Applied Soft Computing xxx (2011) xxx–xxx

Contents lists available at ScienceDirect

Applied Soft Computing

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

A multiclass/multilabel document categorization system: Combining multiple classifiers in a reduced dimension

A. Zelaia*, I. Alegria, O. Arregi, B. Sierra

University of the Basque Country, UPV-EHU, Computer Science Faculty, 649 postakutxa, 20.080 Donostia, Gipuzkoa, Euskal-Herria, Spain

ARTICLE INFO

Article history:
Received 16 December 2009
Received in revised form
20 December 2010
Accepted 12 June 2011
Available online xxx

Keywords:
Document categorization
Vector space models
Multiclassifiers
Distance based classifiers

ABSTRACT

This article presents a multiclassifier approach for multiclass/multilabel document categorization problems. For the categorization process, we use a reduced vector representation obtained by SVD for training and testing documents, and a set of *k*-NN classifiers to predict the category of test documents; each *k*-NN classifier uses a reduced database subsampled from the original training database. To perform multilabeling classifications, a new approach based on Bayesian weighted voting is also presented. The good results obtained in the experiments give an indication of the potential of the proposed approach.

1. Introduction

Document categorization, the assignment of natural language texts, according to their content, to one or more predefined categories is an important component in many information organization and management tasks. Researchers have concentrated their efforts on finding the appropriate way to represent documents, index them and construct classifiers to assign each document to the correct categories. Both, document representation and classification method are crucial steps in the categorization process, and they are the object of this paper.

With respect to document representation, in order to obtain the vector representation of documents latent semantic indexing (LSI) [6], a variant of the vector space model, is used. This technique compresses vectors representing documents into vectors of a lower-dimensional space. LSI, which is based on singular value decomposition (SVD) of matrices [1], has the ability to extract the relations among words and documents by means of their context of use, and has been successfully applied to Information Retrieval tasks.

Once the representation of the documents is determined, a multiclassifier [14] is used to perform the categorization process. We use different training databases obtained from the original one by random subsampling, and a category prediction is given for each of them. Finally, to make the category predictions of testing documents, we use a model inspired in bagging [2] which uses *k*-NN classifiers [4].

Document representation and categorization do not solve the problem of multilabeling; the fact that one document can effectively belong to more than one of the categories considered. The most widely used technique for multilabeling in the literature is based on a binary selection for each category, where each document is tested as belonging or not to each category. In this paper we propose a new approach to multilabeling based on Bayesian voting.

The experiment presented in this article has been evaluated for Reuters-21578 standard document collection.[1] Keeping in mind the results published in the most recent literature, and having obtained promising results in our experiments, we consider the new categorization method presented in this article an interesting contribution for text categorization tasks.

The remainder of this article is organized as follows: Section 2 discusses related work on document categorization for Reuters-21578 collection. Section 3 presents our approach to the multiclass/multilabel text categorization. In Section 4 the experimental setup is introduced, and details are provided about the Reuters database, the preprocessing applied and the parameters to tune. The parameter tuning process is explained in detail in Section 5, and the experimental results are presented and

* Corresponding author.
E-mail addresses: ccpzejaa@si.ehu.es, ana.zelaia@ehu.es (A. Zelaia), acpalloi@si.ehu.es (I. Alegria), acparuro@si.ehu.es (O. Arregi), ccpsiarb@si.ehu.es (B. Sierra).

[1] http://daviddlewis.com/resources/testcollections.

1568-4946/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2011.06.002

discussed in Section 6. Finally, Section 7 contains some conclusions and comments on future work.

2. Related work

Text categorization consists in assigning predefined categories to text documents. In the past two decades, document categorization has received much attention and a considerable number of machine learning based approaches have been proposed. A good tutorial on the state-of-the-art of document categorization techniques can be found in [26].

In the document categorization task, different types of problems can be found,

- single-label vs. multilabel document categorization problems. In single-label document categorization tasks exactly one category is assigned to each document. In the multilabel case, categories are not mutually exclusive because the same document may be relevant to more than one category (1 to m category labels may be assigned to the same document, being m the total number of predefined categories).
- Binary classification problems vs. multiclass classification problems. In binary classification only two categories are involved. Multiclass problems arise when a document can be categorized under more than 2 categories.

Most of the classification systems which handle multilabel data in a multiclass problem decompose the multiclass problem into multiple, independent binary classification problems [16]. In this article we present a classifier which handles multilabel data in a multiclass problem; first, it produces a ranking of possible labels for a given document, expecting that the appropriate labels will appear at the top of the ranking. Then, it selects the number of labels to assign to a document (one or two). See also [20] and [36].

In order to reduce the feature vector representation, many authors use the SVD technique in text categorization problems [32] and [21].

For experimentation purposes, there are standard document collections available in the public domain that can be used for document categorization. The most widely used is Reuters-21578 collection, which is a multiclass (135 categories) and multilabel (the mean number of categories assigned to a document is 1.2) dataset. Many experiments have been carried out for the Reuters collection. However, they have not been performed under the same experimental conditions. So, it is difficult to establish comparisons among them. In order to overcome this problem and to lead researchers to use the same training/testing divisions, the Reuters documents have been specifically tagged, and researchers are encouraged to use one of these divisions. In our experiment we used the "ModApte" split [19].

In this section, the category subsets, evaluation measures and results obtained in the past and in recent years for Reuters-21578, ModApte split are analyzed.

2.1. Category subsets

Concerning the evaluation of the classification system, the TOPICS group of categories that labels Reuters dataset contains 135 categories. However, since many of the categories do not appear in any of the documents, and given that inductive based learning classifiers learn from training examples, these categories are not usually considered at evaluation time. The most widely used subsets are the following:

- Top-10: It is the set of the 10 categories which have the highest number of documents in the training set.
- R(90): It is the set of 90 categories which have at least one document in the training set and one in the testing set.
- R(115): It is the set of 115 categories which have at least one document in the training set.

In order to analyze the relative hardness of the three category subsets, a very recent article has been published by Debole and Sebastiani [5] where a systematic comparative experimental study has been carried out.

The results of the classification system proposed in this article are evaluated according to these three category subsets; once all the test documents have been classified, the evaluation measure is calculated for Top-10, R(90) and R(115).

2.2. Evaluation measures

The evaluation of a text categorization system is usually done experimentally by measuring its effectiveness, i.e. average correctness of the categorization. In binary text categorization, two known statistics are widely used to measure this effectiveness: precision and recall. Precision (Prec_i) is the percentage of documents correctly classified into a given category c_i, and recall (Rec_i) is the percentage of documents belonging to a given category c_i that are indeed classified into it.

$$\mathrm{Prec}_i = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FP}_i} \qquad \mathrm{Rec}_i = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FN}_i}$$

where TP_i are true positives—documents correctly deemed to belong to c_i; FP_i are false positives—documents incorrectly deemed to belong to c_i; and FN_i are false negatives—documents incorrectly deemed not to belong to c_i.

