ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 2, February 2015

Unmasking Spam in Email Messages

Anjali Sharma[1], Manisha[2], Dr. Manisha[3], Dr. Rekha Jain[4]

Abstract: Today e-mails have become one of the most popular and economical forms of communication for Internet users. Thus due to its popularity, the e-mail is going to be misused. One such misuse is the posting of unwelcome, unwanted e-mails known as spam or junk e-mails. E-mail spam has various consequences. It reduces productivity, takes extra space in mail boxes, extra time, extend software damaging viruses, and materials that contains potentially harmful information for Internet users, destroy stability of mail servers, and as a result users spend lots of time for sorting incoming mail and deleting unwanted correspondence. So there is a need of spam detection so that its consequences can be reduced. In this paper, the focus is on various steps of spam detection such as pre-processing, representation of data, and classification for unmasking spam in e-mail messages. Further, some results, after applying the various spam detection steps have been displayed.

Keywords: Spam, Spam detection algorithm, Spam detection techniques, Email classification

I. INTRODUCTION

Spam can be defined as unsolicited (unwanted, junk) email for a recipient or any email that the user do not wanted to have in his inbox. Spam refers to unsolicited commercial email. Also known as junk mail, spam floods Internet users electronic mailboxes. These junk mails can contain various types of messages such as pornography, commercial advertising, doubtful product, viruses or quasi legal services [1].

This paper is structured in the following sequence. A concise literature survey of the email-spam is presented in the previous paragraph. In section II, why we need spam detection is discussed. The aim of research work is presented in section III. Methodology of spam detection algorithm is dealt with in section IV. This section describes pre-processing steps, Representation of Data, and Classification in detail. The results & experiments are shown in section V and followed by the conclusion in section VI.

II. WHY WE NEED SPAM DETECTION?

Spam detection is becoming a big challenge for network resources and users because of some negative effects. Spam causes annoyance and wastes users time to regularly check and delete this large number of unwanted messages [2]. Flooding of mailboxes with spam e-mails waste storage space and overload the server; thus it may lead to losing legitimate e-mails, delaying the server response, or even make it totally unavailable. Hence, spam consumes network bandwidth and server storage space.

Spam has ethical issues like advertising fraudulent advertisements (such as make-money fast), immoral and offensive content (such as adult material and pornographic images) that are harmful to the young generations [3].

Sometimes spam even containing explicit content or malicious code including viruses, rootkits, worms, trojans or other kind of damaging software *etc.*

Spam has been the key to perform “phishing” attacks, in which a bank or another organization is supplanted in order to get valid user credentials, and steal his banking data leading to scam [4].

As a final observation, spam is not only hazardous or a waste of time, but it can be quite troubling. Also, network and email administrators have to employ substantial time and effort in deploying systems to fight spam. There is not a way to measure this damage in terms of money, but no doubt it is far from minor .Hence, it has become an important and indispensable aspect of any recent e-mail system to incorporate a spam filtering subsystem that detect spam .

III. AIM OF THE RESEARCH WORK

There The aim of research presented in this paper is to develop a spam detection algorithm that efficiently classifies a document into spam or non spam and to analyze how accurately they are classified into their original categories. A classifier is used to classify a document to be either spam or non spam.

The methodology used in the research work considered in this paper for the spam detection is summarized under the following 3 steps shown in the Fig. 1

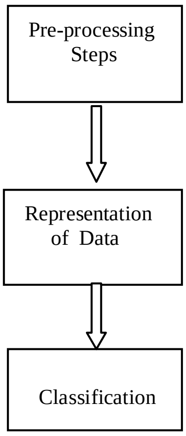

Fig 1: Main Steps in the Spam Detection Algorithm

IV. METHODOLOGY OF SPAM DETECTION ALGORITHM

The basic steps of spam detection are classified as Data Cleaning, Representation of Data, and Classification.

1) Pre-processing Steps:

In filtering of spam, the data cleaning of the textual information is very critical and important. Main objective

DOI 10.17148/IJARCCE.2015.4208

of text data cleaning is to remove data which do not give useful information regarding the class of the document. Furthermore we also want to remove data that is redundant. Most widely used data cleaning steps in the textual retrieval tasks are removing of stop words and performing stemming to reduce the vocabulary [5]. In addition to these two steps we also removed the words that have length lesser than or equal to two.

