Classification with Reject Option in Text Categorisation Systems

Giorgio Fumera Ignazio Pillai Fabio Roli

Dept. of Electrical and Electronic Eng., Piazza d'Armi, 09123 Cagliari, Italy
{fumera,pillai,roli}@diee.unica.it

Abstract

The aim of this paper is to evaluate the potential usefulness of the reject option for text categorisation (TC) tasks. The reject option is a technique used in statistical pattern recognition for improving classification reliability. Our work is motivated by the fact that, although the reject option proved to be useful in several pattern recognition problems, it has not yet been considered for TC tasks. Since TC tasks differ from usual pattern recognition problems in the performance measures used and in the fact that documents can belong to more than one category, we developed a specific rejection technique for TC problems. The performance improvement achievable by using the reject option was experimentally evaluated on the Reuters dataset, which is a standard benchmark for TC systems.

1. Introduction

Text categorisation (TC) problems consists in assigning text documents, on the basis of their content, to one or more predefined thematic categories. This is a typical information retrieval (IR) task, and is currently an active research field, with applications like document indexing and filtering, and hierarchical categorisation of web pages. Approaches to TC have shifted since the '90s from knowledge engineering to the machine learning approach, gaining in flexibility and saving expert labor power. Using the machine learning approach, a classifier is constructed on the basis of a training set of labeled documents, by using inductive learning methods. Several classification techniques have been applied to TC tasks, both symbolic, rule-based methods from the machine learning field, and non-symbolic methods (like neural networks), from statistical pattern recognition (see [8] for a comprehensive overview).

So far, works in the TC literature focused on two main topics: feature selction/extraction, and comparison between different classification techniques [6,9,10,11,12]. In this paper we focus on a technique used in statistical pattern recognition for improving classification reliability, namely, classification with reject option. The reject option consists in withholding the classification of an input pattern, if it is likely to be misclassified. Rejected patterns are then handled in a different way, for instance, they are manually classified. The reject option is useful in pattern recognition tasks in which the cost of an error is higher than that of a rejection.

To the best of our knowledge, so far no work considered the reject option in TC tasks. Nevertheless, we believe that the reject option can be useful also in TC. Accordingly, the objective of this paper is to develop a method for introducing the reject option in TC tasks, and to evaluate its potential usefulness. To this aim, we consider the usual decomposition of N-category problems into N independent binary problems [8]: each binary problem consists in deciding if an input document does or does not belong to one of the N predefined categories. We introduce the reject option by allowing the classifier to withhold assigning or not an input document to the categories for which the decision is considered not sufficiently reliable. To this end, we define a reject rule based on applying two threshold values to the scores provided by the classifier for each category. We then evaluate the performance improvement achievable by using the reject option on the Reuters dataset, which is a standard benchmark for TC systems.

The paper is structured as follows. Sections 2 and 3 provide the theoretical background about TC and about the reject option in statistical pattern recognition. In sections 4 and 5 we describe our method for implementing the reject option in TC tasks. The experimental results are reported in section 6. Conclusions are drawn in section 7.

2. Text categorisation

Formally, TC can be defined as the task of assigning a Boolean value T or F to each pair $(d_j,c_i)\in D\times C$, indicating whether document d_j is to be labeled as belonging to category c_i, where D is a set of documents, and C is a set of N predefined categories. The goal is to find a labeling function (classifier) $\phi:D\times C\rightarrow\{T,F\}$ which approximates at best an unknown target funtion Φ, according to a given performance measure [8].

