Combining Multiple Classifiers for Automatic Text Categorization

Jyh-Jong Tsay
Dept. of Comp. Sci.
Natl. Chung Cheng Univ.
tsay@cs.ccu.edu.tw

Yuan-Gu Wei
Dept. of Comp. Sci.
Natl. Chung Cheng Univ.
wyg88@cs.ccu.edu.tw

Jing-Doo Wang
Dept. of Inform.
Taichung Healthcare and Management Univ.
jdwang@thmu.edu.tw

Abstract

In this paper we study the development of multiple classifier systems in the Chinese text categorization. Our objective is to develop efficient techniques to combine the strength of well-known classifiers such as linear classifiers, decision trees, Bayesian methods, neural networks, and support vector machines. We have experimented with Chinese documents from the Central News Agency and from the Web Openfind. Experiments show that our approaches significantly improve the classification accuracy of individual classifiers for Chinese text categorization as well as for web page classification.

Keywords: Classifier Combination, Multiple Classifier, Text Categorization.

1 Introduction

In recent years we have seen a tremendous growth of online text documents on the Internet, digital libraries and news sources. Effective location of information on these huge resources is difficult without good indexing as well as organization of text collections. Automatic text categorization, which is defined as the task of assigning predefined class (category) labels to text documents, is one of the main techniques that are useful both in organizing and in locating information in huge text collections from, for example, the Internet. Many approaches such as linear classifiers [14], decision trees [19], Bayesian methods [18], neural networks [18] and support vector machines [7, 24], have been extensively studied and used to implement classifier systems for text categorization as well as for web page classification. Although a lot of efforts have been spent on each these methods, we are reaching the limit of further performance improvement.

Multiple classifier systems which aim to combine the strength of individual classifiers to improve overall performance, have been widely studied recently [9, 11, 26]. Multiple classifier systems have been successfully applied to various applications such as handwritten numerals recognition [11]. We believe multiple classifier systems can be a viable alternative in text categorization especially for situations in which the performance of individual classifiers is poor, for example, categorizing web pages in web portals.

There are two different strategies, coverage optimization and decision optimization [10], to designing a multiple classifier system. In coverage optimization, a large committee of weak classifiers (learners) is generated, each classifier trained on a different bootstrap sample of the training data. Higher probabilities are given to instances which are most often misclassified in the subsequent samples so that a specialized classifier that puts more focus on misclassified instances is generated. After training process, the committee's weighted vote determines the class label of a new instance. Boosting [8] and Bagging [4] are two of the successful committee training methods. There is a empirical evaluation of Bagging and Boosting in [15].

In decision optimization, the committee usually consists of a small number of already trained classifiers. The operation of the committee can be either combination-based or selection-based. In combination-based approaches, a combining classifier is used to determine the class label of a new instance from outputs from committee members. In selection-based approaches, a selecting classifier is trained to select a classifier for each new instance from the committee, which can best classify the new instance.

