

# Semantic Smoothing for Bayesian Text Classification with Small Training Data

Xiaohua Zhou, Xiaodan Zhang, Xiaohua Hu  
College of Information Science & Technology, Drexel University  
xiaohua.zhou@drexel.edu, {xzhang, thu}@ischool.drexel.edu

## Abstract

Bayesian text classifiers face a common issue which is referred to as data sparsity problem, especially when the size of training data is very small. The frequently used Laplacian smoothing and corpus-based background smoothing are not effective in handling it. Instead, we propose a novel semantic smoothing method to address the sparse problem. Our method extracts explicit topic signatures (e.g. words, multiword phrases, and ontology-based concepts) from a document and then statistically maps them into single-word features. We conduct comprehensive experiments on three testing collections (OHSUMED, LATimes, and 20NG) to compare semantic smoothing with other approaches. When the size of training documents is small, the bayesian classifier with semantic smoothing not only outperforms the classifiers with background smoothing and Laplacian smoothing, but also beats the state-of-the-art active learning classifiers and SVM classifiers. In this paper, we also compare three types of topic signatures with respect to their effectiveness and efficiency for semantic smoothing.

## 1. Introduction

The task of text classification is to assign one or multiple pre-defined 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, newswire and scientific literature. In past decades, a large number of algorithms, including naïve bayes [13], k-nearest neighbor [22], support vector machines [11], boosting, decision trees [16] and neural network [21], 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 due to 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. Due to 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 some smoothing techniques which assign a

reasonable non-zero probability to those “unseen” terms. Laplacian smoothing, which simply add 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 [10].

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 [25] and two-stage smoothing [26] to smooth unigram language models. Because all these approaches are based on a background collection model, we refer to all of them as background smoothing in this paper. 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 term “auto” should return for the query “car” because both terms are semantically related. Following this intuitive idea, several semantic smoothing approaches [5] [20] [28] has 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 paper a topic signature based semantic smoothing method to address the aforementioned data sparsity problem. The idea of our semantic smoothing 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 doc about *rocket launch* to a given category whose training documents never explicitly present the topic of *rocket launch*, but contain many instances of “*space program*”. The definition of topic signatures will be given later in the paper.

The idea of using multiword phrase or n-grams for text classification is not new. However, to the best of our knowledge, it is the first time used for smoothing purpose in the setting of text classification. The majority of those work [3] [7] [12] [24] 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. [15] and [17] built n-gram and n-multigram language model 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 [1] [4]. 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 proposed semantic smoothing based 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 [30] and conduct comprehensive experiments on three collections, OHSUMED, LATimes, and 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 Laplacian smoothing, but also beats the state-of-the-art active learning classifiers [14] and SVM classifiers [11].

In summary, we make four main contributions in this paper. 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 amount of cooccurrence data, we use a cooccurrence-based algorithm to estimate semantic mappings, which dramatically reduce the cost of obtaining semantic knowledge. Third, we empirically prove that semantic smoothing is more effective than background smoothing and Laplacian 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 and clustering.

The rest of the paper is organized as follows: Section 2 describes the details of the semantic smoothing method for bayesian text classifiers. Section 3 introduces the datasets and protocols for evaluation. Section 4 presents the experimental results. Section 5 shows the result of parameter tuning. Section 6 concludes the paper.

## 2. Naïve Bayes with Semantic Smoothing

### 2.1 Definition of Topic Signatures

There is no strict definition for topic signatures. Any topic carrier can be viewed as a topic signature. In this paper, we consider three types of topic signatures. They are unigrams (single-word features), multiword phrases, and ontological concepts, respectively.

A concept is a unique meaning in a specific domain. It represents a set of synonymous terms in the domain. For example, *C0020538* is a concept about the disease of hypertension in UMLS Metathesaurus; it also represents a set of synonymous terms including *high blood pressure*, *hypertension*, and *hypertensive disease*. In the experiment, we use UMLS concepts as topic signatures for the corpus of OHSUMED.