In general, there is a trade-off between precision and recall. Thus, a classifier is usually evaluated by a measure which combines precision and recall. Various such measures have been proposed along the years. The breakeven point (BEP), the value at which precision equals recall, has been frequently used during the past decade. However, it has been recently criticized by its proposer ([26], footnote 19). Nowadays, the F_1 score is more frequently used. The F_1 score combines recall and precision with an equal weight. Given that Prec_i and Rec_i have been calculated for a given category c_i, the F_1 score for category i is calculated as follows:

$$F_1^i = \frac{2 \cdot \mathrm{Prec}_i \cdot \mathrm{Rec}_i}{\mathrm{Prec}_i + \mathrm{Rec}_i}$$

Since precision and recall are defined only for binary classification tasks, for multiclass problems results need to be averaged to get a single performance value. This is done by calculating the *microaverage* and *macroaverage* of results. In microaveraging, which is calculated by globally summing over all individual cases, categories count proportionally to the number of their positive testing examples. In macroaveraging, which is calculated by averaging over the results of the different categories, all categories count the same. Being $|C|$ the total number of categories in the multiclass problem, microaveraging (F_1^{μ}) and macroaveraging (F_1^{M}) are calculated as follows:

$$F_1^{\mu} = \frac{2\sum_{i=1}^{|C|}\mathrm{TP}_i}{2\sum_{i=1}^{|C|}\mathrm{TP}_i + \sum_{i=1}^{|C|}\mathrm{FP}_i + \sum_{i=1}^{|C|}\mathrm{FN}_i} \qquad F_1^{\mathrm{M}} = \frac{\sum_{i=1}^{|C|}F_1^i}{|C|}$$

Please cite this article in press as: A. Zelaia, et al., A multiclass/multilabel document categorization system: Combining multiple classifiers in a reduced dimension. Appl. Soft Comput. J. (2011), doi:10.1016/j.asoc.2011.06.002

Table 1
Some results reported for the Reuters-21578, ModApte split.

Type	Results reported by	Measure	R(90)	Top-10
SVM	Joachims [16]	BEP	86.4	–
SVM	Dumais et al. [9]	BEP	87.0	92.0
Commitee	Weiss et al. [28]	BEP	87.8	–
MFoM	Gao et al. [11]	F_1^μ	88.42	93.07
SVM	Kim et al. [17]	F_1^μ	87.11	92.21
SVM	Gliozzo and Strapparava [13]	F_1^μ	–	92.80
Combination	Debole and Sebastiani [5]	F_1^μ	78.7	85.20

See [5,30] for a more detailed explanation of the evaluation measures mentioned above. Results presented in this article are microaveraged (F_1^μ) and macroaveraged (F_1^M) F_1 scores.

2.3. Comparative results

Sebastiani [26] presents a table which lists results of experiments for various training/testing divisions of Reuters. Although the results listed by Sebastiani are microaveraged breakeven point (BEP) measures, and consequently, are not directly comparable to the ones presented in this article, we want to point out some of them.

In Table 1 some of the best results reported for the Reuters-21578, ModApte split are summarized. In the first part of the table, the three best results reported in [26] have been extracted. Two of them have been obtained by using support vector machines and the third one by using a commitee of multiple decision trees. As we have said earlier, they are microaveraged BEP measures. In the second part of the table, more recent microaveraged F_1 scores are included. MFoM learning approach has been used in [11,12], SVMs in [17] and domain kernel inside a SVM in [13]. Results reported by [5] give the average effectiveness of any combination of a learning method, a term selection function, a reduction factor and a term weighting policy.

Results for each one of the 10 most frequent categories can also be found in the literature. To facilitate the comparison of results, some of them are shown in Section 6 together with the ones obtained in our experiment.

3. Proposed approach

In this article we propose a multiclassifier based document categorization system which classifies documents represented in a reduced dimensional vector space. Different training databases are generated from the original training dataset in order to construct the multiclassifier. The k-NN classification algorithm is used which, according to each training database, makes a prediction for the testing documents. Finally, a Bayesian voting scheme is used to definitively assign category labels to the testing documents.

In the rest of this section, we provide details of our classification system proposal, particularly the way we construct the multiclassifier and how we obtain and combine the category label predictions. We also explain why and how we perform the dimensionality reduction to the vectors which represent documents.

3.1. The SVD dimensionality reduction technique

The classical vector space model (VSM) has been successfully employed to represent documents in text categorization tasks. The

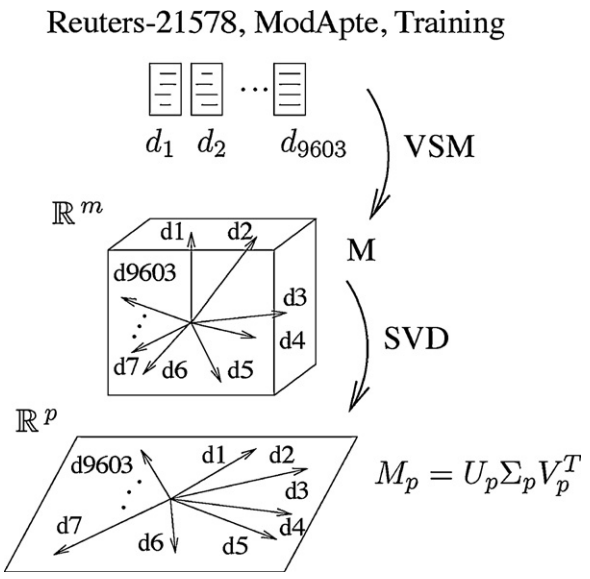


Fig. 1. Vectors in the VSM are projected to the reduced space by using SVD.

newer method of latent semantic indexing (LSI)[2] [6] is a variant of the VSM [25] in which documents are represented in a lower dimensional space by applying the singular value decomposition (SVD) technique. LSI is based on the assumption that there is an underlying latent semantic structure in the term-document matrix that is corrupted by the wide variety of words used in documents. This is referred to as the problem of polysemy and synonymy. The basic idea is that if two document vectors represent two very similar topics, many words will co-occur on them, and they will have very close semantic structures after dimension reduction.

