Error-Correcting Output Codes for Multi-Label Text Categorization

Giuliano Armano[1], Camelia Chira[2], and Nima Hatami[1]

[1] Department of Electrical and Electronic Engineering
University of Cagliari
Piazza D'Armi, I-09123 Cagliari, Italy
[2] Department of Computer Science
Babes-Bolyai University
Kogalniceanu 1, Cluj-Napoca 400084, Romania

Abstract. When a sample belongs to more than one label from a set of available classes, the classification problem (known as multi-label classification) turns to be more complicated. Text data, widely available nowadays in the world wide web, is an obvious instance example of such a task. This paper presents a new method for multi-label text categorization created by modifying the Error-Correcting Output Coding (ECOC) technique. Using a set of binary complimentary classifiers, ECOC has proven to be efficient for multi-class problems. The proposed method, called ML-ECOC, is a first attempt to extend the ECOC algorithm to handle multi-label tasks. Experimental results on the Reuters benchmarks (RCV1-v2) demonstrate the potential of the proposed method on multi-label text categorization.

Keywords: Ensemble learning, Error-Correcting Output Coding (ECOC), Information filtering and retrieval, Multi-label Classification, Multi-label Text Categorization (ML-TC).

1 Introduction

Text Categorization (TC), also known as document classification, plays a key role in many information retrieval (IR) -based systems and natural language processing (NLP) applications. First research on TC goes back to Maron's [1] seminal work on probabilistic text classification. Since then, TC has been used for a number of different applications using techniques from machine learning, pattern recognition and statistics. In [3], TC applications are grouped into hierarchical categorization of web pages, word sense disambiguation, automatic indexing for boolean IR systems, document filtering and organization. Speech categorization as combination of a speech recognition and TC methods, multimedia document categorization through the analysis of textual captions, author identification for literary texts of unknown or disputed authorship, language identification for texts of unknown language, automated identification of text genre, and automated essay grading are some examples for such applications in real-world problems [4, 6] .

The traditional classification problem in pattern recognition refers to assigning any incoming sample to one of two (binary problem) or more (multi-class problem) distinct predefined classes. An even more complex scenario - called multi-label classification - is one in which the classes have overlap between each other. TC or automatically labeling natural language texts with thematic categories from a predefined set is one such task. An instance document or web page about "Persian carpet exhibition" can belong to both "economy" and "art" categories. Despite its multi-label nature, the majority of research studies on TC have considered it as single-label task by assigning the samples into only one of the existing classes. However, this approach simplifies the task and handles it using a huge bibliography of learning algorithms, yet failing to provide a complete solution to multi-label TC.

There are two main approaches in the literature to deal with multi-label classification: *(i) Problem transformation* approaches which transform the multi-label problem into one or more single-label problems, and *(ii) Algorithm adaptation* approaches which extend specific learning algorithms in order to handle the multi-label task directly. Although many approaches have been proposed based on different kinds of classifiers and architectures over a variety of application domains, there is no clear winner method over the rest (see [21] [22] for some recent surveys) and each of them has its own advantages and disadvantages.

Classifier ensembles (also known as Multiple Classifier Systems) is a paradigm based on the *divide-and-conquer* strategy to deal with complex classification problems. The main idea is to use an ensemble of simple *base-classifiers*, each applied to a sub-task, instead of hiring a single classifier expected to take care of the entire task. This strategy typically improves a classification system in terms of stability and classification accuracy (bias-variance reduction). Bootstrap aggregating (i.e., bagging) is a machine learning technique that combines a number of base-classifiers, each trained on a set of bootstrap samples of the original data [16]. The boosting strategy is a fixed point procedure aimed at iteratively generating a set of *week learners* [17]. Random Subspace Ensemble (RSE) [18] creates a set of base classifiers, each using only a (randomly determined) subset of the original feature space. RSE is particularly effective for high-dimensional classification problems. The Mixture of Experts (ME) [13] stochastically partitions the input space of the problem into a number of subspaces, so that experts become specialized on each subspace. The ME uses another expert called *gating network* to manage this process - which is trained together with the experts. Finally, Error-Correcting Output Codes (ECOC) [14] is an ensemble making strategy inspired by the coding theory which decomposes any multi-class problem into some complementary binary sub-problems using a (normally pre-defined) codematrix. The final multi-class solution is obtained by aggregating the binary outputs.

