

# Automatic Word Clustering for Text Categorization Using Global Information

Chen Wenliang, Chang Xingzhi, Wang Huizhen, Zhu Jingbo, and Yao Tianshun

Natural Language Processing Lab  
Northeastern University, Shenyang, China, 110004

{chenwl, zhujingbo, tsyao}@mail.neu.edu.cn

{changxz, wanghz}@ics.neu.edu.cn

## ABSTRACT

This paper presents a cluster-based text categorization system which uses class distributional clustering of words. We propose a new clustering model which considers the global information over all the clusters. The model can group words into clusters based on the distribution of class labels associated with each word. Using these learned clusters as features, we develop a cluster-based classifier. We present several experimental results to show that our proposed method performs better than the other three text classifiers. The proposed model has better results than the model which only considers the information of the two related clusters. Specially, it can maintain good performance when the number of features is small and the size of training corpus is small.

## Categories and Subject Descriptors

I.5.3 [PATTERN RECOGNITION]: Clustering—*Similarity measures*; I.5.4 [PATTERN RECOGNITION]: Application—*Text processing*; I.5.m [PATTERN RECOGNITION]: Miscellaneous

## General Terms

Algorithms, Management, Experimentation

## Keywords

Text Categorization, Word Clustering, Clustering Algorithm

## 1. INTRODUCTION

The goal of text categorization is to classify documents into a certain number of predefined categories. A variety of techniques for supervised learning algorithms have demonstrated reasonable performance for text categorization[5][11][12]. A common and overwhelming characteristic

of text data is its extremely high dimensionality. Typically the document vectors are formed using bag-of-words model. It is well known, however, that such count matrices tend to be highly sparse and noisy, especially when the training data is relatively small. So when the text categorization systems are applied, there are two problems to be counted:

- High-dimensional feature space: Documents are usually represented in a high-dimensional sparse feature space, which is far from optimal for classification algorithms.
- Short of training documents: Many applications can't provide so many training documents.

A standard procedure to reduce feature dimensionality is feature selection, such as Document Frequency,  $\chi^2$  statistic, Information Gain, Term Strength, and Mutual Information[13]. But feature selection is better at removing detrimental, noisy features. The second procedure is cluster-based text categorization[1][2][3][10]. Word clustering methods can reduce feature spaces by joining similar words into clusters. First they grouped words into the clusters according to their distributions. Then they used these clusters as features for text categorization.

In this paper, we cluster the words according to their class distributions. Based on class distributions of words, Baker[1] proposes a clustering model. In clustering processing, we will select two most similar clusters by comparing the similarities directly. But Baker's model only considers two related clusters, when computing the similarity between the clusters without taking into account the information of other clusters. In order to provide better performance, we should take into account the information of all the clusters when computing the similarities between the clusters. This paper proposes a clustering model which considers the global information over all the clusters. The model can be understood as the balance of all the clusters according to the number of words in them.

Using these learned clusters as features, we develop a cluster-based Classifier. We present experimental results on a Chinese text corpus. We compare our text classifier with the other three classifiers. The results show that the proposed clustering model provides better performance than Baker's model. The results also show that it can perform better than the feature selection based classifiers. It can maintain high performance when the number of features is small and the size of training corpus is small.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AIRS '04 Beijing, China

Copyright 2004 ACM 1-58113-000-0/00/0004 ...\$5.00.

In the rest of this paper: Section 2 reviews previous works. Section 3 proposes a global Clustering Model (globalCM). Section 4 describes a globalCM-based text categorization system. Section 5 shows the experimental results. Finally, we draw our conclusions at section 6.

## 2. RELATED WORK

Distributional Clustering has been used to address the problem of sparse data in building statistical language models for natural language processing[7][10]. There are many works[1][2] related with using distributional clustering for text categorization.

Baker and McCallum[1] proposed an approach for text categorization based on word-clusters. First, find word-clusters that preserve the information about the categories as much as possible. Then use these learned clusters to represent the documents in a new feature space. Final, use a supervised classification algorithm to predict the categories of new documents. Specifically, it was shown there that word-clustering can be used to significantly reduce the feature dimensionality with only a small change in classification performance.

## 3. GLOBAL CLUSTERING MODEL BASED ON CLASS DISTRIBUTIONS OF WORDS

In this section, we simply introduce the class distribution of words[1]. Then we propose the Global Clustering Model, here we name it as globalCM. In our clustering model, we define a similarity measure between the clusters, and add the candidate word into the most similar cluster that no longer distinguishes among the words different.

