Category Factor Based Feature Selection for Document Classification

YunHee Kang*
Department of Computer and Communication Engineering
Cheonan University , Cheonan, Korea

ABSTRACT

According to the fast growth of information on the Internet, it is becoming increasingly difficult to find and organize useful information. To reduce information overload, it needs to exploit automatic text classification for handling enormous documents. Support Vector Machine (SVM) is a model that is calculated as a weighted sum of kernel function outputs. This paper describes a document classifier for web documents in the fields of Information Technology and uses SVM to learn a model, which is constructed from the training sets and its representative terms. The basic idea is to exploit the representative terms meaning distribution in coherent thematic texts of each category by simple statistics methods. Vector-space model is applied to represent documents in the categories by using feature selection scheme based on TFiDF. We apply a category factor which represents effects in category of any term to the feature selection. Experiments show the results of categorization and the correlation of vector length.

Keywords *: Document classification, Vector-space model, SVM, Feature selection, Category factor.*

1. INTRODUCTION

With the growth of online information like the World Wide Web (WWW), a large collection of fulltext document is available. Since the information on the Internet grow exponentially, it is difficult to find proper documents and to organize them. There have been many approach to solve the problem such as search engine, meta search engine and directory service. The subject-based directories such as Planet Earth and Yahoo provide a useful browsable organization of information and the efficient navigation path based on hierarchical structure. However, they cannot keep pace with amount of information because they are processed manually. As the growth of the Internet, to reduce information overload, it is necessary to exploit automatic text classification for handling enormous documents [1,2]. Text classification or categorization is the process of organizing information logically. It can be used in many fields such as document retrieval, routing, and clustering. The document classifier is used for classifying documents based on category [1,2,5].

In this paper, a new document classification system is proposed for categorizing web documents in the fields of Information Technology. The system consists of feature selection, term-frequency / inverse-document-frequency (TFiDF) and classification, Support Vector Machine (SVM) [3] for multi-class problem. The basic idea is to exploit the representative terms meaning distribution in coherent thematic texts by simple statistics methods.

The SVM, which is based on minimizing the expected risk of making a classification error, finds an optimal separating hyperplane between binary classes[3,4,6,7]. Using this hyperplane, it is possible to discriminate text collection between a relevant documents set and an irrelevant documents set. To this end, the SVM has recently been applied to a number of applications, e.g., image classification[3,4] and text categorization[6] .

To construct learning vectors for SVM learner, vector-space model is applied to represent documents in a category by using feature selection scheme based on TFiDF which is one of the most common feature extractor for documents. The learning vectors are composed of positive and negative examples which are represented by the set of weighted features.

In some document classification applications, it might be desired to pick a subset of the original features rather then select all of the original features. The benefits of finding this subset of features could be in cost of computations of unnecessary features.

We applied these two methods into multi-class document categorization problems. We apply a category factor which represents effects in category of any term to the feature selection. In experiment, we show the result of classification and evaluate the results based on recall, precision and F-measure criteria.

The rest of the paper is organized as follows: Section 2 describes related works including feature selection and a machine learning method SVM. Section 3 describes the proposed document classification system. In section 4, we show that our proposed system outperforms in experiments. We conclude in section 5.

2. RELATED WORKS

2.1 Feature Selection

Documents can be represented by sparse vectors, which consist of each unique words extracted from the documents.

*Corresponding author. Email: yhkang@cheonan.ac.kr
Manuscript received Aug. 29, 2005 ; accepted Dec. 19, 2005

Typically, the information vector is very high dimensional, at least in the thousands for large collections. The dimensionality reduction is an example of feature selection which is a technique to extract the feature data from observed raw data and is an important issue, especially in the case of high dimensional data. When reducing the vocabulary size, feature selection is done by selecting words that have highest average mutual information with the class variable [9]. Lewis has observed that "many algorithms scale poorly to domains with large numbers of irrelevant features,"[8] and it is not uncommon to have thousands of terms in the vocabulary of a text filtering system[1].

The removed features are not used in the computation anymore. The aim of feature selection method usually determines a low dimensional d features from the set of features m, for which a criterion J will be maximized.

In this work, we extract keywords to represent each document with a well-developed keyword extractor. This extractor excludes stopword which does not give any information about the documents such as 'the', 'of', 'and', 'to'. So this exclusion makes the algorithm fast and efficient. In our proposed system, we can keep and modify easily the stopword list file. In addition, it stems a word to make a robust keyword against conjugation.

