An Empirical Study of Feature Selection for Sentiment Analysis

Pedro L. Varela*, André F. T. Martins*†§, Pedro M. Q. Aguiar*‡, Mário A. T. Figueiredo*§

*Department of Electrical and Computer Engineering, Instituto Superior Técnico, Lisboa, Portugal

†Priberam Labs, Lisboa, Portugal

‡Instituto de Sistemas e Robótica, Lisboa, Portugal

§Instituto de Telecomunicações, Lisboa, Portugal

***Abstract*—Sentiment analysis is a text classification task where the goal is to determine the polarity (positive or negative) of the opinion expressed in a document. This task is typically addressed using machine learning tools, based on the standard bag-of-words description of the documents; the high dimensionality of these features makes feature selection an important step in this class of problems. This paper reports an extensive comparative study of feature selection (FS) methods in sentiment analysis, using two standard classifiers: *naïve Bayes* (NB) and *support vector machines* (SVM). Furthermore, a new *weighted SVM* (WSVM) is proposed, where the features are weighted using the scores of a feature selection method. The proposed WSVM is shown to achieve better performance in the sentiment analysis task than the standard SVM, especially when the weighting is done using the mutual information feature scores.**

I. Introduction

Text categorization is a classification task, defined as automatically assigning predefined category labels to new free text documents. A growing number of statistical machine learning techniques have been recently applied to text categorization, with the most popular being *naïve Bayes* (NB) and *support vector machines* (SVM) [1].

One major difficulty in text categorization problems is the high dimensionality of feature vectors that are typically used for textual data. In fact, each distinct feature appearing in the document collection represents one dimension in the feature space. For a typical document collection, there may exist up to hundreds of thousands of distinct features, usually indicating the presence or absence of terms in each document. This high dimensionality may lead to over-fitting, by adapting a classification system to the a specific training data set. Feature selection (FS) methods are thus important to reduce the feature dimensionality, helping to improving generalization accuracy (reducing over-fitting), as well as reducing the training time and fighting the curse of dimensionality. Sparse models, with just a few features, are also desirable for compactness, low memory footprint and interpretability.

Many FS methods have been proposed in the literature, including Document Frequency (DF), Term Frequency (TF), Term Frequency-Inverse Document Frequency (TF-IDF), Term Contribution (TC), Information Gain (IG), Signed IG (SIG), Gain Ratio (GR), Cross Entropy for Text (CET), Mutual Information (MI), Chi-square (CHI), Correlation Coefficient (CC), Odds ratio (OR) and variations, Gini Index (GINI), Bi-Normal Separation (BNS), Weighted Log Likelihood Ratio (WLLR), Weight of Evidence of Text (WET), GU metric (GU) and GSS coefficient (GSS) [2]–[15].

In this paper, we present an extensive comparative empirical study of all of the above FS methods, in the binary classification task of text sentiment analysis (SA). This is a particular text categorization task, where the goal is to classify texts in terms of the emotion they express. Arguably, the most popular classifiers used for SA are the above mentioned NB and SVM. In this paper, we also propose a modified version of the SVM, termed *weighted* SVM (WSVM), where the features are weighted using the scores of a feature selection method.

II. Methods

A. Feature Selection

Table I presents all the FS methods considered in this paper. The goal of a FS method is to score the features by order of some measure of relevance, estimated from some training set. Most of these methods are defined using some statistical measure of the relationships between each feature w_i (in this paper we consider binary features) and the category variable c_k (which is also binary in the SA problem). In Table I, A is defined as the number of documents of category c_k, in which w_i occurs, B is the number of documents that do not belong to c_k, but in which w_i occurs, C is the number of documents of category c_k in which w_i does not occur, and D is the number of documents that do not belong to c_k, and do not contain w_i. N is the training corpus dimension and N_k is the number of documents belonging to c_k. The final score of methods that are class dependent and have different values for each category, can be determined by two popular methods: the average score of all category values or the maximum value. When applicable, both options were evaluated, with the corresponding scores denoted as FS_{avg} and FS_{max}, respectively.

