Text Categorization with Class-Based and Corpus-Based Keyword Selection

Arzucan Özgür, Levent Özgür, and Tunga Güngör

Department of Computer Engineering, Boğaziçi University,
Bebek, İstanbul 34342, Turkey
{ozgurarz, ozgurlev, gungort}@boun.edu.tr

Abstract. In this paper, we examine the use of keywords in text categorization with SVM. In contrast to the usual belief, we reveal that using keywords instead of all words yields better performance both in terms of accuracy and time. Unlike the previous studies that focus on keyword selection metrics, we compare the two approaches for keyword selection. In corpus-based approach, a single set of keywords is selected for all classes. In class-based approach, a distinct set of keywords is selected for each class. We perform the experiments with the standard Reuters-21578 dataset, with both boolean and tf-idf weighting. Our results show that although tf-idf weighting performs better, boolean weighting can be used where time and space resources are limited. Corpus-based approach with 2000 keywords performs the best. However, for small number of keywords, class-based approach outperforms the corpus-based approach with the same number of keywords.

Keywords: keyword selection, text categorization, SVM, Reuters-21578.

1 Introduction

Text categorization is a learning task, where pre-defined category labels are assigned to documents based on the likelihood suggested by a training set of labelled documents. Many learning algorithms such as k-nearest neighbor, Support Vector Machines (SVM) [1], neural networks [2], linear least squares fit, and Naive Bayes [3] have been applied to text classification. A comparison of these techniques is presented in [4].

Text categorization methods proposed in the literature are difficult to compare. Datasets used in the experiments are rarely same in different studies. Even when they are the same, different studies usually use different portions of the datasets or they split the datasets as training and test sets differently. Thus, as Sebastiani [5] and Yang and Liu [4] argue, most of the results in the literature are not comparable. Some recent studies consider different classification methods by using standard datasets [4,5,6,7], which enable us to compare these.

We use the standard Reuters-21578 dataset in our study. We have used ModApte split, in which there are 9,603 training documents and 3,299 test documents. We have used all the classes that exist both in the training and the test

P. Yolum et al.(Eds.): ISCIS 2005, LNCS 3733, pp. 606–615, 2005.

sets. Our dataset thus consists of 90 classes and is highly skewed. For instance, seven classes have only one document in the training set, and most of the classes have less than ten documents in the training set.

SVM, which is one of the most successful text categorization methods, is a relatively new method that evolved in recent years [4,7]. It is based on the Structural Risk minimization principle and was introduced by Vapnik in 1995 [8]. It has been designed for solving two-class pattern recognition problems. The problem is to find the decision surface that separates the positive and negative training examples of a category with maximum margin. SVM can be also used to learn linear or non-linear decision functions such as polynomial or radial basis function (RBF). Pilot experiments to compare the performance of various classification algorithms including linear SVM, SVM with polynomial kernel of various degrees, SVM with RBF kernel with different variances, k-nearest neighbor algorithm and Naive Bayes technique have been performed [7]. In these experiments, SVM with linear kernel was consistently the best performer. These results confirm the results of the previous studies by Yang and Liu [4], Joachims [1], and Forman [6]. Thus, in this study we have used SVM with linear kernel as the classification technique. For our experiments we used the SVM^{light} system, which is a rather efficient implementation by Joachims [9] and has been commonly used in previous studies [1,4,6].

Keyword selection can be implemented in two alternative ways. In the first one, which we name as *corpus-based keyword selection*, a common keyword set for all classes that reflects the most important words in all documents is selected. In the alternative approach, named as *class-based keyword selection*, the keyword selection process is performed separately for each class. In this way, the most important words specific to each class are determined. This technique has been implemented in some recent studies. One of these studies involves the categorization of internet documents [10]. A method for evaluating the importance of a term with respect to a class in the class hierarchy was proposed in that study. Another study is about clustering the documents [11]. Main focus of that paper is to increase the speed of the clustering algorithm. For this purpose, the authors have tried to make the method of extracting meaningful unit labels for document clusters much faster by using class-based keywords. In both studies, class-based keyword selection approach has been considered, but it was not compared with all words approach or with the corpus-based keyword selection approach.

In SVM-based text categorization, generally all available words in the document set are used instead of limiting to a set of keywords [1,4,7]. In some studies, it was stated that using all the words leads to the best performance and using keywords is unsuccessful with SVM [6,12]. An interesting study by Forman covers the keyword selection metrics for text classification using SVM [6]. While this study makes extensive use of class-based keywords, it naturally does not cover some of the important points. The main focus of the study is on the keyword selection metric; there does not exist a comparison of the class-based and corpus-based keyword selection approaches. Also, all the experiments were performed using boolean weighting algorithm and the study lacks a time complexity comparison between the results.

