-1})$ can be learned from training data.

Ponte and Croft (1998) first introduced the language modeling approach to information retrieval. In this approach, the relevance of a document to a query is defined as the generative probability of the query by the underling model of the document.

$$\mathrm{Rel}(Q,d) \propto p(Q \mid d) \qquad (2.3)$$

In the simple case, the query terms are assumed to be independent of each other. The likelihood of the query by the document can be decomposed into

$$p(Q \mid d) = \prod_i p(q_i \mid d) \qquad (2.4)$$

and similarly, the model parameter $p(q_i \mid d)$ can be estimated from each document.

The language modeling approach to text classification assumes a two-step text generation process. First one samples a class from a mixture class model. Then one samples words according to the class model to generate a text (usually document). Thus, the task of text classification is reduced to finding a class label which maximizes the following joint probability:

$$C(d) = \arg\max_{c_i} p(c_i) p(d \mid c_i) \qquad (2.5)$$

If the multinomial distribution is assumed for the document (McCallum and Nigam, 1998), we have:

$$p(d \mid c_i) = \prod_{j=1}^{|d|} p(w_j \mid c_i) \qquad (2.6)$$

If the bigram language model is assumed for the document (Peng et al., 2004), we then have:

$$p(d \mid c_i) = p(w_1)\prod_{j=2}^{|d|} p(w_j \mid w_{j-1}, c_i) \qquad (2.7)$$

From the aforementioned three applications, we can see that the use of language models in text applications have two steps. In the first step, one inferred a language model from training text. In the second step, the estimated language model was utilized to predict the generative probability of a new text.

2.2 Language Model Smoothing

In this thesis, we apply language models to three applications: information retrieval, text classification, and text clustering. Multinomial distributions are often assumed for those three applications. With this assumption, the document language model can be simply estimated by a maximum likelihood estimator:

$$p(w \mid d) = \frac{c(w,d)}{\sum_i c(w_i,d)} \qquad (2.8)$$

where *c(w, d)* denotes the count of word *w* in the document *d*.

This means the probability will be zero if a word never appears in the training document. Such zero probability should be prevented. Otherwise, the product of all word probabilities will be zero, as illustrated in formula 2.4 and 2.6, no matter how important other words are. To prevent zero probability, the raw language model should be smoothed. Technically, the smoothing is the task to adjust low probabilities upward and high probabilities downward. We want to ultimately obtain a more accurate language model by smoothing. In this section, we would like to review traditional smoothing methods for unigram language models. In section 2.3, 2.4, and 2.5, we will summarize previous work that utilizes semantic knowledge to improve information retrieval, text classification, and text clustering, respectively. We are especially interested in the work utilizing semantic knowledge for language model smoothing.

Additive smoothing (also called Laplace smoothing) is one of the simplest smoothing methods (Lidstone, 1920; Johnson, 1932; Jeffreys, 1948). It simply adds one count to all words in the space.

$$p(w \mid d) = \frac{1 + c(w,d)}{\mid V \mid + \sum_i c(w_i, d)} \qquad (2.9)$$

where V is the vocabulary. This method is designed mainly for the purpose of preventing zero probability. It is frequently used in practice for Bayesian text classification.

Instead of simply adding one count, the Good-Turing (Good, 1953) method adjusts the word counts in the following way. For a word that occurs exactly r times, its count will be adjusted to:

$$c(w,d)^* = (r+1)\frac{n_{r+1}}{n_r} \qquad (2.10)$$

where n_r is the number of words that occur exactly r times in the training data. The final probability will be calculated by the formula below:

$$p(w|d) = \frac{c(w,d)^*}{\sum_i c(w_i,d)^*} \qquad (2.11)$$

The Jelinek-Mercer (Jelinek and Mercer, 1980) method linearly interpolates the maximum likelihood model with a corpus model (also referred to as background collection model).

$$p(w|d) = (1-\lambda)P_{ml}(w|d) + \lambda p(w\,|\,C) \qquad (2.12)$$

where $P_{ml}(w|d)$ and $p(w\,|\,C)$ are the maximum likelihood estimator of the document model and corpus model, respectively; and the coefficient λ controls the influence of the corpus model in the mixture model.

Dirichlet smoothing assumes that words in the document follow Dirichlet distribution (MacKay and Peto, 1995). Each word has a prior count as the parameter of the distribution. Zhai and Lafferty (2001a) used a corpus model to set the Dirichlet parameters, for example,

$$(\mu p(w_1\,|\,C), \mu p(w_2\,|\,C), ..., \mu p(w_n\,|\,C) \qquad (2.13)$$

and the word probability after smoothing becomes

$$p(w|d) = \frac{c(w,d) + \mu p(w\,|\,C)}{\sum_i c(w_i\,|\,d) + \mu} \qquad (2.14)$$

The absolute discounting method (Ney et al., 1994) namely subtracts a constant count from the seen word counts and then interpolates with a corpus model. The model is given by

$$p(w|d) = \frac{\max(c(w,d) - \delta, 0)}{\sum_i c(w_i\,|\,d)} + \sigma p(w\,|\,C) \qquad (2.15)$$

where $\delta \in [0,1]$ is a discount constant and $\sigma = \delta n(d)/c(,d)$. *n(d)* denotes the number of unique words in the document *d*.

2.3 Information Retrieval

In recent years, many methods have incorporated semantics and context information into the language model smoothing. Roughly, these methods were developed to expand either query models or document models for ad hoc information retrieval. Song and Bruza adopted information flow (IF) for query expansion (Song and Bruza, 2003). The context of a concept is represented by a HAL vector; the degree of one concept inferring another can then be computed through vector operators. Song and Bruza also invented a heuristic approach to combine multiple concepts, which enabled information inference from a group of concepts (premises) to one individual concept (conclusion). Thus, their query expansion technique was somehow context sensitive. However, it was difficult to extend it to document model expansions. Besides, the degree to which one individual concept could be inferred from another combined concept was not theoretically motivated; its robustness needs to be further validated.

Similarly, Bai et al. (2005a) used significant term pairs to expand query models. The combination of two terms is helpful to disambiguate their context and thus can capture more sense of the query. The expanded query model based on significant term pairs looked as follows:

$$p(w \mid Q) = (1-\lambda) \sum_{q_i, q_j \in Q} p_R(w \mid q_i q_j) p(q_i q_j \mid Q) + \lambda p_{ML}(w \mid Q) \qquad (2.16)$$

Here the second term is a unigram query model for smoothing purpose, and the first term (query expansion) is based on topic decomposition and mapping. The topic decomposition term $p(q_i q_j | Q)$ is simply assumed to be uniformly distributed. The topic mapping term $p_R(w | q_i q_j)$ is estimated based on term co-occurrence statistics. The coefficient λ controls the influence of the expansion component. Like the information flow approach, this approach is also inappropriate for document model expansions because the distribution of term pairs in a document is obviously not uniform. Besides, the co-occurrence–based estimation algorithm tends to assign higher probability values to general terms than to specific terms.

Berger and Lafferty proposed the statistical translation model for the first time in SIGIR’99. With this model, a term in a document is statistically mapped to query terms as described in the formula below:

$$p(q | d) = \sum_{w} t(q | w) l(w | d) \qquad (2.17)$$

where *t(q|w)* is the translation probability from document term *w* to query term *q*, and *l(w|d)* is the unigram document model. The translation model achieved significant improvement over the simple language model on two TREC collections. However, the model only captures the semantic relationship between individual words and is unable to incorporate the contextual information into the translation procedure. In addition, the training of translation probability requires a large number of real query-document pairs, which are very difficult to obtain. For this reason, Berger and Lafferty used synthetic data in the experiment. Besides, a document often contains a considerable number of unique terms, and thus the model expansion through document and query term mapping is computationally intensive.

The cluster language model (Liu and Croft, 2001) may be the first trial of topic decomposition and mapping for document model expansions. Liu and Croft incorporated cluster information into document model estimation:

$$p(w \mid d) = \frac{N_d}{N_d + u} p_{ML}(w \mid d) + (1 - \frac{N_d}{N_d + u}) p(w \mid cluster) \qquad (2.18)$$

N_d is the length of the document and μ is a parameter for smoothing. The document clusters are very similar to our topic signatures in the sense that both use a set of documents with similar context rather than a single document to estimate a more accurate topic model. However, in their cluster model, a document is associated with a single cluster, which may become problematic for especially long documents, whereas in our model a document can have multiple topic signatures. Furthermore, the clustering for a large collection is extremely inefficient. Lots of decisions need to be made empirically for clustering, based on the domain knowledge and the collection (e.g., the number of clusters, clustering algorithm, static clustering, or query-specific clustering); the topic signature model does not have these problems.

Latent topic models such as pLSI (Hoffman, 1999) assume that a document is generated by a set of topic models with certain distribution. Each topic model is about the distribution of words in a given vocabulary. With topic model assumption, a document is modeled as follows:

$$p(w \mid d) = \sum_{i=1}^{k} p(t_i \mid d) p(w \mid t_i) \qquad (2.19)$$

Here k is the total number of topics in the corpus. The parameter $p(w \mid t_i)$ is the probability of topic t_i generating word w. The parameter $p(t_i \mid d)$ is the probability of topic t_i being generated by document d. Within the framework of latent topic models, a document can be associated with multiple topics, and thus it overcomes the limitation of the cluster language models. Hoffman evaluated the pLSI model for retrieval tasks within the framework of vector space model (Hoffman, 1999). The pLSI model significantly outperformed the LSI model as well as the standard raw term matching method. But the size of four testing collections is far from the representative of realistic IR environments, and the baseline model is also far from state of the art, making the effectiveness of the pLSI model on retrieval unclear.

The idea of topic signature is very similar to the latent topic. The major difference lies in their implementations. The number of free parameters $p(t_i \mid d)$ and $p(w \mid t_i)$ in the latent topic models is mainly in proportion to the number of documents for a large collection, which will cause serious overfitting problems when the Expectation Maximum (EM) algorithm (Dempster et al., 1977) is used for model estimations. The estimation process also lacks scalability because all parameters should be estimated simultaneously. The worst problem is that when a new document is coming, there is no way to estimate the topic mixture $p(t_i \mid d)$. In our approach, we explicitly extract topic signatures from documents in the corpus. Thus, we can estimate each topic signature model $p(w \mid t_i)$ separately. We can also simply use maximum likelihood estimator to approach $p(t_i \mid d)$, whether the document is new or not. In short, the estimation of parameters for topic signature language model is very efficient and scalable as well as applicable to new testing documents.

Wei and Croft (2006) proposed a LDA-based document model for ad hoc retrieval. Unlike the pLSI model where topic mixture is conditioned on each document, the LDA model samples topic mixture from a conjugate Dirichlet prior that remains same for all documents (Blei et al., 2003). This change can solve the overfitting problem and the problem of generating new documents in pLSI. To make up for the possible information loss, the LDA model is further interpolated with a simple language model. The final document model is:

$$\begin{aligned} p(w \mid d) = \lambda(&\frac{N_d}{N_d + u} p_{ML}(w \mid d) + (1 - \frac{N_d}{N_d + u}) p(w \mid coll)) \\ &+ (1 - \lambda) \sum_{i=1}^{k} p(t_i \mid d) p(w \mid t_i) \end{aligned} \tag{2.20}$$

The LDA model improved the retrieval performance of both the simple language model and the cluster language model on five TREC collections (Wei and Croft, 2006). The LDA model is estimated through Gibbs sampling, which is computationally intensive. Thus, compared to the topic signature language model, the LDA model suffers from the computing intensity as well as lack of scalability.

2.4 Text Classification

In literature, there are two lines of work that utilize semantic information to improve text classification performance. One used semantic features such as latent topics, ontological concepts, and compound terms to enhance discriminative classifiers. The other improved the generative classifiers by modeling word dependency that was more meaningful semantically.

Bloehdorn and Hotho (2004) and Yetisgen-Yildiz et al. (2005) extracted ontology-based concepts to supplement single-word features during text classification. Bloehdorn and Hotho (2004) used WordNet and MeSH for concept extraction and further employed a boosting algorithm AdaBoost (Schapire and Singer, 2000) for text classification. Yetisgen-Yildiz et al. (2005) extracted UMLS concepts from Medline abstracts and used SVM (Joachims, 1998) for classification. Both classification systems achieved slight improvement over the baseline, which only used single-word features. However, the reliance on ontologies hinders the extension of their approaches to public domains where there may be no ontologies available. The mapping of text sequence to ontology concepts is not trivial because it involves the sense disambiguation. The inappropriate map may seriously hurt the performance of the text classifier.

Cai and Hofmann (2003) represented a document by a set of weighted latent concepts and then classified the documents by the AdaBoost algorithm. The latent concepts and their weights in different documents were automatically generated by the probabilistic latent semantic analysis model (pLSI) (Hofmann, 1999). The limitation of this approach is that it can not represent a new document by the latent topics learned previously. Theoretically, the LDA (Blei et al., 2003) model can solve this issue because it includes a document model and is able to compute the distribution of latent topics in the new document, assuming the new document is generated by the identical Dirichlet distribution. However, there is very little empirical work to validate the effectiveness of

LDA for new documents in the setting of text classification. The majority of the existing works simply model training documents and testing documents in the same process.

Lewis (1990) used syntactic phrases for document representation. The syntactic phrases are generated from parsing trees, and no statistical constraints are imposed. So a large number of phrases will be generated, but most of them have a low frequency in the collection. In other words, the generated syntactic phrases are very sparse and not qualified as effective features for text classification from the point of view of feature selection. In addition, the syntactic phrases are very noisy and lack semantics, which further deteriorates the performance of text classification.

Bai et al. (2005b) integrated compound terms into in Bayesian text classification. Unlike fixed-length n-grams, their compound terms are length-variable natural phrases. They were motivated to relax the independence assumption of the naïve Bayesian model rather than to take advantage of multiword phrases' discriminative powers. In their approach, when a compound term is identified, the constituent words will not be extracted as single-word features any more. Thus, it is very important to smooth the compound features in the class model. The authors used n-gram statistics to smooth the compound term, which made the smoothing very difficult and inefficient when the compound was very long, say, containing four words or more.

The statistical phrases (e.g., bigram) were used to improve the document representation in (Peng et al., 2004). The statistical phrases can capture the dependency between words and, to some degree, relax the independence assumption of the Bayesian classifier. Therefore, statistical phrases can improve the performance of Bayesian text classification. The experiments on seven datasets showed that the bigram language model with appropriate smoothing outperformed the unigram language model on the task of text classification. In the experiment, they tried five smoothing methods: absolute discounting, Laplace, Good-Turing, Jelinek-Mercer, and Witten-Bell.

However, the size of the vocabulary of grams in n-gram is huge due to the combination. A large amount of training data is needed to obtain a creditable model. The statistical n-multigram language model partially overcomes the limitations of the n-gram model (Shen et al., 2006). It never generates a huge number of word combinations, but it does generate a reasonable number of statistically significant multigrams. The phrases can be at variable length, which makes more sense; the generated multigrams are often meaningful. Shen et al. reported slight improvement over the baseline on a subcollection of RCV1 when using n-multigram model for text classification. However, this model still needs considerable training data. Otherwise very few multigrams will be generated. Although the multiword phrase used in this thesis is also a sort of statistical phrase, it is not required to use a large number of training data. We manage to build phrase dictionaries from large numbers of unlabeled texts and then extract multiword phrases from training data and testing data based on the built dictionaries.

2.5 Text Clustering

There are two major approaches to text (document) clustering: discriminative and generative (Kaufman, L. and Rousseuw, 1990). The discriminative approaches calculate the pair-wise document similarity (or distance) and group documents into clusters that minimize the intra-cluster distance and maximize the inter-cluster distance. It usually suffers from the $O(n^2)$ complexity. The generative approaches attempt to learn generative models from the collection and the clustering procedure is equivalent to finding out the cluster model that generates each document in the collection. The complexity is usually linear to the number of documents, that is, $O(n)$.

Agglomerative hierarchical clustering is a typical similarity-based discriminative clustering approach. According to the method of computing the distance between a document and cluster, agglomerative hierarchical clustering can be further divided into single-linkage, complete linkage, and average linkage. Empirically, average linkage achieved the best result in document clustering. Steinbach, Karypis, and Kumar (2000) concluded that spherical k-means consistently outperformed agglomerative hierarchical clustering on many textual datasets. They attributed the poor performance of agglomerative hierarchical clustering to the sparsity of topic-specific “core” words and density of topic-free “general” words between two documents.

Zhong and Ghosh (2005) conducted a comparative study of generative models for document clustering. Three probabilistic models—multivariate Bernoulli, multinomial, and von Mises-Fisher—were compared within a model-based k-means framework. They found out that multivariate Bernoulli performed consistently worse than multinomial, which was consistent with the finding by McCallum and Nigam (1998) that multinomial was more effective than Bernoulli in text classification. The von Mises-Fisher model performed slightly better than multinomial on most datasets in the experiment, but was roughly in the same magnitude of quality. Spherical k-means was a special case of von

Mises-Fisher. They also compared three generative models to the CLUTO (Karypis, 2002), a graph-partition–based clustering algorithm. The performance of multinomial and von Mises-Fisher was roughly comparable to that of CLUTO. But CLUTO is much more computationally expensive than generative models.

There is little work addressing the model smoothing issue for model-based k-means. In Zhong and Ghosh's comparative study, the simplest Laplace smoothing was used to smooth both Bernoulli and multinomial models. In this thesis, we will study the impact of semantics-based smoothing on the effectiveness of multinomial models for text clustering.

CHAPTER 3: TOPIC SIGNATURE LANGUAGE MODELS

3.1 Introduction

To use language models, we usually need to estimate a model from training data and then compute the generative probability of a given text according to the estimated model. However, it is challenging to estimate an accurate model due to the sparsity of training data. On one hand, many words in the testing text may not appear in the training data; to prevent zero probability, it is required to assign reasonable nonzero probability values to those unseen words. On the other hand, some words in the training data such as stop words are very noisy; it is better to adjust their probability values downward. Thus, the core of the language modeling approach to information retrieval and text mining is to smooth the raw language models. Zhai and Lafferty (2001a and 2002) propose several effective background smoothing techniques that interpolate the document model with the background collection model.

A potentially more significant and effective smoothing method is semantic smoothing, which incorporates human knowledge or word semantics into the language model estimates. The topic signature language model (TSLM) is one of such semantic smoothing methods. Before moving to the details of topic signature language model smoothing, we would like to introduce the framework for semantic smoothing as follows.

$$p(w|d) = (1\text{-}\lambda)p_b(w|d) + \lambda\sum_k p(t_k|d)\,p(w|t_k) \qquad (3.1)$$

Without losing generalization, *d* refers to a document here. It can be interpreted as a cluster or as a class for the application of clustering and classification. The first term is a unigram language model smoothed usually by a corpus-based method such as Jelinek-Mercer, Dirichlet, absolute discount (Zhai and Lafferty, 2001a), or two-stage smoothing (Zhai and Lafferty, 2002). In this thesis, we simply refer to this line of smoothing methods as background smoothing. The second term is a semantic mapping model that statistically maps topics contained in the document to terms. The mapping coefficient (λ) indicates the importance of the semantic mapping component in the mixture model. It is often empirically tuned.

The differences among semantic smoothing methods in literature mainly lie in three aspects: the representation of the topics, the estimation of semantic mapping for each topic $p(w|t_k)$, and the estimation of topic distributions in a document $p(t_k|d)$. The topic representations appearing in literature and our previous work include word (Berger and Lafferty, 1999), combined concept (Song and Bruza, 2003), cluster (Liu and Croft, 2001, multiword phrases (Zhou et al., 2007a and 2007b), ontology-based concept pairs (Zhou et al., 2006c), and topical themes (Wei and Croft, 2006). The estimates of topic distributions include uniform distribution (i.e., all topics are equally treated) (Bai et al., 2005a; Song and Bruza, 2003), maximum likelihood estimate (Berger and Lafferty, 1999; Zhou et al., 2006c, 2007a, 2007b), and topic modeling (Wei and Croft, 2006). The estimates of semantic mapping for each topic in literature are even more diversified, such as document-query pair-based machine translation (Berger and Lafferty, 1999), information flow (Song and Bruza, 2003), co-occurrence with semantic constraint (Cao et al., 2005), and co-occurrence–based mixture language model (Zhou et al., 2006c, 2007a, 2007b).

The topic signature language model is characterized with the following features. First, the topics are represented by any explicit text unit with topical information such as words, multiword phrases, and concepts. Second, the semantic mapping from a topic signature to individual words is based on co-occurrence data. Third, because topic signatures are explicitly extracted, the distribution of topic signatures in a document is estimated within the maximum likelihood estimate (MLE) principle.

The remainder of this chapter will be organized as follows. Section 3.2 introduces three types of topic signatures: words, multiword phrases, and ontological concepts. Section 3.3 shows the method of multiword phrase extraction. Section 3.4 presents the method of ontological concept extraction. Section 3.5 details the semantic mapping estimates.

3.2 Topic Signatures

We don’t have a strict definition of topic signatures. Any text unit that carries topical information and appears in more than one document can be considered a topic signature. For example, individual words, multiword phrases, ontological concepts, and concept pairs are good topic signatures. In this thesis, we compare and contrast the behavior and performance of three types of topic signatures. They are individual words, multiword phrases, and ontological concepts.

Words are the smallest unit in English text. It is straightforward to extract individual words from a document. The space of individual words is relatively small, and it is more computationally efficient to process words than multiword phrases or ontological concepts. However, a single word without context is often ambiguous. For example, the word “mouse” can be interpreted as either computer mouse or biological mouse.

Multiword phrases are defined as rigid noun phrases or collocations in this thesis. A multiword phrase contains two or more individual words which are adjacent to each other in sequence. It often begins with an adjective or a noun and ends with a noun. The semantics of a phrase usually has the following types.

- Organization: International Business Machine Corp.
- Person: George Bush, Ronald Regan
- Location: United States, Los Angeles
- Subject: space program, star wars

Apparently, multiword phrases are usually meaningful and length-variable, which contrasts a multiword phrase from an n-gram (e.g., bigram and trigram). An n-gram has a fixed length and is not necessary meaningful. However, both n-grams and multiword phrases are much more specific than individual words. For example, the phrase “space program” has a specific meaning while individual words “space” and “program” are quite general.

An ontological concept is a unique meaning in a particular domain. It represents a set of synonymous terms in the domain. For example, C0020538 is a concept about the disease of hypertension in the UMLS (Universal Medical Language System, http://www.nlm.nih.gov/research/umls) Metathesaurus; it also represents a set of synonymous terms including high blood pressure, hypertension, and hypertensive disease. Therefore, the concept-based topic signature representation helps to relieve the synonymy and polysemy problems in information retrieval and text mining.

According to whether the topic signature itself is context sensitive, the topic signature language model can be further divided into context-sensitive semantic smoothing (CSSS) and context-insensitive semantic smoothing (CISS). The word-based topic signature language model corresponds to CISS; multiword phrase and ontological concept-based topic signature language models belong to the category of CSSS.

3.3 Multiword Phrase Extraction

Unlike the extraction of bigram and trigram, multiword phrase extraction is not a trivial task. The multiword phrase of interest should be meaningful and should frequently occur in a collection or a domain. Thus, it is impossible to extract phrases from a single document without extra information. We developed a two-stage approach to the multiword phrase extraction. The first stage is to build a multiword phrase dictionary utilizing the statistics of the whole collection. The second stage is to extract multiword phrases from each document, based on the phrase dictionary.

We use a slightly modified version of Xtract (Smadja, 1993) to build a multiword phrase dictionary from a collection of documents. Xtract is designed to extract three types of collocations: predicative relations, rigid noun phrases, and phrasal templates. It begins with extracting significant bigrams using statistical techniques. It then expands 2-Grams to N-Grams, and finally it adds syntax constraint to the collocations. In Fagan's notion of phrases (Croft et al., 1991; Fagan, 1987), the phrases extracted by Xtract are constrained by both statistical and syntactic criteria. In the original version, two words are defined as a bigram if and only if they co-occur within a sentence and their lexical distance is less than five words. Because we are only interested in rigid noun phrases, the first word is limited to an adjective or a noun, and the second word must be a noun. Their distance threshold is set to four words, in our implementation.

Xtract uses four parameters, strength (k0), spread (U0), peak z-score (k1), and percentage frequency (T), to control the quantity and quality of the extracted phrases. In general, the bigger the value of those parameters, the higher quality and less quantity the phrases Xtract extracts. Smadja recommended a setting (k0, k1, U0, T) = (1, 1, 10, 0.75) to achieve good results. In the experiment, we set those four parameters to (1, 1, 4, 0.75). Xtract is an effective approach to the phrase extraction. The precision is about 80 %, which is good enough for our use in information retrieval and text mining—and is also very efficient. For example, it takes only two hours to build the dictionary from the AP89 collection (84,678 documents) using our Java version implementation; Annie (a named entity recognition component of GATE (Cunningham, 2002)) takes about twelve hours to recognize entities from the same collection.

After the phrase dictionary is built, we use a greedy search algorithm to extract all phrases that exist in the dictionary from each single document. To reduce the search space, we tag the part of speech of each sentence first and limit a multiword phrase candidate to a sequence of words that satisfies the following conditions:

(1) Starts with a noun, adjective, or number;

(2) ends with a noun or number;

(3) all words in the middle are a noun or number; and

(4) only the longest sequence is considered. For example, if ABC is a phrase, the subsequences AB and BC are ignored.

Example Sentence:
How the many changes in the former Soviet Union (now the Commonwealth of Independent States) will affect the future of their space program remains to be seen.

Word Index: *change, form, soviet, union, commonwealth, independent, state, affect, future, space, program, remain, see*

Multiword Phrase Index: *Soviet Union, independent state, space program*

Figure 3.1: The demonstration of multiword phrase extraction and indexing. Stop words are removed, and words are stemmed.