TABLE I
FEATURE SELECTION METHODS

Document Frequency (DF)	$\text{DF}(w_i) = A + B$
Term Frequency (TF)	$\text{TF}(w_i) = \sum_{j=1}^{\lvert\mathcal{D}\rvert} N_{ij}$
Term Frequency - Inverse Document Frequency (TF-IDF)	$\text{TF-IDF}(w_i) = \text{TF}(w_i) \log\left(\frac{N}{\text{DF}(w_i)}\right)$
Term Contribution (TC)	$\text{TC}(d_g, d_j) = \sum_{w_i} \text{TF-IDF}(w_i, d_g) \times \text{TF-IDF}(w_i, d_f)$
Mutual Information (MI)	$\text{MI}(w_i, c_k) = \log \frac{N \times A}{(A+B)\times(A+C)}$
Information Gain (IG)	$\text{IG}(w_i) = -\sum_{k=1}^{\lvert\mathcal{C}\rvert} \frac{N_k}{N} \log_2 \frac{N_k}{N} + \sum_{k=1}^{\lvert\mathcal{C}\rvert} \frac{A}{N} \sum_{k=1}^{\lvert\mathcal{C}\rvert} \frac{A}{A+B} \log_2 \frac{A}{A+B} + \sum_{k=1}^{\lvert\mathcal{C}\rvert} \frac{C}{N} \sum_{k=1}^{\lvert\mathcal{C}\rvert} \frac{C}{C+D} \log_2 \frac{C}{C+D}$
Signed IG (SIG)	$\text{SIG}(w_i, c_k) = \text{sign}(AD - CB)\text{IG}(w_i)$
Gain Ration (GR)	$\text{GR}(w_i, c_k) = \frac{\text{IG}(w_i)}{SplitInfo} \qquad SplitInfo = -\sum_k \frac{\lvert N_k\rvert}{\lvert N\rvert} \log_2 \frac{\lvert N_k\rvert}{\lvert N\rvert}$
Cross Entropy for Text (CET)	$\text{CET}(w_i) = \text{DF}(w_i) \sum_{k=1}^{\lvert\mathcal{C}\rvert} A \log_2 \frac{A}{\frac{N_k}{N}}$
Chi square test (CHI)	$\text{CHI}(w_i, c_k) = \frac{N \times (AD-BC)^2}{(A+C)\times(B+D)\times(A+B)\times(C+D)}$
Correlation Coeficient (CC)	$\text{CC}(w_i, c_k) = \frac{\sqrt{N}(AD-CB}{\sqrt{(A+C)\times(B+D)\times(A+B)\times(C+D)}}$
GSS Coeficient (GSS)	$\text{GSS}(w_i, c_k) = AD - CB$
Odds Ratio (OR)	$\text{OR}(w_i, c_k) = \log \frac{AD}{CB}$
Bi-Normal Separation (BNS)	$\text{BNS}(w_i, c_k) = \frac{A}{\text{DF}(w_i)} \frac{N_k}{N} \log \frac{A}{B} \frac{N_k}{N - N_k}$
Gini Index (GINI)	$\text{Gini}(w_i, c_k) = \frac{A^2}{\log_2 \text{DF}(w_i)}$
Weighted Log Likelihood Ratio (WLLR)	$\text{WLLR} = \frac{A}{N_k} \log \frac{A(N-N_k)}{CN_k}$
Weight of Evidence of Text (WET)	$\text{WET}(w_i) = \sum_{k=1}^{\lvert\mathcal{C}\rvert} \text{DF}(w_i) \frac{N_k}{N} \left\lvert \log\left(\frac{A}{B} \frac{N-N_k}{N_k}\right)\right\rvert$
GU Metric (GU)	$\text{GU}(w_i, c_k) = \lvert z\rvert \frac{A(N-N_k)}{BN_k}$

