Classification of Email Using Naïve Bayes And Hiddden Markov Model Algorithm

S Vino David[1], Rashmi Kr[2], Sandhya V[3], Rakshitha P[4], Srinvas Achar[5]
[1, 2, 3, 4, 5] Atria Institute of Technology, Anandnagar, Bangalore

Abstract- *This project investigates a comparison between 2 completely different approaches for classifying emails supported their classes. Naive Thomas Bayes and Hidden Markov Model (HMM).Two completely different machine learning algorithms, each are used for detection whether or not AN email is vital or spam.*

Naive Thomas Bayes Classifier relies on conditional possibilities, it's quick and works nice with little data set. It considers freelance words as a feature. HMM could be a generative, probabilistic model that gives North American nation with distribution over the sequences of observations. HMM's will handle inputs of variable length and facilitate programs return to the foremost possible call, supported each previous selections and current information. Varied mixtures of IP techniques-stop words removing, stemming, summarizing are tried on each the algorithms to examine the variations in accuracy additionally on notice the simplest methodology among them.

Keywords- Email Classification, Hidden Markov Model, Naive Bayes, Natural Language Processing, NLTK, Supervised Learning

I. INTRODUCTION

Email is one in all the foremost vital means that of communication in today's world. Email usage has multiplied considerably round the world. In 2015, the quantity of emails sent and received per day destroyed over 205 billion. This figure is predicted to grow at a median annual rate of three over succeeding four years, reaching over 246 billion by the top of 2019. As of December 2016, spam messages accounted for sixty-one.66 % of email traffic worldwide2. Therefore, filtering these spam emails has become a crying want for email users round the globe. This paper describes the methodologies that may be wont to classify emails into totally different classes like vital and spam. Relative words or sentences are thought-about as feature to classify email messages. The distinction in nature between Naïve Thomas Bayes Classifier and Hidden Andre Markov Model makes it attention-grabbing to match them. Data set has been collected and reprocessed before evaluating accuracy, precision, recall, f-metrics for each algorithm. Stemming, summarizing, removal of stop words- these techniques are utilized in numerous mixtures with the algorithmic programs to research that algorithm on what combination provides the most effective result.

II. RELATED WORK

This paper focuses on identifying vital emails from spam emails. One major consider the categorization is that of a way to represent the messages. Specifically, one should decide that options to use, and the way to use those options to the categorization. M. Aery et al. [1]gave AN approach that relies on the premise that patterns is extracted from a pre-classified email folder and therefore the same is used effectively for classifying incoming emails. As emails consists a format within the sort of headers and body of the e-mail, the correlation between totally different terms is showed within the type graph. They need chosen graph mining as a viable technique for pattern extraction and classification. R. Islam et al. [2] showed the simplest way that projected a multi-stage classification technique victimization totally different fashionable learning algorithms like SVM, Naive Bayes and boosting with an analyser that reduces the False exactness well and will increase classification accuracy compared to similar existing techniques. B. Gustav Klimt et al [3] gave AN approach that introduced Enron corpus as a replacement dataset for this domain. V. Bhat et al. [4] came up with AN approach that derives spam filter known as Beaks.

They classify emails into spam and non- spam. Their pre-processing technique is meant to spot tag- of-spam words relevant to the data-set. X. Wang et al. [5] took an approach that reviews recent approaches to separate spam email, to categories email into a hierarchy of folders, And to mechanically confirm the tasks needed in response to an email. Consistent with E.Yitagesul et al [6], in sender based mostly detection, the e-mail sender info like the literary genre and therefore the email sender user name is employed because the major options. The analysis paper written by S.Teli [7] showed USA a 3 phased system that they designed for his or her approach of spam detection. Within the initial part, the user creates the rule for classification. Rules are nothing, however the keywords/phrases that occur in mails for several legitimate or spam mails. The second part is known as

coaching part. Here the classifier are going to be trained employing a spam and legit emails manually by the user. Then with the assistance of rule the keywords are extracted from classified emails. Once the primary and second phases are completed classifying the emails by given rule starts, victimization this data of tokens, the filter classifies each new incoming email. Here the likelihood of most keyword match is calculated and therefore the standing of a replacement email is confirmed as spam or vital email. Two main strategies for detection spam email are wide used. One is sender based spam detection and the other method is content based spam detection which will consider only the content of an email. This paper talks about the content based spam detection