3.4 Ontological Concept Extraction

In general, the generic ontological concept extraction from free text is still in the infant stage. Biological concept extraction, however, has been extensively studied and has achieved acceptable accuracy. In this thesis, we only address the task of ontological concept extraction in the biomedical domain.

Dictionary-based biological concept extraction is still the state-of-the-art approach to large-scale biomedical literature annotation and indexing. The exact dictionary lookup is a very simple approach, but it always achieves low extraction recall because a biological term often has many variants, while it is impossible to collect and compile all of them into a dictionary. We propose a generic extraction approach—referred to as approximate dictionary lookup—to cope with term variations and we will implement it as an extraction system called MaxMatcher (Zhou et al., 2006d). The basic idea of this approach is to capture the significant words, instead of all words, to a particular concept. The new approach dramatically improves the extraction recall while maintaining the precision.

Example Sentence:

A recent epidemiological study (C0002783, research activity) revealed that obesity (C0028754, disease) is an independent risk factor for periodontal disease (C0031090, disease).

Word Index: *recent, epidemiological, study, research, activity, reveal, obesity, independent, risk, factor, periodontal, disease*

Concept Index: *C0002783, C0028754, C0031090*

Figure 3.2: The demonstration of concept extraction and indexing. Stop words are removed, and words are stemmed.

3.4.1 Approximate Dictionary Lookup

The earlier example of a biological (and ontological) concept, C0020538 from the UMLS Metathesaurus, is a concept about the symptoms of hypertension. It represents a set of synonymous terms including high blood pressure, hypertension, and hypertensive disease. In comparison with individual words, a concept is more meaningful; in comparison with multiword phrases, a concept well solves polysemy and synonymy problems (Zhou et al., 2006b). Therefore, using biological concepts can improve the performance of many applications such as large-scale biomedical literature retrieval, clustering, and summarization.

There are volumes of work addressing the issue of biological concept extraction in literature. However, most of them utilize the special naming conventions or patterns to identify a few types of biological concepts such as genes, proteins, and cells (Change et al., 2004; Collier et al., 2000; Fukuda et al., 1998; Song et al., 2004; Subramaniam et al., 2003; Tanabe and Wilbur, 2002; Zhou et al., 2004). In general, those approaches are designed for very specific types of concepts, and they work efficiently and effectively if the types of biological concepts have unique naming patterns. Many large-scale biomedical applications such as literature retrieval, clustering, and summarization, however, are interested in many rather than a few types of biological concepts, most of which do not have unique naming patterns. For example, UMLS covers 135 semantic types of biological concepts; a typical genomic IR system will index all of them.

The dictionary-based biological concept extraction is still the state-of-the-art approach to large-scale biomedical literature annotation and indexing (Rindfleisch et al., 2000; Zhou et al., 2005; Zhou et al., 2006b). Its major advantage over the pattern-based approach is that it not only recognizes names but also identifies unique concept identities. Among dictionary-based approaches, the exact dictionary lookup is the simplest one but always achieves low extraction recall because a biological term often has many variants, such as morphological variants, syntactic variants, and semantic variants (Chiang et al., 2005), while it is impossible to collect all of them from a dictionary.

To overcome the limitation of exact dictionary lookup, we introduce an approximate dictionary lookup technique. The basic idea of this technique is to capture significant words rather than all words in a concept name. For example, the word "gyrb" is obviously very significant to the concept "gyrb protein"; we treat it as a concept name even if the word protein is not present. So the problem is reduced to measuring the significance of any word to given concept names. In particular, we propose a relative significance score measure in this paper. Suppose a concept (*c*) has *n* concept names denoted as $s_1, \ldots, s_n$, respectively. Let *N(w)* denote the number of concepts whose variant names contain word *w*, and let w_{ji} denote the *i*-th word in the *j*-th variant name of the concept. The significance of *w* to the concept is defined as follows:

$$I(w,c) = \max\{I(w,s_j) \mid j \le n\} \qquad (3.2)$$

$$where:$$

$$I(w,s_j) = \begin{cases} 0 & w \notin s_j \\ \dfrac{1/N(w)}{\sum_i 1/N(w_{ji})} & w \in s_j \end{cases}$$

We use the UMLS Metathesaurus 2005AA version as the dictionary to train the significance score of each word to biological concepts containing that word. The UMLS Metathesaurus has a table called normalized string index, which records all normalized names of each concept. We remove normalized strings containing more than ten words and then use the remaining 2,573,244 strings to build the significance score matrix. A huge matrix, 509,170 rows (words) by 998,774 columns (concepts), is obtained. Because for each word, only a few concepts contain it, we use sparse matrix to make the storage and search more efficiently.

```
Find next starting word $t_s$
$k = 0$
$C = \{c \mid t_s \in T(c)\}$  /* $T(c)$ is the set of words appearing in names of concept $c$ */
For each $c \in C$  $S_c = I(t_s, c)$  /* $I(t_s, c)$ is the score of word $t_s$ to concept $c$ */
While  next word $t$ is not bounary word AND $k < skip$
    $N = \{c \mid t \in T(c) \wedge c \in C\}$
    IF $N = \varnothing$ Then  $k = k + 1$
    Else
        $C = N$
        For each $c \in C$  $S_c = S_c + I(t, c)$
    End If
Wend
$C = \{\text{c} \mid S_c > threshold \wedge c \in C\}$
If $|\text{C}| > 0$ Then
    return concept name and candidate concepts  $c \in C$
End If
```

Figure 3.3: The algorithm for extracting one concept name and its candidate concept IDs. The threshold is set to 0.95; the maximum number (skip) of skipped words is set to 1.

During the stage of extraction, we use a set of simple rules to identify the boundary of a concept candidate. A biological concept name should begin with a noun, a number, or an adjective should end with a noun or a number; it can not contain any boundary words including punctuations (except hyphen, period, and single quote), verbs, and conjunctions and prepositions (except "of"). In other words, whenever a boundary word is encountered, a candidate concept name reaches its end. The detailed searching algorithm is shown in Figure 3.3.

The major advantage of approximate dictionary lookup is that even if a concept name changes the word ordering a little bit, or inserts or deletes a couple of insignificant words, it can still be recognized. According to its definition, the significance score of a concept name should be equal to or greater than 1.0 if no word is missing. Thus, the threshold of significance score should be close to 1.0. If the threshold is too small our approach may falsely recognize “high pressure” as the concept name “high blood pressure”; if the threshold is too high, our approach may fail to recognize “gyrb” as “gyrb protein”. We found that a threshold of 0.95 gave good results for UMLS-based biological concept extraction. Our approach is able to recognize concept names with a couple of insertions such as articles, pronouns, and even nouns. The parameter skip controls the maximum number of insertions. We found that skip=1 gave good results.

The searching results are concept names and corresponding concept IDs. If two or more concept IDs are returned, we need to further figure out the meaning the extracted concept name refers to. The words surrounding the extracted concept name are often indicative to the meaning (Lesk, 1986). Thus, we take surrounding words (4 to the left, and 4 to the right) as the context and use the same algorithm as shown in Figure 3.3 to disambiguate the meaning of the extracted concept name if necessary.

3.4.2 Experimental Results

We evaluate both efficiency and effectiveness of the MaxMatcher. The effectiveness is evaluated on GENIA 3.02 corpus (http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA), which consists of 2,000 human-annotated PubMed abstracts. We compare the result of MaxMatcher with that of two other exact dictionary lookup systems, BioAnnotator

(Subramaniam et al., 2003) and ExactMatcher. The machine-extracted terms are compared with human annotations. Because human annotation is subjective, we provide exact-match-based evaluation and approximate-match-based evaluation, following the evaluation method in Subramaniam et al. For approximate-match, the human annotation should be the substring of the machine annotation, or the machine annotation is the substring of the human annotation.

The comparison among the three systems is presented in Table 3.1. For exact match, MaxMatcher performs significantly better than the other two systems in terms of both precision and recall. For approximate match, the precision of MaxMatcher is comparable to that of the other two systems, but the recall is significantly better than that of the other two.

Table 3.1: The comparison of MaxMatcher to ExactMatcher and BioAnnotator. BioAnnotator actually tested several configurations. But only the configuration with just dictionaries (i.e., exact dictionary lookup) is compared. BioAnnotator was evaluated on GENIA 1.1 (containing 670 human-annotated abstracts of research papers). The dictionary used for BioAnnotator also includes LocusLink and GeneAlias in addition to UMLS.

IE Systems	Exact Match Eva.			Approximate Match Eva.		
	Recall	Precision	F-score	Recall	Precision	F-score
MaxMatcher	57.73	54.97	56.32	75.18	71.60	73.35
ExactMatcher	26.63	31.45	28.84	61.56	72.69	66.66
BioAnnotator	20.27	44.58	27.87	39.75	87.67	54.70

For efficiency comparison, we downloaded the first 10,000 PubMed abstracts published in 2005 and counted the time for annotating these abstracts by MaxMatcher and ExactMatcher, respectively. It takes 510 seconds for MaxMatcher to annotate all 10,000 PubMed abstracts; the average annotation speed is 19.6 abstracts per second. ExactMatcher is faster. It only costs 320 seconds to process those abstracts; the average annotation speed is 31.3 abstracts per second. However, ExactMatcher consumes much more memory (765 megabytes) than MaxMatcher (362 megabytes).

3.5 Semantic Mapping Estimates

The semantic mapping estimate is a task to map a topic signature to each single word in the vocabulary. Formally, denoting ***W*** as the word vocabulary and t_k as the topic signature, the task is to estimate the parameters $p(w_i \mid t_k)$ which satisfy $\sum_{i=1}^{|W|} p(w_i \mid t_k) = 1$.

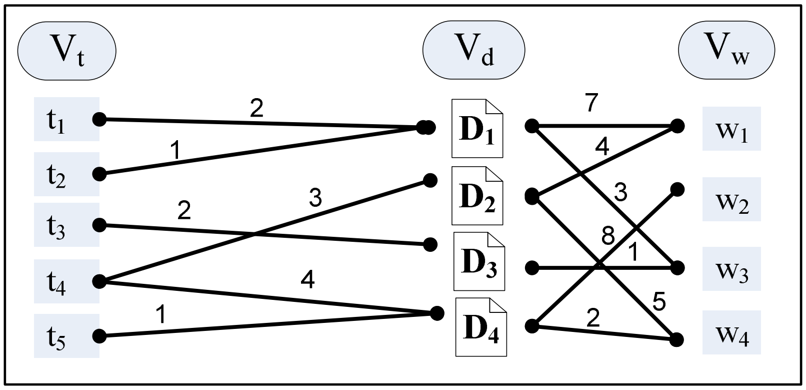


Figure 3.4: Illustration of document indexing. V_t, V_d, and V_w are topic signature set, document set, and word set, respectively. The number on each line denotes the frequency of corresponding topic signature or word in the document.

For each topic signature t_k, we can obtain a set of documents (D_k) containing the signature (see Figure 3.4). Intuitively, we can use the document set D_k to approximate the semantic mapping from t_k to single-word features in the vocabulary. If all words appearing in D_k center on the topic signature t_k, we can simply use maximum likelihood estimate, and the problem is as simple as frequency counting. Some words, however, address topics corresponding to other topic signatures, and some are background words of the whole collection. Therefore, we employ a mixture language model (Zhai and Lafferty, 2001b), as described in equation 3.3, to remove noise, that is, words are generated either by the topic signature mapping model or by the background collection model.

$$p(w|D_k) = (1-\alpha)p(w|t_k) + \alpha p(w|C) \qquad (3.3)$$

When this mixture model is used for text generation, it is unknown what model a word is exactly generated by. It is instead a hidden variable. But the chance of selecting either model is known. Here α is the coefficient accounting for the chance of using the background collection model to generate words. The log likelihood of generating the document set D_k is then:

$$\log p(D_k) = \sum_w c(w, D_k) \log p(w|D_k) \qquad (3.4)$$

Here $c(w, D_k)$ is the document frequency of term w in D_k, that is, the cooccurrence count of w and t_k in the whole collection. The parameters $p(w|t_k)$ can then be estimated by the EM algorithm (Dempster et al., 1977) with the following update formulas:

$$\hat{p}^{(n)}(w) = \frac{(1-\alpha)p^{(n)}(w \mid t_k)}{(1-\alpha)p^{(n)}(w \mid t_k) + \alpha p(w \mid C)} \qquad (3.5)$$

$$p^{(n+1)}(w \mid t_k) = \frac{c(w, D_k)\hat{p}^{(n)}(w)}{\sum_i c(w_i, D_k)\hat{p}^{(n)}(w_i)} \qquad (3.6)$$

As usual, the maximum likelihood estimator initializes the EM algorithm. With respect to the setting of the background coefficient α, the larger α is, the more specific the trained parameters are. When α closes to one, the majority of terms get extremely small probability values. Our study shows a large α (e.g., 0.9) fits for applications such as query expansion, in which only a few of the most important terms are expanded, and a medium α (e.g., 0.5) is good for applications such as text classification and clustering. We also truncate terms with extremely small mapping probabilities for two purposes. First, with smaller mapping space, class model smoothing becomes much more efficient. Second, we assume terms with extremely small probability are noise (i.e., not semantically related to the given topic signature). In detail, we disregard all terms with mapping probability less than 0.0005 and renormalize the mapping probabilities of the remaining terms.

Space [CISS]

space 0.245; shuttle 0.057; launch 0.053; flight 0.042; air 0.035; program 0.031; center 0.030; administration 0.026; develop 0.025; like 0.023; look 0.022; world 0.020; director 0.020; plan 0.018; release 0.017; problem 0.017; work 0.016; place 0.016; mile 0.015; base 0.014…;

Program [CISS]

program 0.193; washington 0.026; congress 0.026; administration 0.024; need 0.024; billion 0.023; develop 0.023; bush 0.020; plan 0.020; money 0.020; problem 0.020; provide 0.020; writer 0.018; d 0.018; help 0.018; work 0.017; president 0.017; house .017; million 0.016; increase 0.016…;

Space Program [CSSS]

space 0.101; program 0.071; NASA 0.048; shuttle 0.043; astronaut 0.041; launch 0.040; mission 0.038; flight 0.037; earth 0.037; moon 0.035; orbit 0.032; satellite 0.031; Mar 0.030; explorer 0.028; station 0.028; rocket 0.027; technology 0.026; project 0.025; science 0.023; budget 0.023…;

(a) Examples from news collection AP89

Breast [CISS]:

breast 0.312; cancer 0.195; tumor 0.056; carcinoma 0.050; woman 0.048; node 0.028; metastasis 0.026; estrogen 0.025; chemotherapy 0.024; lymph 0.020; invasive 0.019; survival 0.016; malignant 0.015;…

Cancer [CISS]:

cancer 0.329; tumor 0.080; breast 0.070; carcinoma 0.055; survival 0.034; chemotherapy 0.033; metastasis 0.030; prostate 0.027; lung 0.026; stage 0.024; therapy 0.023; advance 0.022; node 0.021; risk 0.019;…

Breast Cancer [CSSS]:

breast 0.040; malignant 0.028; tumor 0.022; cancer 0.021; benign 0.011; mcf-7 0.006; carcinoma 0.006; mammary 0.005; neoplasm 0.004; estrogen 0.004; herceptin 0.004; her2 0.004; estrone 0.004; lobular 0.004;…

(b) Examples from Medline

Figure 3.5: The demonstration of context-sensitive and context-insensitive topic signature semantic mappings.

Our estimation of semantic mappings is significantly different from the statistical translation model (Berger and Lafferty, 1999) in two aspects. First, the translation model requires a large amount of document-query pairs, which is very difficult to obtain in practice. Instead, we use co-occurrence data, which are much cheaper to collect. Second, the translation model takes words as topic signatures and is unable to incorporate contextual information into the translation procedure. Our approach can use context-sensitive topic signatures such as multiword phrases and ontological concepts. Consequently, the semantic mapping is more specific. From two examples shown in Figure 3.5 we can see that multiword phrase mapping (e.g., space program) and ontological concept mapping (e.g., breast cancer) are quite coherent and specific. However, if we estimate semantic mappings for its constituent terms separately, both contain mixed topics and are fairly general. Some terms such as "NASA", "astronaut", "moon," "satellite," "rocket," and "Mar," which are highly correlated to the subject of "space program," do appear in the result of phrase mappings, but in neither of the word mappings. Some terms such as "mcf-7," "estrogen," "herceptin," and "her2," which is highly correlated to the subject of "breast cancer," do appear in the result of ontological concept mappings, but in neither of the word mappings.

CHAPTER 4: SEMANTIC SMOOTHING IN INFORMATION RETRIEVAL

4.1 Introduction

The language modeling approach to information retrieval (IR), initially proposed by Ponte and Croft (1998), has been popular with the IR community in recent years because of its solid theoretical foundation and promising empirical retrieval performance. In essence, this approach centers on the document model estimation and the query generative likelihood calculation according to the estimated model. However, it is challenging to estimate an accurate document model due to the sparsity of training data. On one hand, because the query terms may not appear in the document, we need to assign a reasonable nonzero probability to the unseen terms. On the other hand, we need to adjust the probability of the seen terms to remove the effect of the background collection model or even irrelevant noise. Thus, the core of the language modeling approach to IR is to smooth document models. Zhai and Lafferty (2001a and 2002) propose several effective background smoothing techniques that interpolate the document model with the background collection model.

A potentially more significant and effective method is semantic smoothing that incorporates synonym and sense information into the language model (Lafferty and Zhai, 2001a). Berger and Lafferty (1999) adopt semantic smoothing into the language model by statistically mapping document terms onto query terms using a translation model trained from synthetic document-query pairs. The translation model is context insensitive (i.e., it is unable to incorporate sense and contextual information into the language model);

the resulting translation may be mixed and fairly general. For example, the term “mouse” without context may be translated to both “computer” and “cat” with high probabilities. Jin et al. (2002) and Cao et al. (2005) present two other ways to train the translation probabilities between individual terms, but their approaches still suffer the same context-insensitivity problem as Berger and Lafferty. Thus, it is urgent that a framework is developed to semantically smooth document models within the language modeling (LM) retrieval framework.

In this chapter, we propose a topic signature language model for context-sensitive document smoothing. A document is decomposed into a set of weighted topic signatures, and then those topic signatures are mapped into individual terms for the purpose of document expansions. We define a topic signature as either an ontology-based concept or an automated multiword phrase. Because a concept or a multiword phrase itself contains contextual information and its meaning is usually unambiguous, the mapping from topic signatures to individual terms should have higher accuracy and result in better retrieval performance, compared to the semantic translations between single words. For example, “mouse” in conjunction with “computer” could be a topic signature; the signature might be translated to “keyboard” with a high probability but to “cat” with a low probability, because of additional contextual constraints.

The new smoothing method is tested on collections from two different domains in order to show its robustness. The extraction of concepts needs domain ontology. Thus we evaluate the effectiveness of concepts on TREC Genomics Track 2004/2005. The extraction of multiword phrases does not need any external human knowledge and can be applied to any public domains. Therefore we test the effectiveness of multiword phrases on TREC Disks 1, 2, and 3, which contain news articles from several sources including AP, SJM, and WSJ. The experimental results show that significant improvements are obtained over the two-stage language model (Zhai and Lafferty, 2002) as well as the language model with context-insensitive semantic smoothing (CISS).

The remainder of this chapter is organized as follows. In Section 4.2, we describe in details the method of context-sensitive document smoothing. Section 4.3 shows the experimental results on TREC 2004/2005 Genomics Track collections, where topic signatures are implemented as ontology-based concepts. Section 4.4 shows the experimental results on TREC Disks 1, 2, and 3, where multiword phrases are used as topic signatures. Section 4.5 concludes this chapter.

4.2 Context-Sensitive Document Smoothing

In this section, we first define two types of topic signatures and introduce the extraction algorithms. Second, we describe the document expansion (smoothing) using topic signature language models. Last, we discuss the scalability and complexity of the estimation of the topic signature language model.

4.2.1 Context-Sensitive Topic Signatures

The implementation of topic signatures plays a crucial role in our context-sensitive semantic smoothing approach. First, the topic signature must be context sensitive, and thus it should contain at least two terms, unless word sense is adopted. Second, constituents of a topic signature should have syntactic relation. Otherwise, we cannot count their frequency in a document, and it becomes difficult to estimate their distributions. Third, it should be easy and efficient to extract topic signatures from texts. Following these criteria, we recommend two types of topic signatures. One is the ontological concept, and the other is the multiword phrase. In this subsection, we formally define these two types of topic signatures and briefly introduce the corresponding extraction algorithms.

In our previous work (Zhou et al., 2006c), we implemented topic signatures as concept pairs inspired by Harabagiu and Lacatusu's (2005) topic representations. Formally, a topic signature is defined with two order-free components as in $t(w_i, w_j)$, where w_i and w_j are two concepts related to each other syntactically and semantically. Because two concepts in a pair help to determine the context for each other, the meaning of a concept pair is often unambiguous, and its semantic mapping to individual concepts is very specific and accurate. The combination of two concepts, however, causes a large vocabulary space that makes it inefficient to index large collections. The distribution of concept pairs is also quite sparse, and thus it is difficult to obtain sufficient data for many concept pairs in order to estimate their mapping probabilities to individual concepts. Aware of the unambiguousness of a single concept in ontology, we simply use ontological concepts as topic signatures.

In general, the extraction of concepts from texts is still a challenging problem. Fortunately, in the domain of biology and medicine, a large ontology called UMLS was developed, which makes the task of concept extractions possible. The extraction of biological concepts is a hot topic in bioinformatics, and a survey of those methods can be found in “Mining Knowledge from Text Using Information Extraction” (Mooney and Bunescu, 2005). Most approaches segment a sequence of words into phrases but do not further map the identified phrases into concepts. For this reason, we adopt MaxMatcher (Zhou et al., 2006d), a dictionary-based biological concept extraction tool, for UMLS concept extractions.

In order to increase the extraction recall while maintaining the precision, MaxMatcher uses approximate matches between the word sequences in text and the concepts defined in a dictionary or ontology, such as the UMLS Metathesaurus. It outputs concept names as well as unique IDs representing a set of synonymous concepts. The unique concept IDs are used as an index in our experiments. In the example shown in Figure 3.2, the underlined phrases are extracted concept names followed by the corresponding concept ID and semantic type. The details of the algorithm for MaxMatcher can be found in our previous work (Zhou et al., 2006d). MaxMatcher has been evaluated on the GENIA corpus. The precision and recall reached 71.60% and 75.18%, respectively, using approximate match criterion.

The use of phrases has a long history in information retrieval. A typical method for utilizing phrases will identify phrases within queries (e.g., “star wars”, “space program”), scan documents to identify query phrases, and score the document if it contains query phrases (Pickens and Croft, 2000). The recognition of query phrases within documents can be done in one of the following three manners (Pickens and Croft, 2000).

- Boolean: it is also called conjunctive phrases (Croft et al., 1991). All subterms of a query phrase co-occur in a document.

- Adjacent: Exact same form as the query phrase.
- Proximity: All subterms of a query phrase occur in close proximity in a document.

In this thesis, we utilize multiword phrases in a different manner. We treat phrases frequently occurring in a given collection as topic signatures and try to find a set of individual words to represent the topic signature (the multiword phrase). Then we can expand a document language model by statistically mapping topic signatures into query terms (individual words). For this purpose, we only identify multiword phrases within documents. The definition of phrase in this paper is roughly equivalent to the definition of query phrases in traditional phrase models. It is a sort of rigid noun phrase or collocation. It contains two or more individual words that are adjacent to each other in sequence. It often begins with an adjective or a noun and ends with a noun. We use a slightly modified version of Xtract (Smadja, 1993) to extract phrases in documents.

In the experiment, we also tried two other types of multiword phrases in order to increase phrase coverage. One is named entities (person, location, and organization) identified by GATE (Cunningham, 2002). The other is WordNet noun phrases (Miller, 1995). However, the extra phrases did not bring further improvement of IR performance. A possible explanation is that both GATE entities and WordNet noun phrases are purely "syntactic" phrases, and those extra phrases (not extracted by Xtract) are often infrequent in our testing collections. In our language model, the infrequent phrases (topic signature) result in little effect on document expansions.

Table 4.1: Examples of context-sensitive topic signature mappings. The three multiword phrases are automatically extracted from the collection of AP89 by Xtract. We only list the top twenty topical words for each phrase. It is worth noting that the word "third" is removed from indexing as a stop word, and thus it does not appear in the mapping result of the third phrase.

space program		**star wars**		**third world debt**	
Term	Prob.	Term	Prob.	Term	Prob.
space	0.101	star	0.088	debt	0.072
program	0.071	war	0.066	Brady	0.039
NASA	0.048	missile	0.06	loan	0.038
shuttle	0.043	strategy	0.051	world	0.038
astronaut	0.041	defense	0.051	treasury	0.037
launch	0.040	nuclear	0.043	bank	0.035
mission	0.038	space	0.034	Nicholas	0.034
flight	0.037	initialize	0.033	debtor	0.030
earth	0.037	Pentagon	0.032	trillion	0.027
moon	0.035	weapon	0.031	reduction	0.027
orbit	0.032	bomber	0.031	forgive	0.025
satellite	0.031	budget	0.028	monetary	0.025
Mar	0.030	stealthy	0.025	Mexico	0.025
explorer	0.028	program	0.025	economy	0.023
station	0.028	spend	0.024	billion	0.023
rocket	0.027	armed	0.023	reduce	0.022
technology	0.026	fiscal	0.022	burden	0.022
project	0.025	Reagan	0.021	lend	0.021
science	0.023	cut	0.021	creditor	0.021
budget	0.023	Bush	0.019	secretary	0.020

4.2.2 Document Model Smoothing

Suppose we have indexed all documents in a given collection C with both individual words and topic signatures. The probability of mapping a topic signature t_k to any individual term w, denoted as $p(w|t_k)$, is also given. Then we can easily obtain a document model below:

$$p_t(w \mid d) = \sum_k p(w \mid t_k) p_{ml}(t_k \mid d) \qquad (4.1)$$

The likelihood of a given document generating the topic signature t_k can be estimated with

$$p_{ml}(t_k \mid d) = \frac{c(t_k, d)}{\sum_i c(t_i, d)} \qquad (4.2)$$

where $c(t_i, d)$ is the frequency of the topic signature t_i in a given document *d*.