The most used performance measures, precision (π) and recall (ρ), are derived from IR. Precision π_i for the i-th category is defined as the probability that, if a random document d is classified as belonging to c_i, this decision is correct: $\pi_i=P(\Phi(d,c_i)=T|\phi(d,c_i)=T)$. Recall ρ_i for the i-th class is defined as the probability that, if d is to be classified as belonging to c_i, this decision is taken: $\rho_i=P(\phi(d,c_i)=T|\Phi(d,c_i)=T)$. For a given classifier ϕ, these probabilities are usually estimated from a test set of pre-labeled documents, as a function of the number of true positives (TP_i), false positives (FP_i), true negatives (TN_i) and false negatives (FN_i) documents, defined as shown in Table 1. The estimates are computed as:

$$\pi_i=\frac{TP_i}{TP_i+FP_i},\quad \rho_i=\frac{TP_i}{TP_i+FN_i}\ . \qquad (1)$$

Table 1. Contingency table for class *i*

		$\Phi(d,c_i)$	
		T	F
$\phi(d,c_i)$	T	TP_i	FP_i
	F	FN_i	TN_i

Global precision and recall are computed either as micro- or macro-averaged values (depending on application requirements). Micro-averaging consists in applying definition (1) to the global values of TP, FP, TN and FN, obtained by summing over all classes:

$$\pi^{\mu}=\frac{\sum_i TP_i}{\sum_i\left(TP_i+FP_i\right)},\quad \rho^{\mu}=\frac{\sum_i TP_i}{\sum_i\left(TP_i+FN_i\right)}, \qquad (2)$$

while in macro-averaging, the values of π_i and ρ_i given in Eq. (1) are averaged over the classes:

$$\pi^{M}=\frac{\sum_i \pi_i}{N},\quad \rho^{M}=\frac{\sum_i \rho_i}{N}\ . \qquad (3)$$

Precision and recall (both macro- and micro-averaged) take on values in the range [0,1], and are not independent measures: a higher precision correspond to a lower recall, and vice-versa. The optimal classifier would be the one for which π=1, ρ=1. In real tasks, for varying values of the classifier parameters, precision and recall describe a curve in the π-ρ plane like the one of Fig. 1. The working point on this curve is application-dependent.

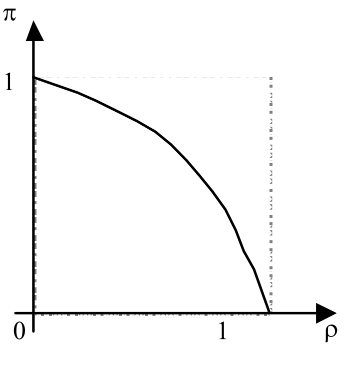

Fig. 1. A typical behaviour of the π-ρ curve

Several effectiveness measures which combine π and ρ have also been defined. A widely used one is the F_1 measure, defined as the harmonic mean of π and ρ:

$$F_1=\frac{2\pi\rho}{\left(\pi+\rho\right)}\ . \qquad (4)$$

In TC systems documents are usually represented with a weight vector, each weight corresponding to a *term* (either a word or a phrase) of the so-called vocabulary, which is the set of all terms occurring in the set D (note that in real tasks D is the available training set). Typically, only terms obtained after stop-word removal and stemming are considered. Weights are mainly computed as *term frequencies* (*tf*), in which they represent the frequency of occurrence of the corresponding terms in a document, and as *inverted term frequencies* (*tfidf*), defined as:

$$tfidf(t_k,d_j)=\#(t_k,d_j)\cdot\log(|D|/\#_D(t_k))\ ,$$

where $\#(t_k,d_j)$ denotes the number of occurrences of term t_k in document d_j, and $\#_D(t_k)$ denotes the number of documents in D in which t_k occurs.

