ISSN: 2321-9653

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

# Anti Spam Robot

[1]Tarun Dugar, [2]Rijul Handa, [3]Mohit Garg, [4]Akshay Gupta

*Student, B.Tech, Department of Computer Science,*
*Bharati Vidyapeeth's College of Engineering,*
*New Delhi,India*

**Abstract** - ***The rapid growth of Internet has popularized E-mail as an effective means to communicate between people. At the same time, it has encouraged a new form of advertising known as spam or junk-email. Sophisticated spammers forge their e-mail headers so that it can bypass many filters relying on address checking. In this project, the filter relies on the actual message content to distinguish spam emails. There are three distinct processes. First, the filter is supplied with many training data, including both genuine and spam e-mails. Second, the filter removes redundant words and smoothes data by applying the Porter Stemming algorithm. Finally, the testing e-mails are passed into the filter for classification.***

**Keywords**— ***Spam,Bayesian,Ham,Porter Stemming***

## I. INTRODUCTION

Unsolicited bulk e-mail, electronic messages posted blindly to thousands of recipients, is becoming alarmingly common. Although most users find these postings – called "spam" – annoying and delete them immediately, the low cost of e-mail is a strong incitement for direct marketers advertising anything from vacations to get-rich schemes. A 1997 study (Cranor & LaMacchia, 1998) found that 10% of the incoming e-mail to a corporate network was spam. Apart from wasting time, spam costs money to users with dial-up connections, wastes bandwidth, and may expose under-aged recipients to unsuitable (e.g. pornographic) content.

Some anti-spam filters are already available. These rely mostly on manually constructed pattern matching rules that need to be tuned to each user's incoming messages, a task requiring time and expertise. Furthermore, the characteristics of spam (e.g. products advertised, frequent terms) change over time, requiring the rules to be maintained. A system that would learn automatically to separate spam from other "legitimate" messages would, therefore, present significant advantages.[2]

Only one attempt has ever been made to apply a machine learning algorithm to anti-spam filtering (Sahami et al., 1998). Sahami et al. trained a Naive Bayesian classifier (Duda & Hart, 1973; Mitchell 1997) on manually categorized legitimate and spam messages, reporting impressive precision and recall on unseen messages. It may be surprising that text categorization can be effective in anti-spam filtering: unlike other text categorization tasks, it is the act of blindly mass-mailing a message that makes it spams, not its actual content. Nevertheless, it seems that the language of spam constitutes a distinctive genre, and that spam messages are often about topics rarely mentioned in legitimate messages, making it possible to train a text classifier for anti-spam filtering.

### *A. Spam*

Spam, in computing terms, means something unwanted. It has normally been used to refer to unwanted email or Usenet messages, and it is now also being used to refer to unwanted Instant Messenger (IM) and telephone Short Message Service (SMS) messages. Spam email is unwanted, uninvited, and inevitably promotes something for sale. Often the terms junk email, Unsolicited Bulk Email (UBE), or Unsolicited Commercial Email (UCE) are used to refer to spam email. Spam generally promotes Internet – based sales, but it also occasionally promotes telephone- based or other methods of sales too.

The term "spam email" generally precludes email from known sources, regardless of however unwanted the content is. One example of this would be an endless list of jokes sent from acquaintances.

### *B. Introduction to Bayesian Network*

ISSN: 2321-9653



A Bayesian network is a directed acyclic graph, which represents a set of nodes and their dependencies. A directed edge from node X to node Y , means Y conditionally depends on X.A node N is said to be conditionally independent of node M if N does not directly connect with M. Each node X is assigned with a probability table, which species the distribution over X, given the value of X's parent.

