Multi-Label Text Classification via Ensemble Techniques

Martin Boroš, Franky and Jiří Maršík

***Abstract*—Text classification is one of the important problems being solved in information retrieval. However, traditional single-label classifiers are no longer sufficient and multi-label approaches are becoming more relevant. There have been a lot of proposals for multi-label learners and in our work, we tried applying ensemble techniques, which have proven to be effective in solving other multi-label classification problems, to combine them. We implemented seven ensemble techniques presented in previous works and evaluated their performance. We have found that some of the ensemble classifiers outperform all of the individual classifiers, namely mean and top$_3$ techniques. We have also found Calibrated Label Ranking to be a very useful multi-label learner for text classification with a small amount of labels. Ensemble techniques have thus proven themselves to be applicable and beneficial to the domain of text classification.**

***Index Terms*—Text classification, multi-label classification, ensemble techniques**

I. Introduction

In many fields, the labeled data may be insufficient in quantity while the unlabeled may be vast. When it comes to the domain of textual data, the reasons for classification of newspaper articles, academic papers or aviation safety reports as in our work are obvious. Manual classification of existing documents or of the extensive amount of newly created documents is unfeasible. It is not only time consuming and expensive, manual annotators may also produce diverse and inadequate classifications. All these limitations have led to the development of automatic multi-label classifiers. Using ensemble techniques Sanden and Zhang [4] were able to obtain better results in multi-label music genre classification than just using a single classifier. In this paper, our aim is to apply and evaluate various ensemble techniques on multi-label text classification.

The data set examined in this paper is a subset of the Aviation Safety Reporting System (ASRS) data set. The collection tmc2007 contains 28596 NASA aviation safety reports in free text form with 49060 discrete attributes corresponding to terms in the collection. Each document is represented as a term incidence vector. The safety reports are provided with 22 labels, each of them representing a problem type that appears during flights. For our purposes, subset containing 2000 randomly selected instances was used. In order to reduce the computational costs of experiments, we used a set of 500 features. The 500 features were selected in compliance with Tsoumakas and Vlahavas [10]. For each label the X^2 feature ranking method was used to obtain a ranking of all features for that label. The top 500 features were selected based on the their maximum rank over all labels [10]. Average cardinality in the collection is 2.2.

The rest of the paper is organized as follows. In the next section we provide a summary of related work. After that, we briefly describe multi-label classification algorithms, classifiers and ensemble techniques. In section 4 we present the experiment setup. In section 5 we discuss the results and finally, section 6 concludes our work.

Manuscript received February 28, 2012; revised April 29, 2012.

The authors are from Shanghai Jiao Tong University, 800 Dong chuan Road, 200240 Shanghai(e-mail: boros@sjtu.edu.cn)

II. Related Work

Ensemble methods combine results of multiple predictive models to achieve better performance than using any of the predictive models separately. Ensemble techniques originated in bagging predictors [1], which is a method that trains multiple versions of a predictor and which relies on their plurality vote when predicting a class. Bagging predictors and other ensemble techniques [4] have shown a substantial increase in accuracy. The ensemble methods also tend to produce better results with models showing high diversity among each other.

Some work on ensemble techniques has been done by Shi, Kong, Yu and Wang [5]. They give a study of multi-label ensemble learning with focus on building a set of learners. Their proposed solution can efficiently improve the generalization ability of multi-label learning system and hence enhance the predictive performance of the classifier.

The work of Kubat, Sarinnapakorn and Dendamrongvit [3], deals with induction in multi-label text classification. They propose an induction technique of a set of subclassifiers that are applied on a same training set but use different features, and how to combine their outputs.

Sanden and Zhang [4] propose a set of ensemble techniques to obtain better results as with individual multi-label classification algorithms. These techniques also help to overcome the drawbacks of individual classifiers. Their experimental study deals with music genre classification but can be beneficial for other domains as well.

III. Multi-Label Classification Algorithms

The task of multi-label classification is to produce output of (d_i, L_i) from a collection of possible labels $C = \{c_1, c_2, \ldots, c_N\}$ for each document in the test dataset of $D_t = \{d_1, d_2, \ldots, d_m\}$, given the training dataset $D_r = \{(d_1, L_1), (d_2, L_2), \ldots, (d_n, L_n)\}$, where $L_i \subseteq C$. The approaches performed to solve this task can be divided into two categories, *problem transformation method* and *algorithm adaptation method* [6].

The *problem transformation method* works by transforming the multi-label classification problem into one

or multiple single label classification problems. Hence, using a single label classifier as a base of multi-label classifier. The *algorithm adaptation method* works by handling the multi-label classification problem directly, by extending the capability of a certain classification algorithm.

We use five different multi-label classification algorithms as components for the ensemble techniques. The details of the algorithms can be found in [8] and [11].

