Two-Stage Feature Selection for Text Classification

Levent Özgür, Tunga Güngör

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

Abstract. In this paper, we focus on feature coverage policies used for feature selection in the text classification domain. Two alternative policies are discussed and compared: corpus-based and class-based selection of features. We make a detailed analysis of pruning and keyword selection by varying the parameters of the policies and obtain the optimal usage patterns. In addition, by combining the optimal forms of these methods, we propose a novel two-stage feature selection approach. The experiments on three independent datasets showed that the proposed method results in a statistically significant increase over the traditional methods in the success rates of the classifier.

1 Introduction

Text classification, which is a sub-domain of classification and has been subject to active research for many years, is a learning task where pre-defined category labels are assigned to documents based on the likelihood suggested by a training set of labeled documents.

There exist several research topics related to text classification that have been extensively studied in the literature, such as the machine learning scheme used for classification, feature representation, generating new feature types (syntactic or semantic features), feature selection, performance measures, etc. In this paper, by using the well-known and the state-of-the-art methods in most of these topics, we mainly focus on the feature selection and feature filtering process and propose a novel two-stage feature extraction approach. Basically feature selection aims at eliminating unimportant and uninformative features using some statistical ranking techniques in order to reach more scalable and accurate solutions.

In traditional studies, all available words in the document set were used as features instead of limiting to a set of keywords and satisfactory results were obtained [8]. Some studies even stated that using all the words leads to the best performance and using keywords may be unsuccessful without optimal parameter tuning [1,3]. On the other hand, some studies reveal that feature selection may improve the performance in terms of accuracy and scalability with a significant cut in the solution vector size [4]. There are different types of feature selection implementations: Filter methods determine a ranking among all features with respect to some statistical metrics, wrapper methods use classical artificial intelligence techniques (e.g. greedy hill climbing) with cross validation, and embedded methods employ a linear prediction

model for optimization [4]. Among them, filter methods are the simplest (in terms of implementation) and the most scalable ones for text classification problems having large feature spaces.

There are various types of feature selection metrics used in the text classification domain, such as chi-square, information gain, tf-idf, odds ratio, pruning, probability ratio, document frequency, and bi-normal separation. Concerning these metrics, there exist many studies analyzing and comparing their performances [3,15], combining them based on specific measurements [14], and proposing supervised and unsupervised selection algorithms [2,13].

The main concern of this paper is not the analysis or extension of these methods and metrics which has already been discussed in many recent studies. Instead, we deal with the coverage policy employed during the feature selection process: corpus-based and class-based feature selection approaches are analyzed using the appropriate metrics. The corpus-based approach uses the same feature vector for the discrimination of all the classes by selecting terms from the whole corpus as global keywords and thus favors the prevailing classes. On the other hand, the class-based approach uses a distinct feature vector for each class by considering the document set of each class separately, so that rare classes are represented equally well as the prevailing classes. In this work, we use two alternative selection approaches within these coverage policies. The first one is corpus-based pruning that takes into account the total frequencies of the terms in the whole dataset and filters the less frequent ones. The second one is class-based tf-idf (term frequency - inverse document frequency) metric that focuses on the frequencies of the terms in the documents of a class and favors those terms that do not commonly occur in other classes.

Corpus-based feature selection is the traditional approach used in classification problems: filtering the rare features that occur less than a threshold value is a classical usage of corpus-based selection [11]. As an alternative to the corpus-based approach, class-based feature selection aims at identifying important features for each class separately. A related study covers several feature selection metrics for text classification using support vector machines (SVM) [3]. While this study makes extensive use of class-based features, it also does not include an explicit comparison of the two approaches. A direct comparison between these approaches was performed with the Reuters dataset by using the tf-idf metric [12]. In that work, optimal results were obtained around 2000 terms and the class-based approach yielded significantly more successful results than the corpus-based approach, especially with the macro-averaged F-measure. Reuters is a highly skewed dataset, so it is an expected result for macro-averaged performance to be much more affected by the class-based coverage of the terms. [15] proposes a new metric named as within class popularity for class-based feature selection. They aim at taking two issues about feature selection into consideration, which are the skewness of a dataset and the global importance of a term. Experiments on three datasets showed more successful results than other classifiers for class-based feature selection. In another study, a scalable architecture was proposed and class-based results were given on the Reuters dataset [7].