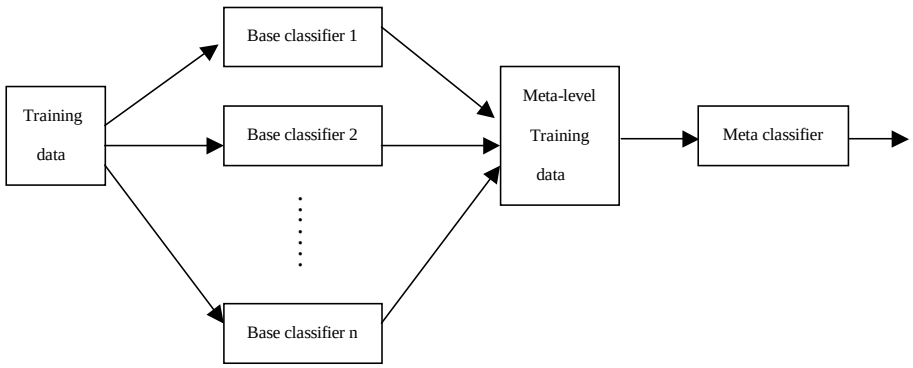

Figure 1: A general model of meta-learner

In this paper, we focus on developing efficient combination-based multiple classifier systems, and develop efficient techniques to combine the strength of well-known classifiers such as linear classifiers, decision trees, Bayesian methods, neural networks, and support vector machines which have been extensively studied. We compare five simple aggregation methods and two meta-learning methods. An aggregation method is like the voting method which aggregates the outputs of the base classifiers to decide which class should be assigned to a new instance. Each of the base classifier is trained with a set of raw training data. However, as shown in Fig 1, the meta classifier is trained with mete-level training data which are the outputs of the base classifiers. The meta classifier does not aim at picking the best base classifier; instead it tries to combine the predictions of the classifiers by learning their biases because the combination is useful only if there is disagreement [15]. The training set(meta-level training data or meta data) for the meta classifier varies according to the levels of information available from the various classifiers [28]. In this paper, we use the unique class label predicted by base classifier because it is the basic information which every classifier should provide.

We have two Chinese data sources, one from the Central News Agency(CNA) [1] and another from the Web Openfind [2] for experiments in this paper . There are 12 classes from CNA, and the training data and the testing data consist of 55257 documents and 5007 documents, respectively. From the web Openfind, we use the web pages at the leaf nodes of that web whose directory is a hierachical structure. There are 80 classes from the Openfind, and we use 10097 and 4411 web pages for the training data and the testing data, respectively. Experiments show that our approaches significantly improve the classification accuracy of individual classifiers for texts classification (CNA) as well as web page classification (Openfind).

The remainder of this paper is organized as follows. Section reviews related base classifiers . Section describes our approaches to classifier combination. Section gives experimental results. Section gives conclusions and further researches.

2 Related Base Classifiers

In this paper we have seven individual classifiers as base classifiers. They are Rocchio, Widrow-Hoff, K-Nearest Neighbor, Naive Bayes, decision trees, Neural Network, and Support Vector Machines. We briefly describe these base classifiers for completeness as follows.

2.1 Rocchio

Rocchio [14] is a batch algorithm for training linear classifiers. The main idea of linear classifier is to construct a feature vector as one representative G_i for each class C_i (category). To classify a request document X, in this paper, we compute the *cosine similarity* between X and each representative G_i, and assign X to the class whose representative has the highest degree of cosine similarity with X.

For each class C_i, linear classifier computes prototype vector $G_i = (g_{i,1}, \ldots, g_{i,n})$, where n is the dimension of the term space and each element $g_{i,j}$ corresponds to the weight of the jth term of G_i. Rocchio compute G_i is as $\vec{G_i} = \frac{\sum_{I \in P_K} I}{|P_K|} - \eta \frac{\sum_{I \in N_K} I}{|N_K|}$ where P_K is the set of positive instances, N_K is the set of negative instances, and η is the parameter to adjust the relative impact of positive and negative instances.

2.2 Widrow-Hoff

Widrow-Hoff [14] is an online learning algorithm for training linear classifiers. It runs through the training example one at a time to update a weight vector at each step.

We denote weight vector before the ith training example by w_i. Initially, the weight vector is set to all zeros, $w_1 = (0,...,0)$. At each step, the new weight vector w_{i+1} is computed from the old weight vector w_i using training example x_i with label y_i. The jth component of the new weight ector is computed as $\omega_{i+1,j} = \omega_{i,j} - 2\eta(\omega_i \cdot x_i - y_i)x_{i,j}$ The parameter $\eta > 0$ is the learning rate which controls how quickly the weight vector ω is allowed to change, and how much influence each new example has on it.

