A Machine Learning Approach for Text and Document Mining

Hrishikesh Deka
Student, Department of Information Technology
Gauhati University Institute of Science and Technology
Guwahati-781014, India
hrishikesh.deka1991@gmail.com

Parismita Sarma
Assistant Professor, Department of Information Technology
Gauhati University Institute of Science and Technology
Guwahati-781014, India
parismita.sarma@gmail.com

Abstract— Text Classification or Text Categorization is performed to automatically categorize a set of documents into its respective categories. With the rapid growth of World Wide Web and increasing electronic documents, the Text Categorization becomes an essential method for knowledge discovery and organizing the information. Different tools of Information Retrieval (IR) and Machine Learning (ML) are used for the classification process. A review on machine learning approach for text classification have been done in this paper and also a new system has been proposed for efficient classification. KNN (K Nearest Neighbor) is one of the most promising classification methods for Information retrieval. The main disadvantage of this method is that it has very high computational complexity. This is due to the fact that it considers all the training samples. A combination of traditional KNN classification algorithm and K-Means clustering algorithm has been proposed to overcome this difficulty. The terms are weighted using Term Frequency- Inverse Document Frequency after the Preprocessing steps are performed. Then the K-Means clustering algorithm will be used to group all the training samples and the cluster centers will be considered as the new training samples. The KNN classification algorithm is then performed to find out the category of the documents. A Decision Tree algorithm will then be performed to find out the sub category of the documents. The accuracy will be evaluated using precision, recall and F measure.

Keywords- Text classification, KNN classification algorithm, Cluster, Decision tree

I. INTRODUCTION

As the internet grows, a large number of text information is now available in the form of computer readable electronic documents. The process of Automatic Information Retrieval thus got much importance in recent years due to the exponential growth of the number of documents in digital form. Text classification or Text categorization is the process to categorize the digital documents into respective categories that describe the contents of the documents. Each document can belong to one or more categories based on their contents. The Preprocessing step is followed by a Text classification algorithm in a Text Categorization process. Feature extraction and Feature Selection are the two main steps of Preprocessing. In Feature extraction, tokenization, stop word removal and stemming are carried. In Feature selection, the term weighting methods are carried out to find out the most relevant information from a set of documents. For text classification, there are different algorithms. K nearest neighbor is one of the most efficient and simple classification algorithms used for text classification [1]. But this algorithm also has some limitations. To overcome these limitations, a clustering method such as K means clustering algorithm can be used.

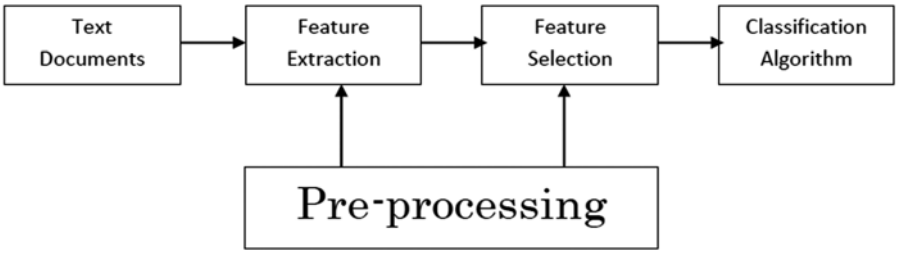

Fig. 1. Text classification process

In this paper, a process of classification of documents into their respective categories is reviewed. After classifying the documents, a decision tree classification algorithm is again used with each category to find out the sub-categories. For example, a set of documents are first classified into categories such as sports, news, entertainment, business etc. Then using the decision tree classifier, each of these categories is again divided into subcategories. An overview of the text classification process is shown in Figure 1.

II. LITERATURE REVIEW

[1] The authors used KNN based machine learning approach to classify different documents into some specific categories. Some newspaper articles are classified into different classes such as 'people', 'places', 'Exchange', 'Organization' and 'topics'. The authors used Reuters-21578 dataset for training and testing purpose. The SGML files of the dataset are first converted into text files. Then the keywords are extracted using bag of words algorithm. Then using the TF-IDF algorithm, the Universal Dictionary is prepared. Then the database is created and using the text classification algorithm, the documents are classified into various categories.

