Expert Systems with Applications 40 (2013) 4871–4886

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Comparison of text feature selection policies and using an adaptive framework

Şerafettin Taşcı, Tunga Güngör *

Boğaziçi University, Computer Engineering Department, Bebek, 34342 İstanbul, Turkey

ARTICLE INFO

Keywords:
Document categorization
Feature selection
Local and global policies
Adaptive keyword selection
Support vector machines

ABSTRACT

Text categorization is the task of automatically assigning unlabeled text documents to some predefined category labels by means of an induction algorithm. Since the data in text categorization are high-dimensional, often feature selection is used for reducing the dimensionality. In this paper, we make an evaluation and comparison of the feature selection policies used in text categorization by employing some of the popular feature selection metrics. For the experiments, we use datasets which vary in size, complexity, and skewness. We use support vector machine as the classifier and tf-idf weighting for weighting the terms. In addition to the evaluation of the policies, we propose new feature selection metrics which show high success rates especially with low number of keywords. These metrics are two-sided local metrics and are based on the difference of the distributions of a term in the documents belonging to a class and in the documents not belonging to that class. Moreover, we propose a keyword selection framework called adaptive keyword selection. It is based on selecting different number of terms for each class and it shows significant improvement on skewed datasets that have a limited number of training instances for some of the classes.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

Text categorization is the task of automatically assigning unlabeled text documents to some predefined category labels by means of an induction algorithm. It has gained great popularity and importance in recent years since the amount of documents in electronic media which necessitate some form of organization and arrangement increased considerably. A large number of statistical techniques and machine learning approaches have been used for this task such as naive Bayes, linear regression, Rocchio classification, neural networks, k-nearest neighbors (k-NN) clustering, and support vector machines (Sebastiani, 2002).

In text categorization, generally a document is represented as a set of words without regard to the grammar and word order. This representation is called the bag of words model. Since a document collection may contain thousands of words, a bag of words representation of a document will probably have a very high dimensionality. This situation is a critical challenge for most learning algorithms. Therefore, normally feature selection is used in text categorization systems for the purpose of reducing the dimensionality. Dimensionality reduction has many benefits such as improving the interpretability of the data, reducing the time and storage requirements, and speeding up the learning process. Moreover, it may improve the classification accuracy since it can prevent overfitting by eliminating the terms that are useless or misleading for the classifier.

Feature selection on textual data is mostly based on feature ranking in which all features are evaluated by a metric that estimates their importance and then the ones with the highest scores are selected. There are mainly two ways for selecting features: locally and globally. In local policy each category is represented with a different set of keywords, while in global policy the feature set is created globally and it is the same for all categories. Local policy helps us to find the most important terms for each class, whereas global policy favors the prevailing classes and gives penalty to classes with small number of training documents.

A fundamental factor that has an impact on performance in text categorization experiments is the characteristics of the dataset used, which are the size of the dataset, the number of terms, and the skewness property. Especially, skewness (class imbalance) is a major determinant of the classification performance (Chawla, Japkowicz, & Kotcz, 2004). Highly skewed datasets are particularly hard to categorize since the common classes may dominate the rare classes. Therefore, feature selection and document classification algorithms may show a biased behavior by classifying the common classes successfully while largely ignoring the rare classes.

In this paper, we study the binary classification problem with support vector machines (SVM), where each document is classified into one of two categories. The document either belongs to a given class or does not. The paper presents an evaluation of feature selection policies by using some popular feature selection metrics. We evaluate the policies by concentrating on the following aspects:

* Corresponding author. Tel.: +90 (212) 359 7094.
E-mail addresses: serafettin.tasci@boun.edu.tr (Ş. Taşcı), gungort@boun.edu.tr (T. Güngör).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.02.019

- Comparison and analysis of different feature selection metrics.
- Evaluation of local and global policies for each feature selection metric.
- Investigation of the effect of datasets with different characteristics on the performances of the metrics.

In addition to the evaluation of feature selection policies and well-known metrics, we propose some new feature selection metrics which are more advanced forms of the Acc2 metric that was studied by Forman (2003). These proposed metrics are two-sided metrics (that is, they take into account the negative features as well as the positive features) and are based on the difference of the distributions of a term in the documents belonging to a class and in the documents not belonging to that class. They are all local metrics and achieve high performance rates even at small number of keywords. This makes them precious especially when the practitioner is constrained to use a limited number of keywords.

We also propose a novel feature selection framework called adaptive keyword selection (AKS) which selects different number of terms for classes that have different sizes. It is inspired by the observation that classification performances are better with high number of keywords in datasets that contain an abundant number of examples for each class, while the performances are better with low number of keywords in skewed datasets that contain very few examples for some of the classes. In accordance with our expectations, it shows significant performance improvements on skewed datasets that have a limited number of training instances for some of the classes.

The rest of the paper is organized as follows: Section 2 presents an overview of the literature about feature selection in text categorization. In Section 3, we describe the existing feature selection methods that are used in this study and the ones that we propose. Section 4 explains the experimental settings; the classifier, the datasets, the evaluation criteria, and the preprocessing steps. In Section 5, we show the results of the experiments and give a comparative and detailed discussion of these results. Section 6 concludes the paper.

2. Related work

Text categorization is a learning task which aims at predicting the category labels of unlabeled documents by using a training set. Therefore, most of the machine learning algorithms such as SVMs, neural networks, naive Bayes, and k-NN can be used for this task. There are several studies in the literature that compare the performances of the learning algorithms in the text categorization domain (Sebastiani, 2002; Sriurai, 2011). It was found that SVM is generally a top performer in this task (Forman, 2003; Joachims, 1998; Lan, Tan, Su, & Lu, 2009; Yang and Liu, 1999).

Feature selection is an important topic in learning tasks where datasets with high number of features are common. There exist a great deal of works about feature selection that are not focused on textual data (Camps-Valls, Mooij, & Schölkopf, 2010; Guyon & Elisseeff, 2003; Rakotomamonjy, 2003; Yu & Liu, 2004). Since textual data have thousands of dimensions, most of the general feature selection methods are not efficient or successful in text categorization. For example, wrapper methods which search the space of all possible feature subsets perform very well on low-dimensional data. However, it is inefficient to use these methods on text documents (Pinheiro, Cavalcanti, Correa, & Ren, 2012; Li, Xia, Zong, & Huang, 2009). Likewise, embedded methods perform feature selection in the process of training and reach a solution faster by avoiding retraining the learning machine when each feature is selected (Grigorescu, Petkov, & Kruizinga, 2002). For instance, the recursive feature elimination method (Chen, Zeng, & van Alphen, 2006) makes use of the change in the objective functions as a ranking criterion when a feature is removed. With a backward elimination strategy, the features that contribute least to the classification are removed iteratively. This is an efficient strategy for texture classification in computer vision systems; however eliminating features iteratively is not feasible in a domain with high dimensionality such as text classification. Therefore, methods used in the text domain are mostly based on the scoring of features.

Feature selection is at least as important as the choice of the induction algorithm in text categorization. Accordingly, many studies to evaluate the feature selection metrics have been done by researchers. However, it is hard to generalize the findings of these studies and compare the result of a study with another one because of the variations in the evaluation metrics and experimental settings such as datasets, classifiers, term weighting methods, feature selection policy, and preprocessing.

Forman (2003) considers local policy and gives a comprehensive evaluation of many feature selection metrics. SVM is used as the classifier and many datasets including skewed datasets as well as homogenous ones are evaluated. A work that includes both of the feature selection policies is carried out by Debole and Sebastiani (2003). Nevertheless, they focus on a new term weighting scheme using the feature selection scores of the terms and thus they do not give a detailed comparison of the policies. Özgür, Özgür and Güngör (2005) compare local and global policies by using SVM as the induction algorithm. However, they use a single dataset and do not analyze the effect of these keyword selection approaches for document corpora of varying class distributions. In the study, local and global policies are named as class-based and corpus-based keyword selection, respectively. In a later study (Özgür and Güngör, 2007), they analyze the two keyword selection policies on datasets with different skewness properties and sizes. However, they only use the tf-idf metric for keyword selection and do not consider the popular feature selection metrics such as information gain, chi-square, and document frequency thresholding. Bakus and Kamel (2006) investigate some more advanced feature selection approaches that employ higher order decisions and that take the feature-to-feature correlation into account when selecting the feature set, such as odds ratio, correlation-based feature selection (CFS) and Markov blanket. Li et al. (2009) compare six popular feature selection metrics on topic-based and sentiment classification tasks. They analyze the methods for low and high feature numbers separately and derive results about the performances of different methods on these cases.

In addition to these comparative studies, some successful methods for feature selection have also been developed. One such example is given by Forman (2003) where a method called bi-normal separation, which is especially successful in highly skewed datasets, is proposed. Another example is gain ratio, which is acquired by normalizing the information gain score of a term by its entropy (Debole & Sebastiani, 2003). An approach based on unsupervised feature selection is proposed by Dasgupta, Drineas, Harb, Josifovski and Mahoney (2007). Their algorithm assigns a univariate importance score to each feature. It then randomly samples a small number of features (independent of the total number of features, but dependent on the number of documents and an error parameter) and solves the classification problem induced on those features. Li et al. (2009) propose a method named as weighted frequency and odds (WFO) that has the effect of combining different feature selection methods by parameter tuning. Experiments on four datasets and a comparison with other methods show that it performs robustly across different domains and gives comparable results. There are methods based on Gini index theory which was used earlier in decision trees for splitting attributes and achieved better categorization precision rates. Singh, Murthy and Gonsalves (2010) discuss how the Gini index can be effectively used for feature selection in text categorization and acquires results comparable to the well-known metrics.

There are also some works that use supervised techniques for term weighting where the scores of the terms in the feature selection phase are also used in the term weighting phase. A method called supervised term weighting is proposed in which the idf part of the tf-idf term weighting formula is replaced by the score of the term that is calculated in the feature selection phase (Debole & Sebastiani, 2003). Likewise, Soucy and Mineau (2005) introduce a method called ConfWeight which is a weighting method based on the statistical estimation of the word importance for a particular categorization problem. Xu, King, Lyu and Jin (2010) apply feature selection in a semi-supervised manner by using a small set of labeled data. They propose a new feature selection approach based on the maximum margin principle to increase the discriminative power of the classifier.

The class imbalance problem is encountered in a large number of practical applications of machine learning and data mining. It has been widely realized that class imbalance gives rise to problems that are either nonexistent in or more difficult to handle than balanced class cases and often causes a classifier to perform suboptimally. The problem is more severe when the imbalanced data are also high dimensional as in the case of text documents. In such cases, feature selection methods are critical to achieve optimal performance (Chen & Wasikowski, 2008). In Forman (2004), it is claimed that most feature selection metrics based on feature scoring can be blinded by a surplus of strongly predictive features for some classes, while largely ignoring features needed to discriminate difficult classes. They propose solutions to this pitfall by scheduling approaches. Zheng, Wu and Srihari (2004) propose a framework for finding the optimal combination of features that signal non-membership (negative) and membership (positive) aspects on imbalanced textual data. In a similar work, the effect of three types of metrics on imbalanced datasets was analyzed (Ogura, Amano, & Kondo, 2011). It was found that metrics that implicitly combine positive and negative features denoting membership and non-membership outperform metrics that are used in a combination setting by explicitly combining positive and negative features.

Due to the existence of various feature selection metrics, methods based on their combinations have also been proposed. Olsson and Oard (2006) combine a few of the popular metrics by taking the maximum, average, or minimum of the scores. It is argued that even a simple combination approach yields results better than the individual metrics. Neumayer, Mayer and Nørvåg (2011) analyze binary combinations of some of the well-known feature selection methods and compare the results with a large number of individual metrics on 18 multi-class datasets. Although the combined metrics give higher performances than the individual ones in most cases, they do not show a consistent behavior and the performances largely depend on the domain.

