An Algorithmic Approach of Keyword Extraction based Text Document Classification

Yoganand.C.S[1], Vadivel.R[2]
PG Student of CSE[1], Assistant Professor of CSE[2]
Adithya Institute of Technology, Coimbatore[1, 2]
info.yoganand@gmail.com[1],vadivelcse@gmail.com[2]

Abstract-The various institutions and industries are converting their documents into electronic text files. The documents may contains applications, personal documents, properties documents etc. The categorization of the text documents are really makes a very big issue. In this paper we propose the various techniques for the document classification process. These documents may be in the form of supervised, unsupervised or semi-supervised documents. The supervised documents are the standard documents which are contains the proper format of data. They can be classified by using the Naïve Bayes model with the help Hidden Markov Model (HMM). The keyword and key Phrases are extracted and used as a training set for the further document classification along with the training dataset. The keyword extraction can be done based on the Word count method and Porter stemming algorithms. Further documents can be classified using Naïve Bayes and Support Vector Machine (SVM), Subspace, Decision Tree and k-Nearest Neighbor (k-NN) methods.

Index Terms-Support Vector Machines (SVM), Hidden Markov Model (HMM), k-Nearest Neighbor (k-NN), Text categorization, mapping models.

1. INTRODUCTION

Nowadays all institutions and private companies keep their files in electronic format in order to reduce the paperwork and, at the same time provide instant access to the information contained. Document clustering and classification in one of the most important text mining methods that are developed to help users effectively navigate, summarize and organize text documents. Document classification can be defined as the task of automatically categorizing collections of electronic documents into their annotated classes based on their contents [3]. Recent years, this has become important due to the advent of large amounts of data in digital form. Document classification in the form of text classification systems have been widely implemented in numerous applications such as spam filtering, emails categorizing, directory maintenance and plagiarism checking processes [6].

Data mining is useful in discovering implicit, potentially valuable information or knowledge and previously unknown from large datasets. Text Document classification denotes the test of assigning raw text documents to one or more pre-defined categories [2]. This is a direct concept from machine learning, which denotes the declaration of a set of labelled categories as a way to represent the documents, and a text classifier trained with a labelled training set. Among these approaches, Bayesian classification has been widely implemented in many real world applications due to its relatively simple training and clustering algorithms [7].

The concept of text categorization is the classification of documents into a fixed number of predefined categories or classes [4]. Each document can be classified into exactly one or more category automatically. Some of the documents are not classified into any category [5]. This is known as supervised learning problem. Since categories may overlap, each category is treated as a separate binary classification problem.Each of the document classification schemes previously mentioned has its own unique properties and associated problems. The decision tree induction algorithms and the rule induction algorithm are simple to understand and interpret the classification [1]. However, these algorithms do not work well when the number of distinguishing features between documents is large. The k-NN algorithm is easy to implement and shows its effectiveness in a variety of problem domains [8]. As a trade-off to its simplicity, Bayesian classification has been reported as one of the poorest-performing classification approaches by many research groups through extensive experiments and evaluations [7].

The paper is organized as follows: section 1 deals with Introduction of my paper. Section 2 includes the literature survey of my concept. Section 3 explains related work, Section 4 gives brief explanation about proposed system, system architecture and workflow of the project. Section 5 contains Experimental setup details and Section 6 contains the Performance analysis, results graph of the document classification and so on.

2. LITERATURE SURVEY

2.1. *Naïve Bayes Model for Textclassification*

Naïve Bayes classifiers which are widely used for text classification in machine learning are based on the conditional probability of features measures belonging to a class, feature selection methods are used for feature selection. An auxiliary feature method is used for text classification [3]. The auxiliary feature is chosen from collection of features which is determined by using existing feature selection method. To improve classification accuracy the corresponding conditional probability is adjusted [5]. The feature with auxiliary feature was found and the probability of the feature with auxiliary feature was adjusted after feature selection.

