A Multiclassifier based Document Categorization System: profiting from the Singular Value Decomposition Dimensionality Reduction Technique

Ana Zelaia
UPV-EHU
Basque Country
ccpzejaa@si.ehu.es

Iñaki Alegria
UPV-EHU
Basque Country
acpalloi@si.ehu.es

Olatz Arregi
UPV-EHU
Basque Country
acparuro@si.ehu.es

Basilio Sierra
UPV-EHU
Basque Country
ccpsiarb@si.ehu.es

Abstract

In this paper we present a multiclassifier approach for multilabel document classification problems, where a set of k-NN classifiers is used to predict the category of text documents based on different training subsampling databases. These databases are obtained from the original training database by random subsampling. In order to combine the predictions generated by the multiclassifier, Bayesian voting is applied. Through all the classification process, a reduced dimension vector representation obtained by Singular Value Decomposition (SVD) is used for training and testing documents. The good results of our experiments give an indication of the potentiality of the proposed approach.

1 Introduction

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

In this paper we concentrate on both issues. On the one hand, we use Latent Semantic Indexing (LSI) (Deerwester et al., 1990), which is a variant of the vector space model (VSM) (Salton and McGill, 1983), in order to obtain the vector representation of documents. This technique compresses vectors representing documents into vectors of a lower-dimensional space. LSI, which is based on Singular Value Decomposition (SVD) of matrices, has showed to have 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.

On the other hand, we construct a multiclassifier (Ho et al., 1994) which uses different training databases. These databases are obtained from the original training set by random subsampling. We implement this approach by bagging, and use the k-NN classification algorithm to make the category predictions for testing documents. Finally, we combine all predictions made for a given document by Bayesian voting.

The experiment we present has been evaluated for Reuters-21578 standard document collection. Reuters-21578 is a multilabel document collection, which means that categories are not mutually exclusive because the same document may be relevant to more than one category. Being aware of the results published in the most recent literature, and having obtained good results in our experiments, we consider the categorization method presented in this paper an interesting contribution for text categorization tasks.

The remainder of this paper is organized as follows: Section 2, discusses related work on document categorization for Reuters-21578 collection. In Section 3, we present our approach to deal with the multilabel text categorization task. In Section 4 the experimental setup is introduced, and details about the Reuters database, the preprocessing applied and some parameter setting are provided. In Section 5, experimental results are presented and discussed. Finally, Section 6 contains some conclusions and comments on future work.

2 Related Work

As previously mentioned in the introduction, 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 (Sebastiani, 2002).

In the document categorization task we can find two cases; (1) the multilabel case, which means that 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), and (2) the single-label case, where exactly one category is assigned to each document. While most machine learning systems are designated to handle multiclass data[1], much less common are systems that can handle multilabel data.

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 been performed in different experimental conditions. This makes results difficult to compare among them. In fact, effectiveness results can only be compared between studies that use the same training and testing sets. In order 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 those divisions. In our experiment we use the "ModApte" split (Lewis, 2004).

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

2.1 Category subsets

Concerning the evaluation of the classification system, we restrict our attention to the TOPICS group of categories that labels Reuters dataset, which contains 135 categories. However, many categories appear in no document and consequently, and because 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 paper has been published by Debole and Sebastiani (Debole and Sebastiani, 2005) where a systematic, comparative experimental study has been carried out.

The results of the classification system we propose are evaluated according to these three category subsets.

2.2 Evaluation measures

The evaluation of a text categorization system is usually done experimentally, by measuring the 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) is the percentage of documents correctly classified into a given category, and recall (Rec) is the percentage of documents belonging to a given category that are indeed classified into it.

In general, there is a trade-off between precision and recall. Thus, a classifier is usually evaluated by means of a measure which combines precision and recall. Various such measures have been proposed. The breakeven point, the value at which precision equals recall, has been frequently used during the past decade. However, it has been recently criticized by its proposer ((Sebastiani, 2002) footnote 19). Nowadays, the F_1 score is more frequently used. The F_1 score combines recall and precision with an equal weight in the following way:

$$F_1 = \frac{2 \cdot \text{Prec} \cdot \text{Rec}}{\text{Prec} + \text{Rec}}$$

[1] Categorization problems where there are more than two possible categories.

