Classification of Skewed and Homogenous Document Corpora with Class-Based and Corpus-Based Keywords

Arzucan Özgür[1] and Tunga Güngör[1]

[1] Boğaziçi University, Computer Engineering Department, Bebek,
34342 İstanbul, Turkey
{ozgurarz, gungort}@boun.edu.tr

Abstract. In this paper, we examine the performance of the two policies for keyword selection over standard document corpora of varying properties. While in corpus-based policy a single set of keywords is selected for all classes globally, in class-based policy a distinct set of keywords is selected for each class locally. We use SVM as the learning method and perform experiments with boolean and tf-idf weighting. In contrast to the common belief, we show that using keywords instead of all words generally yields better performance and tf-idf weighting does not always outperform boolean weighting. Our results reveal that corpus-based approach performs better for large number of keywords while class-based approach performs better for small number of keywords. In skewed datasets, class-based keyword selection performs consistently better than corpus-based approach in terms of macro-averaged F-measure. In homogenous datasets, performances of class-based and corpus-based approaches are similar except for small number of keywords.

1 Introduction

The amount of electronic text information available such as Web pages, digital libraries, and email messages is increasing rapidly. As a result, the challenge of extracting relevant knowledge increases as well. The need for tools that enable people find, filter, and manage these resources has grown. Thus, automatic categorization of text document collections has become an important research issue.

SVM is one of the most successful text categorization methods [1, 2, 3]. It was designed for solving two-class pattern recognition problems [4]. 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 used to learn linear or non-linear decision functions. 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 [5]. In these experiments, SVM with linear kernel was consistently the best performer. These results confirm the results of previous studies [1, 2, 3]. Thus, in this study we use SVM with linear kernel as the classification technique. For our experiments, we use the *SVM*$_{light}$ system [6], which has been commonly used in previous studies [1, 2, 3].

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 and a different set of keywords is used for each class.

Most previous studies focus on keyword selection metrics such as chi-square, information gain, odds ratio, probability ratio, document frequency, and binormal separation [3, 7, 8]. They use either the class-based or the corpus-based 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, 2, 5, 9]. In some studies, it was stated that using all the words leads to the best performance and using keywords is unsuccessful with SVM [3, 9, 10]. An interesting study by Forman covers the keyword selection metrics for text classification using SVM [3]. 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 metrics; and there does not exist a comparison of the class-based and corpus-based keyword selection approaches. In [9], Debole and Sebastiani focus on supervised term weighting approaches and report their results both for class-based keyword selection, which they name as local policy and corpus-based keyword selection, which they call global policy. They use Reuters-21578 in their study, which is a highly skewed corpus. Different from our findings, they report that global keyword selection performs better than local keyword selection and SVM performs best when all the words are used. In [11], Özgür *et al.*, compare class-based and corpus-based keyword selection. However, they use a single dataset, Reuters-21578, and do not study the effect of these keyword selection approaches for document corpora of varying class distributions.

The aim of this paper is to evaluate the use of keywords for SVM-based text categorization and examine how class-based and corpus-based keyword selection approaches perform for datasets with varying class distribution properties. We use six standard document corpora in our study. Classic3 is a homogenous corpus, where all the classes are nearly equally well represented in the training set. Reuters-21578 and Wap corpora are highly skewed. A few of the classes are prevalent in the training set, while some classes are represented with very few documents. Hitech, LA1, and Reviews are neither homogenous nor highly skewed. Our results reveal that using keywords in SVM-based text categorization instead of using all the available words generally leads to better performance. We show that when corpus-based keyword selection is used for highly skewed datasets, less prevalent classes are represented poorly and macro-averaged F-measure performance drops down. In this case, class-based keyword selection is preferable. In homogenous datasets, although class-based approach performs better for small number of keywords, corpus-based approach performs slightly better or similar for large number of keywords. We perform our experiments with the two most commonly used term weighting approaches, boolean and tf-idf weighting. Surprisingly, we find that tf-idf weighting does not always outperform boolean weighting. As the keyword selection metric, we use total tf-idf scores of each term. In this way, keyword selection and term weighting phases are

reduced to a single phase since tf-idf is also used for term weighting. This reduces the overall time of term weighting and keyword selection.

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 six standard datasets we used in the experiments, our experimental methodology, and the results we have obtained. We conclude in Section 5.