2.2. *Support Vector Machine*

The application of Support vector machine (SVM) method is the Text Classification. The SVM need both positive and negative training set which are uncommon for other classification methods [11]. These negative and positive training set are needed for the SVM to seek for the decision surface that best separates the positive from the negative data in the n dimensional space, is called as hyper plane [16]. SVM classifier method is outstanding from other with its effectiveness to improve performance of text classification combining the HMM and SVM where HMMs are used to as a feature extractor and then a new feature vector is normalized as the input of SVMs, so unknown texts are successfully classified based on the trained SVMs, also by combing with Bayes use to reduce number of feature which as reducing number of dimension [13].

2.3. *Decision Tree*

When decision tree is used for text classification it consist tree internal node are label by term, branches departing from them are labelled by test on the weight, and leaf node are represent corresponding class labels .Tree can classify the document by running through the query structure from root to until it reaches a particular leaf, which represents the goal for the classification of the text document [9]. The decision tree classification method is outstanding from other decision support tools with several advantages like its simplicity in interpreting and understanding, even for non-expert users [14]. So it is only used in some applications processes.

2.4. *Decision Rule*

Decision rules classification method uses the rule-based inference to classify documents to their annotated categories. A popular format for interpretable solutions is the Disjunctive Normal Form (DNF) model [19]. A classifier for category ci built by an inductive rule learning method consists of a DNF rule. In the case of handling a dataset with large number of features for each category, strict implementation is recommended to reduce the size of rules set without affecting the performance of the classification [16]. The presents a hybrid method of rule based processing and back-propagation neural networks for spam filtering.

2.5. *Term Frequency/Inverse Document Frequency (TF-IDF)*

A new improved term frequency/inverse document frequency (TF-IDF) approach which uses confidence, support and characteristic words to enhance the recall and precision of text classification. Synonyms defined by a lexicon are processed in the improved TF-IDF approach [17]. It need to find the best matching category for the text document. The term (word) frequency/inverse document frequency (TF-IDF) approach is commonly used to weigh each word in the text document according to how unique it is. In other words, the TF-IDF approach captures the relevancy among words, text documents and particular categories [1]. It put forward the novel improved TF-IDF approach for text classification, and will focus on this approach in the remainder of this paper, and will describe in detail the motivation, methodology, and implementation of the improved TF-IDF approach.

3. RELATED WORK

The Naïve Bayes text document classification will be depends only on the Bayesian rule. The Bayesian rule is given below:

$$P(D_1|D_2) = (P(D_2|D_1) * P(D_1)) / P(D_2) \qquad (1)$$

where $D_1$and $D_1$are the two documents or two constraints of Bayesian rule.

The probability of D_1 happening given $D_2$is determined from the probability of D_2given D_1, the probability of D_1 occurring and the probability of D_2. The Bayes Rule enables the calculation of the likelihood of event D_1 given that D_2 has happened. This is used in text classification to determine the probability that a document $D_2$is of type D_1 just by looking at the frequencies of words in the document. You can think of the Bayes Rule as showing how to update the probability of event D_1 happening given that you've observed D_2.

A category is represented by a collection of words and their frequencies; the frequency is the number of times that each word has been seen in the documents used to train the classifier.Suppose there are n categories C_0 to C_{n-1}. Determining which category a document D is most associated with means calculating the probability that document D is in category C_i, written $P(C_i|D)$, for each category C_i.Using the Bayes Rule, you can calculate $P(C_i|D)$ by computing:

$$P(C_i|D) = (P(D|C_i) * P(C_i)) / P(D) \quad (2)$$

$P(C_i|D)$ is the probability that document D is in category C_i; that is, the probability that given the set of words in D, they appear in category C_i. $P(D|C_i)$ is the probability that for a given category C_i, the words in D appear in that category.$P(C_i)$ is the probability of a given category; that is, the probability of a document being in category C_i without considering its contents. P(D) is the probability of that specific document occurring.