3. Classification with reject option in statistical pattern recognition

In statistical pattern recognition, classification with reject option has been formalised under the minimum risk theory in [1,2,14]. When the costs of correct classifications, rejections, and misclassifications are respectively c_C, c_R and c_E (obviously, $c_C<c_R<c_E$), then the expected risk (i.e. the expected value of the cost of classifying any pattern) is minimised by Chow's rule, which is defined as follows. Consider the class for which the a posteriori probability of $\mathbf{x}$ is maximum: ω_i=argmax$_j P(\omega_j|\mathbf{x})$; if $P(\omega_i|\mathbf{x})\geq T$, assign $\mathbf{x}$ to ω_i, otherwise reject $\mathbf{x}$. T is the so-called reject threshold, whose optimal value depends on classification costs: $T=(c_E-c_R)/(c_E-c_C)$ [2]. Note that when the cost of rejections equals that of misclassifications, one obtains T=0: therefore, no pattern is rejected, and Chow's rule reduces to the well-known Bayes rule.

We point out that the above approach to the reject option can not be immediately applied to TC tasks, for two main reasons. First, Chow's rule applies to classification problems in which each pattern belongs to only one class, while this is not the case of TC problems, as explained in previous sections. Moreover, Chow's rule is aimed at minimising the expected risk of a classifier, while the performance measures in TC tasks are different than the expected risk. In the next section we discuss the way in which the reject option can be introduced in TC tasks.

4. A method for introducing the reject option in text categorisation

As pointed out in [8], N-category TC problems are usually considered as N independent binary problems: for each category c_i, the classifier has to decide whether an input document d does or does not belong to c_i, independently on the other categories. In this framework, it is reasonable to introduce the reject option by allowing a classifier to withhold making a decision, for categories where the decision is considered unreliable. A document can then be rejected from any subset of the N categories, while it can be accepted and automatically categorised as belonging or not to the other categories. Therefore, each document has to be exceptionally handled (typically, by manual classification) only for the categories from which it has been rejected, if any. A text classifier with reject option can then be viewed as implementing N functions $\phi_i{:}D\times C\to\{T,F,R\}$, one for each category, where R stands for the reject decision. The goal of the reject option should be that of increasing the effectiveness measure (Eq. (1)-(4)) by turning as many incorrect decisions (i.e. FP_i and FN_i) as possible into rejections. Obviously, this can be achieved at the expense of exceptionally handling the rejected documents: a trade-off between the achievable performance improvement and the cost of rejections must then be found. However, the cost of handling the rejected documents is clearly application-dependent. In this work, we evaluate the performance improvement that can be achieved by the reject option, with respect to the rate of rejected decisions. By denoting as r_i the number of documents rejected from the i-th category, the rate r of rejected decisions is defined as:

$$r=\left(1/N|D|\right)\sum\nolimits_{i=1}^{N}r_i\ , \tag{5}$$

where $|D|$ is the total number of documents. From now on, we will call r the *reject rate*. Accordingly, we will consider the following goal for the reject option: maximise the performance measure for any given value of the reject rate (5).

Let us now discuss the way in which the decision of rejecting an input document from a category can be implemented. Most classification techniques used so far in TC (like the naïve Bayes and neural networks) are based on computing for any document d a score $s_i\in[0,1]$ for each category c_i, representing the evidence of the fact $d\in c_i$. Scores can be obtained, for instance, from the output of neural network classifiers, or as probabilities computed using a naïve Bayes classifier. The decision of assigning or not assigning d to c_i is usually made by introducing a threshold τ_i for each category, so that d is labeled as belonging to category c_i, only if $s_i(d)\geq\tau_i$, i=1,...,N [8,12]. In other words, if $s_i(d)\geq\tau_i$, then $\phi(d,c_i)=T$, otherwise $\phi(d,c_i)=F$. Note that the threshold values are usually computed from a set of validation documents.

