Feature Reduction for Neural Network Based Text Categorization

Savio L. Y. Lam & Dik Lun Lee
Department of Computer Science
Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong

Abstract

In a text categorization model using an artificial neural network as the text classifier, scalability is poor if the neural network is trained using the raw feature space since textural data has a very high-dimension feature space. We proposed and compared four dimensionality reduction techniques to reduce the feature space into an input space of much lower dimension for the neural network classifier. To test the effectiveness of the proposed model, experiments were conducted using a subset of the Reuters-22173 test collection for text categorization. The results showed that the proposed model was able to achieve high categorization effectiveness as measured by precision and recall. Among the four dimensionality reduction techniques proposed, Principal Component Analysis was found to be the most effective in reducing the dimensionality of the feature space.

1. Introduction

Text categorization is the classification of text documents into a set of one or more categories. In this paper, we propose a text categorization model using an artificial neural network trained by the Backpropagation learning algorithm [8] as the text classifier. In order to take advantage of the neural network model of computation, we formulate text categorization as a learned classification problem. We observe the following properties of text categorization that support this formulation:

1. The set of categories is usually pre-defined before the development of the categorization system, and remains unchanged for a long period of time. This is essential as the number and types of the categories can be taken into account in choosing the topology and size of the neural network classifier.

2. When a categorization system is developed, often there are a considerable number of documents that have already been categorized manually. These manually categorized documents can be used as training examples for training the neural network classifier.

The main difficulty in the application of neural network to text categorization is the high dimensionality of the input feature space typical for textual data. This is because each unique term in the vocabulary represents one dimension in the feature space. For a typical document collection, there are thousands or even tens of thousands of unique terms in the vocabulary.

Because of the high dimensionality in the feature space, the feature vectors are not suitable as input to the text classifier since the scalability will be poor [1, 2]. In order to improve the scalability of the text categorization system, we propose that dimensionality reduction techniques should be employed to reduce the dimensionality of the feature vectors before they are fed as input to the text classifier. In this paper, we study four dimensionality reduction techniques applicable to text categorization.

The rest of the paper is organized as follows. In the next section, we describe briefly the problem with text categorization. In Section 2, we present four dimensionality reduction techniques for text categorization. In Section 3, we discuss the neural network model used for text classification. In Section 4, the effectiveness of the dimensionality reduction techniques are compared, and the performance of the proposed text categorization model is evaluated. Finally, we give the conclusions in Section 5.

2. Dimensionality Reduction

In order to improve scalability of the text classifier, four dimensionality reduction techniques, namely the DF method, the CF-DF method, the TFxIDF method and Principal Component Analysis (PCA) [3], are applied to reduce the feature space. The aim of the techniques is to minimize information loss while maximizing reduction in dimensionality.

2.1. The DF Method

Given a set of training documents together with a specification of which of the pre-defined categories each training document belongs to, the DF method reduces the vocabulary size by term selection based on a local term ranking technique. The categorization information is used for grouping the training documents such that all the documents belonging to the same category are put into the same group. When there are overlaps between the categories, a document may belong to more than one group. After the documents are grouped, we can then form groups of the indexing terms in the vocabulary by putting in a group all terms contained in documents belonging to the same category. This process results in a set of sub-vocabularies corresponding to each category.

In the DF method, terms are ranked based on the *document frequency* (DF) of each term within a document group. For each document group, the document frequency of a term is defined as the number of documents within that particular group containing the term. By choosing the document frequency as the importance measure, we are assuming that the important terms are those that appear frequently within a group of documents belonging to the same category. This is because the set of terms which are good representatives of the category topics should be used by most documents belonging to that category.

Based on the DF importance measure, terms are ranked separately within each sub-vocabulary. For term selection, a parameter d is defined such that within each sub-vocabulary, only the most important d terms with the highest ranks are selected. The sets of selected terms from each sub-vocabulary are then merged together to form the reduced feature set.

By adjusting the selection parameter d, we can control the dimensionality of the reduced feature vectors. A smaller d will result in fewer terms being selected, thus higher reduction in dimensionality.