3. Feature selection metrics

In this section, we explain five feature selection metrics that are well-known in the text categorization domain and are used in this study as well as four new methods that we propose. Section 5 gives a detailed comparison of the proposed methods with the existing ones.

3.1. Existing metrics

Table 1 shows five popular feature selection metrics we use in the analyses (Manning, Raghavan, & Schütze, 2008; Sebastiani, 2002). Note that the scores calculated by these formulae are local scores in the sense that they show the score of a term with respect to a specific class c_i. Let $f(t_k, c_i)$ denote the feature selection score of term t_k specified locally to class c_i and nc is the number of classes. In order to assess the value of t_k in a 'global' sense, either the sum $f_{sum}(t_k) = \sum_{i=1}^{nc} f(t_k, c_i)$, the weighted average $f_{avg}(t_k) = \sum_{i=1}^{nc} P(c_i) f(t_k, c_i)$, or the maximum $f_{\max}(t_k) = \max_{i=1}^{nc} f(t_k, c_i)$ of its category-specific values may be computed. For obtaining the global versions of the local metrics, we use the globalization technique $f_{\max}$ which is claimed to outperform the other techniques (Debole & Sebastiani, 2003).

3.1.1. Information gain (IG)

This metric measures the reduction in the entropy by knowing the presence or absence of a term in a document. It is a very popular term-goodness criterion that is widely-used in the machine learning community.

3.1.2. Chi-square statistics (CHI)

In statistics, the chi-square test is applied to measure the independence of two random variables. In the domain of text categorization, the two random variables are the occurrence of a term t and the occurrence of a class c. It is also used extensively in the text categorization research and in most studies it is claimed to perform comparable to information gain.

3.1.3. Document frequency (DF)

This is a very simple metric which is independent of the class labels. It is based on the assumption that infrequent terms are not reliable and effective in category prediction. It counts the number of documents in which a term appears and selects the terms whose counts are the highest. In spite of its simplicity, it has a performance similar to IG and CHI if the keyword number is not too low.

3.1.4. Accuracy balanced (Acc2)

This is a two-sided metric (it selects both negative and positive features) that is based on the difference of the distributions of a term in the documents belonging to a class and in the documents not belonging to that class. It resembles the $s\chi^2$ (simplified chi-square) metric that was proposed by Galavotti, Sebastiani and Simi (2000) as a simplification of the CHI metric. The difference is that contrary to Acc2, $s\chi^2$ is one-sided (it selects only positive features). In Zheng and Srihari (2003) and Sebastiani (2002), $s\chi^2$ (renamed as GSS coefficient) was studied and claimed to have a performance comparable to IG and CHI while Forman (2003) reports similar results for Acc2.

3.1.5. Term frequency-inverse document frequency (tf-idf)

This is similar to the DF metric in the sense that it is based on the idea that terms which have higher tf-idf scores are more informative for the classification task. It was used by Özgür and Güngör (2007), but was not compared to other metrics. A variant of the tf-idf metric was proposed by How and Kulathuramaiyer (2004) and was claimed to perform comparable to other popular metrics such as IG and CHI.

3.2. Proposed metrics

The proposed metrics are local metrics and each metric is a simple variation of another one taking into account some characteristics of the classification task. The metrics are named as M$_1$, M$_2$, M$_3$, and M$_4$.

3.2.1. $M_{1.}$ Metric

This metric is a more elaborate version of the Acc2 metric. In Acc2, only the number of documents in which a term occurs is

Table 1
Existing feature selection metrics.

Name	Formula
Information gain	$IG(t_k, c_i) = \sum_{c \in \{c_i, \bar{c}_i\}} \sum_{t \in \{t_k, \bar{t}_k\}} P(t, c) \log \frac{P(t, c)}{P(t)P(c)}$
Chi-square	$CHI(t_k, c_i) = N \times \frac{[P(t_k, c_i)P(\bar{t}_k, \bar{c}_i) - P(\bar{t}_k, c_i)P(t_k, \bar{c}_i)]^2}{P(t_k)P(\bar{t}_k)P(c_i)P(\bar{c}_i)}$
Document frequency	$DF(t_k, c_i) = P(t_k, c_i)$
Accuracy balanced	$Acc2(t_k, c_i) = \lvert P(t_k, c_i) - P(t_k, \bar{c}_i) \rvert$
Term frequency-inverse document frequency	$tf - idf(t_k, c_i) = \sum_{d_j \in \{c_i\}} tf(t_k, d_j) \log \frac{D(c_i)}{D(t_k, c_i)}$

Notation:
$P(t_k, c_i)$: percentage of documents belonging to class c_i in which term t_k occurs
$P(\bar{t}_k, c_i)$: percentage of documents belonging to class c_i in which term t_k does not occur
$P(t_k, \bar{c}_i)$: percentage of documents not belonging to class c_i in which term t_k occurs
$P(\bar{t}_k, \bar{c}_i)$: percentage of documents not belonging to class c_i in which term t_k does not occur
$P(t_k)$: percentage of documents in which term t_k occurs
$P(\bar{t}_k)$: percentage of documents in which term t_k does not occur
$P(c_i)$: percentage of documents belonging to class c_i
$P(\bar{c}_i)$: percentage of documents not belonging to class c_i
N: total number of documents in the dataset
$tf(t_k, d_j)$: frequency of term t_k in document d_j
$D(c_i)$: number of documents in class c_i
$D(t_k, c_i)$: number of documents in class c_i in which term t_k occurs

taken into account without considering the number of actual occurrences of the term in the documents. In this method, we multiply the Acc2 score by relative term occurrence frequencies:

$$\mathrm{M}_1(t_k, c_i) = \mathrm{Acc2}(t_k, c_i) \times \left[\frac{A}{t_1} - \frac{B}{t_2}\right]$$

where A and B are the number of occurrences of term t_k in the documents, respectively, belonging to class c_i and not belonging to class c_i; t_1 and t_2 correspond to the number of terms, respectively, in class c_i and in other classes. Note that the M_1 metric is not a simple extension of the Acc2 method from a binary approach (a term either occurs or does not occur in a document) to a frequency approach (number of times a term occurs in a document). Instead, it aims at taking a combination of these two approaches.

3.2.2. $\mathrm{M}_{2.}$ Metric

This metric is also similar to Acc2, but we measure the correlation between a term and a class in a different way. Here, we consider the documents in the whole corpus in which the term appears as a group and we find the proportion of the documents with class label c_i in this group. We also multiply it by the document frequency of term t_k in the whole corpus. Because, without such a modification, a very infrequent and thus uninfluential term can have a similar score as a frequent term that is effective in the classification of many documents, since the document proportions in the group may be similar in both cases. The calculation of the M_2 metric is given by the following formula:

$$\mathrm{M}_2(t_k, c_i) = DF(t_k) \times \left[\frac{C}{d_1} - \frac{D}{d_2}\right]$$

where C and D are the number of documents in category c_i, respectively, in which term t_k occurs and t_k does not occur; d_1 and d_2 correspond to the number of documents in which, respectively, t_k occurs and t_k does not occur; $DF(t_k)$ is the sum of document frequencies of all classes for term t_k .

3.2.3. $\mathrm{M}_{3.}$ Metric

This metric, as in the case of the M_2 metric, integrates a frequency score into the calculation. It is calculated as the multiplication of the M_1 score of a term by the document frequency of the term. Since the M_1 method alone does not take into account the document frequency, it may give similar weights to frequent and rare terms. However, it may be reasonable to prefer the frequent terms since they play a role in the classification of more documents. The formula is given below:

$$\mathrm{M}_3(t_k, c_i) = DF(t_k) \times \mathrm{M}_1(t_k, c_i)$$

3.2.4. $\mathrm{M}_{4.}$ Metric

In the experiments, we observed that despite the fact that the M_1 metric gives very good results for low number of keywords, it is not as good as the global methods when the number of keywords increases. That is, the top keywords determined with respect to the M_1 scores have more discriminative power than the top keywords selected by the other methods. But, as we increase the number of keywords, it seems that the M_1 method cannot select new keywords as good as those in the other methods. For instance, on the Wap dataset, the M_1 method outperforms all other popular metrics up to about 200 keywords; but beyond this point its performance falls behind the performance of others. This phenomenon is related with the fact that a class with a few documents does not include sufficient number of reliable keywords. In other words, there may be only a few keywords occurring several times in such a class and if we select too many keywords most of them will be noise words that have a little or no effect in classification. For handling the deficiency of the M_1 metric with high number of keywords, we select the first n keywords by the M_1 metric, where n is the number of documents in that class. Then we select the remaining keywords from the list of keywords extracted by the global IG metric by scanning the list beginning from the highest scored keyword. Note that the keywords found by IG that have already been selected or that do not occur in the documents in that class are ignored.

3.3. Adaptive keyword selection framework

Different text categorization problems have varying levels of difficulty due to some factors such as class skewness, similarity of classes, very large vocabulary, and insufficient training examples. Especially, when the number of classes increases, the separability of them decreases and therefore more training data are required for successful categorization.

In a multi-class environment, probably the number of training examples for different classes will be unequal. In such imbalanced situations, inevitably rare classes will suffer from the inadequacy of positive training examples for them. It will be difficult for a feature selection metric to find a large number of reasonable keywords for rare classes. Therefore selecting too many features will cause overfitting and reduce the performance in such classes.

In this method, we propose a solution for the issue that different classes in a dataset may require different number of terms for the best performance. In our experiments, we have seen that in datasets where there is a substantial amount of training examples, the performance of the classifier increases as the number of selected keywords increases. However, in datasets with a small training set, the results are better when a small number of keywords is used. Therefore we decided to employ an adaptive framework in such a way that the size of the keyword set is proportional to the size of the training data.

In order to find the best number of keywords for each class, first we divided the classes into groups with respect to the number of documents they contain. Then we carried out several experiments on different datasets to determine the optimal number of keywords for each group. Below is the keyword number selection procedure for a class with respect to the number of documents it contains, where n represents the number of documents in the training set of the class:

$$\text{Use}\begin{cases}\text{100 keywords} & n > 0 \text{ and } n \leqslant 15\\ \text{20 keywords} & n > 15 \text{ and } n \leqslant 30\\ \text{100 keywords} & \text{if } n > 30 \text{ and } n \leqslant 100\\ \text{500 keywords} & n > 100 \text{ and } n \leqslant 200\\ \text{1000 keywords} & n > 200 \text{ and } n \leqslant 500\\ \text{2000 keywords} & n > 500\end{cases}$$

Basically, this framework selects more keywords as the document number in a class increases. It begins with as few as 20 keywords and increases this number up to 2000 keywords. However, as can be seen in the formula above, the situation is different for classes having less than 15 documents. We observed in the experiments that selecting more keywords (for example, 100) for classes that have less than 10–15 training instances improves the results slightly. The reason is that for a class that has such a low number of training documents, a small number of reliable keywords describing the class cannot be determined. We acquire a better classification when we use more keywords, even though some of these keywords do not have much discriminative power.

The AKS framework is a local policy since it processes each class separately according to its size. We tested the AKS framework using local versions of all keyword selection metrics including the proposed ones. As we will see in Section 5, the results of almost all metrics were improved in skewed datasets. Currently the mapping between the class sizes and the keyword numbers is done in an ad hoc manner. Rather than adopting a mapping by intensive experimentation that seems suitable for all the datasets, it seems better to derive these parameters automatically for each dataset separately. This can be done by training the classifier on a development set. We leave this for future work.

4. Experimental settings

In this work, we used SVM as the learning method, since in previous studies it was asserted that SVM is almost always one of the best classifiers in text categorization. It aims at solving binary classification problems by finding a hyperplane in n-dimensional space that separates positive and negative examples with the largest possible margin. In this way, the generalization error on unseen examples is minimized. We used the SVM$^{\text{light}}$ implementation with default parameter settings and a linear kernel (Joachims, 1999).