This paper proposes a method for multi-label TC called ML-ECOC created by extending the ECOC strategy. ML-ECOC modifies the coding/decoding phases of the standard ECOC algorithm making it suitable to the multi-label problems. This modification includes setting up new rules in both coding and decoding phases to avoid the occurrence of any inconsistency while handling

multi-label data. Experiments on the text mining problem of Multi-Label Text Categorization (ML-TC) show a good performance of the proposed ML-ECOC. Comparissons to the state-of-the-art methods from different perspectives are carried out and the obtained results are analysed in detail.

The rest of this paper is organized as follows: the standard ECOC algorithm presented in section 2, the proposed ML-ECOC algorithm is presented in section 3 with full details, section 4 presents the analysis of experimental results on Reuter's version 2 datasets and the comparisons with the state-of-the-art methods from litrature. Last section concludes the paper and discusses some directions of future work.

2 Error-Correcting Output Coding

ECOC is a classifier ensemble method inspired by signal transmission in information theory used to safely send and receive the data. Besides its *error-correcting* capability to recover the errors made in each sub-problem classification level, ECOC has the advantage of decomposing a multi-class problem into some binary sub-problems (*dichotomies*) in machine learning concept. Each sub-problem is tackled by a *dichotomizer* and the final solution for the multi-class problem is created by aggregating the results of the dichotomizers (divide-and-conquer principle). For this reason, ECOC performs well particularly on the problems with large number of classes for which other classifiers normally have difficulties.

Given a classification problem with N_c classes, the main idea of ECOC is to create a binary/ternary *codeword* for each class. Arranging the codewords as rows of a matrix, we define a *codematrix* M, where $M \in \{-1, 0, +1\}^{N_c \times L}$ and L is the code length (coding phase). From a learning point of view, M specifies N_c classes to train L dichotomizers, $f_1 ... f_L$. A classifier f_l is trained according to the column $M(., l)$. If $M(i, l) = +1$ then all examples of class i are positive, if $M(i, l) = -1$ then all its examples are negative *supper-class* and, finally, if $M(i, l) = 0$ none of the examples of class i participate in the training of f_l.

Let $\overline{y} = [y_1 ... y_L]$, $y_l \in \{-1, +1\}$ be the output vector of the L classifiers in the ensemble for a given input x. In the *decoding* phase, the class output that maximizes the similarity measure s (e.g. the Hamming distance) between $\overline{y}$ and row $M(j, .)$ (its codeword) is selected:

$$Class\ Label = ArgMax\ S(\overline{y}, M(j, .)) \tag{1}$$

The ECOC matrix codifies the class labels in order to achieve different partitions of classes, considered by each dichotomizer. The main coding strategies can be divided into problem-independent (or fixed) and problem-dependent. Most popular pre-designed problem-independent codeword constructions satisfy the requirement of high separability between rows and columns in order to increase error-correcting capability and diversity between dichotomies. These strategies include: *1vsA*, using N_c dichotomizers, each trained to discriminate a given class

from the rest of classes; *random techniques*, which can be divided into the *dense-random*, consisting of a binary matrix with high distance between rows with estimated length of $10 \log_2 N_c$ bits per code, and the *sparse-random* strategy based on the ternary symbol and with the estimated length of about $15 \log_2 N_c$. *1vs1* is one of the most well-known coding strategies, with $N_c(N_c - 1)/2$ dichotomizers including all combinations of pairs of classes [12]. Finally, BCH codes [2] are based on algebraic techniques from Galois Field theory and, while its implementation is fairly complex, it has some advantages such as generating ECOC codewords separated by a minimum, configurable Hamming distance and good scalability to hundreds or thousands of categories. Moreover, recently some researchers [10, 9, 11] argue that, unlike the problem-independent strategies where a codematrix is defined without considering the problem characteristics or the classification performance, the selection and the number of dichotomizers must depend on the performance of the ensemble for the problem at hand.

