A Two-Stage Classifier with Reject Option for Text Categorisation

Giorgio Fumera, Ignazio Pillai, and Fabio Roli

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

Abstract. In this paper, we investigate the usefulness of the reject option in text categorisation systems. The reject option is introduced by allowing a text classifier to withhold the decision of assigning or not a document to any subset of categories, for which the decision is considered not sufficiently reliable. To automatically handle rejections, a two-stage classifier architecture is used, in which documents rejected at the first stage are automatically classified at the second stage, so that no rejections eventually remain. The performance improvement achievable by using the reject option is assessed on a real text categorisation task, using the well known Reuters data set.

1 Introduction

Text categorisation (TC) systems are key components of many applications of document managing, like document retrieval, routing, and filtering. With the increased availability of documents in digital form over the last decade, and the consequent need of automatic document management systems, TC has become an active research topic in the machine learning field. TC can be viewed as a classification task, in which a document in natural language must be labeled as belonging or not to thematic categories from a predefined set, on the basis of its content [10]. Accordingly, in recent years, several researches investigated the use of statistical pattern recognition techniques applied to TC (a comprehensive review is given in [10]). In particular, different kinds of classification techniques have been evaluated and compared, like neural networks, k-nearest neighbors, support vector machines, naïve Bayes, and multiple classifier systems [5, 12, 6, 9], as well as feature extraction and selection techniques [11].

In this work, we focus on the reject option, which is a technique used to improve classification reliability in pattern recognition systems. The reject option has been formalised in the context of statistical pattern recognition, under the minimum risk theory, in [1, 2]. It consists in withholding the automatic classification of a pattern, if the decision is considered not sufficiently reliable. Rejected patterns must then be handled by a different classifier, or by a human operator. This requires to find a trade-off between the achievable reduction of the cost due to classification errors, and the cost of handling rejections (costs are obviously application-dependent). Although the reject option turns out to be very

useful in many pattern recognition systems, its use in TC systems has not been considered in the literature so far.

In a previous work, summarised in Sect. 3, we investigated how the reject option can be implemented in a TC system, and whether it can improve its reliability [3]. We implemented the reject option by allowing a text classifier to reject any subset of category assignments, for a given document. Using three different kind of classifiers (neural networks, k-nearest neighbors and support vector machines), we experimentally observed remarkable performance improvements, at the expense of small rates of rejected assignments. However, the rejected category assignments turned out to be spread across a large fraction of documents, making it impractical to handle them manually.

In this paper, we investigate whether the documents with rejected category assignments can be automatically handled. To this aim, we implement a two-stage classifier, based on the multi-stage architecture defined in [8] for pattern recognition systems. In our classifier, described in Sect. 4, documents can be either classified or rejected at the first stage. All the documents rejected at the first stage are then classified at the second stage, so that no rejections eventually remain. The effectiveness of this approach is evaluated by preliminary experiments carried out on the well known Reuters data set. The experimental results are presented in Sect. 5.

2 Text Categorisation

In TC systems, a document is typically represented as a vector of weights $d = (w_1, \ldots, w_T)$, where each w_k is associated to one of the T words that occur in training documents (*bag of words* approach). Weights can be computed in several ways, and are usually related to the frequency of the corresponding words, both in the document and in the whole training set [10]. While traditional classification problems are single-label, TC is a multi-label problem, i.e. each document can belong to any subset of C predefined categories $c_1, \ldots, c_C$. Given an input document d, a text classifier usually provides a score s_i for each category c_i, denoting the likelihood that d belongs to c_i. Several strategies can then be used to decide which categories d should be assigned to, given the scores. One of the most used strategies consists in determining a threshold τ_i for each c_i, after the training phase of the classifier, using a separate validation set. In the classification phase, each score is compared with the corresponding threshold: if $s_i \geq \tau_i$ ($s_i < \tau_i$), then d is labeled as (not) belonging to c_i [13, 10]. For instance, if neural networks are used as base classifiers, C output units can be used, each one related to one category, and their output values are taken as the scores s_i. Instead, the k-nearest neighbors (k-NN) classifier is implemented by first retrieving the k training documents most similar to an input document d. The similarity between two documents d and d' is computed using the cosine measure $\frac{d^{\mathrm{T}} \cdot d'}{\|d\| \cdot \|d'\|}$. Then, the score s_i for each category c_i is computed as the sum of the similarity measures between d and the training documents belonging to c_i, among the k nearest neighbors of d [10].