The aim of this paper is to evaluate the use of keywords for SVM-based text categorization. The previous studies focus on keyword selection metrics such as chi-square, information gain, tf-idf, odds ratio, probability ratio, document frequency, and bi-normal separation [6,13,14]. In this study we use tf-idf and, instead of the keyword selection metric, we focus on the comparison of the two keyword selection approaches, corpus-based keyword selection approach and class-based keyword selection approach. Unlike most studies, we also perform time complexity analysis. We aim to reach better results in less time and space complexity, which enables us to achieve good classification performance with limited machine capabilities and time. There are many situations in which only a small number of words are essential to classify the documents. Our research in this paper involves the inquiry of the optimal number of keywords for texts in text categorization.

The paper is organized as follows: Section 2 discusses the document representation and Section 3 gives an overview of the keyword selection approaches. In Section 4, we describe the standard Reuters-21578 dataset we have used in the experiments, our experimental methodology, evaluation metrics, and the results we have obtained. We conclude in Section 5.

2 Document Representation

Documents should first be transformed into a representation suitable for the classification algorithms to be applied. In our study, documents are represented by the widely used vector-space model, introduced by Salton *et al.* [15]. In this model, each document is represented as a vector **d**. Each dimension in the vector **d** stands for a distinct term in the term space of the document collection. We use the bag-of-words representation and define each term as a distinct word in the set of words of the document collection. To obtain the document vectors, each document is parsed, non-alphabetic characters and mark-up tags are discarded, case-folding is performed (i.e. all characters are converted to the same case-to lower case), and stopwords (i.e. words such as "an", "the", "they" that are very frequent and do not have discriminating power) are eliminated. We use the list of 571 stopwords used in the Smart system [15,16]. In order to define words that are in the same context with the same term and consequently to reduce dimensionality, we stem the words by using Porter's Stemming Algorithm [17], which is a commonly used algorithm for word stemming in English. We represent each document vector **d** as

$\mathbf{d}=(w_1, w_2, \ldots.., w_n)$

where w_i is the weight of i^{th} term of document **d**.

There are various term weighting approaches studied in the literature [18]. Boolean weighting and tf-idf (term frequency-inverted document frequency) weighting are two of the most commonly used ones.

In boolean weighting, the weight of a term is considered to be 1 if the term appears in the document and it is considered to be 0 if the term does not appear in the document:

$$w_i = \begin{cases} 1, & \text{if } tf_i > 0 \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where tf_i is the raw frequency of term i in document d.

tf-idf weighting scheme is defined as follows:

$$w_i = tf_i \cdot \log\left(\frac{n}{n_i}\right) \tag{2}$$

where tf_i is the same as above, n is the total number of documents in the document corpus and n_i is the number of documents in the corpus where term i appears. tf-idf weighting approach weights the frequency of a term in a document with a factor that discounts its importance if it appears in most of the documents, as in this case the term is assumed to have little discriminating power. Also, to account for documents of different lengths we normalize each document vector so that it is of unit length.

In his extensive study of feature selection metrics for SVM-based text classification, Forman used only boolean weighting [6]. However, the comparative study of different term weighting approaches in automatic text retrieval performed by Salton and Buckley reveals that the commonly used tf-idf weighting outperforms boolean weighting [18]. On the other hand, boolean weighting has the advantages of being very simple and requiring less memory. This is especially important in the high dimensional text domain. In the case of scarce memory resources, less memory requirement also leads to less classification time. Thus, in our study, we used both the boolean weighting and the tf-idf weighting schemes.

3 Keyword Selection

Most of the previous studies that apply SVM to text categorization use all the words in the document collection without any attempt to identify the important keywords [1,4]. On the other hand, there are various remarkable studies on keyword selection for text categorization in the literature [6,13,14]. As stated above, these studies mainly focus on keyword selection metrics and employ either the corpus-based or the class-based keyword selection approach, do not use standard datasets, and mostly lack a time complexity analysis of the proposed methods. In addition, most studies do not use SVM as the classification algorithm. For instance, Yang and Pedersen use kNN and LLSF [13], and Mladenic and Grobelnic use Naive Bayes in their studies on keyword selection metrics [14]. Later studies reveal that SVM performs consistently better than these classification algorithms [1,4,6].