2.3 K-Nearest Neighbor

K-Nearest Neighbor is a lazy learning algorithm. Unlike other machine learning algorithms, it doesn't have the training phase. When a new instance comes, we calculate the cosine similarity of the new instance with every instance in the training set. We rank the training instances by their cosine similarity at descending order. Then the top k instances are selected for determining the category of the new instance. For each instance at the top k rank, we add its cosine similarity to the category it belongs. Then we assign the new instance to the category with the highest summation of cosine similarity.

2.4 Decision Tree

Decision tree learning [18] is a practical method for inductive inference. A decision tree based classification learning consists of two steps, tree induction and tree pruning. In tree induction step, a tree is induced from the given training set. In tree pruning step, the induced tree is made more concise and robust by removing any statistical dependencies on the specific training data set. Tree induction consists of two phase at each of the internal nodes. First phase makes a splitting decision based on optimizing a splitting index. We call this phase the splitting determining phase. The second phase is called the splitting phase. It splits the records into children nodes based on the decision made. The process stops when all the leaves have records bearing only one class label.

2.5 Neural Networks

The study of artificial neural network has been inspired in part by the observation that biological learning systems are built of very complex webs of interconnected neurons.

In this paper we use three-layer network model [18] with input layer, hidden layer and output layer. We use the Backpropagation algorithm [16] to update our weights. The Backpropagation algorithm contains two phase, Forward phase and Backward phase. In the Forward phase, we will compute the values of each output layer unit by the weights on the arcs. In the Backward phase, we update the weights on the arcs by gradient descent method [16].

2.6 Support Vector Machines

SVMs (Support Vector Machines) [23] is a new machine learning method invented by Vapnik and his group at AT&T Bell Laboratories in 1995. The main idea of SVMs is to separate the classes with a surface that maximizes the margin between them. Given a set of training examples $(x_{i,1}, x_{i,2}, ..., x_{i,n})$, $i = 1, \ldots, l$ and we can define the class label y_i as follows:

$$y_i = \begin{cases} 1 & \text{if } x_i \text{ is in class 1} \\ -1 & \text{if } x_i \text{ is in class 2} \end{cases}$$

2.7 Naive Bayes

Naive Bayes (NB) probabilistic classifier is commonly studied in machine learning [18, 29]. The basic idea of NB is to use the joint probabilities of terms and classes to estimate the probabilities of classes given a document.

3 Classifier Combination

In this paper we compare five simple aggregation methods and two meta-learning methods. Note that we define a single-method classification as *level-0* classification and the one generated by combining method: *level-1* classification. These methods are described as follows.

3.1 Voting

Voting or Majority Voting [13] is the simplest and intuitive aggregation method. It classifies a document to the class with maximum counts of class predictions of classifiers. There is an assumption that the results of *level-0* classifiers should be diverse if using Voting method. For example, we have three *level-0* classifiers $\{h_1, h_2, h_3\}$ and consider a instance X. If these three classifiers are

identical (i.e., not diverse), then when h_1 is wrong, h_2 and h_3 will also be wrong. However, if the errors made by the classifiers are uncorrelated, then when h_1 is wrong, h_2 and h_3 may be correct, so that a majority vote will correctly classify X.

3.2 Maximum Precision

We use the class precision of each *level-0* classifier as the scoring function. One classifer with high precision on a specific class means that it performs good prediction on that class. We estimate the class precision of each classifier by evaluating performance with the validation set which are derived from the training data. Note that the training data are divided into a training set and a validation set to avoid the overfitting problem [18].

3.3 Behavior Knowledge Space(BKS)

Many methods of classifier combination assume that classifiers are independent to each other for simplicity. To avoid this assumption, we use the knowledge space derived from the decisions of all classifiers on each learned sample.