### 3.1 Class Distribution of Words

Firstly, we define the distribution  $P(C|w_t)$  as the random variable over classes  $C$ , and its distribution given a particular word  $w_t$ . When we have two words  $w_t$  and  $w_s$ , they will be put into the same cluster  $f$ . The distribution of the cluster  $f$  is defined

$$\begin{aligned} P(C|f) &= P(C|w_t \vee w_s) \\ &= \frac{P(w_t)}{P(w_t) + P(w_s)} \times P(C|w_t) \\ &+ \frac{P(w_s)}{P(w_t) + P(w_s)} \times P(C|w_s) \end{aligned} \quad (1)$$

Now we consider the case that a word  $w_t$  and a cluster  $f$  will be put into a new cluster  $f_{new}$ . The distribution of  $f_{new}$  is defined

$$\begin{aligned} P(C|f_{new}) &= P(C|w_t \vee f) \\ &= \frac{P(w_t)}{P(w_t) + P(f)} \times P(C|w_t) \\ &+ \frac{P(f)}{P(w_t) + P(f)} \times P(C|f) \end{aligned} \quad (2)$$

### 3.2 Similarity Measures

Secondly, we turn to the question of how to measure the difference between two probability distributions. Kullback-Leibler divergence is used to do this. The KL divergence between the class distributions induced by  $w_t$  and  $w_s$  is written

$D(P(C|w_t)||P(C|w_s))$ , and is defined

$$- \sum_{j=1}^{|C|} P(c_j|w_t) \log \frac{P(c_j|w_t)}{P(c_j|w_s)} \quad (3)$$

But KL divergence has some odd properties: It is not symmetric, and it is infinite when  $p(w_s)$  is zero. In order to resolve these problems, Baker[1] proposes a measure named "KL divergence to the mean" to measure the similarity of two distributions(Here we name it as  $S_{mean}$ ). It is defined

$$\begin{aligned} &\frac{P(w_t)}{P(w_t) + P(w_s)} \times D(P(C|w_t)||P(C|w_s \vee w_t)) \\ &+ \frac{P(w_s)}{P(w_t) + P(w_s)} \times D(P(C|w_s)||P(C|w_s \vee w_t)) \end{aligned} \quad (4)$$

$S_{mean}$  uses a weighted average and resolves the problems of KL divergence. But it only considers the two related clusters without thinking about other clusters. Our experimental results show that the numbers of words in learned clusters, which are generated by Baker's clustering model, are very different. Several clusters include so many words while most clusters include only one or two words.

We study the reasons of these results. When Equation 4 is applied in the clustering algorithm, it can't work well if the numbers of words in the clusters are very different at iterations.

For example, we have a cluster  $f$  which include only a word(In Baker's clustering model, a new candidate word will be put into an empty cluster). We will compute the similarities between  $f$  and the other two clusters( $f_i$  and  $f_j$ ) using Equation 4. Let  $f_i$  has many words(ie. 1000 words) and  $f_j$  has one or two words. We define:

$$\begin{aligned} S_i &= \frac{P(f)}{P(f) + P(f_i)} \times D(P(C|f)||P(C|f \vee f_i)) \\ &+ \frac{P(f_i)}{P(f) + P(f_i)} \times D(P(C|f_i)||P(C|f \vee f_i)) \\ &= (1 - \alpha_i) \times D_{i1} + \alpha_i \times D_{i2} \end{aligned} \quad (5)$$

$$\begin{aligned} S_j &= \frac{P(f)}{P(f) + P(f_j)} \times D(P(C|f)||P(C|f \vee f_j)) \\ &+ \frac{P(f_j)}{P(f) + P(f_j)} \times D(P(C|f_j)||P(C|f \vee f_j)) \\ &= (1 - \alpha_j) \times D_{j1} + \alpha_j \times D_{j2} \end{aligned} \quad (6)$$

According to Equation 2, if a word is added to a cluster, the word will affect tiny to the cluster which includes many words and affect remarkable to the cluster which includes few words. So the distribution of  $f \vee f_i$  is very similar to  $f_i$  because  $f_i$  has many words and  $f$  has only one word. And then  $D_{i2}$  is near zero.  $\alpha_i$  is near 1 and  $(1 - \alpha_i)$  is near zero because the number of  $f_i$  is very large than  $f$ . We know:

$$S_i \approx D_{i2} \approx 0 \quad (7)$$

So when we compute the similarities between  $f$  and the other clusters using Equation 4,  $f$  will be more similar to the cluster which includes more words.

