International Journal of Advance Research in Science and Engineering
Volume No 06, Special Issue No. (02), September 2017, ICITTESE-17
www.ijarse.com
IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

Text Classification for E-mail Spam Using Naïve Bayesian Classifier

Priyanka Sao[1], Shilpi Chaubey[2], Sonali Katailiha[3]

[1,2,3]Assistant ProfessorCSE Dept, Columbia Institute of Engg&Tech,
Columbia Institute of Engg&Tech, Raipur (India)

ABSTRACT

The documents to be classified may be texts, images, music, etc. Each kind of document possesses its special classification problems. When not otherwise specified, text classification is implies. The Naive Bayes classifier is a simple probabilistic classifier which is based on Bayes theorem with strong and naïve independence assumptions. It is one of the most basic text classification techniques with various applications in email spam detection, personal email sorting, document categorization, language detection and sentiment detection. E-mail spam is the very recent problem for every individual. The e-mail spam is nothing it's an advertisement of any company/product or any kind of virus which is receiving by the email client mailbox without any notification. For protecting our system to spam we are using different spam classification techniques. In this project, we are using the Naïve Bayesian Classifier for spam classification. The Naïve Bayesian Classifier is very simple and efficient method for spam classification. Here we are using the Lingspam dataset for classification of spam and non-spam mails. We are using feature extraction technique for pre-processing the data and producing the better result.

Keyword: *E-mail spam, Classification, Feature Extraction, Naïve Bayesian Classifier, Support Vector Machine*

.

I. INTRODUCTION

Text classifications have carries different phases like text collection feature generation and feature selection and also contain different classifier machine for classify the text data.

1. Document collection: in the first step we are collecting the data or document from the different sources.

2. Document pre processing and representation: Document containing the collection of number of words so we can classify only the meaning full words for this process we require the document pre processing which are use to eliminate the non words from the document.

3. Feature generation & Selection: To reduce the complexity of the document the feature generation is the very important step. The document must be in full text document so we can easily classify the document this process is also called text transformation.

4. Feature Selection: To reduce the redundancy of the document we can use the feature selection techniques in feature selection take the subset of the data from the document.

5. Text Classification: we are using the different application to classify the text. Text classified in three ways: Supervised Unsupervised and semi- Supervised method.

6. Performance Evolution: At the last step compare the performance of the classifier with respect to accuracy frequency and time.

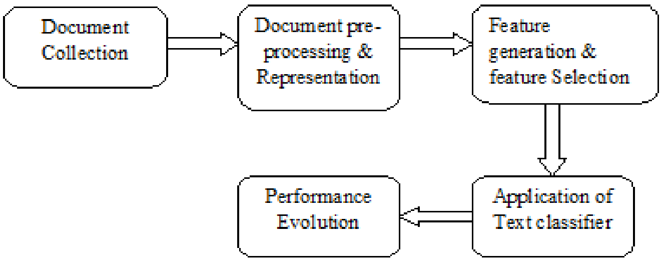

Document Collection
Document pre-processing & Representation
Feature generation & feature Selection
Application of Text classifier
Performance Evolution

Fig 1.1: Text Classification

Here we are applying the text classification technique to classify the E-mail. Email spam is operations which are sending the undesirable messages to different email client. The historical backdrop of Email spam is begin before 2004, however these are the enormous parts that convey spam to the way it is today. Commercialization of the web and united as complete thing of electronic post as a ready to be got to method for news has another face things coming in of not needed data and sends messages

One subset of UBE is UCE (Unsolicited Commercial Email). The inverse of "spam", email which one needs, is called "ham", normally when alluding to a messages mechanized examination, (for example, Bayesian Filtering).

II. RELATED WORK

Web spam which is a major issue throughout today's web search tool; consequently it is important for web crawlers to have the capacity to detect web spam amid creeping. The Classification Models are designed by machine learning order algorithm. [2] The one machine learning algorithm is Naïve Bayesian Classifier which is also used in [1] to separate the spam and non-spam mails. Big Data analyzing framework which is also outline for spam detection. Extricating the feeling from a message is a method for get the valuable data. In Machine learning innovations can gain from the preparation datasets furthermore anticipate the choice making framework hence they are broadly utilized as a part of feeling order with the exceptionally precision of framework. [3]

Most of the research work has already been carried out on improving the efficiency and accuracy of Naïve Bayesian approach. Paul Graham's Naïve Bayesian Machine learning approach is used to improve the efficiency of Bayesian approach. [1] For vast dataset also using the naïve Bayesian algorithm and increment the precision of NBC. [4] The research work has also carried out for increase the accuracy and time efficiency of system. In this paper also present the comparison of classifier with respect to Time consuming [14].