In this paper, we compare the class-based and corpus-based feature selection approaches using three datasets having different characteristics. The main motivation in the paper is not only making a comparison of these policies, but also analyzing their optimal usage patterns and combining these patterns to obtain higher

classification performances. In this respect, we propose a two-stage feature selection approach that combines corpus-based pruning and class-based tf-idf filtering.

2 Proposed System

2.1 Datasets

In this work, we use three well-known datasets from the UCI Machine Learning Repository: Reuters-21578 (Reuters), National Science Foundation Research Award Abstracts (NSF), and Mini 20 Newsgroups (MiniNg20) [5]. These datasets have different characteristics which may be critical for the classification performance. Skewness is one of the key properties of a dataset that is defined as the distribution of the number of documents over classes. A dataset having a low skewness factor indicates that it is a balanced dataset with approximately the same number of document samples for each class. Allowance of multiple classes for documents (indicating that a document may belong to more than one topic), document length (e.g. short abstracts or long news articles), split proportions (training and test sets), formality level (e.g. formal journal documents or informal internet forum messages) are other properties of datasets.

In the experiments, we use the standard splits of the Reuters and MiniNg20 datasets. For NSF, data related with year 2001 was selected randomly and five sections (four sections for training and one section for test) were picked out from this year. We form five different splits, repeat all the tests with these five cross folds, and take their average as the final result.

2.2 Preprocessing, Document Representation, and Term Weighting

For the preprocessing of the documents, we perform all the standard preprocessing operations such as removal of non-alphabetic characters and mark-up tags, case folding, elimination of stopwords, and stemming. We use the Smart system stoplist for the removal of stopwords (ftp://ftp.cs.cornell.edu/pub/smart) and the widely-used Porter stemmer for extracting the root words.

The bag-of-words (bow) form is accepted as the simplest and the most successful approach for document representation in text classification problems. In this standard approach, only the words in the documents are considered as features in the machine learning algorithm used for classification. Using a machine learning algorithm with these basic features with training and test data is the direct, fundamental and conventional architecture for text classification problems [10].

As the term weighting approach, we use the tf-idf metric which is a simple measure that takes the term frequencies into account and that decreases the importance of terms common to the entire dataset by using the document frequencies [10]. For the optimized tf-idf calculation, each document vector is normalized so that it is of unit length to account for documents of different lengths.

2.3 Machine Learning Algorithm

Several studies have compared the performances of different machine learning approaches and in general SVM with linear kernel was shown to yield successful results [3,6,12]. For the fundamental challenges in the text classification domain (high dimensionality, sparse instances, separability of classes), SVM provides efficient solutions by being more immune to the overfitting problem, using an additive algorithm with an inductive bias that suits problems with dense concepts and sparse instances, and employing a basic linear separation model that fits the discrimination of most of the classes [9]. Based on these positive aspects and its success in previous studies, we decided to use SVM with linear kernel as the machine learning module in this work.

2.4 Feature Selection

As stated in the first section, the main motivation in this paper is focusing on both the corpus-based and class-based feature selection approaches and combining them in such a way that will increase the classifer's performance. On the one hand, for corpus-based feature selection, we apply pruning (filtering low-frequency terms) to the whole dataset and perform an analysis for the optimal pruning level using seven different levels between 2 and 30. In the literature, usually an arbitrarily selected and small value (e.g. 2 or 3) has been used for this purpose. On the other hand, we examine class-based feature selection based on tf-idf (which is superior to corpus-based tf-idf as mentioned previously) by extracting a number of the most informative keywords in each class. We experiment with five different number of keywords between 250 and 4000 to determine the optimal number of keywords. In the rest of the paper, we will refer to these two steps as (corpus-based) pruning and (class-based) keyword selection, respectively. Finally, by analyzing the results of these filtering and selection processes and by extracting parameters corresponding to the optimal performances in these experiments, we derive an additional stage that combines the corpus-based pruning with the class-based keyword selection.