[2] The authors provided reviews on different techniques of text classification based on the existing literature. Feature selection and feature extraction are the techniques of document representation. Tokenization, stop word removal and stemming are different steps for feature extraction. Features are selected based on different weighting schemes. Machine learning approaches such as Bayesian classifier, Decision Tree, K-nearest neighbor (KNN), Support Vector Machines (SVM), Neural Networks, Latent Semantic Analysis, Rocchio's Algorithm, Fuzzy Correlation and Genetic Algorithms etc. have been reviewed by the authors.

[3] The authors discussed about various preprocessing techniques related to text mining. Three different preprocessing techniques namely stop word removal, stemming and TF-IDF algorithm are discussed in this paper. There are different stop words removal methods. They are classic method, method based on Zipf's Law, mutual information method and term based random sampling. Stemming is used to identify the root of a word. There are various stemming algorithms available. Among these, Porters stemmer is most frequently used in the field of text mining. This paper also discussed a weighting scheme TF-IDF. The importance of a word to a document can be measured using this method.

[4] The authors presented results of some of the document clustering techniques such as agglomerative hierarchical clustering and K-means clustering. The results showed that K-means clustering and its variants performs well than that of hierarchical techniques. The complexity of hierarchical techniques is quadratic whereas that of K-means is linear. Also bisecting K-means, a variant of traditional K-means algorithm performed better than UPGMA and regular K-means algorithm.

III. TEXT CLASSIFICATION PROCESS

A. Text Documents

In this step, a set of text documents are collected. The documents are of different categories such as business, entertainment, news, sports, tech etc. There are different data sets available for this purpose.

B. Feature Extraction

This is the first step in preprocessing. The full text documents are converted into word format in this step. Most of the features in a text document are noisy and irrelevant. So the feature extraction is carried out to find out a low dimensional vector. It is an important step for the classification process. To make the classification process more efficient and save more space, the feature extraction process should be carried out more effectively [2]. In general, the steps of the feature extraction are:

1. Case folding: All the letters of the text documents are converted to lowercase letters. Also other characters which are not letters are removed considering them as delimiters.
2. Tokenization: A full text document is converted into set of tokens or terms
3. Stop word removal: Some tokens are more important compared to other tokens. Insignificant tokens such as "and", "are" etc. should be removed from the set of tokens.
4. Feature selection: A Vector Space is created in feature selection process to increase the efficiency, scalability and accuracy of the classifier. Feature selection also reduces the dimensionality of the feature space. The importance of each word is measured using different term weighting schemes. Words with maximum importance are then kept and others are discarded. Thus the dimensionality of the feature space is reduced. "Term Frequency – Inverse Document Frequency (TF-IDF)" is a commonly used weighting method. It is a simple and efficient method to find out the importance of the words on a document [3].

Term Frequency (TF) calculates the frequency of appearance of a word in a document. Higher TF value indicates higher importance of a word,

$$TF(t,d) = 0.5 \times \frac{0.5 + f(t,d)}{Maximum\ occerance\ of\ words} \tag{1}$$

Where, *TF (t, d)* is the term frequency of term t in document d and *f (t, d)* is the frequency of appearing of term t in document d.

Inverse document frequency is based on the number of words that appear in all the documents,

$$IDF(t,d) = \log\left(\frac{N}{n_i}\right) \tag{2}$$

Where, N is the total number of documents and n_i is the number of documents containing term i.
Now, TF-IDF is calculated as follows,

$$TF - IDF = TF(t,d) \times IDF(t,d) \tag{3}$$

C. Classification algorithm

Text documents can be classified by unsupervised, supervised or semi-supervised methods. The task of automatic text classification has recently gained much importance and thus extensively studied. Supervised classification techniques are used for automatic text classification process. There are different machine learning approaches such as K Nearest Neighbor, Bayesian Classifier, Decision Tree, Neural Network, Rocchio's Algorithm, Support Vector Machine etc. These machine learning approaches are supervised techniques. In automatic text classification process, pre-defined category labels are assigned to documents based on the likelihood suggested by a training set of labeled documents.