2 Document Representation

In our study, documents are represented by the vector-space model. In this model, each document is represented as a vector **d**, where each dimension stands for a distinct term in the term space of the document collection. We use the bag-of-words representation. To obtain the document vectors, each document is parsed, non-alphabetic characters and mark-up tags are discarded, case-folding is performed, and stop words are eliminated. We use the list of 571 stop words used in the Smart system [12]. We stem the words by using Porter's Stemming Algorithm [13], which is commonly used for word stemming in English. Each document is represented as $\mathbf{d}=(w_1,w_2,\ldots,w_n)$, where, w_i is the weight of i^{th} term of document d.

We use boolean and tf-idf weighting schemes which are most commonly used in the literature. 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. tf-idf weighting scheme is defined as follows:

$$w_i = tf_i \cdot \log\left(\frac{n}{n_i}\right) . \quad (\mathbf{1})$$

where tf_i is the raw frequency of term i in document d, n is the total number of documents in the 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, in order to account for documents of different lengths we normalize each document vector so that it is of unit length. Previous studies report that tf-idf weighting performs better than boolean weighting [14]. 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. Interestingly, we found that boolean approach does not always perform worse than tf-idf approach.

3 Keyword Selection

Most 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, 2, 9]. On the other hand, there are various remarkable studies on keyword selection for text categorization in the literature [3, 7, 8]. As stated above, these studies mainly focus on keyword selection metrics and employ either the corpus-based or the class-based keyword selection approach, and do not use standard datasets. In addition, most studies do not use SVM as the classification algorithm. For instance, in [7] kNN and LLSF are used, and in [8] Naive Bayes is used. Later studies reveal that SVM performs consistently better than these classification algorithms [1, 2, 3].

In this study, rather than focusing on keyword selection metrics, we focus on the two keyword selection approaches, corpus-based keyword selection and class-based keyword selection. These two approaches have not been studied extensively together in the literature. In [9], Debole and Sebastiani perform experiments for both of the approaches. However their study is not extensive in this aspect since their main focus is on supervised term weighting methods and they use only the Reuters-21578 dataset. In contrast to our findings, they report that corpus-based keyword selection performs better than class-based keyword selection and SVM performs best when all the words are used. In [11], Özgür *et al.*, compare class-based and corpus-based keyword selection. However, they use a single dataset, Reuters-21578, and do not study the effect of these keyword selection approaches for document corpora of varying class distributions. In this study, we compare these keyword selection approaches with the alternative method of using all words without any keyword selection. We evaluate the performance of these approaches over datasets with varying class size distributions, i.e. homogenous, skewed, and highly skewed.

We use total tf-idf scores of terms as the keyword selection metric. Although it has not been used as a keyword selection metric in the literature, it has the advantage of leading to the reduction of keyword selection and term weighting phases into a single phase, when tf-idf is also used for term weighting. Our results show that it performs well, since in contrast to the previous studies we could obtain performances better than the approach where all the available words are used with SVM-based text categorization. In corpus-based keyword selection approach, terms that achieve the highest total tf-idf score in the overall corpus are selected as the keywords. To obtain the total tf-idf score of a term, the tf-idf weights of that term in each document are summed. 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. The total tf-idf score of a term is calculated separately for each class. To obtain the total tf-idf score of a term for a specific class, the tf-idf weights of that term in only the documents that belong to that class are summed. This approach gives equal weight to each class in the keyword selection phase. So, less prevailing classes are not penalized.

4 Experiment Results

4.1 Document Data Sets

In our experiments we used six standard document corpora, widely used in automatic text organization research. The contents of these document sets, after preprocessing as described in Section 2, is summarized in Table 1. Classic3 data set contains 1,398 CRANFIELD documents from aeronautical system papers, 1,033 MEDLINE documents from medical journals, and 1,460 CISI documents from information retrieval papers. This dataset is homogenous since all the classes are represented equally well in the training set. This data set is relatively easy, because the classes are disjoint from each other.