2.5 Methods

We basically implement four main approaches in this work: all words (AW), all words with corpus-based pruning (AWP), all words with class-based keyword selection (AWK), and two-stage feature selection with both pruning and keyword selection (AWPK).

The AW method is the baseline method that uses the standard bow approach with all the words in the feature vector. AWP considers all the words in the document collection, but filters them by the pruning process. In this method, the terms that occur less than a certain threshold value in the whole training set are filtered. We name this threshold value as the *pruning level* (PL). PL=*n* ($n \geq 1$) indicates that terms occurring at least *n* times in the training set are used in the solution vector while the others are ignored. Note that PL=1 corresponds to the AW method (i.e. no pruning). We perform

parameter tuning by analyzing different values for each dataset to reach the optimal PL values for the AWP method. We conduct experiments with different pruning levels between 2 and 30: 2, 3, 5, 8, 13, 20, and 30.

In the AWK method, distinct keywords are selected for each class. This approach gives equal weight to each class in the keyword selection phase. We experiment with five different number of keywords (250, 500, 1000, 2000, and 4000) and compare the results with AW that includes all the words as features in the solution vector. The AWPK method is designed as the optimal combination of AWP and AWK by varying the pruning level and the number of keywords parameters. The parameter values that yield the best results in the underlying methods are used for the AWPK experiments.

3 Experiments and Results

Based on the approaches discussed in the previous section, in this section we determine the optimal parameter values (pruning level and number of keywords) for the methods in all the datasets. The experiments were evaluated and the methods were compared with respect to micro-averaged F-measure (MicroF), which is an average of the success rates of the documents, and macro-averaged F-measure (MacroF), which is an average of the success rates of the categories [10].

3.1 Pruning Level Analysis - AWP

In this experiment, the AWP method was implemented with several PL values (PL=1 corresponds to AW) for the three datasets. Table 1 shows the feature number and the MicroF and MacroF success rates for each pruning level. The first column of the table indicates the method and the value of the PL parameter, separated by comma. As can be seen, the pruning process improves the success rate of the classifier and the best results (high accuracies with low feature numbers) are obtained around PL=13 consistently in all the three datasets with two different performance measures. By following the generalization that words occurring less than 10-15 times in a dataset are most probably not a good indicator for the classification of texts [11], we set PL=13 in the pruning-based experiments. This result indicates that the usual belief in the literature that a pruning level of 2-3 suffices to eliminate uninformative terms does not hold.

Method, Parameter	Reuters			NSF			MiniNg20		
	Feature#	MicroF	MacroF	Feature#	MicroF	MacroF	Feature#	MicroF	MacroF
AW	20292	85.58	43.83	13424	64.46	46.11	30970	46.42	43.44
AWP,2	12959	85.55	43.84	8492	64.41	46.21	13102	49.73	47.13
AWP,3	9971	85.52	43.93	6328	64.62	46.42	9092	49.64	47.19
AWP,5	7168	85.51	44.56	4528	64.86	46.49	6000	51.26	48.52
AWP,8	5268	85.73	44.91	3376	64.66	46.38	4169	52.48	49.90
AWP,13	**3976**	**85.84**	**44.85**	**2478**	**64.58**	**46.49**	**2863**	**53.62**	**51.02**
AWP,20	3046	86.02	44.55	1875	64.23	46.67	2025	53.78	51.02
AWP,30	2237	81.29	43.59	1419	63.84	46.21	1384	52.89	50.46

Table 1. AWP Success Rates (Optimal Results Shown in Bold)

3.2 Class-based Keyword Selection Analysis - AWK

In this experiment, the performance of the AWK method was analyzed using different keyword (feature) number parameters. The results are shown in Table 2. The success rates for AW are also included in the table for comparison.

