International Journal of Information Technology and Knowledge Management
July-December 2009, Volume 2, No. 2, pp. 263-267

COMBINING THE CLASSIFIERS AND LSI METHOD FOR EFFICIENT AND ACCURATE TEXT CLASSIFICATION

M. Srinivas[*], K. P. Supreethi[**] & Dr E. V. Prasad[***]

Text classification involves assignment of predetermined categories to textual resources. Applications of text classification include recommendation systems. Personalization, help desk automation, content filtering and routing, selective alerting, and training. This paper describes an experiment for improving the classification accuracy of a large text corpus by the use of dimensionality reduction and multiple-classifier combination techniques. Three different classifiers have been used namely Naive Bayes, Decision Tree and Association rule mining. The results of these classifiers are combined using techniques such as Simple Voting, Weighted Voting and Probability-based Voting. The classification accuracy is further improved by the use of a dimensionality reduction method called Latent semantic indexing (LSI). Experiments conducted on the Reuters 21578 dataset indicate that the combination approach provides an improved and scalable method for text classification. Also, it is observed that concept indexing helps with classification accuracy in addition lo efficiency and scalability.

1. Introduction

Text Classification [9] is the task of deciding whether a piece of text belongs to any of a set of pre-specified categories. It is a generic text processing task useful in indexing documents for retrieval, as a step in natural language processing, for content analysis and filtering, for email filtering, personalization, and in many other applications. A number of classifiers have been used in the past for the task of text classification. Each individual classifier has its strengths and weaknesses. With combination techniques the objective is to emphasize the strengths of individual classifiers while diluting their weaknesses. Such a combination of individual classifiers can be achieved in different ways. These include voting techniques, stacking, grading, bagging, and boosting among others. In this paper, three variations of the voting technique are used. They are Simple Voting, Weighted Voting and Probability-based Voting. Voting in general requires little additional processing to produce the final results.

The three methods of combination are explained in Section 3. A common feature of most text classification problems is the large number of attributes which are required to represent each document. Since each distinct word can be considered as an attribute, the number of attributes could increase rapidly with increase in dataset size. For example the Reuters-21578 dataset has over 2000 attributes in a 1000 documents dataset. These words include words from the dictionary, proper nouns, and acronyms. Such a large set of attributes, presents some problems for any text classification algorithm. The first problem is the limited memory. Mining algorithms that rely on memory are limited in their ability to handle large textual datasets. The second problem is the attribute relevance. Not every word or phrase is equally valuable for the task of classification. As a solution to these problems a dimensionality reduction methods based on LSI [5] was proposed. Interestingly, as a positive side effect, it is also observed that such a dimensionality reduction techniques can produce a visible improvement in the classification results. The combination of dimensionality reduction and disk-based processing techniques allow for the processing of large corpora in a scalable manner.

The rest of the paper is organized as follows: The next section explains the individual classifiers that were used for classification. In Section 3, the three combination techniques are defined, a dimensionality reduction algorithm based on concept indexing is explained in Section 4. Section 5 explains the proposed method in more detail. Experimental evaluation results are presented in Section 6. Section 7 discusses some of the related work and Section 8 concludes the paper.

2. Individual Classifiers

Three individual text classifiers were selected to study their performance for text categorization and use them for the combination techniques. There are many categories of classifiers including probability based, rule based, decision tree based, multivariate regression based, neural network based, and nearest neighbor based classifiers. For this study, the three classifiers were chosen from three different categories. A popular choice for the probability based classifier is the Naive Bayes Classifier [4] as it has been found to be particularly successful in text categorization.

[*] CSE Dept., JNTUACE, Anantapur, INDIA.
E-mail: sreenu2521@gmail.com

[**] CSE Dept., JNTUHCE, Hyderabad, INDIA
E-mail: supreethi.pujari@gmail.com

[***] JNTUKCE, Kakinada, INDIA. *E-mail: drevprasad@gmail.com*

Decision tree algorithms are a good choice for symbolic classification of textual data. C 4.5 [7] (or J 48 which is an implementation of C 4.5) is one of the most popular decision tree algorithms. The third choice was a rule based classifier Association rule mining [8] due to its simplicity. The main aim of this research was not the development of these individual classifiers and hence the standard implementations of the three classifiers which are available in a data mining tool WEKA (Waikato Environment for Knowledge Analysis) [11]: were used. WEKA offers a host of classifiers that can be used and any of them can be plugged into an application by invoking WEKA methods.