The basic pre-processing steps of spam detection algorithm are given as:

A. The word s having length <=2 are removed.
B. All the special characters are removed.
C. Stop words are removed.
D. Porter's Stemming Algorithm is applied to bring the word in their most basic form.

A) Removal of Words Lesser in Length(length<=2): Investigation of English vocabulary shows that almost all such words whose length are lesser than or equal to two contains no useful information regarding class of the document. Examples includes a, is, an, of, to, as, on etc. though there are words which have length of three and are useless like the, for, was, etc but removing all such words will cost us losing some words that are very useful in our domain, like sex, see, sir, fre (often fre is used instead of free to deceive the automatic learning filter). All email of the data set were passed through a filter which removed the words that have length lesser than or equal to two. This removed bundle of words from the corpus that were useless and reduced the size of the corpus to great extend.

For example,
Input string x= " I have a list of people you missed ! "
Output string x= "have list people you missed ! "

B) Removal of Alpha Numeric Words: There were many words found in the corpus that were alpha numeric. Removal of those terms was important as they do not keep on repeating in the corpus and they are just added in the emails to deceive the filter so that our classifier fails to find patterns in the given email.

They do not keep on repeating in the email instances. In this sense they can be considered as unique terms. They are present in large numbers in the corpus and adding them to our features set will have drastic increase in the features set size with little of information.

Counting the number of alpha numeric words in subject line or in the entire email might be helpful as spam's are reported to contain large number of alpha numeric words. So a single feature containing the number of alpha numeric words in an email might be helpful.

For example,
Input string x= " have list people you missed ! "
Output string x= have list people you missed

C) Removal of Stop Words: In information textual retrieval there are words that do not carry any useful information and hence are ignored during spam detection. In general and for document classification tasks we consider them as words intended to provide structure of the language rather than the content and mostly include pronouns, prepositions and conjunctions.

For example,
Input string x= have list people you missed
Output string x= list people missed

Stop Word List for Experiment Set:

"then", "there", "that", "which", "the", "those", "now", "when", "which", "was", "were", "been", "had", "have", "has", "will", "subject", "here", "they", "them", "may", "can", "for", "such", "and", "are", "but", "not", "with", "you", "your", "alone", "anyways", "along", "anywhere", "able", "already", "apart", "about", "also", "appear", "above", "although", "appreciate", "according", "always", "appropriate", "between", "be", "beyond", "became", "both", "because", "brief", "become", "but", "becomes", "by", "becoming", "before"

D) Stemming: The main pre-processing tasks applied in textual information retrieval tasks is the stemming. Stemming is a process of reducing words to its basic form by stripping the plural from nouns (e.g. "books" to "book"), the suffixes from verbs (e.g. "measuring" to "measure") or other affixes [6]. For example, supplies, supplying & supplied matches supply. Originally proposed by Porter on 1980, it defines stemming as a process for removing the commoner morphological and in-flexional endings from words in English. In the context of document classification we can define it to be a process of representing words and its variants with its root. We used the porter stemming algorithms. There are some examples of the words after being stemmed with porter's algorithm.

Few Examples of Words with their Stems

abilities	→	abiliti
caress	→	caress
rats	→	rat
need	→	ne
agreed	→	agre
gathered	→	gather
motoring	→	motor
sing	→	sing
conflated	→	conflat
doubling	→	doubl
hopping	→	hop
protected	→	protect
falling	→	fall

For example,
Input string x= list people missed
Output string x= list peopl miss

2) Representation of Data:

The next main task of spam detection algorithm was the representation of data. The data representation step is needed because it's very hard to do computations with the textual data. The representation should be such that it should reveal the actual statistics of the textual data. Data representation should be in a manner so that the actual statistics of the textual data is converted to proper numbers. Furthermore it should facilitate the classification tasks and should be simple enough to implement.

There exist many term weighting methods which will calculate the weight for term differently:

A) Boolean Weighting: It is the simplest of the term weighting methods where all the data is represented using Boolean values. Mathematically it can be represented as [7]

$$\textbf{\textit{Boolean \_ W}}_{ij} = \begin{cases} \mathbf{1} & \textbf{if } \; \textbf{\textit{tf}}_{ij} > \mathbf{0} \\ \mathbf{0} & \textbf{otherwise} \end{cases} \qquad \text{……..}\textit{Eq(1)}$$

A term will get a weight of 0 in email *j* if it is not present otherwise it will get a weight of 1. Boolean weighting makes the computation easy but does not consider the actual statistics of terms in the emails that's why it does not achieve as high accuracy as some of the other weighting methods does.