3 Multi-Label ECOC for TC

The first application of ECOC algorithm on TC dates back to 1999 [8, 7], However, in these studies, the authors simply use standard single-label classifiers and view the problem as a traditional multi-class classification. Since then, many researchers also used ECOC with different types of classifiers on various applications but with more or less the same assumptions. From the ECOC literature, one can conclude that there are three main possible ways to improve ECOC classifiers: (i) code matrix design, (ii) building binary classifiers, and (iii) decoding step. In TC area, the improvements are mainly limited to the second option i.e. building binary classifiers as accurate as possible. This goal is achieved in [20] by Model-Refinement strategy which is used to adjust the so-called bias in centroid classifiers. The basic idea is to take advantage of misclassified examples in the training data to iteratively refine and adjust the centroids of text data. In [19], Li et al. proposed a simple strategy to improve binary text classification via multi-class categorization (dubbed 2vM) for applications where sub-class partitions of positive and/or negative classes are available. As multi-class categorization may implicitly capture the interactions between sub-classes, detailed subclasses are expected to help differentiating the positive and negative classes with high accuracy.

The reason that all these works are limited to single-label assumption is that an *inconsistency* would occur otherwise in ECOC classification while applying to multi-label data. For instance, imagine a document d belongs to a label set $[1, 3, 5]$, each label representing a content based topic. Also imagine 5-th column of an instance (predefined or given) matrix $M^{7\times 9}$ shown in Figure 1 which is used to create dichotomizer f_5. Considering $d \rightarrow \omega = [c_1, c_3, c_5]$, now the question is which super-class sample d belongs to (+1 or -1)? According to traditional decoding of ECOC, the sample belongs to both super-classes of the dichotomy at the same time. This inconsistensy in assignment of d is not only limited to f_5 but also occures for dichotomies 3,4, 6, 7 and 8. In fact, standard ECOC algorithm

is only capable of single-label prediction for a traditional multi-class problem while it suffers from lack of capability to handle multi-label data in general. Therefore, a modification in the ECOC algorithm is required such that it can directly address multi-label data in both training the dichotomizers and label set prediction without any assumption and limitation. As mentioned before, the only way to address this issue so far was simplifying the problem to single-label classification [7, 8].

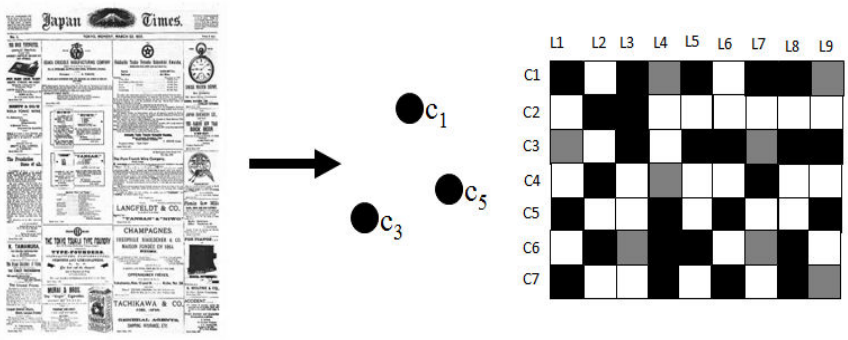

Fig. 1. An instant document d belogs to classes 1, 3 and 5 defined based on its content (Left). An instant codematrix with 7 rows (for 7 class-nodes) and 9 columns. Black, gray and white boxes represent -1, 0 and +1, respectively (Right).

Although the single-label assumption may be true in some TC applications, it certainly limits the application of ECOC to real-world multi-label cases. This is the point where ECOC algorithm requires a major modification to be applicable to multi-label problems. In the following, we introduce the ML-ECOC method to address any multi-label problem without any constraint and restricting assumption.

The main idea of ML-ECOC is to generate a codeword for each category of a TC task with only +1 (positive class) and 0 (don't care) bits. Unlike standard ECOC algorithm, where at least one +1 and one −1 bits are required at each column to define a dichotomy, to be non-zero is all ML-ECOC needs for a column. A classifier defined according to each column of the ML matrix and used to calculate degree of membership of d into a super-class which includes one or more categories. The inconsistency in the dichotomizing process is avoided by defining only positive class and *neutral* set which can not have any overlapping area. It is worth noting that a document belongs to ith positive class *if and only if* at least one of its labels from the label set is in the ith super-class. A document d (Figure 2) either should belong to positive class of ith column or its neutral set. For instance, d is a member of $2, 3, 4, 5, 6, 7$ and 8 positive class sets while should be considered as neutral for 1st and 9th.

