Journal of Theoretical and Applied Information Technology
10th June 2014. Vol. 64 No.1
© 2005 - 2014 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AN INVESTIGATION OF SUMMARY BASED CLASSIFIERS

[1]ESTHER HANNAH. M , [2]PRAKASH KUMAR, [3]SASWATI MUKHERJEE

[1]Associate Professor, Department of Computer Applications, St. Joseph's College of Engineering, Chennai

[2]Student, Department of Computer Applications, St. Joseph's College of Engineering, Chennai

[3]Professor, Department of IST, Anna University, Chennai

E-mail: [1]hanmoses@yahoo.com , [2]prakask90@gmail.com , [3]msaswati@yahoo.com

ABSTRACT

The work presents a comparative investigation of the performance of some of the well-known machine learning classifiers in the task of summarization. The objective of this paper is to evaluate the capability of classification algorithms in predicting summary sentences and compare its prediction performance against ten well-known machine learning models in the context of the DUC 2002 dataset. Classical classification algorithms based on Trees, Rules, Functions, Bayes and Lazy learners are used in the study. We have used 350 text documents from DUC2002 (Document Understanding Conference 2002) for prediction. This paper evaluates the capability of classification algorithms in predicting candidate sentences for summary generation. The results indicate that the prediction performance of machine learning classifiers in the task of text summarization is better than the baseline performance.

Keywords: *Summarization, Classification, Evaluation, Machine Learning, Feature Extraction*

1. INTRODUCTION

The widespread use of computers in today's society means that large quantities of data are stored electronically. This data relates to virtually all facets of modern life and is a valuable resource if the right tools are available for using them, and more than eighty percent of this data is in the form of text. Text can be mined in a systematic, comprehensive and reproducible way, and business critical information can be captured and harvested automatically. Hence high performing text mining tools are the need of the hour.

Text mining extracts precise information based on keywords, entities or concepts, relationships, phrases, sentences and even numerical information in context. Text mining software tools often use computational algorithms based on Natural Language Processing, or NLP, to enable a computer to "read" and analyze textual information. These tools interpret the meaning of the text, identify extracts, synthesize and analyze relevant facts and relationships between different parts of a text document. Therefore, text mining techniques need to be designed to effectively manage large numbers of textual elements with varying frequencies.

As the amount of information on the Internet grows abundantly, it is difficult to select and classify relevant information. Automatic text summarization can automate this work completely or at least assist in the process by producing a draft summary. Traditionally researchers looked at designing statistical models for achieving this. More recently, attention has turned to a variety of machine learning methods that can automatically build models describing the structure at the heart of a set of data.

Down the years, a number of machine learning techniques were used to summarize the essential information in human-readable form, thus helping people can use them to analyze the domain from which the data originates[33] [14] [34]. *Radev et al* (2002) defines a summary as "a text that is produced from one or more texts that conveys important information in the original text(s), and that is no longer than half of the original text(s) and usually significantly less than

© 2005 - 2014 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

that" [2]. How summarization systems are built and, how effectively they perform, what kinds of summarization do people want, how sophisticated should a summarization be, and what milestones should mark quantum leaps in summarization theory & practice are some of the critical issues which are continuously addressed by the NLP community.

The process of summarizing enables us to grasp the essence of a text quicker, and hence heps in reduces the time one spends on getting the gist of a text document. In addition to this, the knowledge gained by summarizing makes it possible to easily analyze and also to critique the original text.

Classification techniques have been found to be successful in systems that capture knowledge and this makes it easier to atomize tasks that are already successfully performed by humans. Data mining, machine learning, database, and information retrieval communities have enjoyed the benefits of classification algorithms and are currently used by applications in diverse domains, such as target marketing, medical diagnosis, news group filtering, email categorization and document organization. If classifier models have made a mark in various information retrieval tasks, then they should be able to perform well in text summarization tasks also. It is therefore motivating to investigate the capability of machine learning classifiers in automatic summary prediction. In this paper we have analyzed the performance of ten classification models and how they are capable of classifying summary sentences. The prediction performances are evaluated in terms of accuracy, precision, recall and F-measure.

The objective of this paper is to evaluate the capability of classification algorithms in predicting summary sentences and compare its prediction performance against ten well-known machine learning models in the context of the DUC 2002 dataset. The compared models are three tree based classifiers techniques: (i) Iterative Dichotomiser (ID3) (ii) J48 (iii) logistic model trees (LMT); two neural networks techniques: (i) Multi-layer Perceptrons (MLP) and (ii) Radial Basis Function (RBF); one Bayesian technique: (i) Naı¨ve Bayes (NB); One Rule based classifier techniques: (i) Single conjunctive rule learner; One Lazy learner: (i) Nearest neighbor classifier(IBK); Two optimization algorithms: (i) Sequential minimal optimization(SMO) (ii) Support vector machine(SVM)

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 provides an overview of the proposed work. Section 4 discusses the conducted empirical evaluation and its results. Section 5 concludes the paper and outlines directions for future work.

2. RELATED WORKS

One of the very first works in automatic text summarization was done by Luhn et al in 1958, demonstrates research work done in IBM, focused on technical documents[13]. Luhn proposed that the 'frequency of word' proves to be a useful measure in determining the significance factor of sentences. Words were stemmed to their roots having the stop words removed. Luhn generated a set of content words that helped to calculate the significance factor of sentences which were then scored, and the sentences become the candidates to be part of the generated summary. In the same year, Baxendate et al proved that the sentence position plays an important role in determining the significance factor of sentences [12].

Almost all the known techniques for classification such as decision trees, rules, Bayes methods, nearest neighbor classifiers, SVM classifiers, and neural networks have been extended to handle text data. Recently, a considerable amount of emphasis has been placed on linear classifiers such as neural networks and SVM classifiers, with the latter being particularly suited to the characteristics of text data. Now-a-days, the advancement of web and social network technologies have led to a tremendous interest in the classification of text documents containing links or other meta-information. These classification methods are used separately or combined with other methods to achieve better performance.

Some authors [21] have proposed to parallelize and distribute the process of text classification. With such a procedure, the performance of classifiers can be improved in both accuracy and time complexity. Recently in the area of Machine Learning the concept of combining classifiers is proposed as a new direction for the improvement of the performance of individual classifiers.

© 2005 - 2014 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

In the context of combining multiple classifiers for text categorization, a number of researchers have shown that combining different classifiers can improve classification accuracy [22], [23]. Qiang et al [24] performed experiments using k-NN LSI, a new combination of the standard k-NN method on top of LSI, and applying a Decomposition, to decompose the vector matrix. The Experimental results showed that text categorization effectiveness in this space was better and it was also computationally less costly, because it needed a lower dimensional space. The authors of [25] present a comparison of the performance of a number of text categorization methods in two different data sets. In particular, they evaluate the Vector and LSI methods, a classifier based on Support Vector Machines (SVM) and the k-Nearest Neighbor variations of the Vector and LSI models. Their results show that overall, SVMs and k-NN LSI perform better than the other methods, in a statistically significant way. Researchers are also applying on pruning techniques on trees, training datasets, etc. Guan and Zhou proposed a training-corpus pruning based approach to speed up the process [19]. By using this approach, the size of training corpus can be reduced significantly while classification performance can be kept at a level close to that of without training documents pruning according to their experiments.

