International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 5 Volume No.6 Aug-Sept – 2022
DOI:10.46647/ijetms.2022.v06i05.137 ISSN: 2581-4621

CLASSIFICATION OF LARGE DOCUMENTS USING MACHINE LEARNING TECHNIQUES

Kandimalla Gopi[1] ,Goli Sushma[2],Srinivasa Rao Vallepu[3]
Assistant professor, Guru Nanak Institutions Technical Campus[1,2,3]
Ibrahimpatnam, Telangana,India

Abstract: Document classification is a growing interest in the research of text mining. Correctly identifying the documents into particular category is still presenting challenge because of large and vast amount of features in the dataset. In regards to the existing classifying approaches, Naive Bayes is potentially good at serving as a document classification model due to its simplicity. The aim of this paper is to highlight the performance of employing Naive Bayes in document classification. Results show that Naive Bayes is the best classifiers against several common classifiers (such as decision tree, neural network, and support vector machines) in term of accuracy and computational efficiency.

Keywords: **Classification, Naive bayes, features**

I. INTRODUCTION

With the explosive growth of the textual information from the electronic documents and World Wide Web, proper classification of such enormous amount of information into our needs is a critical step towards the business success. Recently, numerous research activities have been conducted in the field of document classification, particularly applying in spam filtering, emails categorization, website classification, formation of knowledge repositories, and ontology mapping. However, it is time-consuming and labor intensive for a human to read over and correctly categorize an article manually. Attempts to address this challenge, automatic document classification studies are gaining more interests in text mining research recently. Consequently, an increasing number of approaches have been developed for accomplishing such purpose, including k-nearest-neighbor (KNN) classification, Naive Bayes classification, support vector machines (SVM), decision tree (DT) , neural network (NN), and maximum entropy.

Among these approaches, the Naive Bayes text classifier has been widely used because of its simplicity in both the training and classifying stage. Although it is less accurate than other discriminative methods (such as SVM), numerous researchers proved that it is effective enough to classify the text in many domains. Naive Bayes models allow each attribute to contribute towards the final decision equally and independently from other attributes, in which it is more computational efficient when compared with other text classifiers. Thus, the present study focuses on employing Naive Bayes approach as the text classifier for document classification and thus evaluates its classification performance against other classifiers.

II. PROPOSED METHODOLOGY

In order to generate a document classifier model, Indeed, some data are useless (i.e. do not affect the classification result even removing them, such as stop words) and some carries similar meanings (i.e. the term “bank” and “banks”), therefore a preprocessing phase has been to conduct first. In this way, the dataset can be more precise. After the data preprocessing phase, critical attributes have to be selected. In this study, critical means the importance of such attribute towards the solution class. For example, the term “bank” categorized in “business” class has the highest score in term of term frequency, therefore it is analyzed that “bank” is one of the critical attributes to represent the documents fell in the “business”

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 5 Volume No.6 Aug-Sept – 2022
DOI:10.46647/ijetms.2022.v06i05.137 ISSN: 2581-4621

class. Thus, less important features can be removed and so the computational time can be improved significantly.

As for the classification phase, different classifiers (such as SVM, NN, and DT) are employed to generate the model. However, this study only focused on using Naive Bayes to classify the documents. Given the probabilistic characteristic of Naive Bayes, each training document is vectorized by the trained Naive Bayes classifier through the calculation of the posterior probability value for each existing. Finally, the model is evaluated by a set of testing data. In order to test the classification ability of the model, several evaluation measures (such as precision, recall, and F-measure) are adopted. Furthermore, to interpret whether Naive Bayes is best to use as the classifier, its testing result will be compared with other classifiers results as well.

PHASE 1: DATA PREPROCESSING

It is common to find that several attributes are useless (such as the word "a", "the", etc.). Thus, stopword removing algorithm has been applied. To initialize the algorithm, a set of stopword (such as a, a's, able, about, above, according, accordingly, and across) has set by the human beforehand and hence stored in a text file. Then, the model can simply match the attributes with those preset stopword. After the stopword algorithm, a missing data checking algorithm is adopted. This algorithm is used to identify any missing data and hence interpret a value to it since data mining cannot perform under missing data situation. The third algorithm applied in the preprocessing phase is the stemming. Since some words carry similar meanings but in different grammatically form (such as "bank" and "banks"), therefore it is needed to combine them into one attribute. In this way, the documents can show a better representation (with stronger correlations) of these terms and even the dataset can be reduced for achieving faster processing time.

