A Comparative Analysis of Supervised Multi-label Text Classification Methods

Shweta C. Dharmadhikari
PICT
Pune, India

Maya Ingle
DAVV
Indore, India

Parag Kulkarni
EkLaT Solutions
Pune, India

Abstract: Multi-label classification methods are getting more popular now a days because of their increasing demand in various application domains such as text classification , image classification , functional genomics , music categorization , emotion recognition etc. Multi-label classification methods are falling under two broader categories of problem transformation methods and algorithm adaptation methods. From machine learning perspective both of these types are working under the roof of supervised classification methods wherein the labels are already provided in the training data set. An attempt is made through this paper to present the state of the art supervised text classification techniques and there comparison. The paper also discusses the important results reported so far in text classification domain and also tried to highlight the beneficial directions of the research till date. The experiments are conducted on standard bench mark datasets such as Enron, Bibtex and Slashdot. Moreover, the paper also contains a comprehensive bibliography of selected papers appeared in reputed journals and conference proceedings as an aid for the researchers working in the field of multi-label classification domain.

Keywords: machine learning, multi-label classification, supervised text classification.

1 . INTRODUCTION

In the machine learning context, a large amount of research has been done in traditional single-label and multi-class text classification method. In single-label text classification methods, each training example is associated with a single label l from a set of disjoint labels *L*. But as a matter of fact in most of the real world situations textual data such as documents or web pages are intrinsically belong to multiple labels. The categorization of textual data is perhaps the most dominant multi-label application. In multi-label text classification each training example Y is associated with a set of labels L, ie $Y \subset L$.

In the literature, different methods have been proposed to be applied to multi-label text classification problems. These methods are falling under two broader categories of problem transformation methods and algorithm adaptation methods. Under each of this stated method many algorithms are proposed in the literature. Almost all these existing algorithms are supervised in nature , that means set of labels associated with each instance are already provided in the training data. But , to the best of our very little efforts has been done in providing comparative analysis of these supervised multi-label text classification methods falling under both the groups. In [23] , the authors presented a comparative analysis of some existing methods and they used different evaluation metrics applied to the protein domain. However , by considering increasing number of possible multi-label applications in text domain , there is a need of such a comparative survey to know the state of art.
As a contribution to this important topic, this paper presents a comparative analysis of some of the existing supervised problem transformation and algorithm adaptation methods using standard benchmark text domain datasets of Slashdot, Enron and Bibtex.
This paper provides information about relative strengths and weaknesses of selected 11 supervised problem transformation methods and 5 algorithm adaptation methods. The problem transformation methods considered here are Binary relevance method,Pairwise classification method, Label powerset method ,

Prunned sets method, Ensembles of pruned sets method, Random k-labelsets method, Ranking by pairwise comparison method, Caliberated label ranking method, Collective multi-label classifier, Metalabeler and classifier chains method.
The algorithm adaptation methods considered here are C4.5 , AdaBoost.MH, AdaBoost.MR, Multilabel k-Nearest Neighbours , Back-Propogation Multi-label Learning(BPMLL).
In addition , these methods are evaluated using different evaluation metrics of multi-label classification under the environment of WEKA[24] , MEKA[23] and MULAN[22] framework and provides the comparative results.

The paper is organized as follows : Section 2 & 3 elaborates about the said problem transformation and algorithm adaptation methods and presents their comparison based on their relative merits and demerits; section 4 presents the description about evaluation measures used for multi-label text classification and results that we obtained through their empirical evaluation; In section 5 we have presented the ideas for future work.

2 PROBLEM TRANSFORMATION METHODS

Problem transformation is the process whereby a multi-label problem is transformed into one or more single-label problems. Thus in this scheme, single – label classifiers are employed and their single-label predictions are transformed into a multi-label prediction.
The prime advantage of problem transformation is that it can abstracts away from classifier specifics and be more generally applicable by focusing on issues relevant to all multi-label domains such as modeling label correlations.
Following are the brief descriptions of working methods of various problem transformation methods. Table I enlist all the problem transformation methods described in this paper and also highlights the relative strengths and weaknesses of those.

1] Binary Relevance Method (BR)
It is a popular problem transformation method that learns q binary classifiers, one for each label in L . BR transforms any multi-label problem into L binary problems. It is said to follow "one-vs-rest" paradigm. Each binary classifier is then responsible for predicting the association of a single label.[3][4].

2] Pairwise classification method (PW)
It is a "one-vs-one" paradigm wherein one classifier is associated with each pair of labels. Hence, instead of L binary problems, P = L(L-1)/ 2 binary problems are formed: one for each pair. Each pairwise problem is made up of examples with which either labels are associated, thus forming a decision boundary for these two labels [1][3].

3] Label Powerset Method (LP)

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, which is actually a set of labels. If this classifier can output a probability distribution over all classes, then LP can also rank the labels. The computational complexity of LP depends on the complexity of the base classifier with respect to the number of classes, which is equal to the number of distinct label sets in the training set. It requires as many class labels as in the single-label transformation as there are distinct label sets in the training data [1][3][4].

4] Random k-label sets method(RAkEL)

The random k-labelsets (RA*k*EL) method [6] constructs an ensemble of LP classifiers.
Each LP classifier is trained using a different small random subset of the set of labels. This way RA*k*EL manages to take label correlations into account, while avoiding LP's problems. A ranking of the labels is

produced by averaging the zero one predictions of each model per considered label. Thresholding is then used to produce a bipartition as well.

TABLE I COMPARISON BETWEEN PROBLEM TRANSFORMATION METHODS

Method	**Merits**	**Demerits**
BR	conceptually simple and relatively fast,	*It does not explicitly model label cor-relations. * It can also be affected by class-imbalance.
PW	conceptually simple	* Time complexity is an issue for PW: it is quadratic with respect to the number of labels. *This method is criticized for not dealing well with overlapping labels and struggling to establish disjoint assignments in the multi-label context
LP	can take into account label correlations	*Can also be computationally complex *Leads to overfitting of the training data.
PS	*Run much faster. *It can form new label sets at classification time, and in this way handles irregular labeling, *Can take into account label correlations.	A disadvantage, however, is the reliance on prediction confidence distributions of the base classifier.
EPS	*Provides increased predictive performance. *It allows parallelism *These are scalable	Cannot utilize available unlabeled data for classification
RAkEL	*Computationally simpler *More predictive capability *Can take into account label correlations.	* More time complexity * Cannot utilize available unlabeled data for classification.
RPC	*It is more flexible *It also covers absolute preferences while giving ranking.	*Takes more prediction time and storage capacity
CLR	*It deals with multilabel classification as well as ranking. *It can be generalized	*Computationally expensive. *Cannot utilize available unlabeled data for classification
CML	Taking into account label co-occurances.	It is restricted to pair of labels. It cannot able to model unlabeled data.
ML	*It works efficiently for large scale datasets. *It does not require much cross-validation phase	*It lacks flexibility to be tuned for user-specified precision/recall levels. *Cannot utilize available unlabeled data for classification.
CC	*Provides increased predictive performance * It is scalable , can work with any type of base classifier.	Cannot utilize available unlabeled data for classification

5] Ranking by pair wise comparison (RPC)

Ranking by pairwise comparison (RPC) [7] transforms the multi-label dataset into 2 binary label datasets, one for each pair of labels. Each dataset contains those examples of *D* that are annotated by at least one of the two corresponding labels, but not both. A binary classifier that learns to discriminate between the two labels, is trained from each of these data sets. Given a new instance, all binary classifiers are invoked, and a ranking is obtained by counting the votes received by each label.

6] Caliberated label ranking (CLR)

Calibrated label ranking (CLR) [9] extends RPC by introducing an additional virtual label, which acts as a natural breaking point of the ranking into relevant and irrelevant sets of labels. This way, CLR manages to solve the complete MLR task. The binary models that learn to discriminate between the virtual label and each of the other labels, correspond to the models of BR. This occurs, because each example that is annotated with a given label is considered as positive for this label and negative for the virtual label, while each example that is not annotated with a label is considered negative for it and positive for the virtual label.

7] Instance Differentiation algorithm (INSDIF)

The INSDIF algorithm [10] computes a prototype vector for each label, by averaging all instances of the training set that belong to this label. After that, every instance is transformed to a bag of *q* instances, each equal to the difference between the initial instance and one of the prototype vectors. A two level classification strategy is then employed to learn form the transformed data set.

8] Collective Multi-label classifier(CML)

This method attempts to capture co-occurrence patterns among labels .This classification model learns parameters for pair of labels [14]. This method uses conditional random field for representation of dependencies among the output variables.

9] Metalabeler (ML)

This method obtains the ranking of class membership for each data instance ,and predict the number of top classes from the obtained ranking[19]. This method uses the One-vs-Rest SVM for ranking purpose. The scores obtained from each binary SVM are used to get ranking of the class membership[23]. Then it constructs a meta model to predict the

10] Classifier Chains [CC]

It involves $|L|$ binary classifiers. These are linked along a chain where each classifier deals with the binary relevance problem associated with label $l_j \in L$. The feature space of each link in the chain is extended with the 0/1 label associations of all previous links [4].

11] Pruned Sets Method [PS]

This method is improvement in label –combination method (LC). It treats sets of labels as a single label. This method takes into account the correlations between the class labels. It contains pruning step and sub sampling step. The pruning step removes infrequently occurring label sets from the training data. In sub sampling step it subsamples the label sets which occur frequently in the training data [3][4].

12] Ensembles of Pruned Sets [EPS]

This method uses a Pruned Sets method in an ensemble framework, and uses a voting scheme to produce the prediction confidences. It provides a powerful and general framework. EPS's training algorithm can be used with any multi-label-capable classifier [7].

3 ALGORITHM ADAPTATION METHODS

1] C4.5

This algorithm is based on successful decision tree algorithm. It was adapted in 2001 to handle multi-label data.The output of C4.5 is a decision tree which is constructed from top-down manner [12]. In this tree, for each node attribute which best classifies remaining training examples is chosen. In specific multiple labels were allowed at the leaves of the tree. For this, entropy calculation formula is modified as follows:

$$\text{Entropy}(D) = -\sum_{j=1}^{q} \left(p(\lambda_j)\log p(\lambda_j) + q(\lambda_j)\log q(\lambda_j) \right)$$

where $p(\lambda_j)$ = relative frequency of class λ_j and $q(\lambda_j) = 1 - p(\lambda_j)$.

2] Support Vector Machine with Heterogeneous Feature Kernel (SVM-HF)
This method exploits relationship among the classes. It enhances the basic purely text based SVM learner by augmenting the feature set with |C| extra features, one for each label in the dataset. The cyclic dependency between features and labels is resolved iteratively .Cosine similarity measure is used to calculate the similarity between two documents [5].

3] AdaBoost.MH & AdaBoost.MR

These are two extensions of AdaBoost for multi-label data.In this purpose of using concept of boosting is to find a highly accurate classification rule by combining many weak or base hypotheses, each of which may be moderately accurate. AdaBoost.MH is designed to minimize Hamming loss. In this approach , the goal of the learning algorithm is to predict all of the correct labels[9]. Thus , the learned classifier is evaluated in terms of its ability to predict approximation of the set of labels associated with the given document. AdaBoost.MR is designed to find hypothesis which places the correct labels at the top of ranking.

TABLE II COMPARISON BETWEEN ALGORITHM ADAPTATION METHODS

Method	Merits	Demerits
C4.5	*It allows to choose such attributes which splits the data in the most informative way. *It offers easy learnability	*It does not take into account the correlation among the classes. *It cannot able to utilize the unlabeled data for classification.
AdaBoost.MH& AdaBoost.MR	*Improved accuracy and minimization of Hamming loss error.	*Attempts for generalization results into decrease in performance. *Cannot utilize unlabeled data for classification.
ML-kNN	*Improved performance as compared to other algorithms in terms of hamming loss , ranking loss and coverage. *Can work well on image as well as textual data.	* Cannot utilize unlabeled data for classification.
Back-propogation algorithm for multilabel learning(BP-MLL)	*Outperforms other counterpart in term of ranking loss. *Gives better generalization capability to learning system. *Time cost of making predictions based on the trained model is trivial.	*Computational complexity in training phase is high because of use of neural networks. *Cannot able to utilize unlabeled data for classification.
SVM-HF	*Take into account correlation among classes. *Significant improvement in accuracy for multilabel data.	Accuracy reduces with consideration of unlabeled data

4] Multi-label k- Nearest Neighbours (ML-kNN)

This approach extends the KNN lazy learning algorithm using a Bayesian approach. It uses the maximum a posteriori 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 [22]. In this Euclidean metric is used to measure distances between instances.

5] Back-propogation algorithm for multilabel learning(BP-MLL)

It is the first multi-label neural network algorithm[8][10]. It is derived from the backpropogation algorithm through replacing its error function with new function defined to capture the characteristics of multi-label learning.

4 EXPERIMENTS

This section presents the results obtained from this empirical study. The next three subsection will present the evaluation measures used, dataset information and results for the problem transformation and algorithm adaptation methods.

4.1 EVALUATION MEASURES

In a multi-label text classification problem an example may be associated with set of labels therefore classification of an example may be partially correct or partially incorrect[1][3]. This can happen when a classifier correctly assigns an example to at least one of the labels it belongs to, but does not assign to all labels it belongs to. Also, a classifier could also assign to an example to one or more labels it does not belong to [2][5].

The commonly used performance evaluation measures for multi-label classifiers are broadly categorized in two groups namely bipartition-based and ranking-based [3]. Bipartition-based measures are again having two types called examples-based measures and label-based measure[1][4]. Example-based measures evaluate bipartition over all the examples of the evaluation dataset. Label-based measures decomposes the evaluation process into the separate evaluations for each label. Whereas the ranking-based measures evaluate ranking with respect to the ground truth of multi-label dataset.

However, for the definitions of these measures, let an evaluation dataset of multi-label examples be denoted as $(x_{i,} y_i), i = 1$ to N, $y_i \subseteq L$, is the set of true labels and L={x $_i$ $j =, m\}$ is the set of all labels . Given an examples x , the set of labels that are predicted by an multi-label method is denoted as z. while the rank predicted for a label is denoted as Z_i ,the most relevant label receives the highest rank(1),while the least relevant one receives the lowest rank(M)[13].
Example based measures includes Exact match (accuracy), Hamming loss, Precision ,Recall, F-measure. Label based measure includes macro-averaging and micro-averaging. Whereas ranking based measures includes one-error , coverage and average precision, log-loss.
We evaluated the said algorithms by measuring values of some of the representatives from above measuring techniques. This includes Accuracy , Example based accuracy that is Exact-match, F-measure and log loss.

Exact Match : It is the accuracy measure in the example based scheme. It is computed as :

EXACT-MATCH(D) = $\frac{1}{N}\sum_{i=1}^{N} 1yi = yi$

F-Measure: F-measure is a combination of precision and recall. It is the harmonic average of the two metrics and it is used as an aggregated performance score.

F-Measure = $\frac{2.0 \text{ x precision x recall}}{\text{precision } + \text{ recall}}$

F-Measure = $\frac{1}{N}\sum_{i=1}^{N} \frac{2[Y_i \cap Z_i]}{[Z_i] - \lfloor Y_i \rfloor}$

Accuracy: This measure is proposed by Godbole and Sarawagi in [5] which is independent of example – based and label – based accuracy measures. It is now a days most popular multi-label accuracy measure. It symmetrically measures how close y_i is to Z_i . It is the ratio of the size of the union and intersection of the predicted and actual label sets, taken for each example and averaged over the number of examples.

Accuracy= $\frac{1}{N}\sum_{i=1}^{N}\left[\frac{Y_i \cap Z_i}{Y_i \cup Z_i}\right]$

Log-Loss: This measure is introduced by Jeese Read [4] to overcome some of the limitations of the ranking loss measures. Under this each label error is graded by the confidence at which it was predicted. It also takes into account label relevances at the time of predictions.

LOG-LOSS(D) = $\frac{1}{NL}\sum_{i=1}^{N}\sum_{j=1}^{L}\min\left(-LOG-LOSS(wj, yj), \ln(N)\right)$

4.2 Datasets

We have used three different multi-label datasets namely Slashdot , Enron and Bibtex for the experimentation purpose. Their statistics is described in nutshell in table III.
Enron dataset contains email messages. It is a subset of about 1700 labeled email messages[22]. BibTeX data set contains metadata for the bibtex items like the title of the paper, the authors, etc. Slashdot dataset contains article titles and partial blurbs mined from Slashdot.org[23].

TABLE III : STATISTICS OF DATASETS

Dataset	Examples	Labels	Attributes
Slashdot	3782	22	500
Enron	1702	53	1001
Bibtex	7395	159	1836

4.3 Experimental Results

We evaluated all algorithms under a WEKA-based [24] framework running under Java JDK 1.6 with the libraries of MEKA and Mulan [22][23]. Experiments are run on 32 bit machines with 1.3 GHz clock speed, allowing up to 2 GB RAM per iteration.
For getting results of problem transformation methods we have used Support Vector Machines as the base classifier using WEKA's SMO implementation with default parameters. Ensemble iterations are set to 10 for EPS. Evaluation is done in the form of 5 × 2 fold cross validation on each dataset . Train:test split used is 60:40.
Under the problem transformation methods we have tested the results for few representatives from above given 11 methods. This includes BR which is the basic one , CLR is based on label pairing with ranking approach , Metalabeler which supports large datasets , CC & PS which are advanced ones and also takes into account label correlations , EPS which is advanced one and uses approach of ensembles of classifiers. The results obtained under above stated evaluation measures are given in Table IV.
Under the algorithm adaptation methods we have tested the results for above stated five methods namely C4.5 , AdaBoost.MH & AdaBoost.MR , ML-kNN , BP-MLL and SVM-HF. The results obtained under above stated evaluation measures are given in Table V.

TABLE IV RESULTS OF PROBLEM TRANSFORMATION METHODS

Dataset : Slashdot

Algorithm	Accuracy	Exact Match	F1 Macro	Log Loss
BR	0.43	0.34	0.34	6.4
RAkEL	0.52	0.41	0.35	5.6
METALABELER	0.42	0.51	0.3	4.1
CC	0.47	0.41	0.36	3.3
PS	0.45	0.38	0.31	4.2

Dataset: Enron

Algorithm	Accuracy	Exact Match	F1 Macro	Log Loss
BR	0.39	0.11	0.2	14.5
RAkEL	0.46	0.14	0.21	12.6
METALABELER	0.47	0.51	0.19	17.7
CC	0.44	0.12	0.2	10.2
PS	0.41	0.13	0.15	12
EPS	0.45	0.14	0.16	12.3

Dataset: Bibtex

Algorithm	Accuracy	Exact Match	F1 Macro	Log Loss
BR	0.32	0.12	0.11	18.1
RAkEL	0.38	0.21	0.13	14.7
METALABELER	0.48	0.49	0.19	15.21
CC	0.4	0.08	0.11	17.3
PS	0.37	0.2	0.09	14
EPS	0.41	0.22	0.1	14.57

TABLE V RESULTS OF ALGORITHM ADAPTATION METHODS

Dataset:Slashdot

Algorithm	Accuracy	Exact Match	F1 Macro	Log Loss
C4.5	0.67	0.43	0.38	3.3
AdaBoost	0.84	0.46	0.81	2.7
ML-kNN	0.3	0.24	0.16	3.7
BP-MLL	0.36	0.32	0.34	4.3
SVM-HF	0.85	0.78	0.76	3.1

Dataset:Enron

Algorithm	Accuracy	Exact Match	F1 Macro	Log Loss
C4.5	0.53	0.31	0.33	3.2
AdaBoost	0.71	0.67	0.7	3.1
ML-kNN	0.18	0.2	0.17	4.2
BP-MLL	0.31	0.27	0.32	3.2
SVM-HF	0.76	0.56	0.53	4.2
Dataset:Bibtex				
Algorithm	**Accuracy**	**Exact Match**	**F1 Macro**	**Log Loss**
C4.5	0.43	0.37	0.38	2.7
AdaBoost	0.7	0.62	0.57	2.3
ML-kNN	0.16	0.18	0.19	3.8
BP-MLL	0.27	0.29	0.18	3.1
SVM-HF	0.73	0.48	0.63	3.9

5 CONCLUSIONS AND FUTURE WORK

In this paper we have presented comparative empirical study of some popular supervised multi-label classification methods which are applicable to text domain. In the experimental results it can be observed that more accuracy is obtained in case of algorithm adaptation methods as compare to problem transformation methods. But as the algorithm adaptation methods cannot be generalized thatswhy they are less popular. Also all these methods are using labeled training data for classification. However, in real world obtaining the labeled data is very time-consuming task which needs human-intervention; most of the time unlabeled data is available which is large in number. But unfortunately all the above methods are not able to utilize available unlabeled data for the text classification. So in future there is need to devise the methodology which will effectively utilize the available unlabeled data for multi-label text classification.