We performed experiments on seven datasets with different characteristics shown in Table 2. The last column of the table shows the skewness property (homogenous, medium, highly skewed) of each dataset. We measure skewness by dividing the standard deviation of the class distribution by the mean of the distribution, which is an indication of the amount of imbalance with respect to the dataset size. Classic3 dataset is quite easy to categorize and, as shown in the experiments section, can achieve an accuracy over 99%. Hitech, LA1 and Reviews datasets are relatively homogenous and they contain more training instances per category compared to Wap. Wap dataset is a skewed dataset with 20 classes and very few training instances (1047 documents). Reuters-21578 dataset, a standard dataset in text categorization, has 90 classes and 9603 training instances after ModApte splitting is applied.

Finally, we have conducted experiments on RCV1, a rather new benchmark collection for text classification research. For the experiments, we used the whole dataset and applied the LYRL-2004 split (Lewis, Yang, Rose, & Li, 2004). There are just a few works in the literature that employ the whole RCV1 dataset (e.g. Joachims, 2006). To the best of our knowledge, this paper is one of the first works in the text categorization domain which conducts a detailed set of experiments on the whole RCV1 dataset. Therefore, we dedicate a separate subsection (Section 5.3) for the results of the experiments and a discussion of the RCV1 dataset.

In all experiments, we have removed the stopwords according to the stopwords list of the smart system (ftp://ftp.cs.cornell.edu/pub/smart). In addition, non-alphabetic characters were discarded, all letters were converted to lowercase and stemming was applied by means of Porter's stemmer (1997). For term weighting, we used tf-idf weighting with length normalization (Manning et al., 2008).

Table 2
Properties of the datasets used.

Dataset	# of Training Documents	# of Test Documents	# of Classes	# of Terms	Skewness (sd/mean)
Classic3	2699	1192	3	10930	Homogenous (0.13)
Hitech	1530	770	6	18867	Medium (0.45)
LA1	2134	1070	6	25024	Medium (0.45)
Reviews	2708	1361	5	31325	Medium (0.57)
Wap	1047	513	20	8064	Highly skewed (0.96)
Reuters-21578	9603	3299	90	20308	Highly skewed (3.32)
RCV1	23149	781265	103	46487	Highly skewed (2.03)

Table 3
Micro- and macro-averaged F-measures for Hitech dataset.

	10	30	50	100	200	500	1000	1500	2000	All
Micro-F										
tf-idf(l)	0.551	0.610	0.617	0.638	0.624	0.654	0.644	0.618	0.638	0.649
tf-idf(g)	0.372	0.523	0.559	0.621	0.606	0.645	0.649	0.647	0.666	0.649
IG(l)	0.510	0.610	0.617	0.638	0.630	0.654	0.644	0.634	0.638	0.649
IG(g)	0.430	0.523	0.559	0.621	0.641	0.645	0.649	0.658	0.666	0.649
CHI(l)	0.557	0.590	0.620	0.631	0.636	0.636	0.619	0.630	0.632	0.649
CHI(g)	0.485	0.559	0.597	0.621	0.637	0.633	0.651	**0.670**	0.667	0.649
Acc2(l)	0.558	0.612	0.636	0.649	0.637	0.651	0.659	0.647	0.646	0.649
Acc2(g)	0.521	0.581	0.575	0.606	0.607	**0.657**	0.642	0.637	0.661	0.649
DF(l)	0.501	0.550	0.578	0.624	0.613	0.622	0.644	0.664	0.661	0.649
DF(g)	0.214	0.546	0.538	0.583	0.616	0.609	0.624	0.624	0.629	0.649
M_1	**0.573**	**0.625**	0.637	**0.658**	0.657	0.656	**0.666**	0.661	**0.673**	0.649
M_2	0.547	0.617	**0.638**	0.645	0.635	0.645	0.655	0.646	0.645	0.649
M_3	0.555	0.610	0.637	0.638	0.650	0.652	0.652	0.659	0.653	0.649
M_4	0.571	0.623	0.630	0.657	**0.661**	0.648	0.655	0.633	0.629	0.649
Macro-F										
tf-idf(l)	**0.486**	0.529	0.539	0.577	0.564	0.591	0.573	0.549	0.557	0.558
tf-idf(g)	0.228	0.433	0.461	0.538	0.505	0.572	0.597	0.582	0.602	0.558
IG(l)	0.456	0.529	0.539	0.577	0.571	0.591	0.573	0.555	0.557	0.558
IG(g)	0.301	0.433	0.461	0.538	0.558	0.572	0.597	0.601	0.602	0.558
CHI(l)	0.477	0.495	0.536	0.572	0.567	0.551	0.545	0.552	0.567	0.558
CHI(g)	0.340	0.437	0.509	0.528	0.550	0.570	0.610	**0.611**	0.605	0.558
Acc2(l)	0.459	0.522	0.550	0.571	0.564	**0.596**	0.600	0.583	0.593	0.558
Acc2(g)	0.433	0.507	0.496	0.521	0.522	0.567	0.582	0.567	0.603	0.558
DF(l)	0.397	0.485	0.507	0.549	0.540	0.549	0.592	**0.611**	0.603	0.558
DF(g)	0.141	0.389	0.383	0.461	0.527	0.510	0.524	0.526	0.532	0.558
M_1	0.482	**0.553**	**0.578**	0.582	0.572	0.590	**0.615**	0.609	**0.615**	0.558
M_2	0.449	0.527	0.546	0.568	0.555	0.594	0.597	0.578	0.588	0.558
M_3	0.443	0.533	0.563	0.559	0.566	0.585	0.581	0.597	0.586	0.558
M_4	0.472	0.542	0.556	0.594	0.596	0.578	0.600	0.570	0.561	0.558

We have analyzed the results in terms of micro-averaged and macro-averaged F-measures (Manning et al., 2008) at different keyword selection points. The former reflects the overall performance better, while the latter is preferable in measuring the classifier's performance on rare categories since it gives equal weight to all classes regardless of the frequency of the class. We varied the number of keywords from 10 to 2000 and also compared the results with the case where all keywords are considered (no feature selection). We have not carried out experiments with more than 2000 keywords since we observed in our preliminary experiments that F-measures generally reach their maximum values below 2000 keywords and then start to decline.

5. Results and discussion

In this work, we carried out an extensive set of experiments with local and global policies using different number of terms and different feature selection metrics on seven datasets. In this section, we first give the results for three of the datasets. Then, we will make a detailed comparison of the policies, the existing metrics, and the proposed methods. Finally, we will comment on the findings obtained in this research and their significance with respect to the text categorization domain. The F-measure results for the datasets not included in this section can be found in the appendix.

Tables 3–5 show the micro- and macro-averaged F-measure results of the experiments using the old feature selection metrics as well as the proposed ones. In the tables, the local and global versions of the previous metrics are denoted by (l) and (g), respectively. We give the results of three datasets: Wap, Hitech and Reuters. Wap and Hitech are good examples of skewed and homogenous datasets, respectively, and they are included here in order to evaluate and comment on the performance of the metrics under the skewness criterion. Reuters is given for comparison with the previous work since it is one of the most popular datasets in the text categorization community. In addition to the highest score for each keyword number (shown in bold), the change in the performance of each metric as the keyword number increases provides us important information about the behavior of the approaches.

5.1. Comparison of local and global policies

In previous studies, the well-known feature selection metrics were compared with each other extensively, but their local and global versions were not studied and compared in detail. In this study, one of our main objectives is the comparison of feature selection policies.

Fig. 1 shows the micro-averaged F-measure results for the global and local versions of feature selection metrics on the Reuters dataset as a function of the number of keywords. The success rate without feature selection (all words) is also shown as a straight line for comparison. First, we observe that the local policy performs significantly better than the global policy when the keyword number is low. The reason of this behavior is that, in the test phase, local policy tries to identify a document belonging to a class using keywords specific to that class, whereas global policy uses keywords common to all classes. When there is a very few number of keywords, each keyword selected by the global policy serves as a discriminative feature for several classes and thus fails to identify the correct class. For instance, using 50–100 keywords for a dataset of 90 classes like Reuters implies that there will be about one keyword per class on the average. Such a low number does not seem to be sufficient for a correct classification. On the other hand, local policy makes use of several specific keywords for each class and the decision of whether a document belongs to a particular category or not can be made more accurately. The performance of the global policy approaches to that of the local policy as more keywords are included and it performs better after about

Table 4
Micro- and macro-averaged F-measures for Wap dataset.

	10	30	50	100	200	500	1000	1500	2000	All	AKS
Micro-F											
tf-idf(l)	0.671	0.737	0.741	0.738	0.735	0.722	0.746	0.741	0.749	0.752	0.734
tf-idf(g)	0.134	0.496	0.587	0.655	0.691	0.721	0.740	0.749	0.743	0.752	–
IG(l)	**0.685**	0.735	0.750	0.742	0.747	0.744	0.742	0.758	0.749	0.752	0.769
IG(g)	0.399	0.526	0.577	0.644	0.746	0.753	0.755	0.756	0.755	0.752	–
CHI(l)	0.440	0.714	0.732	0.732	0.720	0.736	0.742	0.756	**0.758**	0.752	0.751
CHI(g)	0.242	0.523	0.540	0.607	0.631	0.712	0.730	0.741	0.749	0.752	–
Acc2(l)	0.639	0.728	0.757	0.770	0.758	0.755	0.752	0.758	0.752	0.752	**0.795**
Acc2(g)	0.221	0.476	0.529	0.629	0.697	0.730	0.743	0.753	**0.758**	0.752	–
DF(l)	0.000	0.567	0.704	0.751	**0.771**	0.747	**0.760**	0.747	0.747	0.752	0.777
DF(g)	0.000	0.341	0.395	0.543	0.657	0.723	0.756	0.757	**0.758**	0.752	–
M$_1$	0.667	0.738	0.762	0.777	0.758	0.750	0.747	0.749	0.748	0.752	0.790
M$_2$	0.603	0.732	0.738	0.757	0.765	0.759	0.756	**0.760**	0.753	0.752	0.774
M$_3$	0.610	0.701	0.735	**0.776**	0.769	**0.761**	0.758	0.748	0.750	0.752	0.793
M$_4$	0.665	**0.739**	**0.769**	0.765	0.767	**0.761**	0.755	0.754	0.754	0.752	0.790
Macro-F											
tf-idf(l)	**0.506**	**0.593**	0.565	0.532	0.507	0.495	0.509	0.477	0.483	0.450	0.543
tf-idf(g)	0.093	0.208	0.306	0.350	0.412	0.442	0.455	0.468	0.455	0.450	–
IG(l)	0.492	0.531	0.548	0.517	0.508	0.508	0.460	0.490	0.482	0.450	0.545
IG(g)	0.052	0.185	0.284	0.375	0.479	0.501	0.473	0.474	0.467	0.450	–
CHI(l)	0.493	0.511	0.520	0.509	0.462	0.491	0.475	0.488	0.491	0.450	0.538
CHI(g)	0.121	0.239	0.256	0.336	0.375	0.451	0.486	0.469	0.478	0.450	–
Acc2(l)	0.435	0.554	0.564	0.551	0.509	0.516	0.488	0.491	0.485	0.450	0.574
Acc2(g)	0.117	0.235	0.278	0.411	0.479	0.489	0.480	0.480	**0.492**	0.450	–
DF(l)	0.000	0.353	0.513	0.550	**0.538**	0.483	**0.524**	0.483	0.481	0.450	0.587
DF(g)	0.000	0.053	0.095	0.237	0.326	0.430	0.474	0.470	0.462	0.450	–
M$_1$	0.481	0.559	0.551	**0.594**	0.534	0.520	0.488	0.484	0.482	0.450	**0.630**
M$_2$	0.386	0.562	0.539	0.558	0.524	0.519	0.492	**0.492**	0.486	0.450	0.539
M$_3$	0.376	0.497	0.546	0.577	0.527	**0.522**	0.495	0.478	0.481	0.450	0.590
M$_4$	0.480	0.554	**0.572**	0.558	0.524	0.497	0.471	0.468	0.467	0.450	0.585

Table 5
Micro- and macro-averaged F-measures for Reuters dataset.