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 will be done using *microaveraging* and *macroaveraging*. 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. See (Debole and Sebastiani, 2005; Yang, 1999) for more detailed explanation of the evaluation measures mentioned above.

2.3 Comparative Results

Sebastiani (Sebastiani, 2002) presents a table where lists results of experiments for various training/testing divisions of Reuters. Although we are aware that the results listed are microaveraged breakeven point measures, and consequently, are not directly comparable to the ones we present in this paper, F_1, we want to remark some of them. In Table 1 we summarize the best results reported for the ModApte split listed by Sebastiani.

Results reported by	R(90)	Top-10
(Joachims, 1998)	86.4	
(Dumais et al., 1998)	87.0	92.0
(Weiss et.al., 1999)	87.8	

Table 1: Microaveraged breakeven point results reported by Sebastiani for the Reuters-21578 ModApte split.

In Table 2 we include some more recent results, evaluated according to the microaveraged F_1 score. For R(115) there is also a good result, $F_1 = 87.2$, obtained by (Zhang and Oles, 2001)[2].

3 Proposed Approach

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

Results reported by	R(90)	Top-10
(Gao et al., 2003)	88.42	93.07
(Kim et al., 2005)	87.11	92.21
(Gliozzo and Strapparava, 2005)		92.80

Table 2: F_1 results reported for the Reuters-21578 ModApte split.

In the rest of this section we make a brief review of the SVD dimensionality reduction technique, the k-NN algorithm and the combination of classifiers used.

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 newer method of Latent Semantic Indexing (LSI) [3] (Deerwester et al., 1990) is a variant of the VSM in which documents are represented in a lower dimensional space created from the input training dataset. It is based on the assumption that there is some 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 used by LSI consists in factoring 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

[2] Actually, this result is obtained for 118 categories which correspond to the 115 mentioned before and three more categories which have testing documents but no training document assigned.

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

space to which document vectors are projected is called the reduced space. Choosing the right dimension p is required for successful application of the LSI/SVD technique. However, since there is no theoretical optimum value for it, potentially expensive experimentation may be required to determine it (Berry and Browne, 1999).

For document categorization purposes (Dumais, 2004), 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.

In Figure 1 we can see an ilustration of the document vector projection. 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. Afterwards, the dimension p is selected, and by applying SVD vectors are projected to the reduced space. Documents in the testing collection will also be projected to the same reduced space.

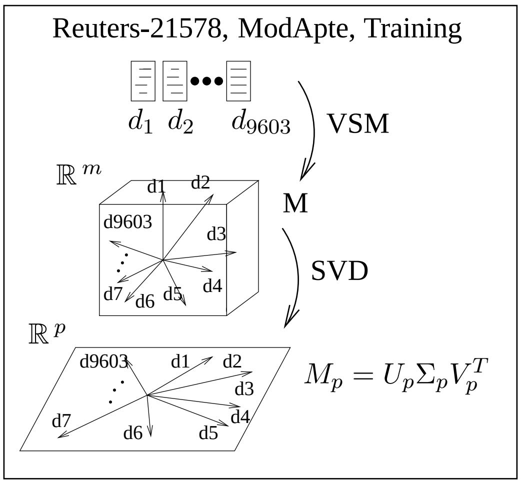
Reuters-21578, ModApte, Training
d_1 d_2 d_{9603}
VSM
$\mathbb{R}^m$
M
SVD
d1 d2 d9603 d3 d4 d7 d6 d5
$\mathbb{R}^p$
d9603 d1 d2 d3 d4 d7 d6 d5
$M_p = U_p \Sigma_p V_p^T$

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

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 (Dasarathy, 1991). In this paper, 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, we calculate the cosine similarity measure.

In Figure 2 an ilustration of this phase can be seen, where some training documents and a testing document q are projected in the $\mathbb{R}^p$ reduced space. The nearest to the q_p testing document are considered to be the vectors which have the smallest angle with q_p. According to the category labels of the nearest documents, a category label prediction, c, will be made for testing document q.

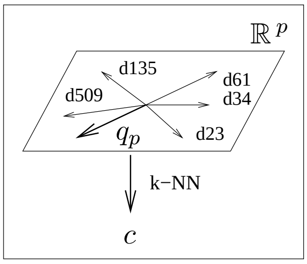

Figure 2: The k-NN classifier is applied to q_p testing document and c category label is predicted.

