Feature Selection based on Sampling and C4.5 Algorithm to Improve the Quality of Text Classification using Naïve Bayes

Viviana Molano[1], Carlos Cobos[1], Martha Mendoza[1], Enrique Herrera-Viedma[2], and Milos Manic[3]

[1] Computer Science Department, University of Cauca, Colombia
{jvmolano, ccobos, mmendoza}@unicauca.edu.co
[2] Department of Computer Science and Artificial Intelligence, University of Granada, Spain
viedma@decsai.ugr.es
[3] Department of Computer Science, University of Idaho at Idaho Falls, Idaho Falls, U.S.A.
misko@uidaho.edu

Abstract. Automatic text classification into predefined categories is an increasingly important task given the vast number of electronic documents available on the Internet and enterprise servers. Successful text classification relies heavily on the vital task of dimensionality reduction, which aims to improve classification accuracy, give greater expression to the classification process, and improve classification computational efficiency. In this paper, two algorithms for feature selection are presented, based on sampling and weighted sampling that build on the C4.5 algorithm. The results demonstrate considerable improvements with regard to classification accuracy - up to 10% - compared to traditional algorithms such as C4.5, Naïve Bayes and Support Vector Machines. The classification process is performed using the Naïve Bayes model in the space of reduced dimensionality. Experiments were carried out using data sets based on the Reuters-21578 collection.

1 Introduction

Thanks to the continued growth of digital information and the increasing accessibility, the classification of text documents has become a task of great interest to the world. The classification task supports key tasks related to electronic trading, search engines, antivirus, email, etc. A great deal of research has been devoted to the subject, and a variety of solutions proposed that apply or adapt such algorithms as Naïve Bayes [1-3], K Nearest Neighbors (KNN) [4-7], Support Vector Machines (SVM) [8, 9] and Neural Networks [10].

The text classification process begins by characterizing the documents. This leads to a structured representation that encapsulates the information in them. A reliable representation of a document is the result of the extraction and selection of its most representative characteristics and its encoding and organization in order to be processed by a classification algorithm. Feature extraction is the process of segmentation and analysis of the text, from which it is possible to differentiate components such as

adfa, p. 1, 2011.
© Springer-Verlag Berlin Heidelberg 2011

paragraphs, sentences, words, relationships of frequencies, among others, that define the document's content or structure. These components represent the characteristics and work at a syntactic or semantic level. The syntactic characteristics (features) refer to statistical data on occurrences of segmented components (words or phrases), while the semantic features are linked with the sense that they are given and relationships that may exist between them. When features have been extracted, it is crucial to measure their amount of representativeness (importance), i.e. measure of the degree of differentiation that these features provide between the two documents. With this in mind, it is determined whether or not features need to be taken into account during the classification process. This is the task of feature selection, which predominantly seeks to reduce dimensionality, improving the accuracy of the classification process. This reduction can also be done by finding nontrivial relationships between features.

With the feature set defined, each document is differentiated according to its content and represented so that it can be processed by a classification algorithm. This algorithm is responsible for categorizing the content, by using a classifier model that is obtained in a training phase with labeled data (with a defined class), or by comparing its similarity to other documents that have a class assigned.

During the process previously explained, the principal points comprise: 1) managing of the high dimensionality of the feature sets obtained in the text collections, and 2) increasing the expressivity of the classification models generated. In seeking to alleviate the previously stated problems, this paper presents a review of the state of the art and proposes two algorithms that apply C4.5 under the concept of sampling and weighted sampling to reduce dimensionality, and build upon Naïve Bayes algorithm for executing the classification process on the reduced feature space. The novel method exhibits better results in classification accuracy and generates models that are easier to understand by users than the methods typically used.

The rest of the paper goes as follows. Section 2 presents recent research work related to text classification. Section 3 describes the proposed algorithm and its variations. Section 4 describes the data set for evaluation and the comparative analysis against C4.5, Naïve Bayes and Support Vector Machines techniques. Finally, the conclusions and future work the authors plan to pursue are presented in Section 5.

2 Related Works

A very widely based state of the art already exists with regard to automatic text classification. As a result, there may be a number solutions designed to meet the varied challenges this field offers. The following takes a brief look at some established methods, first related to document representation (extraction and feature selection) and then focused on the task of classification.

2.1 Document representation: Extraction and Feature Selection

Many researchers have focused their attention on finding the best representation mechanism, knowing that this task is critical to the success of the classification. Vec-

tor Space Model (VSM) based on the model Bag Of Words (BOW), represents a document as a vector of words or phrases associated with their frequency of occurrence, which is commonly calculated using TF-IDF [6, 11, 12]. VSM is the most used method, for its simple implementation, easy interpretation and because it achieves highly significant condensed document content information [11-13]. However, the information it provides is only syntactic in nature and does not take into account the meaning and distribution of terms or structure of the document, in addition to the vectors being high-dimensional [1, 14, 15]. Another widely used model is Latent Semantic Indexing (LSI), which analyzes co-occurrence of high order to find latent semantic relationships between terms. This model finds synonymy and polysemy relationships [11, 15, 16] but has a high computational cost [11].

As a result of the shortcomings of these methods, there are new proposals which explore other data structures and semantic relationships. In [17] a two-level representation is proposed: building a VSM using TF-IDF terms (syntax), and generating concepts, associating each term, depending on the context, with a corresponding definition in Wikipedia (semantic). In [14], graphs to represent both content and structure are used, supported by WordNet. In [16], the authors also use graphs to represent patterns of association between terms. These patterns are roads that are given by the co-occurrence of terms in documents belonging to the same class. In [18] BOW is extended by analyzing grammatical relations between terms to determine patterns of lexical dependency. In [15] a document is represented by a vector that includes concepts, which are combinations of semantically related terms (according to predefined syntactic features). The work done [19] in presents a model for feature extraction composite (c-features) based on the co-occurrence of pairs of terms for each category, regardless of position, order or distance. In [20] the document title importance is highlighted and even though its terms may not be high frequency, they propose to assign greater weight in the feature matrix (TF-IDF), to the terms that it contains. Similarly to [21] except that it analyzes semantically the title to extract concepts before to the weighting.