The SVD technique consists in factoring the term-document matrix M into the product of three matrices, $M = U\Sigma V^T$ where Σ is a diagonal matrix of singular values in non-increasing order, and U and V are orthogonal matrices of singular vectors (term and document vectors, respectively). Matrix M can be approximated by a lower rank M_p which is calculated by using the p largest singular values of M. This operation is called dimensionality reduction, and the p-dimensional space to which document vectors are projected is called the reduced space. The right dimension p must be chosen for successful application of the LSI/SVD technique. However, since there is no theoretical optimum value for p, potentially expensive experimentation may be required to determine it. A very good overview about the SVD technique and the way it is used in information retrieval systems can be found in [1].

For document categorization purposes [8], the testing document q is also projected to the p-dimensional space, $q_p = q^T U_p \Sigma_p^{-1}$, and the cosine is usually calculated to measure the semantic similarity between training and testing document vectors. The use of this reduced dimensional vector representation facilitates conceptual indexing, so that related documents which may not share common terms are still represented by nearby vectors in a p-dimensional vector space.

In Fig. 1 an illustration of the document vector projection can be seen. Documents in the training collection are represented by using the term-document matrix M, and each one of the documents is represented by a vector in the $\mathbb{R}^m$ vector space like in the traditional vector space model (VSM) scheme. Then, the dimension p is selected, and by applying SVD vectors are projected to the reduced

[2] http://lsi.research.telcordia.com, http://www.cs.utk.edu/ lsi.

Please cite this article in press as: A. Zelaia, et al., A multiclass/multilabel document categorization system: Combining multiple classifiers in a reduced dimension. Appl. Soft Comput. J. (2011), doi:10.1016/j.asoc.2011.06.002

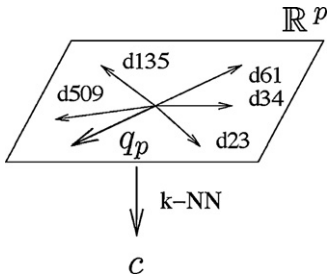


Fig. 2. The k-NN classifier is applied to q_p testing document and c category label is predicted.

space $\mathbb{R}^p$. Documents in the testing collection will also be projected to the same reduced space.

3.2. The k nearest neighbor classification algorithm (k-NN)

k-NN is a distance based classification approach. According to this approach, given an arbitrary testing document, the k-NN classifier ranks its nearest neighbors among the training documents, and uses the categories of the k top-ranking neighbors to predict the categories of the testing document [4]. In the approach presented in this article, the training and testing documents are represented as reduced dimensional vectors in the lower dimensional space, and in order to find the nearest neighbors of a given document, the cosine similarity measure is calculated.

In Fig. 2 an illustration of this phase can be seen, where some training documents and a testing document q_p are projected in the reduced space $\mathbb{R}^p$. The nearest to the q_p testing document are considered to be the vectors which have the smallest angle with respect to q_p, and thus the highest cosine. According to the category labels of the nearest documents, a category label prediction, c, will be made for testing document q_p. Given the reduced size of the training database used, and to look for a variability in category labels, we set k to 1. This implies that the k-NN classifier will give a category label prediction based on the categories of the nearest one.

We decided to use the k-NN classifier because it performs best among the conventional methods [16,30,27,31] on the Reuters-21578 database and because we obtained good results in our previous work on text categorization for documents written in Basque [33]. Besides, the k-NN classification algorithm can be easily adapted to multiclass/multilabel categorization problems such as Reuters.

3.3. The induction and combination of multiple classifiers

The combination of multiple classifiers consists in applying different classifiers to the same classification task and in combining their outcome appropriately. By doing so, a better performance than that of any of the individual components is sought [14]. There are different ways to combine classifiers which improve accuracy over single classifiers. To decide which classifiers to use and how to combine the different outcomes becomes extremely relevant. Concerning the classifiers choice, several approaches have been studied, among them: bagging [2], which uses more than one model of the same paradigm in order to reduce errors; boosting [10], in which a different weight is given to different training documents; random forests [3], an improvement over bagging; bi-layer classifiers [29], where different models from different paradigms are combined in a parallel mode to obtain individual decisions to be used as predictor variables for a new classifier which makes the final decision. There are other combination approaches in serial or semi-parallel architectures [22]. A good review about classifier combination methods can be found in [18].

Methods for voting classification algorithms have been shown to be very successful in improving the accuracy of single classifiers. Typically, three patterns are used: unanimity, simple majority and plurality. As a multiclass problem is to be dealt with, plurality seems to be the most appropriate method. Within the different approaches present in the literature (Weighted Linear Combination, Dynamic Classifier Selection, Naive Bayesian voting, etc.) [26], and due to the characteristics of the categorization task, a Bayesian Weighted voting system has been used in this paper [15].

In our experiment we decided to construct a multiclassifier via bagging. In bagging, a set of training databases is generated by selecting n training documents randomly with replacement from the original training database TD of n documents. When a set of $n_1 < n$ training documents is chosen from the original training collection, the bagging is said to be applied by random subsampling [2]. This is the approach used in our work and the n_1 parameter has been selected via tuning. In Section 4.3 the selection will be explained in a more extended way.

Given a testing document q, each one of the classifiers will make a label prediction based on each one of the training databases. Regarding the combination of the different outcomes, it has to be pointed out that single voting scheme obtains worse results than Bayesian voting in the experiments carried out. In Bayesian voting [7], a confidence value $cv_{c_j}^i$ is calculated for each training database and category c_j to be predicted. These confidence values have been calculated based on the training collection. Confidence values are added by category; the category c_j that gets the highest value is finally proposed as a prediction for the testing document.

In Fig. 3 an illustration of the whole experiment can be seen. First, vectors in the VSM are projected to the reduced space by using SVD. Next, random subsampling is applied to the training database TD to obtain different training databases. Then the k-NN classifier is applied to each one of the training databases $TD_1, \ldots, TD_L$ to make category label predictions. Finally, Bayesian voting is used to combine predictions. c' will be the final category label prediction of the categorization system for testing document q. In some cases, a second category label c'' will also be assigned to the testing document. The conditions required to give this second category label prediction are explained in Section 4.3.

4. Experimental setup

In this section we describe the document collection used in our experiment and give an account of the preprocessing techniques applied and the parameters tuned.

4.1. Document collection

As previously mentioned, the experiment reported in this article was carried out for the Reuters-21578 dataset[3] compiled by David Lewis and originally collected by the Carnegie group from the Reuters newswire in 1987. One of the most widely used training/testing divisions is used, the "ModApte" split, in which 75% of the documents (9603 documents) are selected for training and the remaining 25% (3299 documents) to test the accuracy of the classifier.

Document distribution over categories in both the training and the testing sets is very unbalanced: the 10 most frequent categories, Top-10, account for 75% of the training documents; the rest is distributed among the other 108 categories.[4]

[3] http://daviddlewis.com/resources/testcollections.
[4] It has to be noted that unlabeled documents have been preserved, and thus, our classification system treats unlabeled documents as documents of a new category.