2.2. The CF-DF Method

From the discussion of the DF method, we observe that terms that appear in most documents within the whole training set will always have a high within-group document frequency. Even though these frequently occurring terms are of very low discrimination value, and thus not helpful in distinguishing between documents belonging to different categories, they are likely to be selected by the DF method. The CF-DF method alleviates the problem by considering the discrimination value of a term in the term selection process.

In the CF-DF method, a quantity called *category frequency* (CF) is introduced. To determine the category frequency of a term, the training documents are grouped according to the categorization information, as in the DF method. For any document group, we say that a term appears in that group if at least one of the documents in that group contains that term. For any term in the vocabulary, the category frequency is equal to the number of groups that the term appears in.

By this definition, terms that are concentrated in a few categories will have a low category frequency, while those that are distributed across a large number of categories will have a high category frequency. The idea is that the discrimination value of a term can be measured as the inverse of its category frequency. In other words, we assume that terms that are good discriminators are most likely concentrated in a few categories, and should be considered more important as they are helpful in distinguishing between documents belonging to different categories.

In the CF-DF method, a two phase process is used for term selection. In the first selection phase, we define a threshold t on the category frequencies of the terms, such that a term is selected only if its category frequency is lower than the threshold t. In the second selection phase, the DF method is applied for further term selection to produce the reduced feature set.

2.3. The TFxIDF Method

In the DF method and the CF-DF method, the essential idea is to perform ranking of the terms in the vocabulary based on some importance measure, such that the most important terms can be selected. In both of these methods, the key to minimize information loss as a result of term selection is to define a good importance measure so as to avoid filtering out terms that are useful for the text categorization task. A good measurement of the importance of a term in a document set is the product of the *term occurrence frequency* (TF) and the *inverse document frequency* (IDF). The inverse document frequency of the i^{th} term is commonly defined as [10]:

$$IDF_i = log\frac{N}{n}$$

where N is the number of documents in the document set, and n is the number of documents in which the i^{th} term appears. By this definition, a term that appears in fewer documents will have a higher IDF. The assumption behind this definition is that terms that are concentrated in a few documents are more helpful in distinguishing between documents with different topics.

In order to examine the effectiveness of this measure for term selection, we propose to use the $TF \times IDF$ value to measure the importance of a term for term selection. In other words, the terms are ranked according to their $TF \times IDF$ values, and a parameter d is set such that only the d

terms with the highest $TF \times IDF$ values are selected to form the reduced feature set.

2.4. Principal Component Analysis

Principal component analysis (PCA) [3] is a statistical technique for dimensionality reduction which aims at minimizing the loss in variance in the original data. It can be viewed as a domain independent technique for feature extraction, which is applicable to a wide variety of data. This is in contrast with the other three dimensionality reduction techniques we have discussed, which are domain specific feature selection techniques based on feature importance measures defined specifically for textual data.

In order to perform principal component analysis on the set of training documents, we represent the set of feature vectors by an n-dimensional random vector ($\mathbf{x}$):

$$\mathbf{x} = < x_1, x_2, \ldots, x_n >$$

where n is the vocabulary size, and the i^{th} random variable in $\mathbf{x}$ (x_i) takes on values from the term frequencies of the i^{th} term in the documents. We now find a set of n n-dimensional orthogonal unit vectors, $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$, to form an orthonormal basis for the n-dimensional feature space. We form projections of $\mathbf{x}$ (a_i) onto the set of unit vectors:

$$a_i = \mathbf{x}^T \mathbf{u}_i$$

In doing so, we perform a coordinate transformation in the feature space, such that the unit vectors ($\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$) form the axes of the new coordinate system and transform the original random vector $\mathbf{x}$ into a new random vector $\mathbf{a}$ with respect to the new coordinate system:

$$\mathbf{a} = < a_1, a_2, \ldots, a_n >$$

In principal component analysis, the choice of the unit vectors ($\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$) is such that the projections (a_i) are uncorrelated with each other. Moreover, if we denote the variance of a_i by λ_i, for $i = 1, 2, \ldots, n$, then the following condition is satisfied:

$$\lambda_1 > \lambda_2 > \ldots > \lambda_n$$

In other words, the projections a_i contain decreasing variance, These projections a_i are called the *principal components*. It can be shown [3] that the variance ($\lambda_1, \lambda_2, \ldots, \lambda_n$) corresponds to the eigenvalues of the data covariance matrix $\mathbf{R}$ arranged in descending order, and the unit vectors ($\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$) are the corresponding eigenvectors of $\mathbf{R}$. In order to reduce the dimensionality of the feature space from n to p where $p < n$ while minimizing the loss in data variance, we form a reduced feature space by taking the first p dimensions with the largest variance. In this case, the reduced feature vectors of the documents are represented by the p dimensional random vector:

$$\mathbf{a}_p = < a_1, a_2, \ldots, a_p >$$

3. The Neural Network based Text Classifier

By means of dimensionality reduction techniques, the set of documents to be categorized is transformed into a set of feature vectors in a relatively low dimensional feature space. This set of reduced feature vectors is then fed to the text classifier as input. In this paper, we use a 3-layer feed-forward neural network as the text classifier.

The neural network employed in our study is a 3-layer fully connected feed-forward network which consists of an input layer, a hidden layer, and an output layer. All neurons in the neural network are non-linear units with the sigmoid function as the activation function. In the input layer, the number of input units (r) is equal to the dimensionality of the reduced feature space. In the output layer, the number of output units (m) is equal to the number of pre-defined categories in the particular text categorization task. The number of hidden units in the neural network affects the generalization performance [4, 9]. The choice depends on the size of the training set and the complexity of the classification task the network is trying to learn, and can be found empirically based on the categorization performance.

For classification of the documents, reduced feature vectors representing the documents are fed to the input layer of the neural network classifier as input signals. These input signals are then propagated forward through the neural network so that the output of the neural network is computed in the output layer. As the sigmoid function is used as the activation function in the output units, the output of the neural network classifier is a real-valued classification vector with component values in the range [0, 1]. The classification vector represents a graded classification decision, in which the i^{th} vector component indicates the relevance of the input document to the i^{th} category. If binary classification is desired, a threshold can be set such that a document is considered to be belonging to the i^{th} category only if the i^{th} component of the classification vector is greater than the threshold.

The neural network classifier must be trained before it can be used for text categorization. Training of the neural network classifier is done by the Backpropagation learning rule based on supervised learning [8]. In order to train the neural network, a set of training documents and a specification of the pre-defined categories the documents belong to are required. More precisely, each training example is an input-output pair:

$$T_i = (D_i, C_i)$$

where D_i is the reduced feature vector of the i^{th} training document, and C_i is the desired classification vector corresponding to D_i. The component values of C_i are determined based on the categorization information provided in the training set.

During training, the connection weights of the neural network are initialized to some random values. The training examples in the training set are then presented to the neural network classifier in random order, and the connection weights are adjusted according to the Backpropagation learning rule. This process is repeated until the learning error falls below a pre-defined tolerance level.

4. Performance Evaluation

For performance evaluation, we used a subset of the documents from the Reuter-22173 test collection[1] for training and testing our text categorization model. We used a set of 400 documents as the training set for training the text classifier by Backpropagation. For testing, a test set of 1,508 documents was used. The set of test documents was kept unseen from the system during the training stage. In all experiments, we used a set of 10 categories from the TOPICS group defined in the Reuters test collection.

There are a total of 954 documents belonging to one or more of the 10 chosen categories, with some overlaps between different categories so that some of the documents belong to more than one category. To select the set of documents for training, we randomly chose 200 documents from these 954 documents to be put into the training set. These 200 documents served as positive examples for training the neural network based text classifier. In order to introduce a set of negative examples into the training set, another 200 documents not belonging to any of the 10 categories were randomly selected and added to the training set. These form a training set which consisted of 400 documents, with half of the documents being positive examples and the other half being negative examples.