3. META CLASSIFIERS

Meta classification, using a combination of multiple classifiers, is a two set step process. First, each individual "base" classifier is built using the available training data. Next, the results of the base classifiers are combined to form a higher level, meta-classifier. We have examined three combinations for meta-classification. They are Simple Voting, Weighted Voting and Probability-based Voting. Each of these is described in detail below. A second level of combination, which we called the meta2-classifier, was also explored, which combines the results of the first level metaclassifiers. For the meta2-classifier, we have used the simple voting mechanism. In the examples used below, 3 base-Level classifiers A, E, & C and a dataset having class labels L1, L2 and L3 are assumed.

3.1. Simple Vote

In the simple voting mechanism each base classifier model has a single vote. For each test document, this vote is given to the class label returned by the base model. After all base classifiers have voted, the class label having maximum votes is selected as the correct class label for that document. As an example let's assume that classifier A returns L2, B returns L1 and C returns L2. According to the simple vote L2 will be chosen. In the case of a tie the contention can be resolved either by choosing one of the contending class labels randomly or by ignoring the test data instance all together. The first approach is chosen in this paper.

3.2. Probability Distributed Vote

Each classifier model outputs the probability values indicating the probability of a document belonging to each of the possible classes. We take these probabilities generated by each classifier and sum them up for all class labels. The class label having maximum probability of being correct is chosen as the correct class label. For example: A returns probability 0.5 for L1, 0.3 for L2 and 0.2 for L3. B returns probability 0.1 for L1, 0.4 for L2 and 0.5 for L3. C returns probability 0.3 for L1, 0.4 for L2 and 0.3 for L3. Since L2 has the greatest average probability of 0.36, it is chosen over L1 and L3.

3.3. Weighted Vote

The Weighted Vote mechanism is spilt up into three steps. In the first two steps the base classifier models are trained on the training data and their precision is recorded. This precision is used as the weight for each individual classifier, Thus, greater the precision of any classifier, the larger will be the weight associated with it. The third step is similar to simple vote, the only difference being that each base model vote is multiplied by its weight. This weighted vote is then used for selecting the class label. Here again in case of contention, the same policy as that in the Simple Vote method is used. For example, assume that for the training data classifier A has a precision of 0.7, B has a precision of 0.5 and C has 0.9. Then, if for a test instance A outputs L1 then L1 gets a vote of 0.8, B outputs L2 and hence L2 gets a vote of 0.5 and C outputs L3 and thus L3 gets a vote of 0.9. Here L3 gets selected as it has the highest probability of being the right class label. It can be seen that the contention issue in case of a tie is resolved automatically with the weights. This is an advantage weighted vote has over simple vote, based on the assumption that the training set precision values will rarely agree with each other.

4. LATENT SEMANTIC INDEXING. LATENT SEMANTIC

Indexing (LSI) is an information retrieval method for dimensionality reduction, where the emphasis is on capturing the underlying semantics or "latent" association in the pattern of terms or keywords used across documents. The mapping of original vectors into new vectors is based on the Singular Value Decomposition (SVD) applied to the original data vectors [1, 6]. Hence, let $A_{n\times m}$ matrix of rank *r,* whose rows represent terms and columns represent documents in the corpus. Let the Eigen values of AA^T be $\partial_1 \geq \partial_2 \geq \ldots \geq \partial_r$. Then the SVD of A becomes [6]:

$$A_{n\times m} = U_{n\times r} S_{r\times r} V_{m\times r}^{\ T} \quad (2)$$

$U_{n\times r} - (u_1, u2, \ldots u_r)$ is the term concept matrix, n terms, and r concepts, the columns are orthonormal.

$S_{r\times r} - \text{diag}(\partial_1, \partial_2, \ldots., \partial_r)$. S_u is the strength of concept i

$V_{m\times r} - (v_1, v2, \ldots v_r)$ is the document-concept matrix, m documents, and r concepts, the columns are orthonormal.

LSI works by omitting all but the k largest singular values in (3). Here, k is an appropriate value to represent the dimension of the low dimensional space for the corpus.