The original raw data which are from web page usually include non-text contents such as META tag and images, so we first exclude the non-text data from web page. Nevertheless, even the plain text is not proper because the length of the plain text page is different from others and can be very long. Nevertheless, there are too many keywords to express efficiently the all documents. So we select k keywords by the frequency in whole documents which leads to keyword vector. In addition to this vector, we add the representative keyword vector which was build up by experts. After making a vector which consists of keywords, we make a feature vector from keyword vector to represent each document effectively.

There are many methods to make a feature vector from keyword vector. In this paper, the threshold methods are applied for term-number reduction in text area. Threshold methods are based on removal features, whose frequencies are greater than a defined threshold value. These methods are currently very popular because they are reasonably fast and efficient. On the other hand, they completely ignore the existence of other features and evaluate every feature with its own. Typical examples of threshold methods are: DF(Document Frequency), IG(Information Gain), MI(Mutual Information), $\chi 2$, TS(Term Strength), LSI(Latent Semantic Indexing). For more details see [5].

To be able to classify documents, one must find a way to represent documents in which this representation preserves as much of the original information as possible and also is simple enough in the point of computational complexity [2]. One of the well-established techniques for text in the field of IR(Information Retrieval) is to represent each document as a TF*IDF-vector in the space of keywords that appeared in training documents [2,8], sum all interesting document vectors and use the resulting vector as a model for classification. Each component, d(i) of a document vector is calculated as follows:

$$d(i) = TF(w_i,d) * IDF(w_i) \qquad \text{Eq.(1)}$$

$TF(w_i,d)$ represents the number of times word w_i occurred in document d and $IDF(w_i)= \log D/ DF(w_i)$ where D is the number of documents and document frequency $DF(w_i)$ is the number of documents, where a word w_i occurred in at least once.

The equation Eq.1 means that the more a word appears in documents, the less important measure to represent the documents. On the contrary, calculating TF is independent with other documents and TF means the importance of keyword to measure the difference from other documents. Finally, we have a vector *d(i)* for each document.

2.2 SVM for Document Classification

A number of statistical and machine learning techniques have been applied to text categorization [2]. Among those techniques for classification problem, SVM has been shown to be efficient and effective for text categorization [6,7]. The SVM is a learning method introduced by Vapnik [3] and has been applied efficiently to various problems. This method has a good property that the learning is independent of the dimensionality of the problem, while text categorization problem has very high dimensional data.

A SVM is a classifier based on structural risk minimization [3]. A linear SVM is a hyperplane that separates positive training data from negative training data with maximum margin which minimizes the structural risk. First, SVM is summarized as follows:

Consider a set of data points, $\{(x_1, y_1),..., (x_N, Y_N)\}$, where x_1, y_1 is an input and target data, respectively. An SVM is a model that is calculated as a weighted sum of kernel function outputs. The kernel function of an SVM is written as $K((x_1, y_1)$, which is considered as an inner product in feature space, that is, K is a Gram matrix in feature space. This means that we do not need to transfer all data into feature space. All what we need to calculate is the kernel function such as polynomial, radial basis function and other kernel functions which obey Mercer's condition. We do not cover kernel functions in more detail.

An SVM is a binary classifier and the number of category is M, so in order to apply this binary-classifier into M -class problem, we make M SVMs and each SVM uses *1* class as positive class and *M-1* classes as negative class. In practice, we use SVM^{light}[7] and each SVM uses radial basis function as kernel function. The detail of the overall system is described in next section.

3. THE OVERALL DOCUMENT CLASSIFICATION SYSTEM

For the construction of the classification system, this paper employs preprocessing on the documents selected from the web. In preprocessing, input feature vector is constructed for training(eliminating the unnecessary terms from the terms selected from the data) according to the frequency. The terms that constitute feature vectors are given either the positive weight or the negative weight through the collection of the representative terms of each category [7].

After learning the data set of each categories using SVM, the learner generates a model for classification. The classifier categorizes new documents taken in on the basis of the constructed training model and constructs models by putting

input feature vector, which is made through the process of feature selection from the overall document, in each learner based on category. It is concluded that the categories of the test documents are determined by the value which is generated by each classifier of a learning model. The number of SVM models is equal to that of categories to be created the classification models. Fig. 1 shows the procedure of the document classification.

