Using Typical Testors for Feature Selection in Text Categorization

Aurora Pons-Porrata[1], Reynaldo Gil-García[1], and Rafael Berlanga-Llavori[2]

[1] Center of Pattern Recognition and Data Mining,
Universidad de Oriente, Santiago de Cuba, Cuba
{aurora,gil}@csd.uo.edu.cu
[2] Computer Science,
Universitat Jaume I, Castellón, Spain
berlanga@lsi.uji.es

Abstract. A major difficulty of text categorization problems is the high dimensionality of the feature space. Thus, feature selection is often performed in order to increase both the efficiency and effectiveness of the classification. In this paper, we propose a feature selection method based on Testor Theory. This criterion takes into account inter-feature relationships. We experimentally compared our method with the widely used information gain using two well-known classification algorithms: *k*-nearest neighbour and Support Vector Machine. Two benchmark text collections were chosen as the testbeds: Reuters-21578 and Reuters Corpus Version 1 (RCV1-v2). We found that our method consistently outperformed information gain for both classifiers and both data collections, especially when aggressive feature selection is carried out.

Keywords: feature selection, typical testors, text categorization.

1 Introduction

Text Categorization (TC - also known as text classification) is the task of assigning documents to one or more predefined categories (classes or topics). This task relies on the availability of an initial corpus of classified documents under these categories (known as training data). Depending on the application, TC may be either single-label (i.e., exactly one category must be assigned to each document) or multi-label (i.e., several categories can be assigned to each document).

Text Categorization is an important component in many information management tasks such as spam filtering, real time sorting of email or files into folders, document routing, document dissemination, topic identification, classification of Web pages and automatic building of Yahoo!-style catalogs. That is why during the last decade there has been a great interest from researchers.

TC literature mainly relies on Machine Learning methods such as probabilistic classifiers, decision trees, nearest neighbour classifiers, support vector machines and classifier committees, to mention a few. However, a major difficulty of text categorization problems is the high dimensionality of the feature space, which can be tens or hundreds of thousands of terms for even a moderate-sized text collection.

L. Rueda, D. Mery, and J. Kittler (Eds.): CIARP 2007, LNCS 4756, pp. 643–652, 2007.

Most of the features are irrelevant and others introduce noise that diminishes the classifier effectiveness. Thus, feature selection becomes a crucial task for improving both the efficiency and effectiveness of the classification algorithms. Moreover, feature selection techniques reduce overfitting (i.e., the tendency of the algorithm to better classify the data it has been trained on than new unseen data), and makes the problem more manageable for the classifier.

Let τ be the original set of features and φ a certain feature selection criterion function. Without any loss of generality, let us consider a higher value of φ to indicate a better feature subset. Formally, the problem of feature subset selection consists of finding a subset $\tau' \subseteq \tau$ such that $|\tau'| \ll |\tau|$ and $\varphi(\tau') = \max_{t \subseteq \tau} \varphi(t)$ [1].

According to John et al. [2] there are two main approaches to feature subset selection used in Machine Learning: wrapper and filtering. The idea of the wrapper approach is to select feature subset using the evaluation function based on the same algorithm that will be used for learning on domain represented with selected feature subset. This can result in a rather time consuming process, since, for each candidate feature subset that is evaluated during the search, the target learning algorithm is run usually several times. This approach has the advantage of being tuned to the learning algorithm being used. However, the sheer size of the space of different term sets makes its cost-prohibitive for standard TC applications. On the contrary, in the filtering approach a feature subset is selected independently of the learning method that will use it. It keeps terms that receive the highest score according to a function that measures the "importance" of the term for the TC tasks. Because of computational complexity the filtering approach is preferable over the wrapper approach to feature subset selection in TC. We will explore this solution in this paper.

The filtering methods can be also divided into two categories: Best Individual Feature (BIF) selection methods and global subset selection methods. In the former, some evaluation function that is applied to a single feature is used. All the features are independently evaluated, a score is assigned to each of them and the features are sorted according to the assigned score. Then, a predefined number of the best features is taken to form the best feature subset. BIF selection methods completely ignore the existence of other words and the manner how the words work together. Scoring of individual features can be performed using some of the measures used in machine learning, for instance: information gain [3], document frequency, mutual information, χ^2 statistic [4] and odds-ratio [5]. The mathematical definitions of these functions are summarized in [6]. Yang et al. [4], Mladenic et al. [7], Rogati et al. [8] and Forman [9] give experimental comparison of the above mentioned measures in text categorization tasks. Information gain was reported to work well on text data.