A multiword phrase consists of two or more words adjacent to each other. It is kind of fixed expressions or collocations; people are frequently using it in writings or conversations. *Space Program*, *Third World Debt*, and *Machine Learning* are typical examples of multiword phrases. Multiword phrases can be viewed as n-grams with syntactic and statistical constraints. We use multiword phrases rather than n-grams because the former makes more sense when interpreting its semantics and the former can serve as a dictionary for reuse. The complexity of extracting multiword phrases is a concern. But we only extract multiword phrases from training documents for smoothing purpose; thus, the concern is not serious. In practice, one can use exact string matching to identify multiword phrases defined in a vocabulary. If there is no existing vocabulary for a corpus, tools such as Xtract [18] can be used to automatically compile a multiword phrase vocabulary.

In summary, we introduce ontological concepts and multiword phrases as topic signatures because both are less ambiguous than single-word topic signatures. Thus, they can generate more stable semantic mappings across collections or even domains. In Section 4, we compare the behaviors of three types of topic signatures.

### 2.2 Semantic Smoothing for NB Classifiers

The naïve bayesian classifier is widely used for text classification due to 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:

$$C(d) = \arg \max_{c_i} p(c_i)p(d | c_i) \quad (2.1)$$

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:

$$p(c_i) = \frac{1 + N(c_i, D)}{|C| + |D|} \quad (2.2)$$

where  $N(c_i, D)$  denotes the number of documents with class label  $c_i$  in collection  $D$ . The second term in equation (2.1) 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:

$$p(d | c_i) = \prod_{k=1}^{|d|} p(w_{d_i,k} | c_i) \quad (2.3)$$

Now the problem is reduced to estimating class model, i.e. 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 [14]. For this reason, all experiments in this paper are based on multinomial mixture mode.

The simplest implementation of a multinomial class model is the maximum likelihood estimate with Laplacian smoothing [9]. That is,

$$p_l(w | c_i) = \frac{1 + N(w, c_i)}{|V| + \sum_w N(w, c_i)} \quad (2.4)$$

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, Laplacian smoothing assign 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 proved to be effective for IR [25] [26]. We refer this line of smoothing methods as background smoothing in this paper. The Jelinek-Mercer [10] [25] 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  $\beta$  as shown in equation (2.5):

$$p_b(w | c_j) = (1 - \beta)p_{ml}(w | c_j) + \beta p(w | D) \quad (2.5)$$

where  $p_{ml}(w | c_j)$  is the unigram class model with maximum likelihood estimate and  $p_b(w | c_j)$  denotes the unigram class model with background smoothing. In this paper,  $\beta$  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 [20] [28], 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 (2.5) and the class model ends with the following formula:

$$p_s(w | c_i) = (1 - \lambda)p_b(w | c_i) + \lambda \sum_k p(w | t_k) p(t_k | c_i) \quad (2.6)$$

where  $p_b(w | c_i)$  stands for the unigram class model with semantic smoothing and  $t_k$  denotes the  $k$ -th topic signature and  $p(t_k | c_i)$  is the distribution of topic signatures in training documents of a given class, which can be computed via maximum likelihood estimates. The translation coefficient  $\lambda$  is to control the influence of the semantic mapping component in the mixture model. If the translation 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 regarding the optimization of the translation coefficient. The remaining problem is how to compute the probability of semantic mappings from topic signatures  $t_k$  to single-word feature  $w$ , which will be addressed in the following sub-section 2.3.

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 200 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.

### 2.3 Estimates of Semantic Mappings

For each topic signature  $t_k$ , we can obtain a set of documents ( $D_k$ ) containing the signature. 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. However, some words address topics corresponding to other topic signatures while some are background words of the whole collection. Therefore, we employ a mixture language model as described in equation (2.7) to remove noise, i.e.,

words are generated either by the topic signature mapping model or by the background collection model.