The performance measures used in TC are based on *precision* and *recall*, derived from the field of information retrieval. Precision π_i, for the i-th category, is defined as the fraction of documents that belong to c_i, among the ones that are assigned to c_i by the classifier. Recall ρ_i is defined as the fraction of documents that are correctly assigned by the classifier to c_i, among the ones that belong to c_i. Denoting with TP_i (True Positive) and FP_i (False Positive) the number of documents, out of a given set, correctly and erroneously labeled as belonging to c_i by the classifier, and with FN_i (False Negative) the number of documents erroneously labeled as not belonging to c_i, we have:

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

A global performance measure over all categories can be obtained either by micro- or macro-averaging the above category-related values, depending on application requirements. Micro- and macro-averaged values are defined respectively as follows:

$$\pi^{\mu} = \frac{\sum_{i=1}^{C} TP_i}{\sum_{i=1}^{C} (TP_i + FP_i)}, \quad \rho^{\mu} = \frac{\sum_{i=1}^{C} TP_i}{\sum_{i=1}^{C} (TP_i + FN_i)} \ , \tag{2}$$

$$\pi^{\mathrm{M}} = \frac{1}{C} \sum_{i=1}^{C} \pi_i, \quad \rho^{\mathrm{M}} = \frac{1}{C} \sum_{i=1}^{C} \rho_i \ . \tag{3}$$

The values of precision and recall lie in the range $[0, 1]$, and are not independent on each other: by using different classifier parameters (for instance, different thresholds τ_i) higher values of precision can be achieved at the expense of a lower recall, and vice-versa. Often, the combined measure F_1 is used, defined as the harmonic mean of precision and recall: $F_1^{\mu} = \frac{2\pi^{\mu}\rho^{\mu}}{\pi^{\mu}+\rho^{\mu}}$, $F_1^{\mathrm{M}} = \frac{1}{C}\sum_{i=1}^{C} \frac{2\pi_i^{\mathrm{M}}\rho_i^{\mathrm{M}}}{\pi_i^{\mathrm{M}}+\rho_i^{\mathrm{M}}}$. Also the F_1 measure takes on values in the range $[0, 1]$.

3 Text Categorisation with Reject Option

For single-label classification problems, the meaning of rejecting a pattern is usually to withhold automatically deciding the class to which it should be assigned [1, 2, 8]. The human operator (or the classifier) that handles rejections has thus to decide among the whole set of classes. A slightly different approach was used in [4]: a pattern can be automatically labeled as *not* belonging to any subset of classes, while it is rejected only by the remaining classes. Accordingly, the final decision should be taken among the latter classes only. Anyway, a rejected pattern must be assigned to only one class. To the best of our knowledge, the concept of rejecton has not been extended so far to multi-label problems.

Taking into account that multi-label TC problems involving C categories are usually viewed as C independent two-class problems (each one consisting in deciding whether a document should be assigned or not to the corresponding

category), in [3] we proposed to implement the reject option as follows: given a document d, a text classifier can automatically label d as belonging or not to any subset of the C categories, while it rejects d from the remaining categories, i.e. no decision is taken about these latter categories. Using the decision strategy without reject option, based on category-related thresholds τ_i (see Sect. 2), we experimentally found that the distribution of the scores s_i corresponding to incorrect category assignments, was peaked around the threshold τ_i. In other words, for any given category c_i, lower values of $|s_i - \tau_i|$ correspond to less reliable decisions. Accordingly, to implement the reject option as described above, we used two threshold values for each category c_i, τ_{L_i} and τ_{H_i} (with $\tau_{\mathrm{L}_i} \leq \tau_{\mathrm{H}_i}$), and the following decision strategy:

$$\begin{array}{ll} \text{if } s_i \leq \tau_{\mathrm{L}_i}, & \text{then } d \text{ is labeled as not belonging to } c_i; \\ \text{if } \tau_{\mathrm{L}_i} < s_i < \tau_{\mathrm{H}_i}, & \text{then } d \text{ is rejected from category } c_i; \\ \text{if } s_i \geq \tau_{\mathrm{H}_i}, & \text{then } d \text{ is labeled as belonging to } c_i. \end{array} \tag{4}$$

