An Effective Approach to Auto Email Content Management: Keyword Based Classification

Prof. S.M.Khatri, Prof. Amol Bhilare
Asst. Prof.(Dept. of Computer engineering)
Vishwakarma Institute of Technology
Pune, India 411037
saurabh.khatri@vit.edu
amol.bhilare@vit.edu

Subodh Mankar, Ashwin Lenekar, Swapnil Lawand, Dhruv Lokapure
Student (Dept. of Computer engineering)
Vishwakarma Institute of Technology
Pune, India 411037
subodhmankar27@gmail.com
aklenkear@gmail.com
swapnil.lawand@yahoo.com
dhruv_lokapure@gmail.com

***Abstract*- Today majority of the communication in organizations, big industries, Educational institutions occurs through email. The number of emails received by a working individual in a day has also increased considerably which has increased the overall user workload. Some emails have much higher importance than other mails in the inbox. Time spent in searching a particular email when required reduces the efficiency at which a user performs task.**

These difficulties can be avoided if the incoming email is classified automatically. To achieve automated email classification, many machine learning techniques can be used. Naive Bayes, K-nearest neighbor (K-NN), support vector machines (SVM) are few machine learning algorithms which can be used to classify an email. In this paper a Keyword based approach has been applied to auto manage mailboxes. We have found in our studies that only few selective phrases and words play a role in email classification problem. Thus this approach proves to be much more effective than the above mentioned algorithms both in terms of accuracy and time complexity.

Keywords- Keyword based approach; Naïve Baye; K-NN; SVM; email classification.

I. INTRODUCTION

Users find it very difficult to manage their inbox because of the vast number of sources through which they receive email. Many a times, user needs to go through his old mails for some purpose. If the inbox is not properly managed, user may take much amount of time in completing his tasks. To minimize this effort management of their inbox is necessary task which can be achieved through Text categorization.

The research area of Text Categorization has gained great attention in the recent years due to mass increase of Digital data of all forms such as Web pages, E-learning material, emails, registration forms, online Feedbacks, reviews and so on. The problem of TC can be defined as: Given a set of Training Documents D= {(d1, c1), (d2, c2), (dn, cn)}, where di represents ith document of D and ci represents category of ith document.

Based on the training set a Training Model T will be generated, T be some function of Training Dataset D.

$$T = F(D)$$

We need to classify a new instance of document dj into its most likely category using model T. This TC problem sets the foundation for the information Retrieval system. Email classification also derives its roots from the same concept. It included automatic classification of emails into one of the pre-specified category. But this application may face many problems while categorization of emails because of inclusion of noise while training of documents. In order to perform email classification, an important process is representing each email in vector space such as di= {f1, f2….fn}. An email contains many things which we do not need while classification or it causes problems while classification. We need to reduce the size of the email by removing the stop words such as prepositions, conjunctions etc. Words are used as features in classification. So we need to keep only those words which will provide a valuable input in the classification procedure. We also need to bring the words in their root form by stemming them. This procedure is called pre-processing of document and it plays a very important role in the final accuracy of the algorithm. Good pre-processing increases the probability of correct classification of a document.

The rest of the paper is organized as follows: Section II describes some of the previous work done in TC and a comparison of their accuracy of classification. Section III introduces our proposed model i.e. a Keyword based approach to email classification; Section IV summarizes the experimental set up and system requirements, Section V illustrates the results and observation of all the studied algorithm and Section VI concludes the paper along with future scope

II.RELATED WORK

The study of different machine learning algorithms which are mostly used in the domain of text classification was carried out. The study of those algorithms is described below.

A. Naïve Bayes

Svetlana Kiritchenko and Stan Matwin [1] discuss the Naive Bayes classifier technique is based on Bayesian theorem and is particularly suited when the dimensionality of the input is high. Simple (“naive”) classifica1on method based on Bayes rule relies on very simple representation of document (Bag of words).