III. PROBLEM IDENTIFICATION

In social network Email spam is very crucial matter. There are lots of problem created through spam. The spam is nothing this is unwanted message or mail which the end user doesn't want in our mail box. Because of these spam the performance of the system can be degraded and also affected the accuracy of the system. In the previous study there are many types of spam classifier are present too detect the spam and non-spam mails.

There are different email filtering techniques are also used in spam detection. Mostly popular filters or classifier are: Decision tree classifier, Negative Selection Algorithm, Genetic Algorithm Support Vector Machine

Classifier, Bayesian Classifier etc. From the previous study we identify that Support Vector Machine (SVM Classifier) are used for email spam classification. But it takes very much time for detecting spam. The SVM Classifier has also wrongly classified the messages. So the system can be on a risk. The error rate of SVM Classifier is very high. In this project there is also discussion in the Feature Selection process.

Solution of the Problem:

To solve the problem of previous study in this project we are uses the Naive Bayesian Classifier for classify the spam and non-spam mails. The naive Bayesian Classifier is one of the most popular and simplest methods for classification. Naive Bayesian Classifiers are highly scalable, learning problem the number of features are required for the number of linear parameter. Training of the large data simple can be easily done with Naive Bayesian Classifier, which takes a very less time as compared to other classifier. The accuracy of system is increase using Naïve Bayesian Classifier. And also it takes the less time to classify the e-mail.

IV. METHODOLOGY

In this work we use feature extraction techniques for providing efficient dataset. The feature extraction techniques are used when the input data is too large and it is redundant in nature so feature is extracted to obtain an accurate result. In this work we are using the word-count algorithm for extracting feature from the dataset. Here we use the Lingspam data set which contains total 960 mails in which 700 are train dataset and 260 are test dataset. The train and test data are further divided in two parts spam mails and non-spam i.e. 50% of train dataset are spam dataset and 50% are non-spam dataset as same for the test dataset.

The Feature generation and Selection:

To provide flexible result we are using word-count algorithm for extracting feature of document. With the help of this algorithm we pre-process the dataset and remove the stop-words and non-words in dataset. And then it counts the total number of unique word out of the total word and finds the frequency of that word in a particular document. The main thing about this algorithm is to makes a dictionary. In that dictionary the path of the file is stored which is pre-processed. So the redundancy problem is removed. For counting the word and store the frequency of that word is very helpful to find the unique word.

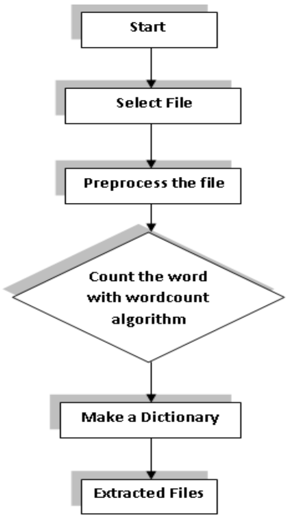


Fig.4.1: Feature generation and selection

Algorithm:

```
Step 1: Select the file from the dataset.
Step 2: Removing the stop-word by preprocessing the document.
Step3: Count the total word of the file and find the unique      word of that file.
Step 4: Find the frequency of words.
Step 5: store the file path and make a dictionary.
Step 6: Extracted Feature.
```

The first is to select the file from the dataset. Then second pre-process the data in which we remove all the duplicate words and symbols or non words present in dataset. Then count the unique word from total number of words. So we can calculate the frequency of that word in a document. The forth step is to make a dictionary and store the path of document this can solve the redundancy problem. The extracted data are received after all steps are complete.