Recent research has shown that the incorporation of linkage information into the classification process can significantly improve the quality of the underlying results. Compared to other sophisticated models, classification models can be quickly generated and are extremely useful tools for many practical data mining applications where both predictive accuracy and the ability to analyze the model are important. Han and Kamber[7] describes data mining software that allow the users to analyze data from different dimensions, categorize it and summarize the relationships which are identified during the mining process. WEKA[6] toolkit is a widely used toolkit for machine learning and data mining that was originally developed at the University of Waikato. WEKA contains tools for regression, classification, clustering, association rules, visualization.

Al-Radaideh, et al[3] applied a decision tree model to predict the final grade of pupils who studied a particular course in an university. For the analysis three different classification methods namely ID3, C4.5, and the Naïve Bayes were used. The outcome of their results indicated that Decision Tree model had better prediction than other models.

3. PROPOSED WORK

Classification techniques have been found to be successful in systems that capture knowledge and this makes it easier to mechanize tasks that are already successfully performed by humans, although genetic algorithms, neural networks, fuzzy logic, etc can perform well. In the present work, we propose a method of modeling the task of summarization as a classification problem where sentences in a document are classified into one of the two classes as '*interesting*' or *'not_so_interesting*'. In the proposed work we employ a classification based view of text summarization, in a way we can exploit the prediction capabilities of various classification learning algorithms. Classifier models which are tree-based, Rule-based, function based, probabilistic-based and Lazy-learning algorithms are generated. The performance of these classifier models in classifying a given sentence as a summary sentence or not is evaluated in terms of

In an earlier work [26] the authors have identified seven features for every sentence of a text document namely title feature, sentence length, term weight, sentence to sentence similarity, proper noun, thematic word and numerical data. For a given sentence 'S' in the original text documents, these seven attributes are extracted from each sentence in each text document. The class attribute is a binary attribute based on the presence or absence of the sentence 'S' in the given model summary document. Classifier models which are tree-based, Rule-based, function based, probabilistic-based and Lazy-learning algorithms are generated. The performance of these classifier models in classifying a given sentence as a summary sentence or not is evaluated.

The Summary based classifier is divided into stages two phases namely the training phase and the prediction phase. The training phase includes steps like preprocessing, feature extraction, generation of training dataset, applying the classification algorithm while the latter applies the classifier model learnt in the training phase to unknown set of features. The

© 2005 - 2014 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

performance of the various classifier algorithms is then evaluated. The system deals with various preprocessing method such as sentence segmentation, tokenization, stop word removal and stemming. After preprocessing, important features for every sentence of the document is extracted.

3.1 Attributes of the Training Dataset

For any task of text mining, features play an important role. The features are attributes that attempt to represent the data used for the task. The system focuses on seven features for each sentence. Each feature is given a value between '0' and '1'. Selecting relevant features and deciding how to encode them for a learning method can have an enormous impact on the learning method's ability to extract a good model. Much of the interesting work in building a classifier is deciding what features might be relevant, and how we can represent them. Although it's often possible to get decent performance by using a fairly simple and obvious set of features, there are usually significant gains to be had by using carefully constructed features based on a thorough understanding of the task at hand.

3.2 The Classifier Models

Classification is a data mining (machine learning) technique used to predict group membership for data instances. The objective of classification is to build a model in training dataset to predict the class of future objects whose class label is not known. The problem of classification is defined as follows. We have a set of training records D = {X1,...,XN}, such that each record is labeled with a class value drawn from a set of k different discrete values indexed by {1...k}. The training data is used in order to construct a classification model, which relates the features in the underlying record to one of the class labels. For a given test instance for which the class is unknown, the training model is used to predict a class label for this instance. We have investigated ten well known classifier models namely (i) Iterative Dichotomiser (ID3) (ii) J48 (iii) logistic model trees (LMT); (iv) Multi-layer Perceptrons (MLP) and (v) Radial Basis Function (RBF); (vi) Naı¨ve Bayes (NB); (vii) Single conjunctive rule learner; One Lazy learner: (viii) Nearest neighbor classifier (IBK) (ix) Sequential minimal optimization(SMO) (x) Support vector machine(SVM). The following subsection gives an overview of each of these classifier models.

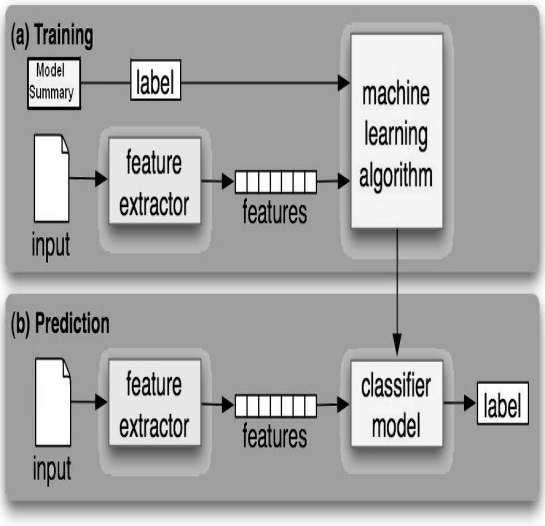

Figure 1 – Machine Learning Classifiers For Summary Prediction

3.2.1 Decision tree based Algorithms

There are many classification algorithms available in literature but decision trees is the most commonly used because of its ease of implementation and easier to understand compared to other classification algorithms[27][28]. One of the most useful characteristics of decision trees is their comprehensibility. People can easily understand why a decision tree classifies an instance as belonging to a specific class.Decision Trees (DT) tree learning algorithms work based on processing and deciding upon attributes of the data[2]. We have used three implementations of the decision trees namely the J48, ID3 and LMT.

3.2.2 Artificial Neural Networks based Algorithms

Artificial Neural Networks (ANN) are one of the common classification methods in data mining. We have used the Multi Layer Perceptron (MLP) neural network and the Radial Base Function (RBF) for our summary prediction. MLP is a feed forward network that makes a model to map input data to output data. Hidden layer in MLP can include various layers between input and

© 2005 - 2014 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

output. The structure of MLP is shown in Figure 2.