PHASE 2: FEATURE SELECTION

Feature selection is one of the most important preprocessing steps in data mining. It is an effective dimensionality reduction technique to remove noise feature. In general, the basic idea of feature selection algorithm to searches through all possible combinations of attributes in the data to find which subset of features works best for prediction. Thus, the attribute vectors can be reduced in number by which the most meaningful ones are kept and the irrelevant or redundant ones are removed and deleted.

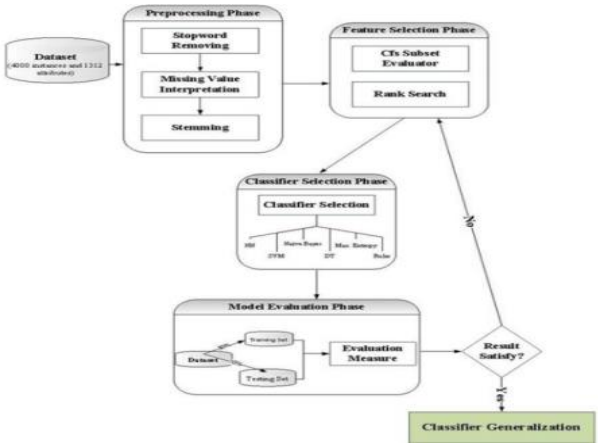

Figure 2.1 :Proposed Methodology

In this study, all the documents in the training data are categorized into four different categories in which the model can simply compute which terms are frequently occurred in such category. In this way, some unless or irrelevant attributes can be filtered out. As discussed by the study, Cfs Subset Evaluator is the best method to get final feature set; whereas rank search or random search is suggested to have a good feature set. Thus, in this study, Cfs Subset Evaluator and rank search are applied as the feature selection algorithm.

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 5 Volume No.6 Aug-Sept – 2022
DOI:10.46647/ijetms.2022.v06i05.137 ISSN: 2581-4621

For example, Table 2.2 summarizes the feature selected by applying the Cfs Subset Evaluator and rank search (with Gain Ratio metric).

Selected Features	administr, admit, agre, bn, britain, busi, capit, chairman, chancellor, chief, ci, company, concern, country, custom, decis, expect, financi, food, growth, hit, hous, increas, job, larg, largest, local, market, meet, offic, policy, poundbn, pressur, price, privat, retail, rock, senior, share, spokesman, start, statement, talk, trade, uk, wam, world, year, attack, defenc, div, fight, hope, lead, michael, team, win, bit, championship, crowd, game, grand, injury, minut, round, season, sport, straight, yearold, ancient, hotel, rive, ski, spectacular, travel

Table 2.1 Features selected

- **PHASE 3: ADOPTION OF DOCUMENT CLASSIFIER – NAIVE BAYES**

After preprocessing and feature selection phases, the numbers of attribute will be significantly reduced and are more precise for the use in building the classification model. "For the classification phase, Naive Bayes is used as the classifier because of its simplicity and good performance in document and text classification", as reported and discussed by Chakrabarti et al.

Naive Bayes classifier is the simplest instance of a probabilistic classifier. The output Pr(C|d) of a probabilistic classifier is the probability that a document d belongs to a class C. Each document contains terms which are given probabilities based on its number of occurrence within that particular documents. With the supervised training, Naive Bayes can learn the pattern of examining a set of test documents that have been well-categorized and hence comparing the contents in all categories by building a list of words as well as their occurrence. Thus, such list of word occurrence can be used to classify the new documents to their right categories, according to the highest posterior probability.

