Expert Systems with Applications 39 (2012) 12851–12857

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

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

A global-ranking local feature selection method for text categorization

Roberto H.W. Pinheiro [a], George D.C. Cavalcanti [a,*], Renato F. Correa [b], Tsang Ing Ren [a]

[a] Federal University of Pernambuco (UFPE), Center of Informatics (CIn), Av. Jornalista Anibal Fernandes s/n, Cidade Universitária, 50740-560 Recife, PE, Brazil
[b] Federal University of Pernambuco (UFPE), Departament of Information Science (DCI), Av. da Arquitetura s/n, CAC, Cidade Universitária, 50740-550 Recife, PE, Brazil

ARTICLE INFO

Keywords:
Text categorization
Feature selection
Filtering method
Variable Ranking
ALOFT

ABSTRACT

In this paper, we propose a filtering method for feature selection called ALOFT (At Least One FeaTure). The proposed method focuses on specific characteristics of text categorization domain. Also, it ensures that every document in the training set is represented by at least one feature and the number of selected features is determined in a data-driven way. We compare the effectiveness of the proposed method with the Variable Ranking method using three text categorization benchmarks (Reuters-21578, 20 Newsgroup and WebKB), two different classifiers (k-Nearest Neighbor and Naïve Bayes) and five feature evaluation functions. The experiments show that ALOFT obtains equivalent or better results than the classical Variable Ranking.

1. Introduction

Due to the growth of digital information, which are plenty available in the Internet, efficient methods to obtain relevant information are necessary. Automatic text categorization are applied to in an attempt to solve this problem. Text categorization aims to automatically assign predefined labels on previously unseen documents according to its content. This task is naturally treated as a supervised learning problem and several algorithms from Machine Learning (ML) approaches have been used in the past years, such as: decision trees (Apte, Damerau, & Weiss, 1998; Lewis & Ringuette, 1994), neural networks (De Souza et al., 2009; Wiener, Pedersen, & Weigend, 1995), k-Nearest Neighbor (kNN) (Lam & Ho, 1998; Tan, 2006), support vector machines (Godbole, Sarawagi, & Chakrabarti, 2002; Lee & Kageura, 2007), Naïve Bayes (Lewis, 1998; McCallum & Nigam, 1998).

When using ML algorithms as text categorization classifiers, documents are represented as a vector of features. A widely used approach for document content representation is the "Bag of Words" (Sebastiani, 2002), in which each word or term that appears in the documents is represented as a feature. Thus, it is common to have tens of thousands of features in a medium-sized corpus. Most of these features are irrelevant, leading to a poor performance of the classifier. Therefore, dimensionality reduction is an essential step, without it the accuracy of the text classifier is compromised, since many ML algorithms cannot handle a high number of features in a reasonable time.

One of major difficulties of text categorization is to perform dimensionality reduction of the feature space. This reduction aims to obtain a significant set of features that allow both document grouping in categories and categories discrimination. This process must be done most automatically as possible in a data-driven way, without needing human ad hoc parameters settings. There are many methods to perform dimensionality reduction. A distinction may be drawn in terms of the nature of the resulting terms: term selection methods or term extraction methods (Sebastiani, 2002).

The feature extraction methods obtain a small set of new features generated by combinations or transformations of the original ones. Latent Semantic Indexing (LSI), Principal Component Analysis (PCA) and Semantic Mapping (SM) are examples of extraction methods. These methods are based on the estimation of principal components (LSI and PCA) or term clustering (SM). A more detailed description about feature extraction methods can be found in (Correa & Ludermir, 2006).

The feature selection methods select a subset of the original set of features using a global ranking metric (Chi-Squared and Information Gain, for example) or a function of the classifier performance that use a selected feature set. Thus, there are two basic approaches to perform feature selection: filter or wrapper.

The basic idea of the wrapper methods (Kohavi & John, 1997) is to generate many different subsets of the feature – based on a defined rule or by random choice – and test each one using a classifier. The number of subsets generated can be predefined by the user, using an automatic rule that observes the behavior of the accuracy rate or by other parameters that differs from method to method. Wrapper methods can select appropriate subsets of

* Corresponding author. Tel.: +55 81 2126 8430x4346; fax: +55 81 2126 8438.
E-mail addresses: rhwp@cin.ufpe.br (R.H.W. Pinheiro), gdcc@cin.ufpe.br (G.D.C. Cavalcanti), renato.correa@ufpe.br (R.F. Correa), tir@cin.ufpe.br (T.I. Ren).
URL: http://www.cin.ufpe.br/~viisar (G.D.C. Cavalcanti).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2012.05.008

features, however the computational cost is very high, which renders these methods unfeasible for text classification problems.

On the other hand, filtering methods (Almuallim & Dietterich, 1991; Kira & Rendell, 1992; Yu & Liu, 2003) are faster than wrapper methods since these approaches select a set of features without repeatedly testing them with a classifier. Generally, Filtering methods select a subset of the features based on the Variable Ranking algorithm (VR) with a specific Feature Evaluation Function (FEF) to globally rank the features. The total number of features selected are defined as parameter by the user. After the feature selection, a classifier is applied using the chosen features.

The high dimensionality of the feature space for text categorization tasks must be taken into consideration when the decision is to use wrappers. It is well-known that wrappers are time consuming. Hence, the most frequent used approach for feature selection is filtering methods.

