Çankaya Üniversitesi Fen-Edebiyat Fakültesi,
Journal of Arts and Sciences Sayı: 11 / Mayıs 2009

Detecting Spam by Weighting Message Words

Mousa Abdoh[1]*, Mohammad Musa*[2]*, Nael Salman*[1]

Abstract

The huge number of spam e-mail received daily by users account, made the necessity of existence of some sort of automated spam filter to detect and remove these unwanted e-mails. Most of the existing spam filters are based on naïve Bayesian methods.

The work presented in this paper introduces a new automated filter based on naïve Bayesian method. The basic idea of this filter is to give each word appears in e-mails a weight based on its frequency in both spam and legitimate mails. This weight value indicates its probable belongings to spam or legitimate. The proposed filter has a preprocessing component which removes all common words.

In the training phase a set of 1300 e-mails (legitimate and spam) has been used for giving weights for non common words.

The classifier uses the weight table generated in the training phase to classify a given e-mail as spam or legitimate. During testing we used 400 e-mails, 200 of them are spam and 200 of them are legitimate, the proposed algorithm achieved a 95% rate of accuracy.

Keywords*:* Spam, Word frequency, Word weight, Classification.

1. Introduction

The internet opened many new channels of communication. One of these is the e-mail communication channel. Initially, e-mail was used only for communication between different persons. Later, e-mail began to be used for advertising products of several types. Many of these products conflict with our Arabic culture, like advertising for Viagra and some sexy products. Such e-mails are called unwanted e-mails or bulk e-mails which are also referred to by the term "SPAM". Vast majority of e-mail users suffer from receiving

1 Palastine Technical University - Kadoorie, Tulkarm, Palestine
e mails: mabdoh@ptuk.edu.ps (M. Abdoh), n.salman@ptuk.edu.ps (N. Salman)

2 Sudan University for Science and Technology, Khartoum, Sudan
e mail: hafiz85@hotmail.com

large numbers of unwanted e-mails. Therefore e-mail users need to find some ways to block these spam e-mails.

The term spam refers to all e-mail messages that contain unwanted advertisements of specific products, other types like e-mails with pornographic content, or viruses.

Some recent studies shows that about 73% of the e-mails sent daily are SPAM. [7]

This huge numbers of spam raise the necessity of automated filters to detect or remove these spam e-mails. In this paper we are developing and implementing an algorithm to handle the spam issue. Our algorithm is based on text classification techniques and methods. Our algorithm used Bayesian rules to find the probability of spam words and use these probabilities to find a value called weight (based on frequency) of word that can classify an e-mail to spam or legitimate.

In Section 2 we discuss the different definitions of the term spam, we provide a historical brief of the term, we discuss briefly the techniques and methods used by spammers to have e-mail lists. And we discuss anti spam techniques and in Section 3 we discuss our proposed algorithm and three stages in the algorithm, implementation of the algorithm, testing, in Section 4 we discuss the implementation stage, in Section 5 we discuss the testing stage according to spam messages type, in Section 6 we introduced a real example of our work, in Section 7 we discuss comparison with some similar filters, in Section 8 we discuss the results and finally in Section 9 we present our conclusions.

2. What Is Spam

There are many definitions for the term spam in the literature. The following are the most widely considered definitions of the term:

1. Spam is an e-mail that is sent to many e-mail users without requesting to receive these e-mails. [5]
2. Spam is unsolicited bulk e-mail (or "junk" e-mail), which can be either commercial (such as an advertisement) or noncommercial (such as a joke or chain letter). [1]
3. Spam is one or more unsolicited messages, sent or posted as part of a larger collection of messages, all having substantially identical content. [1]
4. Spam is: "sending nearly identical message to thousands (or millions) of recipients"
5. Spam is "Irrelevant or inappropriate messages sent on the internet to a large number of newsgroups or users." [2] as defined in the New Oxford Dictionary of English

From the definitions mentioned above we saw that most of these definitions treat spam as unwanted e-mail, and the last definition treat the spam as a message sent to

many recipients, We can conclude that a spam is any e-mail that is sent to large number of users and unwanted to almost all of them.