We have decided to use the k-NN classifier because it has been found that on the Reuters-21578 database it performs best among the conventional methods (Joachims, 1998; Yang, 1999) and because we have obtained good results in our previous work on text categorization for documents written in Basque, a highly inflected language (Zelaia et al., 2005). Besides, the k-NN classification algorithm can be easily adapted to multilabel categorization problems such as Reuters.

3.3 Combination of classifiers

The combination of multiple classifiers has been intensively studied with the aim of improving the accuracy of individual components (Ho et al., 1994). Two widely used techniques to implement this approach are *bagging* (Breiman, 1996), that uses more than one model of the same paradigm; and *boosting* (Freund and Schapire, 1999), in which a different weight is given to different training examples looking for a better accuracy.

In our experiment we have decided to construct a multiclassifier via bagging. In bagging, a set of training databases TD_i is generated by selecting n training examples drawn randomly with replacement from the original training database TD of n examples. When a set of n_1 training examples,

$n_1 < n$, is chosen from the original training collection, the bagging is said to be applied by random subsampling. This is the approach used in our work. The n_1 parameter has been selected via tuning. In Section 4.3 the selection will be explained in a more extended way.

According to the random subsampling, given a testing document q, the classifier will make a label prediction c^i based on each one of the training databases TD_i. One way to combine the predictions is by Bayesian voting (Dietterich, 1998), where a confidence value $\text{cv}^i_{c_j}$ is calculated for each training database TD_i and category c_j to be predicted. These confidence values have been calculated based on the original training collection. Confidence values are summed by category. The category c_j that gets the highest value is finally proposed as a prediction for the testing document.

In Figure 3 an ilustration 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 TD_i. Afterwards the k-NN classifier is applied for each TD_i to make category label predictions. Finally, Bayesian voting is used to combine predictions, and c_j, and in some cases c_k as well, will be the final category label prediction of the categorization system for testing document q. In Section 4.3 the cases when a second category label prediction c_k is given are explained.

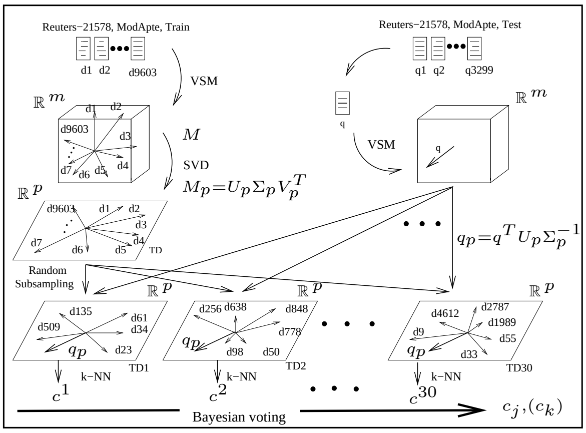
Reuters−21578, ModApte, Train
d1 d2 d9603
VSM
$\mathbb{R}^m$
d1 d2 d9603 d3 d4 d5 d6 d7
M
SVD
$M_p = U_p \Sigma_p V_p^T$
$\mathbb{R}^p$
d9603 d1 d2 d3 d4 d5 d6 d7
TD
Random Subsampling
Reuters−21578, ModApte, Test
q1 q2 q3299
q
VSM
$\mathbb{R}^m$
q
$q_p = q^T U_p \Sigma_p^{-1}$
$\mathbb{R}^p$
d135 d61 d509 d34 q_p d23
TD1
$\mathbb{R}^p$
d256 d638 d848 d778 q_p d98 d50
TD2
$\mathbb{R}^p$
d4612 d2787 d1989 d9 d55 q_p d33
TD30
k−NN
k−NN
k−NN
c^1
c^2
c^{30}
Bayesian voting
$c_j, (c_k)$

Figure 3: Proposed approach for multilabel document categorization tasks.

4 Experimental Setup

The aim of this section is to describe the document collection used in our experiment and to give an account of the preprocessing techniques and parameter settings we have applied.

When machine learning and other approaches are applied to text categorization problems, a common technique has been to decompose the multiclass problem into multiple, independent binary classification problems. In this paper, we adopt a different approach. We will be primarily interested in a classifier which produces a ranking of possible labels for a given document, with the hope that the appropriate labels will appear at the top of the ranking.