The problems of  $S_{mean}$  indicate that we should consider the information of all the clusters when computing the similarity between the two clusters. If we only take into account

---

Input:  
W - the vocabulary includes the candidate words  
M - desired number of clusters  
Output:  
F - the learned clusters

Clustering:

1. Sort the vocabulary by  $\chi^2$  statistic with the class variable.
  2. Initialize the M clusters as singletons with the top M words.
  3. Loop until all words have been put into one of the M clusters.
    - (a) Compute the similarities between the M clusters(Equation 8).
    - (b) Merge the two clusters which are most similar, resulting in M-1 clusters.
    - (c) Get the next word from the sorted list.
    - (d) Create a new cluster consisting of the new word.
- 

**Table 1: The globalCM Algorithm**

the two related clusters, the system will can't work well. In order to resolve the problems, we propose a new similarity measure that considers the global information over the clusters. The similarity between a cluster  $f_i$  and a cluster  $f_j$  is defined

$$S_{global} = \frac{N(f_i) + N(f_j)}{2 \sum_{k=1}^{|M|} N(f_k)} \times S_{mean} \quad (8)$$

Where  $N(f_k)$  denotes the number of words in the cluster  $f_k$ , M is the list of clusters. Equation 8 can be understood as the balance of all the clusters according to the numbers of words in them. In our experimental results show that it can work well even if the numbers of words in the clusters are very different.

### 3.3 Global Clustering Model(globalCM)

Now we introduce a clustering model which use Equation 8. The model is similar to Baker's clustering model[1]. In this paper, we name Baker's model as BakerCM, and our model as globalCM.

In the algorithm, we set M is the final number of clusters. First, we sort the vocabulary by  $\chi^2$  statistic with the class variable. Then the clusters are initialized with the Top M words from the sorted list. Then we will group the rest words into the clusters. We compute the similarities between all the clusters(Equation 8) and then merge the two clusters which are most similar. Now we have M-1 clusters. An empty cluster is created and the next word is added. So the number of clusters is back to M. Table 1 shows the clustering algorithm.

## 4. THE GLOBALCM-BASED TEXT CATEGORIZATION SYSTEM

This section introduces our globalCM-based Chinese text categorization System. The system includes Preprocessing, Extracting the candidate words, Word Clustering, Cluster-based Text Classifier. Word Clustering has been described at Section 3.

### 4.1 Preprocessing

First, the html tags and special characters in the collected documents are removed. Then we should use a word segmentation tool to segment the documents because there are no word boundaries in Chinese documents.

### 4.2 Extracting the Candidate Words

We extract candidate words from the documents: First we use a stoplist to eliminate no-informative words, and then we remove the words whose frequencies are less than  $F_{min}$ . Final, we generate the class distributions of words which is described at Section 3.

### 4.3 The Cluster-based Classifier

Using the learned clusters as features, we develop a cluster-based text classifier. The document vectors are formed using bag-of-clusters model. If the words are included in the same cluster, they will be presented as the single cluster symbol. After representation, we develop a classifier based on these features.

In this paper, we use naïve Bayes for classifying documents. We only describe naïve Bayes briefly since full details have been presented in the paper [9]. The basic idea in naïve Bayes approach is to use the joint probabilities of features and categories to estimate the probabilities of categories when a document is given. Given a document d for classification, we compute the probabilities of each category c as follows:

$$P(c_j|d_i; \hat{\theta}) = \frac{P(c_j|\hat{\theta})P(d_i|c_j; \hat{\theta}_j)}{P(d_i|\hat{\theta})} \approx P(d_i|c_j; \hat{\theta}) \quad (9)$$

$$P(d_i|c_j; \hat{\theta}) = P(|d_i|)|d_i|! \prod_{t=1}^{|F|} \frac{P(f_t|c_j; \theta)^{N_{it}}}{N_{it}!} \quad (10)$$

Where  $P(c_j)$  is the class prior probabilities,  $|d_i|$  is length of document  $d_i$ ,  $N_{it}$  is the frequency of the feature  $f_t$  (Notes that the features are the cluster symbols in this paper.) in document  $d_i$ , F is the vocabulary and  $|F|$  is the size of F,