III. TECHNIQUES FOR RETRIEVING RELEVANT INFORMATION

This paper discusses some techniques to eliminate impertinent knowledge from emails that are stemming, summarizing and stop-words removal. These techniques may be tried along in eight completely different mixtures and every one of them are experimented with. NLTK, one in all the dominant platforms for making Python programs, is meant to support analysis in linguistic communication or closely connected areas. It's varied text process libraries for classification, modernization, stemming, tagging, parsing, linguistics reasoning and wrappers for strong human language technology libraries, several of that are employed in this analysis. Stemming is that the technique of decreasing deviating or derived words to their base kind. For grammatical reasons, documents are progressing to use completely different varieties of a word, like meet, meets, and meeting. In several things, it's helpful for a look for one in all these words to come documents that contain another word within the set. Victimization stemming on the on top of strings, we'll get meet because the base kind [8]. Stemming chops off the ends of words. Algorithms for stemming are studied in applied science since the Nineteen Sixties. The foremost common and effective rule for stemming English is Porter's rule. Porter Stemmer [9] has been foreign from NLTK for stemming purpose.

Lemmatization is that the method of changing the words of a sentence to its lexicon kind. For instance, given the word's amusement, amusing, and amused, the lemma for every and every one would be 'amuse'. This aims to get rid of inflectional endings and to come base or lexicon style of a word. This method involves linguistic approach, like morphological analysis through regular relations compiled in finite-state transducers. Stop words may be a set of ordinarily used words in any language that are excluded out before or once process of linguistic communication knowledge that, during this case, is text. The most reason why stop words are essential to any program is that, once we take away the words that are terribly ordinarily employed in a given language, we will specialize in the necessary words instead. For removing stop words from an email in data set, we have a tendency to search them in NLTK toolkit's given list and therefore the result obtained was terribly correct.

IV. SYSTEM IMPLEMENTATION

1) Naive Bayes Classification

Bayes theorem [11] provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Let's look at the equation below:

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \quad \ldots\ldots\ldots$$

Above,

P(c|x) is the posterior probability of class (c, target) given predictor (x, attributes). P(c) is the prior probability of class.
P(x|c) is the likelihood which is the probability of predictor given class.

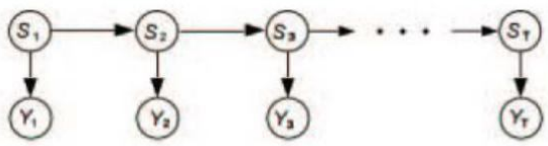
P(x) is the prior probability of predictor.

Bayesian Classification is used as a probabilistic learning method and every feature of this algorithm being classified is independent of the value of any other feature. The goal is to build a classifier that will automatically tag new emails with appropriate category labels. Now the classifier has a list of documents- emails labelled with the appropriate categories. The first step in creating a classifier is deciding what features of the input are relevant, and how to encode those features. So, a feature extractor for documents was defined so that the classifier knows which aspects of the data it should pay attention to. The duplicate words from those emails were removed. This made the checking faster. Now for every word in word_features, if that word existed in each email, it was tagged with the category (important or spam) of that email. Thus, words were found that were labelled as 'important' and 'spam'. These word: label pairs were used as feature set for Naive Bayes Classifier. At this point, there are words in feature set that are labelled as both important and spam. As feature extractor was defined earlier, it can be used to train the classifier to label new emails. Ninety percent of the

feature set was used as train_setwhile the remaining ten percent was used as test_set

2) Hidden Markov Model

HMM is a tool for representing probability distributions over sequence of observations. The HMM assumes that the observation at time t was generated by some process whose state St is hidden from the observer. It also assumes that the state of this hidden process satisfies the Markov property, which is, given the value of St-1, the current state St is independent of all the states prior to t-1. Graphically we can explain it as shown in figure below



The goal is same here- to build a classifier that will automatically tag new emails with appropriate category labels. Two states were used- important and spam. The list word_features were used as observations. And, start probability was set as {'important': 0.5, 'spam': 0.5}. Each word was searched for in both categories and its occurrence in either category was counted separately. Their probability of appearing in spam emails or important emails was used to create the emission probability set. In order to determine transition probability, for each word from spam and important emails, the probability of the next word being spam or important was recorded. HMM was used from sklearn module in python and the start probability, transition probability and emission probability were set. This way, important words and spam words could be identified.