In order to select documents for testing, we put into the test set the rest of the documents belonging to one or more of the 10 categories which were not selected as positive examples in the training set. To test the ability of the categorization model in filtering out documents that should not have any category assigned, we randomly selected 754 documents not belonging to any one of the 10 chosen categories and added them to the test set. This resulted in a test set of 1,508 test documents. Table 1 shows the distribution of documents among the 10 categories in the training set and the test set.

Name of category	No. of training documents	No. of test documents
TOPICS:money-supply	37	155
TOPICS:nat-gas	33	100
TOPICS:soybean	22	102
TOPICS:livestock	26	92
TOPICS:copper	15	65
TOPICS:yen	14	65
TOPICS:jobs	14	64
TOPICS:rice	18	53
TOPICS:cotton	17	48
TOPICS:rubber	12	42
Others	200	754
Total	400	1508

Table 1. The training set and the test set.

The documents in the training set were processed by word extraction, stop words removal, and stemming. For word extraction, we defined a word as any consecutive character sequence contained in the character stream of a document which starts with an alphabet, followed by any number of alphabets or digits. The end of a word is delimited by any non-alphanumeric character. Examples of acceptable words according to this definition include ”john”, ”art”, ”db2”, ”b12”, ”a300s”. The difference between upper case and lower case characters in the words was removed by converting all upper case characters to lower case. After a set of words was extracted, stop words were removed based on a stoplist for general English text. The remaining words were then stemmed using the Porter’s stemming algorithm [7]. After stemming, we merged the sets of stems from each of the 400 training documents and removed the duplicates. This resulted in a set of 4,718 terms in the vocabulary.

In order to create the set of initial feature vectors for representing the training documents, we measured the term frequencies (TF) for each term. The feature vectors were then formed by using the term frequencies as the feature values. This created a set of 400 feature vectors corresponding to each of the 400 training documents, where each feature vector was of dimensionality 4,718.

To reduce the dimensionality of these high dimensional feature vectors, different dimensionality reduction techniques were applied to reduce the dimensionality of the feature vectors from 4,718 to 328, with 93% reduction in dimensionality. This was done in all experiments except for the set of experiments explained in section 4.1 in which the reduced dimensionality was the varying parameter.

To create the set of classification vectors corresponding to each training document, the categorization information specified in the test collection was used. This created a set of 400 classification vectors corresponding to each of the

[1] The test collection can be downloaded by ftp at ciir-ftp.cs.umass.edu:/pub/reuters1.

400 training documents. Each classification vector was a 10-dimensional vector of the form:

$$C_j = < c_{j,1}, c_{j,2}, \ldots, c_{j,10} >$$

where $c_{j,i}$ was set to 1 if document j belonged to the i^{th} category and was set to 0 if document j did not belong to the i^{th} category.

With the set of reduced feature vectors and their corresponding classification vectors, we formed input-output pairs of the form (D_j, C_j) so that each of these pairs represented a training example. We used this set of 400 training examples to train a 3-layer feed-forward neural network using Backpropagation. The neural network had 328 input units, corresponding to the reduced dimensionality (except for the experiments in section 4.1). The output layer had 10 units, in which the output of each unit corresponded to the assignment decision for each of the 10 categories. The number of hidden units was set to 25. During training, training examples were presented to the network in random order, until the learning error was less than 0.01. Note that up to this stage, no test documents from the test set were used in any way for training or dimensionality reduction.

We processed the 1,508 test documents in the same way as the training documents by word extraction, stop words removal and stemming. This created a set of terms from the test documents. In order to keep the vocabulary the same as that of the training documents, we dropped all terms from the test documents that were not in the vocabulary formed from the training documents. Term frequencies were then measured for each of the remaining terms, with zero given to terms appearing in training documents only. By this, we created a set of 1,508 initial feature vectors corresponding to the set of test documents.

We need to reduce the dimensionality of these feature vectors created from the test documents. When the DF method, the CF-DF method and the TFxIDF method were used, dimensionality was reduced simply by dropping any dimension not corresponding to any of the dimensions in the reduced feature space of the training documents. In the case when principal component analysis was used, we created the set of reduced feature vectors from the test documents by multiplying the high dimensional feature vectors with the set of eigenvectors found during principal component analysis of the training documents. In other words, principal component analysis was not necessary on the set of test documents.