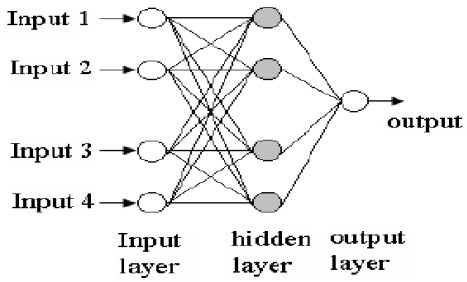

Fig 2 Multi Layer Perception Schema

RBF is another type on Artificial Neural Network. The input of the Neural Network in RBF is linear and the output is nonlinear. The output of this type of ANN is taken from weighted sum of hidden layer's output. The RBF networks are divided in two feed-forward layer. The key to a successful implementation of these networks is to find suitable centers for the Gaussian functions. Figure 3 is illustrates the structure of the RBF.

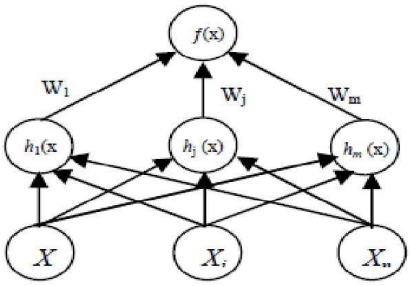

Fig 3 Radial Base Network

3.2.3 Optimization Based Algorithms: This is a standard algorithm that is widely used for practical machine learning. Part is a more recent scheme for producing sets of rules called "decision lists"; it works by forming partial decision trees and immediately converting them into the corresponding rule. We make use of the "sequential minimal optimization" (SMO) algorithm for support vector machines(SVM), which are an important new paradigm in machine learning[18].

3.2.4 Bayesian Algorithms

Bayesian methods are also used as one of the classification solutions in data mining. In our work we use two main Bayesian methods namely naive Bayes[2] and Bayesian networks that are implemented in WEKA software for classification. A NaiveBayes classifier is a simple probabilistic classifier based on applying Bayes' theorem (from Bayesian statistics) with strong (naive) independence assumptions. A more descriptive term for the underlying probability model would be "independent feature model[2]".

3.2.5 Rule Based Algorithms

A Rule Based Classifier is a technique for classifying records using a collection of "If ….Then... " Classification problem .The Rules for the model are represented in Disjunctive Normal Form ,R=(r1 v r2 v ...rk),Where R is called as a *Rule Set* and ri's are called *Classification Rule* or Disjuncts.

3.2.5 Conjunctive Learner: Single conjunctive rule learner is one of the machine learning algorithms and is normally known as inductive learning. The goal of rule induction is generally to induce a set of rules from data that captures all generalizable knowledge within that data, and at the same time being as small as possible.

3.2.6 Lazy Based Algorithms

The compact form proposed in this work represents a rule set without information loss and allows the regeneration of the complete rule set. This form also allows reaching very low support thresholds and mining large rule sets. Rule compression provides a space-effective representation of these large rule sets in the classifier. Nearest neighbors algorithm(IBK) is considered as statistical learning algorithms and it is extremely simple to implement and leaves itself open to a wide variety of variations. The nearest neighbor classifier finds the closest training-point to the unknown point and predicts the category of that training point accordingly to some distance metric.

4. EXPERIMENTAL SETUP AND EMPIRICAL EVALUATION

This section discusses the experimental setup made for investigating how the various nachine learning classifiers perform on summarization data. The empirical study evaluates the capability of various classifier models in predicting summary sentences. We used the open source WEKA[6] machie learning toolkit to conduct this study.

© 2005 - 2014 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4.1 Corpus Used

The TIPSTER program with its two main evaluation style conference series TREC & Document Understanding Conference-DUC (now called as Text Analysis Conference-TAC) have shaped the scientific community in terms of performance, research paradigm and approaches. We used 350 documents from DUC2002 for generating the training dataset. Model summaries provided by DUC2002 were used to evaluate our system.

The main features of the document corpus are:

a. Each document has minimum of 7 sentences and maximum of 33 sentences.

b. The total number of sentences in the corpus is 3960.

4.2 Independent Variables

The independent variables are seven attributes namely: Title Feature, Sentence Length, Term Weight, Sentence to Sentence Similarity, Proper Noun, Thematic Word and Numerical Data. The 'class' identifies whether the chosen sentence is part of the summary document or not. Table 1 shows feature score of 0 to 0.3 is given a class name 'low' a notation 'L'. A score of 0.31 to 0.7 is given the class 'average' under the notation 'A'. The final delimit 0.71 to 1.0 having class name 'high' is having a notation 'H'

(a) Title Feature: The numbers of title words in the sentence 'S' contribute to this feature. This feature is determined by counting the number of matches between the content words in a sentence and the words in the title(S-sentence)

Score(S)= No. Title word in S / No. Word in Title

(b) Sentence Length: The number of words in sentence gives good idea about the importance of the sentence. This feature is mainly very useful to filter out short sentences such as datelines and author names commonly found in articles. The short sentences are not expected to belong to the summary. Normalized length of the sentence is the ratio of the number of words occurring in the sentence 'S' over the number of words occurring in the longest sentence of the document.

Score (S) = No. Word occurring in S / No. Word occurring in longest sentence

(c) Term Weight: Calculating the average of the TF-ISF (Term frequency, Inverse sentence frequency). The frequency of term occurrences within a document has often been used for calculating the importance of sentence 'S'.

Score (S) = Sum of TF-ISF in S / Max (Sum of TF-ISF)

(d) **Sentence to Sentence Similarity:** Similarity between sentences, for each sentence 'S', the similarity between sentence 'S' and each other sentence is computed by the cosine similarity measure. The score of this feature for a sentence 'S' is obtained by computing the ratio of the summation of sentence similarity of a sentence 'S' with each other sentences over the maximum value sentence similarity.

Score (S) = Sum of Sentence Similarity in S / Max (Sum of Sentence Similarity)

(d) Proper Noun: The number of proper noun in sentence, sentence inclusion of name entity (proper noun). Usually the sentence that contains more proper nouns is an important one and it is most probably included in the document summary. The score for this feature is calculated as the ratio of the number of proper nouns in sentence 'S' over the sentence length (S).

Score (S) = No. Proper nouns in S / Sentence Length (S)

(e) Thematic Word: The number of thematic word in sentence, this feature is important because terms that occur frequently in a document are probably related to topic. The number of thematic words indicates the words with maximum possible relativity. The system identifies the top most frequent content words for consideration as thematic. The score for this feature is calculated as the ratio of the number of thematic words in sentence 'S' over the maximum summary of thematic words in the sentence.

Score(S)=No. Thematic Word in S/Max (No. Thematic Word).

(f) Numerical Data: The number of numerical data in sentence, sentence that contains numerical data is important and it is most probably included in the document summary. The score for this feature is calculated as the ratio of the number of numerical data in sentence 'S' over the sentence length (S).