V. RESULTS AND ANALYSIS

Table 5.1 shows that using basic Naive Bayes approach out of 100 important emails, 68 instances were classified correctly, 23 instances were classified incorrectly and 9 instances could not be determined. And out of 100 spams 69 instances were classified correctly, 15 instances were classified incorrectly and 16 instances could not be determined. The total accuracy achieved was 78.28%. Then again using only stop words, out of 100 important emails 83 instances were classified correctly, 13 instances were classified incorrectly and 4 instances could not be determined. And out of 100 spams 57 instances were classified correctly, 25 instances were classified incorrectly and 18 instances could not be determined.

Table 5.1: Evaluation on Test Set (Naive Bayes in DifferentProcesses)

Process	Trained data	Test data	Correctly Classified Instances		Incorrectly Classified instances		Indeterminate instances		Accuracy
			Impor--tant	Spam	Impor--tant	Spam	Impor--tant	Spam	
Basic	5500	200	68	69	23	15	9	16	78.28
Stop Words	5500	200	83	57	13	25	4	18	78.65
Stemming	5500	200	85	52	7	40	8	8	74.46
Lemmatizing	5500	200	72	58	16	29	12	13	74.29
Stop Words + Stemming	5500	200	74	58	14	30	12	12	75.00
Stop Words + Lemmatizing	5500	200	70	67	16	20	14	13	79.19
Lemmatizing + Stemming	5500	200	85	52	6	41	9	7	74.46
Stop Words+ Lemmatizing+ Stemming	5500	200	72	59	15	29	13	12	74.86

The total accuracy achieved was 78.65%. Using both stop words and summarizing 70 instances were classified correctly; 16 instances were classified incorrectly and 14 instances could not be determined. And out of 100 spams 67 instances were classified correctly, 20 instances were classified incorrectly and 13 instances could not be determined. The total accuracy achieved was 79.19% which gives us the best result in this comparison

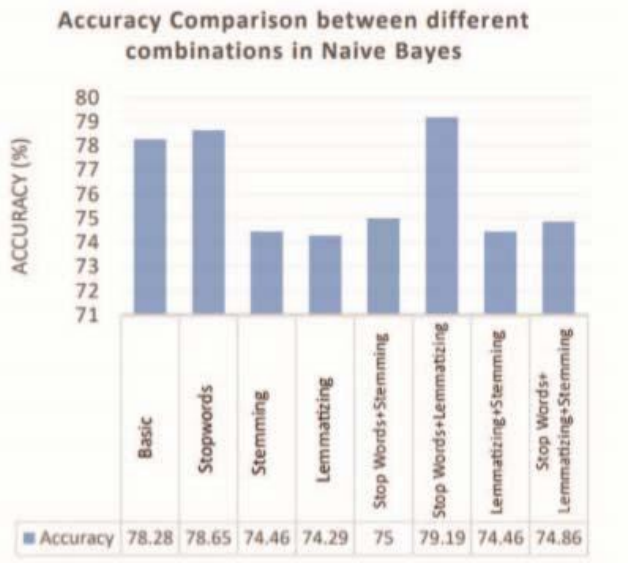


Figure 5.1: Accuracy Comparisons (Naive Bayes)

Figure 5.1 shows accuracy comparison among 8 different combinational approach to Naive Bayes and it shows that using stop words and lemmatizing together gives the best accuracy result which is 79.19%.