To test the performance of the trained text classifier, the set of reduced feature vectors created from the test documents was fed as input to it, and a set of output classification vectors was obtained. Each output vector was a 10-dimensional vector, with each component being a real number in the range [0, 1]. As we were interested in binary categorization only, we set a threshold at 0.5 such that if the i^{th} component of the output vector was greater than 0.5, we considered the decision of the text classifier was to assign the i^{th} category to the corresponding test document. This result was then compared with the categorization information specified for the test documents in the test collection, and the precision and recall were computed by macroaveraging [5].

All the experiments were conducted on Sun SPARCstation 20 machines running SunOS 4.1.4. Principal component analysis was performed using the matrix functions provided in MATLAB version 4.2c [6]. Neural network training and testing were done using the ”trainbpx” function provided in the Neural Network Toolbox of MATLAB, with the momentum constant set to zero and the error goal (tolerance level) set to 0.01. For each run, the time needed for training and testing was within one hour.

4.1. Comparison of Dimensionality Reduction Techniques

In the first experiment, the three feature selection techniques for dimensionality reduction, namely the DF method, the CF-DF method and the TFxIDF method were compared. In the second experiment, we looked at the ability of principal component analysis in reducing the high dimensional feature space into a much lower dimensional space without adversely affecting the categorization effectiveness. In both experiments, we tested the ability of the various techniques in retaining important information in the original feature space by decreasing the reduced dimensionality until there was a sharp drop in categorization effectiveness.

Experiment 1

Figure 1 shows the precision and recall of the DF method, the CF-DF method, and the TFxIDF method when the dimensionality of the reduced feature space was varied.

As shown in the figure, the TFxIDF method was the best among the three techniques in retaining important terms from the original vocabulary. This was observed by the fact that the method was able to reduce the dimensionality of the original feature space to 1,000 before there was a sharp drop in precision and recall due to the loss of terms important for the categorization task. This represented a reduction rate of 78.8%. The CF-DF method was second in effectiveness among the three. It was able to reduce the dimensionality by 36.4% to 3,000 before the precision and recall dropped dramatically. The least effective among the three was the DF method, which was only able to reduce the dimensionality by 15.2% to 4,000. Further reduction resulted in a sharp drop in precision and recall.

An implication of the results obtained in experiment 1 is that the $TF \times IDF$ score is a good measure for term im-

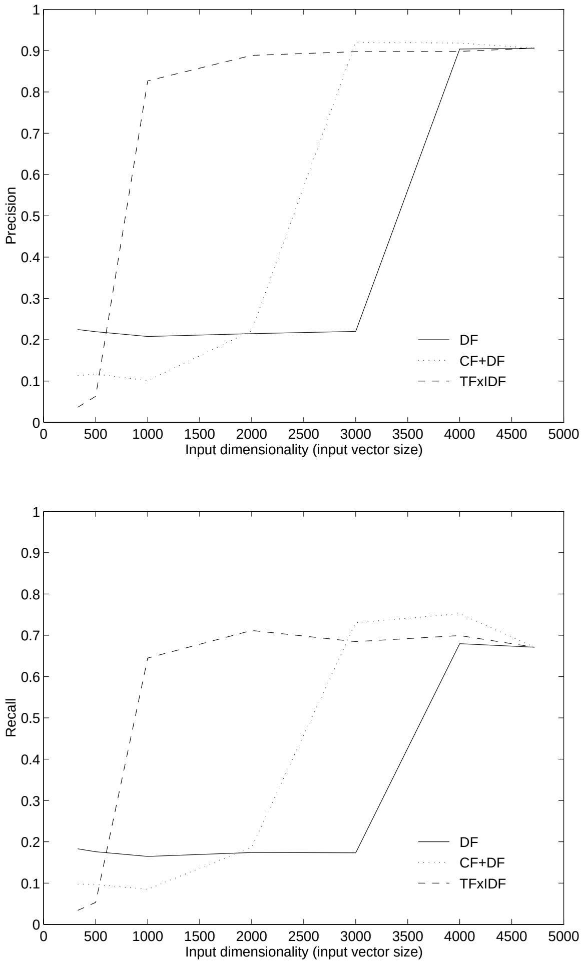

Figure 1. Precision and recall against input dimensionality.