	10	30	50	100	200	500	1000	1500	2000	All	AKS
Micro-F											
tf-idf(l)	0.776	0.812	0.831	0.835	0.838	0.845	0.853	0.850	0.855	0.855	0.850
tf-idf(g)	0.367	0.565	0.625	0.694	0.760	0.811	0.843	0.858	0.860	0.855	–
IG(l)	**0.777**	0.820	0.838	0.842	0.845	0.850	0.856	0.858	0.856	0.855	0.858
IG(g)	0.485	0.661	0.705	0.765	0.815	0.849	0.857	**0.862**	0.861	0.855	–
CHI(l)	0.520	**0.823**	**0.840**	0.842	0.839	0.845	0.852	0.855	0.854	0.855	0.853
CHI(g)	0.231	0.367	0.531	0.626	0.742	0.798	0.844	0.856	**0.862**	0.855	–
Acc2(l)	0.773	0.811	0.835	0.846	0.855	0.860	**0.862**	0.859	0.859	0.855	0.863
Acc2(g)	0.352	0.388	0.513	0.622	0.196	0.814	0.832	0.848	0.860	0.855	–
DF(l)	0.725	0.802	0.820	0.841	0.847	0.854	0.859	0.859	0.859	0.855	0.862
DF(g)	0.412	0.542	0.624	0.679	0.753	0.802	0.839	0.854	0.857	0.855	–
M$_1$	0.773	0.817	0.835	**0.854**	0.857	0.858	0.861	**0.862**	**0.862**	0.855	**0.866**
M$_2$	0.762	0.815	0.828	0.847	0.858	0.861	0.861	0.859	0.860	0.855	0.864
M$_3$	0.690	0.803	0.819	0.846	0.856	**0.863**	0.861	0.861	0.860	0.855	0.862
M$_4$	0.773	0.815	0.823	0.852	**0.860**	0.861	0.857	0.859	0.861	0.855	0.861
Macro-F											
tf-idf(l)	0.494	0.512	0.519	0.508	0.514	0.493	0.495	0.491	0.492	0.438	0.516
tf-idf(g)	0.014	0.031	0.044	0.090	0.163	0.262	0.370	0.417	0.432	0.438	–
IG(l)	0.494	0.530	0.512	0.517	0.496	0.495	0.493	**0.496**	0.490	0.438	0.527
IG(g)	0.034	0.099	0.140	0.195	0.321	0.392	0.457	0.490	0.476	0.438	–
CHI(l)	0.466	0.491	0.493	0.500	0.488	0.493	0.493	0.494	0.491	0.438	0.497
CHI(g)	0.051	0.107	0.163	0.242	0.377	0.439	0.476	0.475	0.482	0.438	–
Acc2(l)	0.492	0.525	0.524	0.527	0.515	**0.513**	**0.500**	0.492	0.489	0.438	0.531
Acc2(g)	0.039	0.113	0.145	0.215	0.193	0.484	0.488	0.492	0.490	0.438	–
DF(l)	0.463	0.497	0.515	**0.539**	**0.532**	0.511	**0.500**	0.491	0.493	0.438	**0.538**
DF(g)	0.010	0.034	0.058	0.090	0.147	0.243	0.364	0.411	0.438	0.438	–
M$_1$	**0.507**	0.512	**0.531**	0.529	0.513	0.505	0.496	0.495	**0.494**	0.438	0.535
M$_2$	0.470	**0.531**	0.519	0.529	0.518	**0.513**	0.499	0.493	0.489	0.438	0.531
M$_3$	0.302	0.459	0.495	0.506	0.507	0.506	0.498	0.492	0.489	0.438	0.499
M$_4$	0.484	0.485	0.477	0.491	0.499	0.491	0.472	0.486	0.478	0.438	0.499

1000 keywords. With this number of keywords, it seems that a feature selection procedure common to all classes can capture the differences in different categories.

As in the case of Reuters, a similar behavior related to the local and global policies is observed in the other skewed datasets Wap and RCV1. However, in homogenous datasets, although the local

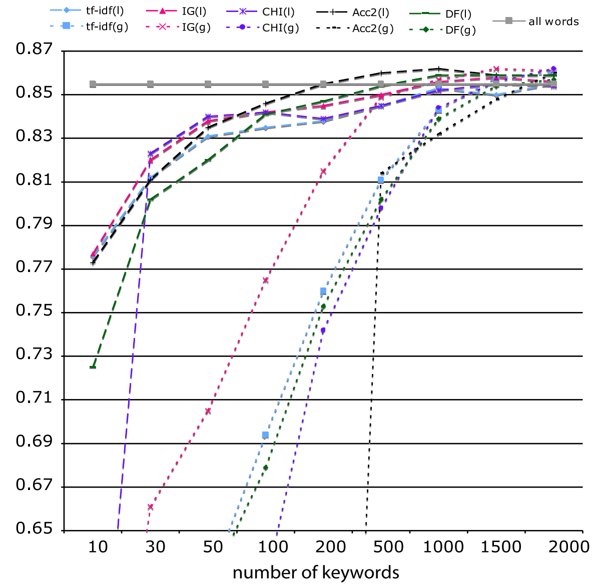

Fig. 1. Micro-averaged F-measures for Reuters dataset.

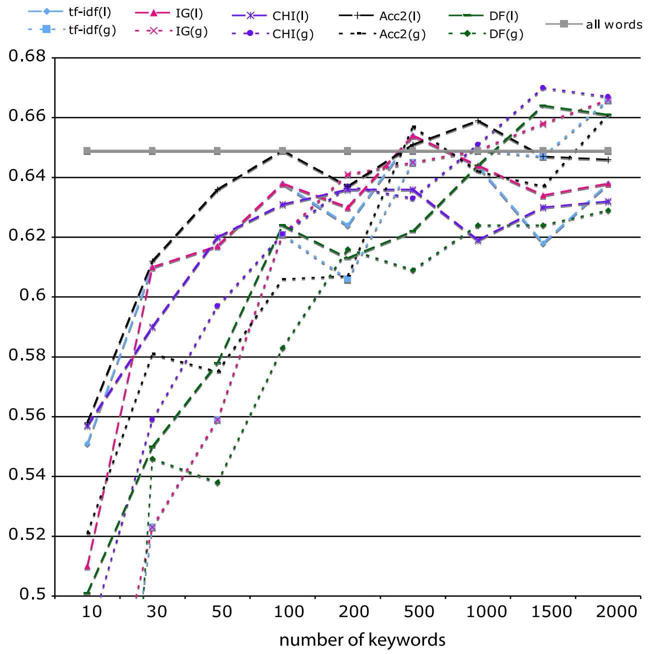

Fig. 2. Micro-averaged F-measures for Hitech dataset.

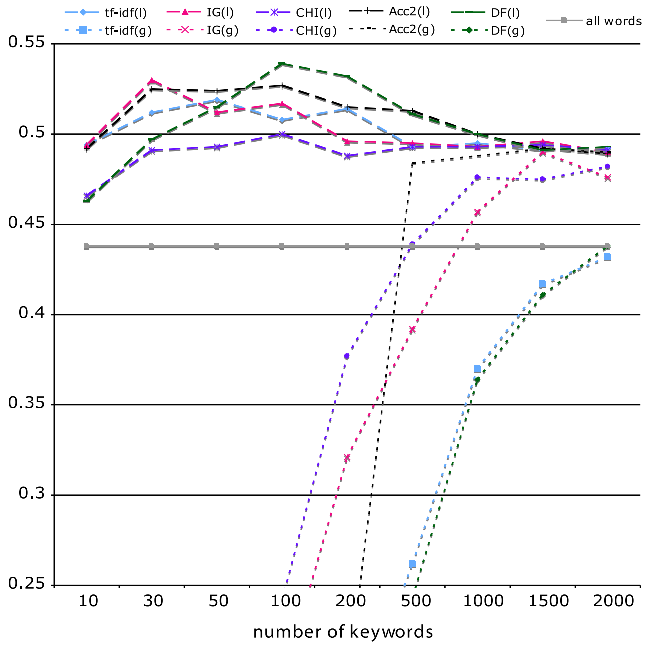

Fig. 3. Macro-averaged F-measures for Reuters dataset.

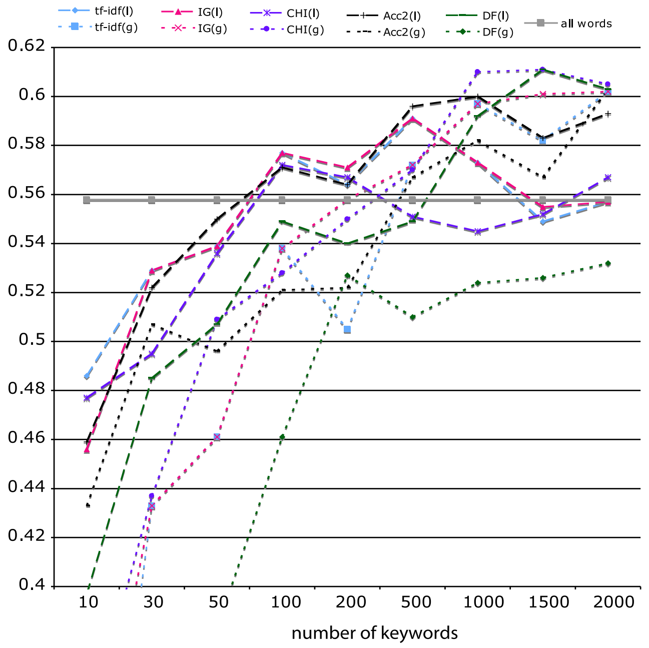

Fig. 4. Macro-averaged F-measures for Hitech dataset.

policy still outperforms the global policy at low keyword numbers, the performance gap between the two policies closes earlier. Fig. 2 shows the micro-averaged results for the Hitech dataset. The performance of the global policy is a little worse than that of the local policy when the number of keywords is low, but for most of the metrics it performs better after 500–1000 keywords. In balanced datasets, all classes will be represented equally well by the selected features and thus a less number of global features will be sufficient for documents belonging to different classes.

When we analyze the macro-averaged F-measure results, we see a similar pattern concerning the relationship between the policies and the feature numbers. Figs. 3 and 4 show the macro-averaged F-measure results for the Reuters and Hitech datasets, respectively. A difference from the micro-averaged success rates is that, in skewed datasets, the superiority of the local policy is more clear in this case. Its performance is almost always higher than that of the global policy and it exceeds the performance of using all words even with 30–50 keywords. This is an interesting result. Using a few class-specific keywords instead of a large number of general keywords classifies the documents belonging to rare classes more accurately. The increase in the classification accuracy of such classes results in better macro-averaged scores.

In order to make a comparison between the feature selection metrics, we show in Table 6 their performances under different environments. The table groups the success rates with respect to two criteria. The first one is the size of the feature set, where feature numbers between 10 and 500 are considered as low keyword

Table 6
Success rates of existing metrics under skewness and feature number criteria.