Let (L_1,..., L_n) be the class labels assigned to the instance X by classifiers F_1,..., F_n, respectively. Every possible combination of class labels is an index regarded as a cell in a look-up table (BKS table) [12]. The table is designed by using the data set Z (joint-distribution of the class labels). Let z_j be an instance in the data set Z, and it is placed in the cell indexed by $F_1(z_j)$,..., $F_n(z_j)$. Each entry in the BKS table is one of the following: a single class label (the one that is most often encountered amongst the element Z in the cell); no label (the cell is empty because no element of Z had the respective combination of class labels); or a set of tied class labels (if more than one class have the same highest number of elements in this cell). The decision for an instance X is made according to the class label of the cell indexed by $F_1(X)$,..., $F_n(X)$. Ties are broken randomly.

3.4 Weighted Voting

As voting method mentioned in section , it is not reasonable to give the same weight to each base classifier. Therefore, we give different weights for joining the class decision according to each classifier. There are studies about the weighting methods in [11, 17]. In this paper, we use the precision on each class evaluated with the validation set as the weight of base classifier.

From another point of view, we could adjust our weighted voting method by taking the probability model into consideration. We have the precision of one class as the probability that the instance X is correctly predicted on that class. Assume that the precision of a classifier F_i on class k is P_{ik}, the probability which F_i mispredicts on class j will be $1 - P_{ik}$. If two classifiers, F_i and F_j, predict the instance to class k, the probability which they both make mistakes on class k is $(1 - P_{ik}) * (1 - P_{jk})$. Note that we assume that each classifier will make errors independently.

3.5 NB Combination

The Naive Bayes(NB) classifier is one of level-0 classification and calculates the related probability for classification as introduced in Section . Because our meta-level training data are the predicted class labels which are the so-called nominal data [27], we can't use that class labels as a numeric value and need some modifications when applying it to meta-level(level-1) classification.

In NB combination method, we calculate the score(probability) of each class with our weighted voting method. We simply calculate scores of the classes which level-0 classifiers predict. In the level-0 training data, a document X is represented as $(t_1, .., t_n)$. However, for the level-1 training data, a document X is represented as $(f_1, \ldots, f_{|F|})$ where f_i is the predicted class label of *level-0* classifier F_i. Therefore, we have $P(X|C_k) = \prod_{j=1}^{|F|} P(f_j|C_k)$.

The term $P(f_j|C_k)$ is defined as $P(f_j|C_k) = \frac{1+|\{F_j(d)=f_j|d\in C_k\}|}{|class|+|C_k|}$, where $|class|$ is the number of classes in the training set and we have it to prevent the zero value of $|\{F_j(d) = f_j|d \in C_k\}|$.

Due to the independent assumption between base classifiers, we have the following equation as $P(X|C_i) = P(F_1 = f_1, F_2 = f_2, ..., F_{|F|} = f_{|F|}|C_i) = \prod_j^{|F|} P(F_j = f_j|C_i)$ We denote $P(F_j = f_j|C_i)$, where f_j is the class label predicted by F_j, as the weight of F_j in class C_i. Because the term $\sum_{j=1}^{|C|} P(X|C_j)P(C_j)$ is the same among these classes, we assign the class label whose $P(X|C)$P(C) value is the maximum to document X.

3.6 Decision Tree

The input data set of the decision tree method used for *level-1* classifier and for *level-0* classifier is different. In *level-0*, we give the document vectors as $(t_1, .., t_n)$ of training set to the decision tree classifier. However, in *level-1*, we use vectors $\{F_1(x), F_2(x),..., F_n(x)\}$ where

$F_i(x)$ is the predicted label by *level-0* classifier F_i.

4 Experimental Results

4.1 Transferring Chinese Documents into Vectors

The process of transforming Chinese documents into vector representations consists of term extraction [5, 21, 22], term selection [30], term clustering [3]. Considering term extraction, there are two main different model to vectorize Chinese texts: the n-gram-based model, and the word-based model [6]. In [6], Chiu has done an extensive comparison for both models for categorizing Chinese texts, and concluded that n-gram-based models perform almost equally well with word-based models. In order to avoid the situation in which a document might contain none of the selected terms, term selection will select a suitable large set of terms which may require large amount of computation time and memory in classification. We then perform term clustering to further reduce the dimension of the vector space. Via the combination of term selection and term clustering, the dimension of term space can be greatly reduced such that the computation of Chinese text categorization is more practical and efficient [25].