Score (S) = No. Numerical data in S / Sentence Length (S)

Journal of Theoretical and Applied Information Technology
10th June 2014. Vol. 64 No.1
© 2005 - 2014 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1: Feature Categories

Feature Score	Class	Notation
0.00 – 0.30	Low	L
0.31 – 0.70	Average	A
0.71 – 1.00	High	H

4.3 Dependent Variables

The study focuses on predicting whether a sentence is 'INTERESTING' or 'NOT_INTERESTING' rather than how many summary sentences it generates. Accordingly, the dependent variable is a Boolean variable : *CLASS* (whether or not the sentence is in summary or not).

Table 2. The training dataset shown above is a function of several inputs, here the number being seven lines. Each input is classified based on the seven parameters discussed in the system implementation section. Line 1, or sentence 1, is based on the title feature and has been given the notation 'H', meaning high, the order of occurrence - highest - being in the category 0.71 to 1.0. The seventh sentence is given a 'L' signifying a low in F7 or its 'numerical data' significance, meaning it has a low or below 0.3 weight of 'effective numbers' involved; similar conditions follow. The class given at the extreme right, is a condition 'NOT_INTERSTING' or 'INTERESTING' with 'NOT_INTERESTING' specifying that the sentence in the summary is not found in the original input ; and 'INTERESTING' specifying that the sentence in the summary is found in the original sentence.

Table 2 Sample Training Dataset

F1	F2	F3	F4	F5	F6	F7	CLASS
H	A	H	A	H	L	L	INTERESTING
A	H	A	A	A	A	L	INTERESTING
L	H	H	A	A	A	L	NOT-INTERESTING
L	L	H	H	A	L	L	NOT-INTERESTING
L	H	H	L	L	L	L	INTERESTING
L	H	H	L	A	L	L	NOT-INTERESTING
L	H	H	L	L	L	H	NOT-INTERESTING

4.4 Prediction Performance Measures

The performance of prediction models for two-class problem (e.g. interesting or not_interesting) is typically evaluated using a confusion matrix which is shown table-3. We have made use of the commonly used prediction performance measures [29] accuracy, precision, recall and F-measure to evaluate and compare prediction models quantitatively. These measures are derived from the confusion matrix.

4.4.1. Accuracy

Accuracy is also known as correct classification rate. It is defined as the ratio of the number of sentences correctly predicted to the total number of sentences. It is calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

4.4.2. Precision

Precision is also known as correctness. It is defined as the ratio of the number of sentences correctly predicted as ' interesting' to the total number of sentences predicted as 'interesting'. It is calculated as follows:

$$Precision = \frac{TP}{TP + FP}$$

4.4.3. Recall

Recall is also known as summary detection rate. It is defined as the ratio of the number of sentences correctly predicted as interesting to the total number of sentences that are actually interesting. It is calculated as follows:

$$Recall = \frac{TP}{TP + FN}$$

Both precision and recall are important performance measures. The higher the precision, the less effort wasted in classifying and checking; and the higher the recall, the fewer interesting sentences go unidentified.[30]

4.4.4 F-measure

F-measure is a way of combining recall and precision scores into a single measure of performance. F-measure considers both precision and recall equally important by taking their harmonic mean[30]. It is calculated as follows:

F-measure = (2*recall*precision)/ (recall + precision)

© 2005 - 2014 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4.5 Parameters initialization

The parameters for each of the investigated prediction model were initialized mostly with the default settings of the WEKA toolkit as follows:

- *Support vector machines (SVM):* the regularization parameter (C) was set at 1; the kernel function used
was Gaussian (RBF); and the bandwidth (c) of the kernel function was set at 0.5.
- *K-nearest neighbor (Lazy learners):* the number of observations (k) in the set of closest neighbor was set at 3.
- *Multi-layer perceptrons (MLP):* a three layered, fully connected, feedforward multi-Layer perceptron (MLP) was used as network architecture. MLP was trained using backpropagation algorithm. The number of hidden nodes varied based on the size and nature of the datasets. Therefore, we used MLP with 4 hidden nodes for DUC 2002. All nodes in the network used the sigmoid transfer function. The learning rate was initially 0.3 and the momentum term was set at 0.2. The algorithm was halted when there had been no significant reduction in training error for 500 epochs with a tolerance value to convergence of 0.01.
- *Radial basis function (RBF):* k-means clustering algorithm was used to determine the RBF center c and width r. The value of k was set at 2.
- *Naive Bayes (NB*): it does not require any parameters to pass.
- *Decision tree (DT*): it uses the well-known J48, Id3, and LMT algorithms to generate decision tree. The confidence factor used for pruning was set at 25% and the minimum number of instances per leaf was set at 2.

In addition to the above parameters initialization, a default threshold (cut-off) of 0.5 was used for all models to classify a module as defect-prone if its predicted probability is higher than the threshold.

4.6 Evaluation Metrics

Evaluation of a summary is perhaps the most difficult task since there is no universal standard to measure and evaluate a summary. When we already have a manual summary in hand, then the automatically generated summary can be compared against it. Since human judgment is expensive, in terms of resources, and is inconsistent, a need for an evaluation standard is a necessity. The study of machine learning classifiers for the task of summarization was evaluated using the following two methods: training set method and 10-fold cross validation. The results of the study are shown in table 3.

4.6.1 Training Set Method

We have used 60 % of the documents in the DUC 2002 corpus for training and the remaining 40% for testing the CBS model. Seven feature attributes as discussed in section 3.1.2 for every sentence present in the input test document is extracted and is given as input to the classifier model. Based on the classifier model learnt in the training phase, each sentence was classified as '*interesting*' or '*not-interesting'*.

4.6.2 Cross-Validation

A 10-fold cross-validation [31] was used to evaluate the performance of the prediction models. Each dataset was randomly partitioned into 10 bins of equal size. For 10 times, 9 bins were picked to train the models and the remaining bin was used to test them, each time leaving out a different bin. This cross-validation process was run 100 times, using different randomization seed values for cross-validation shuffling in each run to ensure low bias.

5 Experimental Results

Obtaining a good estimate of the performance of a learned function (or model) is an important aspect in machine learning. It is needed for comparing different learning algorithms, and is sometimes used as well within a learning algorithm to assess the quality of the model learned so far. The investigation of classifiers on the summarization task was evaluated using the training set test method and the 10-cross fold method, and the results based on the accuracy metric is given below in table-3.