To introduce the reject option, it is reasonable to assume that the reliability of the above decision is higher for values of s_i "far" from τ_i, than for values "near" to τ_i. In other words, the lower $|s_i-\tau_i|$, the more likely is an erroneous decision for category i. Accordingly, we implement the reject option by introducing two threshold values for each category, denoted as τ_{Hi} and τ_{Li}, with $\tau_{Hi}\geq\tau_{Li}$, such that if s_i is between these values, the document is rejected from category i. Formally, the classifier ϕ is the defined as follows:

$$\begin{array}{ll} \text{if } s_i(d)\geq\tau_{Hi}, & \phi(d,c_i)=T,\\ \text{if } \tau_{Li}<s_i(d)<\tau_{Hi}, & \phi(d,c_i)=R,\\ \text{if } s_i(d)\leq\tau_{Li}, & \phi(d,c_i)=F. \end{array} \tag{6}$$

Note that, if $\tau_{Hi}=\tau_{Li}$, one obtains the standard classifier without reject option. We point out that our approach for introducing the reject option by using two threshold values for each category, is analogous to that proposed in [13] for two-class pattern recognition problems.

In order to experimentally evaluate the performance improvement achievable by the reject option, with respect to the number of rejected decisions, we developed algorithms for maximising the performance measure F_1 (4), for any given value of the reject rate (1). These algorithms are presented in section 5.

5. Algorithms for determinining threshold values

When the reject option is implemented in a TC problem as described in section 4, the optimal values of the $2N$ threshold $\{\tau_{H1},\tau_{L1},...,\tau_{HN},\tau_{LN}\}$ are the ones that maximise the performance measure for any given value of the reject rate (1). In practice, it is convenient to implement an algorithm that maximises the performance measure while keeping the reject rate r below the given value. We will denote such value as r_{MAX}. In this work, we focused on the F_1 performance measure, both macro- and micro-averaged. The macro-averaged F_1 is defined as

$$F_1^M=\left(1/N\right)\sum\nolimits_{i=1}^{N}F_{1_i}^M=\left(1/N\right)\sum\nolimits_{i=1}^{N}2\pi_i^M\rho_i^M/\left(\pi_i^M+\rho_i^M\right), \tag{7}$$

while the micro-averaged measure is defined as:

$$F_1^m=2\pi^m\rho^m/\left(\pi^m+\rho^m\right)=2/\left(2+\left(FP+FN\right)/TP\right), \tag{8}$$

where $FP=\sum\nolimits_{i=1}^{N}FP_i, FN=\sum\nolimits_{i=1}^{N}FN_i, TP=\sum\nolimits_{i=1}^{N}TP_i$.

To maximise $F_1{}^M$, we chose to independently maximise each $F_{1i}{}^M$, enforcing the constraint $r\leq r_{\mathrm{MAX}}$ by requiring a maximum reject rate r_i for each category equal to r/N: $r_i\leq r/N$. We also exploited the fact that $F_{1i}{}^M$ is a non-increasing function of any τ_{Li} (this property follows from Eqs. (7),(6),(3),(1)). Since for each class only two thresholds τ_{Hi} and τ_{Li} must be computed, a simple exhaustive search of their optimal values can be performed, using a predefined discretisation step $\Delta\tau$.

To maximise F_1^m, we exploited the following properties (we omit the proof for the sake of brevity):

- F_1^m is a non-increasing function of any τ_{Li} (from (8),(6),(2));
- consider the values of the thresholds of category i, $(\tau_{Li}^*, \tau_{Hi}^*)$, which maximise F_1^m, for any fixed value of the thresholds of the other classes; now, if the thresholds of the other classes are changed so as to increase F_1^m, then values of τ_{Li} and τ_{Hi} such that $\tau_{Hi} < \tau_{Hi}^*$ can not further increase F_1^m (from (8),(6),(2)).