$$p(w|D_k) = (1-\alpha)p(w|t_k) + \alpha p(w|C) \quad (2.7)$$

When this mixture model is used for text generation, it is unknown regarding what model a word is exactly generated by. It is instead a hidden variable. But the chance of selection either model is known. Here  $\alpha$  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:

$$\log p(D_k) = \sum_w c(w, D_k) \log p(w|D_k) \quad (2.8)$$

Here  $c(w, D_k)$  is the document frequency of term  $w$  in  $D_k$ , i.e., 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 [4] with the following update formulas:

$$\hat{p}^{(n)}(w) = \frac{(1-\alpha)p^{(n)}(w|t_k)}{(1-\alpha)p^{(n)}(w|t_k) + \alpha p(w|C)} \quad (2.9)$$

$$p^{(n+1)}(w|t_k) = \frac{c(w, D_k)\hat{p}^{(n)}(w)}{\sum_i c(w_i, D_k)\hat{p}^{(n)}(w_i)} \quad (2.10)$$

As usual, the maximum likelihood estimator initializes the EM algorithm. With respect to the setting of the background coefficient  $\alpha$ , the larger  $\alpha$  is, the more specific the trained parameters are. When  $\alpha$  closes to one, the majority of terms get extremely small probability values. Our study shows a large  $\alpha$  (e.g. 0.9) fits for applications such as query expansion in which only a few most important terms are expanded and a medium  $\alpha$  (e.g. 0.5) is good for applications such as text classification and clustering. We also truncate terms with extremely small translation probabilities for two purposes. First, with smaller number of translation 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 translation probability less than 0.0005 and renormalize the mapping probabilities of the remaining terms.

Our estimation of semantic mappings is significantly different from the statistical translation model [5] 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 cooccurrence 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 ontology-based concepts. Consequently, the semantic

mapping is more specific. From three examples shown in Figure 1, we can see that phrase-word (“space program”) mappings are quite coherent and specific. However, if we estimate semantic mappings for its constituent terms “space” and “program” separately, both contain mixed topics and are fairly general. Some terms such as *NASA*, *astronaut*, *moon*, *satellite*, *rocket*, and *Mar*, which is highly correlated to the subject of space program, do appear in the result of phrase mappings, but in neither of word mappings.

<p><b>Space:</b>  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;</p> <p><b>Program:</b>  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;</p> <p><b>Space Program</b>  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;</p>
--

**Figure 1:** The demonstration of semantic mapping (only top 20 topical terms are listed). All three examples are trained on the 20-newsgroup corpus.

### 3. Datasets and Protocols

#### 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 on individual category first and then be averaged over categories, or globally computed over all categories. The former is called macro-F1 while the later is called micro-F1 [22]. 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 common 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 sub feature space can dramatically improve the performance for many classifiers including the NB classifier using Laplacian smoothing. In all experiments, we choose CHI feature selector [23] 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.

### 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 noise. LATimes contains news articles. OHSUMED consists of scientific abstracts collected from Medline, an on-line medical information database. We selected these three collections in that the sources of selected collections are of diversity.

**Table 1:** the basic statistics of three testing collections

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

20NG has twenty classes each of which contains about 1,000 articles. Total 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 Jan 1, 1989 to December 31, 1990. There are total 111,084 articles distributed in twenty-two sections, e.g. Financial, Entertainment, Sports, etc. We consider the section an article sits in the ground truth of memberships. The articles in 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 its length 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, Calendar. The OHSUMED

corpus contains 13,929 Medline abstract 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 rest 7,400 abstracts.

### 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 Meta data 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, using the algorithm proposed in section 2.3. The parameter  $\alpha$  is set to 0.5.

Multiword phrases are extracted from 20NG and LATimes by a modified version of Xtract [18]. 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 are, the higher quality but less number of phrases Xtract produce. In the experiment, we set those four parameters to (1, 1, 4, 0.75). The detail of the implementation is available in [27].

UMLS concepts are extracted from OHSUMED by MaxMatcher [29]. MaxMatcher is a dictionary-based biological concept extraction tool. In order to increase the extraction recall while remaining 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 [31]. The precision and recall reached 71.60% and 75.18%, respectively, using approximate match criterion.

## 4 Experiment Results

### 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 2. CSSS significantly outperform Laplacian 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 Laplacian smoothing and background smoothing for bayesian text classifiers when the number of training documents is small and the data are sparse.