Table 1. Summary description of document sets

Data set	# of documents	# of classes	# of terms
Classic3	3,891	3	10,930
Hitech	2,300	6	18,867
LA1	3,204	6	25,024
Reviews	4,069	5	31,325
Reuters-21578	12,902	90	20,307
Wap	1,560	20	8,064

The Hitech, LA1, and Reviews [15] datasets are neither highly skewed nor homogenous. They are very high dimensional compared to the number of documents in the training sets. The Hitech data set was derived from the San Jose Mercury newspaper articles, which are delivered as part of the TREC collection [16]. The classes of this document corpora are computers, electronics, health, medical, research, and technology. LA1 data set consists of documents from Los Angeles Times newspaper, used in TREC-5 [16]. The categories correspond to the desk of the paper that each article appeared. The data set consists of documents from entertainment, financial, foreign, metro, national, and sports desks. Reviews data set contains articles from San Jose Mercury Newspaper, that are distributed as part of the TREC collection TIPSTER vol. 3 [16]. The classes of this document corpora are food, movie, music, radio, and restaurant.

The documents in Reuters-21578 v1.0 document collection [17], which is considered as the standard benchmark for automatic document organization systems, 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.

Wap data set consists of 1,560 web pages from Yahoo! subject hierarchy collected and classified into 20 different classes for the WebACE project [18]. This dataset is also highly skewed. Minimum class size is 5, maximum class size is 341, and average class size is 78. Many categories of Wap are close to each other.

In order to divide the Reuters-21578 corpus into training and test sets, mostly the modified Apte (ModApte) split has been used [17]. 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. For the other data sets, we used the initial 2/3 of the documents as the training set and the remaining 1/3 as the test set. Below we report the results for the test sets of the corpora.

4.2 Results and Discussion

Tables 2 and 3 display, respectively, the micro-averaged and macro-averaged F-measure results, for boolean and tf-idf document representations using all words and using keywords ranging in number from 10 to 2000. Bool (cl), tf-idf (cl), and tf-idf (co) stand for class-based approach with boolean weighting, class-based approach with tf-idf weighting, and corpus-based approach with tf-idf weighting, respectively. Micro-averaged F-measure gives equal weight to each document and therefore it tends to be dominated by the classifier's performance on common categories. Macro-averaged F-measure gives equal weight to each category regardless of its frequency and thus it is influenced more by the classifier's performance on rare categories.

In the following discussion, it is assumed that tf-idf weighting is used unless it is stated otherwise. When we examine Classic3 dataset, whose class distribution is homogenous, we observe that micro-averaged and macro-averaged F-measure results are similar. Also, there is not much performance difference among class-based keyword selection and corpus-based keyword selection. For instance, in the case of 30 keywords, both achieve 90% success in terms of micro-averaged F-measure and 88.6% success in terms of macro-averaged F-measure. However, class-based approach converges faster than corpus-based approach and thus performs better for small number of keywords (200 keywords and less). As number of keywords increases performance tends to increase. Although all words approach (10930 words) achieves the highest performance of 99.4%, tf-idf corpus-based approach achieves a very close performance of 99.2% with 1500 keywords. Boolean class-based approach does not perform much worse than the tf-idf class-based approach and it performs generally better than tf-idf corpus-based approach for 100 and less keywords.

Hitech, LA1, and Reviews datasets have neither homogenous nor highly skewed class distributions. Micro-averaged and macro-averaged F-measure results of Reviews dataset are similar to each other. However, macro-averaged F-measure results are considerably less than micro-averaged F-measure results for Hitech and LA1 datasets. When we examine the results on the Hitech dataset, we observe that for 300 and less