Naive Bayesian classier is simply the Bayesian classier relaxed the dependency assumption. In particular, Naive Bayesian assumes that the presence or absence of any node in the Bayesian network does not act any other nodes. For example, considering 'wet grass' and 'cloudy' as the two nodes of the network, although they both contribute to the probability of 'raining' event, the existence of 'wet grass' event does not act the existence of 'cloudy' event and vice-versa.[6]

The biggest advantage of Naive Bayes is the computational overhead reduction, in order to estimate a probability. The quantity P(X j Y) is often impractical to calculate directly, without any independence assumptions.[7] Since each node x1, x2 … xn are conditional independent of each others, given a common class C, its probability can be calculated separately, and the combination of separate probability of each node can be combined to yield an overall probability of the big event. The general formulae for Naive Bayesian in terms of each separate node can be calculated as:

P(XjC) = P(x1jC)P(x2jC) : : : P(xnjC)

Given a classification task, the Bayesian network can be applied to predict the decision outcome. For example, given the number of working hours and the stress level, the Bayesian network can represent the probabilistic relationship between number of working hours and the stress level. Then, if given a particular working hour number, the Bayesian classier can work out the probability of the stress level.

Bayesian 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.

*C. How the Bayesian Spam Filter Works*

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 mail, then it would be reasonable to assume that this email is probably spammed.[5]

## II. EXISTING SYSTEMS

Some anti-spam filters are already available. These rely mostly on manually constructed pattern matching rules that need to be tuned to each user's incoming messages, a task requiring time and expertise. Furthermore, the characteristics of spam (e.g. products advertised, frequent terms) change over time, requiring the rules to be maintained. A system that would learn automatically to separate spam from other "legitimate" messages would, therefore, present significant advantages.

The familiar methods include Bayesian filter, Support Vector Machine (SVM), instance based classifiers, neural network classifiers etc. These methods usually don't process words at the initial stage instead only make use of simple method of word frequency. So these classifiers don't show desired results. Some of the techniques are:

*A. Discretion*

Sharing an email address only among a limited group of correspondents is one way to limit spam. This method relies on the discretion of all members of the group, as disclosing email addresses outside the group circumvent the trust relationship of the group. For this reason, forwarding messages to recipients who don't know one another should be avoided.

*B. Address Munging*

Posting anonymously, or with a fake name and address, is one way to avoid email address harvesting, but users should ensure that the fake address is not valid. Users who want to receive legitimate email regarding their posts or Web sites can alter their addresses so humans can figure out but spammers cannot.

C. *Challenge/Response Systems*

Another method which may be used by internet service providers, by specialized services or enterprises to combat spam is to require unknown senders to pass various tests before their messages are delivered. These strategies are termed challenge/response systems or C/R.

D. *Checksum Based Filtering*

Checksum-based filter exploits the fact that the messages are sent in bulk, that is that they will be identical with small variations. Checksum-based filters strip out everything that might vary between messages, reduce what remains to a checksum, and look that checksum up in a database which collects the checksums of messages that email recipients consider to be spam (some people have a button on their email client which they can click to nominate a message as being spam); if the checksum is in the database, the message is likely to be spam.

## III. PROPOSED SYSTEM

GOAL:

1. Getting a learning set of various Spam and Ham emails.

2. Parsing the individual mails to extract the words of interest.

3. Implementing the Naïve Bayesian Method

PROCESS:

Particular words have particular probabilities of occurring in spam email and in legitimate email. For instance, most email users will frequently encounter the word "Viagra" in spam email, but will seldom see it in other email. The filter doesn't know these probabilities in advance, and must first be trained so it can build them up. To train the filter, the user must manually indicate whether a new email is spam or not. For all words in each training email, the filter will adjust the probabilities that each word will appear in spam or legitimate email in its database. For instance, Bayesian spam filters will typically have learned a very high spam probability for the words "Viagra" and "refinance", but a very low spam probability for words seen only in legitimate email, such as the names of friends and family members.

After training, the word probabilities (also known as likelihood functions) are used to compute the probability that an email with a particular set of words in it belongs to either category. Each word in the email contributes to the email's spam probability, or only the most interesting words. This contribution is called the posterior probability and is computed using Bayes theorem. Then, the email's spam probability is computed over all words in the email, and if the total exceeds a certain threshold (say 95%), the filter will mark the email as a spam.[3]

The initial training can usually be refined when wrong judgments from the software are identified (false positives or false negatives). That allows the software to dynamically adapt to the ever evolving nature of spam.