Other works done in this area apply the concept of clustering. In [9] clusters of words closely related at semantic level (based on co-occurrences of terms across categories) are created and each is treated as a new feature. Some studies have also been done in relation to selection measures: the study in [22] concludes that the best performance is obtained when signed X2 and signed information gain are combined. In [23] it is determined that the measures in which Naïve Bayes achieves the greatest accuracy in the selection task are Multi-class Odds Ratio (MOR) and Class Discriminating Measure (CDM), CDM being the highest simplicity.

All the above mentioned proposals seek to enrich the semantic representation of a document and emphasize the importance of selecting the really significant features prior to classification. However, it is important to note that none of these proposals is clear as to whether all selected features are contributing to the classification process, which indicates that the level of reduction could be carried out further. In most of the work reviewed so far, the selection process and reduction are developed based on the analysis of certain metrics such as Information Gain (IG), Mutual Information (MI), or generally posting frequency. However, what is not taken into account is the inclu-

sion of a classifier, which could contribute to refine the set of features needed to improve the classification task. In many cases a threshold is required, which is difficult to optimally define. In [24], an objective function of feature selection based on probability is presented, which defines a Bayesian adaptive model selection. However, this approach is computationally very expensive.

2.2 Classification

In classification there are also many research papers and hence many proposals developed that revolve around improving the accuracy of the results and reduce computing costs. In [25], the ISOBagC4.5 algorithm is proposed, which implements Isomap for feature reduction and Bagging with C4.5 algorithm for classification. Their results are better than Bagging C4.5 but the optimum values are not defined for the parameters and the complexity of the algorithm Isomap is very high.

In [26] and [27] methods for generating clusters are proposed based on similarity of features using K-means (or an extension thereof). Each cluster is trained to generate a specific classification. These approaches based on clustering have an expensive training phase, especially when large and unbalanced data sets are involved. Furthermore, in [10], it is shown how to generate clusters using a neural network using frequency matrix of terms by document. The results improve as the size of the training set increases.

There are other proposals that have sought to extend and enhance traditional classification algorithms, e.g. [28] proposes the use of KNN with the Mahalanobis distance. [29] authors improve K-NN to reduce the search space of the immediate neighbors. In [13], the importance of data distribution is highlighted. They use a measure of density to increase or decrease the distance between a sample to be classified and its K nearest neighbors. In this work, the increase in accuracy is more visible as the training set grows. [12] describes an algorithm based on KNN classifier with feature selection after taking into account the frequency, distribution and concentration of the data. In [4], an improved KNN is put forward where the parameter K is optimized based on the features selected by cross validation, and that uses IG as a metric for comparison. The accuracy of the results is much higher than conventional KNN, but not very significant compared with SVM. The work proposed in [30] is based on a graph representation where the weights are calculated using KNN (cosine measure) from TF-IDF matrix. On average, the results are more accurate than the comparison algorithms (including SVM, TSVM, and LP), but in the comparison of accuracy by category it is not always better.

The idea presented in [8] is based on combining SVM and KNN by classification in two stages. The first stage uses VPRSVM (SVM based on Variable Precision Rough sets - VPRS) to filter noise and partition the feature space by category (according to the level of confidence in the assignment of the class). The second stage focuses on RKNN (Restrictive K Nearest Neighbor) to reduce class candidates from partitions generated. In [31], the authors propose to construct a combined classifier from SVM, Naïve Bayes and Rocchio that trains with positive data and is capable of generating negative from unlabeled data.

In [1], a Naïve Bayes Multinomial extension (MNB) is shown, which presents a semi-supervised algorithm for learning parameters: Semi-Supervised Frequency Estimate (SFE). Precision results obtained do not exceed MNB for all sets of test data. In [16], the Higher Order Naïve Bayes (HONB) algorithm is put forward; this algorithm takes advantage of the connectivity of the search terms by chains that co-occur among the documents of the same category. This proposal has a search phase connectivity that greatly increases the complexity of Naïve Bayes.

In [32], the authors present the High Relevance Keyword Extraction (HRKE) method to achieve text pre-processing and feature selection. In [33], a modeling language based on n-grams applied to the classification is used. In [34], the learning process is performed based on two types of related documents. A set of pre-labeled documents and other unlabeled documents set. The method performs automatic classification of the second data set through knowledge extracted from the features it shares with the first.

Some researchers elaborated more on the metrics used to compare two documents. For example, in [35], a generalization of the cosine measure using the Mahalanobis distance was proposed. This measure considers the correlation between terms. In [36], some measures for the KNN classification according to the results are explored. In this document, the authors argue that the choice of metric is dependent on the application domain. Other research has been directed toward specific applications of text classification. For example, in [2], Naïve Bayes Shrinkage for analysis based on medical diagnoses is presented, while in [3] web classification by Naïve Bayes algorithm that handles HTML tags and hyperlinks is presented. In [37], an extension of TF-IDF for unbalanced data representation given its distribution for the discovery of behavioral patterns between proteins from published literature is presented.

3 The Newly Presented Methods

The method of feature selection (dimensionality reduction) presented in this paper has four stages: preprocessing, model generation, feature selection and classification. In the following, a detailed description of these stages is presented.