We present a feature selection algorithm for text categorization. The proposed algorithm is a filtering method that ensures the contribution of each document to the selection of the final feature set and the selected features covers all documents in the training set, because at least one feature per document is selected. Since many documents of the same category should select the same features, the size of the final feature set is small when compared with state of the art methods. The proposed method is called ALOFT (At Least One FeaTure).

The proposed algorithm is compared with the Variable Ranking algorithm and evaluated using Naïve Bayes and k-Nearest Neighbor classifiers. Three text categorization datasets with five different Feature Evaluation Functions (FEF) as parameter were used in the experiments. The rest of the paper is organized as follows. Section 2 describes the classical approach used in feature selection by filtering methods, it includes a description of Variable Ranking algorithm and several Feature Evaluation Functions. Section 3 presents the proposed method in details. Section 4 describes the methodology of the experiments, the text categorization benchmarks, the applied ML classifiers, and the metrics and methods to measure the text categorization effectiveness. Section 5 reports the experimental results and analysis. Finally, Section 6 presents the conclusion.

2. Feature selection for text categorization

Even though feature selection algorithms are divided in filter and wrapper methods, we centered the efforts here in the filtering approach. The reason for this choice lies in the scalability of this methods which is a required characteristic when dealing with problems that deals with many features as in text categorization. The filtering approach consists in ranking each feature based on a Feature Evaluation Function (FEF) and selecting the n features that have the highest scores (Forman, 2003; Yang & Pedersen, 1997). Each feature is evaluated by the chosen FEF, the output of this function represents the degree of significance which describes and discriminates the categories. The input parameter n is given by the user or determined experimentally by trial and error.

The feature selection task can be divided into C binary problems (Mladenic & Grobelnik, 1999), in which C is the number of classes. In this case, a different set of features is used for each binary problem. Thus, C classifiers must be trained. In this paper, the feature selection task is treated as a multiclass problem (Chen, Huang, Tian, & Qu, 2009), in which the whole set of feature is used but only one classifier is required.

Section 2.1 shows a classical algorithm used to select features for text categorization and, in Section 2.2, five applied FEFs are described.

2.1. Classical approach

Algorithm 1: *Variable ranking*

```
Require: Integer $n > 0$
1: Load training set $\mathcal{D}_{tr}$
2: for $h = 1$ to $V$ do {Calculate FEF for each term}
3:     $S_h = \text{FEF}(w_h)$
4: end for
5: for $i = 1$ to $n$ do {Select $n$ features with the highest FEF}
6:     $SN = S$
7:     $bestscore = 0.0$
8:     for $h = 1$ to $V$ do
9:         if $SN_h > bestscore$ then
10:            $bestscore = SN_h$
11:            $bestfeature = h$
12:        end if
13:    end for
14:    $SN_{bestfeature} = 0$
15:    $FS_i = bestfeature$
16: end for
17: $\mathcal{D}_{nv}$ an empty dataset
18: for all $d \in \mathcal{D}_{tr}$ do
19:    Insert document $d'$ in $\mathcal{D}_{nv}$
20:    for $h = 1$ to $n$ do
21:        $d'_h = d_{FS_h}$
22:    end for
23: end for
```

Algorithm 1 shows the pseudo-code for the classical approach called Variable Ranking (VR) (Guyon & Elisseeff, 2003). The algorithm is divided in three parts: FEF calculation (lines 2–4); selection of the n best ranked features based on the FEF values (lines 5–16); and, construction of the new training set using the selected features (lines 17–23).

The classical approach presents some disadvantages. The first one is the effort required to find the best value for n. Since it is necessary several tests using different values of n in order to obtain the optimal number. The second problem occurs when the final set of features is small. The classical approach is based only on the global values of FEF. Thus, the chosen features may be too generic and appear in more than one category. It is expected that when the same feature is shared by many categories, the discrimination power of this feature is decreased. Moreover, the feature set selected by the classical approach may not cover the entire training set. In other words, empty vectors can be produced by the selection procedure and these vectors are misclassified.

2.2. Feature selection functions

Several FEFs have been proposed over the years and comparative studies are described in the literature (Forman, 2003; Mladenic & Grobelnik, 1999; Rogati & Yang, 2002; Yang & Pedersen, 1997). In this section, five FEFs are described and applied in the multiclass problem.

The following nomenclature is adopted: w is the evaluated feature (word or term), $P(w|c_j)$ is the probability of the word w to occur in class c_j, $P(w|\bar{c}_j)$ is the probability that the word w does not occur in class c_j, $P(\bar{w}|c_j)$ is the probability that every word but w occurs in class c_j, $P(\bar{w}|\bar{c}_j)$ is the probability that every word but w does not occur in class c_j and $P(c_j)$ is the probability of the class c_j in general.