	Keyword ⩽ 500		Keyword > 500	
	Homogenous	Skewed	Homogenous	Skewed
Micro-averaged F-measure				
Global metrics				
tf-idf	0.758	0.562	0.852	0.793
IG	0.765	0.615	0.856	0.799
CHI	0.760	0.530	0.857	0.791
Acc2	0.781	0.514	0.852	0.799
DF	0.680	0.533	0.845	0.796
Local metrics				
tf-idf	0.786	0.699	0.834	0.791
IG	0.797	0.698	0.850	0.790
CHI	0.793	0.674	0.846	0.792
Acc2	0.804	0.782	0.855	0.807
DF	0.757	0.682	0.846	0.800
Macro-averaged F-measure				
Global metrics				
tf-idf	0.674	0.188	0.819	0.446
IG	0.703	0.227	0.824	0.474
CHI	0.710	0.228	0.825	0.477
Acc2	0.741	0.267	0.819	0.487
DF	0.589	0.150	0.803	0.449
Local metrics				
tf-idf	0.757	0.406	0.799	0.478
IG	0.766	0.440	0.816	0.485
CHI	0.763	0.438	0.812	0.489
Acc2	0.767	0.519	0.825	0.491
DF	0.724	0.443	0.818	0.507

numbers and feature numbers between 1000 and 2000 as high keyword numbers. The second criterion is dataset skewness; we put Reuters, Wap and RCV1 into the group of skewed datasets and the others (Hitech, Reviews, Classic3, LA1) into the homogenous dataset group. All the figures in the table are averages of the success rates of the metrics in the corresponding group. Although taking the average of F-measure scores for different feature numbers in several datasets is not mathematically sound, it provides us an idea about the general performances of the metrics.

In the case of global policy, IG, CHI and Acc2 have similar performances and are the best methods on both homogenous and skewed datasets. When we have a few number of keywords, Acc2 performs slightly better than IG and CHI. It gives the highest micro- and macro-averaged F-measures on homogenous datasets. On skewed datasets, however, while Acc2 is still the best metric in terms of macro-averaged F-measures, IG performs much better than all other metrics in terms of micro-averaged results. This indicates that although the keywords selected by the IG method can classify more documents correctly, they are mostly good discriminators for classes having large number of documents and they fail in identifying the documents in rare classes. On the other hand, the keywords selected by Acc2 give equal chance to both types of classes. As the number of keywords is increased, the performances of all metrics approach to each other. But IG, CHI and Acc2 seem to be a little more successful in this case also.

The superiority of Acc2 over other metrics is emphasized more clearly in local policy. When the keyword number is low, it performs about 10–15% better than the second best metric on skewed datasets and therefore it is a quite successful local classification method for skewed datasets. Acc2 is the best metric for homogenous datasets also, but IG and CHI show similar performances in this case. When we have a large number of keywords, the methods show similar behaviors as in the case of global policy. When we look at the micro-averaged F-measure results, we see that Acc2 is slightly better than the other methods. In macro-averaged scores, DF shows a good performance especially on skewed datasets. This indicates that, for classes containing very few documents, we can achieve high performances by just selecting the high frequency keywords.

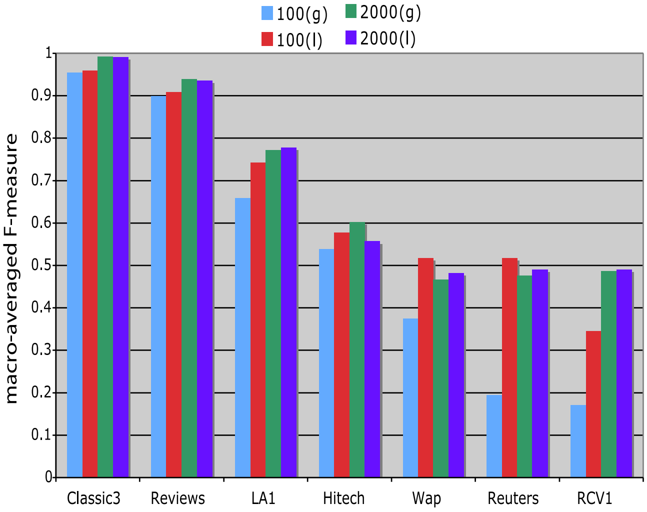

Fig. 5. Comparison of local and global policies (macro-averaged F-measures).

In Fig. 5, we see the macro-averaged F-measure results of IG for 100 and 2000 keywords with local and global policies as the skewness of the datasets increases. The Classic3 dataset is one extreme with only 3 classes while RCV1 is the other extreme with 101 classes. First, the figure shows explicitly that the success rate is inversely proportional to the skewness of the dataset. Second, we observe the relationship between the local and global policies with respect to dataset skewness. With low number of keywords, as the skewness of the dataset increases, the superiority of the local policy over the global policy becomes more apparent. For instance, while the local and global IG performances are similar in the Classic3 dataset for 100 keywords (95.9% and 95.5%, respectively), the local IG value is 2.65 times higher than the global IG value in Reuters for the same keyword number (51.7% and 19.5%, respectively). As discussed above, the reason of this behavior is the insufficiency of a small set of global features. However, the difference between the two policies begins to disappear as we increase the number of features and they perform alike at 2000 keywords.

5.2. Analysis of the proposed metrics and adaptive keyword selection

The success of local policy at low keyword numbers motivated us to concentrate on local feature selection metrics. All of the methods proposed in this paper are local methods. They show slightly different success patterns in different situations depending on the environmental parameters such as the dataset, feature numbers, etc. For instance, M_3 seems to be the best metric on the Reviews dataset while M_1 is the best one for the Hitech dataset. Likewise, M_1 is more successful than M_2 at low keyword numbers for skewed datasets, but it is not so successful when more keywords are selected.

Figs. 6 and 7 show, respectively, the micro- and macro-averaged F-measures for the Wap dataset. In the figures, we compare the new metrics with Acc2(l). We have chosen Acc2 as a representative of the old metrics since it exhibits the best performance as explained in Section 5.1.

The best micro- and macro-averaged F-measure results for the M_1 method both occur with 100 keywords on the Wap dataset (77.7% and 59.4%, respectively). The situation is similar for the M_3 method: 77.6% micro- and 57.7% macro-averaged F-measures with 100 keywords. We observe a significant improvement when compared with the local versions of the Acc2 metric. Acc2 reaches

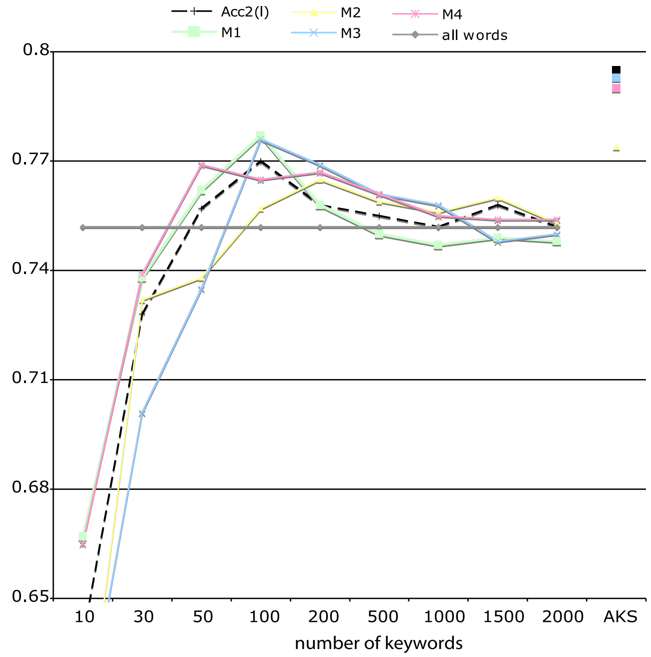

Fig. 6. Micro-averaged F-measures for Wap dataset.

at most 77.0% micro- (100 keywords) and 56.4% macro-averaged (50 keywords) F-measures. M_1 and M_3 give their best results for both scores with the same feature numbers. This is not the case for Acc2 and when we fix the number of keywords, it shows a worse performance. For instance, the macro-averaged score with 100 keywords is just 55.1% and the micro-averaged score with 50 keywords is 75.7%.

On the Hitech dataset (Table 3), for instance, the M_1 method reaches 67.3% micro- (2000 keywords) and 61.5% macro-averaged (1000 and 2000 keywords) F-measures, while Acc2 can achieve at most 65.9% micro- and 60.0% macro-averaged F-measures (both with 1000 keywords). Again, the performance gap between the old and new metrics enlarges as the number of keywords decreases. The situation is similar on Reuters (Table 5) and all the metrics except M_3 perform significantly better than Acc2 for keyword numbers between 100 and 500.

A comparison of the new metrics with the existing metrics other than Acc2 reveals the performance gain more clearly. For instance, on the Wap dataset, all of the new metrics outperform IG(l) and IG(g) when the keyword number is between 100 and 1000. This success on a skewed dataset indicates that the new metrics are very good at finding the best features even when a class does not have too many training documents. The situation is similar for the more homogenous Hitech dataset. The proposed metrics are more successful than CHI(l) for low keyword numbers and than CHI(g) for high keyword numbers. CHI(g) shows the best performance with a high keyword number (1500 keywords), which is a characteristic of a global method, but it is even outperformed by the local M_1 metric at 2000 keywords.

Table 7 lists the micro- and macro-averaged F-measure scores with respect to the dataset skewness and keyword number criteria. As before, we compare the new metrics with the most successful one of the existing metrics (Acc2(l)). M_1 and M_4 seem to be the best methods in terms of both micro- and macro-averaged scores when the number of keywords is low. As the keyword number increases, again the success rates of different metrics approach to each other. With high number of keywords, M_4 shows a poor performance and falls behind Acc2(l). However, M_1 obtains quite high micro- and macro-averaged scores for homogenous datasets and proves to be the best metric. For skewed datasets with this number of keywords, none of the methods outperform Acc2(l) and all have performances similar to Acc2(l). We should note that although M_4 can be regarded as a successful method in general, it has a weakness about the macro-averaged score on skewed datasets. This shows that it cannot handle rare classes as successfully as other methods. As a result, we evaluate the new methods as successful; in most cases they cause an improvement on the performance and in the others they are at least as good as the existing methods regardless of the dataset.

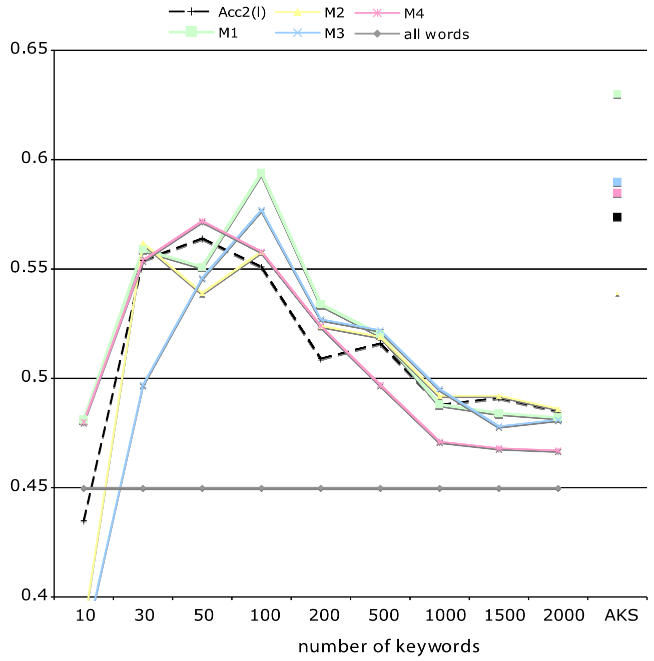

Fig. 7. Macro-averaged F-measures for Wap dataset.

Table 7
Success rates of proposed metrics under skewness and feature number criteria.