As the thresholds τ_i for the case without reject option, also the thresholds $\tau_{\mathrm{L}_i}, \tau_{\mathrm{H}_i}, i = 1, \ldots, C$, should be computed after the training phase of the classifier, on a separate validation set, by maximising the chosen performance measure. However, defining a performance measure for a TC problem with reject option is not straightforward. The measure typically used in statistical pattern recognition is the expected value (named expected risk), of the classification cost of a pattern, for a given decision rule. Different costs are defined for correctly classified patterns and for misclassified ones. When the reject option is used, it is straightforward to take into account the costs of rejections in the definition of the expected risk. In this case, it turns out that minimising the expected risk is equivalent to finding the best trade-off between the misclassification and rejection rates, depending on the corresponding costs [1, 2]. Instead, for TC problems, the precision and recall measures are not based on classification costs. It is thus not straightforward to generalise them to take into account the costs of rejections. Moreover, even defining the cost of rejections is not easy, since they are strongly application-dependent: in general, the cost for manually handling a document with some rejected category assignments could depend both on the time required by a person to read that document, and on the number of rejected assignments.

Nevertheless, when the reject option in a TC system is implemented as described above, it still makes sense to evaluate the performance through the same measures used without reject option, based on precision and recall, provided that the rejected category assignments are not taken into account when computing TP_i, FP_i and FN_i in (1) and (2). Such measure should be considered together with a measure related to the cost of documents with rejected assignments. Given the above difficulties in defining the cost of rejections, in [3] we chose to consider simply the rate of rejected decisions, i.e. the percentage of re-

jected category assignments over all test documents,[1] which will be denoted in the following as the "reject rate".

In [3], we experimentally evaluated the effectiveness of such implementation of the reject option, on the well known Reuters data set. We used neural networks, k-nearest neighbors and support vector machines as base classifiers. We considered all the main performance measures, i.e. the precision-recall curve and the F_1 measure, both micro- and macro-averaged. The C pairs of threshold values $\tau_{\mathrm{L}_i}, \tau_{\mathrm{H}_i}, i = 1, \ldots, C$, were computed by maximising the considered performance measure, while keeping the reject rate below a predefined value. Values of the reject rate in the range $[0, 0.15]$ were considered. For all the classifiers and all the performance measures considered, we found remarkable performance improvements, at the expense of small values of the reject rate. For instance, using the neural network classifier, the values of F_1^{M} increased from 0.39 to 0.47, as the reject rate increased from 0 to 0.03 (i.e. up to 3% of the total category assignments were rejected for test documents), while F_1^{μ} increased from 0.83 to 0.94, as the reject rate increased from 0 to 0.04. However, the analysis of these results showed that such performance improvements were always achieved by rejecting a small number of category assignments from a large fraction of documents. In particular, for most documents (50% to 75% of all test documents, depending on the reject rate) only one category assignment was rejected. Moreover, the number of documents with n rejected assignments decreased for increasing n. This could be a serious problem from the practical viewpoint, if documents with rejected assignments are manually handled, and if the cost of handling them depends mainly on the time required by a person to read a document, rather than on the number of rejected assignments. In this case, handling a single document rejected from ten categories could be much faster than handling ten documents, each rejected from one category, even if the number of rejected assignments is the same. It is worth noting that TC tasks involve usually several dozen categories (in the version of the Reuters data set we used, $C = 90$).

In conclusion, although we found that the reject option can lead to remarkable performance improvements at the expense of a small percentage of rejected category assignments, it turned out that such rejections are spread across a large fraction of documents: handling them manually is thus likely to be impractical for real applications. A possible solution to this problem is to automatically handle the documents with rejected category assignments, through the use of a second-stage classifier. This approach is investigated in the rest of this paper.