Subsequently, it is obvious that this modification requires also different decoding strategy, since standard Euclidean or Hamming distances with $ArgMax$ labeling are not applicable anymore. Let us suppose a predicted codeword $\overline{y}_d = [\overline{y_1}...\overline{y_L}]$, $0 \leq \overline{y_l} \leq +1\}$ is a string assigned to document d (each bit representing the output of a classifier i.e. $\mathcal{P}_l(+1 \mid d)$). The posterior probability of each class using ML-ECOC is calculated as follows:

$$\mathcal{P}(c_N \mid d) = \frac{1}{\mid M(N,.) \mid} \sum_{l=1}^{L} \mathcal{P}_l(+1 \mid d) M(N,l) \tag{2}$$

For each document, ML-ECOC sorts categories by score and assigns YES to each of the t top-ranking categories. Parameter t is an integer ranging from 1 to the number of categories N_c whose value can be either specified by the user or automatically tuned using a validation set. It should be noted that when $t = 1$, this multi-label assignment turns into the standard single-label TC with $ArgMax$ rule. Obviously, it is just typical thresholding strategy adopted to ML-ECOC and the other existing throsholding methods can be applied. The generic ML-ECOC is summarized in Algorithm 1.

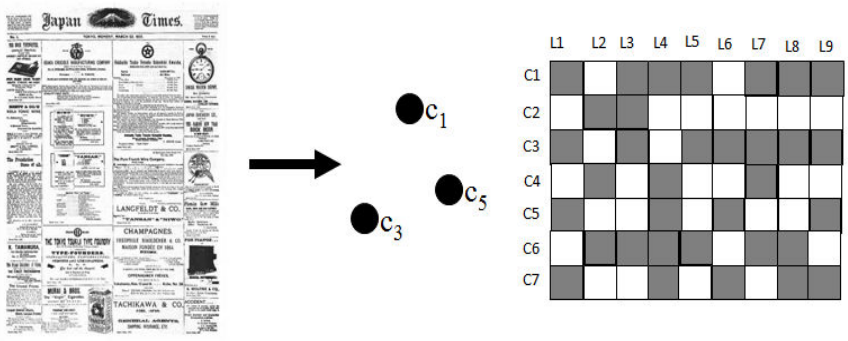

Fig. 2. ML-ECOC defines a binary codeword for each category of TC and sets up the decoding rule such that the problem decomposed in some subsets in which a positive supper-class stands against a neutral set. The number of columns, L varies depend on the coding method. Gray and white boxes represent 0 and +1 which represent positive and neutral data, respectively.

3.1 Why does ML-ECOC work?

The success of the ML-ECOC idea can be attributed to following three factors:

1. Unlike the standard TC approaches trying directly to discriminate different classes, ML-ECOC transfers the entire class space to many super-classes, which are not necessarily carrying meaningful concepts, by mixing them . This is helpful particularly to deal with what is called in the literature *Data sparsity*. This

Algorithm 1 ML-ECOC.

Input: $\mathcal{X}_t$, $\mathcal{T}_t$ training set, $\mathcal{X}_e$, $\mathcal{T}_e$ testing set and f learning algorithm.

Training:
- generate a binary codemtrix $M^{N_c \times L}$ which N_c is the number of categories and L varies with coding strategy.
- for i-th column in M:
  build (create) one-class set made of $\mathcal{T}_i^+$ and $\mathcal{T}_i^*$ supper-classes (positive and neutral sets respectively)
  train i-th classifier f_i with i-th training set

Testing:
- apply $\mathcal{X}_e$ on entire set of f_is
- create a codeword which i-th bit is $f_i(\mathcal{X}_e) = \mathcal{P}_i(+1 \mid \mathcal{X}_e)$
- calculate the posterior probability for each class using Eq. 2

- use multi-label decoding to predict label set

Output: $\bar{\omega} = [\bar{c_p}, \bar{c}_q, \bar{c}_r]$

is a measure for how much data we have for a particular dimension/entity of the model. A dataset is sparse if the number of samples for each class is not enough for a classifier to discriminate it from the rest which is normally the case in the TC problem. Therefore, mixing categories by ML-ECOC decomposing, not only used to define new class-boundaries which might provide additional information in final decision making, but also provides new one-class problems with more samples per positive class (in the case each super-class has more than one category). For instance, each super-class in first dichotomy of Figure 2 is made of 3 categories.