On the basis of the above properties, we developed an algorithm for maximising F_1^m under the constraint $r \leq r_{\text{MAX}}$, based on an iterative search in the space of the $2N$ threshold values. The thresholds are initially set to zero. The above properties allow to reduce the search space by considering at each step only higher values of the thresholds with respect to the current ones. To further reduce the search space, we consider at each step only points in the search space obtained by incrementing the thresholds of only one category at a time, while keeping the ones of the other categories at their current values. Accordingly, at each step, the thresholds of only one category are changed, provided that F_1^m can be increased with $r \leq r_{MAX}$. If no threshold values satisfying this condition are found, the algorithm terminates. The algorithm can be schematised as follows.

```
initialise $\tau_{Hi}=0$, $\tau_{Hi}=0$, $i=1,...,N$
repeat
    for each category $c_i$:
        evaluate $F_1^m$ and $r_i$ for $\tau_{Hi} + k\Delta\tau$, for each $k$ in
        $0,\ldots, maxk$, and for the maximum $\tau_{Li}$ such
        that $r_i \leq r_{MAX}/N$, and $F_1^m$ does not decrease with
        respect to $\tau_{Hi} + k\Delta\tau$ and to the initial value of $\tau_{Li}$;
    select the pair of threshold values $(\tau_{Li}^*, \tau_{Hi}^*)$ that give
    the maximum improvement with respect to the current
    $F_1^m$, if any, and set them as new values for the
    corresponding category;
until a higher $F_1^m$ with $r \leq r_{MAX}$ is found with respect to the
previous step.
```

The parameter $\Delta\tau$ is a predefined discretisation step, while *maxk* determines the number of different values of τ_{Hi} which are explored at each step.

6. Experimental results

The goal of the experiments presented in this section was to evaluate the potential usefulness of the reject option in real TC tasks. More precisely, our aim was to assess the improvement of the performance of a TC system achievable by the reject option, implemented as described in section 4, with respect to the rate of rejected decisions. We conducted our experiments on the Reuters-21578 dataset, which is a standard benchmark for TC systems [8,4,11,3]. This dataset consists of a set of 21,578 newswire stories (provided in standard text files) classified under categories related to economics. In particular, we used the "ApteMod" version of this dataset, which was obtained by eliminating unlabelled documents and selecting the categories which have at least one document in the training set and one in the test set. This version consists of a training set of 7,769 documents and a test set of 3,019 documents, belonging to 90 categories. The vocabulary, extracted from the training set, consists of 16,635 terms, after stemming and stop-word removal.

Our experiments have been performed with three kinds of classifiers, commonly used in the TC literature [4,9,10,11]: *k*-nearest neighbour (*k*-NN), multi-layer perceptron neural network (MLP) and support vector machine (SVM) classifiers.

Pre-processing of the dataset (stop-word removal, stemming and feature extraction from the training set) was performed with the software SMART [7]. In particular, both *tf* and *tfidf* weights have been considered. Due to the high number of features, feature selection was performed on the training set by using the *information gain* criterion [10,11]. Feature selection was performed separately for each classifier, and for macro- and micro-averaged performance measures. Then, the 70% of the documents in the original training set was randomly extracted (by keeping the same proportion between the number of documents in each class) to be used for determining classifier parameters and for training the classifiers, while the remaining 30% of documents were used as validation set for estimating threshold values. In particular, for categories with less than five documents, we always kept at least one document both in the training and in the validation set; for categories with only one document in the original training set, the document was duplicated in the validation set. For statistical significance, the experiments were repeated for ten different randomly generated training sets. Reported results refer to the test set performance, averaged over these ten runs.

For MLPs, we used one hidden layer with 64 units, and 90 output units, as in [11]. The *tf* features were used. The number of input units was equal to that of features, which was set to 1,000. MLPs were traind with the standard backpropagation algorithm, with a learning rate equal to 0.01. For *k*-NNs, the *distance-weighted* version of Yang [10] was used, with 2,500 *tfidf* features. The value of *k* was 45 for macro- and 65 for micro-averaged measures. On the basis of experiments presented in [4,11], we used SVMs with linear kernel, trained with the SVM$^{\text{light}}$ software [5]. 10,000 *tfidf* features were used. One SVM for each category was trained, and the outputs were scaled to the range [0,1] using a sigmoidal function.