Please cite this article in press as: A. Zelaia, et al., A multiclass/multilabel document categorization system: Combining multiple classifiers in a reduced dimension. Appl. Soft Comput. J. (2011), doi:10.1016/j.asoc.2011.06.002

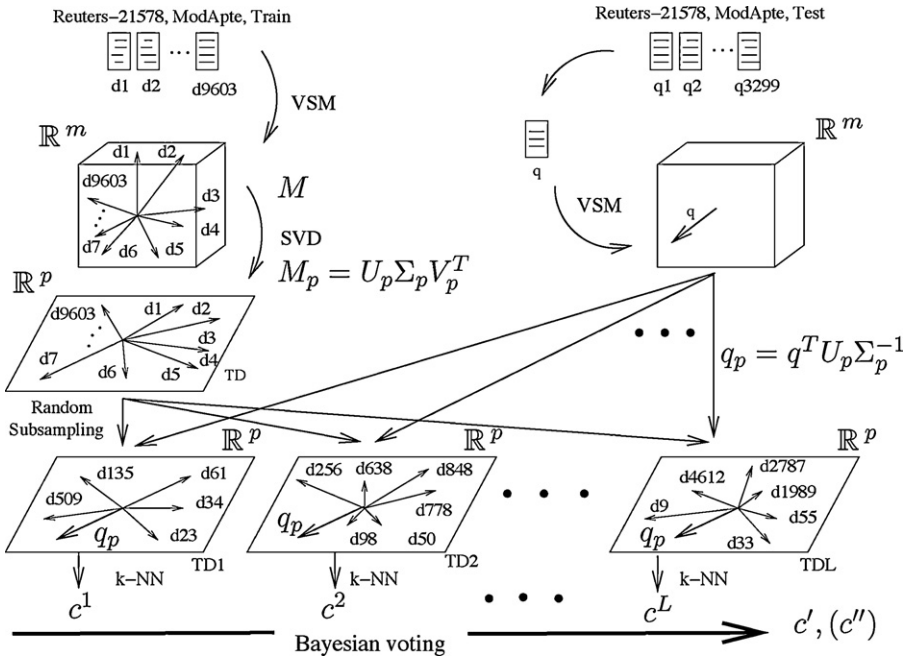


Fig. 3. Proposed approach for multiclass/multilabel document categorization tasks.

According to the number of labels assigned to each document, many of them (19% in training and 8.48% in testing) are not assigned to any category, and some of them are assigned to 12. We decided to keep the unlabeled documents in both the training and testing collections, as it is suggested in [19].[5]

4.2. Preprocessing

The original format of the text documents is in SGML. A preprocessing was performed to filter out the unused parts of a document. Only the title and the body text were preserved, punctuation and numbers were removed and all letters were converted to lowercase. The tools provided in the web[6] were used to extract text and categories from each document. Moreover, the training and testing documents were stemmed by using the Porter stemmer [23].[7] By doing so, case and flection information were removed from words. The experiment was carried out for the two forms of the document collection: the Bag-of-Words (BoW) and the Bag-of-Stems (BoS).

For the dimension reduction, it has to be noted that after preprocessing was applied, the training document collection was represented by 15,591 features, and so, the size of the training matrix created was $15,591 \times 9603$ for the BoW corpus. After applying the Porter stemmer, the number of features was reduced to 11,114, and a matrix of $11,114 \times 9603$ was obtained for the BoS corpus. By applying the SVD, the number of features in both corpora was reduced significantly. Experiments have been performed for dimensions $p = 100, \ldots, 1000$, although in this article we only publish results obtained for $p = 100, 300, 500$, because results obtained for higher dimensions were less significant.

Thus, and as a consequence of having two forms of the document collection (BoW and BoS) and three different dimensions ($p = 100$, 300, 500), we have six different representations of documents: BoW-100, BoW-300, BoW-500, BoS-100, BoS-300 and BoS-500. The experiment was performed and results evaluated for each one of the six different representations. In the illustration of the experiment in Fig. 3, each one of the six representations corresponds to the original training database (TD) to which random subsampling is applied.

4.3. Parameters

In the experimental approach proposed in this article, there were some decisions that needed to be made. We had to determine

(1) how many documents should be selected from the TD to create each one of the training databases: parameter n_1;
(2) which were the cases when a second category label should be assigned to a testing document after Bayesian voting was applied: parameter λ;
(3) which was the appropriate number of training databases that should be created: parameter L.

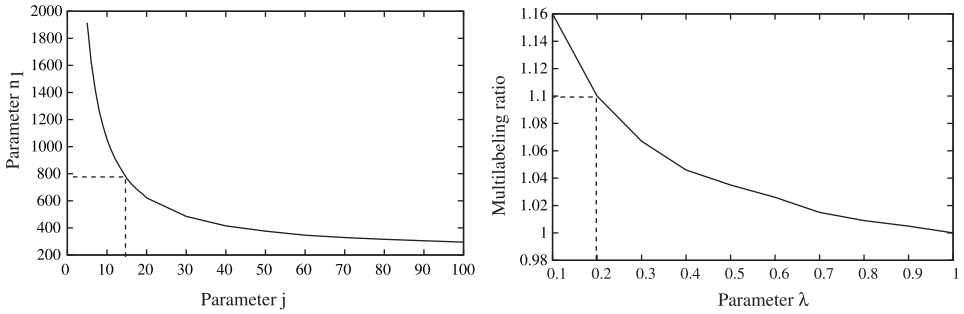
Therefore, a parameter tuning phase was carried out in order to fix the three parameters. This parameter tuning phase was not carried out based on the Reuters original training/testing document collections. Instead, a training subcollection (75%, 7242 docs.) and a validation subcollection (25%, 2361 docs.) were created randomly from the original training document collection of 9603 documents. This subdivision preserved the proportion of documents by category in the original training document collection. For categories with a very low number of documents (less than 4), at least one document in the training subcollection was kept.

In the following subsections, the three parameters are briefly introduced and in the next section the tuning process is explained in more detail.

4.3.1. The size of each of the training databases: parameter n_1

As it was mentioned earlier, the multiclassifier is implemented by random subsampling, where a set of $n_1 < n$ training documents is chosen from the original training collection of n documents

[5] In the "ModApte" Split section it is suggested as follows: "If you are using a learning algorithm that requires each training document to have at least TOPICS category, you can screen out the training documents with no TOPICS categories. Please do NOT screen out any of the 3299 documents—that will make your results incomparable with other studies."
[6] http://www.lins.fju.edu.tw/ tseng/Collections/Reuters-21578.html.
[7] http://tartarus.org/martin/PorterStemmer/.