2. No matter which TC approach is chosen, a class-label is assigned to a document if its corresponding classifier *fires*. In fact, when a category is wrongly detected, there is no any *efficient* way to go back and fix it without the increase of the algorithm complexity and computational cost. However, in ML-ECOC there is no *dedicated* classifier for each category and decisions are made by *consensus* of all classifiers. Therefore, because of its *error-correcting* capability, even if some errors occur in the bit level, the final decision can still be reliable.

3. Another important issue arising while dealing with TC refers to *class-imbalanced* datasets where there is no balance between the positive and negative set of a category. This problem can badly affect the learning process particularly in the *Local Classifier per Category* approach when a category stands against the rest. ML-ECOC keeps more balance between two resulted positive classes and neutrals by having chance of including more than one class in the positive class set. For instance *Sparse-random* method can possibly include more than one category in a positive class resulting into more balanced data. Consequently,

efficient learning of the class boundaries by classifiers results in more accurate prediction.

4 Numerical Experiments and Results

For the text categorization experiments, we have chosen two commonly used multi-label datasets i.e. the Reuters (RCV1-V2) and TMC2007. A brief description of each is given below.

RCV1-V2: Reuters Corpus Volume1-Version2 is a large-scale dataset for text classification task. It is based on the well known benchmark dataset for text classification, the Reuters (RCV1) dataset. We use the topics full set 3 that contains (804,414) news articles. Each article is assigned to a subset of the 103 topics. A detailed description of the RCV1 dataset can be found in [5]. We preprocessed RCv1v2 documents as proposed by Lewis et al. [5] and, in addition, we separated the training set and the testing set using the same split adopted in [5]. In particular, documents published from August 20, 1996 to August 31, 1996 (document IDs 2286 to 26150) are included in the training set, while documents published from September 1, 1996 to August 19, 1997 (document IDs 26151 to 810596) are considered for testing. The result is a split of the 804,414 documents into 23,149 training documents and 781,265 test documents. In order to save computational resources, we have randomly chosen 600 documents (300 training documents and 300 testing documents) as indicated in Table 1.

TMC2007: This is the dataset used for the SIAM 2007 competition organized by the text mining workshop held in conjunction with the 7th SIAM International Conference on Data Mining [25]. This competition sponsored by NASA Ames Research Center, focused on developing text mining algorithms for document classification. It contains 28596 aviation safety reports in free text form, annotated with one or more out of 22 problem types that appear during certain flights [26]. However, in order to save computational resources, we have randomly chosen 300 training documents and 300 testing documents for our experiments. The dataset comes from human generated reports on incidents that occurred during the flights which means there is one document per incident. Text representation follows the boolean bag-of-words model. The goal was to label the documents with respect to the types of problems that were described. This is a subset of the Aviation Safety Reporting System (ASRS) dataset, which is publicly available. Some other statistics of the dataset are given in Table 1.

Table 1. The main characteristics of the selected subset of the datasets.

problem	samples	nominal	numeric	label	cardinality	density	distinct
rcv1v2	600	0	47235	103	2.642	0.026	946
tmc2007	600	49060	0	22	2.158	0.098	1341

In the applications using text categorization as the core task, the computational efficiency is crucial because of very large number of features, classes and samples. Therefore, the need for designing a simple and fast classification system is important. There are many research studies using different kinds of classifiers such as k-nearest neighbors (kNN), support vector machines (SVM), artificial neural networks (ANN), bayesian methods and rocchio classifiers [3]. However, in practice most of them are not applicable as in real-world applications, e.g. search engines and recommender systems, a just-in-time response has great importance. Among them, the naive bayes and centroid classification algorithms are extremely simple and straightforward illustrating competitive performance on text categorization problems. Moreover, they do not need to memorize a huge amount of training data as some other classifiers do (e.g. kNN) and adjust so many parameters (e.g. ANN).

For the experiments presented in the current paper, we used *centroid-based classifiers* as the ECOC dichotomizers. This means that the prototype vector or centroid vector (μ_i^+) is computed for super-class $\mathcal{T}_i^+$ as:

$$\mu_i^+ = \frac{1}{\mid \mathcal{T}_i^+ \mid} \sum_{d \in \mathcal{T}_i^+} d \tag{3}$$

where $\mid \mathcal{T}_i^+ \mid$ denotes the cardinality of set $\mathcal{T}_i^+$, i.e. the number of documents that belong to positive set in the i-th individual and d is a training document.