2.1 Spam History

The term spam is firstly derived from the 1970 Monty Python SPAM sketch, set in a cafe where nearly every item on the menu includes SPAM luncheon meat. SPAM was one of the few meat products that avoided rationing, and hence widely available. [2]

Although the first spam message had already been sent via telegram in 1864, and the first spam message of commercial e-mail occurred in 1978, the term spam for this practice had not yet been applied.

In the 1980s the term was adopted to describe certain users who frequented BBSs (A Bulletin board system, is a computer system running software that allows users to dial into the system over a phone line or Telnet), who would repeat "SPAM" a huge number of times to scroll other users' text off the screen in early chat rooms services like the early days of AOL. [3]

Spam mail later came to be used on multiple posting—the repeated posting of the same message. The unwanted message would appear in many if not all newsgroups, the first usage of this sense was by Joel Furr in the result of the ARMM incident of March 31, 1993, in which a piece of experimental software released dozens of recursive messages onto the news.admin.policy newsgroup.

Commercial spamming started in force on March 5, 1994, when a pair of lawyers, Laurence Canter and Martha Siegel, began using bulk Usenet posting to advertise immigration law services. The incident was called the "Green Card spam".

Within a few years, the focus of spamming (and antispam efforts) moved to e-mail, where it remains today

2.2 How Spammers Collect E-mails?

An e-mailing list means the list of names and e-mail addresses which collected by some individual or an organization. [1]

2.2.1 Harvesting

Harvesting is the simplest method that spammers can purchase or trade lists of e-mail addresses from other spammers. But there are other methods that spammers can bring e-mail lists. One of these methods is "harvesting bots" which spider web pages, posting on Usenet, mailing list archives and other sources to obtain the e-mail list. Another way to harvest e-mail list is the directory harvest attack, where valid e-mail addresses at known domain can be found. Spammers use the directory attack in order to get the e-mail list. Spammers can also harvest e-mails list from a number of other sources. [9]

2.3 Why Bayesian Filtering

Bayesian algorithm is based on the probability of an event occurring in the future, that can be measured from previous occurrence of that event. The same technique can be used to classify e-mail to spam and legitimate, if a word occurs in spam mails and not in legitimate e-mails, then it would be reasonable to assume an e-mail containing this word is probably a spam. Some of the reasons that consider the Bayesian algorithm as a powerful classifying method are that the Bayesian method takes all the message into account, by measuring the words that identify spam and legitimate mails. [8]

2.4 Anti Spam techniques.

2.4.1 White lists

White lists made e-mail users to classify or define “trusted” or good addresses that will be known as legitimate. These e-mail addresses were flagged as legitimate addresses, so any e-mail received from these addresses would be classified as legitimate. [6]

2.4.2 Blacklists

Blacklists, which are also known as domain *name system black lists* (DNSBL), can classify e-mail as spam or legitimate according to the domain that have sent spam e-mails before. These addresses will be in a list and all the messages sent by these address would be blocked. [6]

2.4.3 Content Filters

Content based filters can classify the e-mail into spam or legitimate by examining inside the e-mail contents, most of these content based filters look for words or sentences that refer to spam like sex, Viagra, by now, you have won, and other determinates. [6]

The problem with these types of filters is that the spam is always increasing so in black and white lists these lists will daily increase and to classify the e-mail into spam or legitimate will take more time than other filters. Other problem with these types of filters that some messages from the same black addresses may be sent and would be classified as spam while they are legitimate.

The problem with the content based filters that these filters look for certain words or messages, and the spammers always find new techniques to send their messages, for example instead of writing Viagra they may write it as v-i-a-g-r-a, or these e-mails may not contain the word Viagra but contain other words like visit the pharmacy or some other advertising sentences, that the classifier would not classify these messages correctly. The messages may contain some spammed words while they has been used in legitimate e-mail, so the classifier will classify them as spam, like using Viagra word in e-mail sent from a doctor explaining the danger of Viagra.

Our classifier is content based filter, but works not like these described above, but weights each word in the e-mail and according to the weights it will be classified as spam or legitimate, as described in the next chapter.

