Efficient Content Based Spam Filtering Using Bayesian Method
Himanshu Gupta1 Vasudha Arora2
1M.Tech Scholar 2Assistant Professor
1,2Department of Computer Science and Engineering
1,2Manav Rachna International University, Faridabad, Haryana

Abstract— There are two types of emails one in wanted emails from authorised users and other is unsolicited or unwanted emails, these unsolicited emails are called spams. This is a rapidly growing problem in the domain of emails these days. Emails are used by millions of people and send billions of mails daily all over the world. Over the last 1.5 decade it has become a very big problem. Every day a very huge amount of spam emails are received by the users. Due to this the business in not only this industry but also in every domain loses productivity, and it costs billions of dollars. It also strains the IT infrastructure. These emails become very frustrating for the users. So we need an efficient method to filter these mails and its necessity is increasing day by day. In this paper, we represent a method to do the same based on Naïve Bayes Classifier. It works by evaluating the probability of occurrence of a keyword in spam and in legitimate emails and classify by comparing them.

Key words: spam mails, non-spam mails, Naïve Bayes Classifier, learning dataset, laplacian, stop_words, ignore_words

I. INTRODUCTION

Electronic mail, also known as email or e-mail, is a method of exchanging digital messages from an author to one or more recipients. Modern email operates across the Internet or other computer networks. As the internet is becoming an important part of daily lives of not only business users but also of people from every domain with this the use of emails is also increasing. Also the number of spam emails is increasing day by day. We have many solutions for spam filleting despite that the spam mails are increasing rapidly. For every type of user the spam is not only becomes costly and frustrating but also a big security issue.

What is spam to a person may not be a spam to another. Unfortunately, the spam will be there. This issue is to be addressed at multiple levels at sender, at network and also at receiver’s level [1].

Undesired, unsolicited emails are not only frustrating to its recipients but also it is becoming a big security threat. You never know that if the received email contains some cookie grabbing codes, to know your access credentials (phishing) or it may be linked to a website that installs malicious software to your machine etc. [2].

There are severe problems with spam mails e.g. wastage of time, damage to your machines, spam emails advertising unethical content, and a major one wastage of network resources especially bandwidth [4].

Emails are most widely used mode of communication around the world as it is fast, cheap, reliable and easily accessible therefore it is also spam prone.

There are several methods to detect spam emails such as:

1. Handmade rules to detect spam mails made by experts, which requires domain experts and constant updating of these rules.
2. Blacklisting some particular senders. In this, it is hard to keep track of abuser senders.
3. Allowing emails from a specific domain only. It is hard to keep track of domains that are valid for a specific user.
4. Only receives the emails from the users which are present in the contact list, for this it is required that before sending email the email ids should be communicated.
5. Laws are implemented by the governments against these spammers.

Actually an automated learning algorithm implements a function given by:

\[ F (m, L) = \{ \text{spam if “spam”}, \text{else legi}. \]  

Where \( m \) = message or email,

\( L \) = a Vector of parameters and “Spam” and “legi” are labels assigned to the mail.

In order to classify a new message a machine learning algorithm reads or analysis the mails and their labels from previously collected data. It also checks the occurrence of a word in a spam and legitimate mails either separately or in groups.

This paper shows the similar technique but by using Bayesian techniques of classification. It calculates the probability of each word’s occurrence in a spam mail and non-spam mail using a training database made from past experience. For every word we calculate the probability of being a spam mail and a non-spam mail. If the probability of being a spam mail is greater than that of being a non-spam mail then this mail will be considered as a spam mail. After this the training dataset is updated in the database.

II. CHARACTERISTICS OF SPAM

To get rid of spam mails companies these days become very smart, they use various spam filtering mechanisms because of the obvious reasons of spams such as exploitation of network resources, loss of productivity, wastage of time etc. To deal with spams first of all we need to understand characteristics of spams. The spammers generally use some fixed patterns which become their characteristics. But spammers these days become very clever and use some other techniques to form spams. The proposed system is able to deal with all these types of spam mails.