4.2 Performance Measure

We use the Micro-Accuracy and the Macro-Accuracy for our performance measure. The Micro-Accuracy measure is defined as follows:$Micro - Accuracy = \frac{\sum_{i=1}^{|C|} .H_{i,i}}{\sum_{i=1}^{|C|} |C_i|}$ where C_i is the set of instances belonging to class i and $H_{i,j}$ is the number of instances belonging to class C_i predicted to class C_j.

In order to see the bias situation [25] that some classifiers prefer large classes than small classes, we use Macro-Accuracy for our classifier combination experiments. The Macro-Accuracy measure is defined as follows:$Macro-Accuracy = \frac{\sum_{i=1}^{|C|} R_i}{|C|}$ where $|C|$ is the number of classes, R_i is the recall of class i.

4.3 Data Source

In the paper we have two data sources, one from the Central News Agency(CNA) [1] and another from the Web Openfind [2] for experiments. We give preliminary statistics of data sets that are used later, and briefly explain the preprocessing of Chinese documents which transforms a document into a vector in the vector space model. Note that we also do experiments on the other variations of this two data sets according to different term extraction methods or different numbers of selected term as disscussed in [6, 26]. However, we didn't list those results here for simplicity.

The CNA news articles spanning a period of one year, from 1/1/1991 to 12/31/1991, as the training data, and a period of one month, from 1/2/1992 to 28/2/1992, as the testing data. There are 12 classes from CNA. The training data consists of 55257 news documents and the testing data consist of 5007 news documents. Note that the training data are divided into a training set, 2/3 of the training data, and a validation set, 1/3 of the training data, to avoid the overfitting problem [18]. Considering the CNA news, we extract terms by bigram model and select 36000 terms with highest χ^2 statistic measure [30], then cluster there terms into 4500 groups via distributional clustering [3, 20].

From the web Openfind consisted of 80 classes, we use the web pages at the leaf nodes of that web whose directory is a hierachical structure. We use 10097 web pages for the training data and 4411 ones for the testing data. After term extraction and term selection, we choose 20000 terms from this data set.

4.4 Performance Comparison

In this section, we show the performance improvement by comparing the base classifiers with our combining methods, including Voting, Maximum Precision, BKS, Weighted Voting, NB Combination and Decision Tree.

Table 1 and Table 2 show the result on the CNA news data and web directory data, respectively. In Table 1, almost all the level-1 classifiers, no matter with combining training set or validation set, achieved better accuracy, above 78%, than 77.53% which was the value of the best accuracy achieved by the kNN among the level-0 classifiers.

In Table 2, among the level-1 classifiers, the Weighted Voting achieved the best accuracy, 68.28%, that was better than the accuracy, 61.73%, that was the best accuracy achieved by the WindrowHolf among the level-0 classifiers. The other level-1 classifiers, except the WindrowHolf, at least achieved similar accuracy, about 61%, to that achieved by the WindrowHolf. Note that the difference between "weighted voting" and "weighted voting 2" is that the former used the precision of each class to be the weight

of classifiers and the latter used what we modify to obey the probability model.

In Table 1 and Table 2, voting, naive bayes and our weighted voting methods all outperform the best level-0 classifier, especially these method are very simple. Combining methods, such as Maximum precision and Decision Tree, might fall into the overfitting problem. This situation was similar to that of SVMs which achieved accuracy on training sets are over 95%.