© 2005 - 2014 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table – 3 Classifier Accuracies

Machine Learning Classifiers		Training Set Method		10-fold Cross Validation	
Techniques	**Algorithm**	**% of Correctly Classified Instance**	**% of Incorrectly Classified Instance**	**% of Correctly Classified Instance**	**% of Incorrectly Classified Instance**
Trees	ID3	76.8382	23.1618	59.1912	35.2941
	J48	66.9118	33.0882	66.9118	33.0882
	LMT	59.1912	35.2949	65.0735	34.9265
Rules	Conjunctive	66.9118	33.0882	66.9118	33.0882
Functions	MLP	76.4706	23.5294	62.8676	37.1324
	SVM	68.0147	31.9853	66.1765	33.8235
	SMO	66.9118	33.0882	63.6029	36.3971
	RBFNetwork	68.75	31.25	66.1765	33.8235
Bayes	NavieBayes	67.6471	32.3509	62.8676	37.1324
Lazy	IBK	76.8382	23.1618	61.3971	38.6029

Table 3 shows the percentage of correctly classified and incorrectly classified instants after the various classification models are trained using 605 of DUC data and tested on 40% of the DUC data. The graphical representation of the results of training set method and 10-fold cross validation method are shown in fig4 and fig 5 respectively.

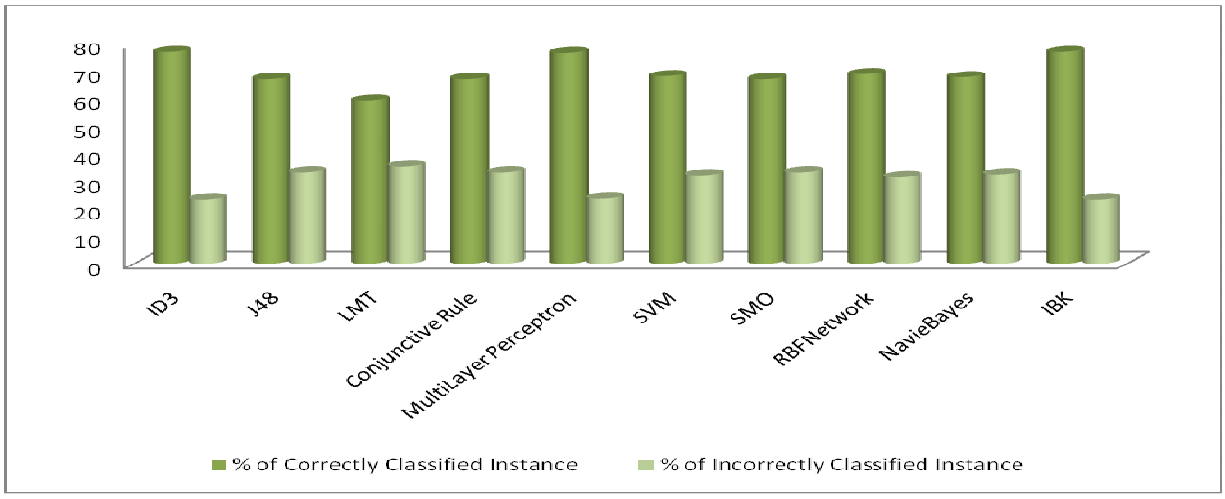

Fig 4: Comparing Various Classifiers Using Training Set Method

Journal of Theoretical and Applied Information Technology
10th June 2014. Vol. 64 No.1
© 2005 - 2014 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

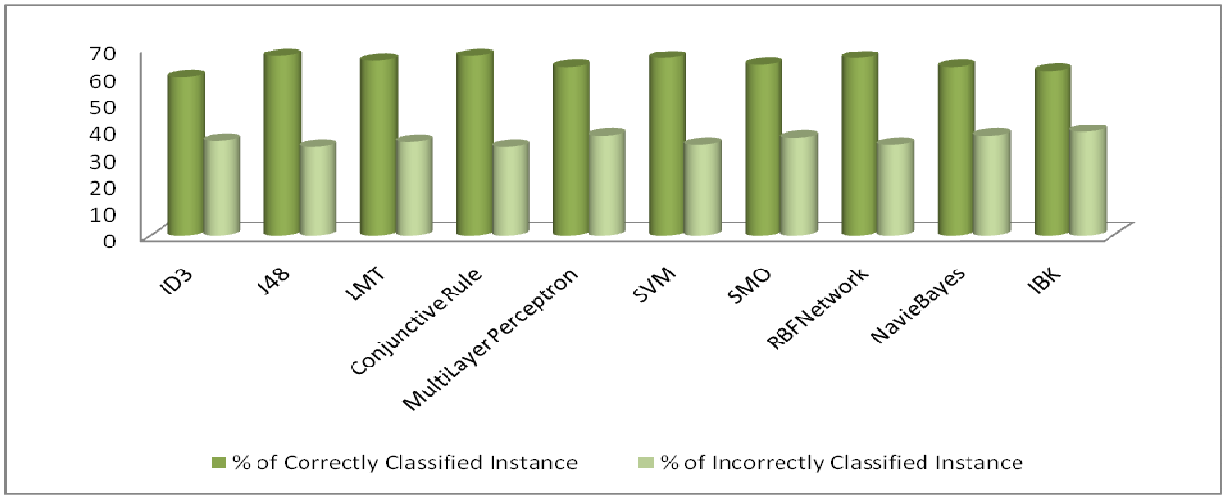

Fig 5: Comparing Various Classifiers Using 10-Fold Cross Validation Method

In addition to the use of the accuracy, we have also made use of one of the most commonly used evaluation metrics which are precision, recall, and F-measure [32] using both the test method shown in Table – 4 shows the recall, precision and F-measure for the various classifier models. The graphical representation is shown in fig - 6 and Fig -7

Table- 4 Shows Comparison Of Different Classifiers Using Two Test Methods

Classifier Algorithm	Training Set Method			10 Cross Fold Method		
	Precision	Recall	F-Measure	Precision	Recall	F-Measure
Id3	0.765	0.768	0.751	0.595	0.626	0.604
J48	0.448	0.669	0.536	0.448	0.669	0.536
LMT	0.595	0.626	0.604	0.444	0.651	0.528
Conjunctive Rule	0.448	0.669	0.536	0.448	0.669	0.536
MLP	0.758	0.765	0.75	0.589	0.629	0.598
SVM	0.784	0.68	0.561	0.53	0.662	0.539
SMO	0.448	0.669	0.536	0.471	0.636	0.526
RBFNetwork	0.676	0.688	0.607	0.603	0.662	0.581
NavieBayes	0.642	0.676	0.628	0.562	0.629	0.571
IBK	0.765	0.768	0.751	0.576	0.614	0.587

Journal of Theoretical and Applied Information Technology
10th June 2014. Vol. 64 No.1
© 2005 - 2014 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