In this study, we focus on the two keyword selection approaches, corpus-based keyword selection and class-based keyword selection. These two approaches have not been studied together in the literature. We also compare these keyword selection approaches with the alternative method of using all words without any keyword selection. Our focus is not on the keyword selection metric, thus we use the most commonly used tf-idf metric. In the corpus-based keyword selection approach, the terms that achieve the highest tf-idf score in the overall corpus are selected as the keywords. This approach favors the prevailing classes and gives penalty to classes with small number of training documents in document corpora where there is high skew. In the class-based keyword selection approach, on the other hand, distinct keywords are selected for each class. This approach gives equal weight to each class in the keyword selection phase. So, less prevailing classes are not penalized. This approach is also suitable for the SVM classifier as it solves two class problems.

4 Experiment Results

4.1 Document Data Set

In our experiments, we used the Reuters-21578 document collection, which is considered as the standard benchmark for automatic document categorization systems [19].

The documents in Reuters-21578 have been collected from Reuters newswire in 1987. This corpus consists of 21,578 documents. 135 different categories have been assigned to the documents. The maximum number of categories assigned to a document is 14 and the mean is 1.24. This dataset is highly skewed. For instance, the "earnings" category is assigned to 2,709 training documents, but 75 categories are assigned to less than 10 training documents. 21 categories are not assigned to any training documents. 7 categories contain only one training document and many categories overlap with each other such as "grain", "wheat", and "corn".

In order to divide the corpus into training and test sets, mostly the modified Apte (ModApte) split has been used [19]. With this split the training set consists of 9,603 documents and the test set consists of 3,299 documents. For our results to be comparable with the results of other studies, we also used this splitting method. We also removed the classes that do not exist both in the training set and in the test set, remaining with 90 classes out of 135. The total number of distinct terms in the corpus after preprocessing is 20,307. We report the results for the test set of this corpus.

4.2 Evaluation Metrics

To evaluate the performance of the keyword selection approaches we use the commonly used F-measure metric, which is equal to the harmonic mean of recall (ρ) and precision (π) [4]. ρ and π are defined as follows:

$$\pi_i = \frac{TP_i}{TP_i + FP_i}, \qquad \rho_i = \frac{TP_i}{TP_i + FN_i} \tag{3}$$

Here, TP_i (True Positives) is the number of documents assigned correctly to class i; FP_i (False Positives) is the number of documents that do not belong to class i but are assigned to class i incorrectly by the classifier; and FN_i (False Negatives) is the number of documents that are not assigned to class i by the classifier but which actually belong to class i.

The F-measure values are in the interval (0,1) and larger F-measure values correspond to higher classification quality. The overall F-measure score of the entire classification problem can be computed by two different types of average, *micro-average* and *macro-average* [4].

Micro-averaged F-Measure. In micro-averaging, F-measure is computed globally over all category decisions. ρ and π are obtained by summing over all individual decisions:

$$\pi = \frac{TP}{TP + FP} = \frac{\sum_{i=1}^{M} TP_i}{\sum_{i=1}^{M}(TP_i + FP_i)}, \quad \rho = \frac{TP}{TP + FN} = \frac{\sum_{i=1}^{M} TP_i}{\sum_{i=1}^{M}(TP_i + FN_i)} \tag{4}$$

where M is the number of categories. Micro-averaged F-measure is then computed as:

$$F(\text{micro-averaged}) = \frac{2\pi\rho}{\pi + \rho} \tag{5}$$

Micro-averaged F-measure gives equal weight to each document and is therefore considered as an average over all the document/category pairs. It tends to be dominated by the classifier's performance on common categories.

Macro-averaged F-Measure. In macro-averaging, F-measure is computed locally over each category first and then the average over all categories is taken. π and ρ are computed for each category as in Equation 3. Then F-measure for each category i is computed and the macro-averaged F-measure is obtained by taking the average of F-measure values for each category as:

$$F_i = \frac{2\pi_i\rho_i}{\pi_i + \rho_i}, \quad F(\text{macro-averaged}) = \frac{\sum_{i=1}^{M} F_i}{M} \tag{6}$$

where M is total number of categories. Macro-averaged F-measure gives equal weight to each category, regardless of its frequency. It is influenced more by the classifier's performance on rare categories. We provide both measurement scores to be more informative.