4.1 Document Collection

As previously mentioned, the experiment reported in this paper has been carried out for the Reuters-21578 dataset[4] compiled by David Lewis and originally collected by the Carnegie group from the Reuters newswire in 1987. We use one of the most widely used training/testing divisions, the "ModApte" split, in which 75 % of the documents (9,603 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 75% of the training documents; the rest is distributed among the other 108 categories.

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 have decided to keep the unlabeled documents in both the training and testing collections, as it is suggested in (Lewis, 2004)[5].

4.2 Preprocessing

The original format of the text documents is in SGML. We perform some preprocessing to filter out the unused parts of a document. We preserved only the title and the body text, punctuation and numbers have been removed and all letters have been converted to lowercase. We have

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

[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 3,299 documents - that will make your results incomparable with other studies."

used the tools provided in the web[6] in order to extract text and categories from each document. We have stemmed the training and testing documents by using the Porter stemmer (Porter, 1980)[7]. By using it, case and flection information are removed from words. Consequently, the same experiment has been carried out for the two forms of the document collection: word-forms and Porter stems.

According to the dimension reduction, we have created the matrices for the two mentioned document collection forms. The sizes of the training matrices created are 15591×9603 for word-forms and 11114×9603 for Porter stems. Different number of dimensions have been experimented $(p = 100, 300, 500, 700)$.

4.3 Parameter setting

We have designed our experiment in order to optimize the microaveraged F_1 score. Based on previous experiments (Zelaia et al., 2005), we have set parameter k for the k-NN algorithm to $k = 3$. This way, the k-NN classifier will give a category label prediction based on the categories of the 3 nearest ones.

On the other hand, we also needed to decide the number of training databases TD_i to create. It has to be taken into account that a high number of training databases implies an increasing computational cost for the final classification system. We decided to create 30 training databases. However, this is a parameter that has not been optimized.

There are two other parameters which have been tuned: the size of each training database and the threshold for multilabeling. We now briefly give some cues about the tuning performed.

4.3.1 The size of the training databases

As we have previously mentioned, documents have been randomly selected from the original training database in order to construct the 30 training databases TD_i used in our classification system. There are $n = 9,603$ documents in the original Reuters training collection. We had to decide the number of documents to select in order to construct each TD_i. The number of documents selected from each category preserves the proportion of documents in the original one. We have experimented to select different numbers $n_1 < n$ of documents, according to the following formula:

$$n_1 = \sum_{i=1}^{115} 2 + \frac{t_i}{j}, \qquad j = 10, 20, \ldots, 70,$$

where t_i is the total number of training documents in category i. In Figure 4 it can be seen the variation of the n_1 parameter depending on the value of parameter j. We have experimented different j values, and evaluated the results. Based on the results obtained we decided to select $j = 60$, which means that each one of the 30 training databases will have $n_1 = 298$ documents. As we can see, the final classification system will be using training databases which are quite smaller that the original one. This gives a lower computational cost, and makes the classification system faster.

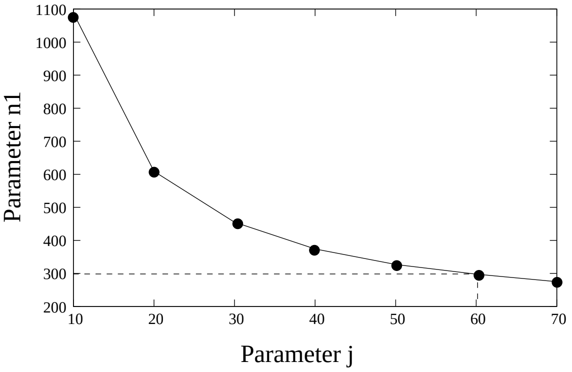

Figure 4: Random subsampling rate.

4.3.2 Threshold for multilabeling

The k-NN algorithm predicts a unique category label for each testing document, based on the ranked list of categories obtained for each training database TD_i[8]. As previously mentioned, we use Bayesian voting to combine the predictions.