3. The Proposed Algorithm

We built an algorithm based on Bayesian rules of conditional probability method. The algorithms are composed of three stages. These stages are: Preprocessing, training and classification.

3.1 Preprocessing Stage

As it is the case in every data mining tool, preprocessing is an essential component in our algorithm. Preprocessing is preparing the data for learning. In our algorithm preprocessing is achieved by two stages. These stages are creating a dictionary of common words (pronouns, prepositions, articles …) and the second stage is removing (cleaning) these words from the downloaded e-mail documents. These steps can be explained as follows:

1) Creating the dictionary: The step is performed manually. We prepare a file containing all common English language words. By a common word, we mean a word that is believed to appear in all kinds of e-mails with similar frequencies. Examples of common words can be, but not limited to, the prepositions "in, on, at …" and the pronouns "he, she, it …". These words can be encountered is all kinds of e-mails with the same frequency.

Removing these words from e-mails before trying to learn from these e-mails is essential to speed up the process of learning without effecting affecting the filtering negatively.

2) Removing common words: Using the dictionary prepared in Step 1, in this step, we remove all these common words from the original e-mail documents. In this step our algorithm is self adapting, that is if the spammers used V-I-A-G-R-A instead of Viagra, our algorithm will notice that by removing the special characters that can be added to any word

As the common word list is sorted, and the algorithm uses binary search. The algorithm has $O(m \log n)$ time complexity where n is the number of words in the dictionary and m is the number of words in the e-mails document (training set). The search in this step is a binary search. This complexity is acceptable and will give excellent impact on the coming work that will speed up in late stages. The output of this stage is a new mail document was all the common words (words in the dictionary) are removed. This file will be used as the input for the second stage in which the actual learning process takes place. Figure 1 illustrates this stage.

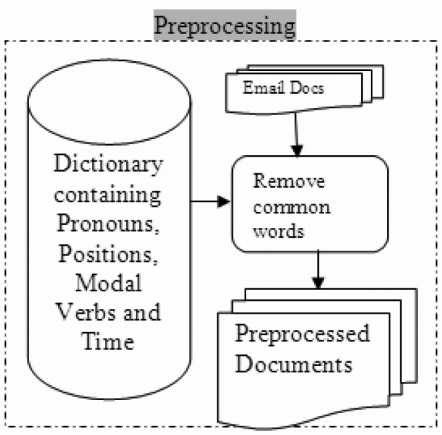

Figure 1: Preprocessing Stage.

3.2 Training Stage

This stage is the heart of our algorithm. In this stage we build our classification rules based on the knowledge we gained from the training documents gathered from many free e-mails like yahoo, gmail, and hotmail. In this stage our program will learn how to classify documents into spam and legitimate (that we will refer to not spam or non spam). The training is totally base on the documents' content. So, our algorithm learns from the existing documents by performing the followings steps as in Figure 2:

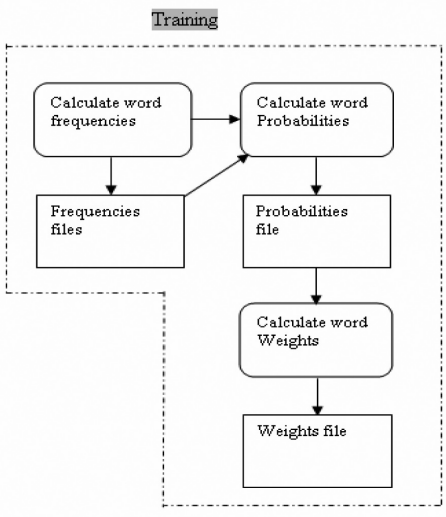

Figure 2: Training Stage.