As opposed to the BIF methods the global selection procedures reflect to a certain extent the dependencies between words. These methods include, for instance, forward and backward selection algorithms and oscillating search. Forward selection algorithms start with an empty set of features and add one feature at a time until the final feature set is reached. Backward elimination algorithms start instead with a feature set containing all features and remove features one at a time. Oscillating search [10] is not dependent on pre-specified direction of search. It is based on repeated modification of the current subset of features by alternating the so-called

down- and up-swings. However, these sequential methods can show to be too slow to yield results in reasonable time, because of their combinatorial nature.

In this paper, we propose a new feature selection method based on Testor Theory [11] for text categorization tasks. This criterion not only takes into account inter-feature relationships but also it is computationally feasible. We experimentally compared our method with information gain using *k*-nearest neighbour and Support Vector Machine classifiers over two benchmark text collections. We found that our method consistently outperformed information gain for both classifiers when aggressive feature selection is carried out.

2 Basic Concepts

Before presenting our feature selection method, we review the main definitions of the *Testor Theory* [11].

Let ζ be the set of training samples, each of them described in terms of features $\tau=\{t_1,\ldots,t_n\}$ and grouped into the classes $C_1,\ldots,C_r$, $r \geq 2$. Each feature t_i takes values in a set D_i, $i = 1,\ldots,n$. Let M be the training matrix, whose rows represent the training samples and columns represent the features describing them.

A comparison criterion of dissimilarity $\psi_i : D_i \times D_i \rightarrow \{0,1\}$ is associated to each t_i (i=1,…,n). Applying these comparison criteria for all possible pairs of objects belonging to different classes in M, a Boolean comparison matrix is built. Notice that the number of rows in the comparison matrix is $m' = \sum_{i=1}^{r-1} \sum_{j=i+1}^{r} |C_i||C_j|$, where $|C_i|$ denotes the number of objects in class C_i.

In the Testor Theory, the set of features $\pi = \{t_{i_1},\ldots,t_{i_k}\} \subseteq \tau$ and their corresponding set of columns $\{i_1,\ldots,i_k\}$ of a matrix M is called a *testor*, if after removing from M all columns except $\{i_1,\ldots,i_k\}$, all rows of M corresponding to distinct classes are different. In terms of the comparison matrix, a testor can be described as a set of features for which a whole row of zeros does not appear in the remainder comparison submatrix, after removing all the other columns. A testor is called *irreducible* (*typical*) if none of its proper subsets is a testor [11].

Thus, the set of all typical testors of a matrix M contains the combinations of features that distinguish the classes.

3 Proposed Feature Selection Method

Broadly speaking, our approach to feature selection is a combination of the individual and global approaches. More specifically, we use individual feature selection by applying word frequency criterion to select a first subset of features. Then, we apply Testor Theory to select the subset of these features that better discriminate the different target classes. Thus, this approach takes profit from both viewpoints: individual filtering speeds-up notably the selection of features and the global one takes into account possible feature correlations and discriminating power. Moreover, Testor Theory provides us a natural method to select for each class, independently of its weight and training set size, the set of features that better discriminates their

examples from the rest of classes. This approach alleviates the problem of unbalanced classes, which is a very common problem in TC, and implies that aggressive feature selection affect to all the classes not only the smaller ones. Next paragraphs describe in detail how Testor Theory is applied to filter features.

In our text categorization problem, ζ is the set of training documents, τ contains all terms occurring in the documents and $C_1,\ldots,C_r$ are the categories. Each document d_j is represented as a vector of term weights $d_j = (w_1^j,...,w_n^j)$. The selection of terms includes removing tags and stop words, lemmatization and proper name recognition. Weights are computed by using the standard *ltc* variant of *tf-idf* function [12], i.e., $w_i^j = (1+\log TF(t_i,d_j)) \cdot \log \frac{|\zeta|}{df(t_i)}$, where $TF(t_i,d_j)$ denotes the number of times t_i occurs in d_j and $df(t_i)$ is the number of documents in ζ in which t_i occurs at least once.

The representative of a category C_i, denoted as $\bar{c}_i$, is calculated as the average of the documents of that category, that is, $\bar{c}_i = \left(\frac{\sum_{d_j \in C_i} w_1^j}{|C_i|},...,\frac{\sum_{d_j \in C_i} w_n^j}{|C_i|} \right)$.