References

[1] A. Santos , A. Canuto, and A. Neto, "A comparative analysis of classification methods to multi-label tasks in different application domains", International journal of computer Information systems and Industrial Management Applications". ISSN: 2150-7988 volume 3(2011), pp. 218-227.

[2] R.Cerri, R.R. Silva , and A.C. Carvalho , "Comparing Methods for multilabel classification of proteins using machine learning techniques",BSB 2009, LNCS 5676,109-120,2009.

[3] G. soumakas , I. Katakis , and L. Vlahavas , "Mining Multi-label Data" , Data Mining and knowledge Discovery Handbook, O. Maimon , L. Rokach(Ed.), Springer, 2nd edition, 2010.

[4] J. Read, B. Pfahringer, G. Homes , and E.Frank , "Classifier chains for multi-label classification"., Proc. Of European Conference on Machine Learning and knowledge discovery in Databases, LNAI 5782(254-269),2009

[5] S. Godbole and S. Sarawagi , "Discriminative methods for multi-labeled classification", 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2004.

[6] G. soumakas , I. Katakis , and L. Vlahavas , "random k-Labelsets for Multi-Label Classification", IEEE Transactions on Knowledge and Data Engineering, 2010.

[7] J. Read, B. Pfahringer, G. Homes , and E.Frank, "Multi-label classification using ensembles of pruned sets", Proc. Of 8th IEEE International conference on Data Mining, Pisa, Italy, pages 995-1000. IEEE computer Society , 2008.

[8] M.L. Zhang, " ML-RBF : RBF neural networks for multi-label learning" , Neural Processing Letters, 29(2), 61-74 , 2009.

[9] R.E. Schapire , and Y. Singer , "BoosTexter: A Boosting-based system for Text Categorization", Machine Learning, 39(2-3) 135-168.

[10] M.L. Zhang, Z. Zhou , "Multilabel neural networks with Applications to Functional Genomics and Text Categorization" , IEEE Transactions on Knowledge and Data Engineering, 18 , 10 , 1338-1351, 2006.

[11] K. Trohidis, G. Tsoumakas , G. Kalliris , and I. Vlahavas , "Multilabel classification of music into emotions", proc. Of 9th International conference on Music Information Retrieval, Philadelphia , PA, USA,2008.

[12] H. Blockeel , L. Schietgat, A. Clare, "Decision Trees For hierarchical multilabel classification: A case study in functional genomics", LNCS 4213(2006) 18-29, 2006.

[13] Ueda , N. Saito, " Parametric mixture models for multi-labeled text", Advances in Neural Information Processing Systems 15 (2003) 721-728.

[14] Ghamrawi , N., McCallum , A. “Collective Multi-label classification”, Proc. Of ACM Conference on Information and knowledge management (CIKM ‘ 05), Bremen, Germany (2005) 195-200.

[15] G. Tsoumakas , G. Kalliris , and I. Vlahavas, “Multi-label text classification for automated tag suggestion”, Proc. Of the ECML/PKDD 2008 Discovery Challenge, Antwerp , Belgium(2008)

[16] G. Tsoumakas , G. Kalliris , and I. Vlahavas, “ Effective and efficient multilabel classification in domains with large number of labels”, Proc. Of the ECML/PKDD 2008 workshop on Mining Multidimensional Data (MMD’ 08)(2008) 30-44.

[17] L. Wang, M.Chang, J. Feng, “Parallel and sequential support vector machines for multi-label classification”, International Journal of Information Technology, Vol. 11, No. 9. (2005), pp. 11-18.

[18] T.Li, c. Zhang , S. Zhu, “Empirical Studies on Multi-label Classification” , Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence (2006).

[19] Cesa- Bianchi , Gentile , Zaniboni, “Incremental algorithms for hierarchical classification”, Journal of Machine Learning Research 7 (2006) 31-54

[20] Lewis, Yang, Rose “ Rev1 : A new benchmark collection for text categorization research”, Journal of Machine Learning Research 5(2004) 361-397.

[21] Yang, Y “An evaluation of statistical approaches to text categorization”, Journal of Information Retrieval 1 (1999) 67-88.

[22]http://mulan.sourceforge.net/datasets.html
[23]http://MEKA.sourceforge.net
[24] www.cs.waikato.ac.nz/ml/weka/