B) Term Frequency: Term frequency counts the number of occurrences of term in a text document.
Mathematically it can be represented as:

$$\textbf{\textit{Term \_ Frequency \_ W}}_{ij} = \textbf{\textit{tf}}_{ij} \qquad \text{…….}\textit{Eq(2)}$$

where, tf_{ij} as the frequency of term i in document j

C) Term Document Frequency inverse document frequency(TF-IDF):
Tf-Idf weighting represent that those terms whose presence is in lesser number of text documents(e-mails) can discriminate well between the classes [8].
In Tf-Idf, we have found normalized term frequency, inverse document frequency and Tf-Idf of each word in document.

$$\textbf{\textit{TFIDF \_W}}_{ij} = \textbf{\textit{tf}}_{ij} \times \textbf{\textit{idf}}$$

$$\textbf{\textit{TFIDF \_W}}_{ij} = \textbf{\textit{tf}}_{ij} \times \textbf{\textit{log}} \; (\textbf{\textit{N/ n}}_{i}) \qquad \text{…..}\textit{Eq(3)}$$

where, $\mathbf{tf_{ij}}$ is normalized term frequency

$tf_{ij=}$ Number of times term t appears in a document) / (Total number of terms in the document)

N is the total number of documents or emails in the corpus $\mathbf{n}_i$ is the number of documents in the corpus where term i appears.

3) Classification:

In Simple terms classification is a task of learning data patterns that are present in the data from the previous known instances and associating those data patterns with the classes. Later on when given an unknown instance it will search for data patterns and thus will predict the class based on the absence or presence of data patterns.
There are many classification techniques that can classify message as a spam or non spam .
Some of the classification techniques that can used in spam detection algorithm are discussed below:

A) Naive Bayes classifier: The Naive Bayes classifier is a simple machine learning technique used to classify spam e-mails. It is a probabilistic classifier; words probabilities play the main rule here. It calculates and uses the probability of certain words occurring in the known examples (messages) to categorize new examples (messages). Every word has certain probability of occurring in spam or ham email in its database. If the total of words probabilities exceeds a certain limit, the filter will mark the e-mail to either category [9]. Here, only two categories are necessary: spam or ham. Instead of using explicit rules (as in heuristic filtering or rule-based learning) or storing all instances and delaying generalization, the Bayesian approach creates a database that contains all terms (or message attributes) and their associated conditional probabilities during the training phase.

B) Support Vector Machines: The Support Vector Machines (SVM) has successes at using as classifying text documents. SVM has encouraged important researches into applying them to spam filtering. SVM are used to embed the data indicating the text documents into a vector space where geometry and linear algebra can be performed [10]. SVM try to create a linear separation between the two classes in the vector space.

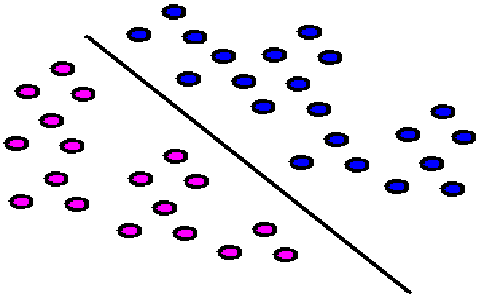

Fig 2: Support Vector Machine

An example is shown above. In this example, the objects belong either to BLUE (ham) class or PINK (spam) class. A boundary is defined using the separating line. On the left side of boundary line where all objects are PINK and to the right side of boundary line where all objects are BLUE are represented as in the Fig2. Any new object falling to the left is labelled, i.e., classified, as PINK (or classified as BLUE should it fall to the right of the separating line).

C) K-Nearest Neighbour Classifier: The k-nearest neighbour (*K*-NN) classifier is considered an example-based classifier, that means that the training documents are used for comparison rather than an explicit category representation, such as the category profiles used by other classifiers[12]. When a new document needs to be categorized, the k most similar documents (neighbours) are found and if a large enough proportion of them have been assigned to a certain category, the new document is also assigned to this category, otherwise not . To decide whether a message is spam or ham, we look at the class of the messages that are closest to it. The comparison between the vectors is a real time process.