Sharma Chakravarthy [7] explains this bag of words approach could be difficult for further processing as this algorithm advances through complex calculations and to deal with it the data available should be as less as possible. In our project the bag of words representation will be of an email received by the user. To reduce the bag of words approach these text files need to be pre-processed. Pre-processing includes removing punctuations, removing stop words, stemming and finding the tf*idf weights of each feature [2]. Term frequency (tf)- tf is the frequency of a particular word in a document. Inverse document frequency idf is log (d/df). After applying above steps features which are remaining will be the main features used further in algorithm.

1) Algorithm:

Now the main task of an algorithm is an input document d is to be categorized into a fixed set of classes’ viz. C= {Reply, Meeting, informative}

a) To carry out the above mentioned task the naïve Bayes classifier first needs to be trained. Training includes calculating the prior probabilities of each class.

E.g. P (Reply) = $N_{reply}/(N_{reply}+N_{informative}+N_{meeting})$

b) Then to construct the Unigram Language Model, Add one smoothing of each feature for every class needs to be calculated.

E.g.P(word|Reply)=$(T_{word}|Reply)/((T_{word}|Reply+1)+(T_{word1}|Meeting+1)+(T_{word2}|Informative+1))$

c) As the training gets completed we need to test the accuracy of the trained classifier. Suppose d6 is the new mail needs to be categorized i.e. tested, so Testing is carried out using following steps Unigram Language Model is applied as

E.g.P(d6|Reply)=P(word1|Reply)*P(word2|Reply)*P(word3|Reply)

d) The final step is to calculate the Posterior probabilities of each class using Bayes Rule.

E.g. P (Reply|d6) = P (d6|Reply)*P (Reply)/P (d6)
P (Meeting|d6) = P (d6|Meeting)*P (Meeting)/P (d6)
P (Info|d6) = P (d6|Info)*P (Info)/P (d6)

Based on the comparison of above calculated three values the actual class of email is determined

P (Meeting|d6)> P (Reply|d6)> P (Informative|d6)

Since the posterior probability for class “Meeting” is greatest thus the mail d6 will be categorized in “Meeting” class. The main advantage of this algorithm is its time complexity. It is considerably very low as compared to other algorithms. It works very fast. It works very accurate with smaller size datasets.

B. K-NN

Pablo Bermejo, Jose A. Gamez, Jose M. Puerta and Roberto Uribe [3] have performed an extensive study on improving knn based e-mail classification into folders generating class balanced datasets. The basic principle behind k-NN is to find the ‘k’ training samples to determine the k- nearest neighbor based on a distance measure. Next, the majority of that k nearest neighbors decides the category of the next instance. The value of k is determined arbitrarily. When a new mail is received the k near neighbors are determined on the basis of a computational model. The document is classified into category where the majority of the neighbors belong.

1) Algorithm

a) Determine k.

b) Train the classifier using documents D1…..Dn for classes C1, C2, and C3.

c) Consider new document D as the testing document

d) Calculate distance d= {D, D1} with all the training data.

e) Sort the distances ‘d’ calculated and determine k nearest neighbours based on the k^{th} minimum distance.

f) Determine the categories C of all these k nearest neighbours.

g) Based on the majority of vote category C of the document D will be decided

KNN algorithm is simple to implement and powerful. The algorithm does not need for tuning complex parameters to build a model. The algorithm is Lazy. New training examples can be added easily. The algorithm shows better results for larger data set.

C. SVM

Bryan Klimt, Shyamsundar Jayaraman, Yiming Yang [4] and Svetlana Kiritchenko and Stan Matwin [1] have examined performance of svm which briefs a support vector machine constructs a hyper plane or a set of hyper planes in a high or infinite dimensional space which can be used for classification, regression or other tasks. A good separation is achieved by the hyper plane that has the largest distance to the nearest training data point of any class. In general, larger the margin, lower is the generalization error of the classifier. To keep the computational load reasonable, the mappings used by SVM schemes are designed to ensure that the dot products may be computed easily in terms of the variables in original space by defining them in terms of kernel function k (x, y). Bo Yu [6] states SVM has a greater ability to generalize, which is the goal in statistical learning. Thus it is one of the most used and accurate classifier.

