# Email Filtering Using Bayesian Method

Nadia Al-Bakri

**Abstract:** Electronic mail is inarguably the most widely used Internet technology today. With the massive amount of information and speed the Internet is able to handle, communication has been revolutionized with email and other online communication systems. However, some computer users have abused the technology used to drive these communications, by sending out thousands and thousands of spam emails with little or no purpose other than to increase traffic or decrease bandwidth.

This paper evaluates the effectiveness of email filtering based on the Bayesian method to construct automatically anti-spam filters with superior performance. Bayesian e- mail classifier is trained automatically to detect spam messages. A test is performed on a large collection of personal e-mails taken from email server using POP3 protocol. The results had shown that using Bayesian method in filtering process yields an enhancement in filter performance.

**keywords:** Bayesian, Spam, Ham, Filter, E-mail, Naïve, pop3.

## 1-INTRODUCTION

Email spam known as unsolicited bulk Email (UBE), junk mail, or unsolicited commercial email (UCE), is the practice of sending unwanted email messages, frequently with commercial content, in large quantities to an indiscriminate set of recipients. Spam in email started to become a problem when the Internet was opened up to the general public in the mid-1990s. It grew exponentially over the following years, and today composes some 80 to 85% of all the email in the world, by a conservative estimate [1]. Pressure to make email spam illegal has been successful in some jurisdictions, but less so in others. Spammers take advantage of this fact, and frequently outsource parts of their operations to countries where spamming will not get them into legal trouble.

Attempts to introduce legal measures against spam mailing have had limited effect [2]. A more effective solution is to develop tools to help recipients identify or remove automatically spam messages. Such tools, called anti-spam filters, vary in functionality from blacklists of frequent spammers to content-based filters. The latter are generally more powerful, as spammers often use fake addresses. Existing content-based filters search for particular keyword patterns in the messages. These patterns need to be crafted by hand, and to achieve better results they need to be tuned to each user and to be constantly maintained a tedious task, requiring expertise that a user may not have [3].

The issue of anti-spam filtering was addressed with the aid of machine learning. A supervised learning method was examined, which learn to identify spam e-mail after receiving training on messages that have been manually classified as spam or non-spam.

Assistance lecturer in computer science department.
AL Nahrain University, Baghdad, Iraq.
Email: nadiaf_1966@yahoo.com

## 2-Types of Spam Filters

Spam filters work using a combination of techniques in order to filter through the messages and separate the genuine messages from the junk mail.

These techniques would rely on the following measures [4]:

- **Word lists** – Lists of words that are known to be associated with spam and are commonly found in unsolicited mail messages, such as 'sex' or 'mortgage'.
- **Blacklists and Whitelists** – These lists contain known IP addresses of spam senders (blacklists) and non-spam senders (e.g. friends and family). Therefore addresses that form part of the contact list are automatically registered as whitelist and any emails originating from these email addresses will be sent directly to the inbox.
- Trend Analysis – By analyzing the history of email sent from an individual, trends can help assess the likelihood of an email being genuine or spam.
- Learning or Content filters – Learning filters such as Bayesian filtering, examine the content of each email sent to and from an email address, and by learning word frequencies and patterns associated with both spam and non-spam messages, it is able to recognize which messages are valid and should therefore be directed towards the inbox, and which are spam and should be sent to Junk.

## 3-Classification of e-mail messages

We now turn to the learning algorithm we experimented with.

### 3.1 Naive Bayesian classification

Bayesian filter is a statistical technique of e-mail filtering. In its basic form, it makes use of a naive Bayes classifier on bag of words features to identify spam e-

mail, an approach commonly used in text classification. Bayesian filtering is based on the principle that most events are dependent and that the probability of an event occurring in the future can be inferred from the previous occurrences of that event. This same technique can be used to classify spam. If some piece of text occurs often in spam but not in legitimate email, then it would be reasonable to assume that this email is probably spam. Naive Bayes classifiers work by correlating the use of tokens (typically words, or sometimes other things), with spam and non-spam e-mails and then using Bayesian inference to calculate a probability that an email is or is not spam. It is one of the oldest ways of doing spam filtering, with roots in the 1990s [2].

From Bayes' theorem and the theorem of total probability, the probability that a document d with vector $\vec{x} = \langle x_1, \ldots, x_n \rangle$ belongs to category c is [5]:

$$P(C = c \mid \vec{X} = \vec{x}) = \frac{P(C = c) \cdot P(\vec{X} = \vec{x} \mid C = c)}{\sum_{k \in \{spam,\ legit\}} P(C = k) \cdot P(\vec{X} = \vec{x} \mid C = k)}$$

In practice, the probabilities $P(\vec{X} \mid C)$ are impossible to estimate without simplifying assumptions, because the possible values of X are too many and there are also data sparseness problems. The Naive Bayesian classifier assumes that n X1 … Xn are conditionally independent given the category C, which yields:

$$P(C = c \mid \vec{X} = \vec{x}) = \frac{P(C = c) \cdot \prod_{i=1}^{n} P(X_i = x_i \mid C = c)}{\sum_{k \in \{spam,\ legit\}} P(C = k) \cdot \prod_{i=1}^{n} P(X_i = x_i \mid C = k)}$$

P (Xi|C) and P(C) are easy to estimate from the frequencies of the training corpus.