(a) Parameter $n1$: the size of each training database

(b) Parameter λ: the threshold for multilabeling

Fig. 4. Tuning of parameters n_1 and λ.

at random ($n = 7242$ during the tuning phase, $n = 9603$ during the experimental phase). Consequently, the size of each training database will vary depending on the value of n_1. The selection of different numbers of documents was experimented, according to the following equation:

$$n_1 = \sum_{i=1}^{115}(2 + \lfloor \frac{t_i}{j} \rfloor), \qquad j = 5, \ldots, 100 \tag{1}$$

where t_i is the total number of training documents in category c_i. Note that values for t_i vary depending on the training document collection referred to, i.e. the original or the subcollection created for the tuning phase.

By dividing t_i by j, the number of documents selected from each category preserves the proportion of documents per category in the original one. However, it has to be taken into account that some of the categories have a very low number of documents assigned to them. By adding 2, at least 2 documents will be selected from each category. In Fig. 4(a) the variation of the parameter n_1 depending on the value of j is outlined.

4.3.2. The threshold for multilabeling: parameter λ

Being Reuters-21578 a multilabel database, we decided to construct a classifier that, in some cases, assigns a second category label to a testing document. The multilabeling ratio we define is based on confidence values which are calculated in the following way: by using the training data, a missclassification matrix is constructed for each of the classifiers, where value in row m column n represents the number of documents that, belonging to class n have been classified as being of class m. The confidence value cv_{c_m} for category c_m is the percentage of documents correctly classified into a given category c_m among those classified as belonging to this category c_m. These confidence values are used as a weight value in Bayesian voting. Given that c' is the category with the highest confidence value in Bayesian voting and c'' the next one, the second category label c'' is assigned when the following relation is true:

$$cv_{c''} > cv_{c'} \times \lambda, \qquad \lambda = 0.1, 0.2, \ldots, 0.9, 1 \tag{2}$$

By applying Eq. (2), and depending on the value of parameter λ, the difference between the confidence values calculated for categories c' and c'' is measured. The lowest multilabeling ratio is obtained when $\lambda = 1$, in which case the classifier becomes single-label because the relation in the equation will never be hold. By reducing the value of parameter λ, different thresholds for the multilabeling ratio are experimented. In Fig. 4(b) the variation of the multilabeling ratio depending on the value of parameter λ is outlined.

4.3.3. The number of classifiers: parameter L

The classification approach presented in this article is based on the construction of a multiclassifier which uses different training databases to make category label predictions. The number of classifiers to construct is a parameter that needs to be tuned. Given that it is computationally too expensive to tune the three parameters at the same time, we decided to tune parameter L after the rest of parameters were tuned and set to their optimal values. So, based on our previous work [34], we decided to create 30 training databases and to tune parameters n_1 and λ previously introduced. Once n_1 and λ were set to their optimal values, parameter L was tuned by creating different numbers of training databases, ranging L from 10 to 300.

5. Parameter tuning

5.1. Tuning the parameter n_1: the size of each training database

In order to decide the optimal value for parameter n_1, the classification experiment was carried out varying j from 5 to 100 according to Eq. (1). Results obtained by using the multiclassifier system composed by 30 k-NN single classifiers appear graphically represented in Fig. 5. In fact, graphics are restricted to the range of parameter j where best results were obtained: $j = 5, \ldots, 20$.

A first glance at the graphics leads us to pay attention to Fig. 5(c) and (d) where the highest results for Top-10, R(90) and R(115) are obtained. Actually, the best ones for R(90) are obtained for the BoS-300 validation subcollection (an average microaveraged F_1 score of 87.57%), even though they are just slightly better than the ones obtained for the BoW-300 subcollection (87.42%); they both correspond to $j = 15$ (see discontinuous lines drawn in the graphics). According to Eq. (1), this implies that each of the training databases will be created by selecting $n_1 = 766$ documents in the tuning phase (see discontinuous line in Fig. 4(a)). It has to be noted that, being j the first parameter to be tuned, results depicted in Fig. 5 correspond to the average of the results obtained for $\lambda = 0.1, \ldots, 1$.

5.2. Tuning the parameter λ: the threshold for multilabeling

The tuning of parameter n_1 in the previous subsection was made based on the average of microaveraged F_1 scores obtained for $\lambda = 0.1, \ldots, 1$ and led us to set j to 15. In Table 2 results calculated for the six forms of the document subcollections are shown explicitly for $j = 15$. It can be seen that in most cases the results obtained by using 300 dimensions are superior than the ones obtained by using 100 and 500 dimensions.

However, it is not clear whether the stemming process improves results; by observing the average of results at the bottom of the table, the best ones are obtained for the stemmed documents

Please cite this article in press as: A. Zelaia, et al., A multiclass/multilabel document categorization system: Combining multiple classifiers in a reduced dimension. Appl. Soft Comput. J. (2011), doi:10.1016/j.asoc.2011.06.002