**Table 2:** Comparisons of context-sensitive semantic smoothing (CSSS) to Laplacian smoothing (Lap) and background smoothing (Bkg). 1% of documents are used for training and the remaining 99% for testing. The parameter  $\lambda$  for CSSS is 0.4. The sign ( $\dagger$ ) means the semantic knowledge is learned from other corpus. The symbols \*\* and \* indicates 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	<b>0.413</b>	**17.3%	**10.9%
20NG	0.427	0.526	<b>0.613</b>	**43.7%	**16.6%
LATimes	0.525	0.538	<b>0.581</b>	**10.7%	**7.9%
LATimes $\dagger$	0.525	0.538	<b>0.559</b>	**6.5%	**3.8%

(b) The result of macro-F1					
Collection	Lap	Bkg	CSSS	vs. Lap	vs. Bkg
OHSUMED	0.205	0.280	<b>0.362</b>	**76.2%	**29.1%
20NG	0.421	0.523	<b>0.613</b>	**45.5%	**17.2%
LATimes	0.492	0.513	<b>0.562</b>	**14.3%	**9.5%
LATimes $\dagger$	0.492	0.513	<b>0.541</b>	**10.0%	**5.4%

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 Micro-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 other two collections are highly skewed and as we pointed out earlier, the result of Micro-F1 is dominated by the performance of some common 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 observation is that the magnitude of improvement of semantic smoothing over the other two approaches depends on the dataset. Take the example of

the improvement of Micro-F1 (Semantic smoothing vs. Laplacian smoothing). On the corpus of 20NG, CSSS achieved the biggest improvement of 43.7%, but only 10.7% and 17.3% on LATimes and OHSUMED. The 20NG corpus has a large vocabulary space of 133K words whereas the average number of unique words per doc in 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 (e.g. multiword phrases and ontology-based concepts) to single word features. This explains why semantic smoothing obtained as much as 43.7% on 20NG.

**Table 3:** Comparisons of CSSS to Lap and Bkg. 33% of documents are used for training and the remaining 67% for testing. The parameter  $\lambda$  for CSSS is tuned to the best.

(a) The result of micro-F1					
Collection	Lap	Bkg	CSSS	vs. Lap	vs. Bkg
OHSUMED	0.660	<b>0.667</b>	0.665	0.8%	-0.2%
20NG	0.771	0.802	<b>0.820</b>	*6.3%	*2.2%
LATimes	0.728	0.726	<b>0.729</b>	0.2%	*0.4%

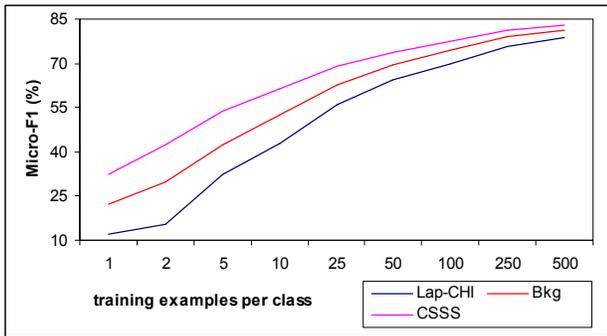
(b) The result of macro-F1					
Collection	Lap	Bkg	CSSS	vs. Lap	vs. Bkg
OHSUMED	0.626	0.639	<b>0.669</b>	*6.8%	4.7%
20NG	0.756	0.787	<b>0.820</b>	*8.5%	*4.1%
LATimes	0.708	0.696	<b>0.719</b>	*1.5%	*3.3%

According to the results of the first experiment, we can say that the more sparse the data, the more effective the semantic smoothing. To further validate this finding, 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 three testing collections. The results are shown in table 3.

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 the performance metric Micro-F1, semantic smoothing achieved significant improve over 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 all testing collections though the magnitude of improved was less than in the case of 1% training data. It is because some small classes still have serious data sparse problem even though one-third documents are selected for training. For example, the smallest four classes in OHSUMED and LATimes have about only 70 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.

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 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 easy to control the size of training data for each class. The results are shown in Figure 2. Clearly, we can see that the effectiveness of semantic smoothing is in inverse proportion to the size of 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 Laplacian smoothing and background smoothing, respectively. With the increase of training documents, data become less sparse and consequently, semantic smoothing goes less effective and ends with slight gains (5.3% and 1.8%) over two baseline smoothing methods.