ALGORITHM:

Classification is a two step task:

Training stages

1. Collection of known emails

2. Preprocessing of emails

3. Creating Hash map of words

4. Calculating probabilities

5. Sorting words in relevant order of probabilities

Classification stages

1. Prepare a set of emails for testing

2. Preprocessing of emails.

3. Generate interesting word list

4. Finding overall spam probability

5. Classifying an email

Training Stages

This is the learning step, which teaches the filter which e-mail is spam and what is genuine. Two sets of training data are fed into the system. For each e-mail, the message content is broken down into smaller words. A word is a consecutive sequence of characters. Two words are separated by one or many spaces. The training data is encoded into a probability table. This table stores the probability of every word found in the training data, categorized into Spam and Non-spam classes. Thus, for some training data, a particular word will have a different probability for the Spam class and a different probability for the Non- spam class. To generate such table, the system first counts the frequency of each individual word in each class.

ISSN: 2321-9653



Classification Stages

When a new email which needs to be classified, is presented to the filtering system, it is broken down into smaller words. The Porter Stemming algorithm is applied to these words. A question arises here: what if the original words are passed to the system without applying the Porter Stemming. The answer is the probability of the word will be extremely small, because the original form of the word is not recorded in the training data. In this case, these words do not contribute to the overall probability of the whole message.[1]

The message processing in this step is similar to the one at the importing training data in step 1. However, to improve the system running time, the small filtering process to remove the most frequent words and rare words is ignored. This does not affect the results at all, because the system returns an extremely small probability for any frequent words appearing as they are not found in the training data.

COMPUTATION:

Let's suppose the suspected message contains the word "replica". Most people who are used to receiving e-mail know that this message is likely to be spam, more precisely a proposal to sell counterfeit copies of well-known brands of watches. The spam detection software, however, does not "know" such facts; all it can do is compute probabilities.

The formula used by the software to determine that is derived from Bayes' theorem :

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

Where:

- $\Pr(S|W)$ is the probability that a message is a spam, knowing that the word "replica" is in it;

$\Pr(S)$ is the overall probability that any given message is spam;

$\Pr(W|S)$ is the probability that the word "replica" appears in spam messages;

$\Pr(H)$ is the overall probability that any given message is not spam (is "ham");

$\Pr(W|H)$ is the probability that the word "replica" appears in ham messages.

Combining Individual Properties

$$p = \frac{p_1 p_2 \cdots p_N}{p_1 p_2 \cdots p_N + (1-p_1)(1-p_2)\cdots(1-p_N)}$$

where:

$p$ is the probability that the suspect message is spam;

$p_1$ is the probability $p(S|W_1)$ that it is a spam knowing it contains a first word (for example "replica");

$p_2$ is the probability $p(S|W_2)$ that it is a spam knowing it contains a second word (for example "watches");

etc...

$p_N$ is the probability $p(S|W_N)$ that it is a spam knowing it contains an $N$th word (for example "home").

To beat Bayesian filters, it would not be enough for spammers to make their emails unique or to stop using individual eye-catching words. They'd have to make their mails indistinguishable from your ordinary mail. And this would severely constrain them. Spam is mostly sales pitches, so unless regular mail is all sales pitches, spams will inevitably have a different character. And the spammers would also, of course, have to change (and keep changing) their whole infrastructure, because otherwise the headers would look as bad to the Bayesian filters as ever, no matter what they did to the message body. Enough is not known about the infrastructure that spammers use to know how hard it would be to make the headers look innocent, but guess is that it would be even harder than making the message look innocent.

FUTURE SCOPE

IV. CONCLUSION

Instead of using the Porter Stemming algorithm, other techniques in the Information Retrieval area could be applied to reduce the size of the vector space representing an e-mail. Another replacement would be the lemmatization in the RASP system.

REFRENCES

[1] P. Pantel, D. Lin: "SpamCop: A Spam Classi_cation & Organization Program" (1998)

[2] B. Medlock: "An Adaptive, Semi-Structured Language Model Approach to Spam Filtering on a New Corpus" (2006)

[3] M. Sahami, S. Dumais, D. Heckerman, E. Horvitz: "A Bayesian Approach to Filtering Junk E-Mail" (1998)

[4](1992).http://www.irdindia.in/Journal_IJACECT/PDF/Vol1_Iss1/2.pdf

[5] http://en.wikipedia.org/wiki/Bayesian_spam_filtering

[6]http://www.gfi.com/whitepapers/why-bayesianfiltering.pdf