The Reuters-21578 is a multilabel database, and therefore, we had to decide in which cases to assign a second category label to a testing document. Given that c_j is the category with the highest value in Bayesian voting and c_k the next one, the second c_k category label will be assigned when the following relation is true:

$$\text{cv}_{c_k} > \text{cv}_{c_j} \times r, \qquad r = 0.1, 0.2, \ldots, 0.9, 1$$

In Figure 5 we can see the mean number of categories assigned to a document for different values

[6] http://www.lins.fju.edu.tw/∼tseng/Collections/Reuters-21578.html

[7] http://tartarus.org/martin/PorterStemmer/

[8] 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

of r. Results obtained were evaluated and based on them we decided to select $r = 0.4$, which corresponds to a ratio of 1.05 categories.

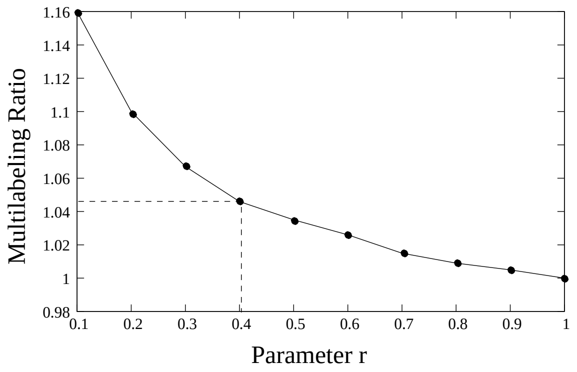

Figure 5: Threshold for multilabeling.

5 Experimental Results

In Table 3 microaveraged F_1 scores obtained in our experiment are shown. As it could be expected, a simple stemming process increases slightly results, and it can be observed that the best result for the three category subsets has been obtained for the stemmed corpus, even though gain is low (less than 0.6).

The evaluation for the Top-10 category subset gives the best results, reaching up to 93.57%. In fact, this is the expected behavior, as the number of categories to be evaluated is small and the number of documents in each category is high. For this subset the best result has been obtained for 100 dimensions, although the variation is low among results for 100, 300 and 500 dimensions. When using higher dimensions results become poorer.

According to the R(90) and R(115) subsets, the best results are 87.27% and 87.01% respectively. Given that the difficulty of these subsets is quite similar, their behavior is also analogous. As we can see in the table, most of the best results for these subsets have been obtained by reducing the dimension of the space to 500.

6 Conclusions and Future Work

In this paper we present an approach for multilabel document categorization problems which consists in a multiclassifier system based on the k-NN algorithm. The documents are represented in a reduced dimensional space calculated by SVD. We want to emphasize that, due to the multilabel character of the database used, we have adapted the classification system in order for it to be multilabel too. The learning of the system has been unique (9603 training documents) and the category label predictions made by the classifier have been evaluated on the testing set according to the three category sets: top-10, R(90) and R(115). The microaveraged F_1 scores we obtain are among the best reported for the Reuters-21578.

	Dimension reduction			
Corpus	100	300	500	700
Words(10)	93.06	93.17	93.44	92.00
Porter(10)	**93.57**	93.20	93.50	92.57
Words(90)	84.90	86.71	87.09	86.18
Porter(90)	85.34	86.64	**87.27**	86.30
Words(115)	84.66	86.44	86.73	85.84
Porter(115)	85.13	86.47	**87.01**	86.00

Table 3: Microaveraged F_1 scores for Reuters-21578 ModApte split.

As future work, we want to experiment with generating more than 30 training databases, and in a preliminary phase select the best among them. The predictions made using the selected training databases will be combined to obtain the final predictions.

When there is a low number of documents available for a given category, the power of LSI gets limited to create a space that reflects interesting properties of the data. As future work we want to include background text in the training collection and use an expanded term-document matrix that includes, besides the 9603 training documents, some other relevant texts. This may increase results, specially for the categories with less documents (Zelikovitz and Hirsh, 2001).

In order to see the consistency of our classifier, we also plan to repeat the experiment for the RCV1 (Lewis et al., 2004), a new benchmark collection for text categorization tasks which consists of 800,000 manually categorized newswire stories recently made available by Reuters.

7 Acknowledgements

This research was supported by the University of the Basque Country (UPV00141.226-T-15948/2004) and Gipuzkoa Council in a European

Union Program.