Table 5.2: Evaluation on Test Set (Hidden Markov Model in Different Processes)

Process	Trained Data	Test data	Correctly Classified instances		Incorrectly Classified instances		Indeterminate instances		Accuracy (Mail Classifi-cation)
			Impor--tant	Spam	Impor--tant	Spam	Impor--tant	Spam	
Basic	5500	200	80	81	17	11	3	8	85.19
Stop Words	5500	200	87	71	11	19	2	10	84.04
Stemming	5500	200	87	91	10	7	3	2	91.28
Lemmatizing	5500	200	83	77	15	17	2	6	83.33
Stop Words + Stemming	5500	200	91	74	7	22	2	4	85.05
Stop Words + Lemmatizing	5500	200	86	68	13	23	1	9	81.05
Lemmatizing + Stemming	5500	200	86	89	10	7	4	4	91.15
Stop Words + Lemmatizing + Stemming	5500	200	91	72	7	23	2	5	84.46

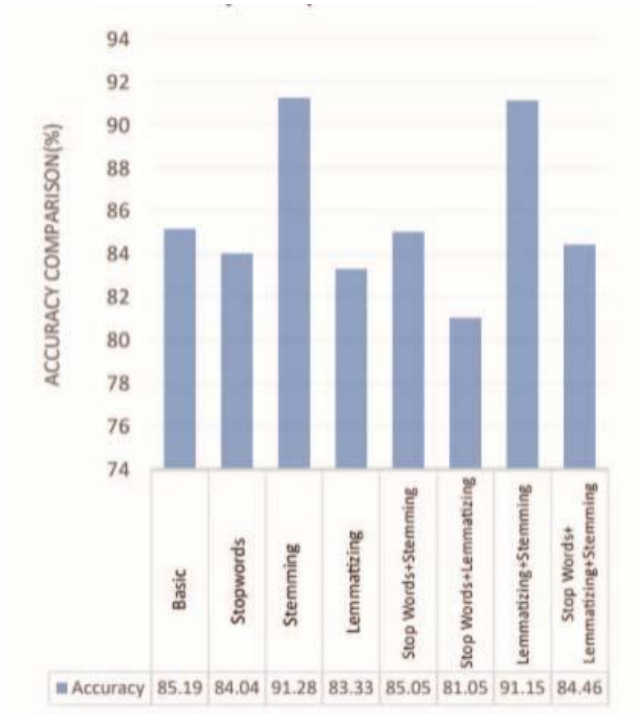


Figure 5.2: Accuracy Comparisons (Hidden Markov Model)

After running both Naive Bayes and HMM algorithm in 8 combinations of 3 different processes along with basic approach we find different classification accuracy for different

Table 5.3: Accuracy Comparison between Naive Bayes and HMM Algorithm.

Process	Accuracy (Mail Classification)	
	Naive Bayes Algorithm	HMM Algorithm
Basic	78.28	85.19
Stop Words	78.65	84.04
Stemming	74.46	91.28
Lemmatizing	74.29	83.33
Stop Words + Stemming	75.00	85.05
Stop Words + Lemmatizing	79.19	81.05
Lemmatizing + Stemming	74.46	91.15
Stop Words + Lemmatizing + Stemming	74.86	84.46