The proposed methodology:

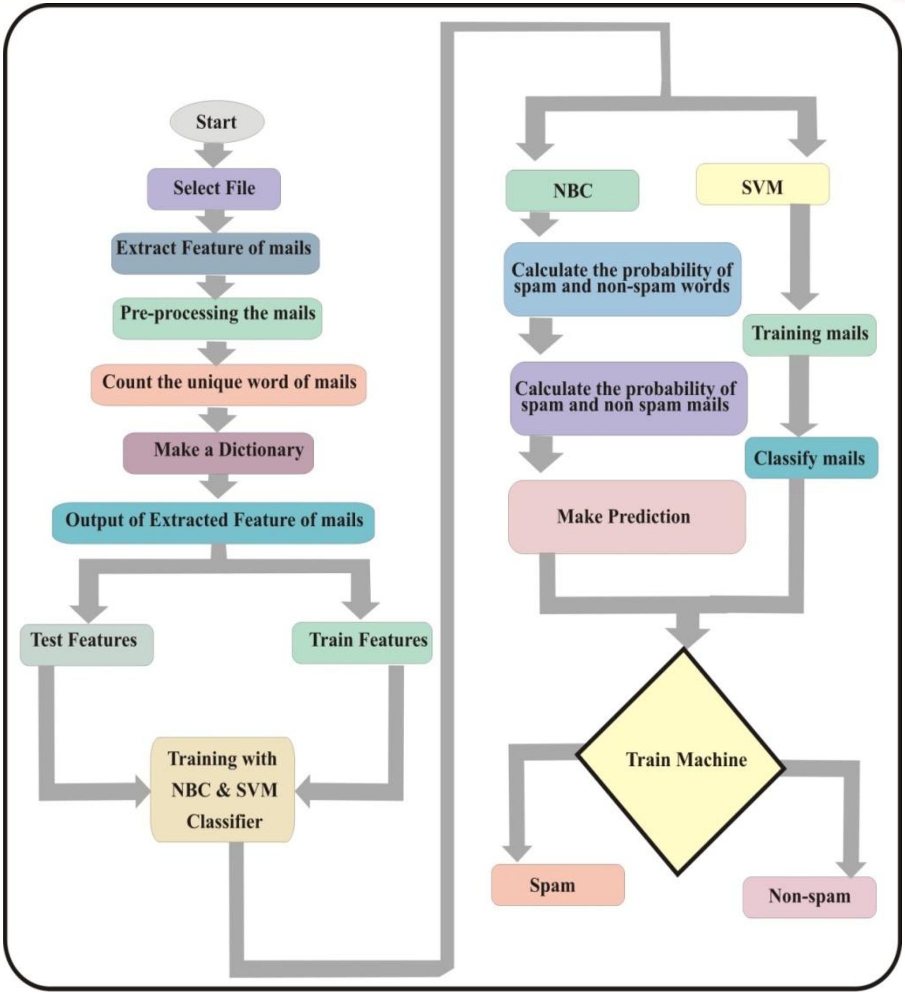

Start
Select File
Extract Feature of mails
Pre-processing the mails
Count the unique word of mails
Make a Dictionary
Output of Extracted Feature of mails
Test Features
Train Features
Training with NBC & SVM Classifier
NBC
SVM
Calculate the probability of spam and non-spam words
Training mails
Calculate the probability of spam and non spam mails
Classify mails
Make Prediction
Train Machine
Spam
Non-spam

Fig 4.2: The Proposed Methodology

Algorithm:

```
Step 1: Select the file
Step2: with help of wordcount algorithm extracting the feature.
Step 3: Apply Naive Bayesian Classifier to training dataset.
Step 4: Find the probability of spam and non-spam mails.
Prob_spam = (sum(train_matrix(spam_indices, )) + 1) ./ (spam_wc + numtokens)
Prob_nonspam =  (sum(train_matrix(nonspam_indices, )) + 1) ./ (nonspam_wc + numtokens)
Step 5: Testing the dataset
log_a = test_matrix*(log(prob_tokens_spam))' + log(prob_spam)
log_b = test_matrix*(log(prob_tokens_nonspam))'+ log(1 - prob_spam)
 if
output = log_a > log_b
then document are spam
else the document are non-spam
Step 6: Classify the spam and non-spam mails.
Step 7: compute the error of the text data and calculate the word which is wrongly classified
Numdocs_wrong = sum(xor(output, text_lables))
Step 8: display the error rate of text data and calculate the fraction of wrongly classified word
Fraction_wrong = numdocs_wrong/numtest_docs
```