Method, Parameter	Reuters		NSF		MiniNg20	
	MicroF	MacroF	MicroF	MacroF	MicroF	MacroF
AWK,250	83.69	51.15	62.04	49.51	56.65	55.72
AWK,500	84.71	50.92	62.92	49.31	56.16	55.01
AWK,1000	85.16	51.72	64.69	49.33	53.68	52.17
AWK,2000	**85.58**	**52.03**	**65.19**	**49.31**	**54.04**	**52.10**
AWK,4000	**85.84**	**52.10**	**65.71**	**49.35**	**55.25**	**53.73**
AW	85.58	43.83	64.46	46.11	46.42	43.44

Table 2. AWK Success Rates (Optimal Results Shown in Bold)

In general, the AWK method with number of keywords between 2000 and 4000 increases the success rates in all the datasets compared to the AW method. Therefore, we can conclude that using a specific set of keywords for each class gives more successful results than using all the words in the feature vector.

When we analyze the results of AWP and AWK together, we see that the improvement of AWP over AW is explicit in the balanced dataset (MiniNg20) while there is less improvement in the skewed datasets (Reuters and NSF). On the other hand, the improvement of AWK over AW is more significant than that of AWP in all the datasets. This performance increment is more explicit in the MacroF measure. In corpus-based approaches documents of rare classes tend to be more misclassified since the words of prevailing classes dominate the feature vector. The MacroF measure gives equal weight to each class in determining the success rate of the classifier. Thus, especially for highly skewed datasets, when the rare classes are not represented well with the selected features, average of correct classifications for rare classes drops dramatically. This is the case for both AW and AWP in skewed datasets that use a common set of features for all the classes. However, with class-based keyword selection, since each class has its own keywords during classification, rare classes are characterized in a more successful way. So, we observe a significant success rate (MacroF) increase with the AWK method in skewed datasets.

3.3 Two-stage Feature Selection Analysis - AWPK

The AWPK method combines the optimal usage patterns of the AWP and AWK approaches. Therefore, the parameters in the method are the pruning level and the number of keywords. In this experiment, we use the optimal values of these parameters determined during the previous analyses for each dataset: pruning level 13 and number of keywords 2000 and 4000. The results are given in Table 3. The table also shows the best performances of AW, AWP, and AWK for comparison.

Method, Parameters	Reuters		NSF		MiniNg20	
	MicroF	MacroF	MicroF	MacroF	MicroF	MacroF
AWPK,13,2000	**86.40**	**53.95**	**66.06**	**50.11**	**57.43**	**55.66**
AWPK,13,4000	**86.70**	**53.98**	**66.10**	**50.12**	**57.43**	**55.66**
AW	85.58	43.83	64.46	46.11	46.42	43.44
AWP,13	85.84	44.85	64.58	46.49	53.62	51.02
AWK,2000	85.58	52.03	65.19	49.31	54.04	52.10
AWK,4000	85.84	52.10	65.71	49.35	55.25	53.73

Table 3. AWPK Success Rates (Optimal Results Shown in Bold)

As can be seen in the table, the two-stage feature selection approach outperforms the previous approaches. Selecting the best 2000-4000 keywords for each class with an initial pruning step significantly improves the best performances of AWP (with PL=13) and AWK (with 2000-4000 keywords) in all the three datasets. So, we can conclude that the incremental effect of corpus-based pruning continues when it is combined with the class-based tf-idf keyword selection metric. As a result, the method proposed in this work, AWPK, yields the best performance.

The significance of the results for the three methods were measured using the statistical sign test. We observed that, in general, each method significantly outperforms its predecessor method. In this sense, AWP and AWK are significantly better than the standard benchmark method AW, and AWPK is significantly better than both AWP and AWK. So, the most advanced method in this study (AWPK) is the optimal method with its two-stage feature selection analysis.