1) Estimate word probabilities that depend on frequencies: In this step our training module uses the output file produced in the preprocessing stage and estimated word frequencies and probabilities. Word frequency means, the number of occurrences of a specific word in the document. We initialize a counter to zero and it will be incremented once the word is next encountered. This complexity of this algorithms is ***O (n)*** where ***n*** is the number of words in the document. Estimating probabilities is achieved using Bayes conditional probability theorem according to which the probability of a word given that the message is spam can be estimated as follows:

$$Ps = \frac{\frac{Fs}{Cs}}{\frac{Fns}{Cns} + \frac{Fs}{Cs}} \tag{1}$$

And

$$Pns = \frac{\frac{Fns}{Cns}}{\frac{Fs}{Cs} + \frac{Fns}{Cns}} \tag{2}$$

Where:

Ps is the probability of a word given the mail is spam.
Pns is the probability of a word given the mail is legitimate.
Fs is the frequency of word in the spam documents.
Fns is frequency of words in the legitimate documents.
Cs is the count of spam documents.
Cns is the count of non-spam mail documents.

The output of this step is a new text file containing a list of words with their frequencies and probabilities.

2) Calculate word weights: In this step we estimate a weight for each word based on its frequency and its probability in spam mail documents and non-spam mail documents, computing each word weight will give this feature more reliable effect than other algorithms that based on classifying the message according to some words and phrases that exist in the spam messages. This step uses the output file produced in the previous step. The weight of every word is estimated using the formula:

$$Weight = \frac{Ps}{Pns} \tag{3}$$

This weight value is calibrated based experimental results on a set of mail documents containing spam messages and non-spam messages.

The algorithm of the program calculating the word weights is described in Figure 4.

This complexity of this algorithm is ***O (n)*** and it is tested with a file. After the cleaning process (in which all common words are removed) the generated file will be used as input to the next stage.

A message is classified as a spam message if the average weight of all words inside the message is greater than or equal to 1 (which means approximately each word has a weight greater than or equal to 1). The word weights are estimated during the training process and stored in a separate text document. If a word in the mail document and does not exist in the weights file then it is given a weight of ZERO.

In this stage we collected the data set from the many free e-mail providers like yahoo, hotmail, and gmail. The number of e-mails collected is 1300, 650 of them is spam and 650 of them is legitimate, this data set is collected manually from these e-mail providers and we read them and classify them to be accurate that these e-mails are classified correctly to spam and legitimate.

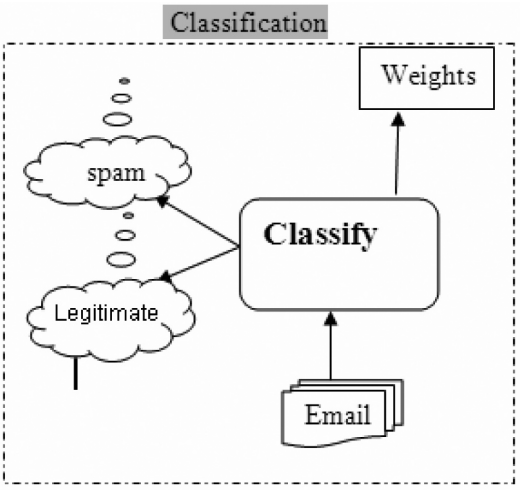

Figure 3: Classification Stage.

```
wordCount = 0
TotalWeight =0
While NOT EOF(e-mailDocument)
Read A word
wordCount = WordCount + 1
      While NOT EOF(frequencies)
      Read CurrentWord, CurrentWeight
     If  word = CurrentWord
     TotalWeight = TotalWight+CurrentWeight
     END While
END While
Result = TotalWeight/WordCount
If Result > 1
Print Spam
Else
Print LEGITIMATE
END
```

Figure 4: The Classification Algorithm

3.3 Classification Stage

In this stage the actual classification of documents takes place depending on the average weight calculated. Our classifier removes the common words from the e-mail,

then finds the total number of words exists, and the sum of the words after the common words removal, by calculating the average weight of the message which is equal to total weight sum divided by the total words number.

$$\text{Average} = \frac{\sum weights}{\sum words}$$

If the average weight > 1 then the e-mail is classified as spam else it is classified as legitimate. Figure 4 illustrates this stage.

4. Implementation

The classifier is implemented using Java in three different classes each of these programs is an implementation of one stage of the classifier. The first stage is the cleaning stage, the program cleans all the words found in the commonwords file from all messages, the output of this stage is a new file after removing all the common words.