Given a category C_i, let $T(C_i) = \{t_1,\ldots, t_{n_{c_i}}\}$ be the most frequent terms in the representative $\bar{c}_i$, i.e., the terms t_j such that $w_j^{\bar{c}_i} \geq \varepsilon$, $j = 1,\ldots, n_{c_i}$ and ε is an user-defined parameter that represents the minimum frequency required to consider a term in the global subset selection process.

For each category C_i, we construct a matrix $MR(C_i)$, whose columns are the terms of $T(C_i)$, and its rows are the representatives of all categories, described in terms of these columns. Notice that this matrix is different in each category.

In order to calculate the typical testors, we considered two classes in the matrix $MR(C_i)$. The first class is only formed by $\bar{c}_i$ and the second one is formed by the other category representatives. Notice that our goal is to distinguish the category C_i from the other categories.

For the calculus of the typical testors, the key concept is the comparison criterion of the values of each feature. In our case, the features that describe the documents are the terms and its values are the weights of terms. The comparison criterion applied to all features is:

$$\psi(w_k^{\bar{c}_i}, w_k^{\bar{c}_j}) = \begin{cases} 1 & \text{if } w_k^{\bar{c}_i} - w_k^{\bar{c}_j} \geq \delta \\ 0 & \text{otherwise} \end{cases},$$

where $w_k^{\bar{c}_i}, w_k^{\bar{c}_j}$ are the weights in the category representative $\bar{c}_i$ and $\bar{c}_j$ in the column corresponding to the term t_k respectively, and δ is a dissimilarity threshold (in our experiments we use δ=0.15). As it can be noticed, this criterion considers the two values (weights of the term t_k) different if the term t_k is frequent in category C_i and not frequent in category C_j.

In order to determine all typical testors of each matrix $MR(C_i)$, we use the algorithm LEX [13], which computes efficiently the typical testors of a data

collection. Finally, the selected feature subset is the union of typical testors of all categories. Notice that, unlike global feature selection methods, the number of desired features is not fixed beforehand, but it depends on the ε parameter.

The proposed feature selection method is summarized as follows:

Algorithm Feature selection
Input: *M*: training matrix.
ε: term frequency threshold.
Output: τ′: set of selected features.
1. For each category *C*:
   a. Construct the matrix *MR*(*C*).
   b. Calculate the typical testors of the matrix *MR*(*C*), regarding two classes: *C* and its complement.
   c. Let *U*(*C*) be the union of all typical testors found in the step b.
2. $\tau' = \bigcup_{C} U(C)$

4 Experimental Results

As mentioned before, Information gain had been one of the best performing feature selection measures for text categorization. It takes into account the presence of the term in a category as well as its absence and can be defined by $IG(t_k, C_i) = \sum_{C \in \{C_i, \bar{C}_i\}} \sum_{t \in \{t_k, \bar{t}_k\}} P(t, C) \cdot \log_2 \frac{P(t, C)}{P(t) \cdot P(C)}$ [6]. In this formula, probabilities are interpreted on an event space of documents (e.g. $P(\bar{t}_k, C_i)$ indicates the probability that, for a random document *d*, term t_k does not occur in *d* and *d* belongs to category C_i), and are estimated by maximum likelihood. We use the maximum value over all categories as the global score.

In this section, we compare the proposed feature selection method with Information gain. With this aim, two high-performing classifiers for the experiments: *k*-Nearest neighbour (parallel implementation, [14]) and Support Vector Machines (LibSVM, [15]) are selected. We used the standard *C*-SVC form of the SVM classifier with *C*=1, the linear kernel and tolerance of termination criterion = 0.1. No data scaling has been done. We also used the binary approach, which extends the one-against-all multi-class method for multi-label classification.

4.1 Data Sets

In our experiments two benchmark text collections were chosen as the testbeds: Reuters-21578[1] and Reuters Corpus Version 1 (RCV1-v2) [16]. The distribution of training documents into the categories is quite unbalanced in both collections.

[1] The Reuters-21578 collection may be freely downloaded from http://kdd.ics.uci.edu.

Reuters-21578 consists of a set of 12902 news stories classified under 135 categories related to economics. In this paper, we used the subset of 90 categories with at least one positive training example and one test example. This collection is partitioned (according to the "ModApté" split we have adopted) into a training set of 7058 documents and a test set of 2740 documents. The dimension of the document space is 26602 terms.

RCV1-v2 collection consists of over 800000 newswire stories that have been manually classified into 103 categories. The topic codes were selected as class labels. This collection is partitioned (according to the LYRL2004 split we have adopted) into a training set of 23149 documents and a test set of 781265 documents. The dimension of the document space is 47152 terms.