The method is based on the Terms by Documents Matrix (TDM) commonly used in Information Retrieval (IR). This matrix is built in the preprocessing stage. This stage use Lucene [38] and includes: terms tokenizer, lower case filter, stop word removal, Porter's stemming algorithm [39] and the building of the TDM matrix. TDM is based on the vector space model [39]. In this model, the documents are designed as bags of words, the document collection is represented by a matrix of D-terms by N-documents, each document is represented by a vector of normalized frequency term (tf_i) by the inverse document frequency for that term, in what is known as TF-IDF value (see Eq. (1)).

$$w_i = \frac{freq_i}{\max\left(freq_i\right)} \times \log\left(\frac{N}{n_i + 1}\right) \tag{1}$$

The proposed method, called 10-WS-C4.5-TDM-NB-TDMR, uses ten (10) samples obtained with weighting techniques (WS). The document representation model is the TDM matrix. Each sample is used to create a specific decision tree based on C4.5 algorithm. Next, all different attributes in the 10 decision trees are used in order to build a reduced TDM matrix of documents (TDMR), and finally, the Naïve Bayes (NB) algorithm is used to classify new documents. **Fig. 1** shows the general pseudo-code of this method, including the model generation stage. An alternative method, called 10-S-C4.5-TDM-NB-TDMR, uses sampling with replacement (S in the name of this method instead of WS in previous one) instead of sampling with weighting, as is shown in **Fig. 2**. The final product of this stage is a list of terms that appears in all C4.5 decision trees. This list of terms is a subset of the D-terms in TDM matrix.

```
Preprocessing
Read text collection.
Create a TDM matrix including: Tokenize, lower case filter, stop word remov-
al, and stemming process.

Model generation
Assign equal weight to each training instance.
Initialize list of terms (L).
For each of I iterations:
   Apply C4.5 to weighted dataset.
   Extract terms (t) from C4.5 tree and include in list (L ← L U t).
   Compute error e of model on weighted dataset and store error.
   If e equal to zero:
     Terminate model generation.
   For each instance in dataset:
     If instance is not classified correctly by model:
          Multiply weight of instance by e / (1 – e).
   End For
   Normalize weight of all instances.
End For

Feature Selection
TDMR ← Reduce TDM matrix to selected terms in List L.
Build a Naïve Bayes model on TDMR and stored.

Classification
Predict class of new instances using Naïve Bayes model on TDMR representa-
tion.
```

Fig. 1. Pseudo-code for 10-WS-C4.5-TDM-NB-TDMR method.

The next stage, called Feature Selection, focuses on the reduction of the TDM matrix. This new TDM matrix is called TDM Reduced (TDMR) and includes only the set of terms stored in the previous built list. Then, a Naïve Bayes (NB) model is applied to this new matrix (TDMR). Finally, the classification stage occurs when users need to classify a new instance (document). The document is represented in the reduced space (same terms on TDMR) and classified based on the Naïve Bayes model previously built and stored. It should be noted that just one model is needed in the classification stage.

```
Model generation
Let n be the number of instances in the training data.
```

```
Initialize list of terms (L)
For each of I iterations:
   Sample n instances with replacement from training data.
   Apply C4.5 to the sample.
   Extract terms (t) from C4.5 tree and include in list (L ← L U t).
End For
```

Fig. 2. Model generation stage in 10-S-C4.5-TDM-NB-TDMR method.

The proposed method has an estimated time complexity of in the preprocessing stage, in the model generation stage (based on complexity of C4.5 algorithm), in the feature selection stage, and in the classification stage, where I is the number of iterations (C4.5 models), m is the size of the training data, n is the number of attributes of the training data, c is the number of classes, and r is the number of attributes of the reduced training data (r <<n). In general, the training phase (preprocessing, model generation, and features selection stages) is, and will therefore have linear complexity with regard to the size of the training dataset and have a quadratic complexity with regard to the number of attributes in the training dataset. The testing (classification) phase is very fast (linear complexity with regard to the number of classes and the number of reduced attributes).

4 Experimentation

Datasets for assessment: The Reuters-21578 collection is commonly used as a neutral third party classifier, using human editors to classify manually and store thousands of news items. In this research a total of one hundred datasets were randomly built (these datasets is called Reuters-100; for details see www.unicauca.edu.co/~ccobos/wdc/reuters-100.htm). On average, datasets have 81.2 documents, 4.9 topics and 1,945 terms. **Table 1** shows detailed information from each dataset.

Measures: There are many different methods proposed for measuring the quality of classification. Three of the best known are precision, recall and F-measure, commonly used in IR [39]. In this research, the measures weighted Precision, weighted Recall and weighted F-measure (the harmonic means of precision and recall) are used to evaluate the quality of solution. The True Positive Rate, the False Positive Rate, the True Negative Rate, and the False Negative Rate were used to compare method results.

Results with datasets: The proposed algorithms were compared with C4.5, Naïve Bayes, and Support Vector Machines algorithms (all of them available in Weka). **Table 1** shows detailed results of Precision, Recall, and F-measure for each dataset. **Table 2** shows general results (mean, standard deviation, minimum value, and maximum value) of Precision, Recall and F-measure over all datasets. **Table 3** shows results of other important indexes, namely: True Positive Rate (TPR), True Negative Rate (TNR), False Positive Rate (FPR), False Negative Rate (FNR), and Receiver Operating Characteristic (ROC). Tests were carried out using cross validation with 10-folds.