4.3 Results and Discussion

Tables 1 and 2 display the micro-averaged and macro-averaged F-measure results for boolean and tf-idf document representations for all words and for keywords ranging in number from 10 to 2000, respectively. From Table 1, we can conclude that class-based keyword selection achieves higher micro-averaged F-measure performance than corpus-based approach for small number of keywords. In text categorization, most of the learning takes place with a small but crucial portion of keywords for a class [2]. Class-based keyword selection, by definition, focuses on this small portion; on the other hand, corpus-based approach finds general keywords concerning all classes. So, with few keywords, class-based approach achieves much more success by finding more crucial class keywords. Corpus-based approach is not successful with that small portion, but has a steeper learning curve that reaches the peak value of our study (86.1%) with 2000 corpus-based keywords, which exceeds the success scores of recent studies with standard usage of Reuters-21578 [4,5].

Boolean class-based approach performs always worse than tf-idf class-based approach for all number of keywords. This is an expected result, previous studies show parallel results with boolean approach [18].

Table 1. Micro-averaged F-measure results

# of keywords	Boolean (class-based)	tf-idf (corpus-based)	tf-idf (class-based)
10	0,738	0,425	0,780
30	0,780	0,543	0,814
50	0,802	0,628	0,831
70	0,802	0,671	0,833
100	0,806	0,697	0,838
200	0,811	0,761	0,838
300	0,819	0,786	0,839
400	0,823	0,804	0,842
500	0,821	0,813	0,848
1000	0,820	0,845	0,854
1200	0,818	0,850	0,855
1500	0,818	0,859	0,853
2000	0,818	0,861	0,855
All words	0,817	0,857	0,857

Table 2. Macro-averaged F-measure results

# of keywords	Boolean (class-based)	tf-idf (corpus-based)	tf-idf (class-based)
10	0,481	0,010	0,500
30	0,469	0,030	0,515
50	0,472	0,051	0,519
70	0,466	0,082	0,510
100	0,443	0,091	0,508
200	0,398	0,162	0,511
300	0,384	0,207	0,492
400	0,385	0,242	0,494
500	0,377	0,263	0,494
1000	0,349	0,373	0,498
1200	0,345	0,388	0,494
1500	0,332	0,425	0,492
2000	0,328	0,431	0,492
All words	0,294	0,439	0,439

Table 3. Classification time in seconds

# of keywords	Boolean (class-based)	tf-idf (class-based)
10	5	3
30	5	5
50	7	7
70	9	6
100	10	14
200	11	14
300	17	17
400	18	22
500	20	31
1000	25	40
1200	27	41
1500	31	42
2000	35	44
All words	43	66

From Table 2, we can conclude that class-based keyword selection achieves consistently higher macro-averaged F-measure performance than corpus-based approach. The high skew in the distribution of the classes in the dataset affects the macro-averaged F-measure values in a negative way because macro-average gives equal weight to each class instead of each document and documents of rare classes tend to be more misclassified. By this way, the average of correct classifications of classes drops dramatically for datasets having many rare classes. Class-based keyword selection is observed to be very useful for this skewness. As stated above, with even a small portion of words (50-100-200), class-based tf-idf method reaches 50% success which is far better than the 43.9% success of tf-idf with all words. Rare classes are characterized in a successful way with class-based keyword selection, because every class has its own keywords for the categorization problem. Corpus-based approach shows worse results because most of the keywords are selected from prevailing classes which prevents rare classes to be represented fairly by their keywords.

Table 3 shows the classification times for class-based boolean and class-based tf-idf approaches. We do not display the results for the corpus-based tf-idf approach as its time-complexity is similar to that of the class-based tf-idf approach. We observe that when we use a small number of keywords in the class-based tf-idf approach we gain a lot from time without losing much from performance. For instance, when we use 70 keywords, the classification phase is 10 times faster than the classification phase in the case where all words are used. In addition, the macro-averaged F-measure performance for 70 keywords is better than the case where all words are used and the micro-averaged F-measure performance is not much worse. Another observation is that time complexity of boolean class-based approach is better than tf-idf class-based approach. This is an expected result because boolean approach consumes less space and performs less operations than tf-idf approach. In situations where we have limited time and space resources, we may sacrifice from performance by using class-based boolean approach, which gives around 82% success rate and can be deemed as satisfying.

5 Conclusion

In this paper we investigate the use of keywords in text categorization with SVM. Unlike the previous studies that focus on keyword selection metrics, we study the performance of the two approaches for keyword selection, corpus-based approach and class-based approach. We use the standard Reuters-21578 dataset and both boolean and tf-idf weighting schemes. We analyze the approaches in terms of micro-averaged F-measure, macro-averaged F-measure and classification time.