	Keyword ⩽ 500		Keyword > 500	
	Homogenous	Skewed	Homogenous	Skewed
Micro-averaged F-measure				
Acc2 (l)	0.804	0.782	0.855	0.807
M$_1$	0.807	0.787	0.857	0.805
M$_2$	0.796	0.777	0.854	0.808
M$_3$	0.802	0.769	0.855	0.806
M$_4$	0.806	0.788	0.851	0.807
Macro-averaged F-measure				
Acc2 (l)	0.767	0.519	0.825	0.491
M$_1$	0.772	0.528	0.828	0.490
M$_2$	0.759	0.514	0.822	0.492
M$_3$	0.762	0.485	0.822	0.489
M$_4$	0.770	0.509	0.819	0.474

A remarkable property of the proposed methods is that they reach their maximum values at about 100 keywords on skewed datasets. This is an important property since it indicates that performance ratios similar as or better than those that can only be obtained using a large number of keywords with the existing methods can be achieved with much less keywords and thus in much less time. This indicates that the methods can determine a small but discriminative set of features for most of the classes. However, when the number of keywords increases, most classes suffer overfitting. Likewise, on homogenous datasets, they acquire quite high scores with 100 keywords, but they preserve these success rates on these datasets as the keyword number increases.

We employed the adaptive keyword selection framework on the skewed datasets Wap and Reuters only, since its logic is based on large differences of document numbers in the classes. The rightmost points in Figs. 6 and 7 show the success rates of the AKS method on the Wap dataset. In order to allow for a general comparison of AKS with all other methods, we give in Table 8 the average performances for low and high keyword numbers of the existing and new methods and the performance of AKS. When we look at the results, we see that AKS improves the performance of almost all local keyword selection metrics on both datasets. The performance gain on the Wap dataset is more clear, while on the Reuters dataset there is a slight improvement.

To measure the significance of the results, we compared the AKS method with a fixed keyword number which usually shows the best performance on each dataset. We have chosen the keyword numbers 100 for Wap and 1000 for Reuters, since the local metrics yield the best results (taking into account both micro- and macro-averaged F-measures) under these keyword numbers. So, for instance, for the Wap dataset, we compared the classification of AKS with the classification obtained using 100 keywords for each of the nine local metrics. On the Wap dataset, the improvement obtained with the AKS version of each metric except tf-idf and CHI is statistically significant. For Reuters, all the metrics except tf-idf perform better under the AKS framework, but only the result of M$_1$ is statistically significant.

We also note that although the success rate of AKS is not always higher than the best success rate (among all feature numbers) for some of the metrics on Reuters, there is usually an insignificant difference in such cases. For instance, the highest micro-averaged score for CHI(l) is 85.5% obtained with 1500 keywords (or, with all the words), while it is 85.3% when the keyword numbers depend on the size of the classes. However, the most important result about the AKS method is that its performance is always better than or similar as the best performance for each metric. This indicates that provided that we can determine the optimal number of keywords for each class, AKS seems to be a very valuable tool for skewed datasets.

Another desirable property of AKS is that it gives rise to high performances in both the micro- and macro-averaged F-measures simultaneously. This is particularly important since no other method has proved to be the best in both of the F-measures at the same time. For instance, if we consider Acc2(l) at 1000 keywords for the Reuters dataset, the micro-averaged F-measure is quite high (86.2%) but the macro-averaged F-measure is only 50.0%. On the other hand, if we select 100 keywords, the macro-averaged F-measure increases to 52.7% but the micro-averaged F-measure decreases to 84.6%. When we use adaptive keyword selection, Acc2(l) micro- and macro-averaged F-measures reach to their highest values (86.3% and 53.1%, respectively). This situation is a consequence of the success of the AKS framework in classifying both rare and common classes correctly.

5.3. Evaluation of the RCV1 dataset

RCV1 (Reuters Corpus Volume 1) is an archive of over 800,000 manually categorized newswire stories made available by Reuters for research purposes (Lewis et al., 2004). It consists of English language stories produced by Reuters journalists over a period of one year (from August 20, 1996 to August 19, 1997). The documents in this collection vary from a few hundred to several thousand words in length. The difference of this corpus from the popular Reuters-21578 dataset is that RCV1 is much larger, containing about 35 times as many documents as the Reuters-21578 collection.

In this study, we carried out several experiments on the RCV1 dataset with all the existing and proposed feature selection metrics. We used the entire corpus in all of the experiments in order to observe the behavioral patterns of the dataset as the metric and the keyword number vary. To the best of our knowledge, this is one of the first works in the text categorization domain which conducts a detailed set of experiments on the whole RCV1 dataset.

Figs. 8 and 9 show the micro- and macro-averaged F-measures, respectively. In these figures, the most striking observation is the success of the DF metric with the local policy. It is always more successful than the other metrics up to 2000 keywords. Furthermore, it is the only metric that has significantly higher macro-averaged F-measure results than using all words and it outperforms the all words strategy for all keyword numbers past 100 keywords. We ascribe the surprising success of DF(l) in RCV1 to the huge size and large dimensionality of the dataset. The other metrics give precedence to discriminative but rare words which in turn causes many documents to be represented by only a few dimensions or by no dimensions after keyword selection is applied. So, the classification is biased towards these very few keywords, weakening its reliability. On the other hand, DF chooses the most common words; so that the majority of the documents in the corpus are well-represented even after keyword selection. The selection of common words does not bring about a serious adverse effect since the useless ones (stopwords) have already been eliminated in the earlier steps.

Similar to the results on the other datasets, the success rates on the RCV1 dataset also affirm the success of the local policy at low keyword numbers. We see that all of the local policies have higher F-measures when the number of keywords is less than 100–200. However, the success rates of the local policy at low keyword numbers are not as high as those in other skewed datasets. We can reach the best performances with just 30–50 keywords on Wap and Reuters, and those performances are preserved or even they decrease beyond this point. On the other hand, on RCV1, the F-measure scores begin with low values and they steadily increase as we increase the feature number.

Finally, if we compare the results of the RCV1 dataset with the results of Reuters-21578, when we use all words, we see that the micro-averaged F-measure of RCV1 is lower (79.7% vs. 85.5%). This

Table 8
Comparison of AKS framework with fixed feature number policy.

	Wap			Reuters		
	Keyword ⩽ 500	Keyword > 500	AKS	Keyword ⩽ 500	Keyword > 500	AKS
Micro-averaged F-measure						
tf-idf	0.724	0.745	0.734	0.823	0.853	0.850
IG	0.734	0.750	0.769	0.829	0.857	0.858
CHI	0.679	0.752	0.751	0.785	0.854	0.853
Acc2	0.735	0.754	0.795	0.830	0.860	0.863
DF	0.590	0.751	0.777	0.815	0.859	0.862
M_1	0.742	0.748	0.790	0.832	0.862	0.866
M_2	0.726	0.756	0.774	0.829	0.860	0.864
M_3	0.725	0.752	0.793	0.813	0.861	0.862
M_4	0.744	0.754	0.790	0.831	0.859	0.861
Macro-averaged F-measure						
tf-idf	0.533	0.490	0.543	0.507	0.493	0.516
IG	0.517	0.477	0.545	0.507	0.493	0.527
CHI	0.498	0.485	0.538	0.489	0.493	0.497
Acc2	0.522	0.488	0.574	0.516	0.494	0.531
DF	0.406	0.496	0.587	0.510	0.495	0.538
M_1	0.540	0.485	0.630	0.516	0.495	0.535
M_2	0.515	0.490	0.539	0.513	0.494	0.531
M_3	0.508	0.485	0.590	0.463	0.493	0.499
M_4	0.531	0.469	0.585	0.488	0.479	0.499

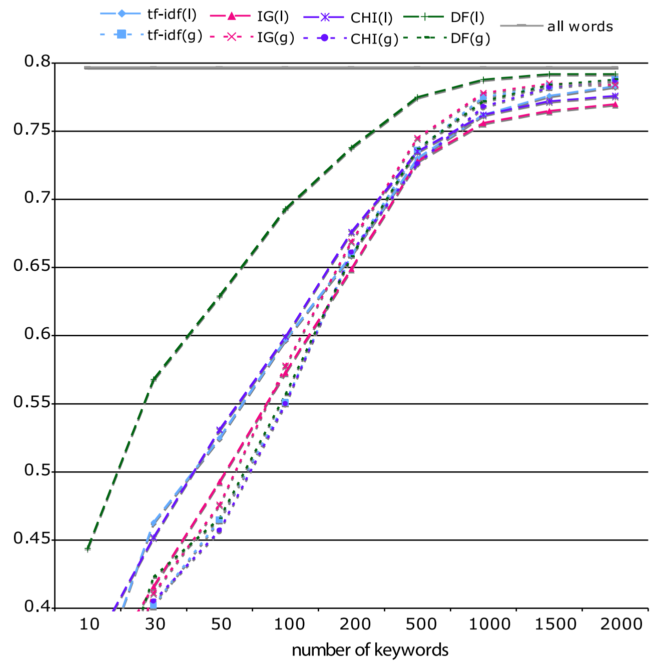

Fig. 8. Micro-averaged F-measures for RCV1 dataset.

situation may be explained by the high class number in the RCV1 dataset. On the other hand, the macro-averaged F-measure is surprisingly higher (47.9% vs. 43.8%). We believe that the reason is the large number of training examples for most of the classes. Obviously, having many training instances for all categories is a valuable quality for a dataset. For example, 27 classes in the Reuters dataset have less than 10 training documents whereas only 5 classes in RCV1 suffer the same problem.

5.4. Summary of the results

In this section, we summarize the results obtained and comment on their implications for a text categorization task.

An obvious observation is that success rate, especially the macro-averaged F-measure, is inversely proportional to dataset skewness (Forman, 2004). Identifying keywords with high discriminative power is difficult for rare classes due to insufficient number of examples. This causes a negative effect on the macro-averaged score which measures the performance on class basis.

Among the existing metrics analyzed in this study, Acc2 seems to be the best one, especially when there is a few number of features. This is more apparent for the local policy of Acc2 on skewed datasets. IG and CHI, which are probably the mostly studied feature selection methods in text categorization, result in scores lower than Acc2 in most cases. However, when we have a large number of features, these three methods show similar behavior, although

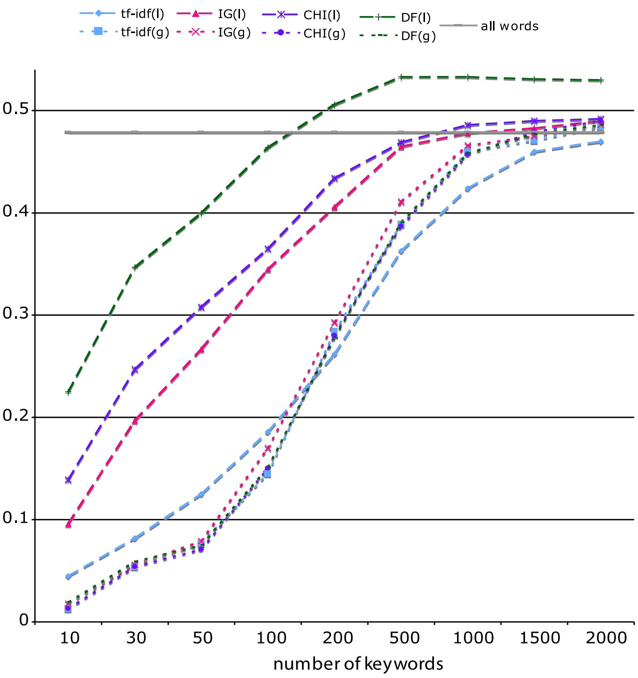

Fig. 9. Macro-averaged F-measures for RCV1 dataset.

Acc2 still yields the best performance with a small margin. The other two methods, tf-idf and DF, seem to give the worst results, although tf-idf performs comparable to IG and CHI on some datasets.