* Best results in bold

				C4.5			NB			SVM			10-S-C4.5-TDM-NB- TDMR			10-WS-C4.5-TDM-NB-TDMR		
Id	**#Docs**	**#Class**	**#Attr**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
0	49	2	1427	0.96	0.96	0.96	0.94	0.94	0.93	0.93	0.92	0.91	0.98	0.98	0.98	**1.00**	**1.00**	**1.00**
1	103	8	2601	0.83	0.83	0.82	0.78	0.75	0.75	0.87	0.86	0.86	**0.94**	**0.92**	**0.93**	0.82	0.83	0.82
2	77	4	1824	0.90	0.90	0.90	0.91	0.90	0.89	0.91	0.91	0.91	0.84	0.82	0.82	**0.94**	**0.94**	**0.94**
3	93	4	2100	0.91	0.90	0.90	0.94	0.94	0.94	**0.97**	**0.97**	**0.97**	0.92	0.90	0.90	**0.97**	**0.97**	**0.97**
4	78	3	1957	0.92	0.92	0.92	0.91	0.91	0.91	0.87	0.87	0.87	**1.00**	**1.00**	**1.00**	0.97	0.97	0.97
5	74	7	1989	0.97	0.97	0.97	0.74	0.73	0.73	0.84	0.81	0.81	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**
6	81	4	1899	0.90	0.91	0.91	0.89	0.89	0.89	0.92	0.93	0.92	0.94	0.95	0.94	**0.96**	**0.98**	**0.97**
7	91	4	2254	**0.95**	**0.95**	**0.95**	0.82	0.81	0.81	0.86	0.85	0.84	0.84	0.81	0.82	0.92	0.90	0.90
8	78	6	1993	0.77	0.76	0.76	0.75	0.71	0.71	**0.89**	**0.88**	**0.89**	0.85	0.82	0.82	0.84	0.83	0.83
9	60	2	1578	0.94	0.93	0.93	0.93	0.92	0.91	**0.98**	**0.98**	**0.98**	**0.98**	**0.98**	**0.98**	**0.98**	**0.98**	**0.98**
10	75	7	2156	0.90	0.89	0.89	0.69	0.65	0.64	0.81	0.73	0.71	0.90	0.89	0.89	**0.96**	**0.96**	**0.96**
11	76	8	1891	0.81	0.80	0.80	0.86	0.84	0.83	0.86	0.82	0.81	0.83	0.82	0.82	**0.92**	**0.91**	**0.91**
12	102	3	2394	0.95	0.94	0.94	0.92	0.92	0.92	0.96	0.96	0.96	0.96	0.95	0.95	**0.98**	**0.98**	**0.98**
13	32	2	838	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	**1.00**	**1.00**	**1.00**
14	71	3	1774	0.96	0.96	0.96	0.93	0.92	0.91	0.92	0.92	0.90	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**
15	91	3	1968	0.88	0.88	0.88	0.93	0.92	0.92	0.93	0.93	0.93	0.94	0.93	0.94	**0.96**	**0.96**	**0.96**
16	68	2	1729	**1.00**	**1.00**	**1.00**	0.96	0.96	0.96	0.99	0.99	0.99	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
17	96	6	2381	**0.88**	**0.88**	**0.88**	0.61	0.60	0.60	0.73	0.69	0.70	0.79	0.79	0.79	0.81	0.80	0.80
18	52	3	1458	0.97	0.96	0.96	0.89	0.87	0.85	0.88	0.85	0.83	0.97	0.96	0.96	**1.00**	**1.00**	**1.00**
19	78	6	1898	0.65	0.67	0.65	0.68	0.64	0.62	0.68	0.73	0.69	**0.85**	**0.81**	**0.81**	0.82	0.78	0.77
20	72	5	1885	0.86	0.88	0.87	0.76	0.76	0.76	0.85	0.85	0.83	**0.94**	**0.94**	**0.94**	0.92	0.93	0.92
21	28	3	838	0.89	0.93	0.91	0.94	0.96	0.95	0.73	0.86	0.79	**0.93**	**0.96**	**0.95**	**0.93**	**0.96**	**0.95**
22	63	2	1519	0.96	0.95	0.95	**1.00**	**1.00**	**1.00**	0.98	0.98	0.98	0.95	0.95	0.95	0.97	0.97	0.97
23	83	6	2264	0.72	0.73	0.72	0.77	0.71	0.71	**0.91**	**0.90**	**0.90**	0.84	0.83	0.83	0.81	0.81	0.81
24	104	5	2565	0.94	0.93	0.93	0.83	0.76	0.76	0.91	0.91	0.91	0.94	0.93	0.93	**0.95**	**0.95**	**0.95**
25	62	4	1372	**1.00**	**1.00**	**1.00**	0.90	0.87	0.86	0.92	0.90	0.90	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
26	91	3	2189	0.87	0.87	0.87	0.91	0.91	0.91	0.93	0.93	0.93	0.94	0.93	0.93	**0.99**	**0.99**	**0.99**
27	69	5	1871	**0.88**	**0.87**	**0.87**	0.77	0.74	0.74	0.79	0.78	0.78	0.85	0.84	0.84	0.85	0.86	0.85
28	41	3	1293	0.94	0.93	0.93	**0.98**	**0.98**	**0.97**	0.88	0.85	0.84	**0.98**	**0.98**	**0.97**	**0.98**	**0.98**	**0.97**
29	81	5	2028	0.90	0.90	0.90	0.78	0.75	0.76	0.87	0.84	0.83	0.82	0.81	0.81	**0.96**	**0.95**	**0.95**
30	46	3	1061	**0.96**	**0.98**	**0.97**	0.90	0.91	0.90	0.88	0.89	0.87	**0.96**	**0.98**	**0.97**	0.94	0.96	0.95
31	73	6	1696	0.81	0.82	0.82	0.78	0.75	0.75	0.84	0.82	0.82	**0.90**	0.89	0.89	**0.90**	**0.90**	**0.90**
32	85	5	2008	0.84	0.85	0.84	0.85	0.85	0.84	**0.92**	**0.93**	**0.92**	0.91	0.91	0.91	0.91	**0.92**	0.91
33	44	3	1716	**1.00**	**1.00**	**1.00**	0.94	0.93	0.93	0.94	0.93	0.93	0.98	0.98	0.98	0.98	0.98	0.98
34	103	9	2445	0.94	0.93	0.93	0.70	0.70	0.69	0.82	0.81	0.81	**0.97**	**0.98**	**0.98**	0.94	0.94	0.94
35	84	7	2045	0.67	0.68	0.66	0.79	0.76	0.76	0.73	0.79	0.76	**0.84**	**0.83**	**0.84**	**0.84**	0.83	0.83
36	33	3	945	**1.00**	**1.00**	**1.00**	0.86	0.82	0.80	0.88	0.85	0.84	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
37	110	8	2317	0.85	0.85	0.84	0.75	0.71	0.71	0.83	0.82	0.81	**0.93**	**0.93**	**0.93**	0.92	0.92	0.92
38	99	6	2290	0.93	0.91	0.91	0.75	0.71	0.71	0.83	0.79	0.78	0.88	0.87	0.87	**0.94**	**0.94**	**0.94**