B. Multinomial Naïve Bayes (MNNB) Classifier

NB is a widely used method for document classification. Given a feature vector table, the algorithm computes the posterior probability that the document belongs to the different classes and assigns it to the class with the highest posterior probability, assuming that the features are conditionally independent, given the class. In this paper, we use the multinomial naïve Bayes (MNNB) [16], which counts the frequency of feature, w_i, given the category, c_k, according to

$$\hat{P}(w_i|c_k) = \frac{1 + \sum_{j=1}^{N} N_{ij}\delta_{jk}}{|\mathcal{V}| + \sum_{i=1}^{|\mathcal{V}|}\sum_{n=1}^{N} N_{in}\delta_{jk}} \tag{1}$$

where $|\mathcal{V}|$ is the number of features and N the number of documents, δ_{jk} is 1 if the j-th document belongs to class c_k and 0 otherwise, and N_{ij} is the number of times that feature w_i occurs in the j-th document. Notice that (1) uses the so-called *add-one smoothing* method, to avoid zero or one probabilities. The class prior probabilities $\hat{P}(c_k)$ are given simply by N_k/N. Finally, the classification rule applied to some document d is

$$C(d) = \arg\max_{c_k \epsilon \mathcal{C}} \left(\log\left(\hat{P}(c_k)\right) + \sum_{i=1}^{|\mathcal{V}|} N_i(d) \log \hat{P}(w_i|c_k) \right)$$

where $\mathcal{C}$ is the set of classes and $N_i(d)$ is the number of times that feature w_i occurs in document d.

C. Support Vector Machine (SVM)

SVM was first introduced by [17] to perform binary classification. Based on the structural risk minimization principle from computational learning theory, an SVM seeks a decision boundary separating the training data into two classes with maximal margin; the resulting classifier makes decisions based only on a subset of the training points (the *support vectors*). Many variants of SVM have been developed and they have been first adopted for text classification tasks in [1]. In this paper, we use the SVM implementation provided by the LibLinear package [18] with a linear kernel.

D. Weighted SVM (WSVM)

Following the work of [19], only binary presence/absence feature values were used with the SVM. However, other feature values or non-linear kernels may be used, such as the scores achieved with FS methods. This proposed method is named *weighted* SVM (WSVM), and referred to as (FS)-WSVM, where FS refers to the method that provides the feature weights. In fact, the binary values used for the SVM are nothing more than the values of the *term presence* (TP) FS method. The rationale behind WSVM is that the decision boundary is determined according to the relevance of the features, since the usual methods for feature scaling (such as TF-IDF) have no information regarding categories.

III. Results and Discussion

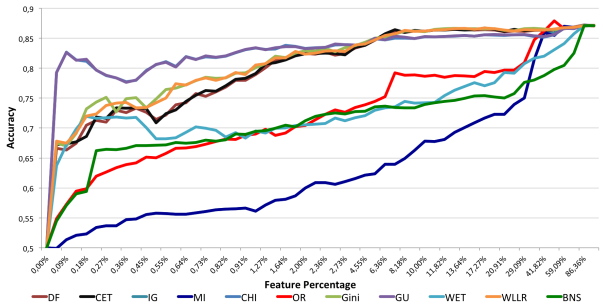
Two balanced (equal numbers of positive and negative instances) corpora of movie reviews were used in this study: an original corpus with 3672 reviews in Spanish and an English Corpus of 2000 reviews. The latter is the corpus introduced in [19], which is a widely used benchmark for SA research. The classification accuracy was assessed using 10-fold cross validation (CV). A feature space composed of unigrams, bigrams and trigrams was used for all evaluations (an n-gram is a sequence of length n of contiguous features). As is common in text classification, punctuation and stop-words were removed, and all the letters were changed to lowercase.

In this particular case of binary classification with a balanced corpus, some FS methods behave similarly (Table II), and only the representative ones are presented in this paper. Furthermore, the fact that both corpora are balanced and we are performing binary classification, many of the FS methods are also equivalent in terms of performance.