IV. PROBLEM STATEMENT

The classification rate of the traditional KNN algorithm is severely degraded when there is noisy or irrelevant data present in the training set. Also, the computational complexity of the algorithm becomes very high when the training set contains a large number of documents. It is not suitable for a real time system. In KNN classification algorithm, a vector space containing feature vectors of all the documents is created. For classification, the distance of the feature vector of the query document from each of the vector in the vector space should be calculated. So, when there are a large number of documents, the computational complexity becomes very high.

To reduce the computational complexity of the algorithm, the traditional KNN algorithm may be combined with some clustering algorithm. This will create a centroid vector from all the feature vectors of a specific category and the new centroid vector will act as the new feature vector for classification. There are different clustering methods available for document clustering. Of these methods, K-Means clustering method and its variants shows [4] better results while dealing with document clustering. So, in this paper, we presented a combination of KNN classification with K-Means clustering algorithm in order to increase efficiency and accuracy of the classification process by reducing computational complexity.

V. PROPOSED METHOD

In this paper, we proposed a method to categorize a set of documents and then to find the category of a query document using the trained dataset. The subcategory of the query document will then be determined based on the content of the documents. The different steps of the classification process are shown in the Fig. 2.

Pre-processing consists of three steps, Tokenization, Stop word removal and term weighting scheme. In tokenization, a common method called 'bag of words' is used to create a set of tokens from the text documents. The extracted tokens are stored in different text documents. All the words are extracted in tokenization process. But some of these words have very less impact in the classification process. Common words like articles, prepositions and pro-nouns etc. should not be considered as they do not define a document [3]. These words are called 'stop words'. Different stop word list is available for different processes. A stop word list can also be created as per the requirement of the system. Removing these words will also reduce dimensionality of the vector space. The characters other than alphabets such as numbers or special characters are also removed. There are various methods for stop word removal as discussed in [3]. These are classic method, methods based on Zipf's law, mutual information method and term based random sampling method. In this paper, we used a classic method to remove the stop words. A pre-compiled list is used to match the words in the document with

the words in the list. If the match occurs, then the word is removed from the document considering it as a stop word.

All the terms do not have equal importance on a document of the training set. To find out the importance of a word to a document, different weighting schemes are suggested by the researchers. Term frequency- Inverse document frequency (TF-IDF) is one of the most common and frequently used weighting schemes in the field of Information retrieval and text mining. The value of TF-IDF will determine how many times a word appears in a document. TF-IDF is calculated as the multiplication of the two terms 'term frequency' and 'inverse document frequency'.

A. K Means clustering

As shown in [4], Hierarchical clustering techniques have a quadratic time complexity while K-means clustering technique and its variants have a linear time complexity. 'Bisecting K-means' is a simple and efficient variant of K-means. It can show better results than normal K-means or agglomerative hierarchical clustering. The nearest neighbors of a document may often belong to a different class because most of the documents share a large number of same words. This is why hierarchical clustering works poorly with document clustering. The cluster quality can be evaluated using two metrics 'entropy' and 'F-measure'. Entropy measures the goodness for un-nested clusters and F-measure determines the effectiveness of hierarchical clustering.

K-means is one of the simplest unsupervised learning algorithms. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assumed k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. The centroids should be placed as far away as possible from each other. In the next step, each point from a given data set is selected and associated it to the nearest centroid. After all the points are associated, k new centroids as barycenters of the clusters resulting from the previous step should be calculated. Then again the same data set points should be associated with the nearest new centroid. This process should be carried out until no more changes occur. Thus the final k centroids are found out. These k centroids will now act as the new training set for the classification algorithm. This process reduces the dimensionality of the vector space thus reducing the complexity and increasing the efficiency of the KNN classification algorithm. The process flow of the K-means algorithm is shown at Fig. 3.