We refer to the above model as a semantic mapping model. As we discussed in the previous subsection, the semantic mapping from context-sensitive topic signatures to individual terms would be very specific. Thus, the smoothed (expanded) document models will be more accurate. However, not all topics in a document can be expressed by topic signatures (e.g., multiword phrases). Take the example of AP88-90. A document in this collection contains 179 unique words but only contains 32 multiword phrases on average (see Table 4.2). If only the semantic mapping model is used, there will be serious information loss. A natural extension is to interpolate the semantic mapping model with a unigram language model. We use the two-stage method (Zhai and Lafferty, 2002) to smooth the unigram language model:

$$p(Q \mid D) = \prod_{q \in Q} \{(1-\gamma) \frac{tf(q, D) + \mu p(q \mid C)}{|D| + \mu} + \gamma p(q \mid C)\} \qquad (4.3)$$

where $p(q|C)$ is the collection background model. γ and μ are two coefficients for tuning. We also refer to this smoothed unigram model as simple language model (SLM) or baseline language model (BLM).

The final document model for retrieval use is described in equation 4.4. It is a mixture model with two components: a simple language model and a semantic mapping model.

$$p_{bt}(w|d) = (1-\lambda)p_b(w|d) + \lambda p_t(w|d) \qquad (4.4)$$

The mapping coefficient (λ) is to control the influence of two components in the mixture model. With training data, the mapping coefficient can be trained by optimizing a retrieval performance measure such as average precision. In the experiments in this thesis, we train the optimal mapping coefficient on one collection and then apply the learned mapping coefficient to other collections.

4.2.3 Scalability and Complexity

In comparison to the simple language models (Zhai and Lafferty, 2002) and traditional probabilistic language models, such as Okapi (Robertson, 1995), the topic signature language model needs the following extra computational costs: (1) the extraction of topic signatures from documents in offline mode, (2) the estimation of topic models for each topic signature in offline mode, and (3) document model expansions based on topic signature mappings in online mode. Fortunately, the additional computation is scalable and, its complexity is acceptable in practice. Furthermore, the issue of scalability and complexity is significantly improved over the statistical translation model (Berger and Lafferty, 1999) and the LDA-based document model (Wei and Croft, 2006).

The extraction of topic signatures is time consuming compared with individual term extraction. However, it does not cause a serious problem because it can be executed in the offline and incremental mode. In the experiment, the dragon toolkit (Zhou et al., 2007c) is used for document indexing. The dragon toolkit implements a Java version of Xtract (Smadja, 1993) for multiword phrase extraction. Take the example of indexing the AP collection in Disk 1, 2, and 3 (about 240,000 news articles) on a Linux server. It takes about fifteen minutes to index individual terms and three hours to index topic signatures (multiword phrases). From this example, we can see that the indexing time for topic signatures is acceptable as an offline task.

Table 4.2: Average numbers of unique words and unique topic signatures per document in six collections.

Collection	avg. # of unique words per doc	avg. # of unique topic signatures per doc
Genomics 2004	71.3	39.2
Genomics 2005	75.2	37.6
AP89	180.1	31.8
AP88-89	178.6	31.7
WSJ90-92	196.6	35.6
SJMN91	164.2	25.3

The estimation of topic models is highly computation-intensive. In general, the parameter space is in proportion to the number of documents in the corpus, the size of vocabulary, and the number of topics; the computational complexity is in proportion to the number of documents, the number of topics, and the number of iterations for convergence. Therefore, the estimation algorithms proposed in “Information Retrieval as Statistical Translation” (Berger and Lafferty, 1999) and “LDA-based document models for ad-hoc retrieval" (Wei and Croft, 2006) are not very scalable and are time-consuming for large collections. For example, the estimation of the LDA model for the AP collection using Gibbs sampling (please refer to Wei and Croft, 2006, for detailed settings) costs about seventy-two hours, whereas our approach uses only forty-five minutes to estimate topic models for all topic signatures. Our approach estimates topic models for each topic signature separately, which dramatically reduces the parameter space and makes the model converge with fewer iteration steps. Thus, our estimation approach increases the scalability and reduces the complexity.

The online document model expansion based on topic models is computationally intensive because it involves the summation of translation probabilities as shown in equation 4.1. The complexity is in proportion to the number of topics for a document. The number of topics is equal to the number of unique terms in the statistical translation model (Berger and Lafferty, 1999), the number of latent topics in LDA-based models (Wei and Croft, 2006), and the number of unique topic signatures in the topic signature language model, respectively. As shown in Table 4.2, the number of topic signatures is significantly less than the document length as well as the number of latent topics in the LDA model (e.g., the optimal number of topics is 800 in Wei and Croft, 2006) in typical testing collections, and thus our approach has the lowest complexity during the stage of online document model expansions.

4.3 Experiments with Ontological Concepts

4.3.1 Evaluation Metrics and Baseline Models

Following the convention of TREC, we use the mean average precision (MAP) as the major performance metric and the overall recall at 1000 documents as a supplemental metric. The noninterpolated average precision is defined as:

$$\frac{1}{|\mathrm{Rel}|}\sum_{D\in \mathrm{Rel}}\frac{|\{D'\in \mathrm{Rel}, r(D')\le r(D)\}|}{r(D)} \qquad (4.5)$$

where *r(D)* is the rank of document *d*, and *Rel* is the set of relevant documents for a query *Q*. By averaging the noninterpolated average precision across all queries of a collection, we obtain the MAP for the collection.

In the experiment, we use the two-stage language model (SLM) (Zhai and Lafferty, 2002) as the first baseline. The exact formula for the two-stage model is described in equation 4.3. To show how strong the baseline is, we also compare the baseline to the famous Okapi model (Robertson, 1993). The exact formula for the Okapi model is shown below:

$$Sim(Q,D)=\sum_{q\in Q}\left\{\frac{tf(q,D)\log(\frac{N-df(q)+0.5}{df(q)+0.5})}{0.5+1.5\frac{|D|}{avg\_dl}+tf(q,D)}\right\} \qquad (4.6)$$

Where:

tf(q, D) is the term frequency of *q* in document *D*.

df(q) is the document frequency for *q*.

avg_dl is the average document length in the collection.

The major difference between the statistical translation model (Berger and Lafferty, 1999) and the proposed topic signature language model is that the latter incorporates the contextual information into the document model expansions (smoothing). Thus, it is very natural to further compare the context-sensitive semantic smoothing (CSSS) to the context-insensitive semantic smoothing (CISS). Because it is difficult to obtain a large number of real query-document pairs, we use word-word co-occurrence data to train a context-insensitive version of mapping probabilities in the experiment. The parameter estimation algorithm is the same as that for the context-sensitive version (i.e., the semantic mapping from topic signature to individual words). The retrieval model is still the mixture of a two-stage language model and a semantic mapping model as described in equation 4.4. But the mapping component is formulated slightly differently:

$$p_t(w|d) = \sum_k p(w|w_k) p_{ml}(w_k|d) \qquad (4.7)$$

It statistically maps each individual word, instead of context-sensitive topic signature, in a document onto query terms.

4.3.2 Testing Collections

Our current implementation of concept-based topic signature extraction needs domain ontology. For this reason, we validate our context-sensitive semantic smoothing method on genomic collections, because UMLS can be used as the domain ontology for this area. The testing collections are TREC Genomic Track 2004 (Hersh et al., 2004) and 2005 (Hersh et al., 2005). The original collection is a ten-year subset of Medline abstracts and contains about 4.6 million abstracts. We only used the subcollection (i.e., the human relevance-judged document pool, with 42,251 documents for 2004 and 35,474 documents for 2005) for our experiment. The ad hoc retrieval tasks of the two tracks include fifty topics (queries). The statistics of the testing collections are shown in Table 4.3.

Table 4.3: The descriptive statistics of genomics track 2004 and 2005 collections

Collections	Word	Concept	Rel./Doc	Q.Len/Q.#
Genomics 2004	92,362	65,257	8,268/42,251	6.4/50
Genomics 2005	80,168	57,879	4,584/35,474	6.0/49

4.3.3 Document Indexing and Query Processing

We index all documents with UMLS-based concepts and individual words. For each document, we record the frequency count of each topic signature (i.e., UMLS concept) and individual words, as well as the basic statistics. For each topic signature and individual words, we record their frequency count in each document and the basic statistics. For word indexing, stop words are removed, and each word is stemmed. For topic signatures appearing in ten or more documents, we estimate their topic models (i.e., semantic mapping probabilities) using the EM algorithms.

The query formulation is fully automated. The extraction of query terms (individual words) from topic descriptions is the same as the process of document indexing. In TREC 2004 Genomics Track, a topic was described in three sections: title, information need, and context. The information provided by section of context is a little noisy. Our pilot study showed that the baseline (both Okapi and the two-stage language model) using context section performed much worse than the one without context. For this reason, we only use the title section and information-need section in the experiment. In TREC 2005 Genomics Track, query #135 was removed because it contained no relevant document.

As stated in "A Hidden Markov Model Information Retrieval System" (Miller et al., 1999), the query terms in the title section are clearly more important than those in the remaining sections. For this reason, we weight query terms according to the sections from which they are extracted. Following the method proposed by Miller et al., we optimize the weight of different sections by maximizing the MAP of the baseline retrieval model. The optimal weights for the title section and the information-need section are 1.0 and 0.2, respectively. In Table 4.4, 4.5, and 4.6, the sign † indicates the initial query is weighted.

4.3.4 Effect of Document Smoothing

We set parameters γ and μ in the two-stage language model to 0.05 and 200, respectively, because the language model achieves the best performance with this configuration. To give readers the sense of how good the baseline language model is, we also report the performance of the Okapi retrieval model in Table 4.4. The Okapi model is slightly better than the two-stage model, but roughly these two models are comparable to each other.

The mapping coefficient (λ) in the topic signature language model is optimized by maximizing the MAP on TREC Genomics Track 04 using an unweighted query. The learned optimal value is 0.3; we apply this learned value to the other two collections. The result is shown in Table 4.5. In order to validate the significance of the improvement, we also run a paired-sample t-test. As expected, the topic signature language model outperforms the two-stage language model in terms of average precision and overall recall at the significance level of 0.01 on both TREC04 and TREC05.

Table 4.4: Comparison of the simple language model (SLM) to the Okapi model on genomics collections. The sign† indicates the initial query is weighted.

Collection	Recall			MAP		
	2SLM	Okapi	Change	2SLM	Okapi	Change
TREC04	6544	6847	+4.6%	0.352	0.369	+4.8%
TREC04†	6680	6869	+2.8%	0.384	0.370	-3.7%
TREC05	4093	4193	+2.4%	0.265	0.270	1.9%

Table 4.5: The comparison of simple language model (SLM) to the topic signature language model (i.e., context-sensitive semantic smoothing, CSSS). The signs ** and * indicate the improvement is statistically significant according to the paired-sample t-test at the level of p<0.01 and p<0.05, respectively. The sign† indicates the initial query is weighted.

Collections		SLM	CSSS	Change
TREC04	MAP	0.352	0.422	+19.9%**
	Recall	6544	7279	+11.2%**
TREC04†	MAP	0.384	0.446	+16.2%**
	Recall	6680	7395	+10.7%**
TREC05	MAP	0.265	0.322	+21.5%**
	Recall	4093	4291	+4.8%**

To see the robustness of the topic signature language model, we change the settings of the mapping coefficient. The variance of the mean average precision (MAP) with the mapping coefficient λ is shown in Figure 4.1. When the mapping coefficient ranges from 0 to 0.9, the topic signature language model always performs better than the baseline on the three collections. This shows the robustness of the new model. More interestingly, the best performance is achieved when the mapping coefficient is around 0.3 for all three curves; after that point, the performance is downward. A possible explanation is that the extracted topic signatures do not capture all points of the document, but the baseline language model captures those missing points. For this reason, when the influence of the semantic mapping model is too high in the mixture model, the performance is downward and even worse than that of the baseline. Therefore, if we can find a better topic signature representation for documents and queries, or we can refine the extraction of topic signatures, the IR performance might be further improved.

Table 4.6: The variance of MAP with the change of the mapping coefficient (λ), which controls the influence of the mapping component in the mixture model.

Mapping Coefficient	Genomics_04 weighted	Genomics_04 unweighted	Genomics_05
0	38.41	35.17	26.51
0.1	43.22	40.32	29.68
0.2	44.24	41.66	31.54
0.3	44.60	42.18	32.24
0.4	44.22	42.20	32.05
0.5	43.59	41.48	32.1
0.6	42.76	40.52	31.82
0.7	41.77	39.62	31.34
0.8	40.58	38.33	30.66
0.9	39.27	36.41	29.68
1	28.45	27.60	25.57

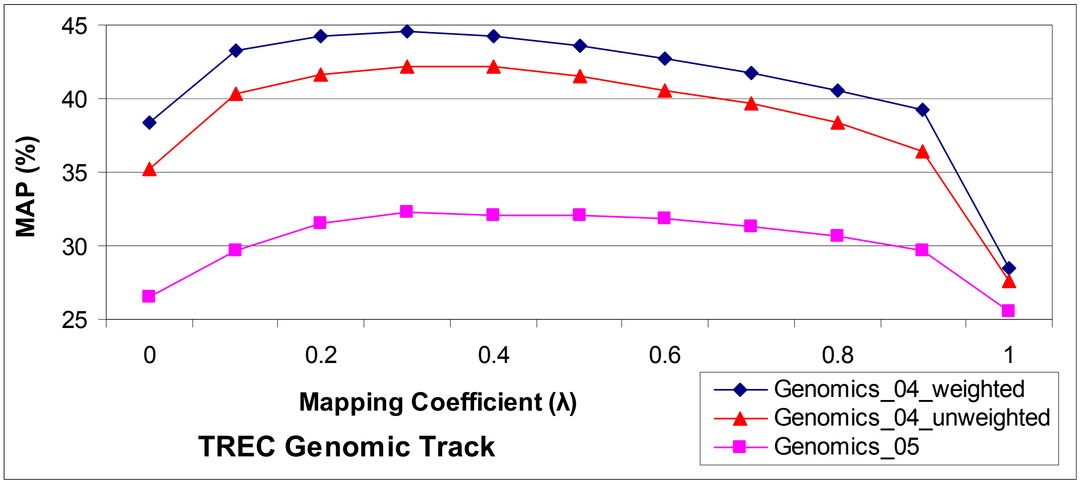


Figure 4.1: The variance of MAP with the change of the mapping coefficient (λ), which controls the influence of the topic signature language model.

4.3.5 Context Sensitive vs. Context Insensitive

Basically, the context-insensitive semantic smoothing (CISS) is based on the word-word mapping as did in (Berger and Lafferty, 1999; Gao et al., 2005; Jin et al., 2002; Lafferty and Zhai, 2001). The comparison of CISS to CSSS is presented in Table 4.7. For each collection, we tune the mapping coefficient (λ) to maximize the MAP. The optimal λ is about 0.3 for all three collections. Firstly, we can see that CISS significantly outperforms the two-stage language model on all three collections. The gain of the CISS model over the baseline language model is consistent with the conclusions of previous work, such as (Berger and Lafferty, 1999; Gao et al., 2005; Jin et al., 2002; Lafferty and Zhai, 2001). However, CISS is slightly less effective than CSSS, as expected.

Secondly, the improvement of CSSS over CISS seems not much on genomics track. On genomics track 2005, there is almost no improvement. A possible explanation is that most document terms are biological terms such as protein, gene and cell names. Compared to general terms such as words in news articles, the meaning of biological and medical terms (e.g., p53, brca1 and orc1) is more consistent even if without additional contextual constraints. Thus, the word-word mapping itself was very specific and accurate in Genomics collections.

Table 4.7: Comparison of the context-sensitive semantic smoothing (CSSS) to the context-insensitive semantic smoothing (CISS) on MAP. The rightmost column is the change of CSSS over CISS.

Collections		SLM	CISS	vs. SLM	CSSS	vs. CISS
Genomics 2004	MAP	0.352	0.408	+15.9%**	0.422	+3.4%*
	Recall	6544	7176	+9.7%**	7279	+1.4%*
Genomics 2004†	MAP	0.384	0.432	+12.5%**	0.446	+3.2%*
	Recall	6680	7359	+10.2%**	7395	+0.5%
Genomics 2005	MAP	0.265	0.322	+21.5%**	0.322	+0.0%
	Recall	4093	4283	+4.6%**	4291	+0.2%

4.4 Experiments with Multiword Phrases

4.4.1 Testing Collections

In this section, we evaluate the effectiveness of automated multiword phrases as topics signatures. Compared to ontological concepts, the extraction of multiword phrases does not need any external human knowledge and can be applied to any public domain. The model is validated on six TREC ad hoc collections from disk 1, disk 2, and disk 3. We select these collections for three reasons. First, these collections are well studied and many published results are available to compare. Second, the content of these collections is all about general news stories on which the Xtract is supposed to work very well on the automated phrase extraction. Third, compared to the vocabulary in genomic collections, the vocabulary of news stories is more ambiguous and thus the context-sensitive semantic smoothing is supposed to take the advantage over the context-insensitive semantic smoothing. The descriptive statistics of these testing collections are shown in Table 4.8.

Table 4.8: The descriptive statistics of six news collections from TREC disk 1, disk 2, and disk 3.

Collections	Word	Phrase	Rel./Doc	Q.Len/Q.#
AP89/1-50	145,349	114,096	3,301/84,678	3.4/47
AP88&89/51-100	204,970	127,736	6,101/164,597	3.4/49
AP88&89/101-150	204,970	127,736	4,822/164,597	4.0/50
WSJ90-92/101-150	135,864	75,687	2,049/74,520	3.8/48
WSJ90-92/151-200	135,864	75,687	2,041/74,520	4.6/49
SJMN91/51-100	173,727	95,986	2,322/90,257	3.4/48

4.4.2 Document Indexing and Query Processing

We build two separate indices, word index and phrase index, for each collection. For word indexing, each document is processed in a standard way. Words are stemmed (using porter-stemmer) and stop words are removed. We use a 319-word stop list compiled by van Rijsbergen. Xtract (Smadja, 1993) is employed to extract multiword phrases from documents. For phrases appearing in ten or more documents, we estimate their mapping probabilities to single-word terms.

The query formulation is fully automated. For each collection, we remove all queries (topics) which contain no relevant documents. Early TREC topics are often described in multiple sections including title, description, narrative, and concept. As many other studies did (Bai et al., 2006; Lafferty and Zhai, 2001; Liu and Croft, 2001; Wei and Croft, 2006; Zhai and Lafferty, 2001b), we use only the section of title. The extraction of query terms from topic descriptions is the same as the process of word indexing. That is, each topic is tokenized and stemmed and stop words are removed. The average length of queries and total number of queries for each collection is listed in Table 4.8.

4.4.3 Effect of Document Smoothing

We set the parameters γ and μ in the two-stage language model to 0.5 and 750, respectively in the experiment because almost all collections achieve the optimal MAP at this configuration. Interestingly, the Okapi model and the two-stage language model have similar retrieval performance in the experiment as shown in Table 4.9. This is also a kind of indication that both baseline models are well tuned.

Table 4.9: The comparison of the simple language model to the Okapi model on six news collections.

Collection/Topics	Recall			MAP		
	SLM	Okapi	Change	SLM	Okapi	Change
AP89/1-50	1621	1618	-0.2%	0.187	0.187	0.0%
AP88-89/51-100	3428	3346	-2.4%	0.252	0.239	-5.2%
AP88&89/101-150	3055	3087	+1.0%	0.219	0.220	+0.5%
WSJ90-92/101-150	1510	1488	-1.5%	0.239	0.249	+4.2%
WSJ90-92/151-200	1612	1624	+0.7%	0.314	0.304	-3.2%
SJMN91/51-100	1350	1348	-0.1%	0.190	0.184	-3.2%

Table 4.10: The effect of document expansions based on phrase-word semantic mapping (i.e., context-sensitive semantic smoothing, CSSS).

Collection/Topics		SLM	CSSS	Change
AP89 1-50	MAP	0.187	0.206	+10.2%**
	Recall	1621	1748	+7.8%**
AP88-89 51-100	MAP	0.252	0.288	+14.3%**
	Recall	3428	3771	+100%*
AP88-89 101-150	MAP	0.219	0.246	+12.3%**
	Recall	3055	3445	+12.8%**
WSJ90-92 101-150	MAP	0.239	0.256	+7.1%**
	Recall	1510	1572	+4.1%*
WSJ90-92 151-200	MAP	0.314	0.334	+6.5%**
	Recall	1612	1620	+0.5%
SJMN91 51-100	MAP	0.190	0.208	+9.5%**
	Recall	1350	1472	+9.0%**

The mapping coefficient (λ) in the topic signature language model is optimized by maximizing the MAP on the collection of AP89 Topic 1-50. The optimal value is 0.3 and we then apply this learned coefficient to other five collections. Interestingly, all collections achieve the best performance when the mapping coefficient is around 0.3. We then compare the result of the topic signature language model to the two-stage language model. The comparison is shown in Table 4.10. In order to validate the significance of the improvement, we also run paired-sample t-test. The incorporation of phrase-word mapping improves both MAP and overall recall over the baseline model on all six collections. Except the recall on the collection of WSJ 90-92 Topic 151-200, the improvements over the two-stage language model are all statistically significant at the level of $p<0.05$ or even $p<0.01$. Considering the baseline model is already very strong, we think the topic signature language model is promising to improve IR performance.

To see the robustness of the topic signature language model, we also change the settings of the mapping coefficient. The variance of MAP with the mapping coefficient λ is plotted in Figure 4.2. In a wide range from 0 to 0.6, the topic signature language model always performs better than the baseline on all six collections. This shows the robustness of the model. For all six curves in Figure 4.2, the best performance is achieved when the mapping coefficient is 0.3; after that point, the performance is downward. A possible explanation is that the extracted topic signatures (multiword phrases) do not capture all points of the document, but the two-stage language model captures those missing points. For this reason, when the influence of the mapping model is too high in the mixture model, the performance is downward and even worse than that of the baseline.

Table 4.11: The variance of MAP with the change of the mapping coefficient (λ), which controls the influence of the mapping component in the mixture model.

Mapping Coefficient	AP89 1-50	AP88-89 51-100	AP88-89 101-150	WSJ90-92 101-150	WSJ90-92 151-200	SJMN91 51-100
0	18.7	25.2	21.9	23.9	31.4	19.0
0.1	19.6	27.7	23.8	25.0	32.8	20.4
0.2	20.2	28.5	24.5	25.5	33.3	20.9
0.3	20.6	28.8	24.6	25.6	33.4	20.8
0.4	20.5	28.4	24.2	25.3	33.3	20.5
0.5	20.2	27.7	23.3	24.7	32.9	20.0
0.6	19.6	26.5	22.1	23.7	32.4	19.2
0.7	19.2	25.0	21.1	22.5	30.6	18.5
0.8	17.8	23.5	19.8	21.2	28.7	17.6
0.9	16.3	21.5	18.2	19.4	26.6	16.3
1	7.7	15.9	11.2	9.9	12.50	10.8

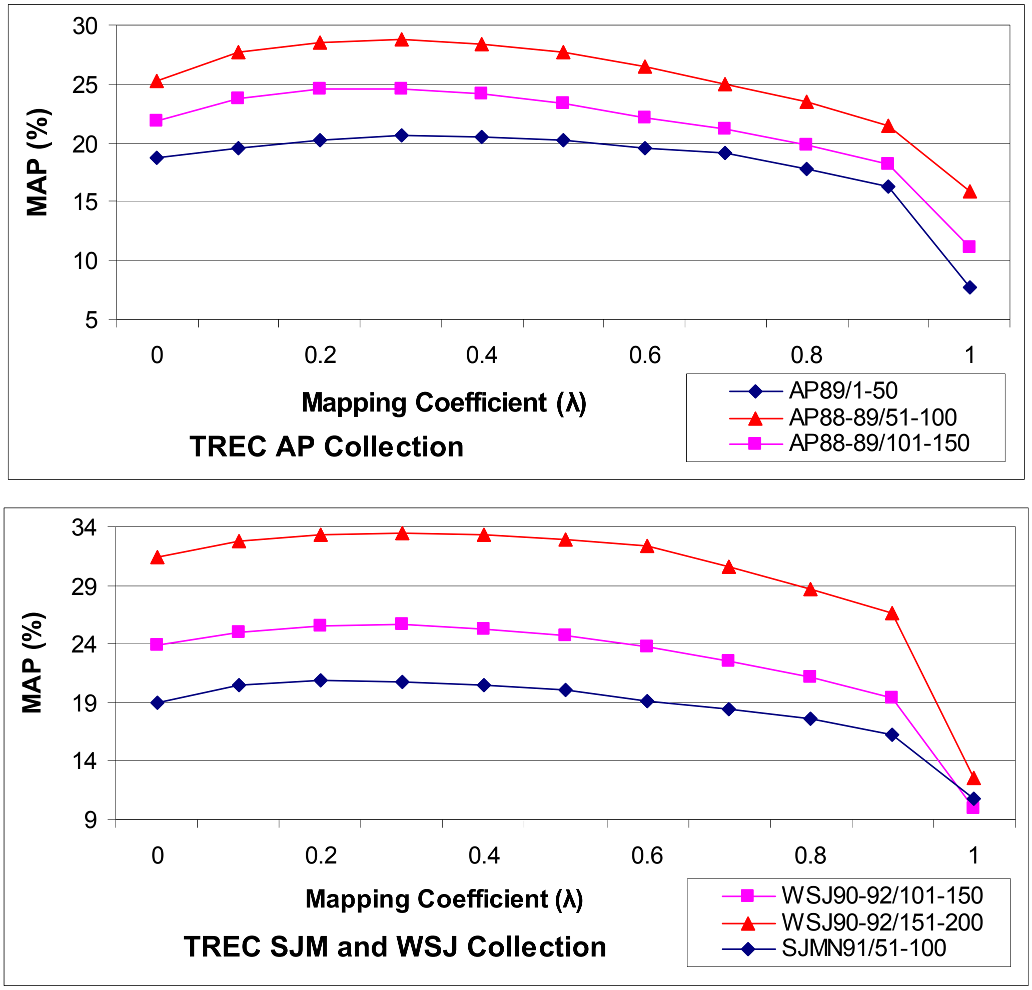


Figure 4.2: The variance of MAP with the mapping coefficient (λ), which controls the influence of the context-sensitive mapping component in the mixture language model.