Random k-Labelset (RAkEL) randomly creates n different subsets $C_i \subset C$ of a label set C with each having k distinct labels. The classification model for each C_i is built using Label Powerset (LP) method that treats each member c_{ij} of powerset of C_i, $P(C_i)=\{\{\},\{c_{i1}\},\{c_{i2}\},\ldots,\{c_{i1},c_{i2},\ldots,c_{ik}\}\}$ as a single label, and uses single label classifier to produce the model. The outputs from n different models of LP classifier are combined to get the final multi-label classification result.

Calibrated Label Ranking (CLR) learns from the training data by creating a model for each distinct pair of (c_i, c_j), where $c_i \neq c_j$. An additional virtual label v is added to the model, resulting in $q(q+1)/2$ models to be built, where q is the number of labels in C. The virtual label v is used to differentiate the positive and negative labels in the final classification results. A model is built for each pair using a single label classifier that only takes training data which contain c_i or c_j (but not both) as its label. The final classification result is produced by combining all models.

Multi-label k-Nearest Neighbour (ML-kNN) extends the idea of kNN method to perform a multi-label classification. Given a test document d, we identify N(d) as the k nearest neighbours of d. The q-dimensional vector $\vec{C_d}$ is created where the i-th dimension of $\vec{C_d}$ represent the number of members in N(d) having the i-th label. The final classification result is calculated using Maximum A Posteriori (MAP) principle, that estimates how likely it is for d to have the i-th label given its j $(j<k)$ nearest neighbours have the i-th label.

Hierarchy of Multilabel Classifiers (HOMER) learns from training data by constructing a hierarchy tree of labels. The root of the tree contains all labels in C. Starting from the root node, the labels contained in the parent node are divided into k children nodes. Each children node contains a subset labels C_i, where C_i is a subset of the labels in its parent node. The process continues recursively in a top down and depth-first manner. A balanced clustering algorithm is proposed in [7] to perform the task of dividing the labels. For each internal (non-leaf) node, a meta-label μ is created, to represent the node's label as a collection of labels of its children. The multi-label classifier is then trained at each node to create a model that classifies its children. In the classification process, a test document d is classified starting from the root to get the final resulting labels.

Instance Based Logistic Regression (IBLR) combines the instance based learner algorithm with logistic regression method. The basic idea is to consider labels of neighbouring instances or documents as additional features. This approach is to ensure that the interdependencies between class labels are taken into the classification. More detailed explanation of this algorithm can be found in [2].

A. Ensemble Techniques

In this paper, we adapt ensemble techniques presented in [4] into our experiment. For a test document *d*, a multi-label classifier K_j produces two kinds of *N*-dimensional vectors, a score vector and a bipartition vector. The score vector $\vec{S^j}=\{s_1^j, s_2^j, \ldots s_N^j\}$ contains probability or confidence values s_i^j for *i*-th label assigned by a classifier K_j. The bipartition vector $\vec{B^j}=\{b_1^j, b_2^j, \ldots b_N^j\}$ contains binary prediction values b_i^j with value 1 if the classifier predicted document *d* can be assigned to *i*-th label and 0 otherwise. The ensemble techniques presented below are categorized based on the type of the output of the classifier.

B. Bipartition-Based Ensemble

Bipartition-based ensemble takes bipartition vector $\vec{B^j}$ from each classification algorithm and combines them together to get the final multi-label classification. We denote the resulting bipartition vector as $\vec{B^{ens}}=\{b_1^{ens}, b_2^{ens}, \ldots b_N^{ens}\}$. The operation to combine the vectors can use simple boolean operations or by simply calculating the number of occurrences of the positive classification for each label.

The Intersection Rule uses the boolean AND on each column i of each vector $\vec{B^j}$, denoted as $b_i^{ens}=\bigwedge_j b_i^j$. This rule represents the agreement by all classifiers.

The Union Rule uses the boolean OR. In order to get the result, each column i in vector $\vec{B^j}$ is combined as $b_i^{ens}=\bigvee_j b_i^j$ A document will be assigned the i-th label if at least one of the classifiers gives value 1 for that label.

The Majority Vote Rule takes the majority of the label assigned by the classifiers, and can be denoted as: $b_i^{ens}=1$ if A(1)>A(0) otherwise 0 where A(1) is the number of classifiers that give value 1 for i-th label and A(0) the number of classifiers that give value 0.

C. Score-Based Ensemble

Score-based ensemble works on score vector $\vec{S}$ of the classification algorithms. We denote the resulting score-based vector as $\vec{S^{ens}}=\{s_1^{ens}, s_2^{ens}, \ldots s_N^{ens}\}$. The resulting classification is determined by using comparisons or by averaging the value for each label.

