Triple-Random Ensemble Multi-label Classification for Image to Text Translation

Gulisong Nasierding[1,3], Grigorios Tsoumakas[2] and Abbas Z. Kouzani[3]

[1]Department of Computer Science, Xinjiang Normal University
No. 19 Xin Yi Rd., Urumqi, P. R China 830054
[2]Department of Informatics, Aristotle University of Thessaloniki
54124 Thessaloniki, Greece
[3]School of Engineering, Deakin University, Geelong, VIC 3217, Australia

gulnas9@gmail.com, greg@csd.auth.gr, kouzani@deakin.edu.au

Abstract. This paper presents a triple-random ensemble learning method for multi-label classification problems, especially aimed at application to image to text translation and automatic image annotation. The proposed randomized learning method integrates the concepts of random subspace, bagging and random k-label sets ensemble learning methods to form an approach to classification of multi-label data. It applies the random subspace method to feature space, label space as well as instance space at the same time. The devised subset selection procedure is executed iteratively. Each multi-label classifier is trained using the randomly selected subsets. At the end of the iterations, the ensemble MLC classifiers are constructed. The proposed method is implemented and its performance is evaluated. The experimental results demonstrate that the proposed method outperforms the examined counterparts in most occasions when tested on six multi-label datasets from different domains. It is shown that the developed method possesses a general usability in dealing with various multi-label classification problems. Therefore, the triple random ensemble learning method is recommended for application to image to text translation system, which is based on the positive outcome of predictive performance of TREMLC on scene image dataset.

Index Terms: Triple-random ensemble, multi-label classification, subspace method, RAkEL, bagging,

1 Introduction

Image to text translation (ITT) is the process of translating a given un-labelled image into a set of semantic concepts or keywords. Automated image annotation can be

considered as a category of image to text translation where the task is to assign a set of semantic concepts to un-labelled image [Duy02, Bar03]. Besides, automated image region annotation is another option for realization of image to text translation [Bar03, Yua07]. The Automated image annotations can be grouped into two categories: statistical model based and classification based approaches [Liu07, Wan08]. The statistical model based approaches give rise to a problem called semantic gap. In order to avoid such problem, classification based approaches have appeared. They can be further divided into single-label and multi-label classification [Tsa08, Wan08], which can be seen in figure 1. However, single-label classification ignores the correlation among semantic concepts associated with the image. Therefore, multi-label classification is emerging as a robust candidate for image annotation problems [Kan06, Wan08].

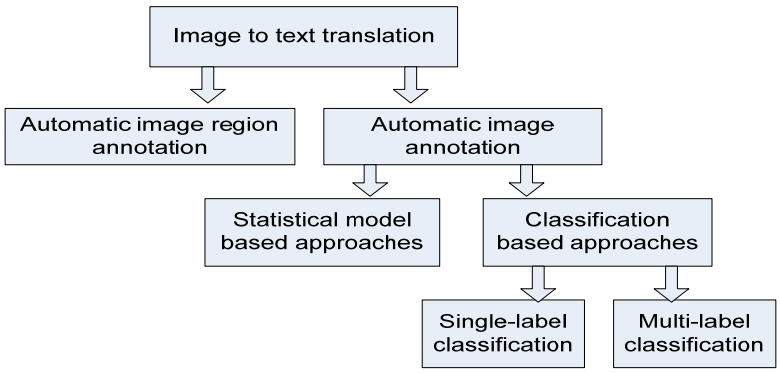


Fig. 1: Categorisation of image to text translation.

Although, a number of multi-label classification methods have been developed for multi-label image classification and automatic image and video annotations [Bou04, Zhou06, Li04, Joh05, Kan06, Wan08, Nas09, Qi07, Dim09], their performances are yet to match the basic requirement of image annotation systems. Therefore, a novel triple random ensemble multi-label classification (TREMLC) method is proposed in this paper, which can be applied to image to text translation and automatic image annotation.

The remaining content of this technical report is organized as follow. Section 2 introduces the related work to this paper include overview of the popular MLC algorithms and the baseline ensemble learning methods. Section 3 describes the proposed TREMLC method. Section 4 provides the experimental setup, including the datasets used for evaluation, the evaluation methods, as well as the experimental settings. Section 5 presents the experimental results and the associated discussions. Additionally, prospective application of the proposed TREMLC method is pointed out in section 6, which is based on the positive outcome of predictive performance of TREMLC on several multi-label datasets. Finally, conclusion and the future direction of this work are given in Section 7.

2 Related Work

This section introduces the concepts of multi-label classification and overviews of the related popular multi-label classification methods. The multi-label classification algorithms in these methods are used as the comparative counterparts for the proposed TREMLC algorithm. Besides, the baseline ensemble learning methods are also briefly introduced.

2.1 Multi-label Learning

Traditional *single-label* classification is concerned with learning from a set of examples that are associated with a single label *l* from a set of disjoint labels *L*, |*L*| > 1. In *multi-label* classification (MLC), on the other hand, the examples are associated with a set of labels $Y \subseteq L$ [1, 10, 11, 12]. In order to describe multi-label problems, using $L = \{l_j : j = 1...M\}$ to denote the finite set of labels in a multi-label learning task and $D = \{(\vec{x}_i, Y_i), i = 1,..., N\}$ to represent a set of multi-label training examples, where $\vec{x}_i$ denotes a feature vector, and $Y_i \subseteq L$ denotes a set of labels of the *i*-th example in *D*. Multi-label classification problems can be found in various domains, examples of these problems include text document classification [13 - 16, 26 - 27], bioinformatics data classification [17-19, 11], music categorization [20-21], scene image classifications [1-2, 22], image and video annotation [3- 9]. Therefore, a variety of MLC approaches have been explored to tackle these problems. Multi-label classification methods can be mainly categorized into two groups: (i) *problem transformation* methods and (ii) *algorithm adaptation* methods [12, 19]. The former

includes methods that are algorithm independent. They transform the multi-label classification task into one or more single-label classification, regression or ranking tasks. The latter one includes methods that extend specific learning algorithms to adapt multi-label learning by handling multi-label data directly [19]. What type of method should be developed for a particular multi-label task depends on the characteristics of the multi-label problem.

As an algorithm adaptation method, the multi-label k-nearest neighbour (ML-KNN) method [11] extends the popular k Nearest Neighbours (kNN) lazy learning algorithm using a Bayesian approach [23]. It uses the maximum a posterioris principle in order to determine the label set of the test instance, based on prior and posterior probabilities for the frequency of each label within the k nearest neighbours.

Binary relevance (BR) method [18] is a popular PT method that learns *M* binary classifiers, one for each different label in L. For the classification of a new instance, BR outputs the union of the labels that are positively predicted by the M classifiers.

Label Power set (LP) is an effective problem transformation method [1, 19]. It considers each unique set of labels that exists in a multi-label training set as one of the classes of a new single-label classification task. Given a new instance, the single-label classifier of LP outputs the most probable class, i.e. a set of labels. Due to the large number of classes produced by the label power set method, many of the classes correspond to a few examples causing difficulties for the learning process.

The random k-label sets (RAkEL) method [24, 16] builds an ensemble of LP classifiers. Each LP classifier is trained using a different small random subset of the set of labels. In such a way, RAkEL is able to take label correlations into account, while avoiding LP's problems. A ranking of the labels is produced and threshold is then used to produce a classification.

The calibrated label ranking (CLR) method [22] learns a mapping from instances to rankings over a finite number of predefined set of class labels. The main idea of the approach is to separate the relevant labels from the irrelevant labels in each example by introducing artificial calibration labels.