As we used a parallel implementation of *k*-Nearest neighbour classifier [14], our experiments are carried out over the entire RCV1-v2 collection. However, SVM is unable to handle such a collection, and consequently we used a small percentage (2%) of it. The documents were randomly chosen, and split into a 70% training set and 30% test set, while maintaining the distribution of the class probabilities in the original training and test sets. The resulting set has 11224 training documents and 4815 test documents.

4.2 Results

The first experiments are conducted to compare the categorization performance of our feature selection method (TT) against Information Gain (IG) using *k*-NN and SVM classifiers on the two above-mentioned Reuters collections. In our experiments, we set $\varepsilon = \{0.1, 0.15, 0.2, 0.25, \ldots, 0.7\}$ to obtain feature subsets of different sizes. For instance, we obtained a subset of 502 features in Reuters-21578 and 639 features in RCV1-v2 collection when ε is fixed to 0.4. Figures 1, 2, 3 and 4 show the classifiers performance w.r.t. different feature subset selections (including all features). Effectiveness is evaluated with both micro-averaged and macro-averaged F1 measures. Whereas micro-F1 depends on the size of the evaluated categories, macro-F1 depends on the number of categories to be evaluated. Thus, a classifier that behaves well on large categories will obtain a high micro-F1 value, but if it does not with small ones it will obtain low macro-F1. This is because in text collections, large categories cover a very large portion of the collection and small categories are much more numerous than large ones. As a result, when applying feature selection it is more difficult to improve macro-F1 values than micro-F1 ones.

Several observations can be made by analyzing the results in Figures 1, 2, 3 and 4. First, our feature selection method consistently outperformed information gain for both classifiers and both data collections at all number of selected features. The increase of performance is much larger for macro-averaged F1 (27% for Reuters-21578 and 18% for RCV1-v2 in average) than micro-averaged F1 (0.37% for Reuters-21578 and 5% for RCV1-v2 in average). This fact seems to suggest that our feature selection method is more insensitive to the problem of unbalanced class distribution. Another interesting observation is that the lesser number of selected features, the higher increase of performance is obtained.

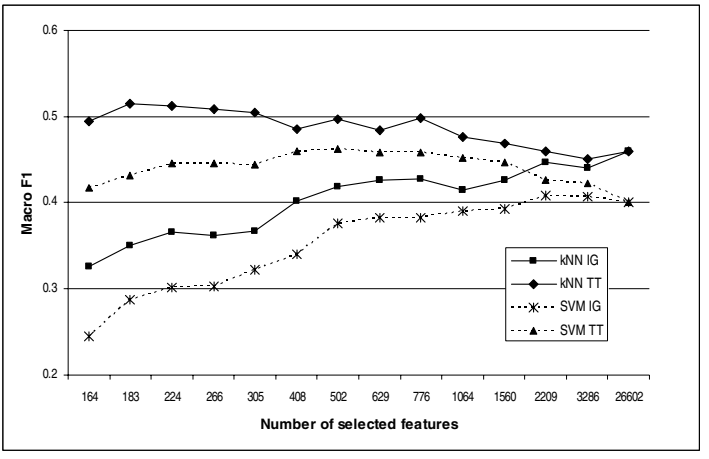

Fig. 1. Macro-averaged F1 scores for Reuters-21578 collection

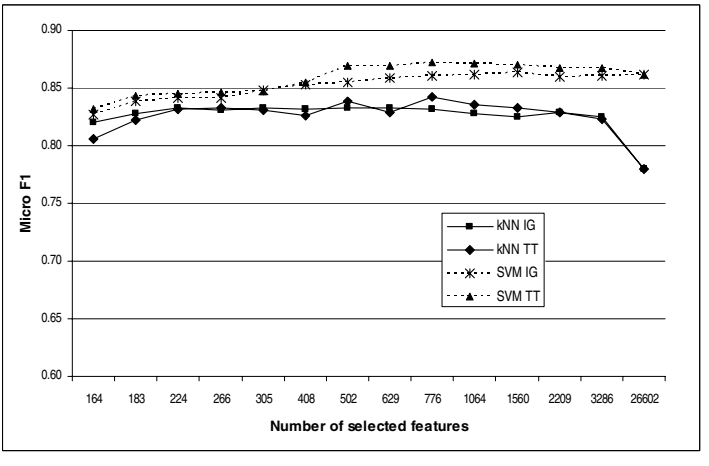

Fig. 2. Micro-averaged F1 scores for Reuters-21578 collection

A second fact that also emerges clearly from the figures is that our method achieves better F1 scores with very aggressive feature selection than those obtained when all features are regarded for both classifiers and both text collections.