- **PHASE 4: MODEL EVALUATION**

To test and evaluate the model, 70% of the dataset are used. Instances are extracted and then served as a benchmarking dataset for machine learning problems. By comparing the actual class of the instance with the predicted one (i.e. generated by the classification model), system performance can be measures in term of recall, precision, and F-measure. These can be mathematically defined as below

$$\text{recall} = \frac{\text{Number of documents retrieved that are relevant}}{\text{Total number of documents that are relevant}} \quad (1)$$

$$\text{precision} = \frac{\text{Number of documents retreived that are relevant}}{\text{Total number of documents that are retrieved}} \quad (2)$$

$$\text{F-measure} = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}} \quad (3)$$

In order to further evaluate the performance of the proposed preprocessing stage, the results of not preprocess and preprocess are compared. However, if the results are worse than that when no preprocessing phase is conducted (i.e. the classification model is not good enough), therefore adjusting and fine-tuning parameters are required (e.g. modifying the technique used in feature selection) and hence re-build the model again. This step will stop until a good classification result is obtained. Furthermore, Naive Bayes classifier will be tested with other classifier (such as DT, SVM, NN) to

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 5 Volume No.6 Aug-Sept – 2022
DOI:10.46647/ijetms.2022.v06i05.137 ISSN: 2581-4621

determine whether Naive Bayes is the best classifier among them.

III. PERFORMANCE EVALUATION DATA DESCRIPTION

The goal of this study is to classify the given specified experimental dataset into four categories (i.e. business, politics, sport, travel) correctly. To start with, it is given 1000 documents for each category to serve as the dataset for generating the classification model. To build and evaluate the classification model, the total 4000 documents will be split into two datasets, namely training set and testing set, in which 30% of the documents will go to the training set whereas the remaining 70% will go to the testing set. In the representation of these documents, they have been vectorized into 1311 attributes (in term of numerical values) and 1 solution attribute (in term of nominal values). No missing data is among the attributes and all the numeric attributes are described in the term frequency/inverse document frequency (TFIDF). An example of the data can be presented as

Figure 3.1.1 and Table 3.1.2 summarizes the description data in both training and testing set.

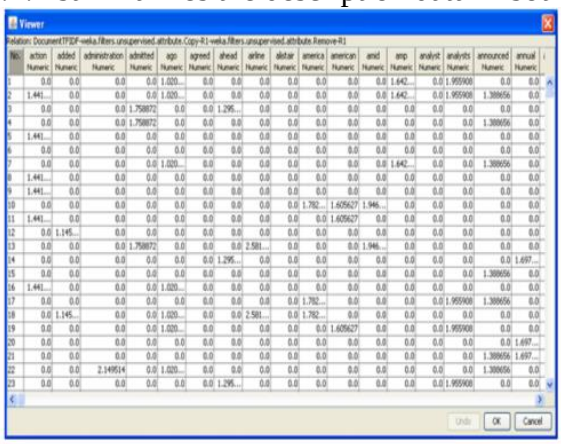

Figure 3.1. Example of Data View

	Training Data	Testing Data
Number of instances	1200	2800
Number of attributes	1312 (Numeric – 1311; Nominal -1)	1312 (Numeric – 1311; Nominal -1)
Missing data	No	No

Table 3.1.2 Data Description in this Study

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

The objective of this evaluation is twofold. First, it determines whether the preprocessing phase is useful to deduce better classification accuracy and performance when compared to the situation that has not been preprocessed the data. Second, it compares the classification accuracy and performance when different classifiers are applied. A dataset with 4000 documents classified in four different categories is used for evaluation. The selected dataset contains four categories of document: business, politic, sports, and travel. All the four categories are easily differentiated. 30% data (i.e. 1200 documents) are extracted randomly to build the training dataset for the classifier. The other 2800 documents are used as the testing dataset to test the classifier. The model is built based on the "Naive Bayes" classifier developed in Weka. Table 4.1 summarizes the result of using Naive Bayes classifier to classify the documents. However, it surprisingly finds that the results of preprocessed dataset (95.5%) are worse than those which have not preprocessed (96.9%). Therefore, it is required to adjust the preprocessed model in order to achieve a better result. Considering the preprocessing phase is common to adopt in all case, therefore the adjustment is made in the feature selection phase.