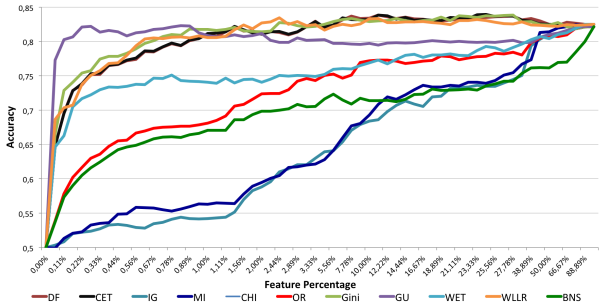
TABLE II
FS methods that perform similarly in this task

FS Represented	FS that have similar behaviour
DF	TF, TFIDF, TC
IG	SIG, GR
CHI	CC, GSS

The accuracies of the MNNB and SVM classifiers as a function of the number of features selected by the several FS methods are shown in Fig. 1-2. Note that the axis of the percentage of features is warped to better show the evaluation of the performance. The SVM is much faster than MNNB, it is less prone to over-fitting, and scales better to considerably larger dimensions, so it is expected that FS has lower impact on its performance. Although the FS methods studied behave differently, using less than 10% of features, with any FS method, significantly hurts the performance. The SVM reacher its performance plateau around 30% of the features, while the MNNB reaches it around 50%. Also, for the MNNB, some FS methods yield a slightly higher accuracy around 20% than with the all the features, arguably due to some loss of generalization ability. The FS methods that consistently performed better with both classifiers were DF, Gini, CET and CHI. GU performed very well with the SVM, but poorly with MNNB. The method that performed worse for both classifiers was MI.



(a) SVM performance



(b) MNNB performance

Fig. 1. Evolution of accuracy with the proportion of features selected for the English corpus

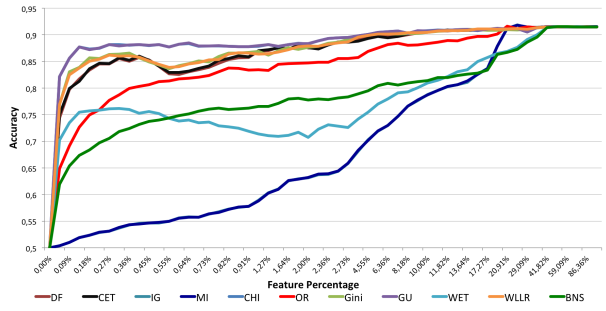
Accuracies achieved with only 10% are not much lower than those achieved with all the feature space, as can be seen in Table III (using the CHI method, for illustration purposes).

TABLE III
SVM and MNNB accuracy with and without FS

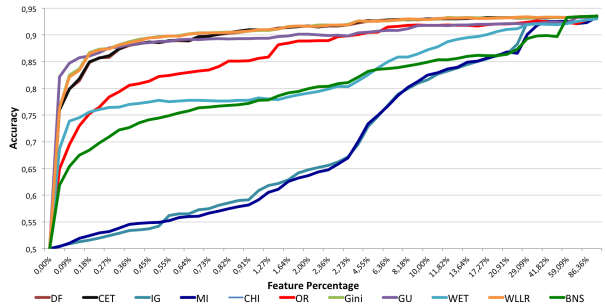
	English corpus		Spanish corpus	
	SVM	MNNB	SVM	MNNB
10%	0.8335	0.8090	0.8780	0.8932
100%	0.8715	0.8215	0.9143	0.9357

A. Weighted SVM

Although MI did not show a good performance, when compared to the other FS methods considered, its scores turned out to be the best for use as weights with the WSVM. The fact that MI assumes that features with high category ratio are more effective for classification, giving rare features a higher score than common features, allows the classifier to discriminate features according to category related information, which may be the reason behind the good performance of this method. In



(a) SVM performance



(b) MNNB performance

Fig. 2. Evolution of accuracy with the proportion of features selected for the Spanish corpus

both corpora, the WSVM performed better than the standard SVM (with binary feature vectors), proving to be an improvement over the SVM for this task. However, the WSVM did not outperform the MNNB classifier on the Spanish corpora.