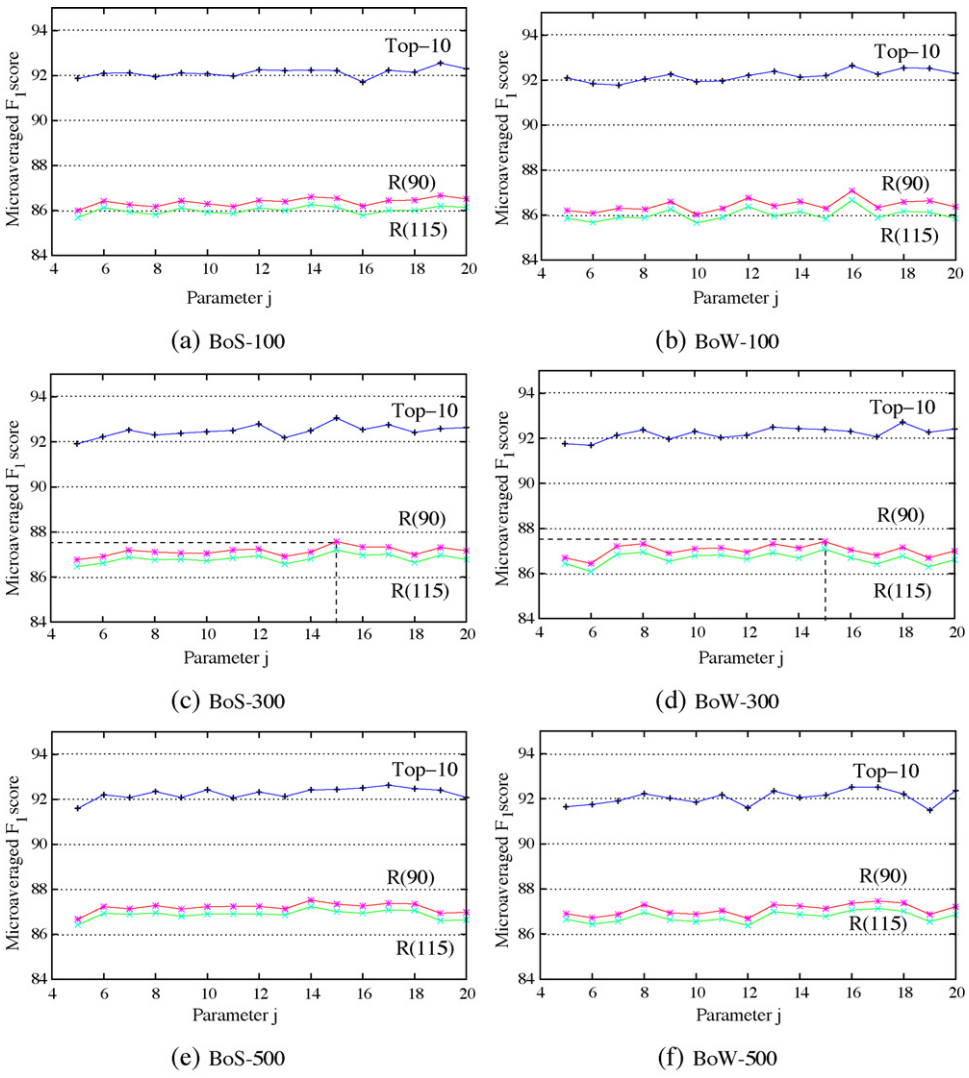


Fig. 5. Average microaveraged F_1 scores measured for the validation subcollection of documents; tuning parameter j.

(BoS-300, 87.57%), but they do not differ much from the ones obtained for the BoW-300 corpus (87.42%) (see also Fig. 5(c) and (d)). The best microaveraged F_1 result in Table 2 without calculating the average (88.96%) is obtained for the BoW-300 corpus.

In any case, the optimal results set parameter λ to 0.2, which according to Eq. (2), gives a multilabeling ratio of 1.1 categories per document in the validation subcollection (see Fig. 4(b)).

Given that the best results were obtained by using 300 dimensions, on the remaining of the tuning phase and during the experimental phase, only the BoW-300 and BoS-300 corpora were used.

Table 2
Microaveraged F_1 scores for $j = 15$ evaluated for the R(90) category subset by using the validation subcollection of documents; tuning parameter λ.

λ	BoS-100	BoS-300	BoS-500	BoW-100	BoW-300	BoW-500
0.1	87.28	88.42	87.90	86.85	88.46	87.89
0.2	87.68	**88.83**	88.54	87.30	**88.96**	88.65
0.3	87.37	88.73	88.55	87.03	88.42	88.42
0.4	86.87	88.40	88.06	86.74	87.97	87.73
0.5	86.60	87.93	87.70	86.34	87.63	87.24
0.6	86.32	87.48	86.93	86.12	87.19	86.86
0.7	86.07	86.98	86.75	85.87	86.77	86.35
0.8	86.00	86.49	86.50	85.68	86.43	86.28
0.9	85.68	86.37	86.32	85.53	86.34	86.06
1	85.57	86.08	86.14	85.40	86.04	85.80
Avg	86.54	**87.57**	87.34	86.29	**87.42**	87.13

5.3. Tuning the parameter L: the number of classifiers

Finally, and being aware that parameters n_1 and λ were tuned by creating 30 training databases ($L = 30$), we proceeded to optimize the number of classifiers to create for the final multiclassifier system, i.e. the number of individual k-NN algorithms to be used by the multiclassifier in order to combine opinions by Bayesian voting. The creation of different numbers of training databases, $L = 10, \ldots,$ 300 was experimented, and results were evaluated for $j = 15$ and $\lambda = 0.2$.

Fig. 6 shows results obtained for both the BoS-300 and the BoW-300 corpora. Graphics seem to suggest that a minimum number of classifiers (around 100) is needed for the multiclassifier system to give promising results. For a higher number of classifiers, the behavior of the system seems to stabilize. The best results for the R(90) category subset sets parameter L to 120 for the BoS-300 corpus (89.86%) and L to 190 for the BoW-300 corpus (89.52%). Once again, final results obtained for BoS-300 and BoW-300 are very similar. That is why it was decided to perform the final experiment for both forms by creating 120 and 190 classifiers, respectively.

Please cite this article in press as: A. Zelaia, et al., A multiclass/multilabel document categorization system: Combining multiple classifiers in a reduced dimension. Appl. Soft Comput. J. (2011), doi:10.1016/j.asoc.2011.06.002

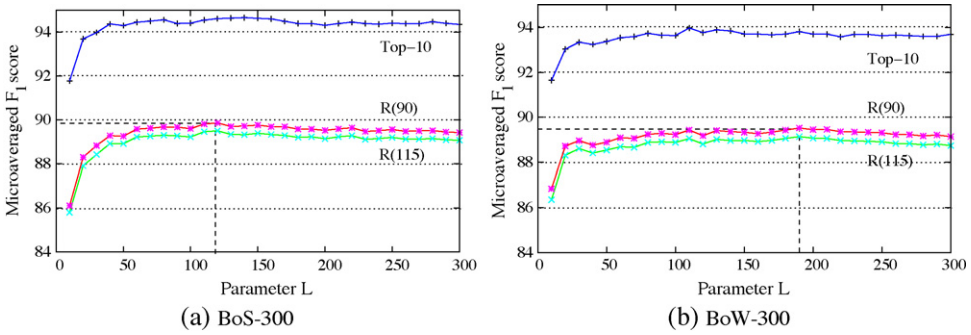


Fig. 6. Microaveraged F_1 scores for $j = 15$ and $\lambda = 0.2$: tuning parameter L.

Table 3
F_1 scores for Reuters-21578, ModApte split obtained for BoS (Bag-of-Stems) and BoW (Bag-of-Words) by using 300 dimensions in the reduced vector space representation.

Our results	Microaveraged scores			Macroaveraged scores		
	Top-10	R(90)	R(115)	Top-10	R(90)	R(115)
BoS-300	94.07	**88.26**	**88.26**	84.41	**52.86**	**41.58**
BoW-300	**94.10**	88.00	87.90	**85.30**	51.04	40.10
Single-BoS-300	83.18	75.59	75.52	59.51	33.23	26.20
Single-BoW-300	82.78	75.26	75.22	59.13	33.92	26.74

Table 4
Best results found in the literature. Results in [5] show the mean of the scores obtained by using different text classifiers.