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 5 Volume No.6 Aug-Sept – 2022
DOI:10.46647/ijetms.2022.v06i05.137 ISSN: 2581-4621

	Correctly Classified Instances	Incorrectly Classified Instances	Precision	Recall	F-Measure
Without Preprocessing and Feature selection	2713 (96.9%)	87 (3.1%)	0.969	0.969	0.969
With Preprocessing and Feature selection	2675 (95.5%)	125 (4.5%)	0.956	0.955	0.955

Table 4.1. Classification Accuracy of Naive Bayes Classifier (by Using the Dataset with Preprocessing and without Preprocessing)

As mentioned above, Cfs Subset Evaluator and rank search (with Gain ration metric) are used for the feature selection. Therefore, another technique for rank search has been tried to adopt. This time, Chi-square feature selection has been adopted and 89 attributes are selected (Figure 4.1). Rather than 75 attributes being selected previously, 89 attributes has been inputted this time. The result has been improved after using Chi-square feature selection, as depicted in Table 3.1 and Figure 4.1. The accuracy has been improved 0.1%. Although the improvement is insignificant, it is proven that preprocessing and feature selection are useful in achieving better classification result. It is believed that different searching technique can help to accomplish different classification result under different situations, in which it needs to take many trials and time to generalizing the best solution. However, due to the time constraints, this study only draws the conclusion of using preprocessing and feature selection can achieve better classification result. Furthermore, another critical point can be found is that the time used to build the model is significantly improved after the number of features has been greatly reduced from 9.66 seconds to around 0.19 seconds (Table 4.1).

After discussing the importance of preprocessing and feature selection, the following experiment is to test whether Naive Bayes is the best classifier among other classifiers. To serve for this purpose, three different classifiers have been applied for testing. These classifiers are: SVM (the “SMO” function in WEKA), NN (the lazy “IBk”), and DT (the tree “J48”). In this experiment, the preprocessed dataset (with 90 attributes) are used for evaluation. Table 4.2 summarized all the accuracy results with the precision, recall, and F-Measure. As shown in the table, the accuracy result of Naive Bayes is the best among other classifiers. Although SVM gets similar r results as Naive Bayes, the time taken to build the model is dissatisfactory. Compared with the times used for building a Naive Bayes classifier (0.19 seconds), SVM requires 2.69 seconds, which is 14 times of Naïve Bayes classifier, as depicted in Table 4.4. As a result, Naïve Bayes is reported to be the best text classifier.

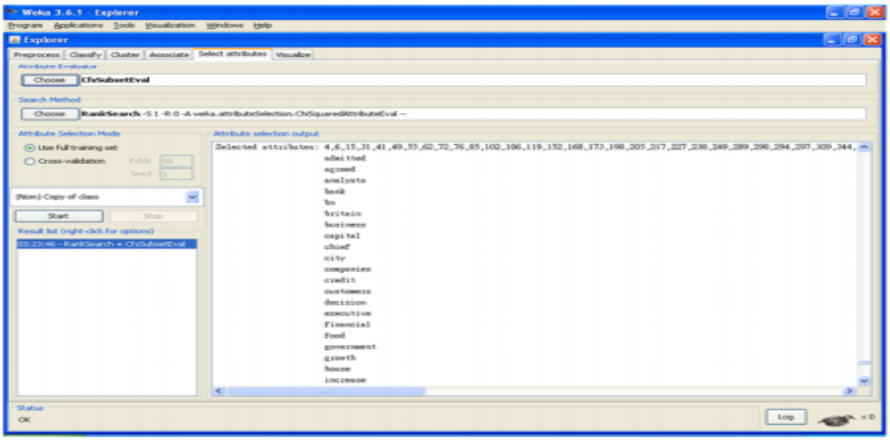

Figure 4.1. Feature selection result using the Cfs Subset Evaluator and Rank Search (with Chi-square feature selection)

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 5 Volume No.6 Aug-Sept – 2022
DOI:10.46647/ijetms.2022.v06i05.137 ISSN: 2581-4621

	Correctly Classified Instances	Incorrectly Classified Instances	Precision	Recall	F-Measure
Without Preprocessing and Feature selection	2713 (96.9%)	87 (3.1%)	0.969	0.969	0.969
With Preprocessing and Feature selection – Gain Ratio	2675 (95.5%)	125 (4.5%)	0.956	0.955	0.955
With Preprocessing and Feature selection – Chi-square	2717 (97.0%)	83 (3.0%)	0.970	0.970	0.970