The Minimum Rule takes the lowest score assigned by classifiers for each label. It is calculated as $s_i^{ens}=min_j(s_i^j)$.

Contrary to the Minimum Rule, the Maximum Rule takes the highest score assigned by classifiers for each label. It is

calculated as $s_i^{ens} = max_j(s_i^j)$.

The Mean Rule takes an average of the value for i-th label from all classifiers. For each column i, the value is calculated as $s_i^{ens} = \sum_j (s_i^j)/M$, where M is the number of classifiers used.

Top-k Rule is proposed in [4], the rule takes the average of the k largest values only. The value k is a constant determined in advance. Value is calculated as $s_i^{ens} = avg(topk_j(s_i^j))$.

IV. Experiment Setup

To perform the evaluation, we used the Mulan [9] open source library. We implemented the ensemble techniques on top of the provided interfaces and used the included evaluation framework to perform 10-fold cross-validation for all the individual multi-label learners and the ensemble techniques.

The dataset used was obtained from the Mulan website (http://mulan.sourceforge.net/datasets.html). The 28596 instances of the TextMining Challenge were randomly stripped down to about 2000 instances to make running the experiments feasible on our equipment. Instead of the full data where every document is represented by 49060 term incidence booleans, we used a stripped down version processed by feature selection which uses only the 500 most important terms. Again, this was done to make the execution of the experiments viable on our machines.

The 5 constituent classifiers were set up using provided default or customary settings. This means that RAkEL was using Label Powerset as the internal multi-label learner which in turn used J48 decision trees for single-label classification. CLR used SVM as the internal classifier, which was trained using the SMO learner with a linear kernel. ML-kNN and IBLR were initialized using the default implementation of their constructors. For HOMER, we used Binary Relevance as the internal classifier which in turn used SVMs for binary classification. The number of clusters was set to 2 and the balanced clustering method was used (these settings were taken from the evaluations done in [7]).

V. Results

The results of our experiments can be seen in Tables I and II with the best achieved values highlighted in bold-face.

First, we give an explanation of the measures and their abbreviations used in the two tables. HL (Hamming Loss), SA (Subset Accuracy), Recall (Example-based Recall), Accu. (Example-based Accuracy), MicroP (Micro-averaged Precision), MicroR (Micro-averaged Recall), MicroF_1 (Micro-averaged F_1), AP (Average Precision), CO (Coverage), OE (One Error) and RL (Ranking Loss) are all evaluation measures described in [11]. IE (Is Error) is the relative frequency of the predicted label set being different from the true label set. ESS (Error Set Size) represents the number of label pairs where an irrelevant label was ranked above a relevant one and is thus basically isomorphic to the Ranking Loss measure. MicroAUC is the micro-averaged area under the ROC curve.

Some expected statistics are conspicuously missing. Example-based precision is not given since for some examples, the positive rate of the classifier might be zero and precision is thus not defined. Therefore, the example-based precision, which is meant to be the average of such precision values, is not defined either. The same goes for the example-based F_1 measure which is a function of the precision and recall measures.

Also missing are all macro-averaged measures. This follows from the fact that for some labels, the statistic cannot be defined due to the contingency tables being degenerate. Therefore, an average over undefined values stays undefined. Micro-averaged measures, on the other hand, are fine, as they average the contingency tables for all the labels and then compute the statistics from the final contingency table, which eliminates the probability of the contingency table being degenerate.

A. Analyzing the Results

Let us start with the bipartition-based classifiers whose results are posted in Table 1. For all the first four example-based measures, CLR seems to be the best individual classifier, which might lead us to think that the ensemble techniques will fair worse as no measure would make us prefer any other method. The micro-averaged measures however reveal that some methods might be actually advantageous in some situations (see RAkEL's micro-averaged precision, which is higher than that of CLR). This paints a different picture than [4] where CLR was not the best performer and if it excelled in something, it was precision. This goes to show that different classifiers end up being more or less useful given the data they are used on.

When we consider the ensemble techniques, performance tends to increase in some measures and decrease in others. The majority vote technique ends up being better in Hamming Loss and Subset Accuracy, but loses to CLR in Accuracy and Micro-averaged F_1. This leads us to believe that bipartition-based ensemble techniques do not offer a significant improvement in general performance. However, one-sided measures like precision and recall can be greatly improved by using the intersection and union techniques which might be handy for specific applications.

Let us now turn to the results yielded by the score-based classifiers on display in Table 2. The individual classifiers are clearly dominated by CLR which offers the best performance for all the evaluation metrics, confirming its appropriateness for the problem at hand. In face of this one-sided result, we might not expect the ensemble methods to provide much of an improvement. However, in all of the metrics but IE, the mean and top_3 ensemble techniques offer better performance than CLR alone. This corroborates the results seen in [4], where the mean and top_3 techniques consistently beat the individual classifiers as well. Similarly to [4], top_3 seems to be the better of the two techniques.