1) Implementation

a) Collection of data set.

b) Manual distribution of files in three classes namely Reply, Meeting and Informative.

c) Training- For implementation purpose we have used Libsvm as the source library for support vector machine. To use libsvm data needs to be in the svm format or vector format. This training file is then supplied to libsvm svm-train.exe which trains the machine based on the content.

d) Now for testing similar file as above is generated. This test file is then given as an input to libsvm svm-predict.exe. Svm-predict.exe generates .model file and .out file which contains the output of testing data.

It gives both good accuracy and less time complexity.

D. Results

TABLE I. ACCURACY OF ALGORITHMS

Paper Title	**Author**	**Publication**	**Algorithm With result dataset**
Email classification with co-training	Svetlana Kiritchenko and Stan Matwin	Unpublished	Naïve Bayes 80.36% (web pages)
Improving KNN-based e-mail classification into folders generating class-balanced datasets	Pablo Bermejo	IPMU'08 pp. 529-536 in June 2008.	K-NN 63.67% Authors personal mail box
Active Learning to Classify Email	Bryan Klimt, Shyamsundar Jayaraman	Unpublished	SVM 90% Enron Corpus

III. PROPOSED WORK

We have discussed a number of different machine learning algorithms till now which include only supervised learning. We have discussed only advantages of all these algorithms in brief. But each of them has their own limitations. Any machine learning algorithm is evaluated on the basis of time it consumes to categorize a particular mail and the accuracy with which it classifies that email. Naïve Bayes is mainly based on probability calculations thus it tends to give more importance to noise than actual words. Noise is the main factor disturbing the accuracy of the algorithm. As the dataset increase results in more number of features. These more number of features result in smaller probability values which are not worth comparing.

If we consider Knn algorithm it is expensive and slow algorithm. To determine the nearest neighbor of a new point x, must compute the distance to all m training examples. The runtime performance of an algorithm is slow, but it can be improved. Incorrect initial labeling of documents may lead to the misclassification.

To overcome the above mentioned problems we have come up with the solution of concentrating on actual words and phrases which are real contributors for the particular category rather than all the available data available in the email. To achieve this goal Keyword based approach was the best solution available which would result more accuracy as well as very less time complexity. Considering this it is important to note that it is supervised learning model. The machine learning algorithms which are used for text categorization consider all the words for classification. But in emails all the words do not contribute towards the categorization although some keywords do. Thus there is need for Keyword based approach.

Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar:

A. Keyword based approach

In keyword based classification the list of keywords are represented, which should capture the category meaning. Classification is based on measuring the similarity between the list of words and the document.

The separate documents for lists of words are created. The words in those documents are collected such as they are related to the categories Meeting, Reply and Informative. The list contains the words and phrases which can distinctly separate the mail from other categories.

When any new mail comes the algorithm checks that mail with the each list of words. The algorithm already has the separate files containing the keyword's list. Algorithm calculates the number of keywords present in the mail. This calculation is done for the files containing keywords of Meeting and keywords of Reply as well as informative keywords. Then the counts of all the calculation are compared and according to comparison category for the mail are decided. If calculation shows the count=0 then by default the mail gets classify into other category.

B. Algorithm

a) Maintain a list of keywords for each category

Reply= {f1, f2, f3, f4, f5}
Meeting= {f6, f7, f8, f9, f10}
Informative= {f11, f12, f13, f14, f15}

b) Consider input document D which is to be categorised into one of the class C.

c) Let us consider input document as

D= {f1, f2, f7, f15}

d) Where f1, f2 € Reply
f7 € Meeting
f15 € Informative

e) Thus weight Wi, Wm, Wr can be assigned to for Informative, meeting and reply respectively.

f) In our case Wr=2, Wm=1 and Wi=1.

g) Thus category C for the document D can be given by max function as

M= (Wr, Wm, Wi)

h) The category associated with returned value from max function is the category, document belongs to. Here (D € Reply)

C. Addition of new category

As of now we have discussed only three categories but what if user wants to have his own category based on his requirements. In such cases we are providing this functionality for user to create their own category based on some keywords which will either be supplied by user or extracted from the mails which user opts to classify in this new category. This provides flexibility to user to classify mails according to their own convenience.