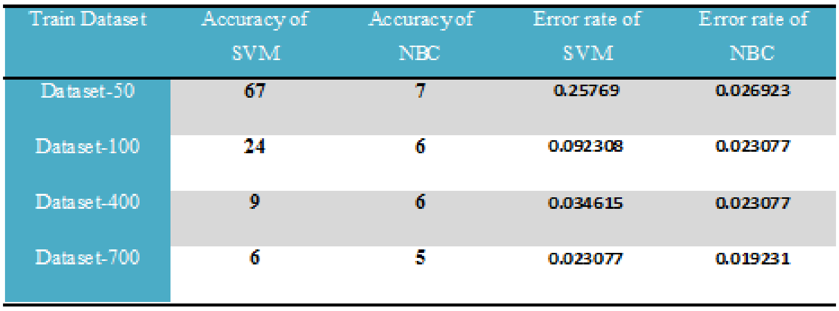
Description: The proposed methodologies explain how the email get classify in spam mails and non spam mails. The first step is to select the file from the dataset and apply the feature extraction technique. For which we are using the word-count algorithm. The next step is training the dataset which are extracted by the feature extraction technique. We can calculate the probability of spam and non-spam words in the document to training the dataset. Then we start to test the data with the help of Naïve Bayesian Classifier for which calculation the probability of spam and non-spam mails and make a prediction which value is higher. If spam words are greater than non-spam words in a mail then the mail is spam mails otherwise non-spam mails.

In the next step we are calculating the words which are wrongly classified by the classifier and calculate accuracy of the classifier and also calculate the error rate of classifier by calculating the fraction of word which is wrongly classified and total number of words in document. For which can find the frequency of the classifier and also calculate the time to compute the classification.

V. Result and Discussion

In this project we are creating an email spam classification system for classify the spam and non-spam mails. For this we are taking the Lingspam dataset to run this experiment. In a Lingspam dataset we are taking total 960 mails in which 700 train dataset and 260 test dataset. Out of 700 train dataset the 350 are spam mails and 350 are non-spam mails. Similarly the 260 test dataset is containing 130 spam mails and 130 non-spam mails.

Here we are present different reading for all four trained dataset which are tested by the classifier i.e. Naive Bayesian Classifier and Support Vector Machine. Hence shown the different readings and calculation of result:



Train Dataset	Accuracy of SVM	Accuracy of NBC	Error rate of SVM	Error rate of NBC
Dataset-50	67	7	0.25769	0.026923
Dataset-100	24	6	0.092308	0.023077
Dataset-400	9	6	0.034615	0.023077
Dataset-700	6	5	0.023077	0.019231

Table 5.1: Resulted readings of different classifier

This reading contains the text data which are classify by the classifier and provide the word which are wrongly classified and error rate of classifier. Hence we can show the overall result which are provided by classifier. And say that the Naive Bayesian Classifier classifies mostly word in accurate way.

When the number of dataset is increase the Naive Bayesian Classifier produce a better result as compared to Support Vector Machine. Hence we can show the graph for display the accuracy and error rate of both classifier (Naive Bayesian Classifier and Support Vector Machine).