Table 4.2. Classification Accuracy of Naive Bayes Classifier

	Times taken to build model (seconds)
Without Preprocessing and Feature selection	9.66
With Preprocessing and Feature selection – Gain Ratio	0.14
With Preprocessing and Feature selection – Chi-square	0.19

Table 4.3 Times taken to build the Naive Bayes classifier (by using thedataset with preprocessing and without preprocessing)

	Correctly Classified Instances	Incorrectly Classified Instances	Precision	Recall	F-Measure
Naïve Bayes	2717 (97.0%)	83 (3.0%)	0.970	0.970	0.970
SVM	2712 (96.9%)	88 (3.1%)	0.969	0.969	0.969
NN	2605 (93.0%)	195 (7.0%)	0.931	0.930	0.930
DT	2551 (91.1%)	249 (8.9%)	0.911	0.911	0.911

Table 4.4. Classification Accuracy of Different Classifiers

	Times taken to build model (seconds)
Naïve Bayes	0.19
SVM	2.69
NN	0
DT	1.8

Table 4.5. Times Taken to Build the Classifiers

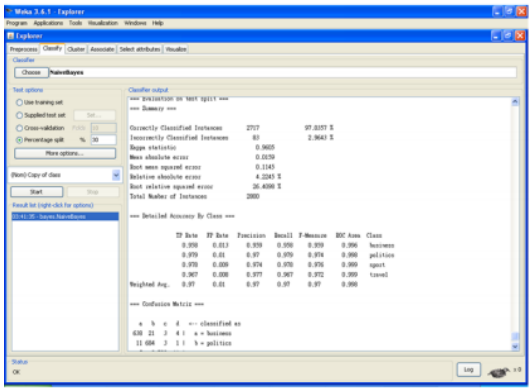

Figure 4.2. Classification Result Using Naive Bayes Classifier with Preprocessed Dataset

V.Conclusion

In this study, Naive Bayes classifier has been discussed as the best document classifier, which satisfies the literature result. Through the implementation of different feature selection and classifier available in WEKA, it is demonstrated preprocessing and feature selection are two important steps to improve the mining quality. There are many words in the documents, therefore when we captured the terms from these documents, thousands of terms are found.

However, there are some terms that are usefulness and uninteresting to the results, it is then important to discover and interpret which features are useful and critical. Concerning numerous searching and selection techniques are available; it is encouraged to apply all these techniques and hence selects the best one for preprocess the data as well as to build the model. Furthermore, the performance of mining result is directly affected by the quality of data. So, preprocessing phase is important to make the data being more precise (so as to achieve a better classification result) and even improve the time used to train and general the model, as proven in the experiment section.

References

[1].Uthurusamy: et al..,"Advances in Knowledge Discovery and Data Mining", AAAI/MIT Press 1996

DOI:10.46647/ijetms.2022.v06i05.137 ISSN: 2581-4621

[2]. Quinlan JR (1993) C4.5: et al..," Programs for machine learning". Morgan Kaufmann Publishers, San Mateo
[3]. Stone PJ (1966) et al..," Experiments in induction". Academic Press, New York
[4]. Quinlan JR (1979) Discovering rules by induction from large collections of examples. In: Michie D (ed)
[5]. Olshen RA, Stone CJ (1984) et al..,"Classification and regression trees,Wadsworth"
[6]. J. Han and M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann Publish, 2001
[7]. K. A. Smith , et al..," On learning algorithm for classification", Applied Soft Computing Dec 2004. pp. 119-138.
[8]. Pang-Ning Tan, Michael Steinbach and Vipin Kumar. Introduction to Data Mining, Addison Wesley, 2006
[9]. Tony R. Martinez. et al..," Functions. Improved Heterogeneous Distance" In Journal of Artificial Intelligence Research (January 1997), pp. 1-34.Cambridge University Press; 2000
[10]. Vapnik V (1995) The nature of statistical learning theory. Springer, New York
[11]. Agrawal R, Srikant R (1994) Fast algorithms for mining association rules. In: Proceeding of the 20th VLDB conference, pp 487–499
[12]. Schapire RE (1997) et al..,"A decision-theoretic generalization of on-line learning and an application to boosting". J Comput Syst Sci 55(1):119–139
[13]. Xindong Wu et.al, "Top 10 Algorithms of Data Mining", Springer-Verlag London, 2007