Results reported by	Microaveraged scores		
	Top-10	R(90)	R(115)
Gao et al. [11]	93.07	88.42	–
Kim et al. [17]	92.21	87.11	–
Gliozzo and Strapparava [13]	92.80	–	–
Yang and Liu [31]	–	85.67	–
Schapire and Singer [27]	–	85.30	–
Debole and Sebastiani [5]	85.20	78.70	78.40

6. Experimental results

The final experiment was conducted with the optimal values for parameters set in the previous section: $j = 15$, $\lambda = 0.2$ and parameter $L = 120$ for the BoS-300 and $L = 190$ for the BoW-300. Results published in this section were calculated by evaluating results obtained for the original Reuters-21578 training-testing document collections. This implies a variation on the final size of each training database to $n_1 = 961$ (see Eq. (1)).

Table 3 shows microaveraged and macroaveraged F_1 scores obtained for the three category subsets. The first thing we want to emphasize is that, as far as we know, the microaveraged evaluation for the Top-10 category subset we achieve is the best one reported so far in the literature: 94.10% microaveraged F_1 score for BoW-300 and 94.07% for BoS-300. Moreover, it has to be noted that these results were obtained by using a pure ModApte split, i.e. without eliminating unlabeled documents. In addition, it is important to make clear that the evaluation was made after all documents in the testing collection were classified.

Results obtained for the R(90) category subset are among the best found in the literature (see Tables 3 and 4 to compare). They reach up to 88.26% microaveraged F_1 score, although they do not outperform results published in [11]. However, it should be noted that in the aforementioned work unlabeled documents were removed from training and testing document collections, and that the classification process was simplyfied by using only R(90) categories.

Results obtained for the R(115) category subset are analogous to the ones obtained for the R(90) subset as it could be expected, since the difficulty of these subsets is similar.

Regarding the macroaveraged performance achieved by our classification system, it can be said that even though the aim was not to optimize macroaveraged results, the system presented in this article behaves positively. Unfortunately, most of the researchers do not report macroaveraged results and consequently it is not easy to establish comparisons. In [11] a macroaveraged F_1 score of 87.78% for the Top-10 subset and 55.57% for the R(90) is reported. They are higher than the ones presented in this article, but once again, it has to be taken into account that the ModApte split is not used in the same way, and therefore, results are not directly comparable.

Analyzing results obtained for BoS-300 and BoW-300, it can be observed that the stemming process slightly improves results in most of the cases (R(90) and R(115)). In our previous work [33] we

Table 5
Results for Reuters-21578, ModApte split, evaluated for the Top-10 category subset, reported by: (a) [28], (b) [35], (c) [11], (d) [17], BoS-300: our F_1 results for BoS-300, BoW-300: our F_1 results for BoW-300.

Category	Train	Test	(a)	(b)	(c)	(d)	BoS-300	BoW-300
Earnings	2877	1087	97.78	98.4	97.9	98.25	**99.45**	**99.45**
Acquisitions	1650	719	95.69	95.4	96.8	95.57	**98.47**	97.86
Money-fx	538	179	76.44	76.0	82.6	75.78	89.58	**89.84**
Grain	433	149	**93.41**	90.3	90.6	92.88	88.37	87.21
Crude	389	189	88.63	84.9	89.7	88.11	**89.87**	89.65
Trade	369	118	75.41	76.3	80.7	75.32	89.54	**90.76**
Interest	347	131	72.95	75.7	79.2	77.99	83.06	**85.83**
Ship	197	89	80.96	83.6	**87.8**	84.09	75.86	73.61
Wheat	212	71	**89.59**	88.5	87.0	84.14	68.53	71.53
Corn	182	56	**89.43**	88.1	89.1	87.27	61.36	67.31
Macroaveraged scores			86.03	85.72	**88.14**	85.94	84.41	85.30
Microaveraged scores					93.07	92.21	94.07	**94.10**

Please cite this article in press as: A. Zelaia, et al., A multiclass/multilabel document categorization system: Combining multiple classifiers in a reduced dimension. Appl. Soft Comput. J. (2011), doi:10.1016/j.asoc.2011.06.002

verified that gain is higher when the stemming process is applied to a highly inflected language.

Results obtained by a single k-NN classifier ($L = 1$, $\lambda = 1$) are shown in Table 3, both for the stemmed (single-BoS-300) and not stemmed (single-BoW-300) corpus, in order to see to what extent the combination of multiple classifiers used in the experiment increases results. Certainly, the use of the multiclassifier contributes to improve results considerably; from an increase of more than 10 points for the microaveraged F_1 scores evaluated for the Top-10 by using the BoS-300 corpus (from 83.18% to 94.07%) to an increase of more than 26 points for the macroaveraged Top-10 BoW-300 (from 59.13% to 85.30%).

In Table 5 the F_1 scores for each one of the 10 most frequent categories are presented. Columns labeled as "Train" and "Test" show the number of documents assigned to each category in the Reuters-21578, ModApte split. The following four columns, labeled as (a)–(d), show F_1 scores reported in the literature. The last two columns, BoS-300 and BoW-300, present F_1 scores obtained by applying the approach proposed in this article.

Results obtained for each of the 10 categories are, in general, very good. Values marked in bold (best results for each category) show that, compared to the results published in the references mentioned in the table, our system obtains the best in 6 out of 10 of the categories. When these results are microaveraged, they are still better than the ones reported by some of the researchers. However, when macroaveraged, results do not improve. This may be because our classification system might not be suited for smaller categories i.e., "Wheat" and "Corn".

7. Conclusions and future work

In this article we present an approach for multiclass/multilabel document categorization problems which consists in a multiclassifier system based on the k-NN algorithm. The classifier was evaluated for the Reuters-21578, ModApte split testing collection, which is a multiclass and multilabel document collection. The microaveraged F_1 scores obtained are among the best reported in the literature, and the macroaveraged performance achieved by our classification system shows a positive behaviour.

Results obtained show that the construction of a multiclassifier, together with the use of Bayesian voting to combine category label predictions, plays an important role in the improvement of results.

A great methodological effort was put into the experimental phase. There were some parameters that needed to be set, but it was not possible to test all the possibilities because of computational load. To compensate, we decided to perform a tuning phase in a sound way by setting parameter n_1, λ and L, in that order, to their optimal values.

We also want to emphasize that we used the SVD dimensionality reduction technique in order to reduce the vector representation of documents. By doing so, documents that originally were represented by 15,000 features in the Bag-of-Words form and by 11,000 in the Bag-of-Lemmas simplify their representation to 300 features, consequently saving space and time.

As future work, we consider adapting the system in order to change the multilabeling ratio. In fact, our system assigns one or two labels to each testing document, but changing parameter λ it should be possible to assign different numbers of labels to documents. Thus, the system could be easily adapted to classify documents in collections with a higher multilabeling ratio.