Generally all of the words in the documents were used for categorization with SVM. Keyword selection was not performed in most of the studies; even, in some studies, keyword selection was stated to be unsuccessful with SVM [6,12]. In contrast to these studies we reveal that keyword selection improves the performance of SVM both in terms of F-measure and time. For instance, corpus-based approach with 2000 keywords performs the best in much less time than the case where all words are used. In the corpus-based approach the keywords tend to be selected from the prevailing classes. Rare classes are not represented well by these keywords. However, in the class-based approach, rare classes are represented equally well as the prevailing classes because each class is represented with its own keywords for the categorization problem. Thus, the class-cased tf-idf approach with small number of keywords (50-100) achieves consistently higher macro-averaged F-measure performance than both the corpus-based approach and the approach where all the words are used. It also achieves higher micro-averaged F-measure performance than corpus-based approach when a small number of keywords is used. This is important as there is a lot of gain from classification time when small number of keywords is used.

When we compare the tf-idf and boolean weighting approaches we see that class-based tf-idf approach is more successful than class-based boolean approach. However, in situations where we have limited time and space resources, we may sacrifice from performance by using class-based boolean approach, which gives around 82% success rate and can be deemed as satisfying.

Acknowledgment

This work has been supported by the Boğaziçi University Research Fund under the grant number 05A103.

References

1. Joachims, T.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features. European Conference on Machine Learning (ECML) (1998)

2. Özgür, L., Güngör, T., Gürgen, F.: Adaptive Anti-Spam Filtering for Agglutinative Languages. A Special Case for Turkish, Pattern Recognition Letters, **25** no.16 (2004) 1819–1831
3. McCallum, A., Nigam, K.: A Comparison of Event Models for Nave Bayes Text Classification. Sahami, M. (Ed.), Proc. of AAAI Workshop on Learning for Text Categorization (1998), Madison, WI, 41–48
4. Yang, Y., Liu, X.: A Re-examination of Text Categorization Methods. In Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval, Berkeley, US (1996)
5. Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys **34** no. 5 (2002) 1–47
6. Forman, G.: An Extensive Empirical Study of Feature Selection Metrics for Text Classification. Journal of Machine Learning Research **3** (2003) 1289–1305
7. Özgür, A.: Supervised and Unsupervised Machine Learning Techniques for Text Document Categorization. Master's Thesis (2004), Bogazici University, Turkey
8. Burges, C. J. C.: A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery Vol. 2 No. 2 (1998) 121–167
9. Joachims, T.: Advances in Kernel Methods-Support Vector Learning. chapter Making Large-Scale SVM Learning Practical MIT-Press (1999)
10. Lin, S-H., Shih C-S., Chen, M. C., Ho, J-M.: Extracting Classification Knowledge of Internet Documents with Mining Term Associations: A Semantic Approach. In Proc. of ACM/SIGIR (1998), Melbourne, Australia 241–249
11. Azcarraga, A. P., Yap, T., Chua, T. S.: Comparing Keyword Extraction Techniques for Websom Text Archives. International Journal of Artificial Intelligence Tools **11** no. 2 (2002)
12. Aizawa, A.: Linguistic Techniques to Improve the Performance of Automatic Text Categorization. In Proceedings of 6th Natural Language Processing Pacific Rim Symposium (2001), Tokyo, JP 307–314
13. Yang, Y., Pedersen J. O.: A Comparative Study on Feature Selection in Text Categorization. In Proceedings of the 14th International Conference on Machine Learning (1997) 412–420
14. Mladenic, D., Grobelnic, M.: Feature Selection for Unbalanced Class Distribution and Naive Bayes. In Proceedings of the 16th International Conference on Machine Learning (1999) 258–267
15. Salton, G., Yang, C., Wong, A.: A Vector-Space Model for Automatic Indexing. Communications of the ACM **18** no.11 (1975) 613–620
16. ftp://ftp.cs.cornell.edu/pub/smart/ (2004)
17. Porter, M. F.: An Algorithm for Suffix Stripping. Program **14** (1980) 130–137
18. Salton, G., Buckley, C.: Term Weighting Approaches in Automatic Text Retrieval. Information Processing and Management **24** no. 5 (1988) 513–523
19. Lewis, D. D.: Reuters-21578 Document Corpus V1.0.
http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html