Finally, in Figure 1 we observe that a good feature selection method enables *k*-NN classifier surpasses SVM's performance.

Our second experiment was focused on evaluating the time performance of our feature selection method (see Fig. 5). It can be seen that the behaviour is exponential as the number of selected features increases. From a practical point of view this is not a problem. Notice that execution times are negligible for aggressive feature selections,

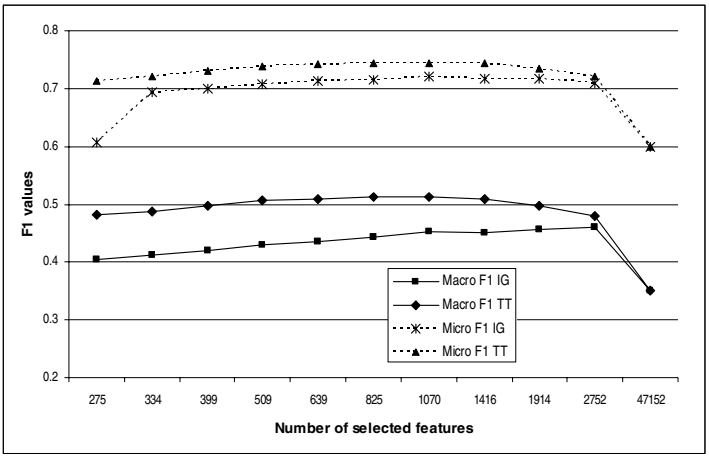

Fig. 3. F1 scores for *k*-NN classifier on the entire RCV1-v2 collection

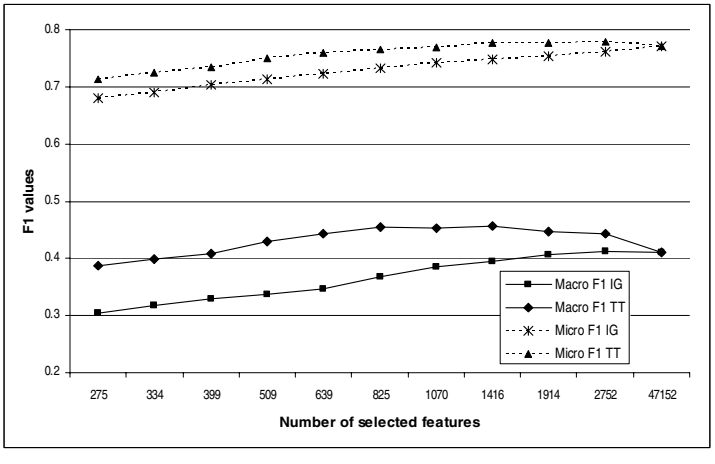

Fig. 4. F1 scores for SVM classifier on the small percentage (2%) of the RCV1-v2 collection

at the same time that good effectiveness improvements are achieved for them (see Figures 1-5). When the feature selection is not so aggressive (e.g. 2209 features), the combinatorial explosion arises but at the same time effectiveness improves very slightly. In this way, this indicates that global methods (as ours) are useful when applying aggressive feature selection, and that individual methods (e.g. *tf-idf*) are useful when selecting large feature subsets.

Comparing to other global methods, our execution times contrast with, for example, the 11 hours that Oscillating search takes to select around 2000 features in a subset of our test collection [17].

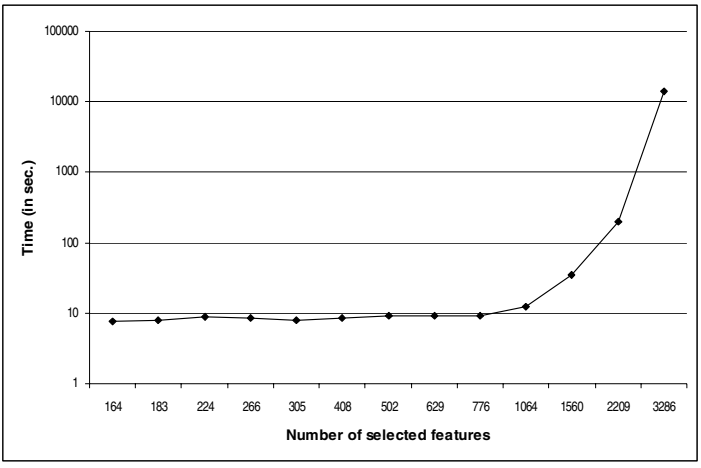

Fig. 5. Computational time of our feature selection method over Reuters-21578 collection

5 Conclusions

In this paper, a feature selection method that combines word frequency criterion and Testor Theory for Text Categorization tasks has been proposed. This method not only takes into account feature relationships and discriminating power but also it is computationally feasible. In this sense, it takes advantages from both individual and global methods for feature selection.