In order to see the bias situation that some classifiers prefer large classes than small classes, we evaluate the value of macro accuracy as follows. Table 3 and Table 4 show the macro accuracy of single methods and combination methods, respectively. We can see that the macro accuracy of our combination methods are stable. Unlike the single-method ones, our combination methods don't prefer to assign instances to the classes which are larger than others. In the web directory data set, we can see the situation more obviously. Many single-method classifiers get zero recall in these classes which are smaller. However, our combination methods don't have this situation.

5 Conclusions and Further Researches

In this paper we develop efficient techniques to combine the strength of well-known classifiers such as linear classifiers, decision trees, Bayesian methods, neural networks, and support vector machines. According to combination-based approaches, we develop and evaluate several classifier combination methods, including Voting, Maximum Precision, BKS, Weighted Voting, NB Combination and Decision Tree. Experiments show that our approaches significantly improve the classification accuracy of individual classifiers for Chinese texts classification as well as web page classification. Furthermore, our combination methods are more stable than single classifiers such that we avoid the bias situation that prefers large classes than small classes.

On the other hand, there is a question whether or not we can reduce the number of classifiers while maintaining the accuracy achieved by combining all classifiers. The classifier selection or model selection, which selects a small subset of most representative models to participate in the combination, is a very important phase to filter out useless models or to choose the most useful classifiers for combination. Beside the combination-based approach, there is another approach call the selection-based approach that a selecting classifier is trained to select a classifier for each new instance from the committee, which can best classify the new instance. We will discuss these related approaches in the future.

Acknowledgment. This work was partially supported by National Science Council, Taiwan, R.O.C., under grant NSC $90-2213-E-194-031$.

References

[1] Central News Agency. http://www.cna.com.tw/index.html.

[2] Openfind. http://www.openfind.com.tw.

[3] Douglas Baker and Kachites McCallum. Distributional clustering of words for text classification. In *Proceedings of the 21th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval(SIGIR'98)*, pages 96–103, 1998.

[4] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[5] Lee-Feng Chien. PAT-Tree-Based keyword extraction for Chinese information retrieval. In *Proceedings of the 20th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval(SIGIR'97)*, pages 50–58, 1997.

[6] Sheng-Bin Chiu. Comparing representatinos for chinese text categorization. Master's thesis, National Chung Cheng University, Taiwan, R.O.C., 2002.

[7] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods.* Cambridge University Press, 2000.

[8] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pages 148–156, 1996.

[9] Giacinto G. *Design of Multiple Classifier Systems.* PhD thesis, Univ. of California, 1998.

[10] Tin Kam Ho. Complexity of classification problems and comparative advantages of combined classifiers. *Lecture Notes in Computer Science*, 1857:97–106, 2000.

[11] T.K. Ho, J.J. Hull, and Srihari. Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1):66–75, 1994.

[12] DY.S. Huang and C.Y. Suen. A method of combining multiple experts for the recognition of unconstrained handwritten numerals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:90–93, 1995.

Table 1: Accuracy of combining methods in the CNA data with 4500 terms.

Combining Methods Comparison CNA News DATA (1991/1/1-1991/12/31) for training, (1992/2/1-1992/2/28) for testing Bigram model, 4500 Terms with Term Clustering, TFIDF Weighting				
	parameter	Testing	Training	Validation
ROCC	ANTA=0.25	0.740207834	0.747442038	0.74003026
SVM	RBF Kernel	0.759192646	0.950582345	0.794391008
NaiveBayes		0.731215028	0.745673234	0.735653302
KNN	K=15	0.775379696	0.859693045	0.813465903
WidrowHolf	Learning Rate=1/4x^2	0.773581135	0.802411016	0.79093267
Neural Network	Rate=0.3 Round=30000	0.73501199	0.924621748	0.78471847
Combining Training set				
Voting	ALL			0.784372502
MaxPrecision	ALL			0.769384492
NaiveBayes	ALL			0.786171063
Decision Tree	ALL			0.785572
BKS	ALL			0.782174261
Weighted Voting	ALL			0.788768985
Weighted Voting 2	ALL			0.789894148
Combining Validation set				
Voting	ALL			0.784372502
MaxPrecision	ALL			0.769522668
NaiveBayes	ALL			0.786299181
Decision Tree	ALL			0.7857
BKS	ALL			0.782304773
Weighted Voting	ALL			0.788895546
Weighted Voting 2	ALL			0.789294987

Table 2: Accuracy of combining methods in the web directory data with 20000 terms.