**Figure 2:** The performance of bayesian text classifiers with semantic smoothing, Laplacian smoothing, and background smoothing on the corpus of 20NG with different number of training document.

## 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 context-insensitive semantic smoothing (CISS, i.e. using words as topic signatures). It is worth noting that CISS is different from the translation model [5] in that the former uses cooccurrence data whereas the latter uses document-query pairs, even though both of them use words as topic signatures. Zhou et al.[28] 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 4. In 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 orcl. These domain-specific terms are much less ambiguous than general conversational 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 which affects the effectiveness of semantic smoothing.

**Table 4:** The comparison of context-sensitive semantic smoothing (CSSS) to context-insensitive semantic smoothing (CISS)

(a) 1% of documents for training

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

(b) 33% of documents for training

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

**Table 5:** The comparison of CSSS to CISS on 20NG corpus with the number of training documents ranging from one to five hundred. The parameter  $\lambda$  is 0.4 for both CISS and CSSS.

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

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. Interestingly, CSSS performs worse than CISS when training data set is very small. With the increase of training documents, the extracted topic

signatures becomes more representative and approach to the true topics associated with those training documents. Eventually, CSSS exceeds CISS.

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. Besides, 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.

### 4.3 Reuse of Semantic Knowledge

One advantage of semantic smoothing over topic models including pLSI and LDA is its reusability. When one 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.

We learn semantic mapping knowledge from TDT2 (64,500 news articles) and then employ it to classify 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. The results are shown in table 2 and 4a (the row with sign †). The improvement of Micro-F1 using CSSS over Lap and Bkg reaches 6.5% and 3.8%, respectively; the improvement of Macro-F1 is 10.0% and 5.4%, respectively. Although the magnitude of improvement is less than the case of using semantic knowledge learned from the same corpus due to the low coverage, it is still statistically significant.

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.

### 4.4 Semantic Smoothing vs. SVM

Support vector machine (SVM) is a powerful learning approach for solving two-class pattern recognition problem [19]. 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 [22]. 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 6:** The comparisons of support vector machines (SVM) to NB 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	<b>0.413</b>	**17.5%	0.206	<b>0.351</b>	**70.7%
20NG	0.472	<b>0.613</b>	**29.9%	0.464	<b>0.609</b>	**31.1%
LATimes	0.524	<b>0.581</b>	**10.8%	0.491	<b>0.554</b>	**12.7%

(b) 33% of documents for training

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

SVM can not handle 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 [2]. Besides, several other factors such as the choice of kernel, scaling, and feature selection can also affect the performance of a 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 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 large number of training document (see table 6-b) are consistent with previous studies [22]. SVM always significantly outperform naïve bayes (see table 3). The bayesian text classifier with semantic smoothing basically has similar performance to naïve bayes and less effective than SVM. However, when the number of training documents becomes extremely smaller (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 6-b. It is mostly likely due to the fact that a large number of features are blind to SVM when training document set is very small and the power of SVM is compromised while a bayesian classifier can expand meaningful features through semantic smoothing.

#### 4.5 Semantic Smoothing vs. Active Learning

Semantic smoothing has proved to be effective in improving classification performance when the size of training dataset is small in our experiments. In literature, active learning also shows its effectiveness in dealing with small number of training samples. 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 [14]. For this reason, we compare semantic smoothing with an active learning classifier proposed by [14]. We choose this approach to compare 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 7. The active learning algorithm uses Laplacian smoothing and seems quite sensitive to the feature selection. The result of active learning reported in table 7 is actually the one with best tuning.