In the Hierarchy Of Multi-label ClassifiERs (HOMER) method [25], a tree-shaped hierarchy of simple multi-label classifier is constructed and each one of the classifiers handles a smaller set of labels compared with the entire large label set L. The better

balanced example distribution and divide-and-conquer strategies are adopted for designing the HOMER. Different approaches for distribution of labels into subsets are presented in the literature for HOMER.

Furthermore, in addition to the aforementioned algorithm adaptation and problem transformation based methods, various multi-label classification approaches are formed by combining and integrating the methods within these two groups [19], such as a probabilistic generative model [26], Adaboost.MH and Adaboost.MR [10], and ML-KNN [11]. A number of baseline algorithms including decision trees and boosting, probabilistic methods, neural networks, support vector machines, and lazy and associative methods are employed for development of multi-label classification and label ranking methods [12, 19]. Besides, the feature dimensionality reduction and the feature selection methods are also explored for multi-label classification [27-29]. However, the development of robust MLC algorithms is still in demand for improving the classification performance.

2.2 Ensemble Learning

Bagging [30], boosting [10] and Random Forests [31] are conventional popular ensemble classification methods that are initially designed to handle single-label classification problems. The results of these ensemble learning methods [30 - 37] are appealing compared to single classifiers. Specifically, the bagging method uses random sub-sampling to train instances. Also, the random subspace method [31] applies the base-level algorithm on randomly selected subset of features at each step of tree construction and selects the best among these to build ensemble classifiers. Breiman [32] combined the concepts of bagging and random subspaces to form random forests, which construct better ensemble classifiers. Attribute bagging method was proposed for improving accuracy of classifier ensembles by using random feature subsets [34]. Bootstrap-inspired techniques [35] and Random feature subset selection for ensemble based classification [36] are also become popular. More recently, Panov et al [37] developed a variant of random forests in order to achieve a similar effect of random forests, which improved ensemble classification performance. However, these methods only targeted single-label classification.

Along with multi-label classification problems have increasingly drawn researchers' attention, development of various ensemble learning methods become prevailed [2, 4, 10, 14, 15, 16, 24, 28]. The results demonstrate that the ensemble strategies can also

bring robustness to multi-label classification performances. For example, a model-shared random subspace bagging method automatically finds shares and combines a number of base models through multiple labels [28]. Johnson et al[Joh05] learned not only relationship between image and words, but also the relationship between image regions and words through multi-class boosting and multi-label weak learners (MLBoost). Furthermore, multi-instance, a multi-label learning framework was proposed by Zhou et al. [Zho06], in which, MIMLBOOST and MIMLSVM learning algorithms were formed. However, these methods ignore sub-sampling the label set in the label space, so that the label correlation was not taken into account. RAndom *k*-label sets Ensemble Learning (RA*k*EL) method [16, 24] was merged by constructing an ensemble of *m* LP classifiers iteratively. RAKEL considered the label correlations in the label space which enables avoiding the learning difficulty where a large number of classes are associated with a few examples. Besides, computational complexity of RAKEL is also reduced comparing with its base multi-label classifier LP. These ensemble methods provide a solid foundation and inspiration for emerging the proposed triple-random ensemble multi-label classification method in this paper.

3 A Triple-Random Ensemble Multi-label Classification Method

This section describes the proposed Triple-Random Ensemble Multi-Label Classification (TREMLC) method. TREMLC is a combination of the random subspace method (RSM) [31, 34], bagging [30, 35] and random k-labelset ensemble learning (RAkEL) [16, 24], where RSM applies the random subspace strategy to feature space, and RAKEL applies the strategic random subspace scheme to label space, whereas bagging [30] brings the random sub-sampling method to instance space. Furthermore, since random forest [32] and its variants [37] employ the random subset selection scheme in both feature space and instance space, TREMLC can be considered combining and extending the ideas of the RSM, bagging and RAkEL, or integration the ideas of Random Forests and RAKEL. That is, TREMLC applies RSM to feature space, label space, as well as instance space. The triple random algorithm can be described in the form of a pseudo code as shown in Figure 1.

- Input: Set of training data *D* of size *N*, set of attributes A of size F, set of labels *L* with size *M*, size of feature subset S_f < F, size of label subset S_l < M, bag percentage *b*, number of models *m*
- Output: An ensemble of LP classifiers h_i, i=1…m.
- FS ← {}, LS ← {}

```
for i =1 to m
  {
   Di ← random selection of N*b% instances from D;
   do
     {
      Fi ← random selection of Sf features from A
     } while (Fi not in FS);
   FS← FS union {F}
    do
     {
      Gi ← random selection of Sl labels from L
     } while (Gi not in LS);
   LS ← LS union {G}
   Ri ← projection of Di to the attribute and label dimensions, F and G.
Train an LP classifier hi based on Ri;
  }
```

(a) TREMLC Training Process.

- **Input**: Set of labels L with size M, number of models m, LP classifiers h_i, sets of attributes F_i and labels G_i, new instance $\vec{X}$
- **Output**: Multi-label classification vector Result

```
for(int i = 1 to m) {
   x' ← projection of x in dimensions of Fi and Gi;
   p = hi(x');
 for(int j = 1 to L)
   {
    SumVote j = SumVote j + Vote(p);
    LengthVote j++;
   }
 }
  for(int j = 1 to M)
 {
   Confj ← SumVotej / LengthVotej;
    if(Confj > threshold)
     {
      Resultj ←1;
     }
 else Resultj ← 0;
   }
```

(b) TREMLC Testing Process.

Fig. 1: Pseudo code for the proposed TREMLC algorithm.

A set of feature subsets, a set of label subsets, and a set of instance sets are selected randomly and iteratively for TREMLC, and the random subset selections are without replacement. By end of iteration, a set of ensemble multi-label classifiers are constructed based on the randomly selected subsets. Note that, LP [16, 19] is used as multi-label Lerner base and Decision Tree [38] is used as base classifier for LP in this problem transformation based TREMLC algorithm.

4 Experimental Setup

This section provides details of the experimental setup. First, it describes the datasets used for the evaluation of the proposed algorithm and the associated counterparts. Next, the evaluation criteria used for measuring the performance of the examined MLC algorithms are presented. Finally, the experimental setting is explained.

4.1 Datasets

The proposed TREMLC algorithm and the examined counterparts are tested on six multi-label datasets in this paper, including *scene* image dataset [1], jmlr2003 image dataset *Corel16k001* [39], multimedia *mediamill* dataset [40] biological *yeast* dataset [17], music categorical *emotions* dataset [20], diagnostic *medical* report dataset [15].

The *scene* image dataset contains 2407 images annotated with up to 6 concepts such as beach, mountain and field. Each image is described with 294 visual numeric features and these features are represented with spatial colour moments in Luv colour space. Each instance in the train and test datasets is labelled with possible 6 object classes as mentioned above [1, 22].

The *Corel16k001* is produced from the first (001) subset of the data *jmlr2003* [39], which is derived from a popular benchmark dataset *eccv2002* [41] by eliminating less frequently appeared keyword classes. That is, 374 keyword classes in *eccv2002* were reduced to 153 in *jmlr2003-001*. Before this stage, images are segmented using normalized cuts, then useful 46 numeric features are extracted from each region/blob and vector quantized. Next, the blobs are clustered into 500 blob clusters. The *Corel16k001* data is created based on 13766 images, and 500 blob clusters are used as nominal features of the dataset.

The *mediamill* dataset is based on the mediamill challenge data set [40]. It contains pre-computed low-level multimedia features from 85 hours of international broadcast

news video of the TRECVID 2005/2006. This dataset contains Arabic, Chinese, and US news broadcasts that were recorded during November 2004, and the contents are annotated with multiple labels. The component used for the evaluation of MLC algorithms are based on still image data from the video shot key frames extracted. The annotation of the *mediamill* data was extended to current 101 concepts from a manual annotation of 39 labels by the TRECVID 2005.

The *yeast* dataset can be used for biological gene function classification evaluation. This dataset contains 2417 gene examples and each of which is related up to a set of 14 functional gene classes from the comprehensive Yeast Genome Database of the Munich Information Center for protein Sequences. Each gene is expressed with 103 numeric features [17, 18, 11].

The *emotions* dataset can be used for evaluating the predictive power of several audio features in a new multi-label feature selection method [20, 21]. The emotion dataset contains a set of 593 songs with 6 clusters of music emotions, which is constructed based on the Tellegen-Watson-Clark model.

The *medical* dataset was constructed from the available data in Computational Medicine Center's 2007 Medical Natural Language Processing Challenge [15]. This dataset contains 978 clinical free text reports and each diagnostic report is related to one or more disease code from the 45 classes [15, 16].

These datasets are widely used as benchmark datasets for evaluating the MLC algorithms [2, 7, 9, 11, 12, 15-16, 19-22, 24-25, 42-43]. Table 1 shows general characteristics of these datasets, including name, number of examples, number of features and number of labels for each dataset, types of attributes, and the domains that these datasets are belonging to. Note that the 'num' in the table 1 refers to numerical attribute dataset, and 'nom' refers to nominal attribute dataset.

Table1. Characteristics of the datasets used.

Datasets Names	Domain	Instances	Attributes	Num. labels
scene	image	2407	294 num.	6
Corel16k001	image	13766	500 nom.	153
mediamill	video	43907	120 num.	101
yeast	biology	2417	103 num	14
emotions	music	593	72 num	6
medical	text	978	1449 nom	45

4.2 **MLC Evaluation Methodology**

The evaluation measures for multi-label classification are different from those of single-label classification [11]. These evaluation methods can be divided into example based measures, label-based measures, and ranking based measures [10-12, 22, 42]. Several MLC evaluation measures from the three types aforementioned are adopted in this work as follows.

Example-based Evaluation Measures bipartitions based on the average differences of the actual and predicted sets of labels over all examples of the evaluation dataset. The hamming-loss refers to average binary classification error. Suppose given the multi-label evaluation dataset *D* contains multi-label examples (x_i, Y_i), i=1, 2,…, N, $Y_i \subseteq L$ is a set of true labels and L= {l_j: j=1…M} is the set of all labels, and x_i is a new instance. Predicted set of labels for the instance x_i by using a MLC method set to be Z_i , and ranking based prediction by using label ranking method for a label l is assumed to be $r_i(l)$. Hence, Hamming-Loss can be calculated as:

$$\text{Hamming-Loss} = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \Delta Z_i|}{M} \tag{1}$$

where $Z_i = h(x_i)$ is a set of labels that predicted by a multi-label classifier *h* for an example x_i. [12]. The smaller the value of the Hamming-loss is the indicative of better performance of the classification.

Label-Based Evaluation Measures: Label based F1-measure refers to the harmonic mean between precision and recall, where the recall refers to the percentage of relevant labels that are predicted and precision refers to the percentage of predicted labels that are relevant. F1-measure is widely used for single-label classification evaluation, which also is applicable for evaluating multi-label classification by using two averaging methods, i.e. Micro and Macro averaging. The F1-measure and micro averaging can be calculated as:

$$\text{F-measure} = \frac{1}{N}\sum_{i=1}^{N}\frac{2|Y_i \cap Z_i|}{|Z_i| + |Y_i|} \tag{2}$$

$$M_{micro} = M\left(\sum_{l=1}^{M} tp_l, \sum_{l=1}^{M} fp_l, \sum_{l=1}^{|M} tn_l, \sum_{l=1}^{M} fn_l\right) \tag{3}$$

where *the* tp_l, fp_l.tn_l, fn_l denote a number of true positive, false positive, true negative and false negative for *l* labels after binary evaluation[12]. The value of the micro F1-measure is the greater the better performance of the classification.

Ranking-based Evaluation Measure: Label based ranking predict the rank of a label. The most relevant label is ranked to receive highest score, while the most irrelevant one is ranked to receive lowest score. The ranking based prediction by using label ranking method for a label *l* is assumed to be $r_i(l)$. There are four ranking-based metrics can be used to measure the label ranking, i.e. one-error, coverage, ranking-loss and average precision [12].

One-error evaluates how many times the top-ranked label is not in the set of proper labels of the instance. One-error is equal to normal classification error for single-label classification problems.

$$One-error = \frac{1}{N}\sum_{i=1}^{N}\delta(\arg\min_{l\in L} r_i(l)) \tag{4}$$

where $\delta_{(l)} = \begin{cases} 1 & if \quad l \notin Y_i \\ 0 & otherwise \end{cases}$

The smaller the value of the one-error is indicative of better performances of the classification.

Coverage evaluates how far we need to cover all the proper labels of the instance on average.

$$Coverage = \frac{1}{N}\sum_{i=1}^{N}\max_{l\in Y_i} r_i(l) - 1 \tag{5}$$

The smaller the value of the coverage, the better performance of the classification is indicated.

Ranking-loss evaluates the average fraction of label pairs that are reversely ordered for the instance.

$$Rank-loss = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|Y_i||\overline{Y_i}|}|\{l_a,l_b): r_i(l_a) > r_i(l_b),(y_a,y_b)\in Y_i\times\overline{Y_i}\} \tag{6}$$

where $\overline{Y}_i$ denotes the complementary set of Y_i with respect to L. The smaller values of the Ranking-loss the better performances of the classification.

Average precision evaluates the average fraction of labels ranked above a particular label $l \in Y_i$, which actually is in Y_i:

$$avg.\Pr ec. = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|Y_i|}\sum_{l\in Y_i}\frac{|l' \in Y_i : r_i(l') \le r_i(l)\}|}{r_i(l)} \tag{7}$$

The larger value of the average precision the better performances of the classification.

4.3 Experimental Setting

This section provides experimental setting for evaluation of the proposed TREMLC algorithm and the selected counterparts. In order to empirical study of the proposed TREMLC, some popular MLC algorithms, as indicated in section2.1, are chosen from the open source MULAN library [19], which is built on top of the open source Weka library [44]. Default parameters are set for the examined MLC algorithms as indicated in the literature. Such as, ML-kNN is run with 10 nearest neighbours and a smoothing factor equal to 1. RAKEL [16, 24] uses Label Powerset [1, 19] as multi-label learner base, and set the size of label subset k= 3, number of models (number of iterations) to be m= 2k, and threshold is set to be 0.5 for all the algorithm evaluations. HOMER distributes the labels evenly and randomly into 3 subsets, and CLR is chosen to be multi-label learner base for the HOMER. Furthermore, Decision tree C4.5 [38] is used as base classifier for all the selected problem transformation based MLC methods in this paper including the proposed TREMLC. ML-KNN is the only algorithm adaptation MLC method among the examined existing methods in the current experimental setting.

The same as RAKEL[24], LP [1, 19] is used as multi-label base learner in TREMLC. The rest of default parameters for TREMLC are set as follow: each subset covers 70% of the original training set in the feature space and instance space, while the number of models is set to be twice size of the label set size of a multi-label dataset, and label subset size is set to be 3. Additionally, the minimum size of models is set to be 200 if m=2L < 200.

Multi-label classification evaluation measures including the example-based Hamming-loss, the label-based micro F1-measure and ranking-based all measures are employed to measure the predictive performances of the examined MLC algorithms. Additionally, the records of the evaluation time for each examined algorithm are also presented in order to estimate the computational complexity of the algorithms. The experiments have been performed on the Victorian Partnership for Advanced Computing machines. The predictive performances are evaluated using the 10-fold cross-validation.

5 Results and Discussion

This section presents experimental evaluation results of predictive performance of the examined MLC algorithms and accompanied discussion to the results.

5.1 Predictive Performance

Predictive performances of TREMLC vs. existing MLC counterparts are given in the following tables. Although the predictive performances of the examined MLC algorithms are obtained in various MLC evaluation measures, the results are presented in this paper only in some popular measures, i.e. example based Hamming-loss, label-based micro F1-measure and ranking-based all the measures.

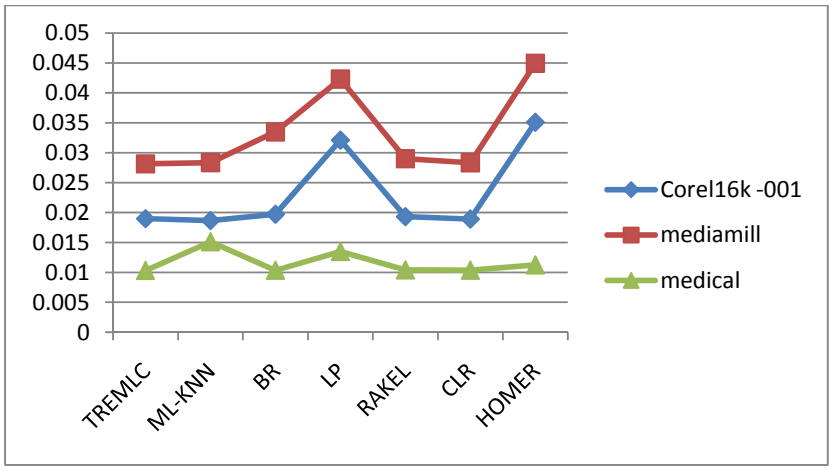
As can be seen from Table 2 and Figure 1, the TREMLC performed the best in terms of Hamming-loss when tested on almost all the selected multi-label datasets, i.e. *scene, mediamill, yeast, emotions and medical,* and performed the second best on Corel16k001. In the second high performance level, ML-KNN presented good results on *scene ,mediamill* and *yeast*, while RAEEL performed nicely on emotions and medical, then a minor inferior to ML-KNN on *scene*, *Corel16k001, mediamill* and *yeast*. Furthermore, ML-KNN performed the best on *Corel16k001,* and CLR also achieved reasonably good results on all the selected datasets. Note that, the performances of examined MLC algorithms are achieved in different level on different datasets under Hamming-loss measure, thus, the presentation in figures are divided into two, i.e. Figure 1(a) presents Hamming-loss measures on *scene, yeast* and *emotions* datasets; and Figure 1(b) presents Hamming-loss measures on *Corel16k001, mediamill* and *medical* datasets. Overall, TREMLC achieved the top performance on five out of six evaluation datasets under Hamming-loss measures.

Table 2: Predictive performances of MLC algorithms measured with *Hamming-loss.*

MLC Algorithms	scene	Corel16k -001	mediamill	yeast	emotions	medical	Top-scores
TREMLC	**0.082821**	0.018989	**0.02814**	**0.18783**	**0.180758**	**0.010319**	5
ML-KNN	0.085309	**0.018669**	0.02834	0.194151	0.26177	0.015112	1
BR	0.136762	0.019729	0.03349	0.245432	0.247401	0.010344	-
LP	0.143819	0.032102	0.042314	0.277901	0.27775	0.013476	-
RAKEL	0.098884	0.019327	0.029003	0.219515	0.217538	0.010411	-
CLR	0.138348	0.018921	0.028317	0.220227	0.242302	0.010364	-
HOMER	0.165357	0.035103	0.04496	0.286063	0.278315	0.011229	-

Note: The smaller value of Hamming-loss, the better performance of the MLC algorithms.

(a)



(b)

Fig.1: Predictive performances of MLC algorithms measured with *Hamming-loss.*

Table 3 and Figure 2 present that TREMLC is the best performing algorithm on *mediamill*, *yeast* and *emotions*, and achieved second high performance on *scene*, and minor difference to the top performers on *medical*. ML-KNN achieved best performance on scene and reached second high performance level on yeast, while BR showed the best on medical, and HOMER the highest performance on Corel16k001. Furthermore, RAKEL reached the second highest position on *emotions* and *medical*, and minor difference to the top performance on *scene*, *mediamill* and *yeast*. In the next level, CLR also performed reasonably well on almost all the selected datasets, except the Corel16k001. Overall, TREMLC achieved the top performance on three out of six evaluation datasets under micro F1-measure, while ML-KNN, BR and HOMER achieved the best performance individually on one dataset only.

Table 3: Predictive performance of MLC algorithms measured with *micro F1-measure.*

MLC Algorithms	scene	Corel16k -001	mediamill	yeast	emotions	medical	Top-scores
TREMLC	0.730163	0.077886	**0.62180**	**0.654469**	**0.680128**	0.803205	3
ML-KNN	**0.737853**	0.012653	0.59346	0.639716	0.468944	0.678398	1
BR	0.619391	0.117836	0.56484	0.585697	0.601974	**0.809087**	1
LP	0.597837	0.123865	0.50677	0.54057	0.548976	0.752437	-
RAKEL	0.697095	0.105382	0.610112	0.620809	0.638645	0.808453	-
CLR	0.627572	0.08579	0.596357	0.615765	0.627627	0.807684	-
HOMER	0.574643	**0.200592**	0.533611	0.589529	0.601781	0.798167	1

Note: The greater value of micro F1-measure, the better performance of the MLC algorithms.

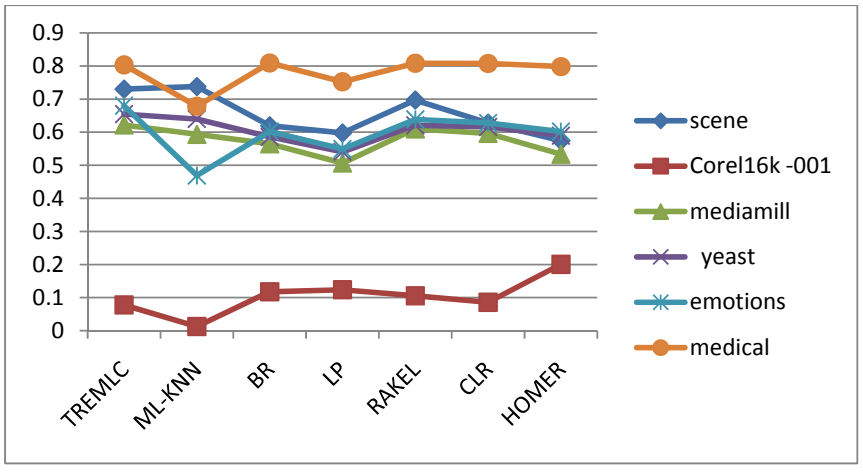


Fig.2 Predictive performance of MLC algorithms measured with *micro F1-measure.*

Table 4 and Figure 3 indicated that TREMLC achieves highest performance on majority the selected datasets in terms of one-error. That is, TREMLC presented the best performance on *scene*, *yeast*, *emotions* and *medical*, while CLR reached to the top performance level on *Corel16k001* and *mediamill*, and climbed to the second top on *medical*. In the second highest performance level, ML-KNN achieved top results on *scene*, *mediamill* and *yeast*, while RAkEL was approaching to the second top on *mediamill* and *emotions*. Besides, BR achieved the second best on *Corel16k001* dataset. Overall, TREMLC achieved the top performance on four out of six evaluation datasets under one-error measure, while CLR achieved the best performances on *Corel16k001* and *mediamill*, which are relatively larger datasets, especially in the respect of label set sizes.

Table 4: Predictive performance of MLC algorithms measured with *one-error*.

MLC Algorithms	scene	Corel16k -001	mediamill	yeast	emotions	medical	Top-scores
TREMLC	**0.204843**	0.743936	0.19045	**0.220118**	**0.248079**	**0.150368**	4
ML-KNN	0.224343	0.728823	0.15562	0.226722	0.379548	0.240311	-
BR	0.413821	0.703473	0.413988	0.399254	0.391328	0.191258	-
LP	0.39343	0.798707	0.298472	0.341367	0.43339	0.205544	-
RAKEL	0.267557	0.739868	0.169722	0.259825	0.300311	0.184126	-
CLR	0.302434	**0.659089**	**0.146195**	0.241629	0.315452	0.160593	2
HOMER	0.446658	0.76609	0.439815	0.283414	0.433531	0.216831	-

Note: The smaller value of one-error, the better performance of the MLC algorithms.

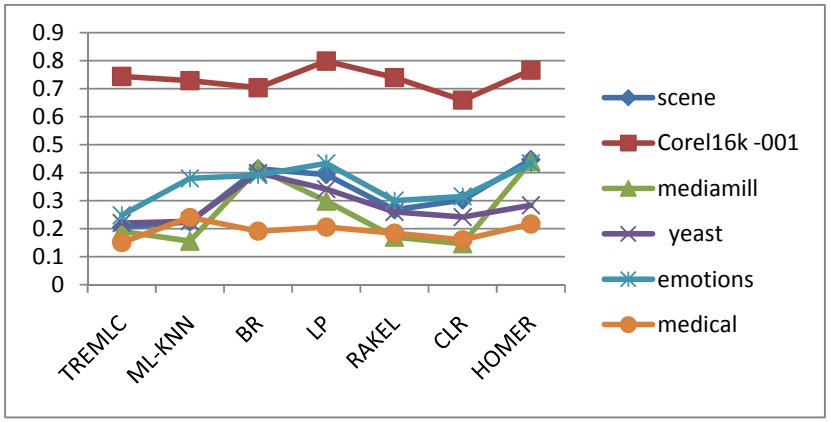


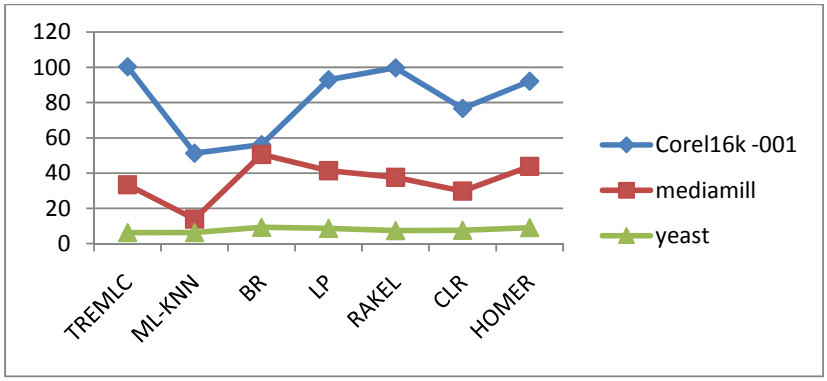
Fig.3 Predictive performance of MLC algorithms measured with *one-error*.

The Tables 5 - 6 showed that TREMLC has been the top performer on relatively smaller label set sized datasets *scene*, *yeast* and *emotions*, and it showed second best performance on *medical* dataset when measured with the coverage and ranking-loss; while ML-KNN showed excellence on the rest of datasets, i.e. *Corel16k001*, *mediamill* and *medical* , and it reached the second best on *scene* and *yeast* under these two measures. In the next level of performance, RAKEL performed nicely on *scene*, *yeast* and *emotions*, while CLR was approaching to this level on *scene*, *mediamill*, *yeast* and *medical*. Note that, BR achieved the second best on *Corel16k001* dataset. In overall ranking, TREMLC achieved the top performance on three out of six evaluation datasets under coverage and ranking-loss measures, while ML-KNN achieved the best performances on the rest of three datasets. Due to the predictive performances of the examined MLC algorithms on different datasets appeared in big gap under the coverage measure, therefore, these performances are presented in two separate figures, i.e. in Figure 4 (a) and (b). These figures support to the evaluation results of predictive performances in Table5. Besides, the predictive performances presented in Figure 5 supports to the evaluation results presented in Table 6.

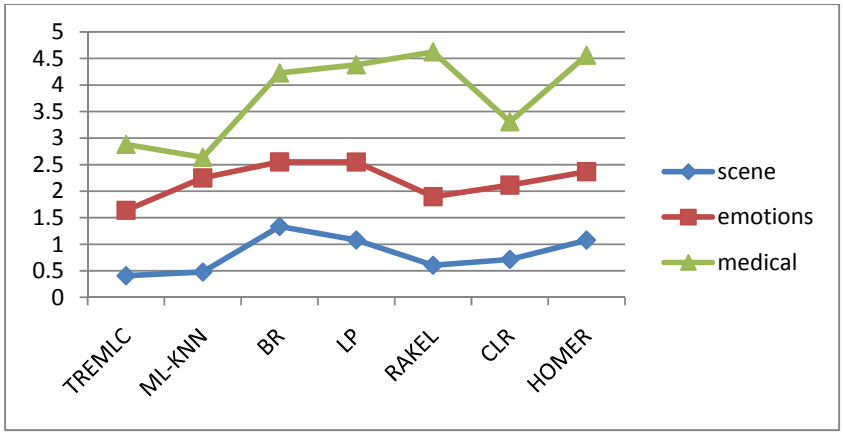
Table 5: Predictive performance of MLC algorithms measured with *coverage.*

MLC Algorithms	scene	Corel16k -001	mediamill	yeast	emotions	medical	Top-scores
TREMLC	**0.407569**	100.3743	33.4002	**6.241081**	**1.639718**	2.880875	3
ML-KNN	0.477343	**51.27858**	**13.9070**	6.263931	2.250424	**2.637839**	3
BR	1.334509	56.14361	50.63293	9.239836	2.550734	4.226667	-
LP	1.079805	92.99812	41.4132	8.654247	2.550367	4.379466	-
RAKEL	0.604065	99.74531	37.65036	7.378855	1.897401	4.6243	-
CLR	0.714613	76.69164	29.79459	7.540849	2.115028	3.306249	-
HOMER	1.078247	92.16098	43.82133	9.055086	2.368051	4.562066	-

Note: The smaller the value of coverage the better performance of MLC algorithms.



(a)



(b)

Fig. 4: Predictive performance of MLC algorithms measured with *coverage.*

Table 6: Predictive performance of MLC algorithms measured with *ranking-loss.*

MLC Algorithms	***scene***	**Corel16k -001**	**mediamill**	**yeast**	**emotions**	**medical**	**Top-Scores**
TREMLC	**0.064957**	0.395274	0.10542	**0.160904**	**0.139232**	0.046994	3
ML-KNN	0.077846	**0.172566**	**0.03888**	0.165541	0.255745	**0.040299**	3
BR	0.246473	0.188266	0.178647	0.309702	0.291476	0.074285	-
LP	0.197284	0.35749	0.158698	0.316854	0.312612	0.076099	-
RAKEL	0.103268	0.389452	0.116	0.211669	0.183373	0.080974	-
CLR	0.12385	0.26427	0.09551	0.210149	0.213045	0.054857	-
HOMER	0.196292	0.344173	0.159397	0.302147	0.273261	0.081113	-

Note: The smaller value of ranking-loss, the better performance of the MLC algorithms.

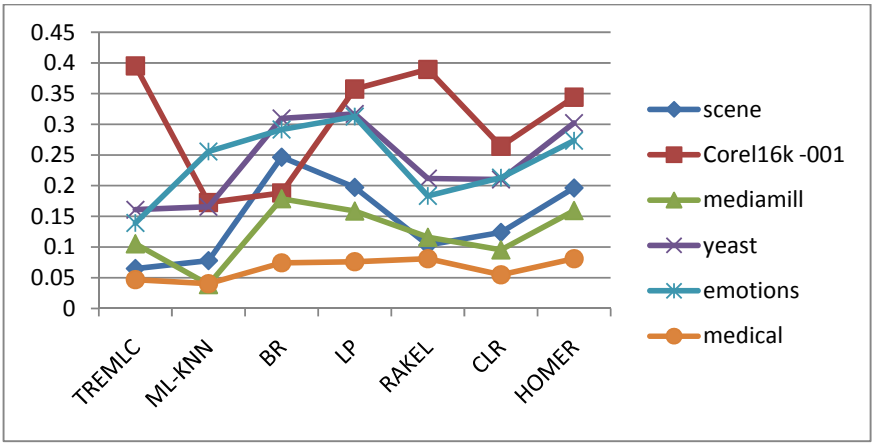


Fig.5 Predictive performance of MLC algorithms measured with *ranking-loss.*

Table 7 and figure 6 present that TREMLC algorithm is outstanding among the counterparts when measure the predictive performances on *scene*, *yeast*, *emotions* and *medical* datasets using average precision. Besides, TREMLC approached to the second best performance on large dataset *mediamill*. Furthermore, ML-KNN reached the best performance on *mediamill* and it approached to the second best level on *scene*, *Corel16k001* and *yeast*, while RAKELand CLR approached to the high performance on *scene*, *mediamill*, *yeast*, *emotions* and *medical* datasets. Note that, BR climbed to the best predictive performance level on the *mediamill*. Figure 6 also provided supportive evidence for this observation. Overall, TREMLC shows excellence on four out of six datasets, while ML-KNN shows the best performance on *mediamill* and BR achieved the best on Corel16k001.

Table 7. Predictive performance of MLC algorithms measured with *average precisions.*

MLC Algorithms	***scene***	**Corel16k -001**	**mediamill**	**yeast**	**emotions**	**medical**	**Top-Scores**
TREMLC	**0.880521**	0.1712	0.69915	**0.771701**	**0.820078**	**0.871313**	**4**
ML-KNN	0.865763	0.287985	**0.75502**	0.765582	0.7141	0.813356	1
BR	0.710852	**0.289205**	0.576282	0.621568	0.701352	0.834109	1
LP	0.739422	0.185362	0.57648	0.645407	0.683013	0.814071	-
RAKEL	0.835592	0.182094	0.691481	0.724137	0.783797	0.826389	-
CLR	0.809449	0.282241	0.699258	0.729328	0.759014	0.851976	-
HOMER	0.71679	0.201736	0.524566	0.64668	0.702491	0.801279	-

Note: The greater value of average precision, the better performance of the MLC algorithms.

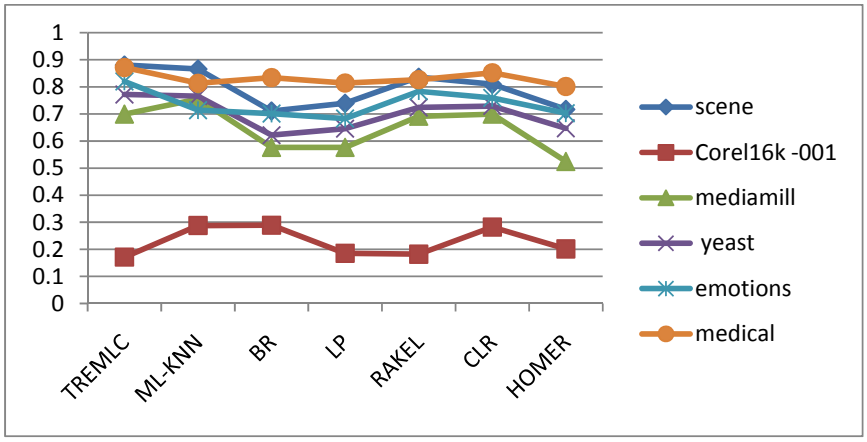


Fig. 6*:* Predictive performance of MLC algorithms measured with *average precisions.*

A peculiar phenomenon can be observed from Tables 2-7 and Figure 1-6 that overall predictive performance levels of all the examined MLC algorithms are quite low on Corel16k001 compare to the performances on the rest of datasets. This is due to the unique characteristics of the Corel16k001 dataset, which can be seen from Table 1. Besides, the *Corel16k001* dataset possesses the nominal data features, which are quite sparse in the features space; and the nominal-featured labels are also sparse in the label space. In order to improve the predictive performances of the examined MLC algorithms on this dataset, additional pre-processing and fine tuning this data is necessary, or specially designed algorithm may need to be explored.

5.2 General Conclusion on Predictive Performance

The predictive performances of examined MLC algorithms on six different datasets are summarized in Tables 8 under six evaluation measures. Table 8 shows that TREMLC achieved the best five out of six predictive performances on *scene* dataset, and gain the second best in sixth measure, micro F1-measure; at the same time, ML-KNN achieved the second best in five out of six evaluation measures and reached to top performance when measured with micro F1-measure on the *scene*. On the *Corel16k001* dataset, three top performances and one second top performance are measured for ML-KNN, while one top performance and three second-top performances are measured for BR. Additionally, CLR achieved one top and one second-top performance when measured with one-error and hamming-loss on the *Corel16k001*. Lastly, RAKEL is measured the a second best on Corel16k001.

Furthermore, TREMLC is the best when measured with Hamming-loss and micro F1-measure on large sized dataset *mediamill*, while the ML-KNN showed excellence when measured with almost all the ranking-based measures, except measured as second best on one-error. CLR also approached to the excellence on mediamill, i.e. it obtained one top performance when measured with one-error, and four second best performance when measured with Hamming-loss and three ranking-based measures. Nevertheless, TREMLC is measured to be outstanding on *emotions* and *yeast* datasets, and it approached to the top on *medical* dataset; while ML-KNN is measured to be top performances under coverage and ranking-loss measures on the *medical*, and it is measured to be the second best under all the selected evaluation measures on the *yeast*. Note that, RAKEL showed the second best performance on all the selected evaluation measures on the *emotions*, while CLR showed four second best

performance on the *medical*. BR is obtained one top and one second top performances on medical when measured with micro F1-measure and Hamming-loss.

Table 8: Ranking predictive performances of examined MLC algorithms on different datasets.

MLC Algo.		*scene*	Corel16k001	mediamill	emotions	Yeast	medical
TREMLC	Hm-loss	^	-	^	^	^	^
	Mic. F1-m	+	-	^	^	^	-
	One-error	^	-	-	^	^	^
	Coverage	^	-	-	^	^	+
	Rank-loss	^	-	-	^	^	+
	Ave. precision	^	-	-	^	^	^
Scoring		**5^, +**	**-**	**2^**	**6^**	**6^**	**3^, 2+**
ML-KNN	Hm-loss	+	^	-	-	+	-
	Mic. F1-m	^	-	-	-	+	-
	One-error	+	-	+	-	+	-
	Coverage	+	^	^	-	+	^
	Rank-loss	+	^	^	-	+	^
	Ave.preci.	+	+	^	-	+	-
Scoring		**^, 5+**	**3^, +**	**3^, +**	**-**	**6+**	**2^**
CLR	Hm-loss	-	+	+	-	-	-
	Mic. F1-m	-	-	-	-	-	+
	One-error	-	^	^	-	-	+
	Coverage	-	-	+	-	-	-
	Rank-loss	-	-	+	-	-	+
	Ave.preci.	-	-	+	-	-	+
Scoring		**-**	**^, +**	**^, 4+**	**-**	**-**	**4+**
BR	Hm-loss	-	-	-	-	-	+
	Mic. F1-m	-	-	-	-	-	^
	One-error	-	+	-	-	-	-
	Coverage	-	+	-	-	-	-
	Rank-loss	-	+	-	-	-	-
	Ave.preci.	-	^	-	-	-	-
Scoring		**-**	**^, 3+**	**-**	**-**	**-**	**^, +**
RAKEL	Hm-loss	-	-	-	+	-	-
	Mic. F1-m	-	-	+	+	-	+
	One-error	-	-	-	+	-	-
	Coverage	-	-	-	+	-	-
	Rank-loss	-	-	-	+	-	-
	Ave.preci.	-	+	-	+	-	-
Scoring		**-**	**+**	**+**	**6+**	**-**	**+**

Note: symbol '^' denotes the best predictive performance, and '+' denotes the second best performance.

Table 8 indicates that TREMLC and MLKNN are not only robust on smaller sized datasets with different type of attributes, i.e. nominal and numerical, but also effective on large sized datasets with both large label set size (e.g. mediamill) and large feature

set size (e.g. medical). Hence, these two can be considered as high performing MLC algorithms and have potential for applying to various multi-label classification problems. Note that, TREMLC showed more robustness compare to ML-KNN overall, which can be observed from Table 8, as well as from Tables 2-7 and Figures 1-6.

5.3 Evaluation Time of TREMLC vs. Counterparts

This section presents evaluation time of examined MLC algorithms. Table 9 shows that the ML-KNN is the most efficient algorithm among the counterparts when tested on all the selected datasets, and BR is second efficient one. The most time consuming MLC algorithms on larger sized dataset *mediamill* are LP and RAKEL, followed by is CLR and TREMLC, especially TREMLC is identified as time consuming algorithm on almost all the datasets. This is due to TREMLC constructs ensemble classifiers with randomly selected subsets iteratively, which is time consuming. The TREMLC achieved high performance in accuracy, but with cost of efficiency, which is considered as important research question for our next step. In the next level of time consuming MLC algorithms, RAkEL is accounted, which is also a randomized ensemble MLC algorithm; it takes time to build ensemble classifiers. Interestingly, LP showed to be efficient algorithm on *Corel16k001*, while it was measured as a most time consuming algorithm on *mediamill*. Again, the characteristics of the datasets play un-ignorable roles for the efficiencies of the MLC algorithms.

Table 9. Evaluation times of the Examined MLC algorithms.

MLC Algorithm	**scene**	**Corel16k001**	**mediamill**	**yeast**	**emotions**	**medical**
TREMLC	172.8003	1231.66	2020.85	117.0322	8.769833	51.2055
ML-KNN	**2.757667**	223.8587	**181.238**	**1.2015**	**0.049833**	**0.1185**
BR	3.281833	486.023	727.2203	3.330667	0.153833	3.496833
LP	2.487	**58.90133**	3207.094	5.336	0.140167	0.7685
RAKEL	16.16567	2371.159	3081.987	26.50683	0.770333	21.08367
CLR	5.075333	1576.314	2577.75	9.6385	0.285	6.285
HOMER	4.356	1088.286	533.3867	4.744833	0.225667	3.688833

Note: The smaller value of evaluation time, the more efficient of an MLC algorithm.

6 Applications

Empirical study of popular multi-label classification methods show that the proposed TREMLC algorithm outperforms examined counterparts when tested on several multi-label datasets from different domains, which can be observed from the tables and figures above. The initial goal for exploration of the TREMLC algorithm was to exploit and develop effective multi-label classification method for image to text translation and automatic image annotation [Nas08, Nas09, Nas10]. Based on experimental evaluation results of the examined MLC algorithms, TREMLC algorithm can be recommended for a number of multi-label classification problems, particularly, image to text translation and automatic image annotation tasks in hand. Since the predictive performance of TREMLC is presented nicely on *scene* image datasets, this can be observed from Tables 2- 8 and Figures 1-6.

Furthermore, one can apply TREMLC for other multi-label classification problems too, such as music categorization based on the predictive performance of TREMLC on *emotions* dataset; biological information categorization based on the predictive performance of TREMLC on *yeast* data; as well as diagnostic *medical* report classification. Additionally, TREMLC also can be suggested for multimedia video news classification based on the positive result of TREMLC on *mediamill*. Moreover, TREMLC also can be considered for image to text translation based on *Corel16k001* with the condition of further processing and transforming the *Corel16k001*dataset to be more suitable for the TREMLC algorithm; alternatively, further optimizing the TREMLC to adapt the multi-label problem that represented with current Corel16k001. To sum up, the proposed TREMLC algorithm possess the general applicability for differently represented multi-label classification problems, therefore, it can be applied for the translation component of image to text translation system [45, 46].

7 Conclusion

This paper presented a triple-random ensemble MLC method, and the proposed TREMLC algorithm is recommended for image to text translation and automatic image annotation. The TREMLC algorithm is formed based on the baseline ensemble learning algorithms random subspace, bagging and k-label set ensemble learning methods. Some popular evaluation measures for multi-label classification were

chosen from three major types, i.e. example-based, label-based and ranking-based, to present the experimental evaluation results of the examined methods. The empirical results show that TREMLC performs better than its examined counterparts when evaluated on a set of selected multi-label evaluation datasets from different domains. Therefore, TREMLC method can be suggested for applying to different representative multi-label classification problems thanks to its general applicability. Hence, TREMLC is particularly, recommended for applying to image to text translation and automatic image annotation task. However, TREMLC needs to be further improved especially from the execution-time efficiency standpoint in our future work.

References

[1] Boutell, M. R., Luo, V., Shen, X., Brown, C. M.: Learning Multi-label scene classification, Pattern Recognition, Vol. 37, 1757 – 1771 (2004)

[2] Zhou, Z. H., Zhang, M. L.: Multi-instance multi-label learning with application to scene classification. In: Scho"lkopf. B., Platt. J. C., Hoffman. T., eds.: NIPS, MIT Press, pp. 1609 – 1616 (2006)

[3] X. Li, L. Wang, Multi-label SVM Active learning for Image Classification, In IEEE 2004 International Conference on Image Processing (ICIP'04), pages 2207-2210.

[4] Johnson, M., Cipolla, R.: Improved Image Annotation and Labelling through Multi-label Boosting. In: Proceedings of BMVC '05 (2005)

[5] Kang, F., Jin, R., Sukthankar, R.: Correlated Label Propagation with Application to Multi-label Learning. In: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), pp. 291-294, 1719-1726 (2006)

[6] Wang, M., Zhou, X., Chua, T.-S.: Automatic Image Annotation via Local Multi-Label Classification. In: Proceedings of the international conference on Content-based image and video retrieval (CIVR'08), pp.17-26, Niagara Falls, Canada (2008)

[7] Nasierding, G., Tsoumakas, G., Kouzani, A. Z.: Clustering Based Multi-Label Classification for Image Annotation and Retrieval. In: 2009 IEEE International Conference on Systems, Man, and Cybernetics (SMC'2009), pp. 4627-4632. Texas, USA (*2009*)

[8] Qi, G., Hua, X., Rui, Y., Tang, J., Mei, T., and Zhang, H.: Correlative Multi-label Video Annotation. ACM SIGMM, pp.17 - 26 (2007)

[9] Dimou, A., Tsoumakas, G., Mezaris, V., Kompatsiaris, I., Vlahavas, I.: An Empirical Study of Multi-Label Learning Methods for Video Annotation. In: the 7th International Workshop on Content-Based Multimedia Indexing, IEEE, Chania Crete (2009)

[10] Schapire, R. E., Singer, Y.: BoosTexter: A Boosting-based System for Text Categorization, *Machine Learning*, 39(2/3):135-168 (2000)

[11] Zhang, M. L., Zhou, Z. H.: ML – KNN: A Lazy Learning Approach to Multi-Label Learning. Pattern Recognition 40(7):2038–2048 (2007)

[12] Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining Multi-label Data, Data Mining and Knowledge Discovery Handbook. Maimon, O., Rokach, L. (Ed.), Springer, 2nd edition (2010)

[13] Katakis, I., Tsoumakas, G., Vlahavas, I.: Multi-label text classification for automated tag suggestion. In: Proceedings of the ECML/PKDD 2008 Discovery Challenge, Antwerp, Belgium (2008)

[14] Esuli, A., Fagni, T., Sebastiani, F.: Boosting multi-label hierarchical text categorization. Information Retrieval, vol. 11, pp. 287–313 (2008)

[15] Read, J., Pfahringer, B., Holmes, G.: Multi-label classification using ensembles of pruned sets.In: 2008, ICDM '08. Eighth IEEE International Conference on Data Mining, pp. 995 – 1000 (2008)

[16] Tsoumakas, G., Katakis, I., Vlahavas, I.: Random k-Labelsets for Multi-Label Classification. IEEE Transactions on Knowledge Discovery and Data Engineering, (2010)

[17] Elisseeff, A., Weston, J.: A Kernel method for multi-labelled classification. Paper presented to Advances in Neural Information Processing Systems, No.14 (2002)

[18] Brinker, K., Hullermeier, E.: Case-based multi-label ranking. In: Proceedings of the 20th International Conference on Artificial Intelligence (IJCAI '07), pp.702–707. Hyderabad, India (2007)

[19] Tsoumakas, G., Katakis, I.: Multi–label Classification: An Overview, *International Journal of Data Warehousing and Mining*, 3(3): pp. 1-13 (2007)

[20] Trohidis, K., Tsoumakas, G., Kalliris, G., Vlahavas, I.: Multilabel Classification of Music into Emotions In: *Proc. 9th International Conference on Music Information Retrieval (ISMIR 2008)*, pp. 325-330, Philadelphia, PA, USA (2008)

[21] Spyromitros, E., Tsoumakas, G., Vlahavas, I.: An Empirical Study of Lazy Multi-label Classification Algorithms. In: Proc. 5th Hellenic Conference on Artificial Intelligence, (SETN 2008). Springer, pp. 401- 406, Syros, Greece (2008)

[22] Fu"rnkranz, J., H"ullermeier, E., Mencia, E. L., Brinker, K.: Multilabel classification via calibrated label ranking. Journal of Machine Learning, pp.73: 133-153 (2008)

[23] Duda, R. O., Hart, P. E., Stork, D. G.: Pattern Classification. John Wiley & Sons, New York (2001)

[24] Tsoumakas, G., Vlahavas, I.: Random k-Labelsets: An Ensemble Method for Multilabel Classification. In: Proceedings of the 18th European on Machine Learning (ECML 2007), pp. 406-417. Warsaw, Poland (2007)

[25] Tsoumakas, G., Katakis, I., Vlahavas, I.: Effective and Efficient Multi-label Classification in Domains with Large Number of Labels. In: Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08), Antwerp, Belgium (2008)

[26] McCallum, A. K.: Multi-label text classification with a mixture model trained by EM, AAAI'99 Workshop on Text Learning1999, // citeseer.ist.psu.edu/ mccallum99 multilabel.html

[27] Chen, W., Yan, J., Ahang, B., Chen, Z., Yang, Q.: Document Transformation for Multi-label Feature Selection in Text Categorization. In: Seventh IEEE International Conference on Data Mining, pp. 451- 456 (2007)

[28]Yan, R., Tesic, J., Smith, J. R.: Model-shared subspace boosting for multi-label classification. In: KDD'07, pp. 834-843, San Jose, California, USA (2007)

[29] Ji, S., Tang, Li., Yu, S., Ye, J.: Extracting shared subspace for multi-label classification. In: Proceedings of the 14th SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, USA (2008)

[30] Breiman, L.: (1996a), Bagging Predictors, Machine Learning, Vol. 24, No. 2, pp. 123-140 (1996)

[31] Ho, T. K.: The Random Subspace Method for Constructing Decision Forests. In: IEEE Transactions on Pattern Analysis and Machine Intelligence 20(8), pp.832-844 (1998)

[32] Breiman, L.: Random Forests, Machine Learning, No.45, pp. 5–32 (2001)

[33] Opitz, D., Maclin, R.: Popular Ensemble Methods: An Empirical Study, Journal of Artificial Intelligence Research 11, pp.169-198 (1999)

[34] Bryll, R., Gutierrez-Osuna, R., Quek, F.: Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets, Pattern Recognition, 36 (6), pp.1291-1302 (2003)

[35] Polikar, R.: Bootstrap-inspired Techniques in Computational Intelligence, Signal Processing Magazine, IEEE, Vol. 24, Issue 4, pp. 59-72. (2007)

[36] DePasquale, J., Polikar, R.: Random Feature Subset Selection for Ensemble Based Classification of Data with Missing Features. In: M. Haindl and F. Roli, Eds. LNCS, vol. 4472, pp.251-260. Springer Berlin (2007)

[37] Panov, P., Džeroski, S.: Combining Bagging and Random Subspaces to Create Better Ensembles. In: LNCS, vol. 4723, Advances in Intelligent Data Analysis VII, pp. 118-129. Springer Berlin (2007)

[38] Quinlan, J. R.: C4.5: Programs for Machine Learning. San Mateo, CA. Morgan Kaufmann (1993)

[39] Barnard, K., Duygulu, P., Forsyth, D., de Freitas, N., Blei, D., Jordan, M. I.: Matching Words and Pictures. Journal of Machine Learning Research, pp.1107-1135, No. 3 (2003).

[40] Snoek, C. G. M., Worring, M., Gemert, J. C. Van, Geusebroek, J.-M., Smeulders, A. W. M.: The Challenge Problem for Automated Detection of 101Semantic Concepts in Multimedia. In: Proceedings of ACM Multimedia, pp. 421-430, Santa Barbara, USA (2006)

[41] Duygulu, P., Barnard, K., de Freitas, J. F. G., Forsyth, D. A.: Object Recognition as Machine Translation: Learning a Lexicon for a Fixed Image Vocabulary. In: Seventh European Conference on Computer Vision (ECCV) (4) pp.97-112, (2002).

[42] Li, T., Zhang, C. and Zhu, S.: Empirical Studies on Multi-label Classification, The 18th IEEE International Conference on Tools with Artificial Intelligence, NOV. 13 – 15, 2006 (ICTAI'06), Washington D.C.

[43] Petrovskiy, M.: Paired Comparisons Method for Solving Multi-label Learning Problem, Proceedings of the Sixth International Conference on Hybrid Intelligent Systems (HIS'06), 2006 IEEE.

[44] Witten, I. H., Frank, E.: Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann (2005)

[45] Nasierding, G., Kouzani, A. Z.: Image to Text Translation: A Review. In: Proceedings of international Conference on Humanized Systems, Beijing, pp378-383, Oct.18-22, (2008)

[46] Nasierding, G. and Kouzani, A. Z. : Image to Text Translation by Multi-label classification. In: 2010 International Conference on Intelligent Computing (ICIC2010), August 18-21, 2010, Changsha, Hunan, China. Lecture Notes in Artificial Intelligence. In press.