TABLE IV
WEIGHTED SVM, SVM AND MNNB COMPARATIVE

	SVM	MNNB	WSVM		
			MI_{max}	CHI	TF-IDF
English Corpus	0.8710	0.8250	0.8795	0.8515	0.8035
Spanish Corpus	0.9150	0.9350	0.9291	0.9007	0.8761

IV. CONCLUSION

In this paper, we presented a comprehensive study of feature selection (FS) methods for text classification, focusing on the particular task of sentiment analysis, using *support vector machines* (SVM) and *naïve Bayes* (NB) classifiers. The results show that different combinations of FS methods and classifiers yield quite different accuracies. The results show that the SVM needs much fewer features to achieve an almost maximal accuracy, and is less affected by the choice of FS method. In comparison, the NB classifier requires more features to perform close to its best, and is more sensitive to the choice of FS method.

The best FS methods for this task are are the same to both classifiers, with the sole exception of the GU method, which performs well with the SVM and poorly with the NB classifier. The difference in performance amongst FS methods, and the coherence between classifiers, allows concluding that choosing a good FS method is very important when used.

Finally, we have also proposed a modified version of the SVM, termed *weighted* SVM (WSVM), where the features are weighted using the scores of a feature selection method. The WSVM (using the mutual information weights) outperforms the standard SVM in all the tests, including the English corpus, which is a widely used benchmark data-set for sentiment analysis.

REFERENCES

[1] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," *Machine Learning: ECML-98*, pp. 137–142, 1998.

[2] Y. Yang and J. Pedersen, "A comparative study on feature selection in text categorization," in *Proceedings of the Fourteenth International Conference on Machine Learning ICML97*, 1997, pp. 412–420.

[3] W. Zhang, T. Yoshida, and X. Tang, "A comparative study of tf* idf, lsi and multi-words for text classification," *Expert Systems with Applications*, 2010.

[4] H. Liu, H. Lieberman, and T. Selker, "A Model Of Textual Affect Sensing using Real-World Knowledge," pp. 1–8, 2003.

[5] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.

[6] ——, *C4.5 - programs for machine learning*. Morgan Kaufmann, 1993.

[7] D. Mladenic and M. Grobelnik, "Feature selection for unbalanced class distribution and naive bayes," in *Proceedings of the 16th International Conference on Machine Learning ICML-99*. Morgan Kaufman, 1999, pp. 258–267.

[8] H. Schütze, D. Hull, and J. Pedersen, "A comparison of classifiers and document representations for the routing problem," in *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1995, pp. 229–237.

[9] H. Ng, W. Goh, and K. Low, "Feature selection, perceptron learning, and a usability case study for text categorization," in *ACM SIGIR Forum*, vol. 31, no. SI. ACM, 1997, pp. 67–73.

[10] L. Galavotti, F. Sebastiani, and M. Simi, "Experiments on the use of feature selection and negative evidence in automated text categorization," *Research and Advanced Technology for Digital Libraries*, pp. 59–68, 2000.

[11] C. Van Rijsbergen, "Information retrieval, chapter 7," *Butterworths, London*, vol. 2, pp. 111–143, 1979.

[12] H. Park, S. Kwon, and H. Kwon, "Complete gini-index text (git) feature-selection algorithm for text classification," in *Software Engineering and Data Mining (SEDM), 2010 2nd International Conference on*. IEEE, 2010, pp. 366–371.

[13] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using em," *Machine learning*, vol. 39, no. 2, pp. 103–134, 2000.

[14] D. Mladenic, "Machine learning on non-homogeneous, distributed text data," *Doctoral Dissertation, University of Ljubljana*, 1998.

[15] G. Uchyigit and K. Clark, "A new feature selection method for text classification," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 21, no. 2, p. 423, 2007.

[16] A. McCallum and K. Nigam, "A comparison of event models for naive bayes text classification," *AAAI-98 Workshop on Learning for Text Categorization*, pp. 41–48, 1998.

[17] V. Vapnik, *The nature of statistical learning theory*. springer, 1995.

[18] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[19] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *Proc. ACL Conference on Empirical Methods in Natural Language Processing*, 2002, pp. 79–86.