Hence, the approximation of $A_{n\times m}$ becomes:

$$A_{k\times m} = U_{n\times k} S_{k\times k} V_{m\times k}^{\ T} \quad (3)$$

Where the column vectors of $A_{n\times m}$ are projected to the k dimensional space spanned by the column vectors of and the rows of are used to represent the documents. Thus, LSI preserves the relative distances in the original data set while projecting it to a lower dimensional space using techniques from linear algebra.

5. PROBLEM SETTING AND PROCEDURE

The Reuters 21587 dataset used in this paper has been used extensively in many tasks related to Text Classification. It consists of a collection of 21587 news stories filed by the Reuters news network. The stories were assembled and indexed with categories by personnel from Reuters Ltd. and Carnegie Group, Inc. The stories are marked up in SGML and include a number of additional data items such as title, author, dateline, company etc. Text Classification is employed to predict the topic based on the contents of the stories and the title. For the experiments in this paper we created subsets of different sizes from the parent set. The size of each subset varied from 100 to 21587 stories. Each subset is then split into two, one for training and the other for testing.

5.1. Procedure

Figure 1 describes the steps of text classification used in this paper. The First Step is data preprocessing. It is usually the most time-consuming part of the entire process. It involves parsing, cleaning, integration, reduction, normalization, and transformation of the data, during parsing each story or document is converted into a document vector, with each of its dimension representing an attribute of the document. Cleaning and integration is done during parsing to handle inconsistencies in document structure. This involves taking care of the missing values and attributes in the data and re-consolidation of the document schema. As an intermediate step a stop list is applied to remove the high frequency terms. These are the words which are deemed irrelevant as they are not able to depict the context of any document. At this point the Porter Stemming Algorithm [6] [l0] is also applied. It removes most suffixes in the English Language using a five step linear algorithm. In each step if the given word satisfies the suffix rule then suffix is conditionally removed. Both, stop-list and stemming reduce the number of terms in the document. Next, data reduction is done on this large dataset in order to improve the efficiency of classification and make the data analysis easier. In conjunction with data reduction a concept indexing based dimensionality reduction is also performed. For this, a centroid matrix is first constructed from the training data. This matrix is then multiplied with training and testing data matrices to create the reduced representation of the two datasets. Next, each document vector is normalized to account for difference in the document lengths. Finally, the document vectors are transformed and written to disk in a format which could be used by the classifier for building the classification model. During the entire preprocessing stage only the centroid matrix and one document vector needs to be in the memory. The rest of the data resides on the disk.

In this project, the standard implementation of the Naïve Bayes [4], Decision Tree [7] and Association Rule Mining [8] given in the data mining tool WEKA [11] are used. First, the preprocessed training dataset is used to build the three models and then the models are tested for accuracy against testing dataset. Next, the results of individual base classifiers are combined to form meta-classifiers. These meta-classifiers are called Simple Vote, Weighted Vote and Probability-based Vote. The results of each of these meta-classifiers are then compared with individual classifiers. We also take this one step further by combining results of the meta-classifiers to form a meta 2-classifier. The combination method used for this med-classifier is again Simple Vote. Hence it is called Meta-Simple Vote Classifier.

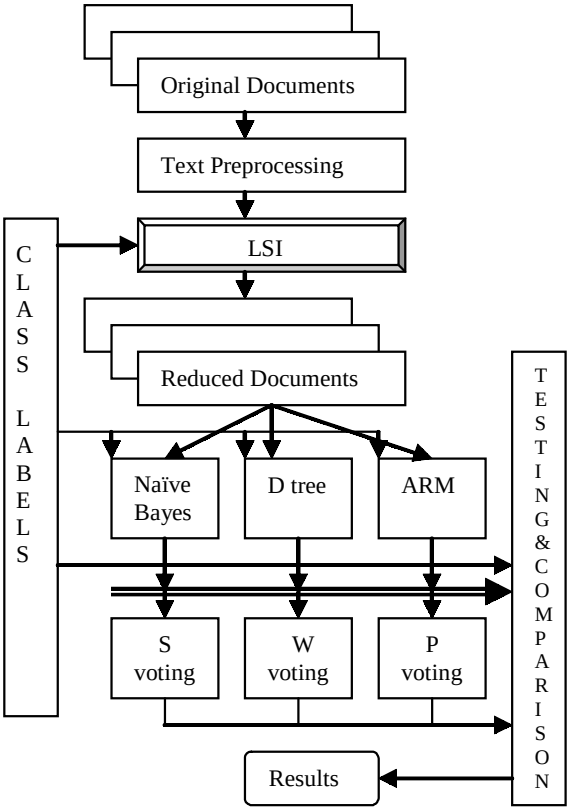

Figure 1: Diagram of the Procedure Used

6. EXPERIMENTAL EVALUATION

In our experimental evaluation we have used accuracy as the performance measure. It is defined as:

$$\text{Accuracy} = \text{number of correctly classified instances} \;/\; \text{total number of instances} \quad (4)$$

In all the experiments reported here percentage split was used as the evaluation technique. This consists of dividing the data into two subgroups. The first subgroup, called the training set, is used for building the model for the classifiers. The second subgroup, called the test set, is used for calculating the accuracy of the constructed model. For the purpose of this work two kinds of experiments were carried out.

6.1. LSI Experiment

The first set of experiments was to run the entire algorithm with and without LSI. Without LSI we were only able to run WEKA for an input data set size of 2000 documents, on a machine having 1GB of Main Memory with maximum allocation to the Java runtime environment. The tests were carried out for dataset sizes from 100 to 1000 in increments of 100 and for a dataset of size 2000 documents. The results are plotted on a graph showing the average difference in performance accuracy for all given classifiers and combination methods. By reducing the dimensionality we are losing some information but still are able to achieve a significant improvement in accuracy measure of our classification. This indicates that dimensionality reduction using LSI deemphasizes or removes the less significant features and emphasizes the important features of the documents. This results in an improvement is the classification accuracy as seen in most cases. The best results are seen with simple voting. document). The computational complexity of LSI is *O* (mn). Then, LSI employed respectively to reduce the dimension from 5612 to *k*, where *k* changes from 15 to 300. Figure 2. Shows Average increase in accuracy of each classifier when Dimensionality Reduction with LSI. Figure 3 Shows Increase in average accuracy for different dataset sizes due to Dimensionality Reduction with LSI. In these we can take different dataset sizes using dimensionality reduction method. We calculate average accuracy for these different datasets.

Classifiers / Combine Methods	*Improvement in Accuracy*
Decision Table	24.54%
Naïve Bayes	23.31%
ARM	24.89%
Simple vote	25.12%
Probability Distribution	22.45%
Weighted vote	24.36%
Average	24.07%

Figure 2: Average Increase in Accuracy of each Classifier when Dimensionality Reduction with LSI

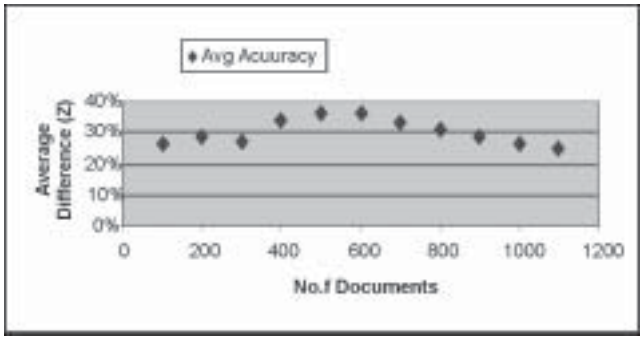

Figure 3: Increase in Average Accuracy for Different Dataset Sizes Due to Dimensionality Reduction with LSI.

6.2. Combination Methods Experiment

The second sets of experiments conducted were to compare the performance of the individual base classifiers with the combination methods. Extensive testing was carried out in this phase. Tests were conducted for data sets in increasing sizes from 1000 documents to 20,000 documents in increments of 1000, and then for the entire dataset. In figure 4 shows the result of the accuracy. This result shows the different classifiers combine methods with the three different classifiers. In this simple voting method give good accuracy than compare to the remaining methods. This method can be combine with the dimensionality reduction method is called LSI will be give better accuracy.