Generally, the characteristics of spams appear in two parts of email first are header of the email and other is content of the message [7]. In the header there are various
types of characteristics. These characteristics and their statistics are shown in table below:

<table>
<thead>
<tr>
<th>Spam Characteristics</th>
<th>% of searched emails</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recipients address not in To: or Cc:</td>
<td>64%</td>
</tr>
<tr>
<td>To: field missing</td>
<td>34%</td>
</tr>
<tr>
<td>To: Field contains invalid email</td>
<td>20%</td>
</tr>
<tr>
<td>No message ID</td>
<td>20%</td>
</tr>
<tr>
<td>Suspect Message ID</td>
<td>20%</td>
</tr>
<tr>
<td>Cc: field contains more than 15</td>
<td>17%</td>
</tr>
<tr>
<td>recipients</td>
<td></td>
</tr>
<tr>
<td>From: is same as To:</td>
<td>6%</td>
</tr>
<tr>
<td>Cc: field contains 5-15 recipients</td>
<td>3%</td>
</tr>
<tr>
<td>To: field contains more between 5-15</td>
<td>2%</td>
</tr>
<tr>
<td>recipients</td>
<td></td>
</tr>
<tr>
<td>Cc: field contains more than 5-15</td>
<td>1%</td>
</tr>
<tr>
<td>recipients</td>
<td></td>
</tr>
<tr>
<td>Bcc: field exists</td>
<td>0%</td>
</tr>
<tr>
<td>To: field is empty</td>
<td>0%</td>
</tr>
<tr>
<td>From: is blank or missing</td>
<td>0%</td>
</tr>
</tbody>
</table>

Table-1: Statistics of Spam Characteristics in Email Header

But in Message Contents, spammers use some kind of words or keywords which companies use to filter spam mails. Some of these words are “click here”, “discount”, “safe”, “earn money” etc. Spams can be blocked by identifying such keywords but it should be done carefully as if not done carefully we can also block legitimate emails also.

It is advised to identify such keywords in groups by identifying patterns of their occurrence together to get better filtering.

III. DEVELOPMENT OF PROPOSED SYSTEM

The proposed system uses Naïve Bayes Classifier to implement the system which is based on Bayes theorem.

Bayes theorem states that: the probability of an event is dependent on the occurrence on another event which is associated with it. Mathematically it can be written as:

\[ P(A|B) = \frac{P(A \cap B)}{P(B)} \]  \hspace{1cm} (2)

\[ P(B|A) = \frac{P(A \cap B)}{P(A)} \]  \hspace{1cm} (3)

\[ P(A \cap B) = P(A)P(B) \]  \hspace{1cm} (4)

If we talk in terms of mathematical equations then our system will work like as follows:

This system uses Bayes theorem which says:

\[ P(cj | d) = \frac{P(d | cj) P(cj)}{P(d)} \]  \hspace{1cm} (5)

Considering each attribute and class label as a random variable and given a record with attributes (A1, A2, ..., An), the goal is to predict class C.

Specifically, we want to find the value of C that maximizes P(C | A1,A2,...An). The approach taken is to compute the posterior probability P(C | A1,A2,...,An) for all values of C using the Bayes theorem.

\[ P(C | A1 A2 ...An) = \frac{P(A1 A2 ...An | C) P(C)}{P(A1 A2 ...An)} \]  \hspace{1cm} (6)

So you choose the value of C that maximizes P(C | A1,A2,...,An). This is equivalent to choosing the value of C that maximizes P(A1,A2,...,An | C) P(C).

Naïve Bayes classifier requires that the conditional probability of every variable must be a non-zero else the overall probability will be zero.

\[ P(X | Ci) = P(X1,......,a | Ci) \]  \hspace{1cm} (8)

To get rid of above problem we predict the probability from Laplacian:

\[ P(A | C) = \frac{(Nc+1)/(Nc+c)} \]  \hspace{1cm} (9)

In our implementation we have assume the probability of non-existing word using Laplacian.

A. Assumptions Used In System

1. Sort the content and keyword dictionary according to language. Each word in learning dataset must have two variables associated with it first is occurrence in non-spam mails and other is occurrence in spam mails.
2. If a word doesn’t exist in the dataset than assume its probability using Laplacian.

IV. ALGORITHM USED

**Input:** keyword list, stop word list, ignore list, email message

**Algorithm**

- **Step 1:** Let M be an email message. Take two variables Nonspampercent and Spampercent initialized to 1. Convert all words to lower case.
- **Step 2:** Find out all special character from M and remove all special character from M.
- **Step 3:** For each word Wi in M
  - If Wi found in stopword list
    - Then W is removed from M.
  - End If.
  - End For.
- **Step 4:** For each word Wj in M
  - If Wj found in learning keyword dataset
    - Take spam value and calculate probability using spam value divided by total spam value and multiply this probability with Spampercent
    - Else Take spam value and calculate probability using spam value divided by total spam value and multiply this probability with Spampercent
    - Add to keyword dataset.
  - End If.
  - End For.
- **Step 5:** For each word Wk in m
  - If Wk found in learning keyword dataset
    - Take non spam value and calculate probability using non spam value divided by total non-spam value and multiply this probability with Nonspampercent
    - Else Take non spam value and calculate probability using non spam value divided by total non-spam value and multiply this probability with Nonspam percent.
    - Add to keyword dataset.
  - End If.
  - End For.
− Step 5: Find out spam and non-spam probability from learning dataset and multiply spam probability with Spam percent to get all word spam percent and similarly multiply non spam probability with non spam percent to get all word non spam percent
− Step 6: If Spam percent > Non Spam percent Then M will be identified as Spam Email. Else M will be legitimate Email