In the testing step, we calculate the similarity of a document d to each centroid by the *cosine* measure,

$$S(d, \mu_i^+) = \frac{d \cdot \mu_i^+}{\mid\mid d \mid\mid \ \mid\mid \mu_i^+ \mid\mid} \tag{4}$$

This similarity can be regarded as the *posterior probability* of the dichotomizer and used for i-th bit of the predicted codeword $\bar{y}_d$.

Consequently, the evaluation of methods to handle multi-label data requires different measures than those used for traditional single-label classification. Various measures are traditionally being used for evaluation of multi-label classification (particularly for document and text applications) such as *classification accuracy*, *precision*, *recall* and *F1*. These are defined below.

$$classification\ accuracy = \frac{1}{n} \sum_{d=1}^{n} \mathcal{I}(\omega_d = \bar{\omega_d}) \tag{5}$$

where $I(true) = 1$ and $I(false) = 0$ and n is the number of documents in a dataset. This is a very strict evaluation measure as it requires the predicted set to be an exact match for the true set in the label set no matter if a classifier makes a mis-classification at only one category or the entire set.

$$precision = \frac{1}{N_c} \sum_{c_i=1}^{N_c} \frac{TP_{c_i}}{TP_{c_i} + FP_{c_i}} \ \ and \ \ recall = \frac{1}{N_c} \sum_{c_i=1}^{N_c} \frac{TP_{c_i}}{TP_{c_i} + FN_{c_i}} \tag{6}$$

where TP, FP and FN stand for the true positive, false positive and false negative for each category, respectively. The F1-score which considers both the *precision* and *recall* of the test set is formulated as:

$$F1 = \frac{2precision.recall}{precision + recall} \tag{7}$$

where an $F1$ score reaches its best value at 1 and worst score at 0.

We have compared the results of the proposed method with some of commonly used TC algorithms. The standard multi-label TC methods used as baseline methods are the big-bang (global method) and Local Classifier per Category (LCC). For all these methods, centroid-based classifiers with the same parameters have been implemented. As shown in Table 2, the proposed ML-ECOC using Dense random and 2vsA codes outperforms the standard TC approaches on the selected datasets by obtaining the maximum F1 scores. One can note that the results for 2vsA code for rcv1v2 data is missing. This is because of large number of classes of RCv1v2 data which make building ECOC classifier unfeasible.

To give more detailed information, Figure 3 shows precision-recall curves corresponding to ML-ECOC and LCC approaches. Because of the superior performance on ML-TC datasets, the LCC approach is used for assessing the comparative performance of ML-ECOC. As clearly shown, the proposed ML-ECOC is able to obtain slightly better results on RCv1-v2 while always winning on TMC2007 data.

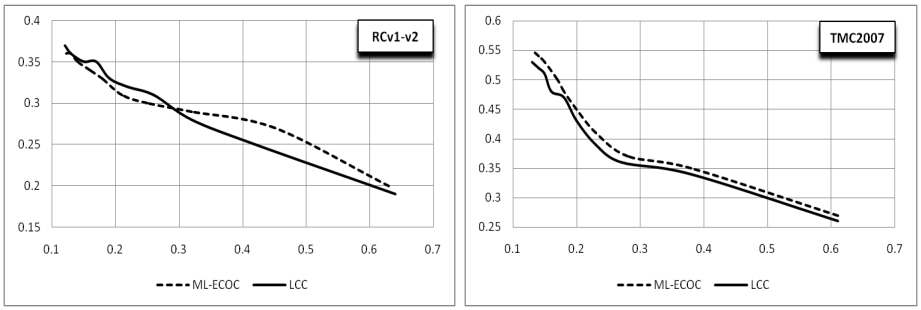

Fig. 3. Precision-Recall curves for the RCV data (left) and TMC2007 (right). X-Y axis represent the precision and recall, respectively.

5 Conclusions

An extension of the ECOC algorithm called ML-ECOC is proposed to tackle multi-label TC problems. To avoid the inconsistency in coding step, the proposed ML-ECOC method decomposes a multi-label problem into some complementary one-class sub-problems unlike the standard ECOC which builds dichotomies.

Table 2. F1 score of the proposed method (PM) using different coding strategies compared to the existing standard text categorization methods on the selected subset of rcv1v2 and tmc2007 datasets (F1 values are reported in percentage).