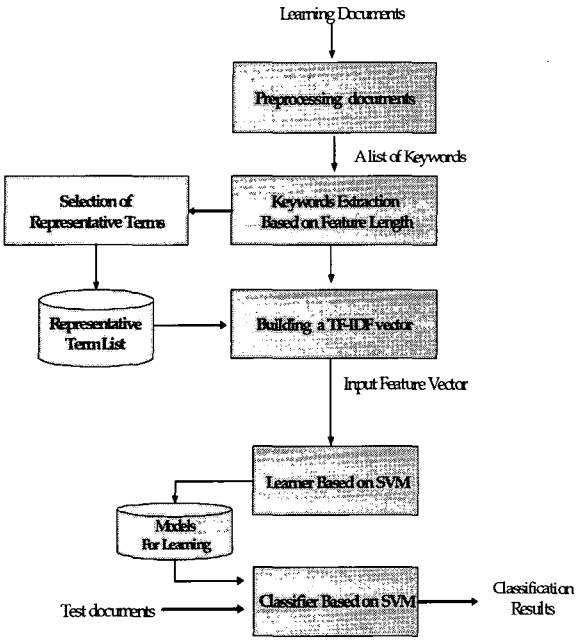

Fig. 1. The Document Classification procedure

As shown in Fig. 1, the documents for training create the extended TF-IDF vectors through the selection process according to frequency after the elimination of HTML tags and the preprocessing of stopwords. The TF-IDF vector is inputted to a SVM learner, and the learner creates the models for classification. The document for classification is to be transformed into an input vector through the preprocessing that is the same as training, and then is classified based on the training models.

4. EXPERIMENTS AND EVALUATION

4. 1 Experimental Design

For this experiment, the SVMs model is constructed by using SVM^{Light} [7] Individual SVMs create models for document classification using training documents. Each SVM is used to learn the training set for the models using both positive examples and negative examples. This experiment draws up the scenario according to two criteria as follows:

- performance evaluation when a new category is to be added.
- performance evaluation according to the length of vector space.

In this experiment, the feature selection module selects the terms which are based on frequency and category relationship. In the first step, representative terms are built, which are based on frequency in any category. To select the appropriate terms, Eq. 2 is applied for calculating category factor which represents effects in category of any term T_j. It can be adjusted the weight value of terms based on frequency and category factor.

$$\text{if } (CDF_{ij}/TDF_j >= \tau) \text{ then } RT_i \leftarrow RT_i + T_j \quad \text{Eq.(2)}$$

, where CDF_{ij} represents frequency of a term i in a category j, TDF_j represents total frequency of term i and $\tau = 0.25$. RT_i denotes a set of representative terms of category j.
Also we consider the length of input vector, which is related with performance of classification. We set the length of vector such as 500, 1000 and 2000. To construct input feature vector and to improve the feature selection, we apply a procedure based on heuristic TF*IDF expression as follows:

1. Sort n terms by frequency order, where n is the number of terms in learning set except for stopword
2. Compute TF * iDF value of input feature vector
3. Adjust TF * iDF value by using category factor

SVM consists of learner and classifier. The following shows the format of input feature for learning and classification:

<class>::= +1 | -1 | 0
<feature> ::= integer
<value> ::= real
<line>::=<class><feature>:<value><feature>:<value>+

For example, the input feature vector shows a part of positive example, where '+' represents a positive example, '1' represents a feature number and '0.241480' represents a feature value.
+1 1:0.241480 2:0.241480 3:-114.326101 4:0.241480 5:-114.326101 6:0.241480 7:0.241480

4.2 Evaluation

In the first experiment, it is evaluated the performance of precision and recall for classifiers formed when a new category is added. In the next experiment, it is evaluated that the performance of precision and recall for classifiers formed by restricted vector length.

Fig. 2 shows performance of each category where the length of the feature vector is 500 and the number of test documents is 126. The classifiers are usually of similar or better effectiveness than other categories(class 1 and 4) which represent *Wire communication* and *Internet*. We observe that the two classes 1 and 4 have similar features. We evaluate the result of the experiment that two categories are more correlated each other. The other categories except for 1 and 4 have the precision rates, which are greater than 70%. For 3 out of 10 categories, category 8, 9 and 10, have 100% precision. In most cases precision levels reached more than 90% by refining sample or more training examples.

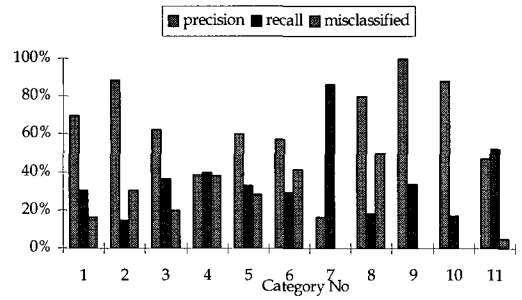

Fig. 2. Experiment-10 Categories

In case that a new category related with XML is added to the category set, Fig. 3 shows the results of the categorization where the length of feature vector is 2000 and the number of test documents is 208. In order to evaluate performance of classification it is compared with it both the original category set and other category set which is added to a new category represents a noise category. As shown in Fig. 3, the overall performance is lower than the first case as shown in Fig.3.