The program works fast enough. On a Pentium4 based PC the program can classify a mail document in a less than a second which can be considered relatively fast. This speed encourages for adoption of the program in mail servers where thousands of mail messages arrive every day.

5. Testing

We tested our algorithm with 400 e-mail, 200 of them are spam and 200 of them are legitimate, our classifier passed in detecting 94% of the e-mails tested and classified them to spam and legitimate.

The dataset that had been tested contains spam e-mails that belongs to many types described in Table1 and in Figure 5.

Table 1: E-mails used in testing

Spam type	**No of e-mails**
Advertising	75
Adults	66
Finance	50
Others	9

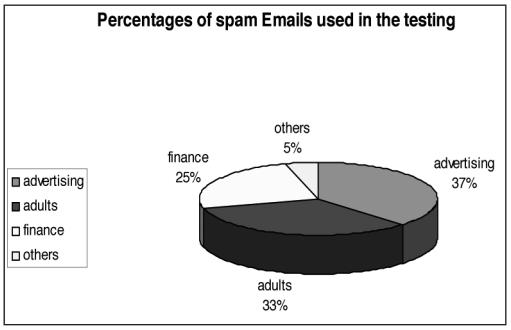

Figure 5: percentage of Spam e-mail types used.

The percentage of positive-negative of e-mails used in data set is 90%, and the percentage of negative- positive of the e-mails used is 88%. As shown in Figures 6 and 7.

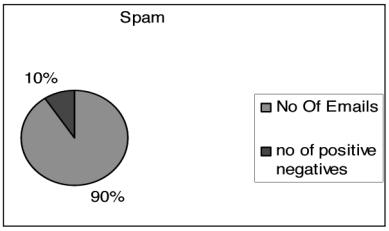

Figure 6: the positive – negative percentage.

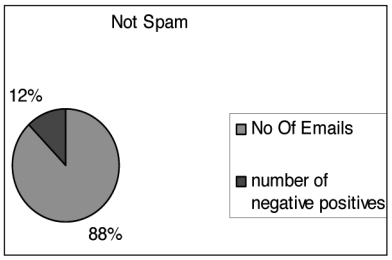

Figure 7: the negative –positive percentage.

At the total our algorithm achieved precision of 94%, and recall of 93%.

Keywords: positive – negative: the e-mails that are spam but classified as legitimate.

Negative – positive: the e-mails that are legitimate but classified as spam.

6. Real Example

The following example is one of the e-mails that had been tested from our dataset which is advertising e-mail, our classifier classified this spam e-mail as spam after calculating the total number of words in the e-mail and the sum of the weights of each word in the e-mail, then finding the average of the message weight, as illustrated in the classification stage of the algorithm. The spam words and their weights found in large table contain more than 2500 word and their weight, and it not possible to put this table

in this paper, this table is the result of the training process. The words and their weights of the following e-mail is part of this large table.

> ***.. Over 200 A-Z medications to choose from ..***
>
> ***We give you FreeViagraPills (Free you 12 pills with any order)***
>
> ***. ViagraPill***
> ***. Cialix Pills***
> ***. PenisGrowth Pack***
> ***. SQMA***
> ***. Phentrimine***
> ***. Levitr***
> ***. Tramadol***
> ***. FemaleViagra***
> ***. & 400 more meds to choose from***
>
> ***Claim your Free 12 ViagraPills here with us***
>
> ***http://kpi.hohisx.com***

Our classifier classified this e-mail as Spam, Table 2 contains all the words of this e-mail and their weights after common words removal.

The total sum of the weights is 234, and the total words count is 30, the average weight for the e-mail is 7.8, therefore the classifier result is

The e-mail is Spam.

7. Comparison with similar works

We compared our results with results obtained from similar works to check the importance of our approach. For this task we chose two spam classification tools. The first of these is Quick Spam Filter and SPATIC obtained from freahmeat.net [10].

The Table 3 contains comparison of our results with some of Bayesian based algorithms with approximately the same number of spam and legitimate e-mails that have been used in training and testing stages.