For all classification methods, local versions outperform the global versions when the keyword number is low. The performance improvement is more significant for skewed datasets and in macro-averaged scores. When the keyword number is increased, local and global policies have similar performances, although the local version of a metric is still slightly better than the global version for skewed datasets.

With low number of keywords two of the proposed metrics, M_1 and M_4, outperform Acc2, while with high number of keywords one of them, M_1, outperforms Acc2. Thus, we observe that the new method M_1 is more successful than Acc2 which is one of the best classification methods studied in the literature. M_4 has a deficiency on skewed datasets in terms of macro-averaged scores, which implies that it should be used with care on such datasets when the class-based performance is important.

The other new metrics M_2 and M_3 are not as successful as Acc2. On the other hand, when we compare them with other existing methods, in most cases they give more successful results and in other cases they show similar performances. Thus, we can say that all of the proposed methods are in general more successful than the widely-used existing methods. Another property of the new methods is that, especially on skewed datasets, they reach their top performances at about 100 features.

An important observation is the high success rate of the AKS version of local methods. For all methods including the old and new ones, making the number of features dependent on the class size causes a significant increase in performance compared to the policy of adopting a fixed number of keywords for all classes. The AKS framework eliminates the disadvantages of using a fixed keyword number: the keywords do not carry sufficient information to discriminate between the classes when there is a few number of keywords and the common keywords selected for different classes have a negative effect on the classification task when there is a large number of keywords. By eliminating these effects, AKS increases both the document-based (micro-averaged) and class-based (macro-averaged) success rates. To the best of our knowledge, this is the first work that varies the number of features with respect to the class size in the text categorization domain.

Another desirable property of the AKS method is that it eliminates the need to determine a suitable keyword number for obtaining high scores in both micro- and macro-averaged F-measures. In other methods, usually the best micro- and macro-averaged F-measure scores occur at different keyword numbers. However, AKS yields results more successful than the other local methods in both of these measures.

The RCV1 dataset shows the best performance under the local DF metric. The superiority of this simple method over other well-known and successful methods on this dataset is an interesting result and we leave a detailed analysis of this behavior for future work. Another observation is that, although local methods are still more successful than global methods for low feature numbers, this difference is not as striking as in other datasets.

6. Conclusions and future research

In this work, we made an extensive study of feature selection policies in text categorization with SVM-based classification. We compared the local and global versions of some of the well-known feature selection metrics by varying the number of selected features from 10 to 2000. In the experiments, we used several datasets with different class skewness, size and complexity. We also introduced some new feature selection metrics that are better than or at least as good as the well-known metrics in all the datasets. The new metrics have shown high performances especially when the keyword number is low (for example, 100–200 keywords). This makes them invaluable when the practitioner is constrained to use a small number of keywords.

Table A.1
Micro- and macro-averaged F-measures for Reviews dataset.

	10	30	50	100	200	500	1000	1500	2000	All
Micro-F										
tf-idf(l)	0.842	0.865	0.889	0.900	0.906	0.918	0.926	0.924	0.920	0.941
tf-idf(g)	0.790	0.869	0.869	0.894	0.935	**0.944**	**0.943**	0.937	0.936	0.941
IG(l)	0.850	0.884	0.900	0.909	0.926	0.930	0.936	0.940	0.942	0.941
IG(g)	0.816	0.897	0.904	0.909	**0.937**	0.943	0.940	0.940	0.941	0.941
CHI(l)	0.828	0.877	0.901	0.907	0.919	0.921	0.928	0.933	0.933	0.941
CHI(g)	0.736	0.896	0.912	**0.923**	0.930	0.940	0.941	**0.944**	0.940	0.941
Acc2(l)	0.842	0.900	0.905	0.917	0.927	0.938	0.942	0.940	0.938	0.941
Acc2(g)	0.829	0.899	**0.921**	0.919	0.930	0.940	0.941	**0.944**	**0.944**	0.941
DF(l)	0.805	0.852	0.868	0.895	0.906	0.918	0.928	0.930	0.933	0.941
DF(g)	0.468	0.791	0.813	0.852	0.897	0.930	0.935	0.933	0.939	0.941
M_1	0.844	0.888	0.902	0.921	0.922	0.934	0.938	0.939	0.936	0.941
M_2	0.850	**0.902**	0.903	0.916	0.923	0.940	0.942	0.940	0.937	0.941
M_3	**0.869**	0.895	0.903	0.914	0.924	0.940	0.941	**0.944**	0.940	0.941
M_4	0.844	0.888	0.902	0.921	0.922	0.936	0.940	0.942	0.940	0.941
Macro-F										
tf-idf(l)	0.847	0.863	0.890	0.904	0.904	0.916	0.916	0.912	0.906	0.928
tf-idf(g)	0.567	0.697	0.693	0.720	0.931	**0.939**	0.939	0.935	0.932	0.928
IG(l)	0.860	0.892	0.886	0.908	0.928	0.928	0.930	0.934	0.936	0.928
IG(g)	0.655	0.871	0.867	0.899	**0.933**	0.938	0.937	0.935	0.939	0.928
CHI(l)	0.841	0.881	0.886	0.905	0.916	0.916	0.919	0.923	0.926	0.928
CHI(g)	0.664	0.905	0.915	**0.923**	0.928	0.937	0.933	0.937	0.935	0.928
Acc2(l)	0.847	0.901	0.905	0.919	0.930	0.935	**0.940**	0.939	0.935	0.928
Acc2(g)	0.840	**0.908**	**0.923**	0.922	0.931	**0.939**	0.939	**0.942**	**0.941**	0.928
DF(l)	0.819	0.869	0.880	0.905	0.909	0.920	0.929	0.931	0.933	0.928
DF(g)	0.295	0.567	0.605	0.678	0.731	0.927	0.935	0.934	0.936	0.928
M_1	0.847	0.888	0.901	0.922	0.925	0.933	0.935	0.935	0.929	0.928
M_2	0.854	0.904	0.903	0.915	0.924	0.937	**0.940**	0.936	0.934	0.928
M_3	**0.866**	0.892	0.907	0.918	0.925	0.935	0.938	0.940	0.935	0.928
M_4	0.847	0.888	0.901	0.920	0.921	0.932	**0.940**	0.939	0.936	0.928

Table A.2
Micro- and macro-averaged F-measures for LA1 dataset.

	10	30	50	100	200	500	1000	1500	2000	All
Micro-F										
tf-idf(l)	0.631	0.731	0.761	0.785	0.789	0.807	0.814	0.812	0.815	0.841
tf-idf(g)	0.465	0.648	0.722	0.767	0.793	0.816	0.817	0.825	0.833	0.841
IG(l)	0.660	0.739	0.765	0.793	0.807	0.830	0.831	0.831	0.833	0.841
IG(g)	0.388	0.664	0.724	0.769	0.804	0.828	0.829	0.833	0.838	0.841
CHI(l)	**0.671**	0.736	0.761	0.788	0.813	0.823	0.833	**0.840**	0.838	0.841
CHI(g)	0.340	0.635	0.663	0.745	0.789	0.822	0.828	0.824	0.838	0.841
Acc2(l)	0.659	0.742	0.764	0.802	**0.817**	0.829	0.835	**0.840**	0.840	0.841
Acc2(g)	0.478	0.687	0.758	0.789	0.812	0.831	0.829	0.830	0.829	0.841
DF(l)	0.318	0.688	0.740	0.766	0.782	0.814	0.815	0.827	0.826	0.841
DF(g)	0.103	0.397	0.642	0.709	0.762	0.799	0.821	0.832	0.827	0.841
M_1	0.669	0.735	**0.772**	0.800	0.813	**0.832**	0.830	0.833	**0.841**	0.841
M_2	0.554	0.730	0.760	**0.804**	0.813	0.831	0.833	0.837	**0.841**	0.841
M_3	0.608	**0.743**	0.761	0.798	0.814	0.826	0.835	0.835	0.835	0.841
M_4	0.669	0.735	**0.772**	0.800	0.811	0.827	**0.836**	0.833	0.836	0.841
Macro-F										
tf-idf(l)	0.552	0.674	0.706	0.728	0.735	0.755	0.762	0.756	0.764	0.777
tf-idf(g)	0.284	0.528	0.628	0.692	0.715	0.752	0.748	0.753	0.765	0.777
IG(l)	0.578	**0.688**	0.714	0.743	0.756	**0.781**	**0.779**	0.775	0.777	0.777
IG(g)	0.301	0.510	0.603	0.658	0.745	0.771	0.764	0.762	0.772	0.777
CHI(l)	**0.607**	0.686	**0.715**	0.741	0.766	0.772	0.778	**0.788**	**0.785**	0.777
CHI(g)	0.318	0.523	0.549	0.651	0.722	0.762	0.765	0.757	0.775	0.777
Acc2(l)	0.546	0.682	0.712	**0.759**	**0.773**	0.770	0.774	0.776	0.781	0.777
Acc2(g)	0.387	0.584	0.677	0.732	0.754	0.769	0.758	0.768	0.759	0.777
DF(l)	0.376	0.582	0.665	0.702	0.715	0.755	0.758	0.767	0.768	0.777
DF(g)	0.117	0.227	0.515	0.588	0.688	0.724	0.756	0.766	0.760	0.777
M_1	0.590	0.680	**0.715**	0.749	0.760	0.773	0.774	0.773	0.782	0.777
M_2	0.455	0.662	0.707	0.750	0.770	0.774	0.771	0.775	0.776	0.777
M_3	0.472	0.675	0.698	0.745	0.763	0.762	0.772	0.770	0.767	0.777
M_4	0.590	0.680	**0.715**	0.749	0.749	0.762	0.771	0.764	0.770	0.777

Another contribution of this study is a new feature selection framework called adaptive keyword selection (AKS) which selects different number of terms for classes that have different sizes. It has shown significant improvements with skewed datasets that

Table A.3
Micro- and macro-averaged F-measures for Classic3 dataset.

	10	30	50	100	200	500	1000	1500	2000	All
Micro-F										
tf-idf(l)	0.653	0.895	0.939	0.951	0.959	0.960	0.964	0.965	0.971	0.994
tf-idf(g)	0.701	0.873	0.901	0.937	0.956	0.981	0.988	**0.992**	**0.992**	0.994
IG(l)	0.735	0.896	0.918	0.958	0.973	0.986	0.989	**0.992**	0.991	0.994
IG(g)	0.702	0.848	0.886	0.956	0.974	0.988	**0.991**	0.990	**0.992**	0.994
CHI(l)	0.638	**0.915**	**0.947**	**0.963**	0.974	0.981	0.987	0.989	0.990	0.994
CHI(g)	0.732	0.848	0.890	0.956	0.972	**0.989**	**0.991**	**0.992**	**0.992**	0.994
Acc2(l)	0.787	0.880	0.926	0.958	0.972	0.985	**0.991**	0.991	0.991	0.994
Acc2(g)	0.736	0.867	0.916	0.944	0.967	0.984	0.988	0.989	0.991	0.994
DF(l)	0.745	0.865	0.883	0.917	0.949	0.964	0.973	0.973	0.978	0.994
DF(g)	0.622	0.800	0.833	0.894	0.943	0.970	0.986	**0.992**	**0.992**	0.994
M$_1$	**0.789**	0.892	0.934	0.956	**0.976**	0.984	0.989	0.989	0.990	0.994
M$_2$	0.743	0.881	0.920	0.955	0.972	0.984	**0.991**	0.991	0.991	0.994
M$_3$	0.766	0.899	0.930	0.955	0.973	0.984	0.989	0.990	**0.992**	0.994
M$_4$	**0.789**	0.892	0.934	0.956	**0.976**	0.983	0.990	0.990	**0.992**	0.994
Macro-F										
tf-idf(l)	0.720	0.880	0.935	0.950	0.957	0.959	0.964	0.964	0.970	0.994
tf-idf(g)	0.665	0.871	0.898	0.936	0.953	0.980	0.989	0.992	**0.992**	0.994
IG(l)	0.728	0.889	0.912	0.959	0.974	0.986	0.990	0.992	0.991	0.994
IG(g)	0.665	0.811	0.863	0.955	0.975	0.988	**0.991**	0.990	**0.992**	0.994
CHI(l)	0.706	**0.908**	**0.945**	**0.963**	0.974	0.981	0.987	0.989	0.990	0.994
CHI(g)	0.709	0.821	0.870	0.956	0.972	**0.990**	**0.991**	**0.993**	**0.992**	0.994
Acc2(l)	**0.761**	0.867	0.923	0.958	0.972	0.985	**0.991**	0.991	0.991	0.994
Acc2(g)	0.690	0.865	0.914	0.944	0.967	0.985	0.988	0.990	0.991	0.994
DF(l)	0.720	0.848	0.871	0.908	0.945	0.964	0.973	0.973	0.978	0.994
DF(g)	0.623	0.798	0.831	0.893	0.941	0.970	0.987	0.992	**0.992**	0.994
M$_1$	0.756	0.877	0.928	0.956	0.976	0.985	0.990	0.990	0.991	0.994
M$_2$	0.723	0.868	0.918	0.954	0.973	0.984	**0.991**	0.991	0.991	0.994
M$_3$	0.737	0.896	0.928	0.955	0.974	0.984	0.990	0.990	**0.992**	0.994
M$_4$	0.756	0.877	0.928	0.956	**0.977**	0.984	0.990	0.991	**0.992**	0.994

Table A.4
Micro- and macro-averaged F-measures for RCV1 dataset.