The experiments were conducted on two benchmark text collections (Reuters-21578 and RCV1-v2) using two high-performing classifiers (*k*-nearest neighbour and SVM). Results show that our method consistently outperformed information gain, especially when aggressive feature selection is carried out. The better performance improvements are obtained with respect to macro-averaged F1. This suggests that the proposed method is not affected by unbalanced class distribution.

The proposed method achieves good F1 scores with very aggressive feature selection, and even better than those obtained when all features are regarded. Thus, it may significantly ease the application of more powerful and computationally intensive machine learning methods to very large text categorization problems which are otherwise intractable.

As future work, we plan to study how this feature selection method can be applied in the context of adaptive document filtering tasks.

Acknowledgments. This work was partially supported by the Research Promotion Program 2006 of Universitat Jaume I, Spain and the CICYT project TIN2005-09098-C05-04.

References

1. Jain, A.K., Duin, R.P.W., Mao, J.: Statistical Pattern Recognition: A Review. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(1), 4–37 (2000)
2. John, G.H., Kohavi, R., Pfleger, K.: Irrelevant Features and the Subset Selection Problem. In: Proceedings of the 11th International Conference on Machine Learning, pp. 121–129 (1994)
3. Lewis, D.D.: An evaluation of phrasal and clustered representations on a text categorization task. In: Proceedings of 15th ACM International Conference on Research and Development in Information Retrieval, Denmark, pp. 37–50. ACM Press, New York (1992)
4. Yang, Y., Pedersen, J.O.: A Comparative Study on Feature Selection in Text Categorization. In: Proc. of the 14th International Conference on Machine Learning, pp. 412–420 (1997)
5. Mladenic, D.: Feature subset selection using in text learning. In: Proceedings of the 10th European Conference on Machine Learning, pp. 95–100 (1998)
6. Sebastiani, F.: Machine Learning in automated text categorization. ACM Computing Surveys 34(1), 1–47 (2002)
7. Mladenic, D., Grobelnik, M.: Feature selection for unbalanced class distribution and naive bayes. In: Proc. of the 16th International Conference on Machine Learning, pp. 258–267 (1999)
8. Rogati, M., Yang, Y.: High-performing feature selection for text classification. In: Proceedings of the 11th International Conference on Information and Knowledge Management, pp. 659–661. ACM Press, New York (2002)
9. Forman, G.: An extensive empirical study of feature selection metrics for text classification. Journal of Machine Learning Research 3, 1289–1305 (2003)
10. Somol, P., Pudil, P.: Oscillating Search Algorithms for Feature Selection. In: Proc. of the 15th IAPR International Conference on Pattern Recognition, Barcelona, pp. 406–409 (2000)
11. Lazo-Cortés, M., Ruiz-Shulcloper, J., Alba-Cabrera, E.: An overview of the evolution of the concept of testor. Pattern Recognition 34(4), 753–762 (2001)
12. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information Processing and Management 24, 513–523 (1988)
13. Santiesteban, Y., Pons-Porrata, A.: LEX: a new algorithm for the calculus of typical testors. Mathematics Sciences Journal 21(1), 85–95 (2003)
14. Gil-García, R., Badía Contelles, J.M., Pons-Porrata, A.: Parallel nearest neighbour algorithms for Text Categorization. In: Kermarrec, A.-M., Bougé, L., Priol, T. (eds.) Euro-Par 2007. LNCS, vol. 4641, pp. 328–337. Springer, Heidelberg (2007)
15. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm
16. Lewis, D., Yang, Y., Rose, T., Li, F.: Rcv1: A new benchmark collection for text categorization research. Machine Learning Research 5, 361–397 (2004)
17. Novovicová, J., Somol, P., Pudil, P.: Oscillating Feature Subset Search Algorithm for Text Categorization. In: Martínez-Trinidad, J.F., Carrasco Ochoa, J.A., Kittler, J. (eds.) CIARP 2006. LNCS, vol. 4225, pp. 578–587. Springer, Heidelberg (2006)