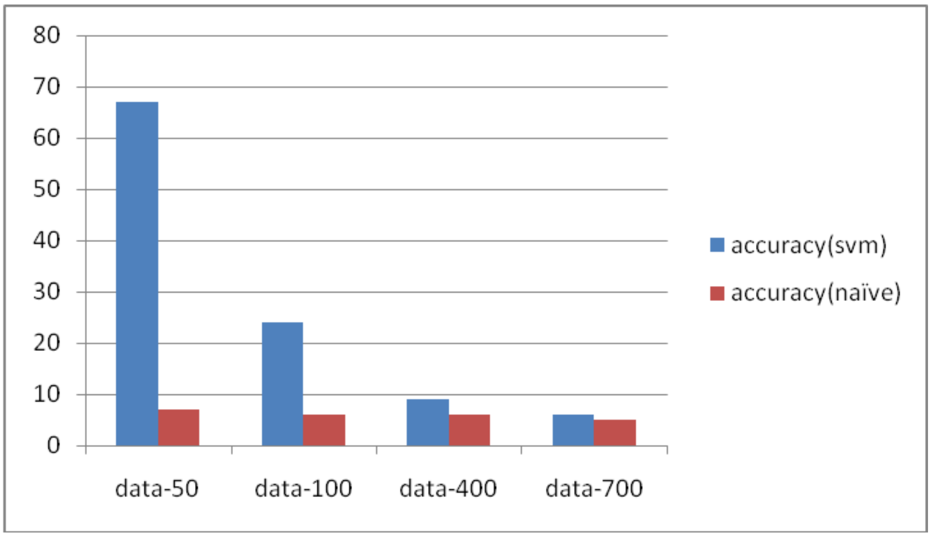


Figure 5.1: representing the accuracy graph

In this graph we can show the accuracy of classifiers (Naive Bayesian Classifier and Support Vector Machine). For this we are calculating the words which are wrongly classified by the classifier. The classifier which are classified more numbers of wrong words is less accurate as compared to the classifier which classified less numbers of wrong word. So here the Naive Bayesian Classifier is more accurate then Support Vector Machine.

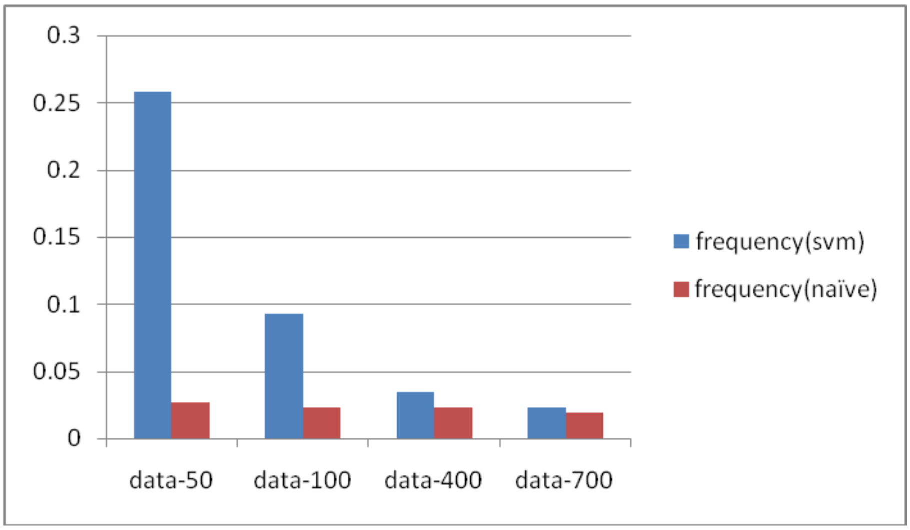


Fig 5.2: representing the frequency graph of classifier

In this graph we show the error rate of classifiers for which we calculate the fraction of a word which are wrongly classified into the total number of words. Hence we can see that the error rate of blue bar is greater than the error rate of red bar, so we can say the Support Vector Machine provide the high error rate than the Naive Bayesian Classifier. We can calculate the time of the classifier and show that the Naïve Bayesian Classifier has higher speed as compared to Support Vector machine.

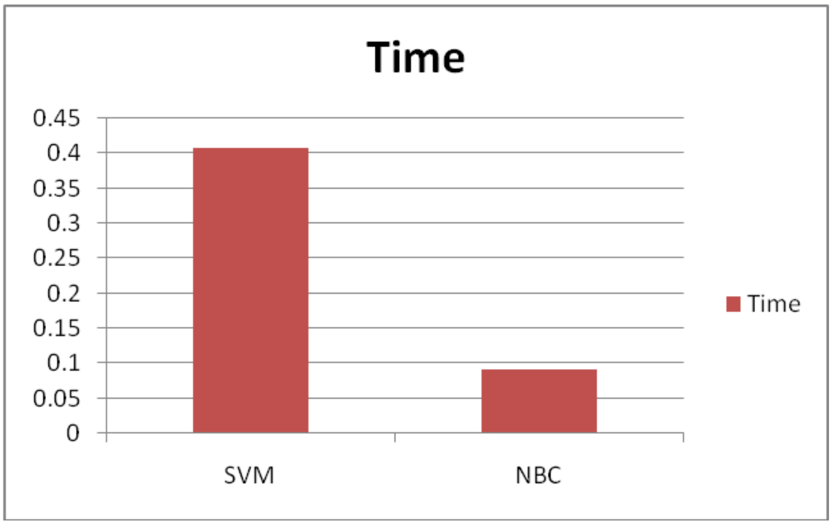


Fig 5.3: Time Consume for Classifiers

In this graph we can show the time consuming of both the classifier here we can see that Naïve Bayesian Classifier has taken very less time to classify the mail as compared to Support Vector Machine.