But we observe that most of categories have more than high precision reached more than 80 %. Also, we observed that the classifiers based on SVM are robust when the category set has noise documents for learning. In this case we got the performance that the average precision is 85.7%, the average recall is 78.8% and F_1 is 0.845.

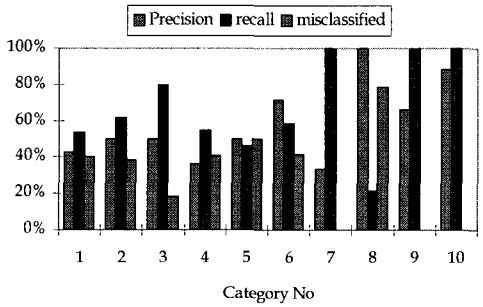

Fig.3. Experiment -11 Categories

Fig. 4 and Fig.5 show that the performance of categorization depends on the length of feature vector. As shown in Fig.5, as the length of feature vector is increased, the performance with respect to recall and precision is also increased. Especially, the category 1, named "wired communication" is higher from 43% to 77% in precision. Also the category 4, named "Internet" is higher from 30% to 71% in precision.

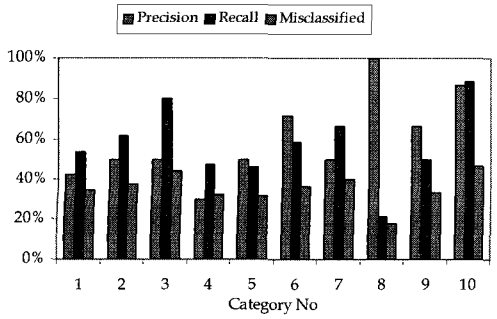

Fig. 4. Vector Length – 500

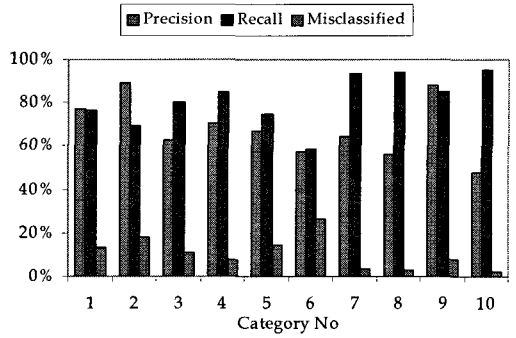

Fig. 5. Vector Length – 2000

5. CONCLUSIONS

In this paper, a document classification method is proposed, which is based on the extended TF*IDF feature selection for SVM. The basic idea is to exploit the representative terms meaning distribution in coherent thematic texts by simple statistics methods. Also we described the feature selection for document classifier of web documents in the fields of Information Technology.

In the first experiment, we evaluated the performance of precision and recall for classifiers formed, which is added to new category. In the next experiment, we evaluated the performance of precision and recall for classifiers formed by vector length. We evaluated the result of experiments that two categories are more correlated each other. The other categories except 1 and 4 have the precision, which is more than 70%. In most cases effectiveness, precision levels reached more than 90% by refining sample or more training. For these experiments we showed that the feature length affects highly performance of classification.

REFERENCES

[1] Tak W.Yan and Hector Garcia-Molina, Sift - a tool for wide-area information dissemination. In Proceedings of the 1995 USENIX Technical Conference, pages 177–186, 1995.

[2] Salton, G. Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer, Addison-Wesley, 1989

[3] V. Vapnik. Statistical Learning Tehory. John Wiley and Sons, Inc., New York, 1998.

[4] Chapelle O., Haffner P., and V Vapnik. SVM for histogram-based image classification. IEEE Trans. on Neural Networks, 10(5):1055--1065, 1999.

[5] Yang Y., J.O. Pdedersen, "A Comparative Study on Feature Selection in Text Categorization," Proc. Of the 14th Internatinal Conference on Machine Learning ICML-97, pp.412-429, 1997.

[6] Joachims, T., "Text categorization with support vector machines: Learning with many relevant features," Proc. European Conference on Machine Learning (ECML), pp. 137-142,1998

[7] Joachims, SVM^{Light}, http://svmlight.joachims.org/, 1998.

[8] D. Lewis, W. A. Gale, A sequential algorithm for training

text classifiers, Proc. SIGIR '94, pp. 3-12. Dublin, Ireland. 1994.
[9] Thomas M. Cover and Joy A. Thomas. Elements of Information Theory. John Wiley, 1991

YunHee Kang
He received the B.E., M.S in computer engineering from Dongguk university, Korea in 1989, 1991 respectively and also received Ph.D. in computer science from Korea university, Korea in 2002. He is an assistance professor in division of computer and communication at Cheonan University. His main research interests include information retrieval, Grid computing, distributed system and multi-agent system.