A message is classified as spam if the following criterion is met:

$$\frac{P(C = spam \mid \vec{X} = \vec{x})}{P(C = legitimate \mid \vec{X} = \vec{x})} > \lambda$$

To the extent that the independence assumption holds and the probability estimates are accurate, a classifier based on this criterion achieves optimal results [6]. In our case,

$$P(C = spam \mid \vec{X} = \vec{x}) = 1 - P(C = legitimate \mid \vec{X} = \vec{x})$$

and the classification criterion is equivalent to:

$$P(C = spam \mid \vec{X} = \vec{x}) > t, \text{ with } t = \frac{\lambda}{1+\lambda},\ \lambda = \frac{t}{1-t}$$

Where t= threshold value and $\lambda$= number of spam messages.

## 4-Email Server Connection

### 4.1 POP3 (Post Office Protocol, version 3)

In computing, the Post Office Protocol (POP) is an application-layer Internet standard protocol used by local e-mail clients to retrieve e-mail from a remote server over a TCP/IP connection [7]. POP supports simple download-and-delete requirements for access to remote mailboxes. Although most POP clients have an option to leave mail on server after download, e-mail clients using POP generally connect, retrieve all messages, store them on the user's PC as new messages, delete them from the server, and then disconnect. Many e-mail clients support POP to retrieve messages.

## 5- The Proposed method Design

The design of the proposed method for email filtering spam messages is discussed below as phases:

### 5.1 Training the proposed E-mail filter

Before email can be filtered using this method, the user needs to generate a database with words. The following steps show the training process.

**A-Connect to the database (Microsoft Access)**

In this step, need to connect to the database by specify the provider of for type of database and the source (location) of database.

**B- Create database of spam and ham words**

1-Microsoft Office Access has been used to create 2 tables. The first table contains two fields (spam words collected from a sample of spam email recognize it as spam because of certain key words (such as "Viagra" and "mortgage") and its occurrences in spam messages and the second table contains the ham words and its occurrences in ham messages. Records for each field has list of some words as illustrated in table 1 and table 2.

Table 1 list of some spam words

| Spam Table | | |
|---|---|---|
| ID | Spam | Frequency |
| 1 | Viagra | 90 |
| 2 | Already!!!! | 100 |
| 3 | price | 40 |
| 4 | Amazing | 7 |
| 5 | drug | 40 |
| 6 | Ambitious | 7 |
| 7 | Amendment | 3 |
| 8 | Free!!! | 35 |

Table 2 list of some ham words

| Ham Table | | |
|---|---|---|
| ID | Ham | Frequency |
| 1 | dinner | 30 |
| 2 | department | 100 |
| 3 | college | 55 |
| 4 | today | 130 |
| 5 | yahoo | 20 |
| 6 | hello | 79 |
| 7 | come | 23 |
| 8 | tonight | 10 |

**Examples of training the filter on these short spam messages:**
Best quality drugs
Worldwide shipping
USPS - Fast Delivery Shipping 1-4 day USA
Professional packaging
100% guarantee on delivery
Best prices in the market
**Examples of training the filter on these short spam messages**:
Important meeting today at noon.
When is the next time you're coming home to visit?
Let's all meet at the diner for breakfast.

## 5.2 Connect to the server

A connection is needed to the server using (POP3) by specifying the server name, port, and security mode. The server name used is Yahoo, and the port is (995).

**Receive emails using POP3 code:**

```
Using pop3 As New Pop3()
pop3.Connect("pop3.filter.com")
pop3.Login("user", "password")
Receive all messages and display the subject
Dim builder As New MailBuilder()
 For Each uid As String In pop3.GetAll()
 Dim email As IMail = builder.CreateFromEml( _
 pop3.GetMessageByUID(uid))
 Console.WriteLine(email.Subject)
 Console.WriteLine(email.Text)
  Next
pop3.Close()
End Using
```

## 5.3 Parse words of current message

The proposed filter will split the message into tokens and build a table of all the tokens it intends to use in the decision making process. Tokens are taken from the body and subject of email. This filter uses single words in the calculations to decide if a message should be classified as spam or not. For each message retrieved from the server, each word is gotten separately by using Regex (regulator Expression).

## 5.4 Elimination of stop words

After initial indexing, it will be discovered that the document index contained useless terms, to decrease the number of terms in the index; it is desired to be filtered by removing stop words. a number of (1500) words is suggested as stop words, including the ordinary stop words similar to "the", "which", "is", and numbers. Also an extracted or suggested stop words similar to "repeat", "high", "width", "second", "first".