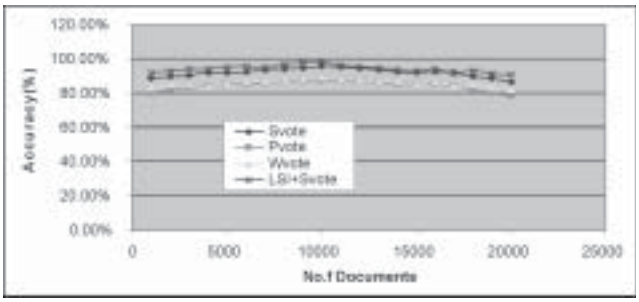

Figure 4: Accuracy of the Classifiers Combining Methods and Dimensionality Reduction Method.

7. RELATED WORK

Reports of various classifier combinations for text classification have been reported in the literature. Single pass combination techniques similar to the Voting mechanisms that were used in this paper are the Bootstrap Aggregation (or Bagging) [2] and Stacked Generalization (or Stacking) [12]. Both of these methods differ from Voting by the fact that the base classifiers use different subsets of the original training data, rather than the complete set. Also, unlike Voting and Bagging, Stacking uses a hierarchical system to combine the results of the base classifiers. Another class of classifiers is the Multi Pass Classifiers for which Boosting [3] is an example. Boosting tries to improve the performance by weighting the examples rather than the classifiers. While it does not guarantee an improvement in performance, it has been demonstrated to do that in various settings.

8. CONCLUSION

We have presented our experiences in using classifier combination methods and concept-based dimensionality reduction techniques for robust and scalable text classification. Our experimental evaluation confirmed the hypothesis that combination based meta-classifiers give better accuracy than individual classifiers for a popular textual dataset, the Reuters 21578 news dataset. Moreover, a significant performance gain was achieved when a concept based supervised dimensionality reduction algorithm was

applied to the original dataset. Most meta-classifiers outperformed the individual classifiers Naive Bayes and ARM by a big margin and Decision Tree by a smaller margin. The only exception was Simple Vote which had slightly lower accuracy when compared to Decision Tree. There was no one clear winner among the classifiers but, both Probability-based and Weighted Vote gave the best results. The superiority of Decision Tree over the Naive Bayes classifier was an unexpected outcome of this experiment and may warrant further research. Application of the Dempster-Shafer evidence combination method, which has been shown to be effective in other domains [l], is another area of further exploration. So using combine methods we can increase the classifiers accuracy. Using dimensionality reduction also we can increase the accuracy. Using these two methods we can increase the accuracy.

REFERENCES

[1] Y. A. Aslandogan and G. A. Mahajani. Evidence Combination in Medical Data Mining. IEEE International Conference on *Information Technology, Coding and Computing.*

[2] L. Breiman. Bagging Predictors. Much. Learn, **24**(2) (1996) 123-140.

[3] Y. Freund. Boosting a Weak Learning Algorithm by Majority. Information and Computation, **121**(2) (1995) 256-285.

[4] G. H. John and P. Langley. Estimating Continuous Distributions in Bayesian Classifiers. Proceedings of the Eleventh Conference on *Uncertainty in Artificial Intelligence.*

[5] Xiao Luo, A. Nur Zincir-Heywood, "Evaluation of Three Dimensionality Reduction Techniques for Document Classification", CCECE 2004 CCGEI 2004, Niagara Falls, May/mai (2004).

[6] M. E. Porter. An Algorithm for Suffix Stripping. Program, **14**(3) (1980) 13 &137.

[7] R. Quinlan. C 4.5 : *Programs for Machine Learning.* Morgan Kaufmann Publishers, San Mako, CA (1993).

[8] Ashok Savasere, Edward Omiecinski and Shamkant Navathe, *"An Efficient Algorithm for Mining Association Rules in Large Databases"* Proceedings of the 21st VLDB Conference Zurich, Swizerland (1995).

[9] T. R. Agrawal and A. Swami. *Database Mining:* A Performance Perspective. IEEE Trans. on Knowledge and Data Engineering, **5**(6), (7) (1993).

[I0] K. Sparck Jones and P. Willet. Readings in *Information Retrieval.* Morgan Kaufmann Publishers. San Francisco, (1997).

[11] I. H. Witten and E. Frank. *Data Mining:* Practical Machine Learning Tools with Java Implementations. Morgan Kaufmann Publishers, San Francisco, (2000).

[12] D. H. Wolpert, Stacked Generalization. Neural Networks. 5:241-259 (1992).