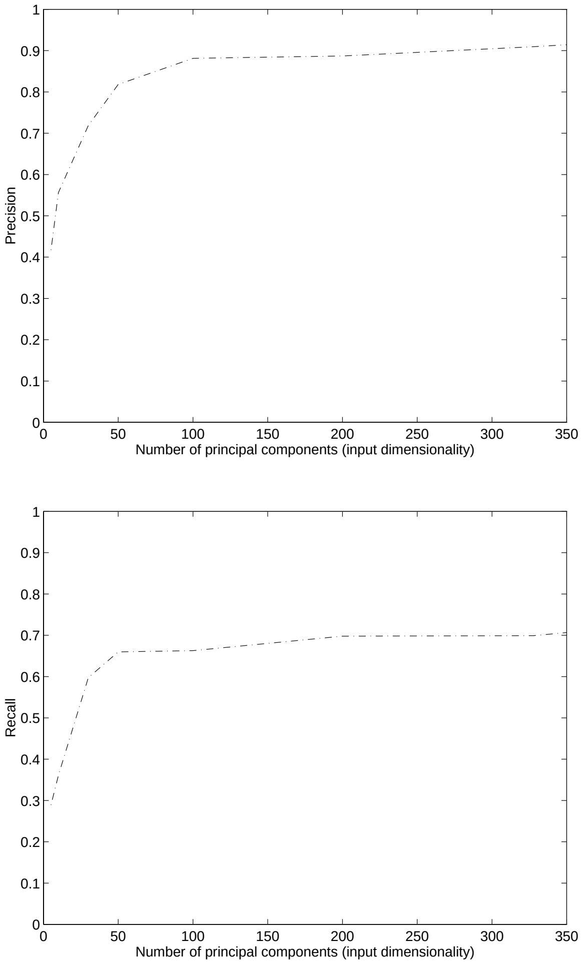

Figure 2. Precision and recall against number of principal components.

portance, which was not surprising given its proven effectiveness for term weighting in text retrieval. Comparing the results obtained for the DF method and the CF-DF method, it was concluded that the addition of the category frequency (CF) was effective in helping to screen out terms of very low discrimination values. However, the effectiveness was not as significant as when the $TF \times IDF$ score was used.

Experiment 2

In the second experiment, principal component analysis was tested for its ability in encoding important information from the high dimensional feature space into a reduced feature space of much lower dimensionality. We used similar testing methodology as in experiment 1, and varied the reduced dimensionality.

Figure 2 plots the precision and recall respectively against the reduced dimensionality. The results showed that principal component analysis was effective in reducing the dimensionality of the feature space from 4,718 to 50, and at the same time maintaining high precision and recall values. This represented a reduction rate of 98.9%, which was much higher than all of the three techniques we have tested in experiment 1.

From the results of experiments 1 and 2, we concluded that principal component analysis was the most effective among the four proposed dimensionality reduction techniques in compressing the high dimensional term space into a much lower dimensional feature space, with minimal adverse effect in categorization effectiveness as measured by precision and recall. We also noticed that the neural network based text classifier was able to obtain reasonably high precision and recall values. This shows that neural networks trained by Backpropagation is effective in performing the text categorization task.

4.2. Varying the Training Set

In the first experiment (experiment 3), we varied the size of the training set by incrementally removing randomly selected training examples from the original training set of 400 examples. In the other experiment (experiment 4), the composition of the training set was varied by changing the ratio of positive and negative examples without changing the overall size of the training set.

Experiment 3

In this experiment, the size of the training set was decreased stepwise from 400 to 50, with a decrement of 50 in each step to find out its effects on precision and recall. Starting with the original training set of 400 training examples, we randomly selected 50 training examples in each step and removed them from the current training set to form the next smaller training set. Precision and recall were computed for each training set in each step. This was done until a training set of only 50 training examples was formed in the last step.