39	103	5	2584	0.88	0.87	0.87	0.84	0.82	0.82	0.87	0.85	0.85	0.93	0.92	0.92	**0.94**	**0.94**	**0.94**
40	103	6	2261	0.82	0.79	0.79	0.73	0.72	0.72	0.88	0.84	0.85	0.80	0.80	0.79	**0.87**	**0.85**	**0.86**
41	72	4	1651	0.93	0.92	0.91	0.89	0.83	0.83	0.90	0.88	0.87	0.97	0.97	0.97	**0.99**	**0.99**	**0.99**
42	32	4	981	**0.94**	**0.97**	**0.95**	0.83	0.84	0.83	0.89	0.91	0.89	0.91	0.94	0.92	0.89	0.91	0.89
43	65	6	1949	0.87	0.83	0.82	0.70	0.65	0.66	0.73	0.66	0.61	**0.94**	**0.94**	**0.93**	0.91	0.91	0.91
44	92	8	2209	0.85	0.86	0.85	0.75	0.72	0.72	0.88	0.89	0.89	**0.92**	**0.90**	**0.90**	0.89	0.89	0.89
45	51	2	1146	**1.00**	**1.00**	**1.00**	0.96	0.96	0.96	0.96	0.96	0.96	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
46	55	2	1476	0.90	0.89	0.89	0.97	0.96	0.96	**0.98**	**0.98**	**0.98**	**0.98**	**0.98**	**0.98**	0.93	0.93	0.93
47	75	4	1620	**0.97**	**0.97**	**0.97**	0.81	0.80	0.80	0.90	0.89	0.89	0.95	0.95	0.95	0.96	0.96	0.96
48	82	4	1824	**0.97**	**0.96**	**0.96**	0.91	0.90	0.90	0.96	0.95	0.95	0.96	0.95	0.95	0.95	0.95	0.95
49	77	6	1796	0.87	0.87	0.87	0.78	0.74	0.75	0.80	0.78	0.78	0.90	0.88	0.88	**0.94**	**0.92**	**0.92**
50	80	2	1971	0.99	0.99	0.99	0.95	0.95	0.95	0.98	0.98	0.98	0.99	0.99	0.99	**1.00**	**1.00**	**1.00**
51	61	4	1665	0.80	0.84	0.81	0.83	0.82	0.81	0.82	0.84	0.81	0.87	0.87	0.87	**0.89**	**0.90**	**0.90**
52	71	3	1781	**0.99**	**0.99**	**0.99**	0.96	0.96	0.96	0.94	0.93	0.93	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**
53	104	9	2422	0.81	0.81	0.80	0.82	0.81	0.79	0.87	0.87	0.86	0.86	0.86	0.85	**0.91**	**0.90**	**0.90**
54	105	7	2036	0.84	0.84	0.84	0.82	0.81	0.81	**0.89**	0.88	0.88	**0.89**	**0.89**	**0.89**	0.85	0.83	0.83
55	96	7	2238	0.88	0.88	0.87	0.80	0.75	0.76	0.93	0.92	0.92	0.85	0.84	0.84	**0.91**	**0.90**	**0.90**
56	105	7	2105	0.84	0.83	0.83	0.68	0.66	0.67	0.78	0.74	0.74	**0.90**	**0.89**	**0.89**	0.82	0.80	0.80
57	86	6	2217	0.86	0.84	0.83	0.86	0.80	0.80	0.90	0.88	0.87	0.89	0.88	0.88	**0.95**	**0.94**	**0.94**
58	115	6	2605	0.87	0.87	0.87	0.77	0.73	0.73	0.90	0.89	0.88	**0.93**	**0.92**	**0.92**	0.90	0.89	0.89
59	89	7	2115	0.72	0.71	0.71	0.73	0.73	0.72	0.79	0.78	0.74	0.89	**0.90**	**0.90**	**0.90**	0.89	0.89
60	95	6	2256	0.78	0.77	0.77	0.75	0.72	0.72	0.80	0.76	0.75	**0.94**	**0.94**	**0.94**	0.91	0.91	0.91
61	102	5	2410	**0.94**	**0.93**	**0.93**	0.82	0.76	0.76	0.86	0.84	0.83	0.92	0.92	0.92	0.93	0.92	0.92
62	56	4	1445	0.95	0.96	0.96	0.91	0.93	0.92	0.89	0.89	0.88	**0.98**	**0.98**	**0.98**	0.96	**0.98**	0.97
63	100	6	2128	0.87	0.85	0.85	0.76	0.77	0.76	0.87	0.87	0.86	**0.89**	**0.88**	**0.88**	0.80	0.81	0.80
64	76	4	1963	0.88	0.88	0.88	0.87	0.87	0.87	**0.92**	0.91	0.90	**0.92**	**0.92**	**0.92**	**0.92**	**0.92**	**0.92**
65	120	3	2744	**0.98**	**0.98**	0.97	0.91	0.89	0.89	0.98	0.98	0.98	**0.98**	**0.98**	**0.98**	**0.98**	**0.98**	**0.98**
66	68	3	1968	0.94	0.94	0.94	0.83	0.81	0.81	0.95	0.94	0.94	**0.96**	**0.96**	**0.96**	**0.96**	**0.96**	0.95
67	78	6	1936	0.87	0.86	0.86	0.72	0.71	0.71	0.84	0.82	0.82	**0.94**	**0.94**	**0.94**	0.90	0.90	0.89
68	69	6	1508	**0.91**	**0.86**	**0.86**	0.80	0.78	0.78	0.83	0.83	0.82	0.90	0.88	0.88	0.89	0.87	0.87
69	65	2	1799	**1.00**	**1.00**	**1.00**	0.93	0.92	0.92	0.99	0.98	0.98	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
70	94	8	2249	0.75	0.76	0.75	0.65	0.63	0.62	0.80	**0.83**	0.81	0.79	0.79	0.78	**0.81**	0.82	**0.81**
71	76	4	2000	0.84	0.83	0.83	0.86	0.86	0.86	0.91	0.91	0.91	0.92	0.92	0.92	**0.97**	**0.97**	**0.97**
72	60	2	1231	0.97	0.97	0.97	**1.00**	**1.00**	**1.00**	0.95	0.95	0.95	0.97	0.97	0.97	0.98	0.98	0.98
73	119	7	2757	0.83	0.82	0.81	0.84	0.82	0.82	**0.93**	**0.92**	**0.92**	0.91	0.90	0.90	0.91	0.90	0.90
74	86	6	2033	0.79	0.77	**0.77**	0.67	0.65	0.65	0.74	0.72	0.72	**0.83**	**0.78**	**0.77**	0.79	0.78	0.78
75	76	3	1645	0.86	0.86	0.85	0.96	0.96	0.96	0.92	0.96	0.94	0.98	0.97	0.97	**0.99**	**0.99**	**0.99**
76	89	7	2161	0.77	0.74	0.75	0.72	0.66	0.67	0.78	0.76	0.77	0.87	0.87	0.86	**0.89**	**0.89**	**0.89**
77	77	5	1833	0.97	0.97	0.97	0.90	0.88	0.88	0.93	0.92	0.92	**1.00**	**1.00**	**1.00**	0.99	0.99	0.99
78	91	7	2300	0.85	0.86	0.85	0.78	0.78	0.78	0.90	0.87	0.87	**0.91**	**0.91**	**0.91**	0.82	0.82	0.82
79	73	8	1759	0.80	0.78	0.78	0.75	0.70	0.70	0.68	0.70	0.68	**0.87**	**0.86**	**0.86**	0.86	0.85	0.84
80	91	7	2245	0.92	0.91	0.91	0.78	0.76	0.76	0.87	0.88	0.87	**0.95**	**0.96**	**0.95**	**0.95**	**0.96**	**0.95**