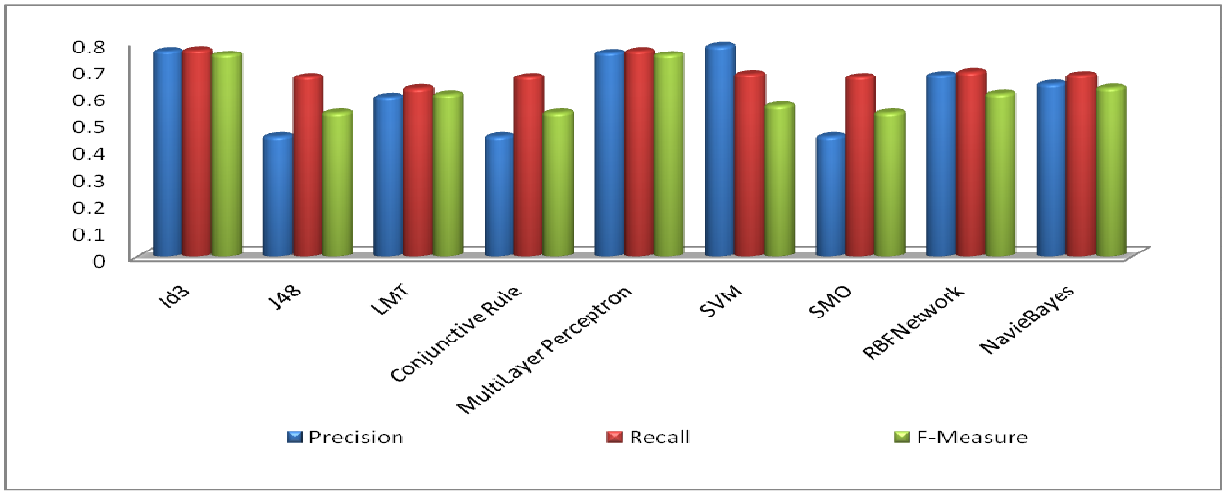

Fig 6 Recall, Precision ANF F-Score -Training Set Method.

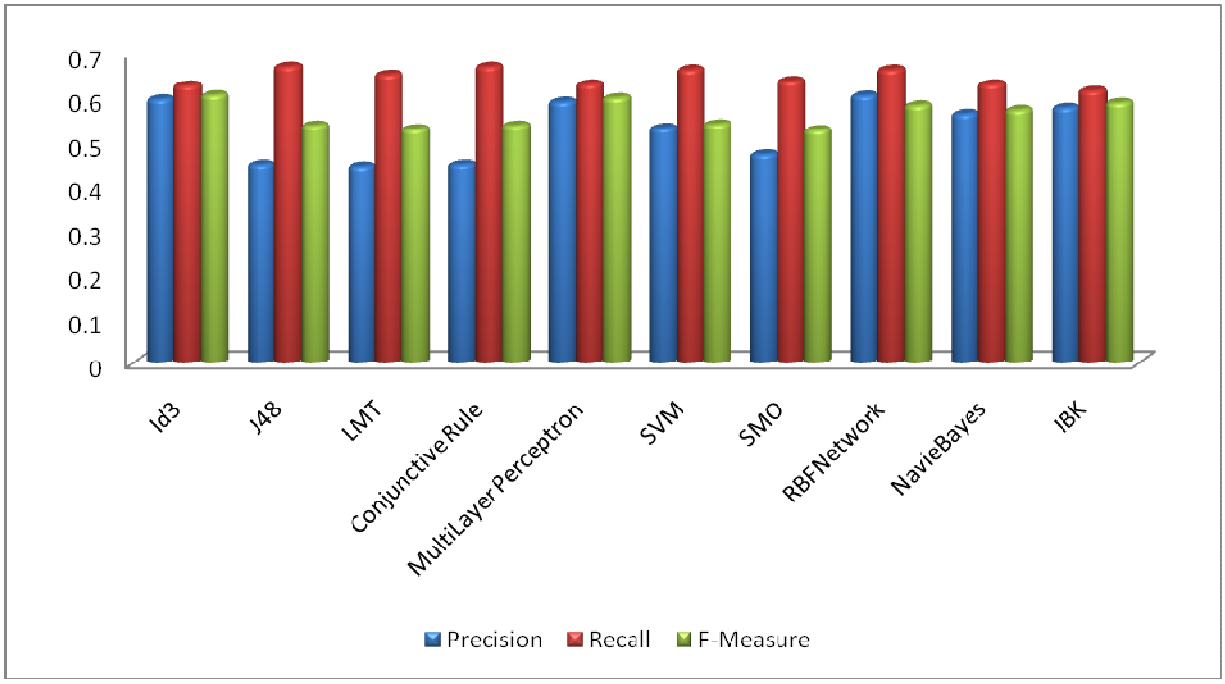

Fig 7 Recall, Precision AND F-Score – 10 CROSS FOLD Method.

5.1. Additional Evaluation metrics and Discussion

In addition to the metrics accuracy, precision , recall and F-measure, we have also investigated some statistics pertaining to mean square error. The following subsections gives an overview of the metrics used the results are shown in table 5.

5.1.1 Kappa statistic: The Kappa[11][18] Statistic can be defined as measuring degree of agreement between two sets of categorized data. Kappa result varies between 0 to 1 intervals. If Kappa = 1, then there is perfect agreement. If Kappa = 0,

© 2005 - 2014 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

then there is no agreement. If values of Kappa statics are varying in the range of 0.40 to 0.59 considered as moderate, 0.60 to 0.79 considered as substantial, and above 0.80 considered as outstanding.

5.1.2 Mean Absolute Error (MAE): Mean absolute error[18] can be defined as sum of absolute errors divided by number of predictions. It is measure set of predicted value to actual value[11] i.e. how close a predicted model to actual model.

5.1.3 ROC Area : ROC[11] provides comparison between predicted and actual target values in a classification[18]. It describes the performance of a model with complete range of classification thresholds or in other words, it has been used for model comparisons. ROC area varies between 0 to 1 intervals . By default classification threshold for binary classification is .5. When the probability of a prediction is 50% or more, the model predicts that class. Changes in the classification threshold affects the predictions made by the model; if the threshold for predicting the positive class is changed from 0.4 to 0.7 then fewer positive predictions will be made. This will affect the distribution of values in the confusion matrix.

5.1.4 Root Mean Square Error (RMSE) : Root mean square error[18] is defined as square root of sum of squares error divided number of predictions. It is measure the differences between values predicted by a model and the values actually observed. Small value of RMSE[11] means better accuracy of model. So, minimum of RMSE & MAE better is prediction and accuracy.

Table 5 Shows Comparison Of Different Classifiers Using Two Test Methods

Algorithms	Training Set Method				10 Cross Fold Method			
	Kappa Statistic	MAE	ROC Area	RMSE	Kappa Statistic	MAE	ROC Area	RMSE
Id3	0.4199	0.301	0.815	0.3879	0.0795	0.4193	0.501	0.5194
J48	0	0.4428	0.5	0.4705	0	0.4428	0.496	0.4705
LMT	0.0795	0.4193	0.501	0.194	0.3391	0.4404	0.536	0.4698
Conjunctive Rule	0	0.4428	0.5	0.4705	0	0.4428	0.496	0.4705
MultiLayer Perceptron	0.4195	0.3391	0.809	0.4001	0.0642	0.4343	0.481	0.5274
SVM	0.0441	0.3199	0.517	0.5656	0.0071	0.3382	0.497	0.5816
SMO	0	0.3309	0.5	0.5752	0.563	0.364	0.478	0.6033
RBF Network	0.1123	0.4213	0.628	0.4588	0.0492	0.4328	0.562	0.47
NavieBayes	0.1387	0.4147	0.63	0.4913	0.00082	0.4352	0.545	0.483
IBK	0.4199	0.3618	0.817	0.3879	0.392	0.4269	0.484	0.5124

The table 4 shows comparisons of different classifiers with the help of two test option i.e. use training method and 10 cross fold method. In this proposed method ten parameters are used to analyze the performance of classifier as well as to find out which method is better. From the 5 statistics, it is clear that the use training set method has better performance than 10 cross fold method.