Besides varying the size of the training set, we also compared the precision and recall for five different settings in which either no dimensionality reduction was done[2], or dimensionality reduction was done by the four proposed dimensionality reduction techniques to form a reduced dimensionality of 328.

Figure 3 shows that the DF method, the CF-DF method and the TFxIDF method have poor precision and recall. This was mainly due to the fact that the reduced dimensionality was kept low at 328 dimensions. From the figure, we also observe that the precision and recall obtained when principal component analysis was used for dimensionality reduction were very close to those obtained when there was no reduction in dimensionality. The implication is that principal component analysis is able to encode most of the information in the original feature space that are useful for text categorization in a much lower dimensional space.

When the size of the training set was decreased, both precision and recall were reduced. However, we observe that the reduction in precision was much smaller than the reduction in recall when the size of the training set was decreased. Recall was reduced as when less number of training examples were presented to the neural network, many test documents belonging to the categories but were not similar to the training documents were missed. On the other hand, precision remained high as the neural network assigned categories to test documents similar to the training documents, and most of these assignments were correct. The much higher sensitivity of recall to variations in the size of the training set implies that a large training set should be used in order to ensure high recall values. On the other hand, in situations where precision is more important, a relatively small training set can be used.

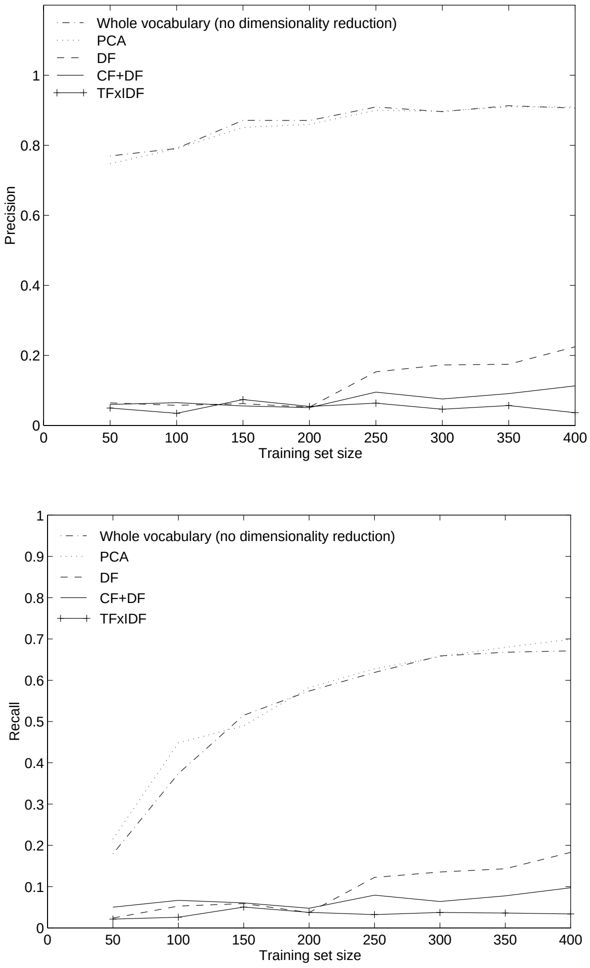

Figure 3. Precision and recall against size of training set.

Experiment 4

In the last experiment, the composition of the training set was changed by varying the ratio of positive and negative examples in the training set. To do this, we started with a training set with all 400 training examples being positive examples. In other words, each training example belonged to at least one of the 10 chosen categories. From this training set, we decreased the number of positive training examples incrementally, from 400 to 200, with a decrement of 40 in each step. During each step, 40 positive examples were

[2] The input to the neural network classifier in this case was the high dimensional feature vectors with dimensionality equals to 4,718.