4 A Two-Stage Classifier with Reject Option

For pattern recognition applications in which a rejection is not acceptable as a final result, a multi-stage classifier architecture was proposed in [8] to automatically treating the rejects. At all stages, but the last one, a pattern can be either classified or rejected. Rejected patterns are fed into the next stage. At the final

[1] Given D documents and C categories, the total number of category assignments is $D \cdot C$.

stage, a decision is taken in any case, so that no rejections eventually remain. The rationale is that each stage should use more informative, and thus more costly measurements than the previous stage. An interesting implementation of such approach has been recently proposed in [7]: a "global" and fast neural network classifier is used at the first stage, followed by a "local" and slower nearest neighbor classifier at the second stage. To speed up the response time of the second stage, it is trained on a subset of the training patterns of the first stage. More precisely, only patterns rejected at the first stage (possibly using a narrower rejection criterion) are used. Moreover, when a test pattern is rejected at the first stage, the second stage decides only among the top-h ranking classes returned by the first stage, using only the training patterns belonging to such classes. In this work, we chose to investigate the two-stage approach proposed in [7], after modifying it to fit the characteristics of a multi-label problem. This approach was implemented as follows.

Let $\mathbb{C}$ denotes the set of all categories $\{c_1, \ldots, c_C\}$, and T' the training set of the first stage classifier. The decision thresholds, that in the following will be denoted as $\tau'_{\mathrm{L}_i}, \tau'_{\mathrm{H}_i}, i = 1, \ldots, C$, are computed using a separate validation set V', by maximising the chosen performance measure, while keeping the reject rate below a predefined value r', and also enforcing that the fraction of documents rejected from each category does not exceed r'. The algorithms we used are described in [3]. In the classification phase, each document is classified according to decision rule (4), using the scores s'_i provided by the first stage classifier, and the thresholds $\tau'_{\mathrm{L}_i}, \tau'_{\mathrm{H}_i}$. For rejected categories, the final decision is taken by the second stage, that we implemented as a modified k-NN classifier.

The training set of the second stage classifier is $\mathrm{T}'' = \bigcup_{i=1}^{C} \mathrm{T}''_i$, where T''_i is the subset of documents of T', for which the decision for category c_i was rejected at the first stage, using a narrower rejection criterion, i.e. using a new set of thresholds $\tau''_{\mathrm{L}_i}, \tau''_{\mathrm{H}_i}$, such that $\tau''_{\mathrm{L}_i} \leq \tau'_{\mathrm{L}_i}$, and $\tau''_{\mathrm{H}_i} \geq \tau'_{\mathrm{H}_i}, i = 1, \ldots, C$. Such thresholds are computed starting from the values of $\tau'_{\mathrm{L}_i}, \tau'_{\mathrm{H}_i}$, and imposing that exactly a fraction $r'' \geq r'$ of documents of T' are rejected from each category, at the first stage. In other words, we require that $|\mathrm{T}''_i| = r'' \cdot |\mathrm{T}'|$. Note that, in general, $\mathrm{T}''_i \cap \mathrm{T}''_j \neq \emptyset$, since a document can be rejected from more than one category. The validation set V'' of the second stage is obtained from V' analogously.

Now, let $\mathbb{C}_{\mathrm{R}}(d) \subseteq \mathbb{C}$ denotes the set of categories for which the decision has been rejected at the first stage, for a given test document d. For each category $c_i \in \mathbb{C}_{\mathrm{R}}(d)$, the final decision is taken at the second stage by first computing a score s''_i, and then comparing it with a single threshold τ''_i: if $s''_i \geq \tau''_i$ ($s''_i < \tau''_i$), then d is labeled as (not) belonging to c_i. The score s_i is computed as follows. Let $\mathrm{T}''_i(d)$ denotes the documents that belong to category c_i, among the k documents of T''_i nearest to d, according to the cosine similarity measure described in Sect. 2. The score s''_i for d is computed as the sum of the similarity measures between d and all the documents in $\mathrm{T}''_i(d)$. The thresholds $\tau''_i, i = 1, \ldots, C$, are computed on the validation set V'', by maximising the same performance measure used at the first stage. We point out that this is a modified version of the k-NN classifier

described in Sect. 2: indeed, the score for each category is computed using a different subset of the training set, instead of using the whole training set.