We also intend to repeat the experiments for the RCV1 Reuters corpus[8] which consists of 800,000 manually categorized documents recently made available.

[8] http://www.daviddlewis.com/resources/testcollections/rcv1/.

Acknowledgements

This work was supported in part by KNOW2 project (TIN2009-14715-C04-01), and by the Basque Country Government under the Research Team Grant.

References

[1] M. Berry, M. Browne, Understanding Search Engines: Mathematical Modeling and Text Retrieval , SIAM, Society for Industrial and Applied Mathematics, Philadelphia, 2005, ISBN: 0-89871-581-4.
[2] L. Breiman, Bagging predictors , Machine Learning 24 (2) (1996) 123–140.
[3] L. Breiman, Random Forests , Machine Learning 45 (1) (2001) 5–32.
[4] B. Dasarathy, Nearest Neighbor (NN) Norms: NN Pattern Recognition Classification Techniques , IEEE Computer Society Press, 1991.
[5] F. Debole, F. Sebastiani, An analysis of the relative hardness of Reuters-21578 subsets , Journal of the American Society for Information Science and Technology 56 (6) (2005) 584–596.
[6] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, R. Harshman, Indexing by latent semantic analysis , Journal of the American Society for Information Science 41 (1990) 391–407.
[7] T. Dietterich, Machine learning research: Four current directions , The AI Magazine 18 (4) (1998) 97–136.
[8] S. Dumais, Latent semantic analysis. , in: ARIST (Annual Review of Information Science Technology), vol. 38, 2004, pp. 189–230.
[9] S. Dumais, J. Platt, D. Heckerman, M. Sahami, Inductive learning algorithms and representations for text categorization , in: Proceedings of CIKM'98: 7th International Conference on Information and Knowledge Management, ACM Press, 1998, pp. 148–155.
[10] Y. Freund, R. Schapire, A short introduction to boosting , Journal of Japanese Society for Artificial Intelligence 14 (5) (1999) 771–780.
[11] S. Gao, W. Wu, C. Lee, T. Chua, A maximal figure-of-merit learning approach to text categorization , in: Proceedings of SIGIR'03: 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2003, pp. 174–181.
[12] S. Gao, W. Wu, C. Lee, A MFoM Learning Approach to Robust Multiclass Multi-Label Text Categorization , in: ICML'04: Proceedings of the Twenty-first International Conference on Machine Learning, 2004, pp. 329–336.
[13] A. Gliozzo, C. Strapparava, Domain kernels for text categorization , in: Proceedings of CoNLL'05: 9th Conference on Computational Natural Language Learning, 2005, pp. 56–63.
[14] T. Ho, J. Hull, S. Srihari, Decision combination in multiple classifier systems , IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (1) (1994) 66–75.
[15] J.A. Hoeting, Methodology for Bayesian Model Averaging: An Update , in: Proceedings—Manuscripts of Invited Paper Presentations, International Biometric Conference, 2002, pp. 231–240.
[16] T. Joachims, Text categorization with support vector machines: Learning with many relevant features , in: Proceedings of ECML'98: 10th European Conference on Machine Learning, 1998, pp. 137–142.
[17] H. Kim, P. Howland, H. Park, Dimension reduction in text classification with support vector machines , Journal of Machine Learning Research 6 (2005) 37–53.
[18] L.I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms , Wiley, 2004.
[19] D. Lewis, 2004. Reuters-21578 text categorization test collection. distribution 1.0. readme file (v 1.3). http://daviddlewis.com/resources/testcollections.
[20] T. Li, S. Zhu, M. Ogihara, Efficient multi-way text categorization via generalized discriminant analysis , in: Proceedings of CIKM'03: Twelfth International Conference on Information and Knowledge Management, 2003, pp. 317–324, http://doi.acm.org/10.1145/956863.956924.
[21] C.H. Li, S.C. Park, Combination of modified BPNN algorithms and an efficient feature selection method for text categorization , Information Processing and Management 45 (2009) 329–340.
[22] J.M. Martinez-Otzeta, B. Sierra, E. Lazkano, A. Astigarraga, Classifier hierarchy learning by means of genetic algorithms , Pattern Recognition Letters 27 (16) (2006).
[23] M. Porter, An algorithm for suffix stripping , Program 14 (3) (1980) 130–137.
[25] G. Salton, M. McGill, Introduction to Modern Information Retrieval , McGraw-Hill, 1983.
[26] F. Sebastiani, Machine learning in automated text categorization , ACM Computing Surveys 34 (1) (2002) 1–47.
[27] R.E. Schapire, Y. Singer, BoosTexter: a boosting-based system for text categorization , Machine Learning 39 (2/3) (2000) 135–168.
[28] S. Weiss, C. Apte, F. Damerau, D. Johnson, F. Oles, T. Goetz, T. Hampp, Maximizing text-mining performance , IEEE Intelligent Systems 14 (4) (1999) 63–69.
[29] D.H. Wolpert, Stacked Generalization , Neural Networks 5 (1992) 241–259.
[30] Y. Yang, An evaluation of statistical approaches to text categorization , Journal of Information Retrieval 1 (1/2) (1999) 69–90.
[31] Y. Yang, X. Liu, A re-examination of text categorization methods , in: 22nd Annual International SIGIR, 1999, pp. 42–49.
[32] B. Yu, Z. Xu, C. Li, Latent semantic analysis for text categorization using neural network , Knowledge-Based Systems 21 (2008) 900–904.

Please cite this article in press as: A. Zelaia, et al., A multiclass/multilabel document categorization system: Combining multiple classifiers in a reduced dimension. Appl. Soft Comput. J. (2011), doi:10.1016/j.asoc.2011.06.002

[33] A. Zelaia, I. Alegria, O. Arregi, B. Sierra, Analyzing the effect of dimensionality reduction in document categorization for basque , in: Proceedings of L&TC'05: 2nd Language & Technology Conference, 2005, pp. 72–75.
[34] A. Zelaia, I. Alegria, O. Arregi, B. Sierra, A multiclassifier based document categorization system: profiting from the singular value decomposition dimensionality reduction technique , in: Proceedings of the Workshop on Learning Structured Information in Natural Language Applications, 2006, pp. 25–32.
[35] T. Zhang, F. Oles, Text categorization based on regularized linear classification methods , Information Retrieval 4 (1) (2001) 5–31.
[36] S. Zhu, X. Ji, W. Xu, Y. Gong, Multi-labelled classification using maximum entropy method , in: SIGIR'05: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2005, pp. 274–281.

Please cite this article in press as: A. Zelaia, et al., A multiclass/multilabel document categorization system: Combining multiple classifiers in a reduced dimension. Appl. Soft Comput. J. (2011), doi:10.1016/j.asoc.2011.06.002