## 5.5-Calculate the number of iterated words

1- Find matched words of current email message with spam and ham words found in database.
2-Find how many times does word of current message has been iterated for both spam and ham words.

## 5.6 Bayesian Filter Process

In this step will apply the Bayesian filter. For each iterated words (spam and current message) divided by the number of total messages.
The formula used by the proposed method which is derived from Bayes' theorem:

$$\Pr(S|W) = \frac{\Pr(W|S) \cdot \Pr(S)}{\Pr(W|S) \cdot \Pr(S) + \Pr(W|H) \cdot \Pr(H)}$$

Pr(S|W) is the probability that a message is spam
Pr(S) is the overall probability that any given message is spam
Pr(W|S) is the probability that the word appears in spam message
Pr(H) is the overall probability that any given message is not spam ( is ham)
Pr(W|H) is the probability that the word appears in ham message.

## 5.7 Calculate the Spamicity

The email filter calculates the word's spamicity and the probability of spam message as shown in the following pseudo code:

**Create table 3 in database with 4 fields, first field to words of current message, spam probability, ham probability and spamicity value of each word.**
**For each word in current message**
**Store word in table 3.**
**If word is stop word then Read next word.**
**Read the frequency of the word in spam table.**
**Read the frequency of the word in ham table.**

IJSER

**If word frequency < =2 then Spamicity=0.4**
**Numerate number of spam messages the filter has been trained on.**
**Numerate number of ham messages the filter has been trained on.**
**Ham probability = frequency of word in ham table / Number of ham messages trained on.**
**Spam probability = frequency of the word in spam table / Number of spam messages trained on.**

**If Ham probability > 1 then**
**Ham probability=1**

**If spam probability > 1 then**
**Spam probability=1**

**Word Spamicity = Spam probability / (Ham probability + Spam probability)**

**Store Word word Spamicity in table 3.**

**Until end of words in current message.**

**Choose 30 words from table 3**

**The spam probability of current message= multiplication of 30 words spamicity/ (multiplication of 30 word spamicity) multiplied by (1-spamicity) for each word.**

**If probability of current message > 0.5 then**

**Spam=Current message**

**Else**
**Ham=current message**

## 6- Result

To validate the proposed filter, a corpus of 1700 actual e-mail messages of which 900 messages are pre-classified as junk and 800 messages are pre-classified as legitimate were conducted. A result was shown that the proposed filter worked more efficiently than other techniques like using public black and white lists. The proposed filter uses 30 most "interesting" words to calculate the message's overall spamicity. These 30 words are the words in the message that have either the highest or lowest spamicity (i.e. are closest to 0 or 1 in value). The spamicity value assigned to each word ranges from 0.0 to 1.0. A spamicity value of 0.5 is neutral, meaning that it has no effect on the decision as to whether a message is spam or not. The spamicity is based on the number of times a word occurs in spam messages as opposed to the number of times it occurs in non-spam messages. Table 3 shows the 4 field's generation.

| token | Spam probability | Ham probability | Spamicity |
|---|---|---|---|
| brother | 177 | 171 | 0.4037 |
| cash | 1318 | 49 | 0.8737 |
| contact | 1552 | 760 | 0.3445 |
| death | 118 | 37 | 0.451 |
| family | 3255 | 172 | 0.829 |
| friend | 456 | 110 | 0.516 |
| blood | 383 | 53 | 0.650 |

**Table 3 the 4 field's generation**

## 7- Conclusion

In examining the growing problem of dealing with junk E-mail, we have found that it is possible to automatically learn effective filter to eliminate a large portion of junk from a user's mail stream. It's also important that the email filter be trained on spam and non-spam messages from user inbox. If an email filter is pre-trained on messages from another site, it won't be able to identify features specific to messages destined for the user. This can easily lead to large numbers of false positives and low spam detection accuracy.
The accuracy of such filters is greatly enhanced by considering not only the full text of the E-mail messages to be filtered, but also a set of hand-crafted features which are specific for the task at hand.
A plan for future is to explore a method deals with phrases besides words.

## 8-Reference

1- https://en.wikipedia.org.
2- Androutsopoulos I., J. Koutsias, K.V. Chandrinos, and C.D. Spyropoulos. 2000b. An Experimental Comparison of Naive Bayesian and Keyword-Based Anti-Spam Filtering with Encrypted Personal Messages. Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Athens, Greece.
3- Cranor, L.F. and B.A. LaMacchia. 1998. Spam! Communications of ACM, 41(8):74–83.
4- Spector, Lincoln. "Guide to Spamming the Spammers". About.com.

5- Friedman, N., D. Geiger and M. Goldszmidt. 1997. Bayesian Network Classifiers. Machine Learning, 29(2/3):131–163.

6- Duda, R.O. and P.E. Hart. 1973. Bayes Decision Theory. Chapter 2 in Pattern Classification and Scene Analysis, pages 10–43. John Wiley.

7- Dean, Tamara (2010). Netw ork+ Guide to Networks. Delmar. p. 519.

IJSER