B. Pre-Conditions For The Algorithm
The pre-conditions for implementing the algorithm for the proposed system are to be imposed in the database. The pre-conditions that are to be imposed in the database of the system are:

1. Sort the content and keyword dictionary according to language. Each word in learning dataset must have two variables associated with it first is occurrence in non-spam mails and other is occurrence in spam mails.
2. Ignore_list<~,!,@,#,$,%,^,&,*,(,),_,..>. It the list of special characters that are to be ignored while operating on the content of the mail. To do this first we need to remove these words from the content then them operate on the rest of the content of the mail.
3. Stop_words<am, is, are, the, etc.>. It is the list of some common words that may possibly be available in most of the mails that are non-spam mails. These words are also being removed first so as to increase the efficiency and speed of the algorithm. These are the words that we use in the English language to make meaningful sentences such as vowels, verbs etc.
4. Keywords<bomb, hack, attack, etc.>. This is another requirement before implementing the algorithm is the set of keywords (malicious) that have the greater probability of their occurrence in the spam mails.

V. EXPERIMENTAL ANALYSIS
After implementing this system we have tested it on around 1000 spam emails and 200 non-spam emails to check how good it works i.e. to check its accuracy. It is a very important to check the accuracy of a system. It can be checked by some measurable entities such as sensitivity and specificity measures, these are used to measure the accuracy.

The evaluation measures that are used in this research are:

(1) True Positive (TP): It tells the no. of spam documents correctly classified as spam True.
(2) True Negative (TN): It tells the number of non-spam documents correctly classified as non-spam.
(3) False Positive (FP): It tells the number spam documents classified as non-spam.
(4) False Negative (FN): It tells the number non-spam document classified as spam.

And the measurements that we have used in this research are:

− Precision: It tells the percentage of positive predictions made by the system is actually correct. It is given by:

\[
\text{Precision} = \frac{TP}{(TP + FP)}
\]

− Sensitivity: It tells the percentage of positive labelled instances that is predicted as positive. It is given by:

\[
\text{Sensitivity} = \frac{TP}{(TP + FP)}
\]

− Specificity: It tells the percentage of negative labelled instances that is predicted as negative. It is given by:

\[
\text{Specificity} = \frac{TN}{(TN + FP)}
\]

− Accuracy: It tells the percentage of total number of predictions that are actually correct. It is given by:

\[
\text{Accuracy} = \frac{(TP+TN) \div (TP+TN+FP+FN)}
\]

After implementing this system we have tested it on around 1000 spam emails and 200 non-spam emails to check how good it works i.e. to check its accuracy. It is a very important to check the accuracy of a system. It can be checked by some measurable entities such as sensitivity and specificity measures, these are used to measure the accuracy.

After doing this experiment, we get to find that 943 emails are classified as spam and 160 are classified as non-spams out of 1000 spam emails and 200 non-spam emails respectively. So the measurements that we get are:

True Positive (TP): 943
True Negative (TN): 155
False Positive (FP): 57
False Negative: 45
Precision = 0.943
Sensitivity = 0.954
Specificity = 0.731
Accuracy = 0.915

VI. CONCLUSIONS AND FUTURE SCOPE
In this research we wanted to filter the spam mails from legitimate mails to get rid of all the problems associated with spams. We have found out the efficiency of this system by collecting some statistical data and by creating a dataset from it. It is based on Naïve Bayes classifier and we conclude that the mail filter based on Naïve Bayes classifier is good learning systems to filter spam mails from legitimate mails with an accuracy percentage of 91.5%. But it should be noted that the dataset should be continuously updating. Also, the accuracy of this system will depend on the volume of mails tested against the system.

We are working to find an efficient method to get the training dataset of keywords to automatically be updated so as to get better performance in this system by using machine learning techniques.

This method is very useful in separating spam mails from legitimate ones but a problem may arise in which there are mails from users of one domain which contains some words that are spam-words and the mails containing such words may be separated as spam mails but these words may not be harmful for the users of another domain such mails, these type of mails are called “gray mails”.

We are working to find an efficient method to detect these grey mails. Gray mail – messages which are not clearly spam or good mail – present significant obstacles to training and evaluating global spam filters. It also indicates key points where more personalized filtering is needed to handle different user preferences. Treating gray mail
campaign with different labels may be an efficient method to handle such mails.

REFERENCES