4.4.4 Context Sensitive vs. Context Insensitive

In news articles, many terms are ambiguous; a term may have a different meaning in different context. Thus, the word-word mapping may be fairly general and contain mixed topics. The phrase-word mapping solves this problem since multiword phrases have very specific meaning and are mostly unambiguous.

Table 4.12: Comparison of the context-sensitive semantic smoothing (CSSS) to the context-insensitive semantic smoothing (CISS) on MAP of six news collections. The rightmost column is the change of CSSS over CISS.

Collections		TSLM	CISS	vs. TSLM	CSSS	vs. CISS
AP89 1-50	MAP	0.187	0.195	+4.3%*	0.206	+5.6%
	Recall	1621	1730	+6.7%*	1748	+1.0%
AP88-89 51-100	MAP	0.252	0.272	+7.9%	0.288	+5.9%*
	Recall	3428	3735	+9.0%*	3771	+1.0%
AP88-89 101-150	MAP	0.219	0.235	+7.3%**	0.246	+4.7%
	Recall	3055	3237	+6.0%*	3445	+6.4%*
WSJ90-92 101-150	MAP	0.239	0.244	+2.1%	0.256	+4.9%*
	Recall	1510	1568	+3.8%**	1572	+0.3%
WSJ90-92 151-200	MAP	0.314	0.324	+3.2%	0.334	+3.1%
	Recall	1612	1646	+2.1%*	1620	-1.6%
SJMN91 51-100	MAP	0.190	0.199	+4.7%*	0.208	+4.5%
	Recall	1350	1427	+5.7%**	1472	+3.2%

The comparison of the context-sensitive semantic smoothing (CSSS) to the context-insensitive semantic smoothing (CISS) is shown in Table 4.12. For each collection, we tune the mapping coefficient (λ) to maximize the MAP of CISS. The optimal λ is about 0.1 for all six collections, which is smaller than the optimal value for CSSS (λ=0.3). It is also a kind of indication that the word-word mapping is much noisier than the phrase-word mapping. From the experimental results, we can first see that CISS greatly outperforms the two-stage language model, and most of the improvements are statistically significant. Second, the CSSS has considerable gain over the CISS especially on the measure of MAP.

In addition, the CSSS is computationally more efficient than the CISS. The CSSS is based on the phrase-word mapping; the CISS is based on the word-word mapping. As shown in Table 4.2, an average document in testing collections contains about 180 unique words but only about 30 unique multiword phrases. In other words, the CSSS is six times faster than the CISS for the construction of co-occurrence data as well as the document model expansions (smoothing).

4.4.5 Other Types of Phrases

The different types of phrases may have different impact on retrieval performance. Fagan reported significant improvement on some collections using statistical phrases, but none with syntactic phrases, in his thesis (1987). In this thesis, we used phrases with both syntactic and statistical constraints extracted by Xtract and obtained very positive results. An interesting question is then raised: Can other types of phrases (e.g., WordNet phrases and Named Entities) still get positive results with the topic signature language model?

To test this idea, we add WordNet noun phrases and named entities including person, organization, and location to the document index and see if the IR performance is further improved or even decreased. WordNet noun phrases are manually selected phrases. The named entities are automatically extracted by GATE (Cunningham, 2002) according to purely syntactic rules. Thus, neither of them is constrained by statistical criteria. Take the example of the AP89 collection. Before adding extra phrases, the collection has 114,096 phrases. After adding WordNet noun phrases and named entities, the number of phrases is increased by about 50,000. However, the increase of phrase coverage does not make any improvement on IR performance. The other five collections are in a similar case.

Examining the extra noun phrases more closely, we find that most of those phrases are infrequent in the testing collections. The majority of phrases frequently occurring in the collection are already extracted by Xtract. Those infrequent phrases will have little effect on the document model expansions and thus have no effect on retrieval performance. Therefore, in order to make the topic signature (phrase) language model effective, we should use phrases that frequently occur in the collection or that are constrained by statistical criteria.

4.5 Conclusions and Future Work

In this chapter, we proposed a topic signature language model for ad hoc text retrieval. This new model decomposed a document into a set of weighted context-sensitive topic signatures and then mapped those topic signatures into individual query terms. Because the topic signature itself contained contextual information, the document model expansion based on topic signatures would be more accurate, compared to the document model expansion based on context-insensitive term mapping proposed in previous work (e.g., Berger and Lafferty, 1999; Gao et al., 2004; Jin et al., 2002), and thus improved the retrieval performance.

We implemented two types of topic signatures in this paper. When domain-specific ontology is available, ontological concepts can be used as topic signatures. Otherwise, automated multiword phrases are an alternative. We evaluated the effectiveness of ontological concepts on TREC Genomics Track 2004 and 2005 and the effectiveness of multiword phrases on TREC Ad hoc Track Disk 1, Disk 2, and Disk 3. The topic signature language model significantly outperformed the two-stage language model on all collections. We further implemented a context-insensitive version of semantic smoothing. It has the same framework as the topic signature language model, but the document model expansion (smoothing) is based on the context-insensitive word-word mapping rather than the context-sensitive signature-word mapping. As expected, it is less effective than the context-sensitive semantic smoothing, though it does achieve significant improvement over the simple language model.

The topic signature language is the linear interpolation of the simple language model and the semantic mapping model. It is required to set the mapping efficient that controls the influence of the semantic mapping component in the mixture model. It is somewhat ad hoc in nature. Fortunately, the experiments showed the robustness of the model. When the mapping coefficient took different values in a wide range (0-0.9 for ontological concepts and 0-0.6 for multiword phrases), the topic signature language model always performed better than the baseline. More interestingly, all collections achieved the best MAP at the same setting (the mapping coefficient is 0.3). This means it is feasible to train the optimal mapping coefficient on one collection and then apply the learned coefficient to other collections in practice.

We also found two factors that would affect the effectiveness of the topic signature language model. One is the degree of the ambiguity of terms in the collection. If terms (e.g., in news collections) are very ambiguous, the topic signature model (i.e., context-sensitive semantic smoothing) can gain much more advantage over the context-insensitive semantic smoothing. The other factor is the occurrence frequency of the topic signatures in the collection. If the topic signatures infrequently occur in the collection, the model has little effect on improving the IR performance.

This chapter made the following contributions. First, we presented a new document representation (i.e., representing a document as a set of weighted topic signatures and terms). The new representation could be applied to other retrieval, summarization, and text classification tasks. Second, we proposed an EM-based method to estimate the semantic relationships between context-sensitive topic signatures and single-word terms by simply using co-occurrence data, and then we formalized the approach to document expansions based on topic signature mapping. Third, we empirically proved the superiority of the context-sensitive semantic smoothing over the context-insensitive semantic smoothing and the simple background smoothing.

Probabilistic topical models such as pLSI (Hoffman, 1999) and LDA (Blei et al., 2003) also take the context into account and, thus, can handle the word polysemy problem. In this chapter, we analyzed their computing complexity in the setting of IR and concluded that these two models were computationally less efficient than the topic signature language model in the stage of offline topic model estimation, as well as the stage of online document model smoothing. However, the comparison of the effectiveness of the three models on retrieval tasks is still unclear. It should be interesting to have a comprehensive comparative study on these three models in future with respect to their efficiency and effectiveness for ad hoc text retrieval.

How to optimize the mixture weights of the topic signature language model remains an open issue. In this chapter, we empirically tuned a fixed mapping coefficient on a training data set and achieved good results. Ideally, the mapping coefficient should be conditioned on each document because the relative information provided by topic signatures varied with documents.

CHAPTER 5: SEMANTIC SMOOTHING IN TEXT CLASSIFICATION

5.1 Introduction

The task of text classification is to assign one or multiple predefined class labels to a text. It has been a hot research topic with the rapid increase of text in digital form, such as web pages, newswires, and scientific literature. In past decades, a large number of algorithms, including naïve Bayes (McCallum and Nigam, 1998), k-nearest neighbor (Yang and Pedersen, 1997), support vector machines (Joachims, 1998), boosting, decision trees (Quinlan, 1986) and neural network (Wiener et al., 1995), have been developed for text classifications. Although some previous studies have shown that SVM outperformed other approaches in many categorization applications, naïve Bayes is still widely used in practice, mostly likely because of its efficient model training and good empirical results.

Naïve Bayesian classifiers face a common issue called data sparsity problem, especially when the size of training data is too small. Because of data sparseness, some terms appearing in testing documents may not appear in training documents of some classes. To prevent zero probability, one has to use smoothing techniques that assign a reasonable nonzero probability to those unseen terms. Laplace smoothing, which simply adds one count to all terms in the vocabulary, is frequently used for Bayesian model smoothing. But it proves to be not effective in many applications (Jelinek, 1990).

The study of language model smoothing has been a hot topic in the community of information retrieval (IR), with the increasing popularity of the language modeling approach to IR. Zhai and Lafferty have proposed several effective smoothing methods including Jelinek-Mercer, Dirichlet, absolute discount (Zhai and Lafferty, 2001a) and two-stage smoothing (Zhai and Lafferty, 2002) to smooth unigram language models. Because all of these approaches are based on a background collection model, we refer to all of them as background smoothing in this thesis. However, a potentially more effective smoothing method is what may be referred to as semantic smoothing, which incorporates context and sense information into the language model. A motivating example for semantic smoothing is that the document containing the term “auto” should return for the query “car” because both terms are semantically related. Following this intuitive idea, several semantic smoothing approaches (Berger and Lafferty, 1999; Wei and Croft, 2006; Zhou et al., 2006c) have been proposed for language modeling IR.

The success of semantic smoothing in text retrieval inspires us to apply it into Bayesian text classification. We propose in this chapter a topic-signature–based semantic smoothing method to address the aforementioned data sparsity problem. The idea of our semantic smoothing method is to extract explicit topic signatures (e.g., words, multiword phrases, and ontological concepts) from training documents and then statistically map them into single-word features. For example, considering the semantics (background knowledge) of the phrase “space program”, we may correctly assign a testing document about rocket launch to a given category whose training documents never explicitly present the topic of rocket launch, but which contain many instances of “space program”. The definition of topic signatures will be given later in the chapter.

The idea of using multiword phrases or n-grams for text classification is not new. However, to the best of our knowledge, this is the first time it has been used for smoothing purposes in the setting of text classification. The majority of those works (Bai et al., 2005b; Bloehdorn and Hotho, 2004; Lewis 1990; Yetisgen-Yildiz and Pratt, 1997) utilized its distinguishing power for classification, with the philosophy that a match of a multiword phrase or n-grams between testing documents and training documents gives more confidence regarding testing documents' membership than a single-word match. Peng et al. (2004) and Shen et al. (2006) built n-gram and n-multigram language models to get more accurate text classifiers. Neither of them used multiword phrases or n-grams to relieve the data sparsity problem. Actually, the distribution of multiword phrases and n-grams is always much sparser than unigrams. When the training document set is extremely small, multiword phrases or n-gram features are too sparse to serve as good features for classification.

Feature (word) clustering is also a common technique for text classification (Al-Mubaid and Umair, 2006; Baker and McCallum, 1998). It groups similar words together and uses word clusters as document features. Such representation accounts for semantic relationships between words and brings higher classification accuracy. Meanwhile, it reduces the high dimensionality. However, its notion and implementation are different from the proposed semantic smoothing approach. The former focuses on document representation, whereas the latter aims at smoothing the language models for different classes.

We implement our semantic smoothing method using the dragon toolkit (Zhou et al., 2007c) and conduct comprehensive experiments on three collections, OHSUMED, the Los Angeles Times (LATimes), and 20-Newsgroups (20NG). The experiments show that when the size of training documents is small, the Bayesian classifier with semantic smoothing not only outperforms the Bayesian classifiers with background smoothing and Laplace smoothing, but also beats the state-of-the-art active learning classifiers (Nigam et al., 2000) and SVM classifiers (Joachims, 1998).

In summary, we make four main contributions in this chapter. First, we propose two new types of topic signature (i.e., multiword phrases and ontology-based concepts), both of which include contextual information, making the semantic mapping more specific and accurate. Second, aware of the existence of large amounts of co-occurrence data, we use a co-occurrence–based algorithm to estimate semantic mappings, which dramatically reduces the cost of obtaining semantic knowledge. Third, we empirically prove that semantic smoothing is more effective than background smoothing and Laplace smoothing for Bayesian text classifiers. Last, we compare the behaviors of three types of topic signatures (i.e., word, multiword phrases, and ontology-based concepts) when they are used as intermediates for semantic smoothing during the task of text classification.

The rest of the chapter is organized as follows: Section 5.2 describes the details of the semantic smoothing method for Bayesian text classifiers. Section 5.3 introduces the datasets and protocols for evaluation. Section 5.4 presents the experimental results. Section 5.5 shows the result of parameter tuning. Section 5.6 concludes the chapter.

5.2 Naïve Bayes with Semantic Smoothing

The naïve Bayesian classifier is widely used for text classification because of its efficient model training and good empirical results. Naïve Bayes (NB) is a maximum a posterior

(MAP) classifier. The assignment of class label to a given document can be formulated as:

$$C(d) = \arg\max_{c_i} p(c_i)p(d \mid c_i) \qquad (5.1)$$

The first term is the class prior. Two commonly used prior distributions are uniform distribution and empirical distribution. In this paper, we use empirical distributions, which can be estimated by the formula below:

$$p(c_i) = \frac{1 + N(c_i, D)}{|C| + |D|} \qquad (5.2)$$

where $N(c_i, D)$ denotes the number of documents with class label c_i in collection D. The second term in equation 5.1 is the conditional probability of the document given the category. Because NB classifiers assume all words are independent of each other, the conditional probability can be further decomposed into the product of individual feature probabilities:

$$p(d \mid c_i) = \prod\nolimits_{k=1}^{|d|} p(w_{d_i,k} \mid c_i) \qquad (5.3)$$

Now the problem is reduced to estimating the class model, that is, the distribution over features for a given class. There are several variants of naïve Bayesian classifiers such as multivariate Bernoulli model and multinomial mixture model, with respect to class models. Previous studies have shown that multinomial mixture model achieves the best accuracy on text classification (Nigam et al., 2000). For this reason, all experiments in this chapter are based on multinomial mixture mode.

The simplest implementation of a multinomial class model is the maximum likelihood estimate with Laplace smoothing (Lidstone, 1920; Johnson, 1932; Jeffreys, 1948). That is,

$$p_l(w \mid c_i) = \frac{1 + N(w, c_i)}{|V| + \sum_w N(w, c_i)} \qquad (5.4)$$

where *N(w, c_i)* is the occurrence frequency of word *w* in all training documents of class c_i, and *V* is the vocabulary of words. Obviously, Laplace smoothing assigns an equal prior probability to all unseen words, which does not make much sense for real textual data. To solve this problem, we introduce two other more effective smoothing approaches, background smoothing and semantic smoothing.

Language modeling has been a hot research topic in the community of IR in recent years. Several smoothing methods based on the statistics from the whole collection have been empirically proven to be effective for IR (Zhai and Lafferty, 2001a; Zhai and Lafferty, 2002). We refer to this line of smoothing methods as background smoothing in this thesis. The Jelinek-Mercer (Jelinek, 1990; Zhai and Lafferty, 2001a) is such a smoothing method. In the setting of NB classifiers, it interpolates a unigram class model with the collection background model, controlled by the parameter β as shown in equation 5.5:

$$p_b(w \mid c_j) = (1-\beta) p_{ml}(w \mid c_j) + \beta p(w \mid D) \qquad (5.5)$$

where $p_{ml}(w \mid c_j)$ is the unigram class model with maximum likelihood estimate and $p_b(w \mid c_j)$ denotes the unigram class model with background smoothing. In this chapter, β is empirically set to 0.5.

The semantic smoothing approach statistically maps topic signatures in all training documents of a class into single-word features. However, as pointed out in previous studies (Wei and Croft, 2006; Zhou et al., 2006c), the mere use of topic signature semantic mapping may lead to information loss. After all, much information is represented by the unigram model. Thus, we linearly interpolate the semantic mapping component with a simple language model as described in equation 5.5, and the class model ends with the following formula:

$$p_s(w|c_i) = (1\text{-}\lambda) p_b(w|c_i) + \lambda \sum_k p(w|t_k) p(t_k \mid c_i) \quad (5.6)$$

where $p_s(w \mid c_i)$ stands for the unigram class model with semantic smoothing, t_k denotes the k-th topic signature, and $p(t_k \mid c_i)$ is the distribution of topic signatures in training documents of a given class, which can be computed via maximum likelihood estimates. The mapping coefficient λ is to control the influence of the semantic mapping component in the mixture model. If the mapping coefficient is set to zero, the class model becomes a simple language model. If it is set to one, the class model becomes a semantic mapping model. Please refer to Section 5.5 regarding the optimization of the mapping coefficient. The remaining problem is how to compute the probability of semantic mappings from topic signatures t_k to single-word feature w, which has been addressed in Chapter 3.

The training of Bayesian classifiers with semantic smoothing takes some extra computational cost over the traditional approaches. First, it needs to extract multiword phrases or ontological concepts from testing or training documents. The complexity of extraction is in proportion to the length of the document. Second, it maps topic signatures (e.g., phrases and concepts) to words. In practice, we map each topic signature to around two hundred significant words, instead of all words in the vocabulary. Thus, the overall complexity would be *O(200n)*, where *n* is the number of extracted topic signatures from the training documents.

5.3 Datasets and Protocols

5.3.1 Evaluation Methodology

The evaluation metrics are precision (*P*), recall (*R*), and F1-measure. F1 is the harmonic average of precision and recall. The formula to compute F1 is $2P \times R/(P+R)$. F1 score can be computed by individual category first and then be averaged over categories, or globally computed over all categories. The former is called macro-F1; the latter is called micro-F1 (Yang and Liu, 1999). If data are evenly distributed over different categories, micro-F1 and macro-F1 are usually similar. However, for highly skewed data, the micro-F1 is often dominated by a few large categories, while the macro-F1 is the better metric to reflect the classification performance on rare categories.

In the experiment, we compare the classification performance upon the change of training data size on each collection. For a given percentage of training data (e.g., 1%), we conduct ten random runs and then average the performance of all runs. Each run has a random partition of training data and testing data controlled by a random seed. For fair comparisons, the partition of training data and testing data is the same to different configurations on all runs in the comparative study.

Feature selection is one of the frequently used techniques for text classification. The appropriate selection of subfeature space can dramatically improve the performance for many classifiers, including the NB classifier using Laplace smoothing. In all experiments, we choose CHI feature selector (Yang and Pedersen, 1997) for NB and manually tune it to the best result. However, the feature selection has no effect on background smoothing and semantic smoothing. Therefore, we do not apply feature selection for these two smoothing methods in the experiments.

Table 5.1: The descriptive statistics of three text classification collections, 20NG, LATimes, and OHSUMED

Dataset Name	20NG	LATimes	OHSUMED
# of categories for classification	20	10	14
# of indexed docs	19,997	21,623	7,400
# of topic signatures	10,902	10,414	28,857
# of signatures per doc	9	8	61
# of unique signatures per doc	7	7	33
# of words in corpus	133,277	63,510	27,676
# of words per doc	157	99	116
# of unique words per doc	91	75	69

5.3.2 Datasets

We evaluate the NB classifier with semantic smoothing on three collections: 20-Newsgroups (20NG), Los Angeles Times (LATimes), and OHSUMED. 20NG is collected from twenty different Usenet newsgroups, and the data are relatively noisy. LATimes contains news articles. OHSUMED consists of scientific abstracts collected from Medline, an on-line medical information database. We selected these three collections because of their diverse sources.

20NG has twenty classes, each of which contains about one thousand articles. A total of 19,997 articles are indexed. LATimes of TREC Disk 5 represents a sampling of approximately 40% of the articles published by the Los Angeles Times in the two year period from January 1, 1989, to December 31, 1990. There are total of 111,084 articles distributed in twenty-two sections, for example, Financial, Entertainment, Sports, et cetera. We consider the section an article sits in to be the ground truth of memberships. The articles in the top fifteen sections are selected for indexing. If a section contains more than 2,000 articles, only the first 2,000 are selected. The articles with a length of less than 200 bytes are excluded. The remaining 21,623 articles were finally indexed. The top ten sections are Metro, Sports, Financial, Late Final, Entertainment, Foreign, National, View, Letters, and Calendar. The OHSUMED corpus contains 13,929 Medline abstracts of the year 1991, each of which was assigned with one or multiple labels out of twenty-three cardiovascular diseases categories. Excluding abstracts with multiple labels, we indexed the remaining 7,400 abstracts.

Table 5.2: The distributions of documents over categories in three text classification collections. The number in the parenthesis following the class label is the number of selected documents for this class.

Collections	Selected Class/Doc	Details
20NG	20/19,660	Atheism (995), guns (994), crypt (993), space (991), religion misc (991), motorcycles (989), politics misc (989), hockey (985), pc hardware (983), ms-windows (982), baseball (982), mideast (982), windows.x (981), electronics (979), autos (978), christian (978), mac hardware (976), med (976), for sale (970), graphics (966)
LATimes	10/17,916	Letters (2001), National (1995), Financial (1991), Foreign (1975), Entertainment (1947), Sports (1923), Metro (1848), Late Final (1776), Calendar (1392), View (1068)
OHSUMED	14/6,657	Cardiovascular Diseases (1175), Neoplasms (1030), Pathological Conditions, Signs and Symptoms (796), Nervous System Diseases (557), Disorders of Environmental Origin (553), Immunologic Diseases (410), Digestive System Diseases (354), Urologic and Male Genital Diseases (342), Nutritional and Metabolic Diseases (323), Respiratory Tract Diseases (250), Skin and Connective Tissue Diseases (233), Musculoskeletal Diseases (223), Bacterial Infections and Mycoses (216), Female Genital Diseases and Pregnancy Complications (195)

5.3.3 Text Processing

For each document, we first identify single-word features from its title and body. The other sections of a document including metadata are ignored. Stop words are removed, and all words are stemmed. Second, we extract context-sensitive topic signatures. Third, we estimate semantic mappings (i.e., the probability of mapping a topic signature to single-word features) for all topic signatures appearing in five or more documents, we use the algorithm proposed in Chapter 3. The parameter α is set to 0.5.

Multiword phrases are extracted from 20NG and LATimes by a modified version of Xtract (Smadja, 1993). Xtract uses four parameters, strength (*k0*), spread (*U0*), peak z-score (*k1*), and percentage frequency (*T*), to control the quantity and quality of the extracted phrases. In general, the bigger those parameters, the higher quality and less number of phrases Xtract produces. In the experiment, we set those four parameters to 1, 1, 4, and 0.75. The detail of the implementation is available in "Semantic Smoothing for Model-based Document Clustering" (Zhang et al., 2006a).

UMLS concepts are extracted from OHSUMED by MaxMatcher (Zhou et al., 2006d). MaxMatcher is a dictionary-based biological concept extraction tool. In order to increase the extraction recall while maintaining the precision, MaxMatcher uses approximate matches between the word sequences in text and the concepts defined in a dictionary or ontology, such as the UMLS Metathesaurus. It outputs concept names as well as unique IDs representing a set of synonymous concepts. MaxMatcher has been evaluated on the GENIA corpus. The precision and recall reached 71.60% and 75.18%, respectively, using approximate match criterion.

5.4 Experiment Results

5.4.1 Semantic Smoothing vs. Lap and Bkg

We first evaluate the context-sensitive semantic smoothing (CSSS) with 1% data for training. For 20NG corpus, 1% training data means each class has about 10 of 1,000 documents for training. The class distribution is highly skewed on the other two collections. For OHSUMED corpus, the largest class and the smallest class use about 12 of 1175 documents and 2 of 195 for training, respectively. The corpus of LATimes is more balanced; the training documents are 20 and 10 for the largest class and the smallest, respectively. The performance (Micro-F1 and Macro-F1) is shown in Table 5.3. CSSS significantly outperformed Laplace smoothing and background smoothing in terms of both Micro-F1 and Macro-F1 on all three collections at the significance level of p<0.01, according to the paired-sample t-test with freedom of nine (i.e., ten runs for each collection). This verifies our hypothesis that semantic smoothing is more effective than Laplace smoothing and background smoothing for Bayesian text classifiers when the number of training documents is small and the data are sparse.

Table 5.3: Comparisons of context-sensitive semantic smoothing (CSSS) to Laplace smoothing (Lap) and background smoothing (Bkg). 1% of documents are used for training and the remaining 99% for testing. The parameter λ for CSSS is 0.4. The symbols ** and * indicate the change is significant according to the paired-sample t-test at the level of p<0.01 and p<0.05, respectively.