3 Conclusions

In this paper, we focused on feature coverage policies (corpus-based or class-based selection of features) used in the text classification domain. First, we analyzed the performances of corpus-based pruning (AWP) and class-based keyword selection with tf-idf (AWK) separately. Then, determining the optimal parameter values for each method, we formed the AWPK method which is a combination of these two approaches. To the best of our knowledge, this is the first work that combines class-based and corpus-based feature selection in the text classification domain.

A possible future work is applying the two-stage feature selection approach to more semantically-oriented text classification methods, such as those using language models, linguistic features, or lexical dependencies. Integrating the concepts of pruning and keyword selection into those methods as two consecutive steps may lead to a higher classification performance.

Acknowledgement

This work was supported by the Boğaziçi University Research Fund under the grant number 05A103D and the Turkish State Planning Organization (DPT) under the TAM Project, number 200K120610.

References

1. Aizawa, A., Linguistic techniques to improve the performance of automatic text categorization, in *Proceedings of 6th natural language processing pacific rim symposium*, 307-314, Tokyo, 2001.
2. Dasgupta, A., P. Drineas, B. Harb, V. Josifovski, and M.W. Mahoney, Feature selection methods for text classification, in *Proceedings of 13th international conference on knowledge discovery and data mining*, 230-239, ACM, San Jose, 2007.
3. Forman, G., An extensive empirical study of feature selection metrics for text classification, *Journal of Machine Learning Research*, 3:1289-1305, 2003.
4. Forman, G., Feature selection for text classification, in Computational methods of feature selection, ed. Liu, H. and Hiroshi M. Chapman and Hall/CRC Press, 2007.
5. Frank, A., and A. Asuncion., *UCI machine learning repository*, University of California, School of Information and Computer Science, Irvine, CA, http://archive.ics.uci.edu/ml, 2010.
6. Gao, Y., and S. Sun., An empirical evaluation of linear and nonlinear kernels for text classification using support vector machines, in *Proceedings of the 7th international conference on fuzzy systems and knowledge discovery (FSKD)*, 1502-1505, IEEE, China, 2010.
7. Ghiassi, M., Olschimke, M., Moon, B. and Arnaudo, P., Automated text classificiation using a dynamic artificial neural network model, *Expert Systems with Applications*, 39:12, 10967-10976, 2012.
8. Joachims, T., Text categorization with support vector machines: learning with many relevant features, in *Proceedings of the European conference on machine learning (ECML)*, 137-142, Springer, 1998.
9. Joachims, T., *Advances in kernel methods: support vector learning*, MIT Press, 1999.
10. Manning, C, P. Raghavan, and H. Schütze, *Introduction to information retrieval*, Cambridge University Press, 2008.
11. Özgür, L., and T. Güngör, Text classification with the support of pruned dependency patterns, *Pattern Recognition Letters*, 31:1598—1607, 2010.
12. Özgür, A., L. Özgür, and T. Güngör, Text categorization with class-based and corpus-based keyword selection, in *Proceedings of the 20th international symposium on computer and information sciences (ISCIS)*, 606-615, Springer, Istanbul, 2005.
13. Shang, W., H. Huang, H. Zhu, Y. Lin, Y. Qu, and W. Zhihai, A novel feature selection algorithm for text categorization, *Expert Systems with Applications*, 33:1-5, 2007.
14. Shoushan, L., X. Rui, Z. Chengqing, and C.R. Huang, A framework of feature selection methods for text categorization, in *Proceedings of the 47th annual meeting of the ACL*, 692-700, Singapore, 2009.
15. Singh, S.R., H.A. Murthy, and T.A. Gonsalves, Feature selection for text classification based on Gini coefficient of inequality, in *Proceedings of the 4th international workshop on feature selection in data mining*, 76-85, India, 2010.