Proposed by Forman (2003), the *Bi-Normal Separation* (BNS) measures the separation between two thresholds calculated using the Normal distribution inverse cumulative probability function (z-score). To avoid the undefined value $F^{-1}(0)$, zero is substituted by 0.0005.

$$\mathrm{BNS}(w) = \sum_{j=1}^{C} |F^{-1}(P(w|c_j)) - F^{-1}(P(w|\bar{c}_j))| \qquad (1)$$

Class Discriminating Measure (CDM) is a derivation of Odds Ratio introduced by Chen et al. (2009) and is defined as:

$$\mathrm{CDM}(w) = \sum_{j=1}^{C} \left| \log \frac{P(w|c_j)}{P(w|\bar{c}_j)} \right| \qquad (2)$$

The well-known Chi-Squared (CHI) measures how independent w is from each class (Debole & Sebastiani, 2003). CHI presented good performance as demonstrated by Rogati and Yang (2002).

$$\mathrm{CHI}(w) = \sum_{j=1}^{C} \frac{[P(w|c_j)P(\bar{w}|\bar{c}_j) - P(w|\bar{c}_j)P(\bar{w}|c_j)]^2}{P(w)P(\bar{w})P(c_j)P(\bar{c}_j)} \qquad (3)$$

Combined with CHI, Information Gain (IG) was reported as one of the best FEFs for multiclass problems (Yang & Pedersen, 1997).

$$\mathrm{IG}(w) = \sum_{j=1}^{C} P(w|c_j) \log \frac{P(w|c_j)}{P(c_j)} + \sum_{j=1}^{C} P(\bar{w}|c_j) \log \frac{P(\bar{w}|c_j)}{P(c_j)} \qquad (4)$$

Since, we are considering text categorization a multiclass problem, Multiclass Odds Ratio (MOR) (Chen et al., 2009) is used instead of the original binary version of Odds Ratio.

$$\mathrm{MOR}(w) = \sum_{j=1}^{C} \left| \log \frac{P(w|c_j)(1 - P(w|\bar{c}_j))}{P(w|\bar{c}_j)(1 - P(w|c_j))} \right| \qquad (5)$$

These FEFs are used as parameter to Variable Ranking and to the proposed feature selection methods in the text categorization experiments.

3. Proposed method

To deal with the problems described in Section 2.1, we introduce a feature selection method called ALOFT (At Least One FeaTure). ALOFT is a heuristic method that selects features for text categorization based on the Bag of Words approach for document content representation. The central idea of this method is to search for a set of features that ensures full coverage of the documents in the training set, i.e., at least one feature per document must be part of the final feature set. Moreover, ALOFT must automatically find the optimal number of features. Based on this strategy, the proposed method guarantees the following points:

- Each document is represented in the feature vector by at least one valued feature (a valued feature is a feature that has nonzero weight and is positive). Thus, all documents in the training set should contribute to the final feature set.
- The algorithm automatically finds the optimal number of features in a data-driven way without a preliminary optimization that searches for the best number of features.
- For a given training set, the algorithm finds at most d features (upper bound), where d is the number of documents.
- Given any FEF, the algorithm finds the lowest number of features that covers all documents in the training set.
- When compared with the classical approach, ALOFT does not require parameter optimization or preliminary tests to find the optimal input parameters. Only the FEF function must be chosen.
- The algorithm is fast and deterministic. Thus, it provides a single solution for FEF and training set.

Algorithm 2: ALOFT

```
1: Load training set $\mathcal{D}_{tr}$
2: for $h = 1$ to $V$ do {Calculate FEF for each term}
3:    $S_h$ = FEF ($w_h$)
4: end for
5: $m = 0$
6: Set $FS$ an empty vector
7: for all $d_i \in \mathcal{D}_{tr}$ do {Select, for each document, the valued
   feature with the highest FEF}
8:    $bestscore = 0.0$
9:    for $h = 1$ to $V$ do
10:       if $w_{h,i} > 0$ and $S_h > bestscore$ then
11:          $bestscore = S_h$
12:          $bestfeature = h$
13:       end if
14:    end for
15:    if $bestfeature \notin FS$ then
16:       $m = m + 1$
17:       $FS_m = bestfeature$
18:    end if
19: end for
20: Set $\mathcal{D}_{nv}$ an empty dataset
21: for all $d \in \mathcal{D}_{tr}$ do
22:    Insert document $d'$ in $\mathcal{D}_{nv}$
23:    for $h = 1$ to $m$ do
24:       $d'_h = d_{FS_h}$
25:    end for
26: end for
```

Algorithm 2 shows the pseudo-code of the proposed feature selection method. A description of the algorithm is given as follows:

- Line 1: A training set $\mathcal{D}_{tr}$ is loaded. The set is composed of $d \in \mathbb{N}^V$ documents, V is the size of the vocabulary (number of features);
- Lines 2–4: For each feature w_h, the FEFs values are calculated and stored in S_h. Thus, S_h represents the importance of the hth feature and $S \in \mathbb{R}^V$;
- Lines 5–19: The new set of features FS is computed. The hth feature is inserted in FS if it is the highest S_h value among all features. However, if this feature is already in FS, it is ignored and the algorithm continues to the next document. At the end of this phase, FS should be a vector with m values, and these values represent the index of the selected features;
- Lines 20–26: The new training set $\mathcal{D}_{nv}$ is constructed. It is composed of $d' \in \mathbb{N}^m$ documents, having m representing the number of selected features. The test set can be generated using the same procedure.

3.1. A toy example

In order to illustrate how the proposed method works, a hypothetical training set (Table 1) composed of 13 documents represented by 9 boolean features (presence or absence of the word) was constructed. We define S to represent the importance of each feature as in Algorithm 2. However, for simplicity, S is defined as a vector of integer values.