VI. Conclusion

We have seen that the ensemble techniques presented in

[4] have universal applications and can be easily used for text classification. We have seen that the top_3 and mean ensemble techniques are the best performers as in Sanden's and Zhang's research. In our situation, one of the preexisting classifiers dominated the other ones in performance, yet still the ensemble techniques benefited from including all of them. Finally, we have also discovered that CLR seems to be a very useful multi-label learner for text classification with a small amount of labels.

This work could be continued by examining more sophisticated ways of integrating the individual classifiers into an ensemble classifier. We might also try adding different multi-label learners to the mix or try creating ensemble classifiers using only some learners which perform exceedingly well. Another direction might be to try and apply ensemble techniques to another problem or field.

TABLE I: EXPERIMENT RESULTS FOR THE BIPARTITION-BASED ENSEMBLE TECHNIQUE

	HL	**SA**	**Recall**	**Accu.**	**MicroP**	**MicroR**	**MicroF$_1$**
RAkEL	0.0702	0.2310	0.5976	0.4967	0.6811	0.5578	0.6127
CLR	0.0699	0.2349	0.6841	**0.5322**	0.6503	0.6475	**0.6485**
ML-kNN	0.0762	0.1697	0.5041	0.4335	0.6666	0.4720	0.5518
HOMER	0.0799	0.2115	0.6604	0.5029	0.5958	0.6218	0.6081
IBLR	0.0572	0.1772	0.5209	0.4429	0.6679	0.4890	0.5642
Intersection	0.0772	0.1433	0.3173	0.3078	**0.8354**	0.2823	0.4214
Union	0.0929	0.1513	**0.8474**	0.5202	0.5221	**0.8194**	0.6374
Majority vote	**0.0639**	**0.2559**	0.6153	0.5300	0.7300	0.5709	0.6402

TABLE II: EXPERIMENT RESULTS FOR THE BIPARTITION-BASED ENSEMBLE TECHNIQUES.

	AP	CO	OE	IE	ESS	RL	MicroAUC
RAkEL	0.7385	5.9043	0.2469	0.5898	5.9793	0.1263	0.8673
CLR	0.8023	2.8701	0.2349	**0.5067**	2.2828	0.0507	0.9399
ML-kNN	0.7204	4.2654	0.3126	0.6371	4.0329	0.0930	0.9048
HOMER	0.6276	8.8283	0.3579	0.6805	10.7272	0.2261	0.7875
ILBR	0.7317	4.0045	0.2972	0.6222	3.7347	0.0855	0.9098
Minimum	0.6344	9.3977	0.2747	0.6765	11.5653	0.2450	0.7716
Maximum	0.7535	2.9771	0.3484	0.5734	2.6181	0.0585	0.9328
Mean	0.8039	**2.8153**	0.2314	0.5102	2.2100	0.0469	**0.9477**
Top_3	**0.8082**	2.8228	**0.2140**	0.5082	**2.2085**	**0.0495**	0.9475

REFERENCES

[1] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123-140, 1996.

[2] W. Cheng and E. Hullermeier, "Combining instance-based learning and logistic regression for multilabel classification," *Machine Learning*, pp. 211-225, 2009.

[3] M. Kubat, K. Sarinnapakorn, and S. Dendamrongvit, "Induction in multi-label text classification domains," *Advances in Machine Learning II,* pp. 225-244, 2010.

[4] C. Sanden and J. Zhang, "Enhancing multi-label music genre classification through ensemble techniques," In *Proc of the 24th international ACM SIGIR conference on Research and development in Information*, pp. 705-714. ACM, 2011.

[5] C. Shi, X. Kong, P. S. Yu, and B. Wang, *Multi-label ensemble learning*, pp. 223-239, 2011.

[6] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining*, vol. 3, no. 3, pp. 1-13, 2007.

[7] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Effective and efficient multilabel classification in domains with large number of labels," In *Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08)*, pp. 30-44, 2008.

[8] G. Tsoumakas, I. Katakis, and I.P. Vlahavas, "Mining multi-label data," *In Data Mining and Knowledge Discovery Handbook*, pp. 667-685. 2010.

[9] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, and I. Vlahavas, "Mulan: A java library for multi-label learning," *Journal of Machine Learning Research*, pp. 2411-2414, 2011.

[10] G. Tsoumakas and I. Vlahavas, "Random k-labelsets: An ensemble method for multilabel classification," *Lecture Notes in Artificial Intelligence*, pp. 406-417, 2007.

[11] G. Tsoumakas, M. Zhang, and Z-H. Zhou, *Learning from multi-label data*, 2009.