81	120	4	2582	0.95	0.95	0.95	0.89	0.84	0.85	0.95	0.95	0.95	**0.98**	**0.98**	**0.97**	0.96	0.96	0.96
82	86	3	1532	0.88	0.88	0.88	0.97	0.97	0.96	0.95	0.95	0.95	0.94	0.93	0.93	**0.98**	**0.98**	**0.98**
83	87	9	2057	0.77	0.74	0.72	0.65	0.68	0.66	0.73	0.77	0.74	**0.78**	**0.76**	**0.76**	0.71	0.74	0.72
84	80	4	2055	0.84	0.84	0.84	0.90	0.90	0.90	0.92	0.91	0.91	0.95	0.95	0.95	**0.96**	**0.96**	**0.96**
85	104	6	1890	0.77	0.77	0.77	0.88	0.87	0.86	0.94	0.94	0.94	**0.96**	**0.96**	**0.96**	0.93	0.92	0.92
86	107	5	2486	0.88	0.88	0.88	**0.91**	0.90	**0.90**	**0.91**	**0.91**	**0.90**	0.86	0.86	0.86	**0.91**	0.90	**0.90**
87	83	7	1642	0.82	0.83	0.82	0.82	0.82	0.81	0.83	0.82	0.82	**0.93**	**0.94**	**0.93**	0.90	0.90	0.90
88	63	4	1904	0.91	0.90	0.90	0.85	0.79	0.78	0.80	0.76	0.73	0.93	0.92	0.92	**0.97**	**0.97**	**0.97**
89	105	5	2599	0.93	0.92	0.92	0.84	0.81	0.81	0.92	0.91	0.91	**0.93**	**0.92**	**0.92**	0.92	0.91	0.92
90	53	4	1465	0.95	0.96	0.95	0.87	0.89	0.88	0.88	0.89	0.88	**1.00**	**1.00**	**1.00**	0.96	0.98	0.97
91	120	4	2149	0.90	0.90	0.90	0.93	0.93	0.93	0.96	0.96	0.96	0.97	0.97	0.97	**0.98**	**0.98**	**0.98**
92	100	3	2166	**1.00**	**1.00**	**1.00**	0.94	0.94	0.94	0.95	0.95	0.95	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
93	68	3	1371	0.97	0.97	0.97	0.96	0.96	0.96	0.96	0.96	0.96	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**
94	120	4	2284	0.96	0.96	0.96	0.93	0.93	0.93	**0.98**	**0.98**	0.97	0.93	0.93	0.93	**0.98**	**0.98**	**0.98**
95	83	4	2211	0.91	**0.90**	**0.90**	0.81	0.78	0.78	0.88	0.87	0.86	0.89	0.89	0.89	**0.92**	**0.90**	**0.90**
96	91	7	2188	0.78	0.76	0.75	0.87	0.87	0.86	0.87	0.85	0.85	0.91	0.91	0.91	**0.92**	**0.92**	**0.92**
97	101	9	2187	0.67	0.65	0.65	0.71	0.67	0.68	0.75	0.76	0.73	**0.80**	**0.78**	**0.78**	0.79	**0.78**	**0.78**
98	80	3	2027	**0.99**	**0.99**	**0.99**	0.89	0.89	0.89	0.96	0.95	0.95	0.98	0.98	0.97	0.95	0.95	0.95
99	95	4	2226	0.86	0.85	0.85	0.85	0.83	0.83	0.94	0.94	0.94	**0.98**	**0.98**	**0.98**	0.97	0.97	0.97

Table 1. Description of Datasets (#Docs for number of documents, #Class for number of classes, #Attr for number of attributes, P for Precision, R for Recall and F for F-Measure)

	C4.5			**NB**			**SVM**			**10-S-C4.5-TDM-NB- TDMR**			**10-WS-C4.5-TDM-NB-TDMR**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Mean	0.89	0.88	0.88	0.84	0.82	0.82	0.88	0.88	0.87	0.92	0.92	**0.92**	**0.93**	**0.93**	**0.92**
Std.Dev.	0.08	0.08	0.08	0.09	0.10	0.10	0.07	0.08	0.08	**0.06**	**0.06**	**0.06**	**0.06**	**0.06**	**0.06**
Min	0.65	0.65	0.65	0.61	0.60	0.60	0.68	0.66	0.61	**0.78**	**0.76**	**0.76**	0.71	0.74	0.72
Max	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.99	0.99	0.99	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

Table 2. General Results Part I: Number of documents (#Docs), number of classes (#Class), number of attributes (#Attr), Precision (P), Recall (R), and F-Measure (F).