To calculate which category D should go in, you need to calculate $P(C_i|D)$ for each of the categories and find the largest probability. Because each of those calculations involves the unknown but fixed value P(D), you just ignore it and calculate:

$$P(C_i |D) = P(D|C_i) * P(C_i) \quad (3)$$

P(D) can also be safely ignored because you are interested in the relative not absolute values of $P(C_i|D)$, and P(D) simply acts as a scaling factor on $P(C_i|D)$.D is split into the set of words in the document, called W_0 through W_{m-1}. To calculate $P(D|C_i)$, calculate the product of the probabilities for each word; that is, the likelihood that each word appears in C_i. Here's the "naïve" step: Assume that words appear independently from other words (which is clearly not true for most languages) and $P(D|C_i)$ is the simple product of the probabilities for each word:

$$P(D|C_i) = P(W_0|C_i) * P(W_1|C_i) * ... * P(W_{m-1}|C_i) \quad (4)$$

For any category, $P(W_{j|}C_i)$ is calculated as the number of times W_j appears in C_i divided by the total number of words in C_i. $P(C_i)$ is calculated as the total number of words in C_i divided by the total number of words in all the categories put together. Hence, $P(C_i|D)$ is:

$$P(W_0|C_i) * P(W_1|C_i) * ... * P(W_{m-1}|C_i) * P(C_i) \quad (5)$$

for each category, and picking the largest determines the category for document D.

The documents are also classified by using the SVM. They use hyperplane method for classification. The hyperplane can be estimated using the same word count and occurrences of words in the document. Hyperplane can classify the text documents based on the weight or term frequency values of the documents. The Multi-Layer Perceptron (MLP) model is used for SVM classification.

4. PROPOSED WORK

The Naïve Bayes and Support Vector Machine algorithms are used for document classification. The keywords can be extracted from documents using porter-stemming algorithm. Fig.1 shows the workflow diagram on the proposed system. The extracted keywords and key-phrases are used as the training set for further document classification. The TF-IDF method is used to calculate the probability of word occurrence in each document. TF is the Term Frequency, it helps to calculate the occurrence of word in each page of the document, IDF is the Inverse Term Frequency, and it helps to calculate the complete word occurrence count in whole documents. The preprocessing steps involved in document classifications are stop words removal and stemming methods.

4.1. *Stop Word Removal*

This is the first step in preprocessing which will generate a list of terms that describes the document satisfactorily. The document is parsed through to find out the list of all the words. The next process in this step is to reduce the size of the list created by the parsing process, generally using methods of stop words removal and stemming. The stop words removal accounts to 20% to 30% of total words counts while the process of stemming reduce the number of terms in the document. Both the process helps in improving the effectiveness and efficiency of text processing as they reduce the indexing file size.Stop words are removed from each of the document by comparing the with the stop word list. This process reduces the number of words in the document significantly since these stop words are insignificant for search keywords. Stop words can be pre-specified list of words or they can depend on the context of the corpus.

4.2. *Stemming*

The next process in phase one after stop word removal is stemming. Stemming is process of linguistic normalization in which the variant forms of a word is reduced to a common form. For example: the word, connect has various forms such as connect, connection, connective, connected, etc., Stemming process reduces all these forms of words to a normalized word connect. Porter's English stemmer algorithm is used to stem the words for each of the document in our stemming process.

4.3. *Feature Selection*

Feature selection is the process of selecting a subset of the terms occurring in the training set and using only this subset as features in text classification. Feature selection serves two main purposes. First, it makes training and applying a classifier more efficient by decreasing the size of the effective vocabulary. This is of particular importance for classifiers that, unlike NB, are expensive to train. Second, feature selection often increases classification accuracy by eliminating noise features. A *noise feature* is one that, when added to the document representation, increases the classification error on new data. Here the Information Gain (IG). These features are selected based on the frequency measurement of the document.

4.4. *Document Representation*

A term-document matrix can be encoded as a collection of n documents and m terms. An entry in the matrix corresponds to the "weight" of a term in the document; zero means the term has no significance in the document or it simply doesn't exist in the document. The whole document collection can therefore be seen as a m x n-feature matrix A (with m as the number of documents) where the element a_{ij} represents the frequency of occurrence of feature **j** in document **i.** This was of representing the document is called term-frequency method. The most popular term weighting is the Inverse document frequency, where the term frequency is weighed with respect to the total number of times the term appears in the corpus. There is an extension of this designated the term frequency inverse document frequency (tf-idf). The formulation of tf-idf is given as follows:-

$$W_{ij} = tf_{i,j} * \log (N / df_i)$$

where W_{ij} is the weight of the term **i** in document **j**, $tf_{i,j}$ = number of occurrences of term **i** in document **j**, N is the total number of documents in the corpus, df_i = is the number of documents containing the term **i**.