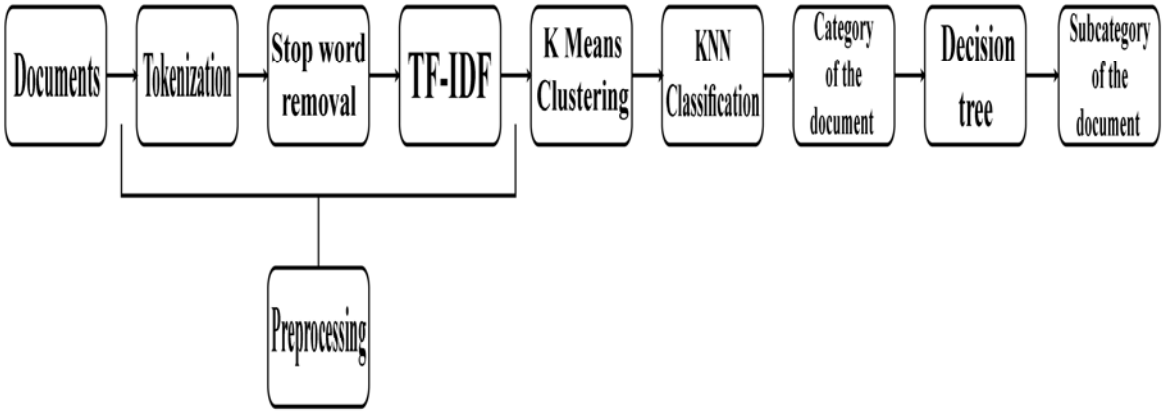

Fig. 2. Proposed methodology

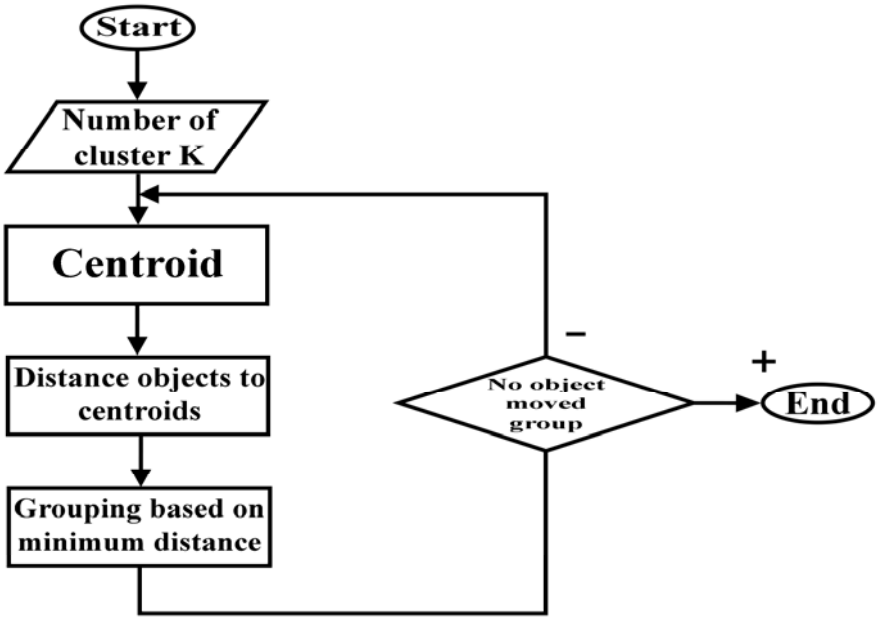

Fig. 3. Process flow of K-means algorithm

K-means clustering algorithm aims at minimizing an 'objective function'. In this process, it is a 'squared error function'. The objective function can be presented as:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 \quad (4)$$

Where $\left\| x_i^{(j)} - c_j \right\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster center c_j, is an indicator of the distance of the n data points from their respective cluster centers.

1) K-means clustering algorithm:

The steps of the algorithm are as follows:

a) Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.

b) Assign each object to the group that has the closest centroid.

c) When all objects have been assigned, recalculate the positions of the K centroids.

Repeat Steps b and c until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

2) Bisecting K-means algorithm

Bisecting K-means algorithm starts with a single cluster of all the documents. The steps of the algorithm as mentioned in [8] are as follows:

a) Pick a cluster to split.

b) Find 2 sub-clusters using the basic K-means algorithm.

c) Repeat step 2, the bisecting step, for a fixed number of times and take the split that produces the clustering with highest overall similarity. (For each cluster, its similarity is the average pair wise document similarity, and we seek to minimize that sum over all clusters.)

d) Repeat steps a, b and c until the desired number of clusters is reached.

B. KNN classification

K Nearest Neighbor is one of the simplest classification algorithms. It is a supervised learning algorithm. The category of the query document is determined based on the category of the K nearest documents in the document space. The feature vector is created from the documents in the training set. Also a weighting scheme such as TF-IDF is also used to reduce the dimension of the feature vector. The feature vector of the query

document is determined and it is compared with the document vector of each document in the training set. Then K closest matching samples are selected as shown in Fig. 4 and based on the maximum matching category, the query document is classified.