The first step of the proposed method (ALOFT) is to calculate the values of the S vector; any FEF can be used. For this example, the values in the table are all hypothetical.

The second step is to select the best features. For each document, the best valued feature is selected based on the S vector.

Table 1
A toy example. The first column (D) represents an index to identify each document, the last column (C) represents the category of the document and the columns between (w_i) are the features. Each line represents one document, except the last one, which represents the S vector.

D	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	C
1	0	1	0	0	0	0	1	0	0	A
2	0	0	0	1	1	0	0	1	0	A
3	1	0	0	1	0	0	0	1	1	A
4	0	1	0	0	0	0	0	0	0	A
5	0	0	0	0	0	0	1	1	0	A
6	0	0	0	0	0	1	0	0	0	B
7	0	0	0	1	0	1	0	0	1	B
8	1	0	1	1	1	0	0	0	1	B
9	0	0	0	0	0	1	0	0	0	B
10	0	0	0	0	0	1	1	0	0	B
11	1	0	1	1	1	0	1	0	0	B
12	0	0	0	1	1	0	0	0	1	B
13	1	0	0	1	0	1	0	0	0	B
S	11	7	4	15	10	8	2	5	13	–

Analyzing the first document (D1), we notice two valued features: w_2 and w_7. The weight w_2 is select because $S_2 = 7$ is greater than $S_7 = 2$, so the index of this word is inserted in the vector $FS = \{2\}$. For Document two, w_4 is selected then $FS = \{2,4\}$. For Document three, w_4 is selected again, however that index is already in FS. In this case, nothing is done and the procedure continues. For Document four, the same happens, but now with w_2. For Document five, w_8 is selected and $FS = \{2,4,8\}$. For Document six, w_6 is selected, then $FS = \{2,4,8,6\}$. From Document 7 to Document 13, no feature is added to FS.

The classical approach selects n features with the highest S. Since ALOFT found 4 features, we use $n = 4$ for a fair comparison. Thus, the classical approach selects w_4,w_9,w_1 and w_5, obtaining $FS = \{4,9,1,5\}$.

Table 2 shows the final feature vectors for the classical and the proposed approaches. For this example, the classical approach presents some problems. For example: six documents (D1, D4, D5, D6, D9 and D10) have no valued feature. Moreover, the selected features are too general, in other words, they can not be used to distinguish one category from another since all features appear in both categories. Some documents have very similar feature vectors even though they belong to different classes, such as: D3 (category A) and D8 (category B) differ only in the last feature w_5. Another example is found analyzing documents D2, D11 and D12.

However, ALOFT does not present these problems. All documents have at least one valued feature and some features are present only in a specific category, for example: w_2 and w_8 are present only in category A; and, w_6 is present only in category B. This is a clear advantage when the objective is classification.

Table 2
The selected features for the classical and the proposed approaches using the training set showed in Table 1.

D	Classical approach				Proposed method				C
	w_4	w_9	w_1	w_5	w_2	w_4	w_8	w_6	
1	0	0	0	0	1	0	0	0	A
2	1	0	0	1	0	1	1	0	A
3	1	1	1	0	0	1	1	0	A
4	0	0	0	0	1	0	0	0	A
5	0	0	0	0	0	0	1	0	A
6	0	0	0	0	0	0	0	1	B
7	1	1	0	0	0	1	0	1	B
8	1	1	1	1	0	1	0	0	B
9	0	0	0	0	0	0	0	1	B
10	0	0	0	0	0	0	0	1	B
11	1	0	1	1	0	1	0	0	B
12	1	1	0	1	0	1	0	0	B
13	1	0	1	0	0	1	0	1	B

4. Experimental settings

We experimentally compare ALOFT with state-of-the-art feature selection methods, which is a result from the combination of Variable Ranking (VR) with five traditional FEFs as presented in Section 2.2. To obtain a fair comparison, the number of features used by the Classical Approach should be the same found by ALOFT when they use the same FEF. In Subsection 4.1 the adopted classifiers are introduced, the data sets are described in Subsection 4.2 and a description of the evaluation methodology is given in Subsection 4.3.

4.1. Classifiers

k-Nearest Neighbor and Naïve Bayes are the classifiers used in this experiment. Besides their simplicity, they are interesting classifiers to evaluate the performance of feature selection methods because both are strongly influenced by the selected features. They are described in the following sections.

4.1.1. k-Nearest Neighbor

To classify an unknown document d_i, the kNN classifier determines its class label as:

$$\text{label}(d_i) = \arg\max_{c_j} \sum_{d_k \in kNN(d_i)} \delta(d_k, c_j) \tag{6}$$

where $\delta(d_k,c_j)$ is a binary function used for the classification of the document d_k with respect to the class c_j, and is defined as:

$$\delta(d_k, c_j) = \begin{cases} 1, & d_k \in c_j \\ 0, & d_k \notin c_j \end{cases} \tag{7}$$

and $kNN(d_i)$ is a classifier that returns the k-Nearest Neighbors of document d_i. Different measure can be used to find the neighbors, particularly for text categorization, the cosine distance is commonly used instead of Euclidean distance because it is usual that the data presents lots of features with zero weight. The cosine distance is defined as:

$$\text{cosine}(x, y) = \frac{\sum_{i=1}^{V} x_i y_i}{\sqrt{\sum_{i=1}^{V} x_i^2}\sqrt{\sum_{i=1}^{V} y_i^2}} \tag{8}$$

where V denotes the size of feature vector from documents x and y.