Journal of Theoretical and Applied Information Technology
10th June 2014. Vol. 64 No.1
© 2005 - 2014 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

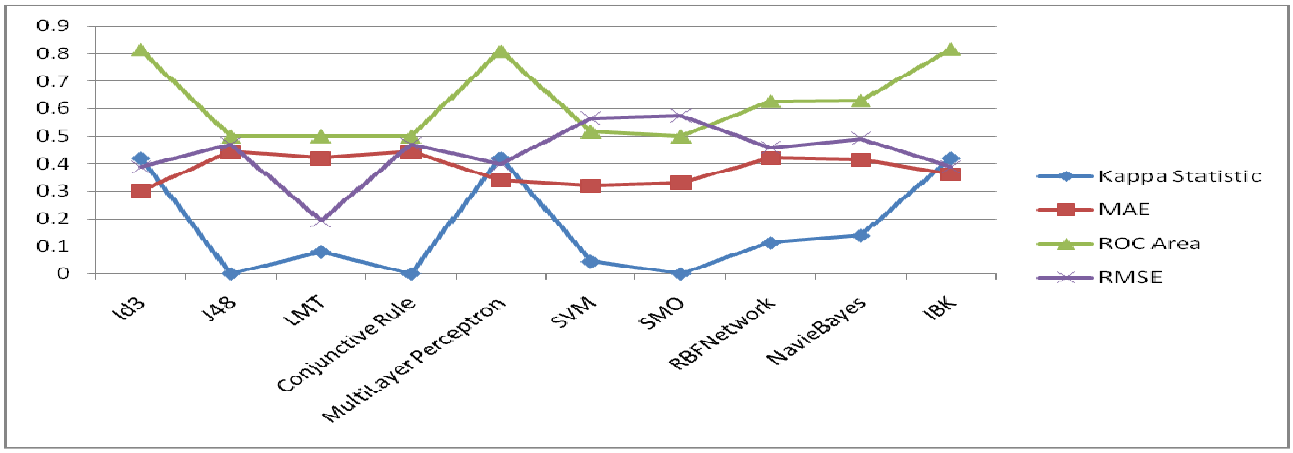

Fig 8 Comparison Of Classifiers Using Training Set Method

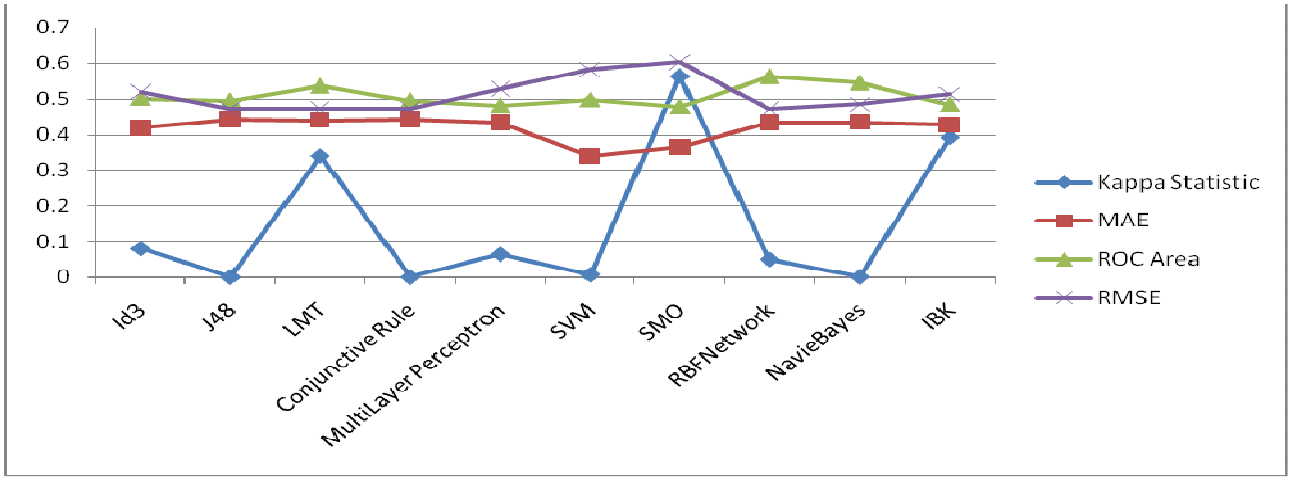

Fig 9 Comparison Of Classifiers Using 10 Cross Fold Method

5.2 Discussion of Results

Table 4 and table 5 shows Using the algorithms available in weka tool like ID3, J48, LMT, Conjunctive, MLP, SMO, SVM, RBFNetwork. The classification of sentences Figure 4 shows the comparison of different classifiers with the help of recall, precision and F-measure using both the training set method and 10 cross fold method. These figures also state that use of training set method has better performance than 10 cross fold method. In table 4 it can be noted that SVM has the highest Precision followed by ID3, but SVM has got a lower Recall than ID3 and hence a lower F-measure.

The value of kappa statistic provides poor agreement of predictions with actual class in both cases but comparing above discussed into two classes namely 'INTERESTING' and 'NOT_ INTERESTNG'. During the study IBK algorithm showed 76.8382% of correct instances and 23.1618% of wrong instances with help of 10 Cross Fold Method option.

method, use training set method has better value than 10 cross fold method. Another reason is the Higher value of ROC means higher positive predictions that can affect matrix which provides the value of sensitivity, specification etc. Analyzing the performance of individual classifiers rather than model, multilayer perceptron has best performance among all these classifiers by using use training set method. Maximum value of Kappa statistic is 0.41, minimum value of RMSE and MAE is l0.09 & 0.25 and maximum value of ROC is 0.87 and the logistic regression has poor performance. The

© 2005 - 2014 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

performance of navie bayes and Bayesian net is same. But in the 10 cross fold method, the performance of navie bayes, ID3, J48, SMO, SVM, Bayesian and multilayer perceptron, the multilayer perceptron shows the best performance among all these method i.e. kappa statistic value (0.01) almost null predications of actual class, value of RMSE & MAE (0.09 &0.2) and value of ROC (0.741).