Combining Methods Comparison Openfind Web Directory Data with 80 classes N-gram-based model, 20000 Terms, TFIDF Weighting			
	parameter	Testing	Training
ROCC	ANTA=1.0	0.606211743	0.672576013
SVM	Linear Kernel	0.615733394	0.999801921
NaiveBayes		0.533665835	0.664652867
KNN	K=15	0.607345273	0.819847479
WidrowHolf	Learning Rate=1/4x^2	0.617320336	0.706942656
Neural Network	Rate=0.3 Round=30000	0.55701655	0.847281371
Combining			
Voting	ALL		0.674450238
MaxPrecision	ALL		0.616413512
NaiveBayes	ALL		0.679437769
Decision Tree	ALL		0.615733
BKS	ALL		0.682611653
Weighted Voting	ALL		0.682838359
Weighted Voting 2	ALL		0.619360689

Table 3: Macro-Accuracy of single methods in the CNA data with 4500 terms.

CNA News DATA (1991/1/1-1991/12/31) for training, (1992/2/1-1992/2/28) for testing Bigram model, 4500 Terms with Term Clustering, TFIDF Weighting							
	Class	ROCC	SVM	NaiveBayes	KNN	WidrowHoff	Neural Network
1	politics	0.779766537	0.903501946	0.634241245	0.927626459	0.832684825	0.853696498
2	economics	0.713763703	0.706455542	0.695493301	0.730816078	0.697929354	0.676004872
3	transport	0.767123288	0.719178082	0.828767123	0.75	0.842465753	0.70890411
4	education	0.702770781	0.697732997	0.740554156	0.753148615	0.755667506	0.647355164
5	sport	0.719626168	0.789719626	0.740654206	0.806074766	0.752336449	0.813084112
6	judiciary	0.709864603	0.767891683	0.754352031	0.736943907	0.725338491	0.723404255
7	stock	0.960784314	0.946078431	0.960784314	0.946078431	0.965686275	0.774509804
8	military	0.576086957	0.452898551	0.815217391	0.376811594	0.663043478	0.489130435
9	agriculture	0.717213115	0.655737705	0.75	0.68852459	0.75	0.692622951
10	religioin	0.671052632	0.407894737	0.684210526	0.394736842	0.592105263	0.421052632
11	finance	0.881578947	0.835526316	0.894736842	0.868421053	0.914473684	0.875
12	social welfare	0.717460317	0.644444444	0.765079365	0.698412698	0.765079365	0.685714286
	MacroAccuracy	0.743090947	0.710588338	0.772007542	0.72313292	0.77140087	0.696706593

Table 4: Macro-Accuracy of combining methods in the CNA data with 4500 terms.