**Table 7:** 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	<b>0.413</b>	**12.1%	0.205	<b>0.351</b>	**71.1%
20NG	0.575	<b>0.613</b>	**6.6%	0.551	<b>0.609</b>	**10.4%
LATimes	0.566	<b>0.581</b>	*2.6%	0.536	<b>0.554</b>	3.3%

Active learning does improve the performance over the baseline naïve bayesian classifier on most collections (see table 2 and 7). However, it is much less effective than semantic smoothing approach. Besides, complexity and robustness are two other concerns regarding active learning. Active learning has an iterative learning process and thus runs very slow compared to semantic smoothing and naïve bayes. The performance of active learning depends on the added unlabeled data. Moreover, it is very sensitive to feature selection. For example, without feature selection, the Micro-F1 of 20NG can be as bad as 0.005; with appropriate feature selection, the result can increase to 0.575. However, active learning looks simpler

than semantic smoothing. It does not have to prepare topic signatures and semantic knowledge in advance.

#### 5. Tuning of Translation Coefficient

Topic signatures are very effective in mapping to single-word features as demonstrated in Figure 1. But the number of extracted topic signatures is often much less than the original single-word features. Therefore, if one only use topic signature based mapping, there 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 2.6 in section 2.1) as many other researchers did [20] [25] [26] [27] [28]. Then, the optimization of mixture weights (i.e. translation coefficient) becomes a problem.

The interpolation-based mixture model is originally designed to smooth multi-gram language models [6]. The mixture weight lambda can be globally optimized using the EM algorithm [4] 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 8. The automatic prediction of the optimal translation coefficient is close to the manually tuned 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.

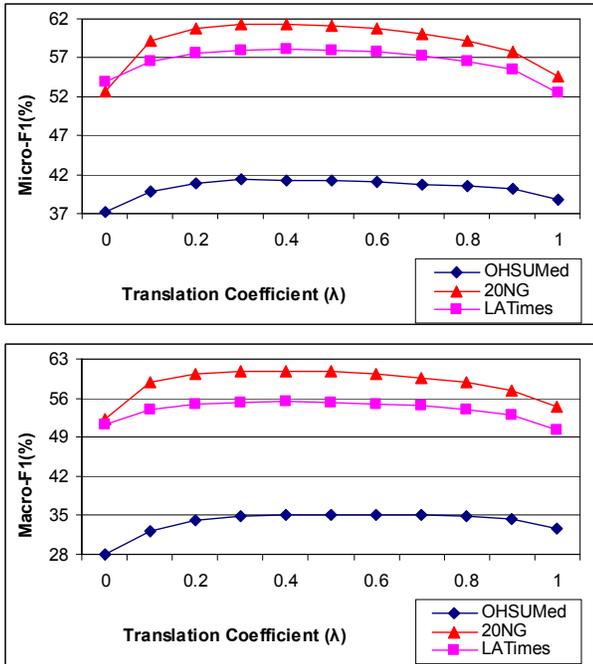
**Table 8:** The comparison of manually parameter tuning to automatic parameter tuning. The parameter  $\lambda$  is the translation coefficient.

Collection	Small training (1%)				Large training (33%)			
	manual		automatic		manual		automatic	
	$\lambda$	mi-F1	$\lambda$	mi-F1	$\lambda$	mi-F1	$\lambda$	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 in [26]. But this method is not very effective in the setting of text retrieval. They still recommend careful tuning of the mixture weight [25] [26]. Most of recent work on mixture language modeling IR also manually tune the mixture weights [20][27][28].

Fortunately, previous studies have shown that similar collections have similar optimal translation coefficients, making held-out training possible. The previous work [28]

showed that 0.3 is a good empirical setting for the translation coefficient in the setting of text retrieval. In this experiment, all of three collections achieved the best result when the translation coefficient was set to 0.4 in the case of 1% training data. When the training data increase to 33%, the data become less sparse and the optimal translation coefficient reduces to 0.1 except the 20NG corpus. As discussed earlier, 20NG corpus is quite sparse. Even if 33% documents are used for training, the features look still sparse and the optimal results are achieved when the translation coefficient set to 0.4.



**Figure 3:** The variance of the classification performance (i.e. micro-F1 and macro-F1) on three testing collections with the change of the translation coefficient which controls the influence of the translation component in the mixture model. The bayesian text classifiers use 1% of documents for training.