Then the Naïve Bayes and Support Vector Machine classification algorithms are applied one by one for the document classification. The classification results will be calculated and compared for both the methods. Finally it produces the better performance on classification accuracy. The k-NN technique helps to cluster or group the classified documents into the proper category. These are all done by using the various steps for classification.

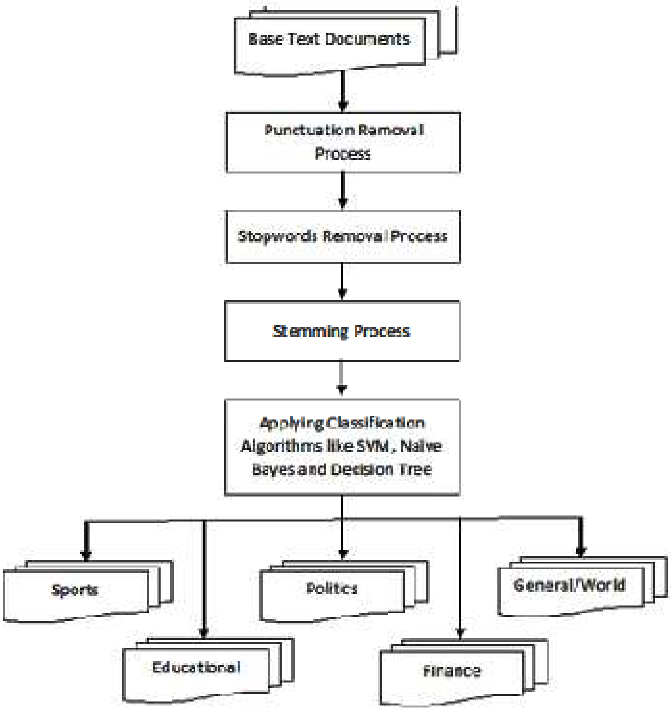
Base Text Documents
Punctuation Removal Process
Stopwords Removal Process
Stemming Process
Applying Classification Algorithms like SVM , Naïve Bayes and Decision Tree
Sports
Politics
General/World
Educational
Finance

Fig. 1. System Architecture

Finally the supervised and unsupervised document classification accuracy will increased by using these various classification algorithms based on keyword extraction processes.

5. EXPERIMENTAL SETUP

To set a benchmark for Document Classification and allow for comparisons of other methods with the proposed approach in this paper. The experiments are performed on Intel Core i -3 3210, 2.3 GHz processor and 4 GB RAM with Windows7 as an operating system and the experiments are implemented in Microsoft visual studio 2008 using C#.Netwith MS SQL server and Java for document analysis and classification process.

To measure the effectiveness of the classification this approach can be applied and verified based on various datasets. The various collections of datasets are tabulated below with their topics. It contains the total of 814 documents belonging to seven different classes (business (B), entertainment (E), health (H), international (I), politics (P), sports (S) and technology (T)) used for training and two test data sets (news items at different time intervals, see Table 1).The Reuters 21578 dataset is used as training dataset for document classification process.

Table 1. Training and Test Datasets

	Categories	**B**	**E**	**H**	**I**	**P**	**S**	**T**
Training Data	No. of documents	130	133	91	110	130	130	90
	Total no. of terms	1848	2048	1213	1974	2070	1659	1364
Test data Set1	No. of documents	110	111	79	80	110	111	79
	Total no. of terms	2155	2583	1535	1999	2439	1952	1618
Test Data Set2	No. of documents	100	101	78	70	101	101	70
	Total no. of terms	2046	2834	1803	2604	2070	1974	1689

The Table 2 contains the comparison of four algorithms. The four algorithms are Naïve Bayes (NB), Nearest Neighbor (NN), Decision Tree (DT) and Sub Space (SS) model. The various algorithm classification measures are tabulated.The NB algorithm performs better on test data set 1. The recognition rate is 83.1%. The SS algorithm performance good on the Test Data Set 2.