IV. EXPERIMENTAL SETUP AND TEST PLAN

To test the performance of each algorithm a sufficient large Dataset is needed and equally rich dataset for testing results is expected. We collected most of our dataset which is specially made available on the Enron corpus for study of machine learning domain and sorted out those mails according to our need. We could collect as much as 100mails for Reply, 360 mails for Informative and 250 mails of meeting category. After collection of mail we processed those mails by removing all the unwanted content and keeping only recipient id, subject, date and content. We supplied 70 mails of each category for training and rests of the mails were used for testing.

If the documents are not well structured, the experiments may not yield authorized results. To do this Pre-processing of each mail plays an important role. It includes processes such as removal of punctuations, removal of stop words, stemming (Eg "Walking" becomes "walk"). Pre- processing helps in reducing the processing time of the algorithm.

We use the underlying algorithm SVM using the library support by LIBSVM coded in Java platform. We run our experiments on following system configurations:

TABLE II. SYSTEM REQUIREMENTS

Component	Recommended Specification	Need
Processor	2.5 GHz	As these algorithms require a heavy work load to be handled.
Operating System	Windows 7	As this application is developed for both desktop as well as mobile phone.
Memory	4 GB RAM	Simultaneous operations will lead to slowing down of the system; as a result an adequate memory will be required for smooth functioning
Software	Eclipse Kepler Android sdk WEKA	For both java coding as well as android application development. For android platform set up. To get a brief overview of algorithm

V. RESULTS AND OBSERVATIONS

On the basis of experimental setup described in section IV we tried to verify performance of all discussed algorithms by considering two major factors: 1.Accuracy of the classification and 2.Time complexity. The results of each algorithm are as follows:

TABLE III. EXPERIMENTAL ACCURACIES OF ALGORITHM

Category	Training Mails	Testing mails	Correctly classified	Accuracy In %
Knn				
Info	70	280	180	57.14
Meeting	70	180	100	55.55
Reply	70	30	18	60
Naïve Bayes				
Info	70	280	87	53
Meeting	70	180	172	96
Reply	70	30	16	53.33
SVM				
Info	70	289	203	70.93
Meeting	70	658	595	91.18
Reply	70	30	11	36.66
Keyword Based Approach				
Info	-	280	250	89
Meeting	-	180	156	87
Reply	-	30	29	98

Accuracy of each algorithm respective to each category and their overall accuracy are compared below:

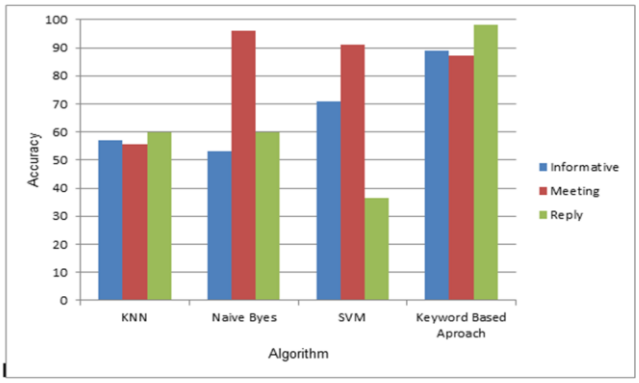

FIGURE I. ACCURACY OF ALGORITHMS ACC. CATEGORY

Overall accuracy of each algorithm is as illustrated below:

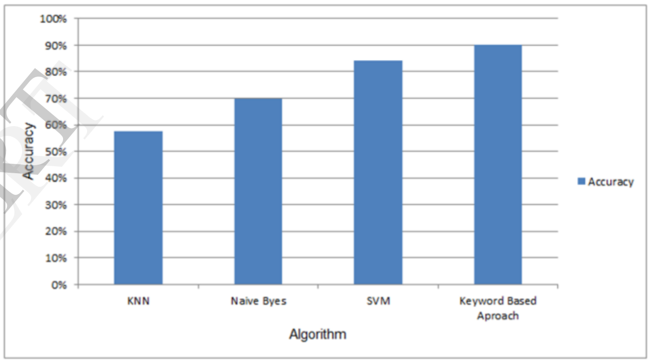

FIGURE II. OVERALL ACCURACY

Time complexity of each algorithm is also detailed below:

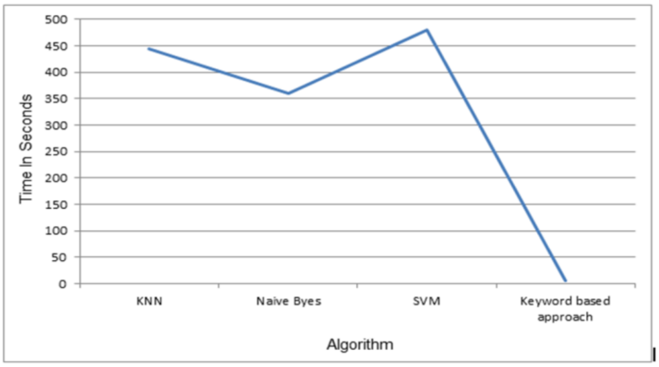

FIGURE III. TIME COMPLEXITY

VI.CONCLUSIONS AND FUTURE SCOPE

From the results discussed in section V we can conclude that the machine learning algorithms which are used for text categorization consider all the words for classification. But in emails all the words do not contribute towards the categorization although some keywords do. Thus there is need for Keyword based approach for multi class email classification. Machine Learning algorithms have a very high possibility of noise to disturb the classification. This might be due to individual preferences as each individual will have their own priority of what is important to them.

In addition to that these algorithms take comparatively more time than Keyword based approach. Thus based on our study we can conclude that Keyword based approach is best suitable for multi class email classification.

REFERENCES

[1] Svetlana Kiritchenko and Stan Matwin "Email Classification with Co-Training" Unpublished

[2] Yiming Yang, Jan O. Pedersen "A comparative study on Feature selection in Text categorization" In Mellon University Pittsburg.

[3] Pablo Bermejo, Jose A. Gamez, Jose M. Puerta, Roberto Uribe "Improving KNN-based e-mail classification into folders generating class-balanced datasets" Proceedings of IPMU'08 pp. 529-536 in June 2008.

[4] Bryan Klimt, Shyamsundar Jayaraman, Yiming Yang "Active Learning to Classify Email" Unpublished.

[5] Jose M. Carmona-Cejudo, Gladys Castillo, Manuel Baena-Garcia, Rafael Morales-Bueno "A comparative study on feature selection and adaptive strategies for email foldering using the ABC-DynF framework" In University of Malaga, Spain 2013.

[6] Bo Yu a,*, Zong-ben Xu b "A comparative study for content-based dynamic spam classification using four machine learning algorithms" In Knowledge-Based Systems 21 (2008) 355–362.

[7] Sharma Chakravarthy, Aravind Venkatachalam, Aditya Telang "A Graph-Based Approach for Multi-Folder Email Classification" Unpublished.

[8] Alfred Krzywicki and Wayne Wobcke "Incremental E-Mail Classification and Rule Suggestion Using Simple Term Statistics" A. Nicholson and X. Li (Eds.): AI 2009, LNAI 5866, pp. 250–259, 2009.

[9] Kristof Coussement, Dirk Van den Poel "Improving customer complaint management by automatic email Classification using linguistic style features as predictors" Decision Support Systems 44 (2008) 870–882