In the Fig. 4, C1 and C2 represent document class and D1 and D2 represents the test documents. It is a 4-NN method in which each test document is compared to 4 neighboring document class and based on maximum matching, the document class is assigned.

1) KNN classification algorithm

The KNN algorithm does not require any prior knowledge about the training data set for classification. On the basis of similarity of the documents, the algorithm performs the classification process [5]. Suppose we have a query document X. The process of KNN classification algorithm to classify this document is as follows [6] [7]:

Suppose C_1, $C_2 \ldots C_j$ are the j training categories. N is the total number of the training samples. The m-dimension feature vector is obtained after preprocessing (tokenization, stop word removal) and applying weight to each term by using TF-IDF method. Then,

a) Obtain a feature vector of same dimension as that of the training samples from the document X in the form ($X_1, X_2 \ldots X_m$).

b) Calculate the similarities between all training samples and document X. The similarity between *i*th document d_i ($d_{i1}, d_{i2} \ldots d_{im}$) is as follows:

$$SIM(X, d_i) = \frac{\sum_{j=1}^{m} X_j \cdot d_{ij}}{\sqrt{\left(\sum_{j=1}^{m} X_j\right)^2} \sqrt{\left(\sum_{j=1}^{m} d_{ij}\right)^2}} \tag{5}$$

c) Choose k samples which are larger from N similarities of $SIM(X, d_i)$, $(i = 1,2,\ldots,N)$, and treat them as a KNN collection of X. Then calculate the probability of X belong to each category respectively with the following formula.

$$P(X, C_j) = \sum_{d_i \in KNN} SIM(X, d_i) \cdot y(d_i, C_j) \tag{6}$$

Where, $y(d_i, C_j)$ is a category attribute function, which satisfies

$$y(d_i, C_j) = \begin{cases} 1, d_i \in C_j \\ 0, d_i \notin C_j \end{cases}$$

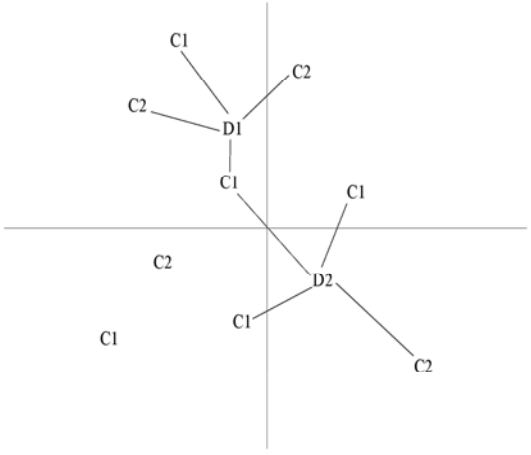

Fig. 4. KNN classification

d) Judge document X to be the category which has the largest P (X, C_j).

2) Distance metric and normalization

The performance of KNN classifier is dependent on the value of K and also on the distance metric used [5] [8]. The distance of the K^{th} nearest neighbor determines the radius of the local region. Different values of K changes the probability of a document to be in a specific class. Too small or too large value of K will adversely affect the classification process. Data sparseness and noisy, ambiguous or mislabeled points are the main problems if K value is chosen to be very small. Again, Outlier points within the neighborhood from other classes are the main problem if very large value of K is chosen. To find the nearest K points, different distance metrics are used. Euclidean distance, Minkowski distance and Manhattan distance are some of the examples of distance metrics used [5].

To find Euclidean distance between two points *X* and *Y:*

$$\sqrt{\sum_{1\le i\le n}(x_i - y_i)^2} \tag{7}$$

The Minkowski distance between the two points *X* and *Y* is given below:

$$\sum_{1\le i\le n}|x_i - y_i| \tag{8}$$

Again, the Manhattan distance between two points *X* and *Y* is given below:

$$\left(\sum_{1\le i\le n}|x_i - y_i|^p\right)^{1/p} \tag{9}$$

Where p is a positive integer.