D) Artificial Neural Network: An artificial neural network is a group of interconnected nodes these nodes are called as neurons. The well known example of artificial

DOI 10.17148/IJARCCE.2015.4208

neural network is the human brain. The term artificial neural network has moved around a huge class of models and machine learning methods. The central idea is to extract linear combinations of the inputs and derived features from input and then model the target as a nonlinear function of these features [10].

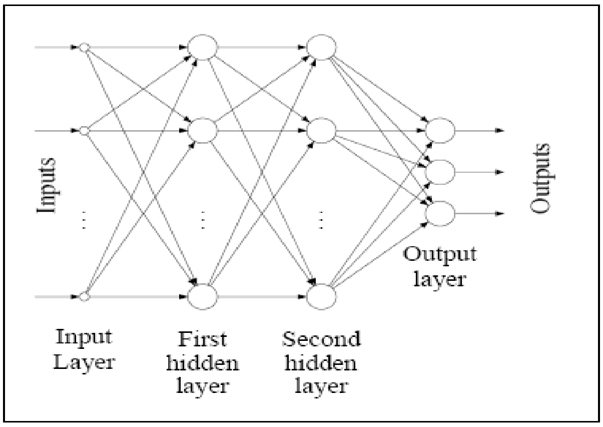

Fig 3: an artificial neural network

ANN is an adaptational system that changes its structure based on internal or external information that flows through the network during the learning phase. They are generally familiar with model complex relationships between inputs and outputs or to find patterns in data. The neural network must first be “trained” to categorize emails into spam or non spam starting from the particular data sets. This training includes a computational analysis of message content using huge representative samples of both spam and non-spam messages [11]. To generate training sets of spam and non-spam emails, each email is attentively reviewed according to the definition of spam.

V. EXPERIMENTS AND RESULTS

In this section, the data sets of selected emails that used to conduct experiments are presented. Next, a pre-processing of our data by the system are given. Finally, a set of experiments are presented followed by the results and their discussion.

1) The Data Sets: Data set of five different spam e-mails is used to conduct experiments. Table 1 shows data sets of five spam emails.

Table 1: The Data Set of Spam emails (corpora)

Email 1	Win a £1000 cash prize or a prize worth £5000
Email 2	Adult 18 Content Your video will be with you shortly
Email 3	100 dating service call ; 09064012103 box334sk38ch
Email 4	You have 1 new voicemail. Please call 08719181503
Email 5	88800 and 89034 are premium phone services call 08718711108

2) Workflow of data cleaning of Emails:

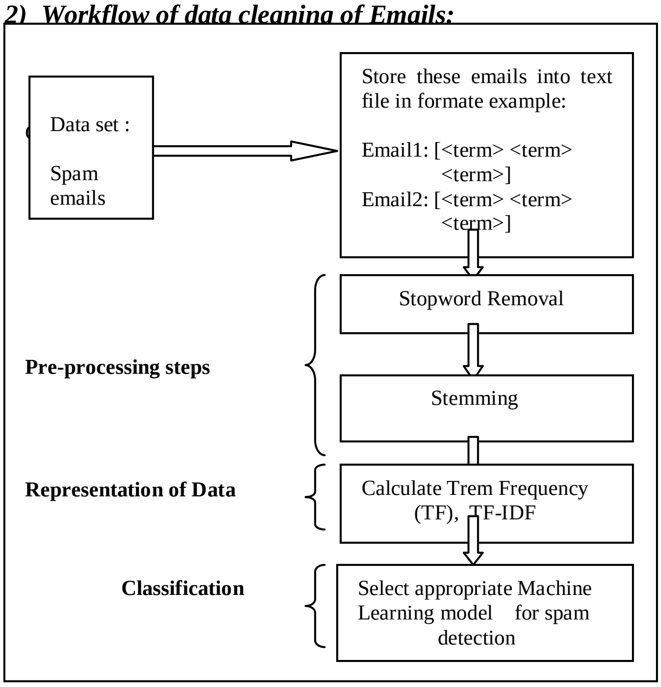
Data set :
Spam emails
Store these emails into text file in formate example:
Email1: [<term> <term> <term>]
Email2: [<term> <term> <term>]
Stopword Removal
Pre-processing steps
Stemming
Representation of Data
Calculate Trem Frequency (TF), TF-IDF
Classification
Select appropriate Machine Learning model for spam detection