* Best results in bold	**TPR**	**TNR**	**FPR**	**FNR**	**ROC**
C4.5	0.884	0.962	0.038	0.116	0.926
NB	0.824	0.932	0.068	0.176	0.904
SVM	0.876	0.934	0.066	0.124	0.927
10-S-C4.5-TDM-NB- TDMR	**0.920**	**0.975**	**0.025**	**0.080**	**0.981**
10-WS-C4.5-TDM-NB-TDMR	**0.927**	**0.977**	**0.023**	**0.073**	**0.985**

Table 3. General Results Part II.

On average, the results on all 100 datasets show that 10-WS-C4.5-TDM-NB-TDMR and 10-S-C4.5-TDM-NB- TDMR are better (based on all index: precision, recall, f-measure, true positive rate, true negative rate, false positive rate, false negative rate, and receiver operating characteristics) than other methods; therefore, the general performance of the proposed methods are better in Reuters-100 collection. Improvements in precision, recall, F-measure, TPR, and FNR are between 4% and 10%. Improvements in TNR and FPR are between 1.5% and 4.5%. Improvements in ROC are between 6% and 8%.

The feature selection process allows a more understandable model to be obtained. The models are more compact and clear to users. They are also very light and computationally very cheap (in classification stage). With 10-S-C4.5-TDM-NB-TDMR the average feature reduction is 99.06%. For example, the data set 92 with 2166 attributes is reduced to 3 attributes and the data set 35 with 2045 attributes is reduced to 47 attributes.

Some specific datasets do not follow the general tendency, for example, dataset number 1 shows better results for 10-S-C4.5-TDM-NB-TDMR and then for SVM.

Therefore, it is necessary to review the pruned process on C4.5 trees and some tuning parameters (for example the number of iterations or models). Also, it is necessary to use concepts instead of terms in the Term by Document Matrix (TDM) e.g. using tools based on science mapping to identify the concepts [40].

5 Conclusions and future work

Two novel methods for feature selection and text classification, called 10-S-C4.5-TDM-NB-TDMR and 10-WS-C4.5-TDM-NB-TDMR, were presented in this paper. These approaches are aimed at applications such as spam filtering, where additional clarity, efficiency, and ease of use is needed for human operators to be effective. The methods presented were tested on publicly available datasets (Reuters-100). Comparisons with C4.5, Naïve Bayes, and Support Vector Machine techniques demonstrated consistent improvements of up to 10% in precision, recall and F-measure. TPR (true positive rates), FNR (false negative rates), and ROC (receiver operating characteristic), demonstrated similar improvements.

As future work, the authors are planning on including ontologies and parts of speech detection techniques in the preprocessing stage. Also, a detailed study will be conducted to define the best value for number of iterations or number of models it is required to use in the model generation stage. It is necessary to evaluate the proposed model over different test sets, such as LingSpam, and evaluate other combinations of models, e.g. C4.5 with Neural Networks or CART with Naïve Bayes. Finally, tuning some parameters of C4.5 and Naïve Bayes algorithms in order to increase the accuracy of the entire method will be considered.

6 Acknowledgments

This paper has been developed with the Federal financing of Projects FuzzyLIng-II TIN2010-17876, Andalucian Excellence Projects TIC5299 and TIC-5991, and Universidad del Cauca under Project VRI-2560.

7 References

1. Su, J., J. Sayyad-Shirab, and M. Stan, *Large Scale Text Classification using Semi-supervised Multinomial Naive Bayes.* Proceedings of the 28th International Conference on Machine Learning (ICML-11), 2011: p. 97--104.
2. Laur, E.J.M., #237, and A.D. March, *Combining Bayesian Text Classification and Shrinkage to Automate Healthcare Coding: A Data Quality Analysis.* J. Data and Information Quality, 2011. **2**(3): p. 1-22.
3. He, Y., J. Xie, and C. Xu. *An improved Naive Bayesian algorithm for Web page text classification*. in *Fuzzy Systems and Knowledge Discovery (FSKD), 2011 Eighth International Conference on*. 2011.
4. Ambert, K.H. and A.M. Cohen, *k-Information Gain Scaled Nearest Neighbors: A Novel Approach to Classifying Protein-Protein Interaction-Related Documents.* Computational Biology and Bioinformatics, IEEE/ACM Transactions on, 2012. **9**(1): p. 305-310.