We can also see the robustness of semantic smoothing for bayesian text classification from Figure 3. In a wide neighbor of the optimal translation coefficient, semantic smoothing outperforms background smoothing. On 20NG and OHSUMED, semantic smoothing always beats background smoothing regardless the setting of the translation coefficient. On LATimes, semantic smoothing wins positive gain over background smoothing except the setting point of one.

In short, one can take the following empirical rules with respect to the choice of the translation coefficient: if data are very sparse, set the translation coefficient to 0.3~0.5; decrease the value when data become less sparse;

when sufficient training data are provided, stop using semantic smoothing.

## 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 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, ontology-based concepts to single-word features is viewed as CSSS while 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 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 future. Our experiments verified this hypothesis. With 60% vocabulary coverage, the semantic knowledge learned from other corpus can still significant improves the accuracy of text classification over Laplacian 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 Laplacian 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, Laplacian smoothing, and background smoothing is getting down. 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 less 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 translation coefficient to control the influence of the two components. The optimization of the translation coefficient is still an open problem. We proposed an automatic parameter tuning method which 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 the optimal value, but sometimes is quite far. This is also the problem of IR community when mixture language models are used for retrieval. Fortunately, first of all, 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 translation coefficient. if data are very sparse, set the translation coefficient to 0.3~0.5; decrease the value when data become less sparse; when sufficient training data are provided, stop using semantic smoothing.

For future work, we will continue working on the optimization of the translation coefficient. Another direction will be focused on the reuse of semantic knowledge. In this paper, the experiment showed that it was quite promising to reuse the semantic knowledge. In future, we will be more interested in factors which affect the effectiveness of semantic knowledge reuse for various applications such as text classification and text retrieval.

## 7. Acknowledgement

This work is supported in part by NSF Career Grant IIS 0448023, NSF CCF 0514679, PA Dept of Health Tobacco Settlement Formula Grant (No. 240205 and No. 240196), and PA Dept of Health Grant (No. 239667).

## References

[1] 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

- [2] 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.
- [3] Bai, J., Nie, J.-Y., and Cao, G., "Integrating Compound Terms in Bayesian Text Classification," *Web Intelligence 2005*, France.
- [4] Baker, L.D. and McCallum, A.K., "Distributional Clustering of Words for Text Classification," *Proc. Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval*, 1998.
- [5] 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.
- [6] Blei, D., Ng, A. and Jordan, M., "Latent Dirichlet allocation," *Journal of machine Learning Research*, 3, 2003, pp 993-1022.
- [7] 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.
- [8] 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.
- [9] Hoffman, T., "Probabilistic latent semantic indexing," *1999 ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99)*, 1999, pp. 50-57
- [10] 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.
- [11] 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.
- [12] 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.
- [13] 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.
- [14] 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
- [15] Peng, F., Schuurmans, D. and Wang, S., "Augmenting naive bayes classifiers with statistical language models," *Information Retrieval*, 7(3-4):317-345, 2004.
- [16] Quinlan, J.R., "Induction of decision trees," *Machine Learning*, 1(1): 81-106, 1986
- [17] 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.
- [18] Smadja, F. "Retrieving collocations from text: Xtract," *Computational Linguistics*, 1993, 19(1), pp. 143--177.
- [19] Vapnik, V.N., *the Nature of Statistical Learning Theory*, Springer, 1995.
- [20] 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
- [21] 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*
- [22] 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.
- [23] 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.
- [24] 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.
- [25] 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.
- [26] 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)*.
- [27] Zhang, X., Zhou, X., and Hu, X., "Semantic Smoothing for Model-based Document Clustering," in *2006 IEEE International Conference on Data Mining (IEEE ICDM06)*, Dec. 18-22, 2006, Hong Kong, pp. 1193-1198,
- [28] Zhou, X., Hu, X., Zhang, X., Lin, X., and Song, I.-Y., "Context-sensitive Semantic Smoothing for Language Modeling Approach to Genomic Information Retrieval," In *the 29th Annual International ACM SIGIR Conference (ACM SIGIR 2006)*, Aug 6-11, 2006, Seattle, WA, USA, pp. 170-177.
- [29] 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, Guangxi, China, Page 1145-1149
- [30] 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)*, October 29-31, 2007, Patras, Greece
- [31] GENIA, <http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA>