Integrated Intelligent Research (IIR) International Journal of Data Mining Techniques and Applications
Volume: 04 Issue: 02 December 2015 Page No.52-58
ISSN: 2278-2419

Text Categorization of Multi-Label Documents For Text Mining

Susan Koshy[1], R. Padmajavalli[2]
[1]Assistant Professor, St.Thomas College of Arts and Science, Chennai,
[2]Associate Professor, Department of Computer Applications, Bhaktavatsalam Memorial College for Women, Chennai

Abstract-Automated text categorisation has been considered as a vital method to manage and process vast amount of documents in digital form that are widespread and continuously increasing.Traditional classification problems are usually associated with a single label.Text Categorization uses Multi-label Learning which is a form of supervised learning where the classification algorithm is required to learn from a set of instances, each instance can belong to multiple classes and then be able to predict a set of class labels for a new instance. Multi-label classification methods have been increasingly used in modern applications such as music categorization, functional genomics (gene protein interactions) and semantic annotation of images besides document filtering, email classification and Web search. Multi-label classification methods can be broadly classified asProblem transformation and Algorithm adaptation. This paper presents anoverview of single-label text classificationand an analysis ofsome multi-label classification methods.

Indexterms-Text categorization, single-label categorization, multi-label categorization

I. INTRODUCTION

Text Mining is the discovery of new, previously unknown information, by automatically extracting information from different written resources. A key element is the linking together of the extracted information together to form new facts or new hypotheses to be explored further by more conventional means of experimentation. Text mining is dependent on the various preprocessing techniques that infer or extract structured representations from raw unstructured data sources and is defined by preparatory techniques. The following preprocessing techniques are required to prepare raw unstructured data for text mining.

- Preparatory processing
- General purpose NLP tasks
- Problem-dependent tasks.

Preparatory processing converts the raw representation into a structure suitable for further linguistic processing. The general purpose Natural Language Processing tasks process text documents using the general knowledge about natural language. The tasks may include tokenization, morphological analysis, Parts Of Speech tagging, and syntactic parsing.The problem-dependent tasks prepare the final representation of the document meaning by text categorization and information extraction. Text categorization (sometime called text classification) tasks tag each document with a small number of concepts or keywords[12].The automated categorization (or classification) of texts into topical categories has been generating interest from as early as 1960's. With the unlimited availability of on-line documents, automated text categorization has envisaged an increased and renewed interest. Text categorization (TC – also known as text classification, or topic spotting) is the task of automatically sorting a set of documents into categories (or classes, or topics) from a predefined set[1].Text Classification tasks can be divided into two categories, supervised document classification where some external mechanism provides information on the correct classification for documents or to define classes for the classifier, and unsupervised document classification or document clustering, where the classification must be done without any external reference, this system do not have predefined classes[3].Machine learning approach automatically builds a classifier by "learning", from a set of previously classified documents. A major issue of text categorization is high dimensionality of feature space and dimensionality reduction needs to be done by feature extraction.

II. BACKGROUND

Categorization involves identifying the main themes of a document by placing the document into a pre-defined set of topics. When categorizing a document, a computer program will often treat the document as a "bag of words." It does not attempt to process the actual information as information extraction does rather, categorization only counts words that appear and, from the counts, identifies the main topics that the document covers[13]. Categorization often relies on a thesaurus for which topics are predefined, and relationships are identified by looking for broad terms, narrower terms, synonyms, and related terms. The goal of text categorization is to classify a set of documents into a fixed number of predefined categories[14][15]. Each document may belong to more than one class. Using supervised learning algorithms the objective is to learn classifiers from known examples called labeled documents and perform the classification automatically on unknown examples called unlabeled documents. Consider a set of labeled documents from a source $D = [d_1, d_2 \ldots d_n]$ belonging to a set of classes $C=[c_1,c_2,\ldots,c_p]$. The text categorization task is to train the classifier using these documents, and assign categories to new documents. In the training phase, the 'n' documents are arranged in 'p' separate folders, where each folder corresponds to one class. In the next step, the training data set is prepared via a feature selection process. Text data typically consists of strings of characters, which are transformed into a representation suitable for learning. In the feature space representation, the sequences of characters of text documents are represented as sequence of words. Feature selection involves tokenizing the text, indexing

and feature space reduction. Text can be tokenized using term frequency (TF), inverse document frequency (IDF), term frequency inverse document frequency (TF-IDF) or using binary representation.

III. DEFINITION OF THE PROBLEM

The general text categorization task can be formally defined as the task of approximating an unknown category assignment function F :D× C → {0, 1}, where D is the set of all possible documents and C is the set of predefined categories. The value of F(d, c) is 1 if the document 'd'belongs to the category 'c'and 0 otherwise. The approximating function M :D× C → {0, 1} is called a classifier, and the task is to build a classifier that produces results as "close" as possible to the true category assignment function F[17].

A. Single-Label versus Multi-label Categorization

Depending on the properties of F, we can distinguish between single-label and multi-label categorization. In multi-label categorization the categories overlap, and a document may belong to any number of categories. In single-label categorization, each document belongs to exactly one category. Binary categorization is a special case of single-label categorization in which the number of categories is two. The multi-labelcase can be solved by |C| binary classifiers (|C| is the number of categories), one for each category, provided the decisions to assign a document to different categories are independent from each other[9][12].Multiclass classification means a classification task with more than two classes; e.g., classify a set of images of fruits which may be oranges, apples, or pears. Multiclass classification makes the assumption that each sample is assigned to one and only one label: a fruit can be either an apple or a pear but not both at the same time.Multi-label classification assigns a set of target labelsto each sample. This can be thought as predicting properties of a data-point that are not mutually exclusive, such as topics that are relevant for a document. A text might be related to any of the topics religion, politics, finance or education at the same time or none of these[9][12].

IV. TEXTCLASSIFICATIONPROCESS

A. Documents Collection

This is first step of the classification process is to collect different types of document which may be in different formats such as like html, .pdf, .doc, etc

B. Pre-Processing

The next step is to prepare the documents which are in different formats and this is the preprocessing step.

a) Tokenization
A document is treated as a string, and then partitioned into a list of tokens.

b) Removing stop words
Stop words such as "the", "a", "and", etc. are frequently occurring, so theinsignificant words need to be removed.

c) Stemming word
Applying the stemming algorithm that converts different word form into similarcanonical form. This step is the process of conflating tokens to their root form, e.g. connection toconnect, computing to compute

C. Indexing

The documents representation is one of the pre-processing techniques that is used to reduce the complexity of the documents and make them easier to handle. The documents have to be transformed from the full text version to a document vector. The most commonly used document representation is called vector space model where documents are represented by vectors of words.The drawbacks of this modelare high dimensionality of the representation, loss of correlation with adjacent words and loss of semantic relationship that exist among the terms in a document. To overcome these problems, term weighting methods are used to assign appropriate weights to the term. Some of the ways to determine the term weight are boolean weighting, word frequency weighting, tf-idf and entropy. The major drawback of this model is that it results in a huge sparse matrix(most elements are zero), which raises a problem of high dimensionality[12].

D. Feature Selection

After pre-processing and indexing the important step of text classification, is feature selection to construct vector space, which improves the scalability, efficiency and accuracy of a text classifier.[4] The main idea of Feature Selection (FS) is to select subset of features from the original documents. FS is performed by keeping the words with highest score according to predetermined measure of the importance of the word. Because of for text classification a major problem is the high dimensionality of the feature space. Many feature evaluation metrics have been notable among which are information gain (IG), term frequency, Chi-square, expected cross entropy, Odds Ratio, the weight of evidence of text, mutual information, Gini index[11].

E. Classification

The automatic classification of documents can be done in three ways, unsupervised, supervised and semi-supervised methods. Some of the popular techniques of classification use machine learning approaches such as Bayesian classifier, Decision Tree, K-nearest neighbor(KNN), Support Vector Machines(SVMs), Neural Networks and Rocchio's Method.

V. APPLICATIONS

A. Automated authorship attribution

Authorship attribution is the science of determining the author of a text document, from a predefined set of candidate authors or inferring the characteristic of the author from the characteristics of documents written by that author[12].

B. Automatic Document Distribution

Text classification also allows the efficient automatic distribution of documents via email or fax by eliminating the time consuming, manual process of faxing or mailing. And this can be achieved by first classifying the documents according to sender and message type[12].

C. Automated survey coding

Survey coding is the task of assigning a symbolic code from a predefined set of such codes to the answer that a person has given in response to an open-ended question in a questionnaire (survey). Survey coding has several applications, especially in the social sciences, ranging from the simple classification ofrespondents on the basis of their answers to the extraction of statistics on political opinions, health, and customer satisfaction etc. are represented as sequence of words[12].

VI. CLASSIFIER ARCHITECTURE

Text classification is a fundamental task in document processing. The goal of text classification is to classify a set of documents into a fixed number of predefined categories/classes. A document may belong to more than one class. When classifying a document, a document is represented as a "bag of words". It does not attempt to process the actual information as information extraction does. Rather, in simple text classification task, it only counts words (term frequency) that appear and, from the count, identifies the main topics that the document covers e.g. if in the document, a particular word comes frequently then it is assigned as its topic (or class) [6] [7][4].Classification is a two-step process. Model Construction:

The set of documents used for model construction iscalled training set. It describes a set of predetermined classes.Each document or sample in the training set is assumed to belongto a predefined class (labeled documents). The model is represented as classification rules, decision trees, or mathematical formulae [7] [8].This is called the Training or Learning phase.Mode Usage: This is the 2nd step in classification and is also calledTesting Phase or Classification Phase, it is used for classifyingfuture or unlabeled documents. The known label of testdocument/sample is compared with the classified result toestimate the accuracy of the classifier. For e.g. the labeled documents of the training set, is used further to classify unlabeled documents. Test set is independent of training set [6][7] [8].According to Nawei Chen and Dorothea Blostein there are four aspects to characterize classifier architecture: (1) document features and recognition stage, (2) feature representations, (3) class models and classification algorithms, and (4) learning mechanisms [2].

VII. CATEGORIZATION TECHNIQUES

There are two approaches to text categorization knowledge engineering and machine learning. The knowledge engineering approach is focused around manual development of classification rules. A domain expert defines a set of sufficient conditions for a document to be labeled with a given category. The development of the classification rules can be quite labor intensive and tedious in this approach.In the machine learning approach, the classifier is built automatically by learning the properties of categories from a set of pre-classified training documents and is an instance of supervisedlearning because the process is guided by applying the known true category assignment function on the training set.

A. Statistical classification

The most widely used type of classifier is Naïve Bayesian classifier which is a probabilistic classifier. Among other statistical classifiers Bayesian classifier is simple and effective.

B. Functional classification

The k-nearest neighbor's algorithm (k-NN) is a method for classifying objects based on closest training examples in the feature space. K-NN is a type of instance-based learning, or lazy learning. It can also be used for regression. The k-nearest neighbor algorithm is one the simplest of all machine-learning algorithms. The space is partitioned into regions by locations and labels of the training samples.

A point in the space is assigned to the class c if it is the most frequent class label among the k nearest training samples. Usually Euclidean distance is used as the distance metric however this will only work with numerical values (continuous values). In cases such as text classification another metric, such as the overlap metric (or Hamming distance) can be used.

C. Neural classification

Neural network is a massively parallel distributed processor made up of simple processing units (neurons), which have a way for storing knowledge from experience and making it available for use. Knowledge is acquired by the network from its environment via a learning process and stored in inter neuron connections (synaptic weights). In principle use of the neural network for any classification task is straightforward, a neural network is taken, data of the feature vector is fed to the inputs of the network and categorization comes from the outputs. Each output is directly assigned a category and if the strongest signal comes out of the neural network on the output number 3, for example, then the object being classified belongs to the third category.

The difference in strength between the strongest output signal and other output signals indicate the confidence the network has in this classification [9]. If no output is strong enough, then the classification can be rejected. However the fundamental problem in the use of a neural network is making the actual design of the network. Theoretically it is possible to construct neural networks of any complexity, but it is very hard to mathematically predict if a given neural network design will be able to excel in a particular classification task. Given this complexity the researchers have concentrated on simple and predictable neural network designs for most practical tasks in the field of text classification and only use more complex designs in the newer and more complex fields of image and speech recognition[10].

VIII. MULTI-LABEL CLASSIFICATION METHODS

We can group the existing methods for multi-label classification into two main categories: a) problem transformation methods, and b) algorithm adaptation methods. Problem transformation methods transform the multi-label classification problem either into one or more single-label classification or regression problems. Algorithm adaptation methods are those that extend specific learning algorithms in order to handle multi-label data directly.

A. Problem transformation methods

Binary Relevance (BR) is a widely-used problem transformation method [18]. BR considers the prediction of each label as an independent binary classification task. BR builds M binary classifiers, one for each different label L (where M = L). For the classification of a new instance, BR outputs the union of the labels 'l_i' that are positively predicted by the M classifiers. The drawback of this method is that it assumes that the labels assigned to an example are independent and correlations among the possible labels are ignored[5].LabelPowerset (LP) is a simple problem transformation method [1]. LP considers each unique set of labels that exists in a multi-label training set as one of the labels of a new single-label classification task. Given a new instance, the single-label classifier of LP outputs the most likely label, which is actually a set of labels. It is an advantage of LP that it takes into account label correlations. However the complexity increases from the large number of label subsets and the majority of these classes are associated with very few examples [19][5].Random k-labelsets (RAkEL) constructs an ensemble of LP classifiers [22]. Each LP classifiers is trained using a different small random subset of the set of labels. An average decision is calculated for each label 'l_i' in L, and the final decision is positive for a given label if the average decision is larger than a given threshold 't'. The RAkEL aims to take into account label correlations and avoids the above problems of LP [19][5].

Classifier Chains (CC) [20] involves |L| binary classifiers as in a binary relevance method. Classifiers are linked along a chain where each classifier deals with the binary relevance problem associated with label 'l_j' ∈ L. The feature space of each link in the chain is extended with the 0/1 label associations of all previous links.Pruned Sets (PS) for multi-label classification is centered on the concept of treating sets of labels as single labels. This allows the classification process to inherently take into account correlations between labels. By pruning these sets, PS focuses only on the most important correlations, which reduces complexity and improves accuracy [21].Ensembles of Classifier Chains (ECC) trains 'm' CC classifiers $C_1,C_2,\ldots,C_m$. Each C_k is trained with a random chain ordering (of L) and a random subset of D. Hence each C_k model is likely to be unique and able to give different multi-label predictions. These predictions are summed by label so that each label receives a number of votes. A threshold is used to select the most popular labels which form the final predicted multi-label set[20].Ensembles of Pruned Sets (EPS) combine pruned sets in an ensemble scheme. PS is particularly suited to an ensemble due to its fast build times and, additionally, the ensemble counters any over-fitting effects of the pruning process and allows the creation of new label sets at classification time[21].

B. Algorithm Adaptations Methods

Extensions of single-label classifiers have been developed which adapt their internal mechanisms to allow their use in multi-label problems. New algorithms can be developed specifically for specific multi-label problems. There are various algorithm adaptation methods proposed based in different algorithms, such as decision trees, probabilistic methods, neural networks, support vector machines, lazy and associative methods and boosting to name a few. Extension of decision tree algorithm, named C4.5, k-Nearest Neighbors (kNN) lazy learning and two extensions of AdaBoost algorithm are proposed, AdaBoost.MH and AdaBoost.MR.ML-kNN (Multi-Label k Nearest Neighbours) extends the popular k Nearest Neighbors (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 neighbors [23].

IX. LEARNING AND EVALUATION OF CLASSIFIER FOR SINGLE LABEL

A classifier by itself does not have knowledge and any knowledge that is required for classification must come to a classifier either by directly translating expert knowledge or from learning. Two major types of learning are supervised learning and unsupervised learning[16].When the network is given more examples to learn from, the results will be better. Two basic measures of successareprecision and recall. First, a confusion matrix is computed. For a simple case of two categories it is a 2x2 matrix where test cases are distributed as follows, first cell is the number of test cases that were correctly assigned to the first category (True Positive), the second is the number of test cases that should have been in the first category, but were classified as belonging to the second one (False Negative), and third and fourth cell respectively for category one that should have been false and vice versa.

Table 1. Confusion Matrix

True Positives	False Negative
False Positives	True Negative

Precision=TP/(TP+FP)
Recall= TP/(TP+FN)

In the simplest supervised learning case all the example data (pairs of input vectors and correct output vectors) is randomly divided into three parts namely training, testing and validation data sets. When the training starts the network is shown examples from the training set in random order and any errors are corrected by back propagation. The process goes through all the training set data multiple times until the recall on the last iteration is higher than a predetermined minimum recall threshold. Sometimes a fixed number of iterations are used. When the training is finished, the testing process starts. In the

testing process all examples from the testing data set are given to the classifier and the precision over the testing data is computed.

If the precision is lower than a predetermined minimum generalisation threshold (70-80%), then the system goes back to the learning stage for a fixed number of iterations. The final stage is the verification stage and is similar to the testing stage and average error across the whole data set is computed. The precision measure, F1 measure computed at the verification stage is the final precision of the classifier and is the measure by which different classification systems can be compared. If the value is good then the network is ready to be used and the learning is complete. However if the verification precision of the network is unsatisfactory then no amount of learning will help.To improve the result the structure of the classifier system must be changed[2].

X. EVALUATION METRICS FOR MULTI LABEL

The evaluation of multi-label classifiers requires different measures than those used in the case of single-label problems. In the single-label problems the classification of an example is correct or incorrect, but in a multi-label problem a classification of an example may be partially correct or partially incorrect. 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 one or more labels it does not belong to. Several measures have been proposed in the literature for the evaluation of multi-label classifiers. According to [18], these measures can be broadly categorized in two groups: bipartition-based and ranking-based. Bipartition-based measures are example based or label-based. Let an evaluation dataset of multi-label examples be denoted as (x_i, Y_i), i=1, …, N, where $Y_i \subseteq L$ is the set of true labels and $L=\{\lambda_j: j=1 \dots M\}$ is the set of all labels. Given an example x_i, the set of labels that are predicted by a multi-label method is denoted as Z_i, while the rank predicted for a label λ is denoted as $r_i(\lambda)$. The most relevant label receives the highest rank (1), while the least relevant one receives the lowest rank (M) [18].

A. Example-based Measures

a) Hamming Loss

Hamming Loss takes into account prediction errors (incorrect label) and missing errors (label not predicted). Then, hamming loss evaluates the frequency that an example-label pair is misclassified, i.e., an example is associated to the wrong label or a label belonging to the instance is not predicted. The best performance is reached when hamming loss is equal to 0. The hamming loss should be less for a better performance of the classifier.

$$\text{Hamming Loss}=\frac{1}{N}\sum_{i=1}^{N}\frac{\left|Y_i \Delta Z_i\right|}{M} \qquad (1)$$

b) Accuracy

Accuracy symmetrically measures how close Y_i(true label) is to Z_i(predicted label). 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.

$$\text{Accuracy}=\frac{1}{N}\sum_{i=1}^{N}\frac{\left|Yi \cap Zi\right|}{\left|Yi \cup Zi\right|} \qquad (2)$$

c) Precision

Precision can be defined as the percentage of true positive examples from all the examples classified as positive by the classification model.

$$\text{Precision}=\frac{1}{N}\sum_{i=1}^{N}\frac{\left|Yi \cap Zi\right|}{\left|Zi\right|} \qquad (3)$$

d) Recall

Recall is the percentage of examples classified as positive by a classification model that are true positive.

$$\text{Recall}=\frac{1}{N}\sum_{i=1}^{N}\frac{\left|Yi \cap Zi\right|}{\left|Yi\right|} \qquad (4)$$

e) F-Measure

F-Measure or F-Score 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.

$$\text{F-Measure}=\frac{1}{N}\sum_{i=1}^{N}\frac{2\left|Yi \cap \right.}{\left|Zi\right| - \left|Yi\right|} \qquad (5)$$

f) Subset Accuracy

Subset Accuracy is a very restrictive accuracy metricwhich considers a classification as correct if all the labels predicted by a classifier are correct.

$$\text{Subset Accuracy}=\frac{1}{N}\sum_{i=1}^{N} I\left(\left|Zi\right| = \right. \qquad (6)$$

B. Label-based Measures

The calculation of these measures for all labels can be done using two averaging operations, known as macro-averaging and micro-averaging. Consider a binary evaluation measure F(tp, tn, fp, fn) that is calculated based on the number of true positives (tp), true negatives (tn), false positives (fp) and false negatives (fn). Micro-averaged precision represents the ratio of examples correctly classified as l (tp) and incorrectly (fp) classified as l. Micro-averaged recall represents the ratio of examples *correctly* classified as l, and all examples actually pertaining to the class l (fn). Micro-averaged F-measure represents a harmonic mean of Micro-Precision and Micro-Recall. |L| represents the number of labels.Macro-average precision is computed first by computing the precision for each label separately, and averaging over all labels. The same procedure is used for computing the macro-averaged recall. Macro-averaged F-measure represents a harmonic mean of Macro-Precision and Macro-Recall[18].

a) One-error

One-error evaluates the frequency of the top-ranked label that was not in the set of true labels. The best performance is reached when one-error is equal to 0. The smaller the value of one-error is, the better is the performance.

b) Coverage

Coverage is defined as the distance to cover all possible labels assigned to a sample *x*. It is loosely related to precision at the

International Journal of Data Mining Techniques and Applications
Volume: 04 Issue: 02 December 2015 Page No.52-58
ISSN: 2278-2419

level of perfect recall. The smaller the value ofCoverage, the better is the performance.

c) Average Precision

Average precision is the average precision taken for all the possible labels and it can evaluate algorithms as a whole. It measures the average fraction of labels ranked above a particular label $l \in Y_i$which is actually in Y_i. The best performance is reached when average precision is equal to 1[18].

XI. EXPERIMENTAL RESULTS

MEKA, a multi label extension of WEKA frameworkis used to briefly study the performance of two datasets Enron and Slashdot.WEKA framework is a collection of machine learning algorithms for data *mining* tasks. WEKA contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization and is well-suited for developing new machine learning schemes.MEKA is based on the WEKA Machine Learning Toolkit and it includes several multi-label methods from scientific literature.The MEKA project provides an open source implementation of methods for multi-label learning and evaluation. WEKA is open source software issued under the GNU General Public License.The EnronandSlashdot datasets available on the MEKA framework which are already created and compiled into WEKA's ARFF(attribute relation file format) have been used.

They are all text datasets, parsed into binary-attribute format using WEKA's StringToWordVectorfilter. Enron dataset is a subset of the Enron email dataset as labelled by the UC Berkeley Enron Email Analysis Project and the Slashdot dataset are articles and partial blurbs mined from Slashdot.org. In these experiments the J48 base classifier is used for BR and CC multi label classifiers and RAkEL is used as the base classifier for MULAN multi label classifier. Experiments are run on 32 bit machine with 3.40 GHz clock speed. The following results were obtained for problem transformation methods of classification on Enron and Slashdot multi label datasets namely Binary relevance and Classifier Chains and MULAN.

Table 2. Description of datasets

Dataset	Class	Attribute	Instances
Slashdot	22	1101	100
Enron	53	1054	100

Table 3. Performance of classifiers on Enron dataset

Enron dataset	BR	CC	MULAN
Accuracy	0.812	0.79	**0.816**
Hamming loss	**0.013**	0.014	0.014
One error	0.176	0.147	**0.118**
Average precision	**0.04**	0.036	0.036
F1 micro average	**0.739**	0.713	0.706
F1 macro average by example.	**0.845**	0.813	0.835
F1 macro average, by label	**0.038**	0.037	0.018
Total_time(build&test)	6.433	**2.267**	5.229

Table4. Performance of classifiers on Slashdot dataset

Slashdot dataset	BR	CC	MULAN
Accuracy	**0.147**	**0.147**	0.113
Hamming loss	0.126	**0.059**	0.092
One error	0.912	**0.853**	**0.853**
Average precision	**0.284**	0.219	0.21
F1 micro average	**0.217**	0.185	0.127
F1 macro average by example.	**0.21**	0.147	0.123
F1 macro average, by label	**0.085**	0.082	0.083
Total_time (build&test)	4.919	**4.375**	11.403

XII. DISCUSSION

The Enron dataset contains 1702 instances and the Slashdot dataset contains 3782 instances and for this study the first hundred instances of eachdatasetwere used. Evaluation measures such as Accuracy, Hamming Loss, One Error, Average precision, F1micro average,F1 macro average and Total time to build and test the classifierwith the optimum performance is indicated in bold in the tables. The Classifier Chain problem transformation method has the least build time for both the datasets and the Binary Relevance classifier shows better performance for the other evaluation metrics.This shows that BR approach is more advantageousand thus validates the findings from literature because of its low computational complexity compared with other multi-label methods. The disadvantage is that it becomes complex when more labels are added to instances[8]. BR ignores label dependency, as it makes the strong assumption of label independency[8].

XIII. CONCLUSION

Text mining is dependent on various preprocessing techniques that infer or extract structured representations from raw unstructured data sources by preparatory techniques.The problem-dependent tasks is a preparatory technique which prepares the final representation of the document meaning by text categorization and information extraction. Text categorization or text classification tasks tag each document with a small number of concepts or keywords.Thispaperused multi label text classification methods such as Binary Relevance, Classifier Chains and MULAN to classify 100 instances each of the datasets for evaluating the classifier.

The Classifier Chain problem transformation method has the least build time for both the datasets and the Binary Relevance classifier shows better performance metrics in accuracy, average precision,F1 micro average and F1 macro average.The currently available classifiers use labeled training data for classification but in realtime obtaining the labeled data is a very cumbersome task which needs human-intervention. Most data available are unlabeled and new methods have to be devised to effectively utilize the available unlabeled data for multi-label text classification.Active learning refers to the task of devising a ranking function that ranks a set of additional unlabeled examples in terms of how much further information they would carry for retraining a better classifier. Further study needs to be done to minimize the human-labeling efforts and multi-label active learning approach can be used to reduce the required labeled data without sacrificing the classification accuracy.