	10	30	50	100	200	500	1000	1500	2000	All
Micro-F										
tf-idf(l)	0.329	0.463	0.525	0.597	0.659	0.730	0.762	0.776	0.783	0.797
tf-idf(g)	0.191	0.402	0.465	0.551	0.660	0.736	0.775	0.783	0.788	0.797
IG(l)	0.338	0.416	0.493	0.573	0.649	0.728	0.756	0.765	0.770	0.797
IG(g)	0.274	0.411	0.476	0.578	0.669	0.745	0.778	0.785	0.784	0.797
CHI(l)	0.364	0.452	0.531	0.599	0.676	0.735	0.762	0.772	0.776	0.797
CHI(g)	0.192	0.405	0.457	0.550	0.661	0.726	0.768	0.782	0.787	0.797
Acc2(l)	0.283	0.381	0.476	0.567	0.668	0.742	0.772	0.782	0.789	0.797
Acc2(g)	0.260	0.330	0.387	0.513	0.648	0.720	0.769	0.779	0.785	0.797
DF(l)	**0.444**	**0.568**	**0.629**	**0.693**	**0.738**	**0.775**	**0.788**	**0.792**	**0.792**	0.797
DF(g)	0.281	0.423	0.466	0.557	0.658	0.737	0.772	0.784	0.788	0.797
M$_1$	0.257	0.381	0.476	0.567	0.668	0.742	0.772	0.783	0.789	0.797
M$_2$	0.268	0.391	0.471	0.575	0.670	0.745	0.773	0.784	0.789	0.797
M$_3$	0.203	0.382	0.445	0.549	0.660	0.741	0.776	0.786	0.791	0.797
M$_4$	0.259	0.383	0.474	0.570	0.669	0.743	0.776	0.786	0.788	0.797
Macro-F										
tf-idf(l)	0.045	0.082	0.125	0.186	0.262	0.363	0.424	0.460	0.470	0.479
tf-idf(g)	0.013	0.054	0.077	0.145	0.284	0.389	0.460	0.471	0.484	0.479
IG(l)	0.096	0.197	0.267	0.345	0.406	0.465	0.478	0.483	0.490	0.479
IG(g)	0.018	0.056	0.079	0.170	0.293	0.411	0.466	0.475	0.486	0.479
CHI(l)	0.139	0.247	0.308	0.365	0.434	0.469	0.486	0.490	0.492	0.479
CHI(g)	0.013	0.054	0.071	0.150	0.280	0.387	0.457	0.479	0.490	0.479
Acc2(l)	0.103	0.147	0.220	0.310	0.402	0.478	0.498	0.505	0.507	0.479
Acc2(g)	0.017	0.027	0.050	0.134	0.333	0.426	0.484	0.499	0.505	0.479
DF(l)	**0.225**	**0.347**	**0.400**	**0.464**	**0.506**	**0.533**	**0.533**	**0.531**	**0.530**	0.479
DF(g)	0.019	0.059	0.075	0.151	0.278	0.391	0.458	0.477	0.486	0.479
M$_1$	0.072	0.147	0.220	0.310	0.402	0.478	0.498	0.505	0.507	0.479
M$_2$	0.018	0.042	0.149	0.295	0.401	0.481	0.501	0.508	0.508	0.479
M$_3$	0.011	0.035	0.077	0.201	0.342	0.450	0.490	0.500	0.508	0.479
M$_4$	0.056	0.128	0.198	0.264	0.377	0.455	0.488	0.500	0.498	0.479

have a limited number of training instances. In addition, it gives us the opportunity to get very high micro- and macro-averaged F-measures simultaneously.

Future work includes the evaluation and comparison of the feature selection policies with newer term weighting approaches such as supervised term weighting (Debole & Sebastiani, 2003; Soucy and Mineau, 2005) and with learning algorithms apart from support vector machines. In addition, we plan to work on adjusting the parameters of the AKS framework automatically depending on the characteristics of each dataset.

Acknowledgment

This work was supported by the Boğaziçi University Research Fund under the Grant number 05A103.

Appendix A

In this appendix, we give the success rates for the datasets not included in Section 5. Tables A.1–A.4 show the micro- and macro-averaged F-measures for Reviews, LA1, Classic3, and RCV1 datasets, respectively.

References

Bakus, J., & Kamel, M. S. (2006). Higher order feature selection for text classification. *Knowledge Information Systems, 9*(4), 468–491.

Camps-Valls, G., Mooij, J., & Schölkopf, B. (2010). Remote sensing feature selection by kernel dependence measures. *IEEE Geoscience and Remote Sensing Letters, 7*(3), 587–591.

Chawla, N. V., Japkowicz, N., & Kotcz, A. (2004). Editorial: Special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter, 6*(1), 1–6.

Chen, X. -w., & Wasikowski, M. (2008). FAST: A roc-based feature selection metric for small samples and imbalanced data classification problems. In *Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 124–132). Las Vegas.

Chen, X.-w., Zeng, X., & van Alphen, D. (2006). Multi-class feature selection for texture classification. *Pattern Recognition Letters, 27*(14), 1685–1691.

Dasgupta, A., Drineas, P., Harb, B., Josifovski, V., & Mahoney, M. W. (2007). Feature selection methods for text classification. In *Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data min*ing (pp. 230–239). SanJose.

Debole, F., & Sebastiani, F. (2003). Supervised term weighting for automated text categorization. In *Proceedings of the 18th ACM Symposium on Applied Computing* (pp. 784–788). ACM Press.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research, 3*, 1289–1305.

Forman, G. (2004). A pitfall and solution in multi-class feature selection for text classification. In *Proceedings of the 21st International Conference on Machine Learning* (pp. 297–304). Alberta.

Galavotti, L., Sebastiani, F., & Simi, M. (2000). Experiments on the use of feature selection and negative evidence in automated text categorization. In *Proceedings of the 4th European conference on research and advanced technology for digital libraries* (pp. 59–68). Lisbon.

Grigorescu, S. E., Petkov, N., & Kruizinga, P. (2002). Comparison of texture features based on Gabor filters. *IEEE Transactions on Image Processing, 11*(10), 1160–1167.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research, 3*, 1157–1182.

How, B. C., & Kulathuramaiyer, N. (2004). An empirical study of feature selection for text categorization based on term weightage. In *Proceedings of the IEEE/WIC/ACM international conference on web intelligence* (pp. 599–602).

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the European conference on machine learning* (pp.137–142).

Joachims, T. (1999). Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, & A. Smola (Eds.), *Advances in Kernel methods – support vector learning*. MIT Press.

Joachims, T. (2006). Training linear SVMs in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 217–226). Philadelphia.

Lan, M., Tan, C. L., Su, J., & Lu, Y. (2009). Supervised and traditional term weighting methods for automatic text categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 31*(4), 721–735.

Lewis, D. D., Yang, Y., Rose, T. G., & Li, F. (2004). RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research, 5*, 361–397.

Li, S., Xia, R., Zong, C. & Huang, C. -R. (2009). A framework of feature selection methods for text categorization. In *Proceedings of the 47th annual meeting of the ACL and the 4th IJCNLP of the AFNLP* (pp. 692–700). Singapore.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *An introduction to information retrieval*. Cambridge University Press.

Neumayer, R., Mayer, R., & Nørvåg, K. (2011). Combination of feature selection methods for text categorisation. In *Proceedings of the 33rd European conference on advances in information retrieval* (pp. 763–766). Dublin.

Ogura, H., Amano, H., & Kondo, M. (2011). Comparison of metrics for feature selection in imbalanced text classification. *Expert Systems with Applications, 38*, 4978–4989.

Olsson, J. S., & Oard, D. W. (2006). Combining feature selectors for text classification. In *Proceedings of the 15th ACM International Conference on information and knowledge management* (pp. 798–799). Arlington, Virginia.

Özgür, A., & Güngör, T. (2007). Classification of skewed and homogeneous document corpora with class-based and corpus-based keywords. In *Proceedings of the 29th German conference on artificial intelligence* (pp. 91–101). Bremen.

Özgür, A., Özgür, L., & Güngör, T. (2005). Text categorization with class-based and corpus-based keyword selection. In *Proceedings of international symposium on computer and information sciences* (pp. 607–616). İstanbul.

Pinheiro, R. H. W., Cavalcanti, G. D. C., Correa, R. F., & Ren, T. I. (2012). A global-ranking local feature selection method for text categorization. *Expert Systems with Applications, 39*, 12851–12857.

Porter, M. F. (1997). An algorithm for suffix stripping. In K. S. Jones & P. Willet (Eds.), *Readings in information retrieval* (pp. 313–316). San Francisco: Morgan Kaufmann.

Rakotomamonjy, A. (2003). Variable selection using SVM-based criteria. *Journal of Machine Learning Research, 3*, 1357–1370.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys, 34*(1), 1–47.

Singh, S. R., Murthy, H. A. & Gonsalves, T. A. (2010). Feature selection for text classification based on Gini coefficient of inequality. In *Proceedings of the 4th international workshop on feature selection in data mining* (pp. 76–85). India.

Soucy, P., & Mineau, G. W. (2005). Beyond TFIDF weighting for text categorization in the vector space model. In *Proceedings of the international joint conference on artificial intelligence* (pp. 1130–1135). Edinburgh.

Sriurai, W. (2011). Improving text categorization by using a topic model. *Advanced Computing: An International Journal, 2*(6), 21–27.

Xu, Z., King, I., Lyu, M. R.-T., & Jin, R. (2010). Discriminative semi-supervised feature selection via manifold regularization. *IEEE Transactions on Neural Networks, 21*(7), 1033–1047.

Yang, Y., & Liu, X. (1999). A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval* (pp. 42–49). Berkeley.

Yu, L., & Liu, H. (2004). Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research, 5*, 1205–1224.

Zheng, Z., & Srihari, R. (2003). Optimally combining positive and negative features for text categorization. In *Proceedings of the ICML workshop on learning from imbalanced datasets II*. Washington.

Zheng, Z., Wu, X., & Srihari, R. (2004). Feature selection for text categorization on imbalanced data. *ACM SIGKDD Explorations Newsletter, 6*(1), 80–89.