For simplicity in calculations, all the attribute values of the data set should be normalized before finding the Euclidean distance. Binary values are assigned to all the attributes by normalization. 1 represents the highest value and 0 represents the lowest value. The min-max normalization of an attribute A is calculated as:

$$v' = (v - \min_A)/(\max_A - \min_A) \tag{10}$$

Here, v is the value of A which is to be normalized in the range [0, 1]. Again, $\max_A$ and $\min_A$ are the maximum and minimum values of A respectively.

3) Variants of KNN

There are different variants of KNN classification algorithm present. Mainly the variants can be divided into two broad parts as mentioned in [5]. The first category tries to alter the result influencing factors that may improve the performance. For examples, selecting optimum K value, use of distance metric, assigning weights to the attributes value etc. In the second category, the concept of evolutionary computing is introduced. The KNN classification algorithm is used with the evolutionary computing techniques to give a hybrid classification technique.

In this paper, we used a combination of clustering and classification techniques to increase the efficiency of the process and also to improve the accuracy. Here, K-means algorithm is used for the clustering and KNN for the classification process.

C. Decision tree

In this paper, KNN classification algorithm is used to classify a set of documents into their respective categories and also to classify a query document based on the training data set. For example, suppose there is a set of documents with categories sports, business, news, entertainment etc. Then KNN classification algorithm is used to classify each document and also a query document after the training phase is completed. But the sports articles can also have articles of different sports such as cricket, football, rugby etc. Again, entertainment articles can also be of different categories such as movies, music, computer games etc. These are sub-categories of the main classes. To classify a query document into its respective sub-category, a decision tree classifier is used in this paper. The category of the document will first be found out with the help of KNN classifier and then using the decision tree, the sub-category of the document will be determined.

Decision tree is an inductive learning method. The algorithm works well with noisy data set and also has capability to learn disjunctive expressions. There are different algorithms for decision tree classification. ID3, C4.5 and C5 are the most popular decision tree algorithms. All the decision tree algorithms follow a top down

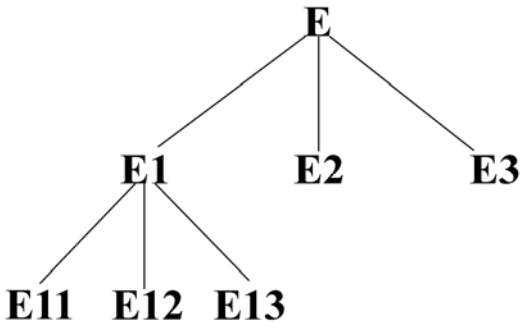

Fig. 5. Decision tree

approach. Based on the information gain, attributes are selected which have highest values at each level. In a decision tree, the leaf nodes represent the category of the documents and the branches show the parameters that leads the query document to its respective category. A query document will be put in the root node of the document tree and then based on the branch parameters; it will reach a specific leaf which shows the category of the document. An example of a decision tree is shown in Fig. 5.

1) C4.5 algorithm:

C4.5 is a successor of ID3 algorithm to prepare a decision tree. This algorithm was developed by Ross Quinlan [9]. The decision tree generated by C4.5 algorithm can be used as a statistical classifier. The training set used for the classification contains pre-classified samples. Each sample is an m-dimensional vector. Suppose S (s1, s2…) is the training set consists of samples which are already classified. Each sample si is represented as an m-dimensional vector (x1i, x2i…xmi), where xj represents attribute values or features of the sample, as well as the class in which si falls. Based on the difference of entropy the splitting occurs. The attribute having maximum information gain is selected at each level of the tree.

The algorithm has a few base cases:

- All the samples in the list belong to the same class. When this happens, it simply creates a leaf node for the decision tree saying to choose that class.
- None of the features provide any information gain. In this case, C4.5 creates a decision node higher up the tree using the expected value of the class.
- Instance of previously-unseen class encountered. Again, C4.5 creates a decision node higher up the tree using the expected value.

In C4.5 algorithm, a few improvements have been made to its predecessor ID3 algorithm. Some of these improvements are:

- In C4.5, both continuous and discrete attributes are used. The continuous attributes are handled using a threshold value. The list is then divided into two parts; one which has the values above the threshold value and another which has values equal or less than the threshold value.
- The attributes which have missing values can also be handled in C4.5 algorithm. The missing attributes are represented as '?' and is not considered for gain and entropy calculations.
- The attributes are handled with different costs for different attributes.
- Pruning trees after creation - C4.5 goes back through the tree once it's been created and attempts to remove branches that do not help by replacing them with leaf nodes.