Table 2. Micro-averaged F-measure Results

	# words	10	30	50	70	100	200	300	400	500	1000	1500	2000	All
Classic3	bool (cl)	0.824	0.892	0.930	0.937	0.936	0.940	0.942	0.947	0.953	0.955	0.961	0.965	0.981
	tf-idf (co)	0.768	0.900	0.901	0.926	0.937	0.956	0.963	0.971	0.981	0.988	0.992	0.992	0.994
	tf-idf (cl)	0.845	0.900	0.944	0.948	0.950	0.959	0.955	0.957	0.960	0.964	0.965	0.971	0.994
Hitech	bool (cl)	0.543	0.534	0.533	0.551	0.567	0.593	0.587	0.578	0.569	0.585	0.578	0.586	0.581
	tf-idf (co)	0.377	0.518	0.539	0.586	0.603	0.606	0.614	0.627	0.623	0.643	0.647	0.659	0.649
	tf-idf (cl)	0.553	0.597	0.619	0.628	0.629	0.625	0.631	0.622	0.644	0.621	0.618	0.627	0.649
LA1	bool (cl)	0.598	0.685	0.699	0.718	0.730	0.739	0.759	0.764	0.783	0.790	0.791	0.791	0.797
	tf-idf (co)	0.467	0.648	0.723	0.754	0.766	0.793	0.806	0.816	0.816	0.817	0.824	0.833	0.841
	tf-idf (cl)	0.634	0.731	0.761	0.773	0.784	0.789	0.801	0.809	0.807	0.814	0.812	0.815	0.841
Reviews	bool (cl)	0.800	0.834	0.844	0.846	0.858	0.882	0.891	0.893	0.898	0.900	0.903	0.903	0.915
	tf-idf (co)	0.778	0.862	0.869	0.886	0.894	0.934	0.939	0.942	0.944	0.943	0.937	0.936	0.941
	tf-idf (cl)	0.843	0.867	0.891	0.901	0.901	0.906	0.912	0.914	0.918	0.926	0.924	0.920	0.941
Reuters 21578	bool (cl)	0.738	0.780	0.802	0.802	0.806	0.811	0.819	0.823	0.821	0.820	0.818	0.818	0.817
	tf-idf (co)	0.425	0.543	0.628	0.671	0.697	0.761	0.786	0.804	0.813	0.845	0.859	0.861	0.857
	tf-idf (cl)	0.780	0.814	0.831	0.833	0.838	0.838	0.839	0.842	0.848	0.854	0.853	0.855	0,857
Wap	bool (cl)	0.650	0.674	0.713	0.715	0.713	0.721	0.736	0.757	0.754	0.758	0.766	0.762	0.759
	tf-idf (co)	0.135	0.496	0.585	0.607	0.655	0.691	0.716	0.723	0.721	0.740	0.749	0.743	0.752
	tf-idf (cl)	0.688	0.736	0.748	0.746	0.751	0.736	0.728	0.726	0.722	0.746	0.741	0.747	0.752

Table 3. Macro-averaged F-measure Results

	# words	10	30	50	70	100	200	300	400	500	1000	1500	2000	All
Classic3	bool (cl)	0.811	0.880	0.930	0.938	0.935	0.938	0.939	0.946	0.951	0.953	0.959	0.965	0.980
	tf-idf (co)	0.769	0.886	0.899	0.925	0.936	0.953	0.962	0.971	0.980	0.989	0.992	0.992	0.994
	tf-idf (cl)	0.829	0.886	0.941	0.946	0.949	0.957	0.955	0.956	0.959	0.964	0.964	0.970	0.994
Hitech	bool (cl)	0.397	0.413	0.421	0.434	0.453	0.469	0.472	0.465	0.446	0.451	0.461	0.456	0.433
	tf-idf (co)	0.230	0.371	0.466	0.497	0.507	0.505	0.508	0.538	0.530	0.538	0.582	0.598	0.558
	tf-idf (cl)	0.489	0.555	0.577	0.571	0.574	0.565	0.565	0.570	0.589	0.567	0.549	0.561	0.558
LA1	bool (cl)	0.476	0.590	0.620	0.656	0.662	0.670	0.686	0.696	0.713	0.729	0.726	0.728	0.729
	tf-idf (co)	0.284	0.530	0.628	0.674	0.692	0.715	0.738	0.745	0.752	0.748	0.752	0.765	0.777
	tf-idf (cl)	0.552	0.674	0.706	0.712	0.727	0.735	0.745	0.760	0.755	0.762	0.756	0.764	0.777
Reviews	bool (cl)	0.767	0.814	0.829	0.830	0.843	0.871	0.881	0.879	0.881	0.885	0.884	0.885	0.874
	tf-idf (co)	0.558	0.692	0.693	0.710	0.720	0.931	0.928	0.933	0.939	0.939	0.935	0.932	0.928
	tf-idf (cl)	0.847	0.864	0.894	0.903	0.904	0.904	0.911	0.912	0.916	0.916	0.912	0.906	0.928
Reuters 21578	bool (cl)	0.481	0.469	0.472	0.466	0.443	0.398	0.384	0.385	0.377	0.349	0.332	0.328	0.294
	tf-idf (co)	0.010	0.030	0.051	0.082	0.091	0.162	0.207	0.242	0.263	0.373	0.425	0.431	0.439
	tf-idf (cl)	0.500	0.515	0.519	0.510	0.508	0.511	0.492	0.494	0.494	0.498	0.492	0.492	0.439
Wap	bool (cl)	0.442	0.453	0.482	0.502	0.502	0.480	0.476	0.490	0.488	0.488	0.482	0.477	0.448
	tf-idf (co)	0.092	0.208	0.306	0.321	0.350	0.412	0.416	0.435	0.442	0.455	0.468	0.455	0.450
	tf-idf (cl)	0.550	0.593	0.590	0.550	0.533	0.507	0.505	0.497	0.495	0.509	0.477	0.482	0.450