It is worth noting that this approach does not require to modify the performance measures based on precision and recall, to take into account the cost of rejections, since rejections are automatically handled. Instead, it could be necessary to find a trade-off between the achievable performance improvement, and the increased computational complexity of the two-stage architecture. This issue is discussed in the next section.

5 Experimental Results

In this section, we present the results of a first set of experiments aimed at evaluating whether the performance of a TC system can be improved using the two-stage classifier with reject option described in Sect. 4. The experiments have been carried out on the Reuters-21578 data set ("Mod-Apté" version), a standard benchmark for TC systems [10]. This data set consists of newswire stories classified under categories related to economics. After discarding unlabeled documents, and retaining only categories with at least one document both in the training set and in the test set, we obtained 7,769 training documents and 3,019 test documents belonging to $C = 90$ categories, with a vocabulary (extracted from the training set) of $T = 16{,}635$ words, after stemming and stopword removal. We represented each document using the bag of words approach. The weights were computed using the well known TF-IDF strategy [10]. The 75% of documents of the original training set, randomly extracted, were used as the training set for the first stage classifier, T′, while the remaining documents were used as validation set V′. Feature selection was performed on the training set, using the Information Gain criterion [10]. For these experiments, we used two different classifiers at the first stage: a multi-layer perceptron neural network (MLP), and a Naïve Bayes classifier (NB). For the MLP classifier, we used a number of input units equal to the number of document weights, and one output unit for each category. The MLP was trained with the standard back-propagation algorithm. The number of hidden neurons and of features (weights) was determined using the validation set, and was set respectively to 50 and 1,000. For the NB classifier, the number of features was set to 250. For the k-NN classifier at the second stage, we used 2,500 features, and a value of k equal to 10.

In Table 1 we report the first results obtained using the micro- and macro-averaged F_1 performance measures. The reported values of F_1 refer to the test set, and are average values over ten runs of the experiments, carried out using ten randomly generated training sets T′. We considered two values of the reject rate r', 0.05 and 0.10, and three different values of r'' (see Table 1), to evaluate the effect of different sizes of the training and validation sets at the second stage. The first row of Table 1, shows the F_1 values obtained by the first stage classifier without the reject option ($r' = r'' = 0$). The results obtained using the MLP and NB classifier at the first stage are reported respectively in columns "MLP+k-

Table 1. Average test set micro- and macro-averaged F_1 values, obtained by two-stage classifiers with reject option (columns "MLP+k-NN", "MLP+k-NN*" and "NB+k-NN"), and by a standard k-NN classifier without reject option (column "k-NN"). The reject rate on the test set (r') is reported in the first column

reject rates		k-NN		MLP+k-NN		MLP+k-NN*		NB+k-NN	
r'	r''	F_1^μ	F_1^M	F_1^μ	F_1^M	F_1^μ	F_1^M	F_1^μ	F_1^M
0	0	0.820	0.532	0.846	0.447	0.843	0.395	0.700	0.214
0.05	0.05			0.842	0.468	0.822	0.445	0.722	0.195
0.05	0.10			0.853	0.474	0.824	0.450	0.752	0.355
0.05	0.15			0.856	0.474	0.819	0.450	0.753	0.356
0.10	0.10			0.848	0.474	0.824	0.462	0.738	0.345
0.10	0.15			0.853	0.479	0.821	0.463	0.786	0.430
0.10	0.20			0.854	0.481	0.816	0.463	0.825	0.456

NN", and "NB+k-NN". For comparison, we also show the results achieved using a standard k-NN classifier at the second stage, trained using the same training set of the MLP at first stage ("MLP+k-NN*"), and a single standard k-NN classifier without the reject option ("k-NN"). For the two-stage classifier implemented as described in Sect. 4, the use of the reject option lead to an increase of the micro-averaged F_1 from 0.846 to 0.854 (using an MLP at the first stage), and from 0.700 to 0.825 (NB at the first stage). The macro-averaged F_1 increased from from 0.447 to 0.481 (MLP), and from 0.214 to 0.456 (NB). In particular, although the performance of the NB classifier without the reject option was quite poor when using the micro-averaged F_1, the reject option lead to a percentage improvement greater than 200%.