Problem	big-bang	LCC	ML-ECOC (drand)	ML-ECOC (2vsA)
rcv1v2	37.5	30.1	**32.9**	32.7
tmc2007	31.3	35.7	34.9	**36.5**

Multi-label relationship is taken into account in the testing phase by using a novel decoding strategy adopted for ECOC algorithm. Experimental results on Reuters datasets confirm the potential of the proposed ML-ECOC on multi-label classification with large number of categories.

Recently, some studies [23, 24] try to increase ECOC reliability by proposing a reject mechanism. One interesting future research line refers to multi-label text categorization with a reject option.

Acknowledgments. Camelia Chira acknowledges the support of Grant PN II TE 320, Emergence, auto-organization and evolution: New computational models in the study of complex systems, funded by CNCS Romania.

References

1. Maron, M. 1961. Automatic indexing: an experimental inquiry. J. Assoc. Comput. Mach. 8, 3, 404-417.
2. Lin S. and Costello. D. J. Error Control Coding, Second Edition. Prentice-Hall, Inc. (2004)
3. F. Sebastiani, Machine learning in automated text categorization, ACM Computing Surveys, Volume 34 Issue 1, 2002.
4. SABLE, C. L., Hatzivassiloglou, V. 2000. Textbased approaches for non-topical image categorization. Internat. J. Dig. Libr. 3, 3, 261-275.
5. Lewis, D. D., Yang, Y., Rose, T. G., and Li, F. (2004). RCV1: A new benchmark collection for text categorization research. Journal of Machine Learning Research, 5, 361-397
6. Schapire, R. E., Singer, Y. 2000. BoosTexter:a boosting-based system for text categorization. Mach. Learn. 39, 2/3, 135-168.
7. R. Ghani, Using Error-Correcting Codes for Text Classification, 17th International Conference on Machine Learning, 2000.
8. A. Berger, Error-Correcting Output Coding for Text Classification, In Proceedings of IJCAI-99 Workshop on Machine Learning for Information Filtering, 1999.
9. Pujol, O. Radeva, P. Vitria, J. Discriminant ECOC: A heuristic method for application dependent design of error correcting output codes, IEEE Transactions on PAMI 28 (6), 1001-1007 (2006)
10. J. Zhou, H. Peng, C. Y. Suen, Data-driven decomposition for multi-class classification, Pattern Recognition, 41 67-76 (2008)
11. Hatami, N. Thinned-ECOC ensemble based on sequential code shrinking. Expert Systems with Applications. 39 (2012) 936-947.

12. Hastie, T. Tibshirani, R., Classification by pairwise grouping, The Annals of Stat. 26 (5), 451-471 (1998)
13. Jordan, M. I. and R. A. Jacobs (1994). Hierarchical mixtures of experts and the EM algorithm. Neural Computation 6(2), 181-214.
14. T. G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. Journal of Arti?cial Intelligence Research, 2:263-286, 1995.
15. Breiman, Leo (2001). "Random Forests". Machine Learning 45 (1): 5-32.
16. Leo Breiman (1996). Bagging predictors. Machine Learning 24 (2): 123-140.
17. Yoav Freund and Robert E. Schapire A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55(1):119-139, 1997.
18. Ho, Tin (1998). "The Random Subspace Method for Constructing Decision Forests". IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (8): 832-844.
19. Baoli Li, Carl Vogel: Improving Multiclass Text Classification with Error-Correcting Output Coding and Sub-class Partitions. Canadian Conference on AI 2010: 4-15
20. Songbo Tan, Gaowei Wu, Xueqi Cheng: Enhancing the Performance of Centroid Classifier by ECOC and Model Refinement. ECML/PKDD (2) 2009: 458-472
21. G. Tsoumakas, I. Katakis, Multi-Label Classification: An Overview, International Journal of Data Warehousing and Mining, 3(3):1-13, 2007
22. Wei Bi, James T. Kwok: MultiLabel Classification on Tree- and DAG-Structured Hierarchies. ICML 2011: 17-24
23. G. Armano, C. Chira, and N. Hatami, Ensemble of Binary Learners for Reliable Text Categorization with a Reject Option, HAIS 2012, in press
24. Paolo Simeone, Claudio Marrocco, Francesco Tortorella: Design of reject rules for ECOC classification systems. Pattern Recognition 45(2): 863-875 (2012)
25. http://www.cs.utk.edu/tmw07/
26. A. Srivastava and B. Zane-Ulman. Discovering recurring anomalies in text reports regarding complex space systems. In IEEE Aerospace Conference, 2005.