4.1.2. Naïve Bayes

There are two different models of Naïve Bayes classifiers for text categorization: Multi-Variate Bernoulli Event Model and Multinomial Event Model (McCallum & Nigam, 1998). In this paper, we choose the second model because it shows a better performance for text categorization.

To classify an unknown document d_i, the Naïve Bayes determines the class label as:

$$\text{label}(d_i) = \arg\max_{c_j}\{P(c_j|d_i)\} \tag{9}$$

using the Bayes rule:

$$P(c_j|d_i) = \frac{P(d_i|c_j)P(c_j)}{P(d_i)} \tag{10}$$

$P(d_i)$ is equal for all classes, so, this term can removed from the equation, simplifying the decision rule:

$$\text{label}(d_i) = \arg\max_{c_j}\{P(d_i|c_j)P(c_j)\} \tag{11}$$

The probability $P(c_j)$ can be calculated dividing the number of documents in class c_j by the number of documents in all corpus. The probability $P(d_i|c_j)$ is computed as:

$$P(d_i|c_j) = P(|d_i|)|d_i|!\prod_{k=1}^{V}\frac{P(w_k|c_j)^{n_{ik}}}{n_{ik}!} \tag{12}$$

where $|d_i|$ is the sum of all the weights in document d_i,V is the size of the feature vector and n_{ik} is the number of appearance of word w_k in document d_i. The probability $P(w_k|c_j)$ is estimated using equation:

$$P(w_k|c_j) = \frac{1 + N_{c_jk}}{V + N_j} \tag{13}$$

where N_{c_jk} is the number of word w_k in class c_j and N_j is the number of words in class c_j.

4.2. Data collection

Three datasets with different characteristics were employed to analyze the behavior of the proposed method using different types of data:

- The Reuters-21578 Collection[1] contains documents collected from the Reuters newswire in 1987. It is a standard text classification benchmark and contains 135 categories in the original version. We adopted a subset of the top ten categories having 9980 documents. This configuration is also adopted by many previous works (Chang, Chen, & Liau, 2008; Chen et al., 2009; Shang et al., 2007). The feature vector contains 10,987 words;
- The 20 Newsgroup corpus[2] contains 19,997 articles taken from the Usenet newsgroup collection. All documents are used in the experiments (Bekkerman, El-Yaniv, Tishby, & Winter, 2003; Nigam, McCallum, Thrun, & Mitchell, 1998; Xue & Zhou, 2009). The feature vector contains 46,834 words;
- The WebKB corpus[2] is a collection of 8282 web pages obtained from four academic domains. The original data set has seven categories, but only four of them are used: course, faculty, project and student. This subset contains 4199 documents and was introduced by Nigam et al. (1998). This reduced database is also used by other researchers (Bekkerman et al., 2003; Xue & Zhou, 2009). The feature vector contains 21,324 words.

4.3. Evaluation methodology

We measure the effectiveness of the methods using the micro averaged and macro averaged F1 (Sebastiani, 2002). The performance of the F1 classifier for a category is a combination of precision and recall. When effectiveness is computed for several categories, the results for individual categories must be averaged. For the computation of the Micro averaged F1 (Micro-F1) the categories count is proportional to the number positive examples, while in the macro averaged F1 all categories count are considered the same. Micro averaged F1 (Macro-F1) is dominated by F1 for common categories while macro averaged F1 is dominated by F1 for rare categories. Micro-F1 can be calculated as:

$$\text{Micro} - \text{F1} = \frac{2 \times R_{\text{micro}} \times P_{\text{micro}}}{(R_{\text{micro}} + P_{\text{micro}})} \tag{14}$$

and the definitions of micro-Precision and micro-Recall are respectively:

$$P_{\text{micro}} = \frac{\sum_{j=1}^{C} TP_j}{\left(\sum_{j=1}^{C} TP_j + \sum_{j=1}^{C} FN_j\right)} \tag{15}$$

$$R_{\text{micro}} = \frac{\sum_{j=1}^{C} TP_j}{\left(\sum_{j=1}^{C} TP_j + \sum_{j=1}^{C} FP_j\right)} \tag{16}$$

where C is the number of existing class, TP_j is the number of correctly classified documents for class c_j, FP_j is the number of incorrectly classified documents for class c_j and FN_j is the number of incorrectly classified documents to other class else j.

Macro-F1 s defined similarly as Micro-F1:

$$\text{Macro} - \text{F1} = \frac{2 \times R_{\text{macro}} \times P_{\text{macro}}}{(R_{\text{macro}} + P_{\text{macro}})} \tag{17}$$

but using macro values for precision:

$$P_{\text{macro}} = \frac{\sum_{j=1}^{C} P_j}{C} \tag{18}$$

$$P_j = \frac{TP_j}{(TP_j + FN_j)} \tag{19}$$

And recall:

$$R_{\text{macro}} = \frac{\sum_{j=1}^{C} R_j}{C} \tag{20}$$

$$R_j = \frac{TP_j}{(TP_j + FP_j)}. \tag{21}$$

where P_j and R_j are the values for a single class and j is the index of that class.