Table 2. Comparison of Four algorithms

Test Data's		**NB**	**NN**	**DT**	**SS**
Test data set1	No. of misclassifications	115	165	178	139
	Recognition rate (%)	**83.1**	75.7	73.8	79.6
Test data set2	No. of misclassifications	125	179	144	111
	Recognition rate (%)	79.87	71.18	76.8	**82.13**

The Table 3 contains of IDF calculation of documents based on the total number of documents(N) and Document frequency (n). The IDF value is also based on the Number of Categories (C) of the documents. The IDF is the Inverse Document Frequency of the document. The IDF is calculated based on the probability of the Document Frequency.

Table 3 IDF Calculation

Total No of Documents (N)	12	No. of Categories (C)			2
Document Frequency (n)	7	5	3	3	6
IDF log $\left(\frac{N}{n}\right)$	0.2341	0.4771	0.6021	0.6021	0.3010

6. PERFORMANCE ANALYSIS

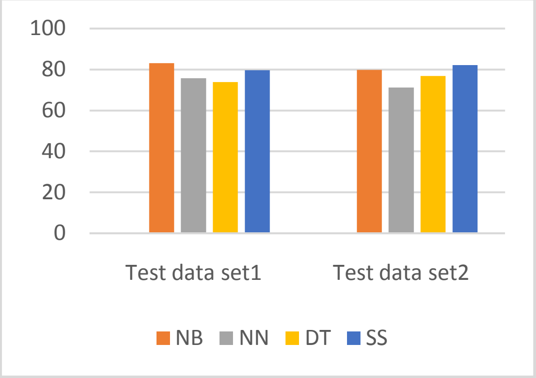

Fig.2. Best case and Average Case of Classification

The performance analyses of different algorithms are discussed in detail and explain in graph as follows. The Fig. 2.Can be plotted based on the Table 2 values. In x- Axis the number of datasets and various algorithms are taken. In y-Axis the no of documents are taken. The Fig.3 expose the best case and average case performance accuracy of the document classification. The Fig. 4 exposes the worst case of document classification measure.

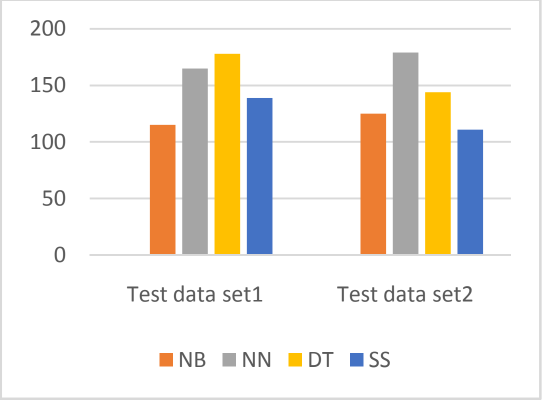

Fig.3. Worst Case of Classification

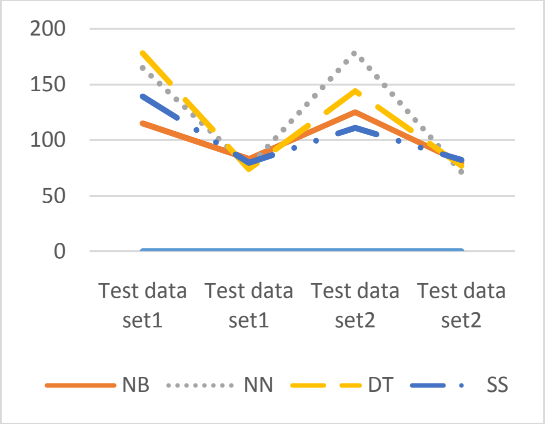

Fig. 4. Chart for Algorithm Comparison

7. CONCLUSION

In this paper a system has been proposed for the organization of a document in terms of content. It comprises two stages where first one is the extraction of keywords and key phrases for the document classification, second one is the process of classifying the documents based on the keywords and training dataset. This content based classification is applicable in the plagiarism processing and in the machine learning processes. In future by combining these different algorithms to improve the classification accuracy. The future enhancement idea includes the WordMap creation based on the relationships between the keywords. It also improves the classification performance.