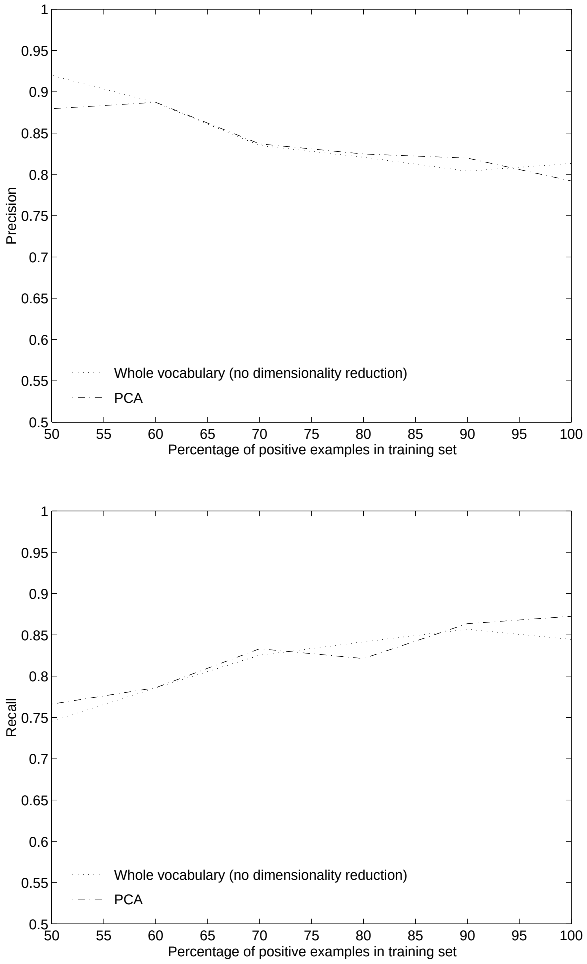

Figure 4. Recall and percentage of positive examples in training set.

randomly chosen from the current training set and removed. These were replaced by 40 negative examples, which were documents not belonging to any of the 10 categories. In this way, we were able to change the ratio of positive and negative examples without changing the overall size of the training set. Precision and recall were computed in each step.

The results of the experiment are shown in figure 4. As shown in the figure, recall increased with an increase in the percentage of positive examples in the training set. However, the opposite was true for precision.

The results show that increasing the number of positive examples in the training set was important for improving recall. However, the presence of negative examples in the training set was also important for improving precision. This can be explained by the fact that negative training examples can improve the ability of the text categorization system in screening out documents not belonging to any of the pre-defined 10 categories. This represents a tradeoff between improving recall and improving precision when the ratio of positive and negative examples in the training set was changed.

5. Conclusions

We studied and empirically tested the categorization effectiveness and feasibility of a text categorization model based on a 3-layer feed-forward neural network training by Backpropagation. Four dimensionality reduction techniques were proposed to reduce the high-dimension feature space typical for textual data into a low-dimension input space for the neural network.

Experiments were conducted using the proposed model to categorize real-world full-text newswire articles contained in the Reuters-22173 test collection for text categorization. The results showed that Backpropagation learning in neural networks was able to give good categorization performance as measured by precision and recall. By comparing the four techniques, principal component analysis was found to be the most effective as measured by the amount of reduction in dimensionality. In the experiments conducted, a high reduction rate of 98.9% was achieved by principal component analysis with insignificant decrease in categorization effectiveness.

References

[1] E. B. Baum and H. David. What size net gives valid generalization? *Neural Computation*, 1(1):151–160, 1989.

[2] G. E. Hinton. Connectionist learning procedures. *Artificial Intelligence*, 40(1–3):185–235, 1989.

[3] I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 1986.

[4] K. Knight. Connectionist ideas and algorithms. *Communications of the ACM*, 33(11):59–74, 1990.

[5] D. D. Lewis. Evaluating text categorization. In *Proceedings of the Speech and Natural Language Workshop*, 1991.

[6] MathWorks, Inc. *MATLAB : high-performance numeric computation and visualization*. MathWorks Inc., Natick, Mass., 1992.

[7] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[8] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In *Parallel distributed processing : explorations in the microstructure of cognition*, chapter 8. MIT Press, Cambridge, MA, 1986.

[9] D. E. Rumelhart, B. Widrow, and M. A. Lehr. The basic ideas in neural networks. *Communications of the ACM*, 37(3):87–92, 1994.

[10] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.