We use the t-test of the combined variance to compare the performance of the methods (Correa & Ludermir, 2006). The t-test is applied on the average and the standard deviation of the micro and macro F1 obtained by each method in the test set on a 10-fold cross-validation experiment.

For the comparison of the methods performance using t-test, the following convention for the P-value are used: "$\gg$" and "$\ll$" mean that the P-value is lesser than or equal to 0.01, indicating a strong evidence that a method results in a greater or minor value for the effectiveness measure than another method, respectively; "$>$" and "$<$" mean that the P-value is greater than 0.01 and lesser or equal to 0.05, indicating a weak evidence that a method results in a greater or minor value for the effectiveness measure than another method; "$\sim$" means that the P-value is greater than 0.05 indicating that it does not have significant difference when compared the performance of two different method.

5. Analysis of experiments

5.1. The effectiveness of the text categorization

Table 3 shows the results of the experiment applied on three datasets for kNN and Naïve Bayes using features selected by ALOFT and VR with five FEFs as parameter. The maximum mean values of Micro-F1 and Macro-F1 (without significant difference to the maximum value) obtained by ALOFT and VR in each dataset are described in bold. The FEF that maximize the performance of both ALOFT and VR for all the datasets is CHI. The FEF that minimize the performance of ALOFT is IG and for VR is CDM. With the goal of maximizing the effectiveness in text categorization and minimizing the number of chosen features, the best choice of the FEF is CHI for ALOFT.

For each dataset, a different FEF maximize the mean of Micro-F1 and Macro-F1 for ALOFT, VR and both kNN and NB classifiers: CHI for 20 Newsgroup, MOR for Reuters10, and BNS for WebKB. For each dataset, the classifier that shows the best performance for

[1] Available at http://dit.unitn.it/~moschitt/corpora.htm.

[2] Available at http://www.cs.cmu.edu/textlearning/.

Table 3
Performance of all the presented methods.

Dataset	Classifier	FEF	*m*	ALOFT		VR	
				macro-F1	micro-F1	macro-F1	micro-F1
20 Newsgroup	k-NN	BNS	18 (1.00)	69.53 (4.79)	71.52 (4.22)	44.26 (1.24)	52.39(1.75)
		CHI	32 (0.00)	**93.41**(**0.90**)	**93.41**(**0.91**)	**93.41** (**0.90**)	**93.41**(**0.91**)
		CDM	393 (8.10)	**93.74** (**0.94**)	**93.74** (**0.92**)	87.18(1.09)	87.61 (0.98)
		IG	10 (1.73)	51.07 (1.95)	54.83 (1.84)	51.98 (4.72)	55.67(4.27)
		MOR	129 (46.45)	91.00 (1.69)	91.03 (1.63)	88.35 (1.65)	88.47 (1.53)
	NB	BNS	18 (1.00)	66.05 (5.25)	68.63 (4.71)	43.01 (0.74)	53.46 (0.70)
		CHI	32 (0.00)	**91.88** (**1.08**)	**92.01**(**1.02**)	**91.88** (**1.08**)	**92.01**(**1.02**)
		CDM	393 (8.10)	**91.78** (**0.67**)	**91.90** (**0.65**)	84.15 (1.03)	84.93 (0.89)
		IG	10 (1.73)	49.95 (1.84)	54.77 (1.63)	52.59 (2.82)	56.31 (2.37)
		MOR	129 (46.45)	87.66 (1.98)	87.73 (1.89)	84.47 (1.39)	84.70 (1.20)
Reuters10	k-NN	BNS	142 (3.74)	**57.17** (**4.00**)	**79.51**(**1.97**)	52.84 (3.24)	**75.91**(**2.04**)
		CHI	67 (2.89)	**55.66**(**4.09**)	**78.97**(**2.35**)	**55.45**(**2.66**)	**76.28**(**2.19**)
		CDM	243 (7.42)	**56.97** (**3.95**)	**79.78**(**2.10**)	**53.91**(**4.83**)	**77.56**(**2.23**)
		IG	64 (3.37)	53.42 (3.40)	78.22 (2.00)	**54.74**(**2.54**)	**77.27**(**1.35**)
		MOR	135 (5.26)	**57.54**(**2.91**)	**80.35**(**2.07**)	**55.02**(**3.61**)	**77.70**(**2.83**)
	NB	BNS	142 (3.74)	**60.95**(**3.33**)	**80.29**(**1.49**)	57.81(3.36)	**77.03** (**2.94**)
		CHI	67 (2.89)	**59.62**(**5.41**)	**79.35**(**2.17**)	**61.56** (**3.79**)	**77.58** (**2.48**)
		CDM	243 (7.42)	**60.91** (**3.86**)	**81.11** (**2.74**)	54.01 (3.99)	**77.99** (**2.56**)
		IG	64 (3.37)	57.27 (3.67)	78.48 (2.18)	**60.57** (**3.60**)	**78.07** (**2.50**)
		MOR	135 (5.26)	**62.13**(**3.56**)	**80.47** (**1.57**)	**61.40** (**3.59**)	**78.25**(**2.56**)
WebKB	k-NN	BNS	151 (11.22)	**80.24** (**2.70**)	**82.11** (**2.80**)	**75.37** (**5.15**)	**79.43** (**3.63**)
		CHI	40 (3.21)	**79.10** (**6.29**)	**81.66**(**4.56**)	**75.22** (**2.15**)	**79.26**(**2.76**)
		CDM	622 (13.66)	76.47 (3.99)	**80.26**(**3.90**)	55.85 (5.51)	63.86 (5.48)
		IG	41 (3.65)	**78.81** (**3.84**)	**81.67**(**3.37**)	**74.52**(**3.29**)	**78.88**(**3.73**)
		MOR	202 (7.37)	**78.95**(**3.71**)	**81.50**(**3.11**)	**73.38**(**4.12**)	76.38 (3.23)
	NB	BNS	151 (11.22)	**82.34 (5.62)**	**84.31**(**4.94**)	**76.47**(**4.77**)	**81.23**(**3.34**)
		CHI	40 (3.21)	**79.09** (**6.27**)	**82.28**(**4.18**)	**73.63**(**4.67**)	**78.78**(**3.47**)
		CDM	622 (13.66)	79.15 (3.14)	**83.36**(**2.22**)	60.40 (3.47)	71.12 (3.10)
		IG	41 (3.65)	**78.36** (**6.26**)	**81.86**(**4.48**)	**74.61**(**3.17**)	**79.28** (**2.83**)
		MOR	202 (7.37)	**82.57**(**3.99**)	**84.55**(**3.42**)	**76.87**(**3.60**)	**80.64**(**3.24**)