VI. CONCLUSION

The text classification is a general way to specify the classification of different classifier. E-mail spamming is one of the text classification techniques. E-mail spam is big problem in daily life; to solve this problem the spam classification system is created to identify the spam and non-spam mails. Spam mails are advertisement of any company or any kind of virus that could be harming your account. To solve this problem create an email spam classification system and identifies the spam and non-spam mails.

Here we are using the Naïve Bayesian Classifier for classification of spam and non spam mails. After calculation we find that naïve Bayesian classifier has more accurate the support vector machine. The number of words which are wrongly classified is very less in naïve Bayesian Classifier, So we can say that Naïve Bayesian Classifier produce better result than Support Vector Machine. For study it evaluate that Naïve Bayesian classifier is very less time consume as compared to other Classifier because the error rate is very low in Naïve Bayesian Classifier.

REFERENCES

[1] Sharma K. and Jatana N. (2014)“Bayesian Spam Classification: Time Efficient Radix Encoded Fragmented Database Approach” IEEE 2014 pp. 939-942.

[2] Sharma A. and Anchal (2014), "SMS Spam Detection Using Neural Network Classifier",ISSN: 2277 128X Volume 4, Issue 6, June 2014, pp. 240-244.

[3] Ali M. et al (2014), , "Multiple Classifications for Detecting Spam email by Novel Consultation Algorithm", CCECE 2014, IEEE 2014, pp. 1-5.

[4] Liu B. et al (2013) “Scalable Sentiment Classification for Big Data Analysis Using Na¨ıve Bayes Classifier” IEEE 2013 pp.99-104.

[5] Belkebir R. and Guessoum A. (2013), "A Hybrid BSO-Chi2-SVM Approach to Arabic Text Categorization", IEEE 2013, pp. 978-984.

[6] Blasch E. et al (2013), Kohler, “Information fusion in a cloud-enabled environment,” High Performance Semantic Cloud Auditing, Springer Publishing.

[7] Allias N. (2013) “A Hybrid Gini PSO-SVM Feature Selection: An Empirical Study of Population Sizes on Different Classifier” pp 107-110.

[8] Prasad N. et al (2013) "Comparison of Back Propagation and Resilient Propagation Algorithm for Spam Classification", Fifth International Conference on Computational Intelligence, Modelling and Simulation, IEEE 2013, pp. 29-34.

[9] Jia Z. et al (2012) “Research on Web Spam Detection Base on Support Vector Machine” IEEE 2012 pp. 517-520.

[10] Panigrahi P. (2012) , "A Comparative Study of Supervised Machine Learning Techniques for Spam E-Mail Filtering", Fourth International Conference on Computational Intelligence and Communication Networks, IEEE 2012, pp. 506-512.

[11] Clark J. et al (2010), “A Neural Network Based Approach to Automated Email Classification.

[12] Sun X. et al (2009), “Using LPP and LS-SVM For Spam Filtering”, School of Information Science and Engineering Henan University of Technology IEEE 2009, pp. 4244-4246.

[13] Hmeidi I. and Hawashin B. (2008),”Performance of KNN and SVM classifiers on full word Arabic articles,” Advanced Engineering Informatics, vol. 22, no. 1, pp. 106-111

[14] Rajeshwari R.Pet al (2017),”Text classification of student data set using naïve bayes classifier and KNN Classifier” International Journal of Computer Trends and Technology (IJCTT) – Volume 43 Number 1, pp 8-12

[15] Jain A. and Mishra R. (2016),” TEXT CATEGORIZATION: BY COMBINING NAÏVE BAYES AND MODIFIED MAXIMUM ENTROPY CLASSIFIERS”, International Journal of Advances in Electronics and Computer Science, ISSN: 2393-2835, pp 122-126.

Website:

1. http://www.Anti-spamtechniquesWikipedia.org
2. http://www.EmailspamWikipedia.org
3. http://www.TextcategorizationScholarpedia.com