6. CONCLUSION

This paper has empirically evaluated the capability of ten machine learning classifiers in predicting summary sentences and compared its prediction in the context of the DUC 2002 dataset. The investigation of the classifier models were evaluated using well known metrics namely accuracy, recall, precision and F-measure. The Models were tested using two training dataset and 10-cross fold methods.

However the investigation was carried out only for the English language. The experiment can be used carried out only with well formatted documents with respect to the grammatical rules of the English language. Only a little number of classifier are discussed. The classifier cannot understand an exhaustive set of mathematical and special symbols and considers them as individual strings. The study deals with a syntactic approach. It is a known fact that the performance of an algorithm is dependent on the domain and the type of the data set, and hence study may be domain-dependent. Similar experiments can be conducted for all sub-tasks of textmining. The future work will focus on attribute selection attribute techniques. Images can also be compared by changing the features of the extraction categories. Future work can include some other languages. The future work will be focused on using the other classification algorithms of data mining.

REFERENCES

[1] Abdullah H. Wahbeh, Qasem A. Al-Radaideh, Mohammed N. Al-Kabi, andM. Al-Shawakfa.," A Comparison Study between Data Mining Tools over some Classification Methods",(IJACSA),pp.18 – 26,2011.

[2] Avellaneda D.A., "Natural Texture Classification: A Neural Network Models Benchmark", pp. 325-329, 2009.

[3] AI-Radaideh Q.A., AI-Shawakfa E.W., and AI-Najjar M.I., "Mining student data using decision trees", International Arab Conference on Information Technology(ACIT'2006), Yarmouk University, Jordan,2006.

[4] Cortez P, and Silva A., "Using Data Mining To Predict Secondary School Student Performance", In EUROSIS, Brito A and Teixeira J (Eds.), pp.5-12, 2008.

[5] Edmundson H.P., "New Methods in Automatic extracting", Journal of the Association for Computing Machinery 16 (2), pp.264-285, 1969.

[6] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.

[7] Han J and Kamber M., "Data Mining: Concepts and Techniques," Morgan Kaufmann,2000

[8] Hijazi S.T, and Naqvi R.S.M.M., "Factors Affecting Student's Performance: A Case of Private Colleges", Bangladesh e-Journal of Sociology, Vol. 3, No. 1, 2006.

[9] Hovy E. H., "Automated Text Summarization. In R. Mitkov (ed), The Ox-ford Handbook of Computational Linguistics, chapter 32, pages 583-598.Oxford University Press, 2005.

[10]Jiawei Han, Michelin Kamber., "Data Mining Concepts and Techniques" [M], Morgan Kaufmann publishers, USA, pno70-181, 2001.

[11]Justin T*, et al.*, "Comparison of different classification methods for emotion recognition," pp.700- 703,2007.

[12]Kupiec J., Pedersen J., and Chen F., "A Trainable Document Summarizer", In Proceedings of the Eighteenth Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR), Seattle, WA, pp.68-73, 1995.

[13]Luhn H.P., "The Automatic Creation of Literature Abstract", IBM Journal of research and development, vol. 2, pp.159-165,1958.

[14]Mani, I., House, D., Klein, G., et al.," The TIPSTER SUMMAC Text Summarization Evaluation. In Proceedings of EACL, 1999.

[15]Roiger et al.,"Data mining: a tutorial- based primer by Richard J.Roiger and Michael W. Geatz, Boston, Massachusetts, Addison Wesley.

[16]Shilpa Dhanjibhai Serasiya,

© 2005 - 2014 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

NeerajChaudhary., "Simulation of Various Classifications Results using WEKA" International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-1, Issue-3, August,pno 156, 2012.

[17] WEKA., the University of Waikato, available at:www.cs.waikato.ac.nz/ml/weka/ , 2011.

[18] Witten I. H., Frank E, and Hall M.A., "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, 2011.

[19] Guan J., Zhou S., "Pruning Training Corpus to Speedup Text Classification", DEXA 2002, pp. 831-840

[20] M. IKONOMAKIS, S. KOTSIANTIS and V. TAMPAKAS, "Text Classification Using Machine Learning Techniques", WSEAS TRANSACTIONS on COMPUTERS, Issue 8, Volume 4, August 2005, pp. 966-974

[21] Verayuth Lertnattee, Thanaruk Theeramunkong, "Parallel Text Categorization for Multi-dimensional Data", Lecture Notes in Computer Science, Volume 3320, Jan 2004, Pages 38 – 41.

[22] Bao Y. and Ishii N., "Combining Multiple kNN Classifiers for Text Categorization by Reducts", LNCS 2534, 2002, pp. 340-347.

[23] Sung-Bae Cho, Jee-Haeng Lee, Learning Neural Network Ensemble for Practical Text Classification, Lecture Notes in Computer Science, Volume 2690, Aug 2003, Pages 1032 – 1036.

[24] Wang Qiang, Wang XiaoLong, Guan Yi, A Study of Semi-discrete Matrix Decomposition for LSI in Automated Text Categorization, Lecture Notes in Computer Science, Volume 3248, Jan 2005, Pages 606 – 615.

[25] Ana Cardoso-Cachopo, Arlindo L. Oliveira, An Empirical Comparison of Text Categorization Methods, Lecture Notes in Computer Science, Volume 2857, Jan 2003, Pages 183 – 196

[26] Esther Hannah, T.V.Geetha, Saswati Mukherjee, (2011), "Automatic Extractive Text Summarization Based On Fuzzy Logic: A Sentence Oriented Approach", Swarm, Evolutionary,and Memetic Computing, Part I, LNCS 7076, pp. 530–538.

[27] Thair Nu Phyu, (2009), "Survey of Classification Techniques in Data Mining", Proceedings of the International MultiConference of Engineers and Computer Scientists, vol. 1, pp. 978-988.

[28] Matthew N. Anyanwu, Sajjan G. Shiva, 'Comparative Analysis of Serial Decision Tree Classification Algorithms', International Journal of Computer Science and Security, (IJCSS) Volume (3) : Issue (3) 231- 238.

[29] Witten, I., Frank, E., 2005. Data Mining: Practical Machine Learning Tools and Techniques, second ed. Morgan Kaufmann, San Francisco.

[30] Koru, A., Liu, H., 2005. Building effective defect-prediction models in practice. IEEE Software, 23–29.

[31] Kohavi, R., 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI), pp. 1137– 1143.

[32] Lin, C.-Y. and E.H. Hovy, (2003), "Automatic Evaluations of Summaries using N-gram, Co-occurrence statistics ",HTL-NAACL, pp. 150-157.

[33] Cristianini, N., Shawe-Taylor, J., 2000. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, Cambridge, UK..

[34] Chen, W., Hsu, S., Shen, H., 2005. Application of SVM and ANN for intrusion detection. Computers and Operations Research 32, 2617–2634.