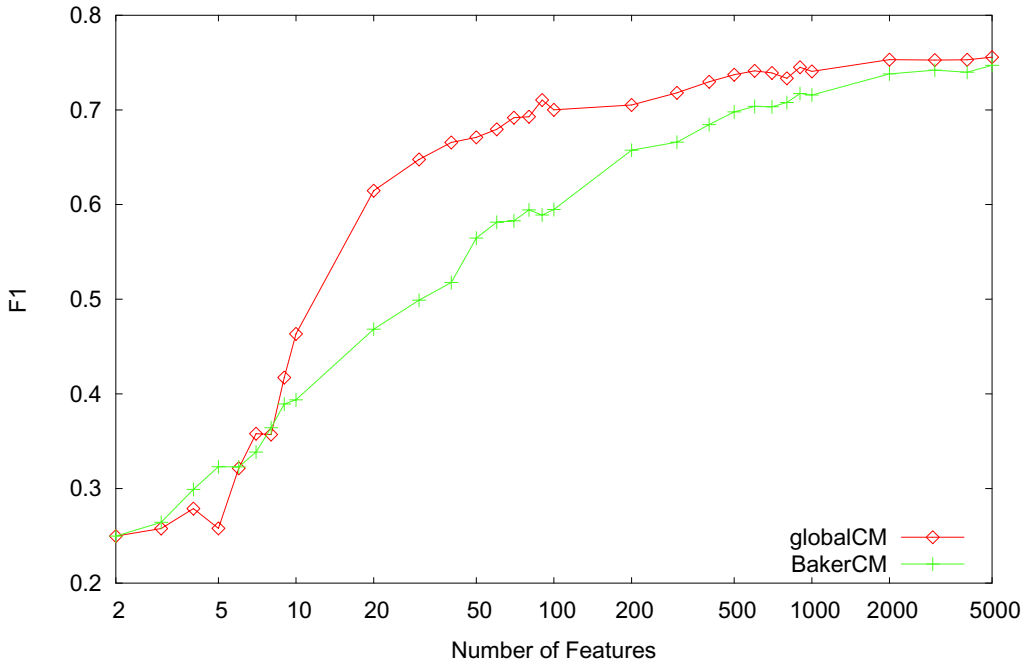


Figure 1: globalCM Vs BakerCM

$f_t$  is the  $t^{th}$  feature in the vocabulary, and  $P(f_t|c_j)$  thus represents the probability that a randomly drawn feature from a randomly drawn document in category  $c_j$  will be the feature  $f_t$ . The probability is estimated by the following formulae:

$$P(f_t|c_j; \theta) = \frac{1 + \sum_{i=1}^{|D|} N_{it}P(c_j|d_i)}{|F| + \sum_{s=1}^{|F|} \sum_{i=1}^{|D|} N_{is}P(c_j|d_i)} \quad (11)$$

## 5. EVALUATION

In this section, we provide empirical evidence to prove that the globalCM-based text categorization system is a high-accuracy system.

### 5.1 Performance measures

In this paper, a document is assigned to only one category. We use the conventional recall, precision and F1 to measure the performance of the system. For evaluating performance average across categories, we use the micro-averaging method. F1 measures is defined by the following formula[6]:

$$F1 = \frac{2rp}{r+p} \quad (12)$$

Where r represents recall and p represents precision. It balances recall and precision in a way that gives them equal weight.

### 5.2 Experimental Setting

The NEU\_TC data set contains Chinese web pages collected from web sites. The pages are divided into 37 categories according to "China Library Categorization"[4]<sup>1</sup>. It

<sup>1</sup>China Library Categorization includes 38 categories. We use 37 categories of all, except category Z(综合性图书/Comprehensive Books)

consists of 14,459 documents. We do not use tag information of pages. We use the toolkit CipSegSDK[14] for word segmentation. We removed all words that have less than two occurrences( $F_{min} = 2$ ). The resulting vocabulary has 75480 words.

In experiments, we use 5-fold cross validation where we randomly and uniformly split each category into 5 folds and we take four folds for training and one fold for testing. In the cross-validated experiments we report on the average performance.

### 5.3 Experimental Results

We compare our globalCM-based classifier with the other three clustering and feature selection algorithms: BakerCM-based classifier,  $\chi^2$  statistic based classifier, and document frequency based classifier. These two feature selection methods are the best of feature selection methods according to Yang's experiments[13].