(a) The result of micro-F1

Collection	Lap	Bkg	CSSS	vs. Lap	vs. Bkg
OHSUMED	0.352	0.372	0.413	**17.3%	**10.9%
20NG	0.427	0.526	0.613	**43.7%	**16.6%
LATimes	0.525	0.538	0.581	**10.7%	**7.9%

(b) The result of macro-F1

Collection	Lap	Bkg	CSSS	vs. Lap	vs. Bkg
OHSUMED	0.205	0.280	0.351	**71.0%	**25.2%
20NG	0.421	0.523	0.609	**44.6%	**16.4%
LATimes	0.492	0.513	0.554	**12.5%	**7.8%

Taking a closer look at the results, we have two interesting findings. One is the different improvement pattern on Micro-F1 and Macro-F1. The 20NG corpus achieved similar improvements over Lap and Bkg in terms of Micro-F1 and Macro-F1, whereas Macro-F1 was improved much more than Mciro-F1 on the other two collections. The classes in 20NG corpus are almost in equal size, and thus it has similar effect on Micro-F1 and Macro-F1. The class labels in the other two collections are highly skewed, and as we pointed out earlier, the result of Micro-F1 is dominated by the performance of large categories. However, for the metric of Macro-F1, the performance of each category is treated equally regardless the size of the category. This means, from the fact that Macro-F1 obtained much more improvement than Micro-F1, we can conclude that semantic smoothing is especially effective for small classes. It is reasonable because small classes contain too few training examples, and data sparsity is a serious problem.

The second finding is that the magnitude of improvement of semantic smoothing over two other smoothing methods depends on the dataset. Take the example of the improvement of Micro-F1 (Semantic smoothing vs. Laplace smoothing). On the corpus of 20NG, CSSS achieved the biggest improvement of 43.7%, but it only achieved 10.7% and 17.3% on LATimes and OHSUMED. The 20NG corpus has a large vocabulary space of 133 thousand words, whereas the average number of unique words per document in the three collections is similar. In other words, 20NG corpus is sparser; many words in the testing document do not appear in the training documents. The semantic smoothing is very effective in solving such a sparse data problem by statistically mapping topic signatures to single word features. This explains why semantic smoothing obtained as much as 43.7% on 20NG.

Table 5.4: Comparisons of CSSS to Lap and Bkg. 33% of documents are used for training. The parameter λ for CSSS is tuned to the best.

(a) The result of micro-F1

Collection	Lap	Bkg	CSSS	vs. Lap	vs. Bkg
OHSUMED	0.660	0.667	0.665	0.8%	-0.2%
20NG	0.771	0.802	0.820	*6.3%	*2.2%
LATimes	0.728	0.726	0.729	0.2%	*0.4%

(b) The result of macro-F1

Collection	Lap	Bkg	CSSS	vs. Lap	vs. Bkg
OHSUMED	0.626	0.639	0.640	*2.2%	0.2%
20NG	0.756	0.787	0.816	*7.9%	*3.6%
LATimes	0.708	0.696	0.700	**-1.2%	**0.5%

To further validate the finding that the more sparse the data, the more effective the semantic smoothing, we conduct another experiment with as many as 33% data for training. With so many training documents, sparsity will not be a serious problem for most categories in the three testing collections. The results are shown in Table 5.4. As expected, the semantic smoothing in the case of 33% training data is much less effective than in the case of 1% training data. In terms of Micro-F1, the semantic smoothing achieved significant improvement over the other two smoothing approaches on 20NG because, as we mentioned earlier, data on 20NG corpus is very sparse. However, the Macro-F1 metric was still significantly improved on majority of testing collections, though the magnitude of improvement was less than in the case of 1% training data. It is because some small classes still have serious data sparse problems even though one-third of the documents are selected for training. For example, the smallest four classes in OHSUMED have only about seventy documents for training. This fact is consistent with the finding from the previous experiment that the more sparse the data, the more effective the semantic smoothing for Bayesian text classifiers.

Table 5.5: The comparisons of Bayesian text classifiers with three smoothing techniques (CSSS, Lap, and Bkg) on 20NG with the number of training documents ranging from one to five hundred.

(a) The result of micro-F1

Training Data Size	Micro-F1				
	Lap	Bkg	CSSS	vs. Lap	vs. Bkg
1 docs	0.121	0.222	0.324	**167.9%	**45.9%
2 docs	0.155	0.298	0.422	**171.9%	**41.8%
5 docs	0.322	0.426	0.539	**67.5%	**26.5%
10 docs	0.427	0.526	0.613	**43.7%	**16.6%
25 docs	0.558	0.628	0.688	**23.3%	**9.6%
50 docs	0.643	0.694	0.736	**14.4%	**6.0%
100 docs	0.698	0.744	0.773	**10.8%	**4.0%
250 docs	0.756	0.791	0.812	**7.3%	**2.6%
500 docs	0.787	0.814	0.828	**5.3%	**1.8%

(b) The result of macro-F1

Training Data Size	Macro-F1				
	Lap	Bkg	CSSS	vs. Lap	vs. Bkg
1 docs	0.096	0.203	0.294	**205.5%	**44.6%
2 docs	0.142	0.290	0.404	**184.0%	**39.4%
5 docs	0.312	0.420	0.531	**70.3%	**26.5%
10 docs	0.421	0.523	0.609	**44.5%	**16.4%
25 docs	0.557	0.623	0.684	**22.8%	**9.8%
50 docs	0.642	0.684	0.732	**14.1%	**7.0%
100 docs	0.692	0.732	0.769	**11.2%	**5.1%
250 docs	0.741	0.776	0.807	**8.9%	**4.0%
500 docs	0.771	0.799	0.824	**6.9%	**3.1%

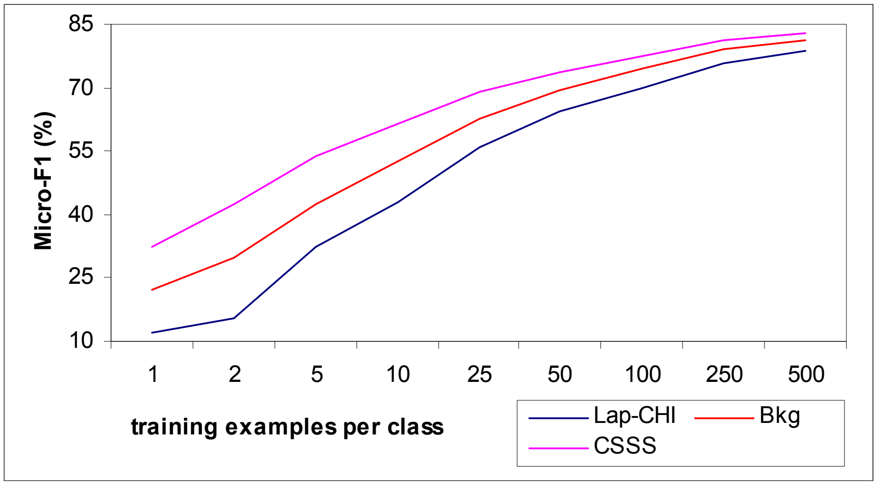


Figure 5.1: The performance of Bayesian text classifiers with semantic smoothing, Laplace smoothing, and background smoothing on the corpus of 20NG with different number of training documents.

To see more clearly the variance of the effectiveness of semantic smoothing with the change of the training data size, we evaluate the 20NG corpus with a number of training documents ranging from one to five hundred. We select 20NG for demonstration because the class labels have a uniform distribution in this corpus and thus it's easy to control the size of training data for each class. The results are shown in Figure 5.1. Clearly, we can see that the effectiveness of semantic smoothing is in inverse proportion to the size of the training documents. In the case of one-document training, semantic smoothing has the biggest gain of 167.9% and 45.9% in terms of Micro-F1 over Laplace smoothing and background smoothing, respectively. With the increase of training documents, data become less sparse, and consequently, semantic smoothing becomes less effective and ends with slight gains (5.3% and 1.8%) over the two baseline smoothing methods.

5.4.2 Context Sensitive vs. Context Insensitive

Our semantic smoothing method provides a framework which can incorporate various topic signatures. If the topic signature itself is context sensitive (e.g., multiword phrases and ontology-based concepts), we refer to it as context-sensitive semantic smoothing (CSSS); otherwise we refer to it as context-insensitive semantic smoothing (CISS) (i.e., using words as topic signatures). It is worth noting that CISS is different from the translation model (Berger and Lafferty, 1999) in that the former uses co-occurrence data, whereas the latter uses document-query pairs, even though both of them use words as topic signatures. Zhou et al. (2006c) has shown that CSSS performs slightly better than CISS in the setting of text retrieval because CSSS can take the advantage of contexts and make more specific and accurate mapping, but their effectiveness remains unclear for text classification.

The comparison of CSSS to CISS is shown in Table 5.6. Overall, CSSS gains slight improvement over CISS, though the semantic mapping result of CSSS looks much better than CISS. More surprisingly, CSSS achieves the largest gain over CISS on the corpus of OHSUMED, in which single-word meanings are supposed to be more consistent than in other collections, and thus CSSS should take less advantage. OHSUMED is a biomedicine corpus, and many single-word terms are gene, protein, and cell names such as p53, brca1, and orc1. These domain-specific terms are much less ambiguous than general conversional terms. However, another important fact is that we extract much more topic signatures in OHSUMED than in the other two collections (see Table 1). In OHSUMED, the number of extracted unique topic signatures is about half of the number of unique single-word terms, but the rate is only one-tenth in the other two collections. In other words, extracted topic signatures in OHSUMED are more representative than in other collections. We think this is an influential factor that affects the effectiveness of semantic smoothing.

Table 5.6: The comparison of context-sensitive semantic smoothing (CSSS) to context-insensitive semantic smoothing (CISS) on Bayesian text classification.

(a) 1% of documents for training

Collection	Micro-F1			Macro-F1		
	CISS	CSSS	Change	CISS	CSSS	Change
OHSUMED	0.401	0.413	**2.8%	0.344	0.351	*2.2%
20NG	0.623	0.613	*-1.6%	0.616	0.609	-1.2%
LATimes	0.577	0.581	0.8%	0.549	0.554	0.9%
LATimes†	0.558	0.559	0.2%	0.529	0.530	0.3%

(b) 33% of documents for training

Collection	Micro-F1			Macro-F1		
	CISS	CSSS	Change	CISS	CSSS	Change
OHSUMED	0.663	0.665	**0.4%	0.636	0.640	**0.7%
20NG	0.801	0.820	**2.4%	0.786	0.816	**3.8%
LATimes	0.724	0.729	**0.8%	0.693	0.700	**1.0%

To further validate this hypothesis, we compare CISS and CSSS on 20NG corpus and change the number of training documents from one to five hundred. The result is reported in Table 5.7. Interestingly, CSSS performs worse than CISS when the training data set is very small. With the increase of training documents, the extracted topic signatures become more representative and approach the true topics associated with those training documents. Eventually, CSSS exceeds CISS.

Table 5.7: The comparison of CSSS to CISS on 20NG corpus with the number of training documents ranging from one to five hundred. The parameter λ is 0.4 for both CISS and CSSS.

Training Data Size	Micro-F1			Macro-F1		
	CISS	CSSS	Change	CISS	CSSS	Change
1 docs	0.389	0.324	**-16.7%	0.367	0.294	**-20.0%
2 docs	0.474	0.422	**-10.9%	0.464	0.404	**-13.0%
5 docs	0.566	0.539	*-4.8%	0.558	0.531	*-4.8%
10 docs	0.623	0.613	*-1.6%	0.616	0.609	-1.2%
25 docs	0.676	0.688	**1.7%	0.668	0.684	*2.4%
50 docs	0.713	0.736	**3.1%	0.702	0.732	**4.4%
100 docs	0.749	0.773	**3.2%	0.736	0.769	**4.5%
250 docs	0.791	0.812	**2.7%	0.775	0.807	**4.1%
500 docs	0.812	0.828	**2.0%	0.797	0.824	**3.4%

In summary, two factors influence the effectiveness of CSSS compared to CISS. One is the ambiguity of single-word terms in the corpus. The more ambiguous the single-word terms, the more effective the CSSS is. The other is the relative number of extracted topic signatures. The more topic signatures, the more effective the CSSS is. Moreover, CSSS takes less computational complexity and runs faster than CISS because the magnitude of unique topic signatures is often much smaller than single-word terms and thus needs less mappings.

5.4.3 Reuse of Semantic Knowledge

One advantage of semantic smoothing over topic models including pLSI and LDA is its reusability. When one document or one set of documents comes, we can extract topics signatures and then map to single-word terms according to previously learned semantic knowledge. LDA is also able to predict the distribution of topics in a new document, but it assumes the prior Dirichlet distribution of the new document is similar to or the same as that of the previously learned document set. Thus, it may be problematic crossing domains or collections.

To verify the reusability of semantic knowledge, we design two experiments. In one experiment, we learn semantic mapping knowledge from TDT2 (64,500 news articles) and then employ it to classify the LATimes collection. Although TDT2 and LATimes are in the same domain of news articles, the overlapping of multiword phrases and words is not very high. 6,269 of 10,414 multiword phrases in LATimes appear in TDT2 and 39,735 of 63,510 words in LATimes appear in TDT2. The phrase and word coverage rates are 60% and 63%, respectively. In another experiment, the semantic mapping is learned from a subcollection of 280,000 Medline abstracts published in the first half year of 2000 and then utilized to classify the OHSUMED collection (7,400 Medline abstracts published in 1991). The words and concepts in the second experiment have a complete coverage, that is, all words and concepts in the training collections appear in the testing collections.

Table 5.8: The classification result using external semantic mapping knowledge and its comparison to internal knowledge. 1% data is used for training. † indicates the result is based on external semantic mapping knowledge.

(a) The result of micro-F1

Collection	Lap	Bkg	CSSS	vs. Lap	vs. Bkg
OHSUMED	0.352	0.372	0.413	**17.3%	**10.9%
OHSUMED†	0.352	0.372	0.428	**21.6%	**15.1%
LATimes	0.525	0.538	0.581	**10.7%	**7.9%
LATimes†	0.525	0.538	0.559	**6.5%	**3.8%

(b) The result of macro-F1

Collection	Lap	Bkg	CSSS	vs. Lap	vs. Bkg
OHSUMED	0.205	0.280	0.351	**76.2%	**29.1%
OHSUMED†	0.205	0.280	0.364	**77.6%	**30.0%
LATimes	0.492	0.513	0.562	**14.3%	**9.5%
LATimes†	0.492	0.513	0.541	**10.0%	**5.4%

(c) External knowledge vs. internal knowledge

Collection	Internal	External	Change
OHSUMED Micro-F1	0.413	0.428	3.6%**
OHSUMED Macro-F1	0.351	0.364	3.7%**
LATimes Micro-F1	0.581	0.559	-3.8%**
LATimes Macro-F1	0.562	0.541	-3.7%**

The experiment results are shown in Table 5.8. First, the semantic smoothing with external mapping knowledge still significantly outperforms the Laplace smoothing and the background smoothing on both collections. Second, the performance of external knowledge is comparable to that of the internal knowledge. On the collection of LATimes, the external knowledge performs slightly worse than the internal knowledge, which is mainly due to the low coverage of the external knowledge. On the collection of OHSUMED, the external knowledge even achieves slightly better results than the internal knowledge. The possible explanation is that the external knowledge is learned from a much larger collection and consequently, the semantic mapping is more robust and reasonable.

This finding is of practical value. It means the learned semantic knowledge can serve as a dictionary for future use. One then does not have to prepare semantic knowledge by himself, but download online semantic knowledge resources fitting his application in future. It is somehow time consuming to prepare high-quality semantic knowledge. The reusability of semantic knowledge brings great convenience and high feasibility to the wide use of semantic smoothing for Bayesian text classification and other related applications.

5.4.4 Semantic Smoothing vs. SVM

Support vector machine (SVM) is a powerful learning approach for solving two-class pattern recognition problem (Vapnik, 1995). Within the SVM framework, an example (document) is represented as a vector and the learning process is equivalent to finding a "decision surface" which "best" separates positive and negative training examples. Previous empirical studies have shown that SVM using linear kernel could outperforms many other text classifiers including Naïve Bayes (Yang and Liu, 1999). However, in previous studies, a large number of training examples are used to learn the support vectors, making the performance of SVM classifiers with small training data unclear. Thus, we compare SVM classifiers and NB classifiers with semantic smoothing in the case of small training data.

Table 5.9: The comparisons of support vector machines (SVM) to bayesian classifier with context-sensitive semantic smoothing (CSSS)

(a) 1% of documents for training

Collection	Micro-F1			Macro-F1		
	SVM	CSSS	Change	SVM	CSSS	Change
OHSUMED	0.351	0.413	**17.5%	0.206	0.351	**70.7%
20NG	0.472	0.613	**29.9%	0.464	0.609	**31.1%
LATimes	0.524	0.581	**10.8%	0.491	0.554	**12.7%

(b) 33% of documents for training

Collection	Micro-F1			Macro-F1		
	SVM	CSSS	Change	SVM	CSSS	Change
OHSUMED	0.680	0.665	**-2.2%	0.646	0.640	-0.9%
20NG	0.797	0.820	**2.8%	0.793	0.816	**2.9%
LATimes	0.781	0.729	**-6.7%	0.765	0.700	**-8.5%

SVM cannot handle a multi-class classification problem directly. It is required to decompose a multi-class classifier into a set of binary classifiers and then combine the results to predict the label of a testing document. The mechanisms of decomposition and combination are not trivial, but a hot research topic (Allwein et al., 2000). Furthermore, several other factors such as the choice of kernel, scaling, and feature selection can also affect the performance of an SVM text classifier. Thus, we try different configurations and report the best tuned result. The best configuration uses a linear kernel, one-versus-all (OVA) code matrix as well as a loss-based multi-class decoder (hinge loss function is used) and does not apply any feature selection or vector scaling. The binary SVM classifier uses SVM-light 6.01.

The results of SVM with a large number of training documents (see Table 5.9b) are consistent with previous studies (Yang and Liu, 1999). SVM always significantly outperform naïve Bayes (see Table 5.4). The Bayesian text classifier with semantic smoothing has similar performance to naïve Bayes and is less effective than SVM. However, when the number of training documents becomes extremely small (e.g., 1% in our experiment), SVM performs no better than naïve Bayes and significantly less than the Bayesian classifiers with CSSS as shown in Table 5.9a. It is mostly likely because a large number of features are blind to SVM when the training document set is very small, and the power of SVM is compromised while a Bayesian classifier can expand meaningful features through semantic smoothing.

5.4.5 Semantic Smoothing vs. Active Learning

In our experiments, semantic smoothing has proven to be effective in improving classification performance when the size of the training dataset is small. In literature, active learning also shows its effectiveness in dealing with a small number of training samples. Active learning typically estimates an initial classifier from a few labeled seed documents; then it iteratively assigns class labels to unlabeled documents and uses all documents to re-estimate a new classifier until the classifier converges (Nigam et al., 2000). For this reason, we compare semantic smoothing with an active learning classifier proposed by Nigam et al. (2000). We choose this approach for comparison because it has the state-of-the-art performance and is also within the framework of Bayesian classifiers. The comparison result is shown in Table 5.10. The active learning algorithm uses Laplace smoothing and seems quite sensitive to the feature selection. The result of active learning reported in Table 5.10 is actually the one with the best tuning.

Table 5.10: The comparison of active learning (AL) to context-sensitive semantic smoothing (CSSS). 1% of documents are used for training and the remaining 99% for testing. Among testing documents, 50% will be used to iteratively optimize the Bayesian classifier during active learning.

Collection	Micro-F1			Macro-F1		
	AL	CSSS	Change	AL	CSSS	Change
OHSUMED	0.368	0.413	**12.1%	0.205	0.351	**71.1%
20NG	0.575	0.613	**6.6%	0.551	0.609	**10.4%
LATimes	0.566	0.581	*2.6%	0.536	0.554	3.3%

Active learning does improve the performance over the baseline naïve Bayesian classifier on most collections (see Table 5.3). However, it is much less effective than the semantic smoothing approach. Likewise, complexity and robustness are two other concerns regarding active learning. Active learning has an iterative learning process and thus runs slow compared to semantic smoothing and naïve Bayes. The performance of active learning depends on the added unlabeled data. Active learning, however, looks simpler than semantic smoothing. It does not have to prepare topic signatures and semantic knowledge in advance.

Nigam et al. argue that unlabeled data contain information about the joint distribution over feature other than labels, and thus they can sometimes be used together with a sample of labeled data to increase classification accuracy. Semantic smoothing and active learning are similar in the sense that both use co-occurrence data to improve the accuracy of text classifiers. In semantic smoothing, co-occurrence data are employed to estimate the semantic mapping between topic signatures and single-word features. There is, however, a significant difference regarding the implementations of the two approaches.

Active learning typically estimates an initial classifiers from a few labeled seed documents; then it iteratively assign class labels to unlabeled documents and use all documents to re-estimate a new classifier until the classifier converges (Nigam et al., 2000). Thus, unlabeled document are assumed to be generated from the same set of class models as the labeled documents. This is sometimes still a strong assumption. With this assumption, one can not arbitrarily use a huge number of external unlabeled texts (e.g., texts collected from the Internet) to obtain a more accurate estimation of the joint distributions over features. As shown in Table 5.11, when noise (i.e., an unlabeled document outside the collection) is added, the performance of active learning is dropped quickly. Take the example of the LATimes collection. The active learning performs worse than the naïve Bayes when 40% unlabeled documents are selected from TDT2. The method of semantic smoothing does not have such a limitation. Even if the semantic knowledge is completely learned from TDT2 corpus, it still significantly outperforms the naïve Bayes.

Table 5.11: The classification performance of active learning with noise on the corpus of LATimes. Active learning uses 8,876 unlabeled documents. ALXX means XX percentage of unlabeled documents are from the corpus of TDT2. CSSS† denotes semantic smoothing with semantic knowledge completely learned from TDT2 corpus.

	NB	AL	AL10	AL20	AL30	AL40	CSSS	CSSS†
Micro-F1	0.525	0.566	0.562	0.552	0.538	0.519	0.581	0.559
Macro-F1	0.492	0.536	0.529	0.516	0.502	0.481	0.554	0.530

In semantic smoothing, the learning and the use of semantic knowledge are two independent processes. Theoretically, one can use any available texts to estimate semantic mappings. Because topic signatures such as multiword phrases and ontological concepts carry some contextual information, we think they are able to cross documents, collections, and even domains. In the stage of using semantic knowledge, it is not necessary to pick up semantic knowledge (i.e., topic signature mappings) from the same source. Instead, one can combine semantic knowledge from different sources. Naturally, for real applications, the closer the testing collection to the texts from which semantic knowledge are estimated, the more effective. In our experiment, even though the vocabulary coverage of the testing collection and the collection for the semantic knowledge estimation is only about 60%, the results for text classification and clustering are still very good. In short, semantic smoothing does not have limitations on the selection of unlabeled documents and it can arbitrarily combine and reuse semantic knowledge; it is closer to real world settings compared to active learning.

5.5 Tuning of the Mapping Coefficient

Topic signatures are very effective in mapping to single-word features, as demonstrated in Figures 3.5 and 4.1. But the number of extracted topic signatures is often much less than the original single-word features. Therefore, if only topic-signature–based mapping is used, one may suffer serious information loss. To solve this problem, we linearly interpolate the topic signature–based semantic mapping with a simple language model (see equation 5.6), as many other researchers have done (Wei and Croft, 2006; Zhai and

Lafferty, 2001a; Zhai and Lafferty, 2002; Zhang et al., 2006a; Zhou et al., 2006c). Then the optimization of mixture weights (i.e., mapping coefficient) becomes a problem.

The interpolation-based mixture model is originally designed to smooth multi-gram language models (Blei et al., 2003). The mixture weight lambda can be globally optimized using the EM algorithm (Dempster et al., 1977) with the objective function of maximizing the posterior probability of generating a text collection. However, the objective of text classification is not the maximum posterior probability of the text, but the classification accuracy. The inconsistency of two objectives leads to the ineffectiveness of this automatic parameter optimization approach for text classification. We implement this approach, and the results are shown in Table 5.12. The automatic prediction of the optimal mapping coefficient is close to the manually tuned parameter on the corpus of 20NG, but quite far from the best results on the other two collections, especially in the case of large training data.

Table 5.12: The comparison of manual parameter tuning to automatic parameter tuning. The parameter λ is the mapping coefficient.

Collection	Small training (1%)				Large training (33%)			
	manual		automatic		manual		automatic	
	λ	mi-F1	λ	mi-F1	λ	mi-F1	λ	mi-F1
20NG	0.4	0.613	0.4	0.613	0.4	0.820	0.3	0.818
OHSUMED	0.4	0.413	0.7	0.408	0.1	0.665	0.8	0.627
LATimes	0.4	0.581	0.6	0.578	0.1	0.729	0.7	0.716

The language modeling approach to IR has encountered the same problem. Zhai and Lafferty propose a modified EM-based algorithm to find the optimal mixture weight for their two-stage language models (Zhai and Lafferty, 2002). But this method is not very effective in the setting of text retrieval. They still recommend careful tuning of the mixture weight (Zhai and Lafferty, 2001a and 2002). In most of the recent work on mixture language modeling IR, the mixture weights are also manually tuned (Wei and Croft, 2006; Zhang et al., 2006a; Zhou et al., 2006c).

Fortunately, previous studies have shown that similar collections have similar optimal mapping coefficients, making held-out training possible. The previous work (Zhou et al., 2006c) showed that 0.3 is a good empirical setting for the mapping coefficient in the setting of text retrieval. In the experiment of text classification, all three collections achieved the best result when the mapping coefficient was set to 0.4 when 1% documents were used for training. When the training documents increased to 33%, the data became less sparse, and the optimal mapping coefficient reduced to 0.1, except for the 20NG corpus. As discussed earlier, the 20NG corpus is quite sparse. Even if 33% documents are used for training, the features still look sparse and the optimal results are achieved when the mapping coefficient set to 0.4.

We can also see the robustness of semantic smoothing for Bayesian text classification from Figure 5.2. In a wide range around the optimal mapping coefficient, semantic smoothing outperforms background smoothing. On 20NG and OHSUMED, semantic smoothing always beats background smoothing regardless the setting of the mapping coefficient. On LATimes, semantic smoothing wins positive gain over background smoothing, except with the setting point of one.

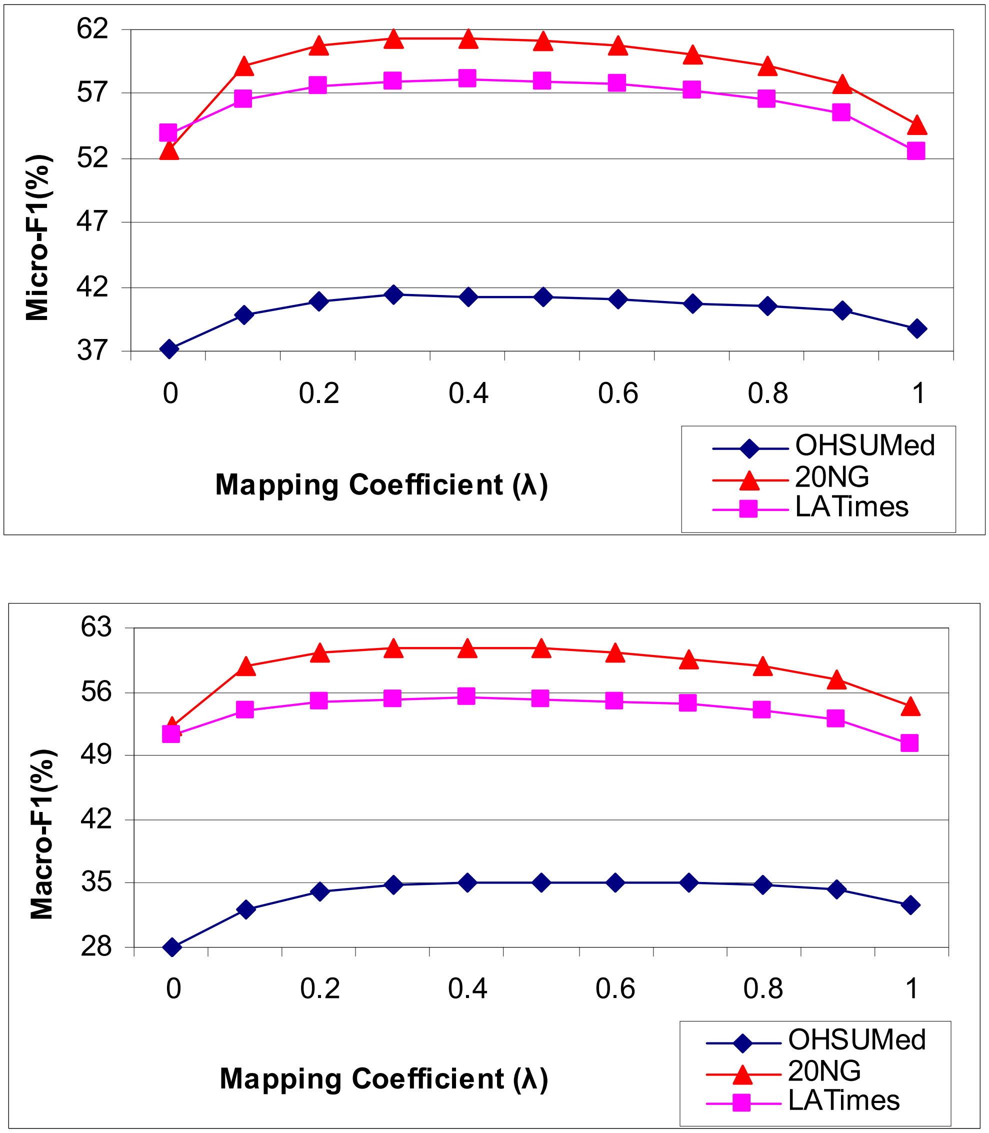


Figure 5.2: The variance of the classification performance (i.e., micro-F1 and macro-F1) on three testing collections with the change of the mapping coefficient, which controls the influence of the mapping component in the mixture model. The Bayesian text classifiers use 1% of documents for training.

Table 5.13: The variance of the classification performance on three testing collections with the change of the mapping coefficient. The Bayesian text classifiers use 1% of documents for training.

Mapping Coefficient	Micro-F1			Macro-F1		
	OHSUMED	20NG	LATimes	OHSUMED	20NG	LATimes
0	37.21	52.60	53.84	28.03	52.31	51.34
0.1	39.82	59.25	56.58	32.24	58.85	53.91
0.2	40.89	60.74	57.58	34.00	60.31	54.82
0.3	41.32	61.25	58.00	34.85	60.83	55.23
0.4	41.28	61.32	58.12	35.10	60.89	55.35
0.5	41.18	61.15	57.97	35.20	60.74	55.22
0.6	40.99	60.70	57.76	35.16	60.30	55.04
0.7	40.78	60.03	57.29	35.04	59.65	54.63
0.8	40.50	59.15	56.59	34.81	58.79	53.98
0.9	40.09	57.77	55.54	34.29	57.45	53.00
1	38.78	54.60	52.55	32.56	54.37	50.16

In short, one can take the following empirical rules with respect to the choice of the mapping coefficient: if data are very sparse, set the mapping coefficient to 0.3~0.5; decrease the value when data become less sparse; when sufficient training data are provided, stop using semantic smoothing.

5.6 Conclusions

We proposed a novel semantic smoothing method for Bayesian text classification. The core idea of the smoothing method is to identify explicit topic signatures such as multiword phrases and ontological concepts in documents and then statistically map them onto single-word features. According to whether the topic signature itself is context sensitive, the smoothing method is further categorized into context-sensitive semantic smoothing (CSSS) and context-insensitive semantic smoothing (CISS). The semantic mapping from multiword phrases, ontological concepts to single-word features, is viewed as CSSS; the word-word semantic mapping is considered CISS. We then evaluated the behavior of CSSS and CISS on three collections: 20NG, LATimes, and OHSUMED.

We estimated semantic mappings between topic signatures and single-word features using co-occurrence data and an EM-based algorithm. Because it is cheap to collect co-occurrence data, the acquisition of a large amount of semantic mapping knowledge becomes feasible. In terms of mapping quality, context-sensitive topic signatures perform much better than context-insensitive ones such as single-word terms. Without contextual constraints, the mapping result is fairly general and often contains mixed topics. Compared to topic models, topic signature is a more intuitive and lightweight representation of topics. It is also easy to be identified and stored. Topic signatures, especially context-sensitive ones, can cross documents, collections, and even domains, which make it possible to reuse learned semantic knowledge in the future. Our experiments verified this hypothesis. With 60% vocabulary coverage, the semantic knowledge learned from other corpus can still significantly improve the accuracy of text classification over Laplace smoothing and background smoothing.

The effectiveness of semantic smoothing for Bayesian text classification depends on the degree of the data sparsity. In general, the sparser the data, the more effective the semantic smoothing is. When the size of training documents is small, the Bayesian classifier with semantic smoothing not only outperforms the classifiers with background smoothing and Laplace smoothing, but also beats the state-of-the-art active learning classifiers and SVM classifiers. With the increase of training documents, the gap among semantic smoothing, Laplace smoothing, and background smoothing is decreasing. This finding is of great practical value because it is always expensive to get the labeled training documents for real applications.

CSSS performs slightly more effectively than CISS for text classification. But if the number of training documents is too small, say, only one or two, CISS runs more effectively than CSSS, because too few extracted context-sensitive topic signatures may misrepresent the topics associated with the training documents. However, CSSS is always more efficient than CISS, whether the size of training documents is small or large. A document contains a smaller number of context-sensitive topic signatures (e.g., multiword phrases or concepts) than words, on average. Thus, CSSS needs much less time complexity than CISS during semantic mapping.

Semantic smoothing uses a mixture language model with the mapping coefficient to control the influence of the two components. The optimization of the mapping coefficient is still an ongoing problem. We proposed an automatic parameter tuning method that obtains the optimal value by maximizing the generative probability of the testing documents. However, this approach is not robust. Sometimes the estimated parameter is quite close to the optimal value, but other times it is quite far. This is also the problem of the IR community when mixture language models are used for retrieval. First of all, fortunately, the proposed semantic smoothing is quite robust; it beats the baseline smoothing methods in a wide range. Second, there are rules of thumb available to the tuning of the mapping coefficient. If data are very sparse, set the mapping coefficient to

0.3~0.5; decrease the value when data become less sparse; and when sufficient training data are provided, stop using semantic smoothing.

In the future, we will continue working on the optimization of the mapping coefficient. We will also focus on the reuse of semantic knowledge. In this chapter, the experiment showed that it was quite promising to reuse the semantic knowledge. In future work, we will be more interested in factors that affect the effectiveness of semantic knowledge reuse for various applications such as text classification and text retrieval.

CHAPTER 6: SEMANTIC SMOOTHING IN TEXT CLUSTERING

6.1 Introduction

Document clustering algorithms can be categorized into agglomerative and partitional approaches according to the underlying clustering strategy (Kaufman and Rousseuw, 1990). The agglomerative approaches initially assign each document into its own cluster and repeatedly merge pairs of clusters with the shortest distance until only one cluster is left. The partitional approaches iteratively re-estimate the cluster generative model (or calculate the cluster centroid) and reassign each document into the closest cluster until no further documents can be moved. The clustering result of the agglomerative approach is free of the initialization and gives a very intuitive explanation of why a set of documents are grouped together. However, in comparison with partitional approaches, it suffers from the $O(n^2)$ clustering time and performs poorly in general (Steinbach et al., 2000).

Recent advances in document clustering have shown that, in general, model-based partitional clustering approaches are more efficient and effective than agglomerative clustering approaches (Zhong and Ghosh, 2005). However, there are also two identified problems, the density of class-independent general words and the sparsity of class-specific core words, with the model-based approaches. Model-based partitional approaches estimate cluster models instead of document models. A cluster often contains much more than one document. Thus, the data sparsity problem is not as serious as in pairwise document similarity calculation. But if the size of the dataset for clustering is small or the dataset is extremely skewed on different classes, the sparsity of core words will still be a serious problem. Besides, no matter how many documents a cluster has, general words always dominate the cluster; thus, discounting the effect of general words is always helpful to improve cluster quality.

Discounting seen words and assigning reasonable counts to unseen words are two exact goals of the probabilistic language model smoothing. To the best of our knowledge, the effect of model smoothing has not been extensively studied in the context of document clustering. Most model-based clustering approaches simply use Laplace smoothing to prevent zero probability (McCallum and Nigam, 1998; Nigam et al., 2000; Zhong and Ghosh, 2005), while most similarity-based clustering approaches employ the heuristic TF-IDF scheme to discount the effect of general words (Steinbach et al., 2000). In contrast, the study of language model smoothing has been a hot topic in the community of information retrieval (IR) with the increasing popularity of the language modeling approach to IR in recent years (Berger and Lafferty, 1999; Zhai and Lafferty, 2001a; Zhai and Lafferty, 2002; Zhou et al., 2006c). In this chapter, we will adapt the smoothing techniques used in IR to the context of document clustering and hypothesize that the document or cluster model smoothing can significantly improve the quality of document clustering.

We evaluate our semantic smoothing method in conjunction with a model-based k-means algorithm on three datasets: 20-Newsgroups, the Los Angeles Times, and OHSUMED. The model-based k-means with semantic smoothing consistently achieves better results than with simple background smoothing and Laplace smoothing. The performance of model-based k-means with semantic smoothing is also better than the spherical k-means that is considered one of the best algorithms for text clustering (Dhillon and Modha, 2001). The rest of the paper is organized as follows: Section 6.2 describes the clustering algorithm, that is, model-based k-means with semantic smoothing. Section 6.3 shows the evaluation method and the datasets. In section 6.4, we present and discuss the experiment results. Section 6.5 concludes the chapter.

6.2 The Clustering Method

6.2.1 Model-based K-Means

The model-based k-means uses a Bayesian classifier for document assignment in iteration. For this reason, we also call it Bayesian text clustering in this thsis. It is a generalized version of the standard k-means (Zhong and Ghosh, 2005). It assumes that there are *k* parameterized models, one for each cluster. Basically, the algorithm iterates between a model re-estimation step and a sample re-assignment step, as shown in Figure 6.1. The implementation of cluster model estimation depends on the word distribution assumption made on the dataset. Zhong and Ghosh (2005) compared several generative models for document clustering and found out that the multinomial model consistently outperformed the multivariate Bernoulli model. For this reason, we chose the multinomial model for evaluation. Based on the naive Bayes assumption, the log likelihood of document *d* generated by the *j*-th multinomial cluster model is:

$$\log p(d \mid c_j) = \sum_{w \in V} c(w,d) \log p(w \mid c_j) \qquad (6.1)$$

where $c(w,d)$ denotes the frequency count of word *w* in document *d*, and *V* denotes the vocabulary. Thus, the problem remains to estimate parameters $p(w \mid c_j)$ for the cluster model.

```
Algorithm: Model-based K-Means

Input: dataset $D = \{d_1,...,d_n\}$, and the desired number of clusters k.

Output: trained cluster models $\Lambda = \{\lambda_1,...,\lambda_k\}$ and the document assignment $Y = \{y_1,...,y_n\}, \mathrm{y_i} \in \{1,...k\}$

Steps:
1. Initializes document assignment Y.
2. Model re-estimation: $\lambda_i = \arg\max_{\lambda} \sum_{d \in c_i} \log(d \mid \lambda)$
3. Sample re-assignment: $\mathrm{y_i} = \arg\max_{j} \log p(d_i \mid \lambda_j)$
4. Stops if Y does not change, otherwise go to step 2
```

Figure 6.1: The framework of the model-based k-means algorithm

6.2.2 Semantic Smoothing

The parameter estimation of multinomial models is as simple as counting word frequency in the cluster. However, one has to smooth the model in order to prevent zero probability caused by a sparse data problem. As we do for the Bayesian text classification, we compare three smoothing methods for model-based k-means. They are Laplace smoothing, background smoothing, and semantic smoothing. The exact formulas for these three smoothing methods are very similar to the ones used by Bayesian text classifiers, as described in Chapter 5. The only difference is that the word statistics are from the labeled training document in text classification, while the word statistics are from the whole collection of unlabeled documents in text clustering.

With Laplace smoothing, the cluster model is formalized as below:

$$p(w \mid c_j) = \frac{1 + c(w, c_j)}{|V| + \sum_w c(w, c_j)} \qquad (6.2)$$

where $c(w,c_j)$ is the frequency count of word *w* in the *j*-th cluster. Obviously, Laplace smoothing assigns to all unseen words of a given cluster a fixed probability.

The Jelinek-Mercer approach (Jelinek, 1990; Zhai and Lafferty, 2001a) is one of the most frequently used background smoothing approaches. In the setting of model-based k-means, it interpolates a unigram cluster model with a collection background model, controlled by the parameter β as shown in the equation (6.3):

$$p_b(w \mid c_j) = (1-\beta) p_{ml}(w \mid c_j) + \beta p(w \mid D) \qquad (6.3)$$

where $p_{ml}(w \mid c_j)$ is a unigram cluster model with maximum likelihood estimate and $p_b(w \mid c_j)$ denotes the cluster model with the background smoothing. The coefficient β is empirically set to 0.5 in the experiment.

We linearly interpolate the semantic mapping component with a simple language model as described in equation 6.3, and the cluster model ends with the following formula:

$$p_s(w \mid c_j) = (1\text{-}\lambda) p_b(w \mid c_j) + \lambda \sum_k p(w \mid t_k) p(t_k \mid c_j) \quad (6.4)$$

where t_k denotes the *k*-th topic signature and $p(t_k \mid c_j)$ is the distribution of topic signatures in a given cluster, which can be computed via maximum likelihood estimates.

The mapping coefficient λ is to control the influence of the semantic mapping component in the mixture model.

6.3 Evaluation Methodology and Datasets

The clustering quality is evaluated by three extrinsic metrics, purity (Zhao and Karypis, 2001), entropy (Steinbach et al., 2000), and normalized mutual information (NMI) (Banerjee and Ghosh, 2002). In this thesis, we only analyze the NMI results because NMI is an increasingly popular measure of cluster quality. NMI is defined as the mutual information between the cluster assignments and a pre-existing labeling of the dataset normalized by the arithmetic mean of the maximum possible entropies of the empirical marginals, that is,

$$NMI(X,Y) = \frac{I(X;Y)}{(\log k + \log c)/2} \qquad (6.5)$$

where *X* is a random variable for cluster assignments, *Y* is a random variable for the pre-existing labels on the same data, *k* is the number of clusters, and *c* is the number of pre-existing classes. One can refer to the paper, "Frequency sensitive competitive learning for clustering on high-dimensional hperspheres" (Banerjee and Ghosh, 2002), or to any text books to get the details of computing mutual information *I(X; Y)*. NMI ranges from 0 to 1. The bigger the NMI, the higher quality the clustering is. NMI is better than the other common extrinsic measures such as purity and entropy in the sense that it does not necessarily increase when the number of clusters increases.

In the experiment, we first compare the effectiveness of three smoothing methods for model-based k-means: Laplace smoothing, background smoothing, and semantic smoothing. Since semantic smoothing is further divided into context-sensitive semantic smoothing (CSSS) and context-insensitive semantic smoothing (CISS), we also compare CSSS to CISS. To get the sense of how good the model-based k-means is, we finally run the spherical k-means (Dhillon and Modha, 2001). Spherical k-means uses cosine similarity and usually normalizes the vector to remove the bias that arises because of the length of a document. Empirical studies have shown that spherical k-means is far more effective in text clustering than other schemes. In our experiments, the spherical k-means takes two different scoring schemes: normalized TF and TF-IDF. In summary, we compare the effectiveness of six text clustering methods, four for model-based k-means and two for spherical k-means.

Our pilot study shows that Laplace smoothing performs poorly on a large vocabulary space. Thus, we only keep terms which appear in five or more documents. However, the size of the vocabulary space makes no difference on the other five clustering schemes.

K-means is a type of EM-based clustering algorithm. The final clustering result depends on the initialization. Thus, we conduct ten runs with random initialization and take the average as the final result. During the comparative experiment, each run has the same initialization except for the Laplace smoothing. There are two ways to initialize document clusters. One is to randomly select one document for each cluster. The other is to randomly assign all documents into the given number of clusters. The second initialization scheme may lead to overfitting problems. Our pilot study shows that the aforementioned clustering methods, except Laplace smoothing, perform poorly with the second initialization scheme. However, Laplace smoothing sometimes receives very bad results with the first initialization scheme. Thus, we try both initialization schemes for Laplace smoothing and report the better one.

The datasets used in the experiment include 20-Newsgroups, the Los Angeles Times, and OHSUMED. Since these datasets are the same as the ones used in the text classification experiment, please see the details in Chapter 5 regarding their descriptions and text processing process.

The effect of semantic smoothing on small datasets is stronger than on large datasets, because small datasets have serious data sparse problems. To test this effect on model-based text clustering, we conduct experiments on both large and small datasets. A large dataset contains all the documents from selected classes. To build small datasets, we randomly pick one hundred documents from each selected class of a given dataset and then merge them into a big pool for clustering. For each collection, we create five small datasets and average the experiment results.

6.4 Experiment Results

The clustering quality is measured by three metrics, purity, entropy, and NMI. The clustering results in the metrics of purity and entropy are shown in Tables 6.1 and 6.2, respectively. In general, the results of purity and entropy are consistent with the result of NMI. Our analysis and comparison in this section are based on the NMI metric.

Table 6.1: The purity results of four variants of model-based k-means (CSSS, CISS, Lap, and Bkg) and two variants of spherical k-means (NTF and TF-IDF)

(a) small dataset, 100 documents per class

Collection	NTF	TF-IDF	Lap	Bkg	CISS	CSSS
20NG	0.217	0.414	0.191	0.256	0.482	0.423
OHSUMED	0.195	0.293	0.184	0.278	0.340	0.323
LATimes	0.322	0.327	0.258	0.270	0.449	0.434

(b) large dataset, all documents are used for clustering

Collection	NTF	TF-IDF	Lap	Bkg	CISS	CSSS
20NG	0.231	0.481	0.472	0.428	0.557	0.545
OHSUMED	0.277	0.414	0.357	0.348	0.410	0.405
LATimes	0.351	0.468	0.502	0.489	0.518	0.531

Table 6.2: The entropy results of four variants of model-based k-means (CSSS, CISS, Lap, and Bkg) and two variants of spherical k-means (NTF and TF-IDF)

(a) small dataset, 100 documents per class

Collection	NTF	TF-IDF	Lap	Bkg	CISS	CSSS
20NG	2.445	1.758	2.328	2.091	1.483	1.494
OHSUMED	2.390	2.170	2.429	2.175	2.010	2.056
LATimes	1.828	1.866	1.884	2.022	1.506	1.514

(b) large dataset, all documents are used for clustering

Collection	NTF	TF-IDF	Lap	Bkg	CISS	CSSS
20NG	2.411	1.467	1.249	1.511	1.271	1.180
OHSUMED	2.231	1.841	1.978	2.022	1.825	1.825
LATimes	1.825	1.483	1.407	1.420	1.377	1.308

6.4.1 The Comparison of Three Smoothing Methods

We compare the effectiveness of Laplace smoothing, background smoothing, and context-sensitive semantic smoothing (CSSS). CSSS outperforms Lap and Bkg on all small dataset clustering tasks, as described in Table 6.3a. All improvements are statistically significant at the level of p<0.01 according to the paired-sample t-test. It also performs significantly better than Lap and Bkg on large dataset clustering, as described in Table 6.3b. But the magnitude of improvement on large datasets is much smaller than on small datasets. This finding is similar to what we find in the text classification experiment. The difference is that CSSS only obtains slight or no improvement over Lap and Bkg for the text classification task with a large number of training documents, whereas CSSS still wins a significant gain for large dataset clustering tasks.

Table 6.3: The NMI results of model-based k-means clustering with four smoothing techniques, that is, context-sensitive semantic smoothing (CSSS), context-insensitive semantic smoothing (CISS), Laplace smoothing (Lap), and background smoothing (Bkg). The symbols ** and * indicate the change is significant according to the paired-sample t-test at the level of p<0.01 and p<0.05, respectively.

(a) small dataset, 100 documents per class

Collection	NTF	TF-IDF	Lap	Bkg	CSSS	vs. Lap	vs. Bkg
20NG	0.176	0.391	0.240	0.201	0.441	**83.6%	**119%
OHSUMED	0.090	0.172	0.080	0.090	0.212	**164%	**135%
LATimes	0.200	0.185	0.145	0.122	0.322	**122%	**164%

(b) large dataset, all documents are used for clustering

Collection	NTF	TF-IDF	Lap	Bkg	CSSS	vs. Lap	vs. Bkg
20NG	0.192	0.506	0.493	0.489	0.564	**14.5%	**15.5%
OHSUMED	0.085	0.232	0.180	0.165	0.239	**32.8%	**44.6%
LATimes	0.201	0.349	0.382	0.371	0.420	**9.8%	**13.2%

A plausible explanation is that semantic smoothing well solves the overfitting problem of model-based k-means. Model-based k-means is an EM-styled iterative clustering algorithm. Documents in the whole collection tend to group into several large clusters, and all cluster models then quickly converge. The overfitting problem is especially severe when the dataset is highly skewed. With Lap or Bkg smoothing, documents tend to group into a few large clusters and quickly converge to local maxima. With semantic smoothing, small clusters still have great chance to grow, because small clusters could share many significant common words with other documents by semantic mapping. Our experiments do show that model-based k-means with semantic smoothing takes more iteration steps to converge than the other two smoothing approaches.

Table 6.4: The NMI comparison of CSSS to CISS on model-based k-means.

(a) small dataset, 100 documents per class

Collection	CISS	CSSS	vs. CISS
20NG	0.476	0.441	**-7.3%
OHSUMED	0.227	0.212	**-6.7%
LATimes	0.332	0.322	**-3.1%

(b) large dataset, all documents are used for clustering

Collection	CISS	CSSS	vs. CISS
20NG	0.571	0.564	-1.1%
OHSUMED	0.238	0.239	0.3%
LATimes	0.395	0.420	**6.2%

6.4.2 Context Sensitive vs. Context Insensitive

We compare context-sensitive semantic smoothing (CSSS) to context-insensitive semantic smoothing (CISS) in terms of effectiveness. In the experiment of text classifications, CSSS is slightly more effective than CISS, especially when the number of topic signatures in a class is large enough to reflect the underlying topics associated with the class. The pattern for text clustering is somehow different, as described in Table 6.4.

On small datasets, CISS shows slightly more effectiveness than CSSS. On large datasets, CISS and CSSS are comparable to each other.

We think the difference can also be attributed to the overfitting problem of model-based k-means. As discussed in Chapter 5, when a category is too small to contain a sufficient number of topic signatures, CISS seems more effective than CSSS. During the model-based k-means clustering, there are always many intermediate small clusters, especially when the dataset to cluster is small. Thus, CISS can help those intermediate small clusters to grow more effectively than CSSS.

6.4.3 Reuse of Semantic Knowledge

We design two experiments to verify the reusability of semantic knowledge. In one experiment, we learn semantic mapping knowledge from TDT2 and then utilize it to cluster the LATimes collection. In another experiment, we partition the OHSUMED collection (7,400 Medline abstracts published in 1991) using the semantic mapping knowledge learned from a subcollection of 280,000 Medline abstracts published in the first half of the year 2000. The design of the reusability experiment is the same as the one for text classification. Please refer to Section 5.4.3 for details.

The semantic smoothing using the external semantic knowledge still significantly outperforms Laplace smoothing and background smoothing within the framework of model-based k-means clustering as shown in Table 6.5. This result of semantic smoothing with external knowledge is also considerably better than the best result of spherical k-means. The performance of the external knowledge is comparable to that of the internal knowledge. On the collection of LATimes, the external knowledge performs worse than the internal knowledge, which is mainly due to the low coverage of the external knowledge. On the collection of OHSUMED, the external knowledge achieves slightly worse results than the internal knowledge. The possible explanation is that the external knowledge is learned from a much larger collection, and consequently, the coverage of the semantic mapping is very high.

Table 6.5: The clustering result on small datasets using external semantic mapping knowledge and its comparison to internal knowledge. † indicates the result is based on external semantic mapping knowledge.

(a) NMI Result

Collection	NTF	TF-IDF	Lap	Bkg	CSSS	vs. Lap	vs. Bkg
OHSUMED	0.090	0.172	0.080	0.090	0.212	**164%	**135%
OHSUMED†	0.090	0.172	0.080	0.090	0.207	**159%	**130%
LATimes	0.200	0.185	0.145	0.122	0.322	**122%	**164%
LATimes†	0.200	0.185	0.145	0.122	0.245	**69.3%	**101%

(b) External knowledge vs. internal knowledge

Collection	Internal	External	Change
OHSUMED	0.212	0.207	-2.4%*
LATimes	0.322	0.245	-23.9%**

6.4.4 The Comparison to Spherical K-Means

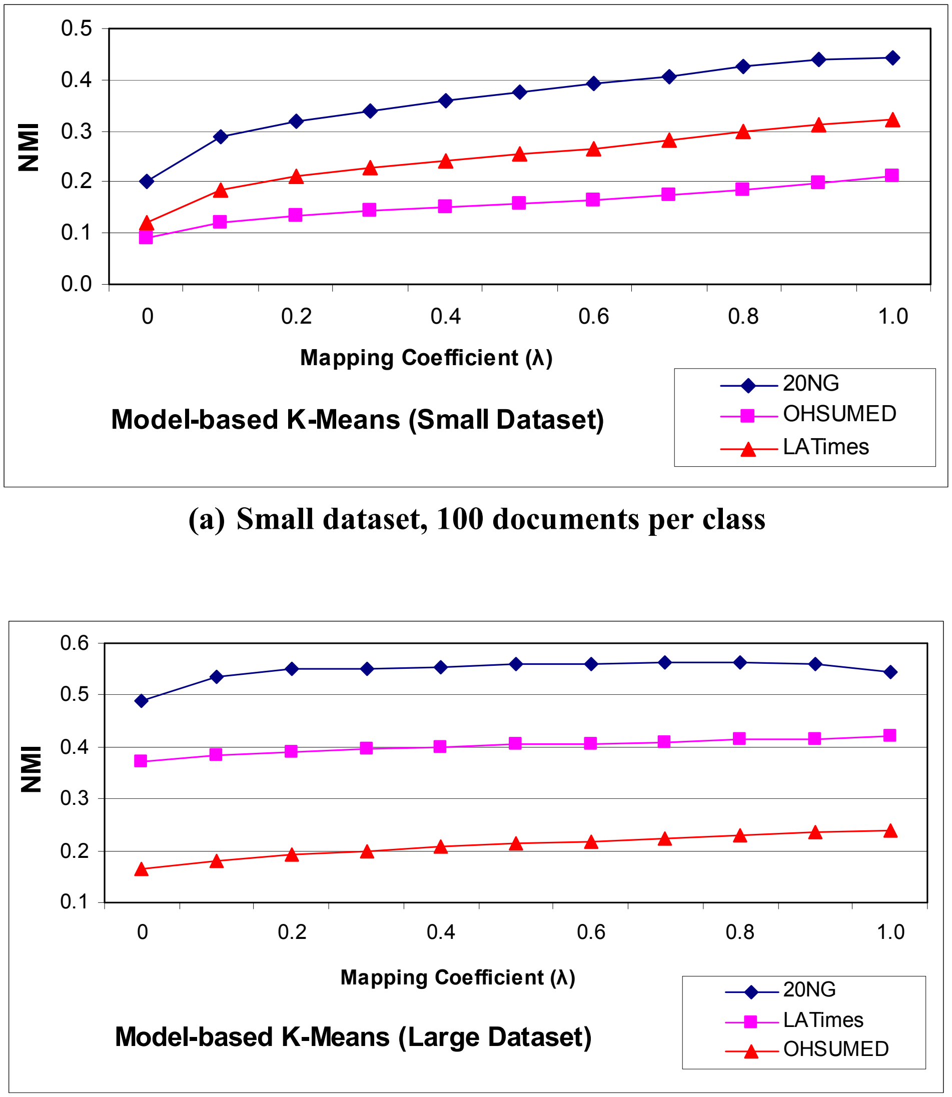
We compare the effectiveness of spherical k-means to the model-based k-means using semantic smoothing. The semantic smoothing has been empirically proven to be more effective than Laplace smoothing and background smoothing, but the comparison to other state-of-the-art clustering approaches remains unclear. Spherical k-means is one of the most effective clustering approaches to text clustering (Dhillon and Modha, 2001), and its comparison to semantic smoothing is shown in Table 6.6. Since both have two variants in our experiment, we compare the best results of these two clustering approaches. The semantic smoothing significantly outperforms spherical k-means on all collections (both small dataset and large dataset). This shows the effectiveness of the semantic smoothing approach to text clustering.

Table 6.6: The comparison of the best semantic smoothing results (CISS or CSSS) to the best spherical k-means results (NTF or TF-IDF).

Collection	Small dataset			Large dataset		
	spkmeans	mkmeans	change	spkmeans	mkmeans	change
20NG	0.391	0.476	**21.7%	0.506	0.571	**12.7%
OHSUMED	0.172	0.227	**31.9%	0.232	0.239	2.8%
LATimes	0.200	0.332	**66.5%	0.349	0.420	**20.2%

6.4.5 Tuning of Parameters

In the experiment of text classification and text retrieval (Zhou et al., 2006), the mapping coefficient seems to be empirically optimal in the range of 0.3~0.5, since too small a weight cannot take advantage of semantic smoothing and too large a weight may cause information loss. The optimal pattern for the mapping coefficient in the setting of text clustering is simpler, as shown in Table 6.7 and Figure 6.2. All collections achieve the best result when the mapping coefficient is set to one (i.e. the cluster model is a purely semantic mapping model).



(a) Small dataset, 100 documents per class

(b) Large dataset, all documents are used for clustering

Figure 6.2: The variance of the NMI with the change of the mapping coefficient, which controls the influence of the mapping component in the mixture model.

Table 6.7: The variance of the NMI with the change of the mapping coefficient (λ). The clustering algorithm is the model-based k-means with context-sensitive semantic smoothing (CSSS).

Lambda	Small Datasets			Large Datasets		
	20NG	OHSUMED	LATimes	20NG	LATimes	OHSUMED
0	0.201	0.090	0.122	0.489	0.371	0.165
0.1	0.287	0.120	0.185	0.536	0.383	0.181
0.2	0.318	0.134	0.211	0.549	0.389	0.192
0.3	0.340	0.144	0.227	0.552	0.397	0.200
0.4	0.361	0.151	0.241	0.554	0.400	0.207
0.5	0.377	0.158	0.254	0.560	0.407	0.215
0.6	0.394	0.165	0.265	0.559	0.404	0.218
0.7	0.407	0.173	0.281	0.562	0.408	0.224
0.8	0.425	0.183	0.298	0.564	0.413	0.229
0.9	0.440	0.197	0.314	0.560	0.416	0.235
1.0	0.441	0.212	0.322	0.545	0.420	0.239

Model-based k-means has serious overfitting problems. Many documents tend to group into a few large clusters and quickly converge at local maxima. The semantic smoothing is very effective in helping those small clusters to grow and jump out of the local maxima. This may explain why the best results are often achieved when the semantic mapping component is fully used, even though the full use of the mapping component may bring some information loss.

6.5 Conclusion

Model-based k-means text clustering is very similar to bayesian text classification except that the former does not require labeled texts for training whereas the latter requires labeled texts. Most findings of topic signature language models (i.e., semantic smoothing) in the settings of text classification are also found in the settings of text clustering. First of all, semantic smoothing performed significant better than two baselines, Laplace smoothing and background smoothing. Second, the model-based k-means with semantic smoothing also significantly outperform spherical k-means, which has been proven to be one of the most effective approaches to text clustering. Third, semantic smoothing is a robust technique. Semantic mapping knowledge learned from one collection could be effectively used to cluster another collection of documents. For example, when semantic knowledge learned from TDT2 was used to cluster LATimes articles, the performance was still better than model-based k-means with Laplacian smoothing and background smoothing as well as spherical k-means.

However, the topic signature language model brings extra advantages to the model-based k-means text clustering probably because it solves the overfitting problem of k-means algorithm. K-means is an EM-styled climbing algorithm and achieves local maxima rather than global maxima. In other words, a good part of documents probably falsely grouped into a few large clusters simply because a large cluster tends to share more common information to a single document than a small cluster. The topic signature language model introduces an extra semantic mapping component in addition to the unigram component. Consequently, the clustering algorithm is somehow transformed to concept clustering and then mapping each document into concept groups. Thus, the overfitting problem was dramatically relaxed. Unlike information retrieval and text classification which has the optimal mapping coefficient around 0.3~0.5, the task of text clustering achieved the best performance when the mapping coefficient is set to 1.0 in which the text clustering simply becomes the concept (topic signature) clustering.

In the task of information retrieval and text classification, the semantic smoothing will degrade the performance if the mapping coefficient is too large, i.e. overusing the semantic mapping component. The semantic smoothing shows much more robustness in the task of text clustering. It always beats the baseline smoothing methods when the mapping coefficient ranges from 0 to 1. Besides, the topic signature language model presents much more effectiveness in text clustering than in text classification. The model does not work when the training dataset is large in text classification. But the model not only makes considerable improvement on small dataset clustering, but also on large dataset clustering that is not supposed to have serious data sparse problems.

In the task of information retrieval and text classification, the context sensitive semantic smoothing (CSSS) is a little bit more effective than the context insensitive semantic smoothing (CISS). However, CISS performs slightly more effective than CSSS in the task of text clustering, when the collection to cluster is small, and two algorithms performs similarly when the collection to cluster is large. Again, we attribute this phenomenon to the overfitting issue of k-means. A small collection contains two few number of multiword phrases or ontological concepts. Therefore, word clustering makes much more sense than phrase clustering and concept clustering. In other words, CISS is more effective in handling overfitting issue than CSSS.

CHAPTER 7: CONCLUSIONS AND FUTURE WORK

We proposed a novel semantics-based language model called topic signature language model for information retrieval, text classification, and text clustering. The core idea of the topic signature language model is to identify topic signatures such as multiword phrases and ontology-based concepts in documents, and then statistically map topic signatures to single-word features for model smoothing purposes. According to whether the topic signature itself is context sensitive, the smoothing method is further divided into context-sensitive semantic smoothing (CSSS) and context-insensitive semantic smoothing (CISS). The semantic mapping from multiword phrases, ontology-based concepts to single-word features, is viewed as CSSS, and the word-word semantic mapping is considered CISS.

7.1 The Summary of Model Effectiveness

7.1.1 The Comparison of Applications

In ad hoc information retrieval, each document is considered a document language model. Because a document is usually short, there is a serious data sparsity issue of estimating the language model for each document. The semantic smoothing with whichever topic signatures (i.e., single-word terms, multiword phrases, or ontological concepts) significantly outperformed background smoothing approaches. The language model with semantic smoothing also beat famous Okapi models. CSSS performs slightly better than CISS in terms of both recall and average precision.

The effectiveness of semantic smoothing for Bayesian text classification depends on the data sparsity. In general, the sparser the data, the more effective the semantic smoothing is. When the size of training documents is small, the Bayesian classifier with semantic smoothing not only outperforms the classifiers with background smoothing and Laplace smoothing, but it also beats the state-of-the-art active learning classifiers and SVM classifiers. With the increase of training documents, the gap among the three smoothing methods is decreasing. This finding is of great practical value because it is expensive to get the labeled documents for real applications. CSSS performs slightly more effectively than CISS on the task of text classification. But if the number of training documents is too small, say, only one or two, CISS runs more effectively than CSSS, mostly because too few extracted context-sensitive topic signatures may misrepresent the topics associated with the training documents. However, this also could be an advantage of CSSS over CISS. A document contains fewer context-sensitive topic signatures (e.g., multiword phrases or concepts) than words on average. Thus, CSSS needs much less time complexity than CISS during semantic mapping.

Semantic smoothing also wins significant gain over Laplace smoothing and background smoothing for model-based k-means text clustering. The gained improvement in text clustering is even larger than in text classification. It not only makes considerable improvement on small dataset clustering but also on large dataset clustering, which is not supposed to have serious data sparse problems. Model-based k-means is a sort of EM algorithm and has serious overfitting problem. It tends to force most documents into a few large clusters and quickly converges on local maxima. Semantic smoothing helps small clusters grow and jump out of local maxima. With semantic smoothing, small clusters still have considerable chance to absorb documents from large clusters, because semantic mapping makes small clusters share much more significant terms with documents in other clusters. Unlike in the task of text classification, CISS looks a bit more effective than CSSS in the task of text clustering when the dataset to cluster is small. Documents contain much more number of words than context-sensitive topic signatures. Thus, CISS may be more effective in overcoming the overfitting problem. But again, CISS runs much slower than CSSS. The model-based k-means with semantic smoothing also significantly outperform spherical k-means, which has been proven to be one of the most effective approaches to text clustering.

Semantic smoothing uses a mixture language model with the mapping coefficient to control the influence of the two components, a simple language model and a semantic mapping model. The optimization of the mapping coefficient is still an ongoing problem. In the application of text classification, we proposed an automatic parameter tuning method which computed the optimal mapping coefficient by maximizing the generative probability of the testing documents. However, this approach is not robust. Sometimes the estimated parameter is quite close to the optimal value, but sometimes is quite far. This is also the problem for mixture language modeling approach to text retrieval. Fortunately, first of all, the proposed semantic smoothing is quite robust; it beats the baseline smoothing methods in a wide range setting of the mapping coefficient. Second, there are rules of thumb available to the tuning of the mapping coefficient. For information retrieval, the mapping coefficient between 0.3~0.5 always achieves good results. In the case of text classification, if data are very sparse, set the mapping coefficient to 0.3~0.5; decrease the value when data become less sparse; when sufficient training data are provided, stop using semantic smoothing. In the case of text clustering, the rule is even simpler; setting the mapping coefficient to 0.8~1.0 always gets good results.

Table 7.1: The summary of the effectiveness of the topic signature language models for different applications.

Applications	Small (Training) Dataset	Large (Training) Dataset	Optimal Mapping Coefficient
Retrieval	Very effective	Not applicable	0.3-0.5
Classification	Very effective	Not effective	0.3-0.5
Clustering	Very effective	Effective	0.8-1.0

7.1.2 The Comparison of Domains

We applied the topic signature language models to two domains in the thesis. One is the general news domain. The other is the specialized biomedical domain. The new model improved the performance of information retrieval, text classification, and text clustering on both domains. However, the model in the domain of biomedical literature performed slightly better than in the domain of news. The domain difference in effectiveness could be attributed to two sources.

First, biomedical literature contains a large number of multiword terms (e.g., high blood pressure, breast cancer) and one term often has many synonyms. This fact makes the ontological concept representation meaningful. A meaningful concept will never be broken down into several separate words; several synonyms will be represented by the same concept identities.

Second, the domain ontology benefited from the extraction of high quality topic signatures. We used statistical approaches to automate the extraction of multiword phrases in the news domain. Statistical approaches only extracted frequently occurring phrases, and some of them were even noisy. On the contrary, we used UMLS as the dictionary to extract concepts from biomedical literature. The dictionary-based extraction approach does not have statistical constraints and can extract more meaningful topic signatures than statistical approaches.

7.1.3 Knowledge Reusability

In the thesis, we also evaluated the effectiveness of external knowledge. In other words, we learned semantic knowledge from one collection and applied it to another collection. The experiment results showed that the topic signature language model with external semantic knowledge worked very well in general. In the experiment of text classification and clustering, it still outperformed background smoothing and Laplace smoothing. In the domain of biomedical literature, the external knowledge even achieved a slightly better result than the internal knowledge. As a rule of thumb, the effectiveness of external knowledge depends on the coverage of topic signatures, that is, the percentage of topic signatures of the testing collection that also appears in the training collection. The higher the coverage, the more effective the model is. In practice, the coverage of topic signatures will not be a serious issue because we can linearly merge semantic knowledge learned from multiple collections, which greatly raises the coverage of topic signatures.

7.2 The Comparisons of Three Topic Signatures

We have introduced three types of topic signatures in the thesis. They are multiword phrases, ontology-based concepts, and single-word terms. In this section, we give a brief comparison. The use of topic signatures in ad hoc information retrieval, text classification, and clustering involves three stages: (1) the extraction of topic signatures, (2) the estimation of semantic mapping, and (3) the incorporation of semantic mapping into language models. In the first stage, single-word term extraction is the easiest. The accuracy of automated multiword phrase extractions is acceptable. For example, Xtract, the one used in this thesis, has an accuracy of 80%. But it is much less efficient than word extraction. The extraction of concepts needs domain ontology. This is a limitation because not all domains have ontology available. Besides, the mapping of ontological concepts is not a trivial task. In our case, the F-score for the MaxMatcher, which can extract UMLS concepts from texts, is about 70%. Compared to multiword phrase extraction, however, one can extract more concepts than multiword phrases because ontological concept extraction has no statistical constraints.

The semantic mapping between single-word terms is less effective and efficient than multiword phrases and ontological concepts. First, because a single-word term is unable to incorporate contextual information into the semantic mapping procedure, the mapping result is fairly general and contains mixed topics. Second, a typical document contains a larger number of words than ontological concepts and multiword phrases and, thus, takes more time to get the parameter estimation because it involves the calculation of the co-occurrence matrix, which has the time complexity in proportion to the square of average document length.

Table 7.2: The summary of the comparison among three types of topic signatures. P, C, and W denote multiword phrases, ontology-based concepts, and single-word terms, respectively. The signs ">" and "<" mean "better than" and "worse than", respectively.

Tasks	Efficiency	Effectiveness
Extraction	P,C<W	C<P<W
Semantic Mapping	P,C>W	P,C>W
Information Retrieval	P>C>W	C,P>W
Text Classification	P>C>W	C, P>W
Text Clustering	P>C>W	C,P<W

In the stage of online semantic smoothing, the topic signature of single-word terms has the highest time complexity because the number of single-word terms is usually much higher than the number of multiword phrases or ontological concepts. During semantic smoothing, the complexity is in proportion to the number of topic signatures for mapping. With respect to the effectiveness of the three types of topic signatures, it is difficult to predict. Several factors determine the effectiveness of semantic smoothing, for example, the sparsity of data, the representative of extracted topic signatures, and the consistency of single-word terms in a corpus. However, in general, we have the following rules. In the application of information retrieval, multiword phrases and ontology-based concepts are more effective than single-word terms. For text classification, multiword phrases and ontology-based concepts are slightly more effective than single-word terms. But when the number of training documents is too small, say, one or two documents per class, single-word terms are more effective than the other two. For text clustering, single-word terms seem more effective than the other two on small datasets and comparable to the other two on large datasets.

7.3 The Comparison to Other Models

The statistical translation model and latent topic models, such as LDA and pLSI, are two representative language models that address the issue of utilizing semantic knowledge for language model smoothing. We would like to summarize the strengths and weaknesses of the topic signature language model compared to the aforementioned two models in the aspects of data acquisition, scalability, reusability, and complexity.

The statistical translation model requires paired corpora for training. For example, it uses large numbers of query-document pairs to train the translation model. It is often very difficult or expensive to collect such paired training data. The latent topic models and the topic signature language model use co-occurrence data, which is easy to collect. However, the topic signature language model has to extract the topic signatures prior to semantic mapping estimates. For general domains such as news collection, it is quite effective to use some automated algorithm (e.g., Xtract) to extract topic signatures such as multiword phrases. For some specific domains such as biomedical literature, we have to use domain ontology to extract meaningful topic signatures. This is the limitation of the topic signature language model.

Both the statistical translation model and latent topic models estimate all model parameters simultaneously. This means the parameter space will be in proportion to the number of training documents as well as to the size of the word space. In other words, both models are not scalable to huge collections. On the contrary, the topic signature language model estimates semantic mapping individually for each topic signature and thus is highly scalable to large collections.

The latent topic models will learn semantic profiles for each latent topic themes. Since the topic theme is latent and abstract, it is difficult to apply latent topics learned in one collection to another. Some latent topic models such as pLSI can't estimate the distribution of latent topics in a new document because it does not have a document model. Some latent topic models such as LDA include a document model and are able to estimate the distribution of latent topics in new documents. But it assumes the content of the new document should be similar to the training collections. Both translation models and topic signature language models result in semantic mappings from explicit words or topic signatures to words. Since it is straightforward to extract explicit words or topic signatures in new documents, the semantic mapping knowledge learned from training data can be easily applied to new documents and collections. However, the translation model results in semantic mapping from single word to single word, that is, it is unable to incorporate contextual information and word sense into the translation procedure. As we know, individual words without context could be very ambiguous. Therefore, it may be problematic to apply translation knowledge learned in one collection to another. The topic signature language model does not have this issue if the topic signature itself is context sensitive.

The complexity of the three models in the testing stage is of the same magnitude. In the testing stage, the three models will statistically map words, topic signatures, and latent topics to individual words, respectively. The complexity is thus subject to the number of words, topic signatures, and latent topics in a document. In practice, the three numbers are in the same magnitude.

7.4 The Contribution of the Thesis

The contribution of the paper is five-fold. First, we developed a novel topic signature language model that could incorporate semantic knowledge into a traditional language model. This new language model can be easily applied to many text applications such as information retrieval, text classification, and text clustering.

Second, we developed an efficient co-occurrence–based semantic knowledge learning method. This method does not require labeled training data. Instead, it uses co-occurrence data, which is quite cheap to collect. It learns semantic mapping knowledge individually for each topic signature and, therefore, is highly scalable to large datasets. The knowledge unit is explicit topic signatures such as multiword phrases and ontological concepts, rather than latent topic themes. The topic signature somehow self-contains contextual information, and the mapping results are usually specific and accurate. Thus, the semantic knowledge of topic signatures can be reused in new documents and collections.

Third, we applied the topic signature language model to three applications (information retrieval, text classification, and text clustering) and evaluated these applications on two different domains (news and biomedical literature). In general, it was safe to conclude that the topic signature language model significantly outperformed the baseline language models and the majority of the state-of-the-art nonlanguage modeling approaches on all these applications and domains. The summarization in Section 7.1 gave users high level guidance regarding how to use the topic signature language model properly. For example, how to set the optimal mapping coefficient of the model; when one can use the model and when one cannot.

Fourth, we compared and contrasted the effectiveness and the efficiency of three types of topic signatures: individual words, multiword phrases, and ontological concepts. A short summary was described in Section 7.2. The comparison gave guidance regarding how to choose topic signatures for different applications and domains.

Last, we developed the dragon toolkit (http://dragon.ischool.drexel.edu) for academic use. The dragon toolkit implemented the topic signature extraction approaches; semantic mapping knowledge learning methods; and the topic signature language models for information retrieval, text classification, and text clustering. It was written in Java, the platform-independent language, and is free to public. Since its first release in April 2007, the dragon toolkit has been downloaded by more than one thousand researchers or research groups worldwide, in the community of information retrieval and text mining.

7.5 Future Work

The topic signature language model is a framework rather than a specific algorithm for semantics-based language model smoothing. It can be extended in many aspects. First, it is able to accommodate more types of topic signatures. For example, significant word pairs or concept pairs can be used as topic signatures as well. We have tested this topic signature in information retrieval and proven it was effective in improving the accuracy of information retrieval. The complexity of this topic signature, however, still needs to be improved.

Second, other types of semantic knowledge could be incorporated into the topic signature language model. In addition to the semantic mapping knowledge, a semantic category of topic signatures is also effective in smoothing language models, especially for bigram language models. The core idea is that words, phrases, and concepts in the same semantic category should share many similar language characteristics.

Third, we plan to apply the topic signature language models to more text applications. Because of time constraints, we only evaluated the model on three applications (information retrieval, text classification, and text clustering). We believe future studies will show that the model is also effective in improving the performance of applications like speech recognition, question classification, and content-based image annotations.

LIST OF REFERENCES

[Al-Mubaid and Umair, 2006] Al-Mubaid, H. and Umair, S., "A New Text Categorization Technique Using Distributional Clustering and Learning Logic," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, Vol. 18, no.9, pp. 1156-1165, 2006

[Allwein et al., 2000] Allwein, E.L., Schapire, R.E., and Singer, Y., "Reducing multiclass to binary: A unifying approach for margin classifiers," *Journal of Machine Learning Research*, 1:113–141, 2000.

[Bahl et al., 1983] Bahl, L.R. , Jelinek, F., and Mercer, R.L., "A maximum-likelihood approach to continuous speech recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1983, 179-190

[Bai et al., 2005a] Bai, J., Song, D., Bruza, P., Nie, J.Y., and Cao, G., "Query Expansion Using Term Relationships in Language Models for Information Retrieval," In *Proceedings of the ACM 14th Conference on Information and Knowledge Management (CIKM)*, November 2005, Bremen, Germany.

[Bai et al., 2005b] Bai, J., Nie, J.-Y., and Cao, G., "Integrating Compound Terms in Bayesian Text Classification," *Web Intelligence 2005*, France.

[Baker and McCallum, 1998] Baker, D. and McCallum, A., "Distributional clustering of words for text classification," *21st ACM International Conference on Research and Development in Information Retrieval (SIGIR'98)*, 1998, pp. 96-103.

[Banerjee et al., 2003] Banerjee, A., Dhillon, I.S., Ghosh, J. and S. Sra, "Generative Model-based Clustering of Directional Data," *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 19-28, August 2003

[Banerjee and Ghosh, 2002] Banerjee, A. and Ghosh, J., "Frequency sensitive competitive learning for clustering on high-dimensional hperspheres," *Proc. IEEE Int. Joint Conference on Neural Networks*, pp. 1590-1595.

[Berger and Lafferty, 1999] Berger, A. and Lafferty J., "Information Retrieval as Statistical Translation," In *Proceedings of the 22nd ACM SIGIR Conference on Research and Development in IR*, 1999, pp.222-229.

[Blei et al., 2003] Blei, D., Ng, A. and Jordan, M., "Latent Dirichlet allocation," *Journal of machine Learning Research*, 3, 2003, pp 993-1022.