Thresholds estimation was performed on the validation set, with the algorithms described in section 5. We used a threshold increment unit $\Delta\tau$ equal to 0.001, and a value of *maxk* for F_1^m equal to 100. Values of the maximum allowed reject rate r_{MAX} between 0 and 15% were considered.

In Figs. 2-7 we report the test set macro- and micro-averaged values of F_1 versus the reject rate, averaged over the ten runs, for the three classifiers used.

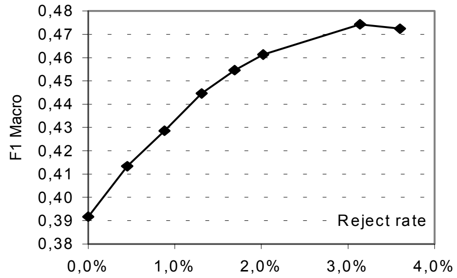

Fig. 2. Test set F_1^M vs reject rate (MLP)

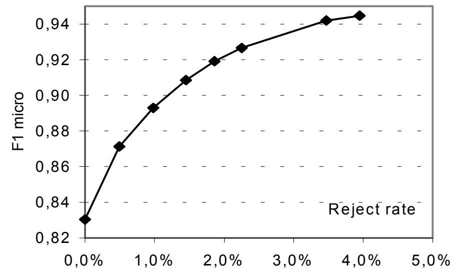

Fig. 3. Test set F_1^m vs reject rate (MLP)

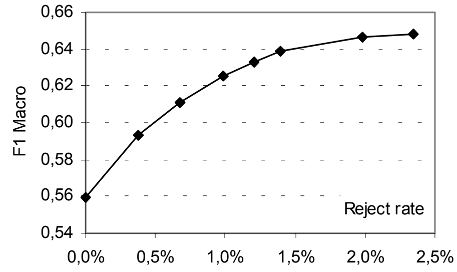

Fig. 4. Test set F_1^M vs reject rate (*k*-NN)

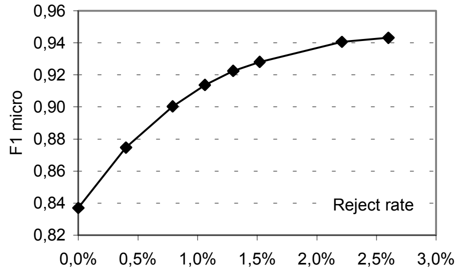

Fig. 5. Test set F_1^m vs reject rate (*k*-NN)

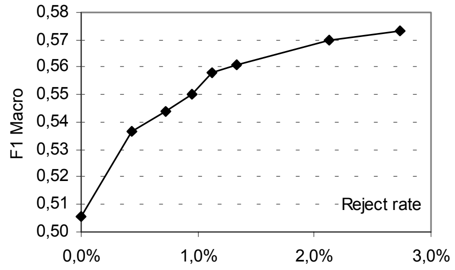

Fig. 6. Test set F_1^M vs reject rate (SVM)

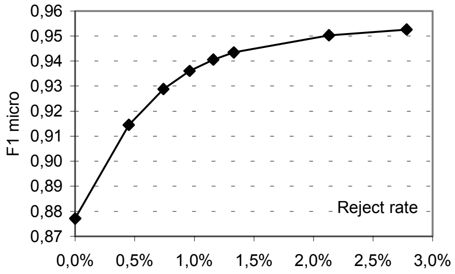

Fig. 7. Test set F_1^m vs reject rate (SVM)

Note that macro-averaging gives much worse performances than micro-averaging. This is due to the fact that Reuters categories have a very different generality, and macro-averaging is dominated by the performance of the system on rare categories, as noted in [12] and [8].