keywords class-based approach achieves better micro-averaged F-measure performance than corpus-based approach and for 1000 and less keywords it achieves better macro-averaged F-measure performance. On the other hand, corpus-based approach achieves the highest performance for 2000 keywords, i.e. 65.9% micro-averaged and 59.8% macro-averaged F-measure performance. These results are higher than the all words approach (18867 words), which achieves 64.9% and 55.8% micro-averaged and macro-averaged F-measure results, respectively. In terms of macro-averaged F-measure performance, class-based approach with 50 and more keywords and corpus-based approach with 1500 and 2000 keywords achieve better results than the all words approach. Boolean class-based approach with 200 keywords achieves higher F-measure performance than boolean all words approach. Although boolean class-based approach performs worse than tf-idf class-based approach, it performs better than tf-idf corpus-based approach for 10 and 30 keywords.

Over LA1 dataset, class-based approach performs better than corpus-based approach for 100 and less keywords in terms of micro-averaged F-measure. Macro-averaged F-measure results of class-based approach are generally higher than that of the corpus-based approach. Only for 2000 keywords, corpus-based approach achieves slightly better macro-averaged F-measure performance than class-based approach (76.5% versus 76.4%). All words approach achieves the best performance of 84.1% micro-averaged and 77.7% macro-averaged F-measure. The closest performance to these results is achieved by the corpus-based approach with 2000 keywords, 83.3% micro-averaged and 76.5% macro-averaged F-measure. Boolean class-based approach performs worse than tf-idf class-based approach, but it performs better than tf-idf corpus-based approach for 10 and 30 keywords.

Over Reviews dataset, tf-idf corpus-based approach achieves the highest micro-averaged (94.4%) and macro-averaged (93.9%) F-measure performance with 500 keywords. These results are even higher than the all words approach (31325 words), which achieves 94.1% micro-averaged and 92.8% macro-averaged F-measure performance. For 100 and less keywords class-based approach achieves higher performance than corpus-based approach both in terms of micro-averaged and macro-averaged F-measure. There is a gap between macro-averaged F-measure results. For instance, while class-based approach achieves 90.3% macro-averaged performance for 70 keywords, corpus-based approach achieves only 71.0% performance. Even boolean class-based approach performs better than tf-idf corpus-based approach in terms of macro-averaged F-measure for 100 and less keywords.

Reuters-21578 and Wap datasets have highly skewed class distributions. Thus, there is a large gap between micro-averaged and macro-averaged F-measure results. For both datasets, 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 datasets 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. For instance, in Reuters-21578 dataset, 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. In Wap dataset, class-based approach with 30 keywords achieves the highest performance in terms of macro-averaged F-measure (59.3%), which is considerably higher than the macro-averaged F-measure performance of all words approach (45.0%). Also, tf-idf class based approach for small number of keywords (100 keywords and less) achieves better or similar performance compared to the case where all words are used. 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. In text categorization, most of the learning takes place with a small but crucial portion of keywords for a class [19]. 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. For instance, for the Reuters-21578 dataset, it leads to the peak micro-averaged F-measure 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 [1, 20].

Boolean class-based approach generally performs worse than tf-idf class-based approach for all number of keywords. This is an expected result, since it does not take into account term frequencies and inverse document frequencies. However, surprisingly, for Wap dataset, for 300 and more keywords, boolean approach achieves higher micro-averaged F-measure performance than tf-idf class-based and corpus-based approaches. Also, boolean all words approach performs better than tf-idf all words approach in terms of micro-averaged F-measure and performs similar in terms of macro-averaged F-measure. In addition, boolean approach achieves the highest micro-averaged F-measure performance in the overall for 2000 keywords (76.2%). Thus, in this case boolean approach may be preferred to tf-idf approach since it is simpler and needs less memory and time.