REFERENCES

[1] Fabrizio Sebastiani. "Text categorization" In Alessandro Zanasi (ed.), Text Mining and its Applications, WIT Press, Southampton, UK, (2005), pp. 109-129.
[2] Mahinovs, Aigars, Ashutosh Tiwari, Rajkumar Roy, and David Baxter. "Text classification method review.", Decision Engineering Centre, Cranfield University,UK,(2007)
[3] Chen, Nawei, and Dorothea Blostein. "A survey of document image classification: problem statement, classifier architecture and performance evaluation.", International Journal of Document Analysis and Recognition (IJDAR), Volume 10 Issue. 1, (2007), pp. 1-16.
[4] Goller, Christoph, Joachim Löning, Thilo Will, and Werner Wolff. "Automatic Document Classification-A thorough Evaluation of various Methods." ISI 2000 (2000) pp. 145-162.
[5] Zhang, M., and Z. Zhou. "A review on multi-label learning algorithms", Knowledge and Data Engineering, IEEE, Volume 26 Issue 8, (2013), pp. 1819 - 1837.
[6] Gupta, Vishal, and Gurpreet S. Lehal. "A survey of text mining techniques and applications", Journal of emerging technologies in web intelligence, Volume 1 Issue 1, (2009), pp. 60-76.
[7] Jiawei Han, Michelin Kamber. "Data Mining Concepts and Techniques", Morgan Kaufmann publishers,USA, ISBN 978-1-55860-901-3, (2001), pp. 70-181.
[8] Cherman, Everton Alvares, Maria Carolina Monard, and Jean Metz. "Multi-label problem transformation methods- a case study.", CLEI Electronic Journal, Volume14, Issue 1 (2011)
[9] Sebastiani, Fabrizio. "Machine learning in automated text categorization.", ACM computing surveys (CSUR), Volume 34, Issue. 1, (2002), pp. 1-47.
[10] Haykin, Simon, and Richard Lippmann. "Neural Networks, A Comprehensive Foundation.", International Journal of Neural Systems Volume 5, Issue. 4 (1994) pp.363-364.
[11] Yang, Yiming, and Jan O. Pedersen. "A comparative study on feature selection in text categorization.", In International Conference on Machine Learning, Volume. 97, (1997), pp. 412-420.
[12] Feldman, Ronen, and James Sanger, "The text mining handbook: advanced approaches in analyzing unstructured data", Cambridge University Press, ISBN-13 978-0-511-33507-5, (2007).
[13] KjerstiAas and Line Eikvil "Text Categorization: A Survey" ,Report No. 941. (1999).
[14] Korde, Vandana, and C. Namrata Mahender. "Text classification and classifiers: A survey.", International Journal of Artificial Intelligence & Applications (IJAIA), Volume 3 Issue 2, (2012) pp. 85-99.
[15] Tsoumakas, Grigorios, and Ioannis Katakis. "Multi-label classification: An overview.", Dept. of Informatics, Aristotle University of Thessaloniki, Greece (2006).
[16] Sebastiani, Fabrizio. "Machine learning in automated text categorization.", ACM computing surveys (CSUR), Volume 34 Issue1 (2002), pp 1-47.
[17] Boutell, Matthew R., et al. "Learning multi-label scene classification.", Pattern recognition, Volume 37 Issue 9, (2004), pp 1757-1771.
[18] Tsoumakas, Grigorios, Ioannis Katakis, and Ioannis Vlahavas. "Mining multi-label data", Data mining and knowledge discovery handbook. Springer US, (2010), pp 667-685.
[19] McCallum, Andrew. "Multi-label text classification with a mixture model trained by EM", In AAAI'99Workshop on Text Learning, (1999)pp. 1-7.
[20] J Read, Jesse, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. "Classifier chains for multi-label classification", Machine learning, Volume 85 Issue. 3, (2011), pp. 333-359.
[21] Read, Jesse, Bernhard Pfahringer, and Geoffrey Holmes. "Multi-label classification using ensembles of pruned sets", In Data Mining, 2008. ICDM'08. Eighth IEEE International Conference, IEEE, 2008, pp. 995-1000
[22] Tsoumakas, Grigorios, Ioannis Katakis, and Ioannis Vlahavas. "Random k-labelsets for multilabel classification." IEEE Transactions on Knowledge and Data Engineering, Volume 23 Issue. 7, (2011), pp. 1079-1089
[23] Zhang, Min-Ling, and Zhi-Hua Zhou. "ML-KNN: A lazy learning approach to multi-label learning.", Pattern recognition , Volume 40 Issue 7, (2007) pp. 2038-2048