Fig 4: Workflow of Spam Detection Algorithm

3) Results:

Email 1: Win a £1000 cash prize or a prize worth £5000

Email Terms	cash	prize	win	worth	£1000	£5000
Word Frequency	1	2	1	1	1	1
Normalized Term Frequency	0.166	0.333	0.166	0.166	0.166	0.166
TF_IDF	0.116	0.232	0.116	0.116	0.116	0.116

Email 2: Adult 18 Content Your video will be with you shortly

Email Terms	adult	content	shortly	video
Word Frequency	1	1	1	1
Normalized Term Frequency	0.25	0.25	0.25	0.25
TF_IDF	0.175	0.175	0.175	0.175

VI. CONCLUSION

In this paper, various spam detection steps such as pre-processing step or data cleaning step (like removing of stop words, stemming), representation of data, and then classification for unmasking spam in e-mail messages are discussed. These steps play a crucial role in classifying emails as spam or non spam emails given. Further some results of pre-processing task are also displayed.

DOI 10.17148/IJARCCE.2015.4208

REFERENCES

[1] Masurah Mohamad, Khairulliza Ahmad Salleh, "Independent Feature Selection as Spam-Filtering Technique: An Evaluation of Neural Network", Malaysia.

[2] El-Sayed M. El-Alfy, "Learning Methods For Spam Filtering", College of Computer Sciences and Engineering King Fahd University of Petroleum and Minerals, Saudi.

[3] Upasna Attri & Harpreet Kaur, "Comparative Study of Gaussian and Nearest Mean Classifiers for Filtering Spam E-mails", Global Journal of Computer Science and Technology Network, Web & Security, USA, Volume 12 Issue 11 Version June 2012.

[4] Enrique Puertas Sanz, José María Gómez Hidalgo,José Carlos Cortizo Pérez, "Email Spam Filtering", Universidad Europea de Madrid Villaviciosa de Odón, 28670 Madrid, SPAIN.

[5] Nouman Azam, "Comparative Study of Features Space Reduction Techniques for Spam Detection", Department of Computer Engineering College of Electrical and Mechanical Engineering National University of Sciences and Technology.

[6] Thamarai Subramaniam, Hamid A. Jalab and Alaa Y. Taqa , "Overview of textual anti-spam filtering techniques", International Journal of the Physical Sciences Vol. 5(12), pp. 1869-1882, 4 October, 2010.

[7] Nick Evangelopoulos ,"Introduction to the vector space model", Text Mining Notes, University of North Texas, January 2007.

[8] M. Basavaraju, Dr. R. Prabhakar, "A Novel Method of Spam Mail Detection using Text BasedClustering Approach ", International Journal of Computer Applications (0975 – 8887) Volume 5– No.4, August 2010.

[9] Ann Nosseir , Khaled Nagati and Islam Taj-Eddin, "Intelligent Word-Based Spam Filter Detection Using Multi-Neural Networks" IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 2, No 1, March 2013.

[10] Alia Taha Sabri, Adel Hamdan Mohammads, Bassam Al-Shargabi, Maher Abu Hamdeh, "Developing New Continuous Learning Approach for SpamDetection using Artificial Neural Network (CLA_ANN)", European Journal of Scientific Research, ISSN 1450-216X Vol.42 No.3 (2010), pp.511-521.

[11] Ann Nosseir , Khaled Nagati and Islam Taj-Eddin, **"**Intelligent Word-Based Spam Filter Detection Using Multi-Neural Networks" , IJCSI International Journal of Computer Science Issues, Egypt**,** Vol. 10, Issue 2, No 1, March 2013.

[12] A. K. Uysal, S. Gunal, S. Ergin, E. Sora Gunal, "The Impact of Feature Extraction and Selection on SMS Spam Filtering", ELEKTRONIKA IR ELEKTROTECHNIKA, ISSN 1392-1215, VOL. 19, NO. 5, 2013.

DOI 10.17148/IJARCCE.2015.4208