#### 5.3.1 Experiment 1: globalCM VS BakerCM.

In this experiment, we provide empirical evidence to prove that the globalCM based text classifier provides better performance than that based on BakerCM. Figure 1 shows the experimental results.

From Figure 1 we can find that globalCM provides better performance than BakerCM in most different features size cases. With 100 features, globalCM provides 10.6% higher than BakerCM. Only when the number of features is less than 7, BakerCM can provide the similar performance to globalCM.

#### 5.3.2 Experiment 2: globalCM-based classifier VS Feature-Selection-based classifiers.

In this experiment, we use three different size of training corpus: 10%, 50%, 100% of the total training corpus(Here we name them as T10, T50 and T100). And we select two fea-

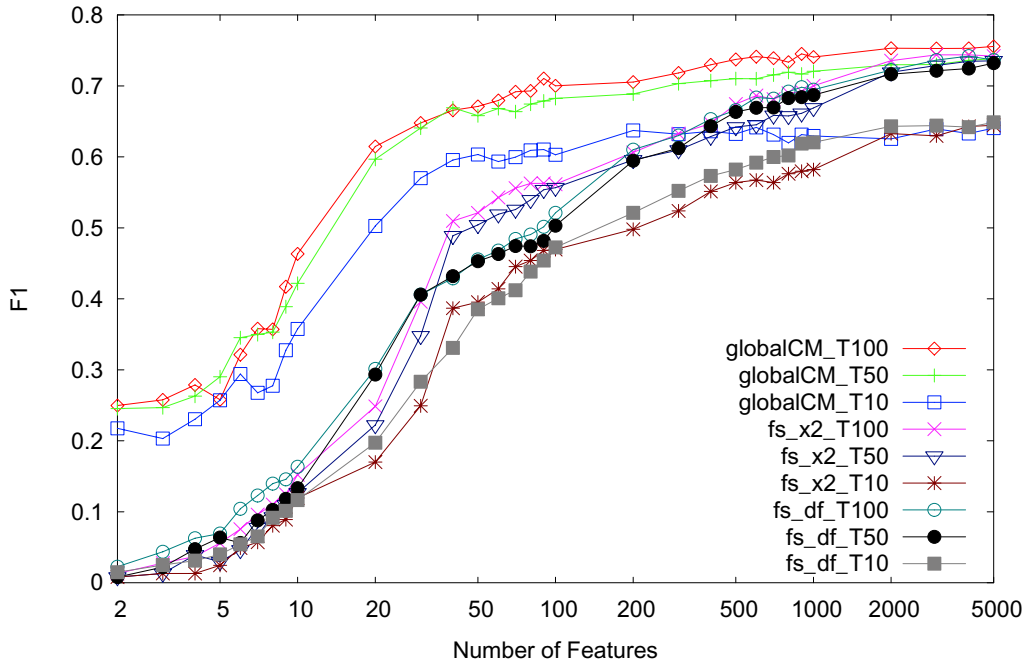


Figure 2: globalCM-based classifier vs Feature-Selection-based classifiers

ture selection methods: document frequency and  $\chi^2$  statistic for text categorization.

Figure 2 shows the effect of varying the amount of features with 3 different amounts of training dataset, where globalCM denotes our clustering model, fs\_x2 denotes  $\chi^2$  statistic feature selection method and fs\_df denotes document frequency feature selection method. For 3 different quantities of documents for training, we keep the number of features constant, and vary the number of documents in the horizontal axis.

Naturally, the more documents for training are used, the better the performance is. The best result of globalCM with T100 training corpus is 75.57%, 1.73% higher than the best result with T50 and 11.42% higher than the best result with T10. The best result of fs\_x2 with T100 training corpus is 74.37%, higher than the result of the other two training corpus.

Then, we study the results of the T100 training corpus. Notice that with only 100 features globalCM achieves 70.01%, only 5.6% lower than with 5000 features. In comparison, fs\_x2 provides only 56.15% and fs\_df provides only 52.12%. When with 1000 features, fs\_x2 can yields the similar result as globalCM with 100 features. Even with only 50 features, globalCM provides 67.10%. The best of globalCM is 1.20% higher than the best of fs\_x2 and 1.37% higher than fs\_df. The performance indicates that globalCM is providing more accuracy. And it can maintain near 70% with only 100 or less features while feature selection based classifiers have fallen into the 50s.