5. Wajeed, M.A. and T. Adilakshmi. *Semi-supervised text classification using enhanced KNN algorithm.* in *Information and Communication Technologies (WICT), 2011 World Congress on*. 2011.
6. Trstenjak, B., S. Mikac, and D. Donko, *KNN with TF-IDF based Framework for Text Categorization.* Procedia Engineering, 2014. **69**(0): p. 1356-1364.
7. Bhadri Raju, M.S.V.S., B. Vishnu Vardhan, and V. Sowmya, *Variant Nearest Neighbor Classification Algorithm for Text Document*, in *ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India- Vol II*, S.C. Satapathy, et al., Editors. 2014, Springer International Publishing. p. 243-251.
8. Li, W., D. Miao, and W. Wang, *Two-level hierarchical combination method for text classification.* Expert Systems with Applications, 2011. **38**(3): p. 2030-2039.
9. Jung-Yi, J., L. Ren-Jia, and L. Shie-Jue, *A Fuzzy Self-Constructing Feature Clustering Algorithm for Text Classification.* Knowledge and Data Engineering, IEEE Transactions on, 2011. **23**(3): p. 335-349.
10. Saha, D. *Web Text Classification Using a Neural Network.* in *Emerging Applications of Information Technology (EAIT), 2011 Second International Conference on.* 2011.
11. Zhang, W., T. Yoshida, and X. Tang, *A comparative study of TF-IDF, LSI and multi-words for text classification.* Expert Systems with Applications, 2011: p. 38(3): p. 2758-2765.
12. Shi, K., et al., *Efficient text classification method based on improved term reduction and term weighting.* The Journal of China Universities of Posts and Telecommunications, 2011. **18, Supplement 1**(0): p. 131-135.
13. Shi, K., S. Shanghai Jiaotong Univ., China , and L.L.H.L.J.H.N.Z.W. Song, *An improved KNN text classification algorithm based on density.* Cloud Computing and Intelligence Systems (CCIS), 2011 IEEE International Conference on, 2011: p. 113 - 117.
14. Jiang, C., et al., *Text classification using graph mining-based feature extraction.* Knowledge-Based Systems, 2010. **23**(4): p. 302-308.
15. Sun, Y., X. Liu, and X. Cui. *The Mining of Term Semantic Relationships and its Application in Text Classification*. in *Intelligent Computation Technology and Automation (ICICTA), 2012 Fifth International Conference on*. 2012.
16. Ganiz, M.C., C. George, and W.M. Pottenger, *Higher Order Naïve Bayes: A Novel Non-IID Approach to Text Classification.* Knowledge and Data Engineering, IEEE Transactions on, 2011. **23**(7): p. 1022-1034.
17. Yun, J., et al., *A multi-layer text classification framework based on two-level representation model.* Expert Systems with Applications, 2012. **39**(2): p. 2035-2046.
18. Özgür, L. and T. Güngör, *Text classification with the support of pruned dependency patterns.* Pattern Recognition Letters, 2010. **31**(12): p. 1598-1607.
19. Figueiredo, F., et al., *Word co-occurrence features for text classification.* Information Systems, 2011. **36**(5): p. 843-858.
20. Tian Xia ; Dept. of Comput. & Inf., S.S.P.U., Shanghai, China ; Yi Du, *Improve VSM text classification by title vector based document representation method.* Computer Science & Education (ICCSE), 2011 6th International Conference on, 2011: p. 210 - 213.
21. Zhang, P.Y., *The Application of Semantic Similarity in Text Classification.* Modern Development in Materials, Machinery and Automation, 2013. **346**: p. 141-144.
22. Hiroshi Ogura, H.A., Masato Kondo, *Comparison of metrics for feature selection in imbalanced text classification.* Expert Systems with Applications, 2011. **38**(5): p. 4978-4989.
23. Chen, J., et al., *Feature selection for text classification with Naïve Bayes.* Expert Systems with Applications, 2009. **36**(3, Part 1): p. 5432-5435.
24. Guozhong Feng, J.G., Bing-Yi Jing, Lizhu Hao, *A Bayesian feature selection paradigm for text classification.* Information Processing & Management, 2012. **48**(2): p. 283-302.

25. Duan, F.L.J.F.L.W.H.Z.R., *A method based on manifold learning and Bagging for text classification.* Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), 2011 2nd International Conference on, 2011: p. 2713 - 2716.
26. Yan Li, E.H., Korris Chung, *A subspace decision cluster classifier for text classification.* Expert Systems with Applications, 2011. **38**(10): p. 12475-12482.
27. Nizamani, S.M., N.; Wiil, U.K.; Karampelas, P., *CCM: A Text Classification Model by Clustering.* Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on, 2011: p. 461 - 467.
28. Suli, Z. and P. Xin. *A novel text classification based on Mahalanobis distance*. in *Computer Research and Development (ICCRD), 2011 3rd International Conference on*. 2011.
29. Nedungadi, P., H. Harikumar, and M. Ramesh. *A high performance hybrid algorithm for text classification.* in *Applications of Digital Information and Web Technologies (ICADIWT), 2014 Fifth International Conference on the*. 2014.
30. Subramanya, A. and J. Bilmes, *Soft-supervised learning for text classification*, in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*2008, Association for Computational Linguistics: Honolulu, Hawaii. p. 1090-1099.
31. Shi, L., et al., *Rough set and ensemble learning based semi-supervised algorithm for text classification.* Expert Systems with Applications, 2011. **38**(5): p. 6300-6306.
32. Lee, L.H., et al., *High Relevance Keyword Extraction facility for Bayesian text classification on different domains of varying characteristic.* Expert Systems with Applications: An International Journal, 2012. **39**(1): p. 1147-1155.
33. Farhoodi, M., A. Yari, and A. Sayah. *N-gram based text classification for Persian newspaper corpus.* in *Digital Content, Multimedia Technology and its Applications (IDCTA), 2011 7th International Conference on*. 2011.
34. Meng, J., H. Lin, and Y. Li, *Knowledge transfer based on feature representation mapping for text classification.* Expert Systems with Applications: An International Journal, 2011. **38**(8): p. 10562-10567.
35. Mikawa, K.I., T.; Goto, M., *A proposal of extended cosine measure for distance metric learning in text classification.* Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on 2011: p. 1741 - 1746.
36. Wajeed, M.A.A., T., *Different similarity measures for text classification using KNN.* Computer and Communication Technology (ICCCT), 2011 2nd International Conference on, 2011: p. 41 - 45.
37. Xu, G., et al., *Improved TFIDF weighting for imbalanced biomedical text classification.* Elsevier Science Energy Procedia, 2011: p. 2360-2367.
38. Gospodnetic, O., E. Hatcher, and D. Cutting, *Lucene in action.* 2005: Mannaging.
39. Manning, C., P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*, 2008, Cambridge University Press: Cambridge, England.
40. Cobo, M.J., et al., *Science Mapping Software Tools: Review, Analysis and Cooperative Study among Tools.* Journal of the American Society for Information Science and Technology, 2011. **62**(7): p. 1382-1402.