Table 2: words and their weights

Word	weight
Over	12
200	0
A-Z	0
medications	23
choose	13
Give	20
FreeViagraPills	0
Free	13
12	0
Pills	33
Order	0
ViagraPill	18
Cialix	0
Pills	33
PenisGrowth	0
Pack	0
SQMA	0
Phentrimine	0
Levitr	0
Tramadol	0
FemaleViagra	0
400	0
More	0
Meds	3
Choose	13
Claim	7
Free	13
12	0
ViagraPills	33
http://kpi.hohisx.com	0

Table 3: Comparison with other algorithms

Filter	**Precision**	**Recall**
Quick Spam Filter[10]	94.9%	79.1%
SPASTIC [10]	88.5%	43.3%
Our Algorithm	94%	93%

8. Discussion of results

We started our research by considering the spam e-mails and legitimate e-mails, we downloaded a dataset from many free e-mail providers, this data set contains 1300 e-mails for training stage, 650 of them are Spam and 650 of them are legitimate, and 400 e-mail of them for testing, 200 of them is spam and 200 of them is legitimate.

All the 1300 e-mail messages had been classified manually to spam and legitimate. Our classifier algorithm consists from three steps:

1 – The common words removal that removes any common word. All of these words grouped in one file called commonwords, from the files input1 that contains all the spam e-mails and input2 that contains all the legitimate e-mails, the output of this step is the files spam.txt file that contains all the words in input1 not found in commonwords.txt and legitimate.txt that contains all the words in input2 not found in commonwords.txt

2 – Training step, in this step the program reads the files spam.txt and legitimate. txt and computes the frequencies of each word, and then estimates the weight for each word.

3 – Classification step, in this step the program classifies a given e-mail as spam or legitimate according to the average weight of the e-mail words. The weight of each word is calculated according to probability of the given word in spam file and legitimate file.

This formula is to calculate the weight:

$$Weight = \frac{Ps}{Pns}$$

The average weight is the sum of the weights to all words divided by the number of words.

$$\text{Average} = \frac{\sum weights}{\sum words}$$

After testing 400 e-mails, 200 of them are Spam and 200 of them are legitimate, we calculated the precision and recall for this data set, the precision is 94%, and recall is 93%.

9. Conclusion

Because of the increasing number of spam messages, we started our research to detect these messages. To detect these spam messages we learned the spam messages features. From these features we focused on the text contents.

After reading these spam e-mails that our classifier classifies them as legitimate, we found that most of them contains words that does not have any weight, that is the 650 spam messages for training is not enough, We manually tried to put weights to some words in these e-mails, the classifier classifies them as spam.

Some of legitimate e-mails also have been classified as spam, because they contain some words that have a high weight as the words Viagra, and girls.

The algorithm described can be used in e-mail server and serve as a gateway filter that can mark or eliminate e-mail with potential spam content.

By comparing the results of our algorithm with other algorithms mentioned in Table3, our algorithm gives a better performance over both the versions of Bayesian based algorithms in terms of both spam precision and spam recall.

References

[1] http://www.monkeys.com/spam-defined/ visited on Apr, 29th, 2008

[2] The wikipedia webpage http://en.wikipedia.org/ visited on Nov, 5th, 2007

[3] Bayes theory webpage http://www.trinity.edu/cbrown/bayesWeb/index.html visited in Jan, 20th, 2008.

[4] The wikipedia webpage http://en.wikipedia.org/ visited on Nov, 5th,2007

[5] Hoffman,paul, "unsolicited Bulk – mail, definition and problems" UBE - DEF IMCR- 004 Oct, 5th,1997.

[6] Duncan Cook, Jacky Hartnett, Kevin Manderson and Joel Scanlan, "Catching Spam Before it Arrives".

[7] Spam Watches Webpage, http://spamwatchers.com/2008/03/20/facts-and-Figures-about-spam/ visited on Apr, 18th, 2008.

[8] Richard O. Duda, Peter E. Hart, David G. Stork, "Pattern Classification" Wiley – Inter science Publication, 2001.

[9] Wikipedia, URL: http://en.wikipedia.org/wiki/E-mail_address_harvesting visited on Apr, 30th, 2009.

[10] Freshmeat website: URL:http://freshmeat.net/ visited on Mar, 10th, 2009.