When we study the results of the other two training corpus, we can find that globalCM can maintain good performance with small training corpus. With T10 training corpus and 50 features, globalCM achieves 60.33%. It is near 20% higher than fs\_x2 and fs\_df. To our surprise, it is similar to the results of the feature selection based classifier using

T100 training corpus and 200 features.

Then we study the reasons why our cluster-based text classifier performs better than the feature selection based classifier. In feature selection method, the system discards some words that are infrequent. But in our clustering algorithm merges them into the clusters instead of discards them. So it can preserves information during merging.

## 6. CONCLUSIONS AND FUTURE WORK

In this paper, we present a cluster-based text categorization system which uses globalCM. While considering the global information over all the clusters, globalCM can group the words into the clusters more effectively. So it can yield better performance than the model which doesn't think about global information.

We conduct the experiments on a Chinese text corpus. The experimental results indicate that our globalCM-based text categorization system can provide better performance than feature selection based systems. And it can maintain high performance when the size of training corpus is small and the number of features is small.

Future work includes collecting the phrases as candidate features for learning algorithm because words forming phrases are a more precise description of content than words as a sequence of keywords[8]. For example, 'horse' and 'race' may be related, but 'horse race' and 'race horse' carry more circumscribed meaning than the words in isolation. We also plan to look at techniques for learning words from unlabeled documents to overcome the need for labeled documents.

## 7. ACKNOWLEDGEMENTS

This research was supported in part by the National Natural Science Foundation of China & Microsoft Asia Research (No. 60203019) and the Key Project of Chinese Ministry of

## 8. REFERENCES

- [1] L. D. Baker and A. K. McCallum. Distributional clustering of words for text classification. In W. B. Croft, A. Moffat, C. J. van Rijsbergen, R. Wilkinson, and J. Zobel, editors, *Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval*, pages 96–103, Melbourne, AU, 1998. ACM Press, New York, US.
- [2] R. Bekkerman, R. El-Yaniv, N. Tishby, and Y. Winter. On feature distributional clustering for text categorization. In W. B. Croft, D. J. Harper, D. H. Kraft, and J. Zobel, editors, *Proceedings of SIGIR-01, 24th ACM International Conference on Research and Development in Information Retrieval*, pages 146–153, New Orleans, US, 2001. ACM Press, New York, US.
- [3] R. Bekkerman, R. El-Yaniv, N. Tishby, and Y. Winter. Distributional word clusters vs. words for text categorization. *Journal of Machine Learning Research*, (3):1183–1208, 2003.
- [4] C. L. C. E. Board. *China Library Categorization(The 4th ed.)*. Beijing Library Press, Beijing, 1999.
- [5] T. Joachims. Text categorization with support vector machines: learning with many relevant features. In C. Nédellec and C. Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pages 137–142, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE. Published in the “Lecture Notes in Computer Science” series, number 1398.
- [6] Y. Ko and J. Seo. Automatic text categorization by unsupervised learning. In *Proceedings of COLING-00, the 18th International Conference on Computational Linguistics*, Saarbrücken, DE, 2000.
- [7] L. Lee. *Similarity-Based Approaches to Natural Language Processing*. PhD thesis, Harvard University, Cambridge, MA, 1997.
- [8] S. Lee and M. Shishibori. Passage segmentation based on topic matter. *Computer Processing of Oriental Languages*, 15(3):305–340, 2002.
- [9] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization*, 1998.
- [10] F. C. N. Pereira, N. Tishby, and L. Lee. Distributional clustering of english words. In *Meeting of the Association for Computational Linguistics*, pages 183–190, 1993.
- [11] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.
- [12] Y. Yang and X. Liu. A re-examination of text categorization methods. In M. A. Hearst, F. Gey, and R. Tong, editors, *Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval*, pages 42–49, Berkeley, US, 1999. ACM Press, New York, US.
- [13] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In D. H. Fisher, editor, *Proceedings of ICML-97, 14th International Conference on Machine Learning*, pages 412–420, Nashville, US, 1997. Morgan Kaufmann Publishers, San Francisco, US.
- [14] T. Yao, J. Zhu, L. Zhang, and Y. Yang. *Natural Language Processing - A research of making computers understand human languages*. Tsinghua University Press, Beijing, 2002.