CNA News DATA (1991/1/1-1991/12/31) for training, (1992/2/1-1992/2/28) for testing Bigram model, 4500 Terms with Term Clustering, TFIDF Weighting								
	Class	Voting	MaxPrecision	NaiveBayes	Decision Tree	BKS	Weight Vote	Weight Vote 2
1	politics	0.893385214	0.93307393	0.846692607	0.902723735	0.887159533	0.889494163	0.891828794
2	economics	0.728380024	0.658952497	0.723507917	0.694275274	0.716199756	0.721071864	0.721071864
3	transport	0.804794521	0.794520548	0.818493151	0.787671233	0.787671233	0.815068493	0.815068493
4	education	0.750629723	0.675062972	0.758186398	0.785894207	0.758186398	0.750629723	0.755667506
5	sport	0.778037383	0.862149533	0.822429907	0.820093458	0.808411215	0.829439252	0.829439252
6	judiciary	0.742746615	0.798839458	0.762088975	0.781431335	0.746615087	0.760154739	0.764023211
7	stock	0.960784314	0.965686275	0.965686275	0.960784314	0.960784314	0.965686275	0.965686275
8	military	0.536231884	0.264492754	0.615942029	0.532608696	0.547101449	0.536231884	0.528985507
9	agriculture	0.721311475	0.713114754	0.75	0.713114754	0.737704918	0.721311475	0.721311475
10	religioin	0.565789474	0.25	0.578947368	0.460526316	0.513157895	0.5	0.447368421
11	finance	0.907894737	0.894736842	0.914473684	0.855263158	0.861842105	0.914473684	0.914473684
12	social welfare	0.733333333	0.736507937	0.749206349	0.714285714	0.726984127	0.73968254	0.742857143
	MacroAccuracy	0.760276558	0.712261458	0.775471222	0.750722683	0.754318169	0.761937008	0.758148469

[13] Gareth James. *Majority Vote Classifiers: Theory and Applications.* PhD thesis, Dept. of Statistics, Univ. of Stanford, 1998.

[14] David D. Lewis, Robert E. Schapire, James P. Callan, and Ron Papka. Training algorithms for linear text classifiers. In *Proceedings of the 19th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval(SIGIR'96)*, pages 298–306, 1996.

[15] Richard Maclin and David Opitz. An empirical evaluation of bagging and boosting. In *AAAI/IAAI*, pages 546–551, 1997.

[16] Howard B. Demuth Martin T. Hangan. *Neural Network Design.* PWS Publishing Company, 1995.

[17] C.J. Merz. *Classification and Regression by Combining Models.* PhD thesis, Department of Computer Science, University of Salerno, 1998.

[18] Tom M. Mitchell. *Machine Learning.* The McGraw-Hill Companies, Inc, 1997.

[19] J. Ross Quinlan and Ross Quinlan. *C4:5: Programs for machine learning.* Matthias Zehe, 1993.

[20] Jyh-Jong Tsay and Jing-Doo Wang. Term selection with distributional clustering for Chinese text categorization using n-grams. In *Research on Computational Linguistics Conference XII, Taiwan, R.O.C*, pages 151–170, 1999.

[21] Jyh-Jong Tsay and Jing-Doo Wang. Design and evaluation of approaches for automatic Chinese text categorization. *International Journal of Computational Linguistics and Chinese Language Processing(CLCLP)*, 5(2):43–58, August 2000.

[22] Jyh-Jong Tsay and Jing-Doo Wang. A scalable approach for Chinese term extraction. In *2000 International Computer Sympoyium(ICS2000), Taiwan, R.O.C*, pages 246–253, 2000.

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

[24] V. Vapnik. *Statistical Learning Theory.* Wiley, 1998.

[25] Jing-Doo Wang. *Design and Evaluation of Approaches for Automatic Chinese Text Categorization.* PhD thesis, National Chung Cheng University, Taiwan, R.O.C., 2002.

[26] Yuan-Gu Wei. A study of multiple classifier systems in automated text categorization. Master's thesis, National Chung Cheng University, Taiwan, R.O.C., 2002.

[27] Ian H. Witten and Eibe Frank. *Data Mining Practical Machine Learning Tools and Techniques with JAVA Implementations.* Morgan Kaufmann, 1999.

[28] L. Xu, A. Krzyzak, and C.Y. Suen. Methods of combining multiple classifiers and their applications to handwriting recognition. *SMC*, 22(3):418–435, 1992.

[29] Yiming Yang. An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval*, 1(1/2):67–88, 1999.

[30] Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning (ICML'97)*, pages 412–420, 1997.