[Bloehdorn and Hotho, 2004] Bloehdorn, S. and Hotho, A., "Boosting for text classification with semantic features," *In the Workshop on Text-based Information Retrieval (TIR-04) at the 27th German Conference on Artificial Intelligence*, Sep 2004.

[Bunescu et al., 205] Bunescu, R., Ge, R., Kate, J.R., Marcotte, M.E., Mooney, J.R., Ramani, A., and Wong, Y.-W., "Comparative Experiments on Learning Information Extractors for Proteins and their Interactions," *Artificial Intelligence in Medicine*, 2005, 33(2), pp. 139-155

[Cai and Hofmann, 2003] Cai, L. and Hofmann, T., "Text Categorization by Boosting Automatically Extracted Concepts," *26th Annual International ACM SIGIR Conference on Research & Development on Information Retrieval*, 2003.

[Cao et al., 2005] Cao, G., Nie, J.Y., and Bai, J., "Integrating Word Relationships into Language Models," *Proceedings of the 28th annual international ACM SIGIR conference on Research and Development in Information Retrieval*, 2005, pp. 298 - 305

[Chang et al., 2004] Chang, J.T., Schütze, H., and Altman, R.B., "GAPSCORE: finding gene and protein names one word at a time," *Bioinformatics*, Vol. 20, No. 2, pp. 216-225, 2004.

[Chagoyen et al., 2006] Chagoyen, M., Carmona-Saez, P., Shatkay, H., Carazo, J.M., and Pascual-Montano, A., "Discovering semantic features in the literature: a foundation for building functional associations," *BMC Bioinformatics*, 2006, 7(1):41

[Chen and Goodman, 1998] Chen, S. and Goodman, J., "An empirical study of smoothing techniques for language modeling," Technical Report TR-10-98, Computer Science Group, Harvard University.

[Chiang and Yu, 2005] Chiang, J.-H. and Yu, H.-C., "Literature extraction of protein functions using sentence pattern mining," *IEEE Transactions on Knowledge and Data Engineering*, 17(8), Aug. 2005 Page(s):1088 – 1098

[Collier et al., 2000] Collier, N., Nobata, C., and Tsujii, J., "Extracting the names of genes and gene products with a Hidden Markov Model," *Proc. COLING 2000*, 201--207, 2000

[Croft et al., 1991] Croft, W.B., Turtle, H.R., and Lewis, D.D., "The use of phrases and structured queries in information retrieval," In *Proceedings of the 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1991, pp. 32--45

[Cunningham, 2002] Cunningham, H., GATE, "A General Architecture for Text Engineering," *Computers and the Humanities*, 2002, Vol. 36, pp. 223-254

[Deerwester et al., 1990] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, Richard Harshman, "Indexing by Latent Semantic Analysis," *Journal of the American Society of Information Science*, 1990, 41(6): 391- 407

[Dempster et al., 1977] Dempster, A.P., Laird, N.M., and Rubin, D.B., "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, 1977, 39: 1-38.

[Dhillon and Modha, 2001] Dhillon, I.S. and Modha, D.S., "Concept Decompositions for Large Sparse Text Data Using Clustering," *Machine Learning*, 42(1):143-175, 2001

[Ding et al., 2003] Ding, J., Berleant, D., Xu, J., and Fulmer, A.W., "Extracting Biochemical Interactions from MEDLINE Using a Link Grammar Parser," In *the 15th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'03)*, 2003.

[Fagan, 1987] Fagan, J., "Experiments in Automatic Phrase Indexing for Document Retrieval: A Comparison of Syntactic and Non-Syntactic Methods," Ph.D. Thesis, Technical Report 87-868, Cornell University, Computer Science Department, 1987.

[Fukuda et al., 1998] Fukuda, K., Tamura, A., Tsunoda, T., and Takagi, T., "Toward information extraction: Identifying protein names from biological papers," In

Proceedings of Pacific Symposium on Biocomputing, pages 707--718, Maui, Hawaii, January 1998.

[Gao et al., 2004] Gao, J., Nie, J.-Y., Wu, G. and Cao, G., "Dependence language model for information retrieval," In *SIGIR-2004*, Sheffield, UK, July 25-29

[Good, 1953] Good, I.J., "The population frequencies of species and the estimation of population parameters," *Biometrika*, 1953, 40 (3 and 4): 237-264

[Grefenstette, 1992] Grefenstette, G., "Use of syntactic context to produce term association lists for information retrieval," *Proceedings of the 15th annual international ACM SIGIR conference on Research and Development in Information Retrieva*l, 1992, pp. 89 - 97

[Hao et al., 2005] Hao, Y., Zhu, X., Huang, M., and Li, M., "Discovering patterns to extract protein-protein interactions from the literature: Part II," *Bioinformatics*, 2005, 21(15): 3294-3300

[Harabagiu and Lacatusu, 2005] Harabagiu, S. and Lacatusu, F., "Topic themes for multi-document summarization," *2005 ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'05)*, 2005, pp. 42-48

[Hatzivassiloglou and McKeown, 1997] Hatzivassiloglou, V. and McKeown, K.R., "Predicting the Semantic Orientation of Adjectives," *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL-EACL 97),* 1997, pp.174-181

[Hersh et al., 2004] Hersh, W. et al., "TREC 2004 Genomics Track Overview," the *Thirteenth Text Retrieval Conference*, 2004.

[Hersh et al., 2005] Hersh, W. et al., "TREC 2005 Genomics Track Overview," the *Fourteenth Text Retrieval Conference*, 2005.

[Hirschman et al., 2002] Hirschman, L., Park, J., Tsujii, J., Wong, L., and Wu, C., "Accomplishments and challenges in literature data mining for biology," *Bioinformatics*, 18:1553--1561, 2002

[Hoffman, 1999] Hoffman, T., “Probabilistic latent semantic indexing,” *1999 ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99)*, 1999, pp. 50-57

[Jeffreys, 1948] Jeffreys, H., “*Theory of Probabilities*,” Clarendon Press, Oxford, Second Edition, 1948

[Jelinek and Mercer, 1980] Jelinek, F. And Mercer, R. “Interpolated estimation of markov sourceparameters from sparse data,” *Pattern Recognition in Practice*, E. S. Gelsema and L. N. Kanal, Eds., 1980, pp. 381–402.

[Jelinek, 1990] Jelinek, F., “Self-Organized Language Modeling for Speech Recognition,” *WeiBel A and Lee K-F, Eds., Readings in Speech Recognition*, Morgan Kaufmann, Los Altos, CA, 1990, pp. 450-505.

[Jinsen et al., 2006] Jinsen, L., Saric, J., and Bork, P., “TextPresso: An Ontology-based Information Retrieval and Extraction System Biological Literature,” *Nature Reviews Genetics*, 2006, Vol. 7, 119-129

[Jin et al., 2002] Jin, R., Hauptmann, A., and Zhai, C., “Title Language Model for Information Retrieval,” *2002 ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'02)*, 2002, pp. 42-48

[Joachims, 1998] Joachims, T., “Text categorization with support vector machines: Learning with many relevant features,” In *Proceedings of European Conference on Machine Learning*, pages 137-142, 1998.

[Johnson, 1932] Johnson, “W.E. Probability: deductive and inductive problems,” *Mind*, 41:421-423, 1932

[Karypis, 2002] Karypis, G., “CLUTO – a clustering toolkit,” Dept. of Computer Science, University of Minnesota, 2002.

[Kaufman and Rousseuw, 1990] Kaufman, L. and Rousseuw, P.J., “*Finding Groups in Data:* an Introduction to Cluster Analysis,” John Wiley and Sons, 1990.

[Kankar et al., 2002] Kankar,P., Adak, S., Sarkar, A., Murali, K. and. Sharma, G., "MedMeSH Summarizer: Text Mining for Gene Clusters," *Proceedings of the Second SIAM International Conference on Data Mining*, Arlington (SDM'02), 2002

[Lafferty and Zhai, 2001] Lafferty, J. and Zhai, C., "Document Language Models, Query Models, and Risk Minimization for Information Retrieval," In *Proceedings of the 24th ACM SIGIR Conference on Research and Development in IR*, 2001, pp.111-119.

[Lesk, 1986] Lesk, M., "Automatic Sense Disambiguation: How to Tell a Pine Cone from and Ice Cream Cone," *Proceedings of the SIGDOC'86 Conference, ACM*, 1986.

[Lewis, 1990] Lewis, D.D., "Representation quality in text classification: An introduction and experiment," In *Proceedings of a Workshop on Speech and Natural Language*, Hidden Valley, Pennsylvania, 1990.

[Li and Abe, 1998] Li, H. and Abe, N., "Word Clustering and Disambiguation Based on Co-occurrence Data," *Proceedings of COLING-ACL 98*, pp. 749-755.

[Lidstone, 1920] Lidstone, G.J., "Note on the general case of the Bayes-Laplace formula for inductive or a posteriori probabilities," *Transactions of the Faculty of Actuaries*, 8:182-192, 1920

[Liu and Croft, 2001] Liu, X. and Croft, W.B., "Cluster-based retrieval using language models," In *Proceedings of the 24th ACM SIGIR Conference on Research and Development in IR*, 2001, pp.186-193.

[MacKay and Peto, 1995] MacKay, D.J.C. and Peto, L., "A hierarchical Dirichlet language model," *Natural Language Engineering*, 1995, 1(3), 1-19.

[McCallum and Nigam, 1998] McCallum, A. and Nigam, K., "A comparison of event models for naive Bayes text classification," AAAI Workshop on Learning for Text Categorization, 1998, pp. 41–48.

[McQueen, 1967] McQueen, J., "Some Methods for classification and Analysis of Multivariate Observations," *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, 1967, Berkeley, University of California Press, 1:281–297

[Mei and Zhai, 2005] Mei, Q. and Zhai, C., "Discovering Evolutionary Theme Patterns from Text -- An Exploration of Temporal Text Mining," *Proceedings of the 2005 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining , (KDD'05)*, 2005, pp.198–207

[Miller et al., 1999] Miller, D., Leek, T., and Schwartz M.R., "A Hidden Markov Model Information Retrieval System," In *Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval*, 1999, pp 214-221.

[Miller, 1995] Miller, G. A., "WordNet: a lexical database for English," *Communications of the ACM*, 1995, 38(11), pp. 39–41

[Mooney and Bunescu, 2005] Mooney, R. J. and Bunescu, R., "Mining Knowledge from Text Using Information Extraction," *SIGKDD Explorations* (special issue on Text Mining and Natural Language Processing), 7, 1 (2005), pp. 3-10.

[Müller et al., 2004] Müller, H.-M., Kenny, E.E., and Sternberg, W.P., "TextPresso: An Ontology-based Information Retrieval and Extraction System Biological Literature," *PLoS Biology*, 2004 2(11): 1984-1998

[Ney et al., 1994] Ney, Hermann, Ute Essen, and Reinhard Kneser, "On structuring probabilistic dependences in stochastic language modeling," *Computer, Speech, and Language*, 1994, 8:1-38.

[Nigam et al., 2000] Nigam,K., McCallum, A., Thrun, S., Mitchell, T., "Text Classification from Labeled and Unlabeled Documents using EM," *Machine Learning*, Volume 39 , Issue 2-3 (May-June 2000), pp103-134

[Peng et al., 2004] Peng, F., Schuurmans, D. and Wang, S., "Augmenting naive bayes classifiers with statistical language models," *Information Retrieval*, 7(3-4):317-345, 2004.

[Pickens and Croft, 2000] Pickens, J. and Croft, W.B., "An exploratory analysis of phrases in text retrieval". In *RIAO2000 Conference Proceedings*, pp. 1179-1195, Paris, France.

[Ponte and Croft, 1998] Ponte, J. and Croft, W.B., "A Language Modeling Approach to Information Retrieval," In *Proceedings of the 21st ACM SIGIR Conference on Research and Development in IR*, 1998, pp.275-281.

[Quinlan, 1986] Quinlan, J.R., "Induction of decision trees," *Machine Learning*, 1(1): 81-106, 1986

[Ray and Craven, 2001] Ray,S. and Craven,M., "Representing sentence structure in hidden markov models for information extraction," *In Proceedings of the 17th international Joint Conference on Artificial Intelligence (IJCAI 2001)*, Morgan Kaufmann, pp. 1273–1279.

[Rasmussen, 1992] Rasmussen, E., "Clustering Algorithms," in W. Frakes and R. Baeza-Yates (Eds.), *Information Retrieval: Data Structure and Algorithms*, 1992, 419-442, Prentice Hall.

[Rijsbergen, 1979] C.J. Van Rijsbergen, *Information Retrieval*, Butterworths, London, Second edition, 1979.

[Rindfleisch et al., 2000] Rindfleisch, T.C., Tanabe, L., and Weinstein, J.N., "EDGAR: Extraction of Drugs, Genes and Relations from the Biomedical Literature", *Proceedings of Pacific Symposium on Bioinformatics*, Hawaii, USA, pp. 514-525, 2000.

[Robertson, 1993] Robertson, S.E. et al. "Okapi at TREC-4", *In the Fourth Text Retrieval Conference*, 1993.

[Schapire and Singer, 2000] Schapire, R.E. and Singer, Y., "Boostexter: A boosting-based system for text categorization," *Machine Learning*, 39(2/3):135{168, 2000.

[Sleator et al., 1993] Sleator, D. and Temperley D., "Parsing English with a Link Grammar," *Third International Workshop on Parsing Technologies*, 1993.

[Shen et al., 2006] Shen, D., Sun, J.-T., Yang, Q., and Chen, Z., "Text Classification Improved through Multigram Models," *In Proceedings of the ACM Fifteenth Conference on Information and Knowledge Management (CIKM 06)*, Arlington, USA. November 6-11, 2006.

[Smadja, 1993] Smadja, F., “Retrieving collocations from text: Xtract,” *Computational Linguistics*,” 1993, 19(1), pp. 143--177.

[Song and Bruza, 2003] Song, D. and Bruza P.D., “Towards Context-sensitive Information Inference,” *Journal of the American Society for Information Science and Technology (JASIST)*, 2003, Vol. 54, 321-334.

[Song et al., 2004] Song, Y.-I., Kim, S.-B., and Rim, H.-C., “Terminology Indexing and Reweighting methods for Biomedical Text Retrieval,” In *Proceedings of the SIGIR'04 Workshop on Search and Discovery in Bioinformatics*, Sheffield, UK, ACM, July 2004.

[Steinbach et al., 2000] Steinbach, M., Karypis, G., and Kumar, V., “*A Comparison of document clustering techniques*,” Technical Report #00-034, Department of Computer Science and Engineering, University of Minnesota, 2000.

[Subramaniam et al., 2003] Subramaniam, L., Mukherjea, S., Kankar, P., Srivastava, B., Batra, V., Kamesam, P. and Kothari, R., “Information Extraction from Biomedical Literature: Methodology, Evaluation and an Application,” *In the Proceedings of the ACM Conference on Information and Knowledge Management*, New Orleans, Louisiana, 2003.

[Takamura and Matsumoto, 2002] Takamura, H. and Matsumoto, Y., “Two-dimensional Clustering for Text Categorization,” *Proceedings of Sixth Conference on Natural Language Learning (CoNLL-2002)*, 2002, pp. 29--35

[Tanabe and Wilbur, 2002] Tanabe, L. and Wilbur, W., “Tagging gene and protein names in biomedical text,” *Bioinformatics*, Vol. 18, No. 8, pp.1124-1132, 2002.

[Turtle, 1990] Turtle, H., “Inference Networks for Document Retrieval,” Ph.D. Thesis, University of Massachusetts, COINS Technical Report 90-92, 1990.

[Vapnik, 1995] Vapnik ,V.N., *The Nature of Statistical Learning Theory*, Springer, 1995.

[Wang and McCallum, 2005] Wang, X. and McCallum, A., “A Note on Topical N-grams,” *UMass Technical Report UM-CS-2005-071*, 2005

[Wei and Croft, 2006] Wei, X. and Croft, W.B., “LDA-based document models for ad-hoc retrieval,” In *Proceedings of the 29th ACM SIGIR Conference on Research and Development in IR*, pp. 178-185

[Wiener et al., 1995] Wiener, E., Pedersen, J.O., and Weigend, A.S., “A neural network approach to topic spotting,” In *Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval SDAIR*, 1995, pp. 317-332.

[Willett, 1988] Willett, P., “Recent trends in hierarchical document clustering: a critical review,” *Information Processing and Management*, 1988, 24(5): 577-597

[Wu and Hu, 2005] Wu, D. and Hu, X., “An Efficient Approach to Detect a Protein Community from a Seed,” in the *2005 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, pp135-141

[Xiong et al., 2005] Xiong, H., He, X., Ding, C., Zhang, Y., Kumar, V., and Holbrook, S.R., “Identification of Functional Modules in Protein Complexes via Hyperclique Pattern Discovery,” *Proc. of the Pacific Symposium on Biocomputing (PSB 2005)*, January 2005, pp 221-232.

[Yang, 1999] Yang, Y., “An evaluation of statistical approaches to text categorization,” *Information Retrieval*, 1(1-2):69-90, 1999.

[Yang and Liu, 1999] Yang, Y. and Liu, X., “A re-examination of text categorization methods,” *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pp 42--49, 1999.

[Yang and Pedersen, 1997] Yang, Y. and Pedersen, J.O., “A comparative study on feature selection in text categorization,” In *Proceedings of International Conference on Machine Learning*, 1997, pp. 412-420.

[Yetisgen-Yildiz and Pratt, 2005] Yetisgen-Yildiz, M. and Pratt, W., “The effect of feature representation on Medline document classification,” In *Proceedings of the American Medical Informatics Association Fall Symposium*, Washington D.C., 2005.

[Yi et al., 2003] Yi, J., Nasukawa, T., Bunescu, R.C., and Niblack, W., “Sentiment Analyzer: Extracting Sentiments about a Given Topic using Natural Language Processing

Techniques," *The Third IEEE International Conference on Data Mining(ICDM'03)*, 2003, pp. 427-434

[Yoo et al., 2006] Yoo I., Hu X., Song I-Y, "Integration of Semantic-based Bipartite Graph Representation and Mutual Refinement Strategy for Biomedical Literature Clustering," In *the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006, Philadelphia, PA, USA.

[Yoo and Hu, 2006] Yoo I., Hu X., "Clustering Large Collection of Biomedical Literature Based on Ontology-Enriched Bipartite Graph Representation and Mutual Refinement Strategy," 10th Pacific-Asia Conference, PAKDD 2006, Singapore, April 9-12, 2006, pp303-312.

[Zha, 2002] Zha, H., "Generic summarization and Keyphrase extraction using mutual reinforcement principle and sentence clustering", *Proceedings of the 25th Annual ACM SIGIR Conference*, Tampere, Finland (2002) 113—120.

[Zhai and Lafferty, 2001a] 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 Conference on Research and Development in IR*, 2001, pp.334-342.

[Zhai and Lafferty, 2001b] Zhai, C. and Lafferty, J., "Model-based Feedback in the Language Modeling Approach to Information Retrieval," In *Proceedings of the 10th International Conference on Information and Knowledge Management*, 2001, pp.403-410.

[Zhai and Lafferty, 2002] Zhai, C. and Lafferty, J., "Two-Stage Language Models for Information Retrieval," *2002 ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'02)*, 2002.

[Zhai et al., 2004] Zhai, C., Velivelli, A., and Yu, B, "A cross-collection mixture model for comparative text mining," *Proceedings of ACM KDD 2004 (KDD'04)*, 2004, pp. 743-748

[Zhao and Karypis, 2001] Zhao, Y. and Karypis, G, "*Criterion functions for document clustering: experiments and analysis*," Technical Report, Department of Computer Science, University of Minnesota, 2001.

[Zhang et al., 2006a] Zhang, X., Zhou, X., and Hu, X., "Semantic Smoothing for Model-based Document Clustering," in proceedings of the *IEEE International Conference on Data Mining (ICDM06)*, Dec. 18-22, 2006, Hong Kong, 1193-1198

[Zhang et al., 2006b] Zhang, X., Wu, D., Zhou, X., and Hu, X, "A Language Modeling Text Mining Approach to the Annotation of Protein Community", In *the 6th IEEE Symposium on Bioinformatics and Bioengineering* (*BIBE 2006*), 12-19

[Zhong and Ghosh, 2005] Zhong, S. and Ghosh, J., "Generative model-based document clustering: a comparative study," *Knowledge and Information Systems*, 8(3): 374-384, 2005.

[Zhou et al., 2004] Zhou, G.-D., Zhang, J., Su, J., Shen, D., and Tan, C.-L., Recognizing Names in Biomedical Texts: A Machine Learning Approach, *Bioinformatics*, 20(7), 1178-1190, 2004.

[Zhou et al., 2006a] Zhou, X., Hu, X., Lin, X., Han, H., and Zhang, X., "Relation-based Document Retrieval for Biomedical Literature Databases," *The 11th International Conference on Database Systems for Advanced Applications (DASFAA 2006)*, 12 - 15 April, 2006, Singapore, pp. 689-701

[Zhou et al., 2006b] Zhou, X., Zhang, X., and Hu, X., Using Concept-based Indexing to Improve Language Modeling Approach to Genomic IR, *The 28th European Conference on Information Retrieval (ECIR' 2006)*, 10 - 12 April, 2006, London, UK, pp. 444-455.

[Zhou et al., 2006c] Zhou, X., Hu, X., Zhang, X., Lin, X., and Song, I.-Y., "Context-Sensitive Semantic Smoothing for the Language Modeling Approach to Genomic IR," I*n the 29th Annual International ACM SIGIR Conference (ACM SIGIR 2006)*, Aug 6-11, 2006, Seattle, WA, USA

[Zhou et al., 2006d] Zhou, X., Zhang, X., and Hu, X., "MaxMatcher: Biological Concept Extraction Using Approximate Dictionary Lookup", In the *9th biennial The Pacific Rim International Conference on Artificial Intelligence (PRICAI 2006)*, Aug 9-11, 2006, Guilin, China, 1145-1149

[Zhou et al., 2007a] Zhou, X., Zhang, X., and Hu, X., "Semantic Smoothing of Document Models for Agglomerative Clustering," in the Twentieth International Joint Conference on Artificial Intelligence (*IJCAI 2007*), Jan. 6-12, 2007, India, 2922-2927

[Zhou et al., 2007b] Zhou, X., Hu, X. and Zhang, X., "Topic Signature Language Model for Ad hoc Retrieval," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 19 no 9, 1276-1287, Sept., 2007

[Zhou et al., 2007c] Zhou, X., Zhang, X., and Hu, X., "Dragon Toolkit: Incorporating Auto-learned Semantic Knowledge into Large-Scale Text Retrieval and Mining," In *proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007)*, October 29-31, 2007, Patras, Greece. 197-201

[Zhou et. al, 2008] X. Zhou, X. Zhang, and X. Hu, “Semantic Smoothing for Bayesian Text Classification with Small Training Data,” accepted by the *2008 SIAM International Conference on Data Mining* (*SDM2008*), April 24-26, Atlanta, Georgia (27%)

VITA

Contact

Xiaohua Zhou, xiaohua.zhou@drexel.edu, 215-840-6193

Education

- 2003.9~2008.12 College of Information Science & Technology (IST), Drexel University
  Philadelphia, PA 19104, USA, Ph.D. (expected in June 2008)
- 1999.9~2002.3 Shanghai Jiao Tong University (SJTU), Shanghai, 200030, China
  M.S. in Management Science and Engineering, 2002
- 1995.9~1999.7 Shanghai Jiao Tong University (SJTU), Shanghai, 200030, China
  B.S. in Automatic Control, B.A. in International Trade

Research Interest

Biomedical Literature Mining, Information Retrieval, Text Mining, Question Answering

Selected Publications

- **X. Zhou**, X. Hu, and X. Zhang, "Topic Signature Language Model for Ad hoc Retrieval," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 19 no 9, 1276-1287, Sept., 2007
- X. Zhang, X. Hu, and **X. Zhou**, "A Comparative Evaluation of Different Link Types on Enhancing Document Clustering," *SIGIR 2008*, 20-24 July 2008, Singapore (17%)
- **X. Zhou**, X. Zhang, and X. Hu, "Semantic Smoothing for Bayesian Text Classification with Small Training Data," *SDM 2008*, April 24-26, Atlanta, Georgia (27%)
- **X. Zhou**, X. Zhang, and X. Hu, "Semantic Smoothing of Document Models for Agglomerative Clustering," *IJCAI 2007*, Jan. 6-12, 2007, India, 2922-2927 (15.7%)
- **X. Zhou**, X. Hu, X. Zhang, and X. Shen, "A Segment-based Hidden Markov Model for Real-Setting Pinyin-to-Chinese Conversion," *CIKM 2007*, November 6-8, 2007, Portugal, 1027-1030 (26%)
- **X. Zhou**, X. Hu, X. Zhang, X. Lin, and I.-Y. Song, "Context-sensitive Semantic Smoothing for Language Modeling Approach to Genomic Information Retrieval", *SIGIR 2006*, 170-177 (18.5%)
- X. Zhang, **X. Zhou**, and Xiaohua Hu, "Semantic Smoothing for Model-based Document Clustering," *ICDM 2006*, 2006, Hong Kong, 1193-1198 (20%)

Academic Services

- Program Committee Member of 2007 IEEE DMIR Workshop and 2006 IEEE DMB Workshop
- Ad hoc Journal Reviewer: TKDE, KAIS, Bioinformatics, IJSOI
- External Referee: ER2004, eCOMO2004, DGOV2004, CAiSE2005, NSF ITR Program 2005

Software

- *Dragon Toolkit:* A Java-based Toolkit for Language Modeling, Information Retrieval and Text Mining (http://dragon.ischool.drexel.edu)

Honors and Awards

- Research Day Winners, Drexel University, 2006 and 2007
- Dean's Award (Research Day), IST, Drexel University, April 2006
- SIGIR Student Travel Award, August 2006