As one can expect, for the same value of the reject rate r', higher improvements are achieved using a larger training set at the second stage (i.e., higher values of r''). In particular, for small values of r'', the values of F_1 can decrease with respect to the case without reject option. This can be due to the small size of the training and validation sets of the second stage.

It is interesting to note that, in some cases, the two-stage classifier with the reject option outperformed the standard k-NN classifier without the reject option. This happened for the micro-averaged F_1, when a MLP was used at the first stage, and also for the NB classifier, but only for the highest values of r' and r''. Moreover, using a standard k-NN classifier at the second stage, with the same training set of the first stage ("MLP+k-NN*"), always lead to a worse performance than that achieved implementing the second stage as in Sect. 4 ("MLP+k-NN").

Finally, we point out that the performance obtained using the two-stage classifier, on the whole test set, was worse than that achieved by the first stage on only the *accepted* category assignments. For instance, for the MLP classifier at the first stage, such improvement was about 0.08 for both the micro- and macro-averaged F_1, for a reject rate $r' = 0.10$. This is not surprising, since, obviously, not all rejected category assignments are turned into correct assignments by the

second stage classifier, as they could, in principle, if rejections were manually handled.

The above preliminary results show that the use of the reject option can improve the reliability of a TC system, even when rejections are automatically handled by a second stage classifier. These results indicate that higher improvements can be achieved by using higher reject rates r' at the first stage, and larger sizes of the training set of the second-stage classifier. In particular, it should be noted that the increase in computational complexity, due to the use of a larger training set for the k-NN classifier at the second stage, can be limited by the fact that most documents are rejected by only one category, as we experimentally observed.

References

1. Chow, C.K.: An optimum Character Recognition System Using Decision Functions. IRE Trans. on Electronic Computers **6** (1957) 247–254
2. Chow, C.K.: On Optimum Error and Reject Tradeoff. IEEE Trans. on Information Theory **16** (1970) 41–46
3. Fumera, G., Pillai, I., Roli, F.: Classification with Reject Option in Text Categorisation Systems. In: Proc. 12th International Conference on Image Analysis and Processing. IEEE Computer Society (2003) 582–587
4. Ha, T.M.: The Optimum Class-Selective Rejection Rule. IEEE Trans. on Pattern Analysis and Machine Intelligence **19** (1997) 608–615
5. Joachims, T.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In: Proc. 10th European Conference on Machine Learning (1998) 137–142
6. Li, Y.H., Jain, A.K.: Classification of Text Documents. The Computer Journal **41** (1998) 537–546
7. Giusti, N., Masulli, F., Sperduti, A.: Theoretical and Experimental Analysis of a Two-Stage System for Classification. IEEE Trans. on Pattern Analysis and Machine Intelligence **24** (2002) 893–904
8. Pudil, P., Novovicova, J., Blaha, S., Kittler, J.: Multistage Pattern Recognition with Reject Option. In: Proc. 11th IAPR Int. Conf. on Pattern Recognition, Vol. 2. (1992) 92–95
9. Schapire, R.E., Singer, Y.: BoosTexter: a Boosting-Based System for Text Categorization. Machine Learning **39** (2000) 135–168
10. Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys **34** (2002) 1–47
11. Yang, Y., Pedersen, J.O.: A Comparative Study on Feature Selection in Text Categorization. In: Proc. 14th Int. Conf. on Machine Learning (1997) 412–420
12. Yang, Y., Liu, X.: A Re-Examination of Text Categorization Methods. In: Proc. 22nd ACM Int. Conf. on Res. and Dev. in Inf. Retrieval (1999) 42–49
13. Yang, Y.: A Study on Thresholding Strategies for Text Categorization. In: Proc. 24th ACM Int. Conf. on Res. and Dev. in Inf. Retrieval (2001) 137–145