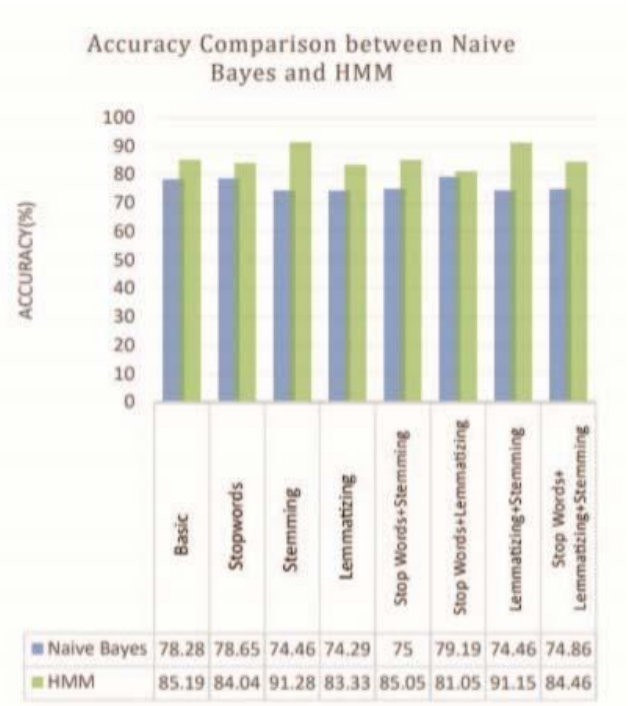


Figure 5.3: Accuracy Comparison between Naive Bayes and HMM Algorithm

VI. CONCLUSION

In summary, we have a tendency to propose a comparative approach to email classification victimization Naive Thomas Bayes Classifier and HMM. We have a tendency to reason emails by considering solely text half from body of the message. As a result of we have a tendency to take into account relative words and sentences as feature. Once running constant variants on each the algorithms, we have a

tendency to compare the results and used HMM for classification as a result of it gave higher accuracy. The structure of our analysis has been in-built such how that with correct data set and minor affray it will work to classify texts in any variety of classes

REFERENCES

[1] Jeffrey Dean, Sanjay Ghemawat, "Map Reduce: Simplified data processing on large clusters," Communications of the ACM, Vol.51, Issue.1, 2008, pp.107-113.

[2] Hadoop [Internet], http://hadoop.apache.org.

[3] Sudheesh Narayanan, "Securing Hadoop : Implement robust end-to-end security for your Hadoop ecosystem," 1st Vol, PACKT Publishing, 2014

[4] So Hyeon Park and Ik Rae Jeong, "A Study on Security Improvement in Hadoop Distributed File System Based on Kerberos," Journal of the Korea Institute of Information Security and Cryptology, Vol.23, Issue.5, 2013, pp.803-813

[5] Liu Yi, Hadoop Crypto Design [Internet], https://issues.apache.org/jira/secure/attachment/12571116 /Ha doop Crypto Desi gn.pdf

[6] Seonyoung Park and Youngseok Lee, "A Performance Analysis of Encryption in HDFS," Journal of KISS : Databases, Vol.41, Issue.1, 2014, pp.21-27

[7] Byeong-yoon Choi. "Design of Cryptographic Processor for AES Rijndael Algorithm," The Journal of The Korean Institute of Communication Sciences, Vol.26, Issue.10, 2001, pp. 1491-1500

[8] Yong Kuk Cho, Jung Hwan Song, and Sung Woo Kang, "Criteria for Evaluating Cryptographic Algorithms based on Statistical Testing of Randomness," Journal of the Korea Institute of Information Security and Cryptology, Vol.11, Issue.6, 2001, pp.67-76.

[9] ARIA Development Team, Block Encryption Algorithm

[10] ARIA [Internet], http://glukjeoluk.tistory.com/attachment/ok110000000002 .pd f.

[11] Korea Internet & Security Agency, ARIA specification [Internet], http://seed.kisa.or.kr/iwt/ko/bbs/EgovReferenceDetail.do? bbs Id=BBSM STR_000000000002&nttId=39&pageIndex=1&searchCn d= &searchWrd =.

[12] Jeffrey Root, Intel ® Advanced Encryption Standard Instructions(AESNI) [Internet] https://software.intel.com/en-us/articles/intel-advancedencryption- standard-instructions-aes-ni.

[13] Weizhong Zhao, Huifang Ma, Qing He, "Parallel k-means clustering based on mapreduce," In: IEEE International Conference on Cloud Computing. Springer Berlin Heidelberg, Vol.5931 p. 674-679, 2009.

[14] HuiGao, Jun Jiang, Li She, Yan Fu, "A New Agglomerative Hierarchical Clustering Algorithm Implementation based on the Map Reduce Framework," International Journal of Digital Content Technology and its Applications, Vol.4 Issue.3, 2010, pp.95-100

[15] [Internet] https://dumps.wikimedia.org/enwiki/[15] [Internet] https://dumps.wikimedia.org/metawiki/

[16] MODIS [Internet] https://modis.gsfc.nasa.gov/