5 Conclusion

In this paper, we investigated the use of keywords in text categorization with SVM. Unlike previous studies that focus on keyword selection metrics, we studied the performance of the two approaches for keyword selection, corpus-based approach and class-based approach, over datasets of varying class distribution properties. We used six standard document corpora and both boolean and tf-idf weighting schemes.

In text categorization literature, 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 [3, 9, 10]. In contrast to these studies, we observed that keyword selection generally improves the performance of SVM. This is quite important since there is considerable gain in terms of classification time and memory when small number of keywords is used.

For all datasets (homogenous, skewed, and highly skewed) class-based approach performs better than corpus-based approach for small number of keywords (generally 100 and less keywords) in terms of micro-averaged F-measure. Corpus-based approach generally achieves higher micro-averaged F-measure performance for larger number of keywords. There is not much difference between micro-averaged and macro-averaged F-measure values and between class-based and corpus-based approaches in homogenous datasets. On the other hand, for skewed and highly skewed datasets, there is a gap between micro-averaged and macro-averaged F-measure results. In highly skewed datasets, class-based keyword selection approach performs consistently better than corpus-based approach and the approach where all words are used, in terms of macro-averaged F-measure. 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. Therefore, class-based keyword selection approach should be preferred to corpus-based approach for highly skewed datasets. It should also be preferred when small number of keywords will be used due to space and time limitations.

When we compare the tf-idf and boolean weighting approaches, surprisingly we see that boolean approach is not always worse than tf-idf approach although it is simpler. It can be preferred to tf-idf approach especially in cases where there are limited space resources.

Acknowledgement

This work was supported by the Boğaziçi University Research Fund under the grant number 05A103. The authors would like to thank Levent Özgür for helpful discussions.

References

1. 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 (1996)
2. Joachims, T.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In: European Conference on Machine Learning (ECML) (1998)
3. Forman, G.: An Extensive Empirical Study of Feature Selection Metrics for Text Classification. Journal of Machine Learning Research 3 (2003) 1289–1305
4. Burges, C.J.C.: A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery 2(2) (1998) 121–167
5. Özgür, A.: Supervised and Unsupervised Machine Learning Techniques for Text Document Categorization. MS Thesis, Boğaziçi University, Istanbul (2004)
6. Joachims, T.: Advances in Kernel Methods-Support Vector Learning. Chapter Making Large-Scale SVM Learning Practical. MIT (1999)

7. 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
8. 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
9. Debole, F., Sebastiani, F.: Supervised Term Weighting for Automated Text Categorization. In: Proceedings of SAC-03, 18th ACM Symposium on Applied Computing. ACM Press (2003) 784–788
10. Aizawa, A.: Linguistic Techniques to Improve the Performance of Automatic Text Categorization. In: Proceedings of 6th Natural Language Processing Pacific Rim Symposium. Tokyo (2001) 307–314
11. Özgür, A., Özgür, L., Güngör T.:Text Categorization with Class-Based and Corpus-Based Keyword Selection. In: Proceedings of ISCIS'05. Lecture Notes in Computer Science 3733. Springer Verlag (2005) 607–616
12. ftp://ftp.cs.cornell.edu/pub/smart/ (2004)
13. Porter, M. F.: An Algorithm for Suffix Stripping. Program 14 (1980) 130–137
14. Salton, G., Buckley, C.: Term Weighting Approaches in Automatic Text Retrieval. Information Processing and Management 24(5) (1988) 513–523
15. Karypis G.: Cluto 2.0 Clustering Toolkit. http://www.users.cs.umn.edu/~karypis/cluto (2004)
16. TREC. Text Retrieval Conference. http://trec.nist.gov (1999)
17. Lewis, D.D.: Reuters-21578 Document Corpus V1.0. http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html
18. Han, E-H.S., Boley, D., Gini, M., Gross, R., Hastings, K., Karypis, G., Kumar, V., Mobasher, B., Moore, J.: WebAce: A Web Agent for Document Categorization and Exploration. In: Proceedings of the 2nd International Conference on Autonomous Agents (1998)
19. Özgür, L., Güngör, T., Gürgen, F.: Adaptive Anti-Spam Filtering for Agglutinative Languages. A Special Case for Turkish. Pattern Recognition Letters 25(16) (2004) 1819–1831
20. Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys 34(5) (2002) 1–47