Table 4
ALOFT × VR: t-test results.

Dataset	Classifier	Measure	Feature evaluation functions				
			BNS	CHI	CDM	IG	MOR
20 Newsgroup	*k*NN	macro-F1	≫	∼	≫	∼	≫
		micro-F1	≫	∼	≫	∼	≫
	NB	macro-F1	≫	∼	≫	<	≫
		micro-F1	≫	∼	≫	∼	≫
Reuters10	*k*NN	macro-F1	>	∼	∼	∼	∼
		micro-F1	≫	>	>	∼	>
	NB	macro-F1	>	∼	≫	<	∼
		micro-F1	≫	∼	>	∼	>
WebKB	*k*NN	macro-F1	>	>	≫	>	≫
		micro-F1	>	∼	≫	∼	≫
	NB	macro-F1	>	>	≫	∼	≫
		micro-F1	∼	>	≫	∼	>

both ALOFT and VR are k-NN for 20 Newsgroup, NB for Reuters10 and NB for WebKB.

Table 4, which is derived form Table 3, shows the t-test results when comparing the performance of ALOFT versus VR. ALOFT has the same or better performance than VR in 97% of the cases (58 of 60 comparisons). In 62% of the cases (37 of 60 comparisons), ALOFT has superior performance than VR. We can see that the CDM, MOR and BNS Feature Evaluation Functions improve the results of ALOFT rather than VR. ALOFT has weak significant inferior results than VR in 3% of the cases (2 of 60 comparisons) using the FEF IG with NB classifier in Macro-F1 effectiveness measure.

Fig. 1 shows all the Micro-F1 means from Table 3 in a plot, providing a general view of the results from the experiments. We can observe that the VR results are worst than the ALOFT results specially in the case of high number of features. Most of the ALOFT samples (20 of 30) has mean Micro-F1 bigger than 80 and most of VR samples (22 of 30) has mean Micro-F1 inferior to 80. Observing the samples plotted in the same number of features, (two values for ALOFT and two for VR for a given pair dataset-FEF) it is possible to confirm that the mean values of Micro-F1 for ALOFT has shows to be higher than the ones of VR.

6. Conclusion

In this paper, a filtering method for feature selection called ALOFT is proposed. One advantage of the method is that it requires as parameter only a FEF, this represents an advantage since it is not necessary to tune the number of features to be selected.

The experiment shows the effectiveness of the ALOFT approach. The performance of the proposed method is compared with the

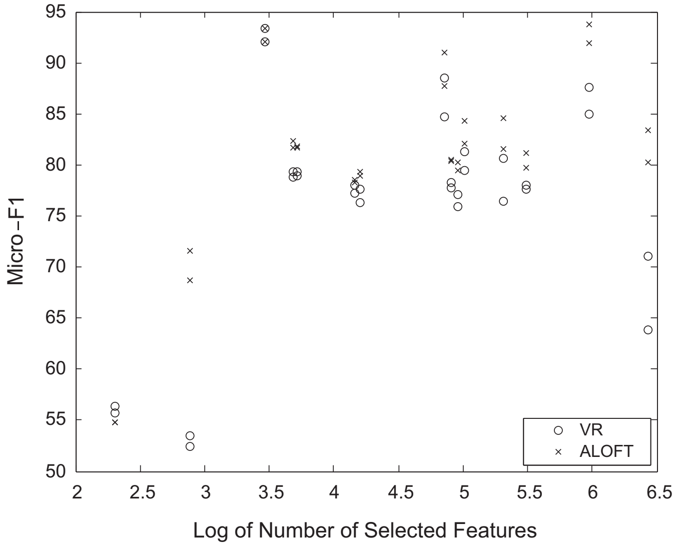

Fig. 1. Mean Micro-F1 for ALOFT and VR runs ignoring differences among FEF, classifiers and corpus.

well established Variable Ranking method for feature selection in text categorization (Forman, 2003; Mladenic & Grobelnik, 1999; Rogati & Yang, 2002; Yang & Pedersen, 1997). Experimental results on three corpora show that the proposed method needs less number of features to cover all training set and it achieves similar or better performance than VR, depending of the dataset and FEF. For all the datasets, the best or near-best results of ALOFT are generated using CHI as FEF.