References

Berry, M.W. and Browne, M.: Understanding Search Engines: Mathematical Modeling and Text Retrieval. SIAM Society for Industrial and Applied Mathematics, ISBN: 0-89871-437-0, Philadelphia, (1999)

Breiman, L.: Bagging Predictors. Machine Learning, **24**(2), 123–140, (1996)

Cristianini, N., Shawe-Taylor, J. and Lodhi, H.: Latent Semantic Kernels. Proceedings of ICML'01, 18th International Conference on Machine Learning, 66–73, Morgan Kaufmann Publishers, (2001)

Dasarathy, B.V.: Nearest Neighbor (NN) Norms: NN Pattern Recognition Classification Techniques. IEEE Computer Society Press, (1991)

Debole, F. and Sebastiani, F.: An Analysis of the Relative Hardness of Reuters-21578 Subsets. Journal of the American Society for Information Science and Technology, **56**(6),584–596, (2005)

Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K. and Harshman, R.: Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science, **41**, 391–407, (1990)

Dietterich, T.G.: Machine-Learning Research: Four Current Directions. The AI Magazine, **18**(4), 97–136, (1998)

Dumais, S.T., Platt, J., Heckerman, D. and Sahami, M.: Inductive Learning Algorithms and Representations for Text Categorization. Proceedings of CIKM'98: 7th International Conference on Information and Knowledge Management, ACM Press, 148–155 (1998)

Dumais, S.: Latent Semantic Analysis. ARIST, Annual Review of Information Science Technology, **38**, 189–230, (2004)

Freund, Y. and Schapire, R.E.: A Short Introduction to Boosting. Journal of Japanese Society for Artificial Intelligence, **14**(5), 771-780, (1999)

Gao, S., Wu, W., Lee, C.H. and Chua, T.S.: A Maximal Figure-of-Merit Learning Approach to Text Categorization. Proceedings of SIGIR'03: 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 174–181, ACM Press, (2003)

Gliozzo, A. and Strapparava, C.: Domain Kernels for Text Categorization. Proceedings of CoNLL'05: 9th Conference on Computational Natural Language Learning, 56–63, (2005)

Ho, T.K., Hull, J.J. and Srihari, S.N.: Decision Combination in Multiple Classifier Systems. IEEE Transactions on Pattern Analysis and Machine Intelligence, **16**(1), 66–75, (1994)

Joachims, T. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. Proceedings of ECML'98: 10th European Conference on Machine Learning, Springer 1398, 137–142, (1998)

Kim, H., Howland, P. and Park, H.: Dimension Reduction in Text Classification with Support Vector Machines. Journal of Machine Learning Research, **6**, 37–53, MIT Press, (2005)

Lewis, D.D.: Reuters-21578 Text Categorization Test Collection, Distribution 1.0. http://daviddlewis.com/resources/testcollections README file (v 1.3), (2004)

Lewis, D.D., Yang, Y., Rose, T.G. and Li, F.: RCV1: A New Benchmark Collection for Text Categorization Research. Journal of Machine Learning Research, **5**, 361–397, (2004)

Porter, M.F.: An Algorithm for Suffix Stripping. Program, **14**(3), 130–137, (1980)

Salton, G. and McGill, M.: Introduction to Modern Information Retrieval. McGraw-Hill, New York, (1983)

Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys, **34**(1), 1–47, (2002)

Weiss, S.M., Apte, C., Damerau, F.J., Johnson, D.E., Oles, F.J., Goetz, T. and Hampp, T.: Maximizing Text-Mining Performance. IEEE Intelligent Systems, **14**(4),63–69, (1999)

Yang, Y. An Evaluation of Statistical Approaches to Text Categorization. Journal of Information Retrieval. Kluwer Academic Publishers, **1**,(1/2), 69–90, (1999)

Zelaia, A., Alegria, I., Arregi, O. and Sierra, B.: Analyzing the Effect of Dimensionality Reduction in Document Categorization for Basque. Proceedings of L&TC'05: 2nd Language & Technology Conference, 72–75, (2005)

Zelikovitz, S. and Hirsh, H.: Using LSI for Text Classification in the Presence of Background Text. Proceedings of CIKM'01: 10th ACM International Conference on Information and Knowledge Management, ACM Press, 113–118, (2001)

Zhang, T. and Oles, F.J.: Text Categorization Based on Regularized Linear Classification Methods. Information Retrieval, **4**(1): 5–31, Kluwer Academic Publishers, (2001)