Let us analyse the performance improvement achieved by using the reject option. We first point out that the reject rate achieved on the test set is always much lower than the maximum required reject rate, which was 15%. In particular, the maximum value of the reject rate (i.e., the rate of rejected decisions) on the test set is about 4% for MLPs (Figs. 2,3), 3% for *k*-NN (Figs. 4,5), and 3.5% for SVMs (Figs. 6,7). Note now that F_1 always increases for increasing values of the reject rate (except for the final part of the curve in Fig. 2). Moreover, a significant increment of F_1 was always obtained by using the reject option, especially for macro-averaged F_1. In particular, the maximum increment of macro-averaged F_1 was about 20% with MLPs, 16% with *k*-NN, and 14% with SVM classifiers; the increment of micro-averaged F_1 was 13% with MLPs, 12% with *k*-NN, and 8% with SVM classifiers.

7. Conclusions

In this work we proposed a method for introducing the reject option in text categorisation problems, and experimentally evaluated the performance improvement achievable by using the reject option on a real TC task. The experimental results showed that the reject option can allow to significantly improve the performance of a text categorisation system, at the expense of a reasonably small rate of rejected decisions for each category.

As pointed out in section 4, the cost of handling rejected documents strongly depends on the particular application. The approach proposed in this work was based on the usual decomposition of *N*-category TC problems into *N* binary independent problems. On the basis of our experimental results, this approach proved to be useful for applications in which the cost of rejections depends mainly on the rate of rejected decisions. Our future work will be devoted to analyse the case in which the cost of rejections is significantly affected also by the number of rejected documents. In this case our approach should be modified, in order to guarantee a small number of rejected documents.

References

[1] C.K. Chow, "An Optimum character recognition system using decision functions", *IRE Trans. Electronic Computers* 6, 1957, pp. 247-254.

[2] C.K. Chow, "On Optimum Error and Reject Tradeoff", *IEEE Trans. on Inf. Theory* 16, 1970, pp. 41-46.

[3] V. Dasigi et al., "Information fusion for text classification – an experimental comparison", *Pattern Recognition* 34, 2001, pp. 2413-2425.

[4] T. Joachims, "Text categorization with support vector machines: learning with many relevant feature", Proc. 10th European Conf. on Machine Learning, Chemnitz, Germany, 1998, pp. 137–142.

[5] T. Joachims, "Making large-scale SVM learning practical", in *Advances in Kernel Methods – Support Vector Learning*, MIT Press, 1999, pp. 169-184.

[6] D.D. Lewis, "An evaluation of phrasal and clustered representations on a text categorization task", Proc. 15th ACM Int. Conf. on Research and Development in Inf. Retrieval, Kobenhavn, DK, 1992, pp. 37–50.

[7] G. Salton, *Automatic Text Processing: The Transformation, analysis, and Retrieval of Information by Computer*, Addison-Wesley, Pennsylvania, 1989.

[8] F. Sebastiani, "Machine Learning in Automated Text Categorization", *ACM Computing Surveys* 34, 2002, pp. 1–47.

[9] E. Wiener, J.O. Pedersen, and A.S. Weigend, "A neural network approache to topic spotting", Proc. 4th Annual Symp. on Doc. Analysis and Inf. Retrieval, Las Vegas, USA, 1995, pp. 317–332.

[10] Y. Yang, and J.O. Pedersen, "A comparative study on feature selection in text categorization", Proc. of 14th Int. Conf. on Machine Learning, Nashville, USA, 1997, pp. 412–420.

[11] Y. Yang, and X. Liu, "A re-examination of text categorization methods", Proc. of 22nd ACM Int. Conf. on Research and Development in Inf. Retrieval, Berkeley, US, 1999.

[12] Y. Yang, "A Study on Thresholding Strategies for Text Categorization", Proc. of SIGIR-01, Louisiana, US, 2001.

[13] F. Tortorella, "An optimal reject rule for binary classifiers", Proc. Int. Workshop on Statistical Pattern Recognition, Alicante, Spain, 2000, pp. 611-620.

[14] G. Fumera, F. Roli and G. Giacinto, "Reject option with multiple thresholds", *Pattern Recognition* 33, 2000, pp. 165-167.