REFERENCES

[1] Bhamidipati. N. L and Pal. S. K, “Stemming via distribution-based word segregation for classification and retrieval,” IEEE Trans. Syst., Man, Cybern. B, Cybern, vol. 37, no. 2, pp. 350–360, Apr. 2007.

[2] Chakrabarti.S, Roy.S, and Soundalgekar.M.V, “Fast and Accurate Text Classification via Multiple Linear Discriminant Projection,” VLDB J., Int’l J. Very Large Data Bases, pp. 170-185, 2003.

[3] Cunningham.P, Nowlan.N, Delany.S.J, and Haahr.M, “A Case-Based Approach in Spam Filtering that Can Track Concept Drift,” Proc. ICCBR Workshop Long-Lived CBR Systems, 2003.

[4] Dino Isa, Lam Hong Lee, V.P. Kallimani, and R. RajKumar, “Text Document Preprocessing with the Bayes Formula for Classification Using the Support Vector Machine” in IEEE Transactions on Knowledge and Data Engineering, Vol. 20, No. 9, September 2008

[5] Han.E.H, Karypis.G, and Kumar.V, “Text Categorization Using Weight Adjusted k-Nearest Neighbor Classification,” Dept. of Computer Science and Eng., Army HPC Research Center, Univ. of Minnesota, 1999.

[6] Hartley.M, Isa.D, Kallimani.V.P, and Lee.L.H, “A Domain Knowledge Preserving in Process Engineering Using Self-Organizing Concept,” technical report, Intelligent System Group, Faculty of Eng. and Computer Science, Univ. of Nottingham, Malaysia Campus, 2006.

[7] J.G.Liang, X.F.Zhou, P.Liu, L.Guo, S.Bai “An EMM-based Approach for Text Classification” in Procedia Computer Science 17, 506 – 513, 2013

[8] Kerner. Y.H, Gross.Z, and Masa.A, Automatic extraction and learning of key phrases from scientific articles. In Computational Linguistics and Intelligent Text Processing, pages 657–669, 2005.

[9] Lam Hong Lee, Dino Isa, Wou Onn Choo, Wen Yeen Chue, “High Relevance Keyword Extraction facility for Bayesian text classification on different domains of varying characteristic“ in Expert Systems with Applications 39, 1147–1155, 2012

[10] Matsuo. Y and Ishizuka. M, Keyword extraction from a single document using word co-occurrence statistical information. International Journal on Artificial Intelligence, 13(1):157–169, 2004.

[11] McCallum.A and Nigam.K, “A Comparison of Event Models for Naïve Bayes Text Classification,” J. Machine Learning Research 3, pp. 1265-1287, 2003.

[12] Sang-Bum Kim, Kyoung-Soo Han, Hae-Chang Rim, and Sung Hyon Myaeng, “Some Effective Techniques for Naive Bayes Text Classification” in IEEE Transactions on Knowledge and Data Engineering, Vol. 18, No. 11, November 2006.

[13] Wang.J, Yao.Y, and Liu. Z. J, “A new text classification method based on HMM-SVM,” in Proc. Int. Symp. Commun. Inf. Technol., Sydney, N.S.W., Australia, Oct. 17–19, 2007, pp. 1516–1519.

[14] Wei Zhang, Feng Gao, “An Improvement to Naive Bayes for Text Classification” in Procedia Engineering 15, 2160 – 2164, 2011

[15] Wen Zhang, Taketoshi Yoshida, Xijin Tang, “Text classification based on multi-word with support vector machine” in Knowledge-Based Systems 21, 879–886, 2008.

[16] Yang.Y, “An evaluation of statistical approaches to text categorization,” J. Inf. Retrieval, vol. 1, no. 1/2, pp. 69–90, 1999.