For future work, we plan to conduct further experiments, using another FEFs, to obtain an optimal balance between performance and number of selected features. Additionally, a wider comparison using more datasets, classification algorithms and FEFs would be preferred as the performance of filter feature selectors can vary with the classification algorithm and dataset chosen.

Both the ALOFT and VR technique are applicable to a wide range of Feature Evaluation Functions. In this work, both ALOFT and VR use univariate ranking, without taking into account interactions between the features. Feature interactions are commonly found, either where one feature turns another redundant, or where two features are combine to obtain a more predictive power than the sum of their univariate powers. There are bi-variate ranking algorithms, which take into account pairwise feature interactions, and these will be investigate in future works.

References

Almuallim, H., & Dietterich, T. G. (1991). Learning with many irrelevant features. In *AAAI* (pp. 547–552).

Apte, C., Damerau, F., & Weiss, S. (1998). Text mining with decision trees and decision rules. In *Workshop on learning from text and the web – Conference on automated learning and discovery*.

Bekkerman, R., El-Yaniv, R., Tishby, N., & Winter, Y. (2003). Distributional word clusters vs. words for text categorization. *Journal of Machine Learning Research, 3*, 1183–1208.

Chang, Y., Chen, S., & Liau, C. (2008). Multilabel text categorization based on a new linear classifier learning method and a category-sensitive refinement method. *Expert Systems with Applications, 34*, 1948–1953.

Chen, J., Huang, H., Tian, S., & Qu, Y. (2009). Feature selection for text classification with Naive Bayes. *Expert Systems with Applications, 36*, 5432–5435.

Correa, R. F., & Ludermir, T. B. (2006). Improving self-organization of document collections by semantic mapping. *Neurocomputing, 70*, 62–69.

De Souza, A., Pedroni, F., Oliveira, E., Ciarelli, P., Henrique, W., Veronese, L., et al. (2009). Automated multi-label text categorization with VG-RAM weightless neural networks. *Neurocomputing, 72*, 2209–2217.

Debole, F., & Sebastiani, F. (2003). Supervised term weighting for automated text categorization. In *ACM symposium on applied computing* (pp. 784–788).

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

Godbole, S., Sarawagi, S., & Chakrabarti, S. (2002). Scaling multi-class support vector machines using inter-class confusion. In *ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 513–518).

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

Kira, K., & Rendell, L. (1992). The feature selection problem: Traditional methods and a new algorithm. In *National conference on artificial intelligence* (pp. 129–129).

Kohavi, R., & John, G. (1997). Wrappers for feature subset selection. *Artificial Intelligence, 97*, 273–324.

Lam, W., & Ho, C. (1998). Using a generalized instance set for automatic text categorization. In *ACM SIGIR conference on research and development in information retrieval* (pp. 81–89).

Lee, K., & Kageura, K. (2007). Virtual relevant documents in text categorization with support vector machines. *Information Processing & Management, 43*, 902–913.

Lewis, D. (1998). Naive (Bayes) at forty: The independence assumption in information retrieval. In *European conference on machine learning* (pp. 4–15).

Lewis, D., & Ringuette, M. (1994). A comparison of two learning algorithms for text categorization. In *Symposium on document analysis and information retrieval* (Vol. 33, pp. 81–93).

McCallum, A., & Nigam, K. (1998). A comparison of event models for naive bayes text classification. In *Workshop on learning for text categorization* (pp. 41–48).

Mladenic, D., & Grobelnik, M. (1999). Feature selection for unbalanced class distribution and naive bayes. In *International conference on machine learning* (pp. 258–267).

Nigam, K., McCallum, A., Thrun, S., & Mitchell, T. (1998). Learning to classify text from labeled and unlabeled documents. In *National conference on artificial intelligence* (pp. 792–799).

Rogati, M., & Yang, Y. (2002). High-performing feature selection for text classification. In *International conference on information and knowledge management* (pp. 659–661).

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

Shang, W., Huang, H., Zhu, H., Lin, Y., Qu, Y., & Wang, Z. (2007). A novel feature selection algorithm for text categorization. *Expert Systems with Applications, 33*, 1–5.

Tan, S. (2006). An effective refinement strategy for KNN text classifier. *Expert Systems with Applications, 30*, 290–298.

Wiener, E., Pedersen, J., & Weigend, A. (1995). A neural network approach to topic spotting. In *Symposium on document analysis and information retrieval* (Vol. 332, pp. 317–332).

Xue, X., & Zhou, Z. (2009). Distributional features for text categorization. *IEEE Transactions on Knowledge and Data Engineering, 21*, 428–442.

Yang, Y., & Pedersen, J. (1997). A comparative study on feature selection in text categorization. In *International conference on machine learning* (pp. 412–420).

Yu, L., & Liu, H. (2003). Feature selection for high-dimensional data: A fast correlation-based filter solution. In *International conference on machine leaning* (pp. 856–863).