VI. CONCLUSION

Text mining or text categorization has gained much popularity in recent years due the rapid growth in the electronic documents and internet. There are many classification algorithms have been developed over the years. But it can be observed that K nearest neighbor algorithm has shown better results than other algorithms when carried out with proper preprocessing methods. Also, it can show better results with noisy datasets. The calculation complexity being the only disadvantage, it can also be reduced by combining the traditional KNN with some clustering methods. In this paper, a combination of clustering and classification method is proposed for classifying a set of documents to their respective categories. Again, to find the sub-categories of the query document, a decision tree algorithm is used as classifier. There are different variants of these algorithms which

can be effectively used for classification. Thus, an efficient classification algorithm can improve the accuracy of text mining process.

ACKNOWLEDGMENT

I, Hrishikesh Deka, student of M.Tech 4th Semester, Department of Information Technology, Gauhati University, have submitted my review paper entitled "A Machine learning approach for text and document mining" under the guidance of Parismita Sarma, Assistant Professor, Department of Information Technology, G.U. I would like to express my sincere gratitude and appreciation to all those who gave me the possibility to complete this paper. Special thanks to my guide Parismita Sarma Madam, our Head of the Department Mr. Mirzanur Rahman sir, and all the teachers and researchers related to this paper, whose stimulating suggestions, coordination and encouragement helped me a lot.

REFERENCES

[1] Bijalwan V, Kumar V, Kumari P and Pascual J, "KNN based Machine Learning Approach for Text and Document Mining," *International Journal of Database Theory and Application*, vol. 7, No. 1, pp. 61-70, 2014

[2] Khan A, Baharudin B, Lee LH, Khan K, "A review of Machine Learning algorithms for text documents classifications," *Journal of advances in information technology*, vol. 1, no. 1, February 2010.

[3] Dr. Vijayarani S, Ms. Ilamathi J, Ms. Nithya, "Preprocessing Techniques for Text Mining – An Overview," *International Journal of Computer Science & Communication Networks*, vol. 5(1).

[4] Steinbach M, Karypis G, Kumar V, "A Comparison of Document Clustering Techniques," Department of Computer Science / Army HPC Research Center, University of Minnesota, 2000.

[5] Lamba A, Kumar D, "Survey on KNN and It's Variants," *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 5, Issue 5, May 2016.

[6] Yong Z, Youwen L and Shixiong X, "An Improved KNN Text Classification Algorithm Based on Clustering," *Journal of computers*, vol. 4, no. 3, March 2009.

[7] Buana PW, Jannet S D.R.M., I Ketut Gede Darma Putra, "Combination of K-Nearest Neighbor and K-Means based on Term Re-weighting for Classify Indonesian News," *International Journal of Computer Applications (0975 – 8887)*, Volume 50 – No.11, July 2012.

[8] Imandoust SB, Bolandraftar M, "Application of K-Nearest Neighbor (KNN) Approach for Predicting Economic Events: Theoretical Background," *Int. Journal of Engineering Research and Applications*, Vol. 3, Issue 5, Sep-Oct 2013, pp.605-610.

[9] Quinlan JR, "C4.5:Programs for machine learning," Morgan Kaufmann Publishers,1993.

[10] Li YH and Jain AK, "Classification of Text Documents," *The Computer Journal*, 41(8), pp.537-546, 1998.

[11] Bhumika, Prof Sehra SS, Prof Nayyar A, "A review paper on algorithms used for text classification," *International Journal of Application or Innovation in Engineering & Management (IJAIEM)*, vol. 2, Issue 3, March 2013.

[12] R Shrihari C, Desai A, "A review on knowledge discovery using text classification techniques in text mining," *International Journal of Computer Applications (0975 – 8887)*, vol. 111 – No 6, February 2015.

[13] Trstenjak B, Mikac S, Donko D, "KNN with TF-IDF Based Framework for Text Categorization," *24th DAAAM International Symposium on Intelligent Manufacturing and Automation*, 2013.