Recent Developments in Bayesian Approach in Filtering Junk E-mail

D. Ihalagedara & U. Ratnayake

Department of Electrical and Computer Engineering,
The Open University of Sri Lanka

E-mail: dhammikai@pdn.ac.lk

Abstract

Junk mail is one of the main problems in Internet. There are several methods for the automated construction of filters to eliminate such unwanted messages from user's mail system. This paper is mainly concerned about the Bayesian filtering method and its different types of applications in junk e-mail filtering. Bayesian technique is trained automatically to detect spam messages. Several implementations that use Bayesian techniques are available as software. Any user can apply this software in different layers of client side or server side. But spammers are now trying to defeat Bayesian filters by including random dictionary words and/or short stories in their messages. The Bayesian filter can be moderated to block the new spammer techniques. The efficiency of the Baysian filter is greater than the other e-mail filters. If any one wants to filter spam out of email, it is strongly recommended not to automatically delete messages. The same is true for your real email; instead of deleting it, move it to another folder. That way, you'll build a collection of spam and non-spam messages, which will come in handy for training filters.

Key words Bayesian, spam, email filtering

1. Introduction

The number of users connected to the Internet is increasing daily. The electronic mail (E-mail) is quickly becoming the fastest and most economical feature in the Internet users. Every e-mail user can function his/her mail account and mailboxes as he/she needs. Unfortunately some virtues that have made e-mail popular have also enticed flooding of unwanted e-mail.

With the proliferation of direct markets on the Internet and the increased availability of enormous e-mail address mailing lists the volume of junk mail has grown widely in past years. This junk mail often referred colloquially as "Spam". There is no standardized definition for spam, however in generally the word "spam" is used to refer to unwanted "garbage" e-mail massages. Spam constitutes a major problem for both e-mail users and Internet Service Providers (ISP). The spam is costly for both users and ISP.

As a result of this growing situation some automated and manual methods for filtering such a junk from legitimate e-mail are needed. Many of junk mail filtering products are available, [4] which allow users to handcraft a set of logical rules to filter junk mail. The problem with the manual system is construction of rule sets to detect junk mail. It points out the need for adaptive methods for dealing with this problem. The automated learning rules to classify e-mail are introduced in [4]. While such approaches have shown some success for general classification tasks based on the text of message, the average number of spam messages received continues to increase exponentially. Figure 01 shows recent statistics on the number of spam messages received by one e-mail user and taken from [5].

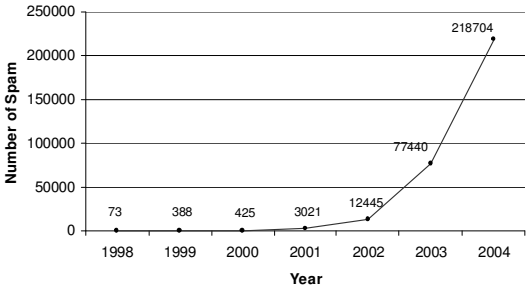

Figure 01: Annual Spam Evolutions

The spam cost to the ISP can be seen at two levels; increase of load of e-mail servers and waste of bandwidth. The slower Internet access is arising according to the bandwidth of the ISPs.

Spam filtering can be applied at the client level or server level. Several solutions and techniques for filtering spam were proposed in the literature. They are based on the header

analysis, address lists, key word lists, digital signatures and content statistical analysis (Bayesian Technique) [6, 7].

The Bayesian technique is the elaborate solution in the spam filter, which constitutes the main core of so many spam filtering software. Generally Baysian technique is used in conjunction with the other techniques in the spam filtering process. They can be applied as several layers. The Baysian filter divides manually the corps of a high number of e-mail messages into two classes; legitimate (ham) or illegitimate (garbage or spam).

2. Anti Spam Technologies

Due to the huge increase of spam in the past years the researches pay more attention for filtering spam. Many researches are presently working in the implementation of new filters that prevent spam from reaching their destination either by blocking it at the sever level or the client level. In January 2003 and 2004, a conference on spam took place at MIT in Cambridge and the Coalition Against Unsolicited Commercial E-mail (CAUCE) [8] was established. While CAUCE is trying to introduce legislations that would make spamming illegal [cauce], some research groups and companies are trying to fight/block spam.

2.1. Centralized filtering server

This architecture used a single anti-spam filter that runs on centralized organization-wide mail server.

2.2. Gateway filtering

All inbound e-mail is routed through a filtering gateway before being delivered to the mail server. Gateway services work well with web based and mobile access to e-mail.

2.3. List-based filtering

This method is richer than the other methods, also it is different to the other methods. This method is operating at the server level. Today, black-listing and white-listing are ineffective, although server-based solutions adopt them as an auxiliary technique often to be integrated with challenge/response. Black-listing resources have become less effective since spammers learned to change their source address to get around the recipient's defenses.

2.4. Rule based filtering

Rule based filters assign a spam score to each e-mail based on whether the e-mail contains features typical of spam massages, such as keywords and HTML formatting like fancy fonts and background colors.

2.5. Heuristic filtering

This method uses baseline artificial intelligence to deliver an automated spam detection process [10]. These automated mechanisms categorize incoming e-mail messages as spam or legitimate based on known spam patterns. Main advantage is that no human actions are needed for filtering here.

2.6. Receipt-time filtering

Once an SMTP server accepts an incoming SMTP connection, it can use a wide variety of techniques to detect and reject spam. An effective heuristic test is to see if the incoming connection has valid reverse DNS (rDNS), giving the sending host's domain name as well as IP address. While there's no technical requirement that all sending hosts have rDNS, many people have noted that most hosts without rDNS are only spam. In mid-2003 AOL started rejecting all mail from hosts without rDNS, which impel the few legitimate senders without working rDNS to get theirs in order.

2.7. Content filtering

Once the SMTP server has decided to accept a message, the sender transfers the entire set of message headers and the message body. (For SMTP purposes, the message headers are just part of the message, and do not affect message delivery.) Many filtering schemes work on the header and body.

2.8. Hybrid filtering

While all of the filtering techniques above can be somewhat effective, a combination of many of them usually works better than any individual one.

Many spam filters can be applied as a series. Typically a mail server will use DNS based blacklists (DNSBLs) to reject some mail, then use body filters on the mail that makes it pass the DNSBLs.
Some other available filtering methods are: Collaborative Spam filtering, Statistical methods, Content-based filtering, Checksum-based filtering, Sender-supported whitelists and tags [22].

3. The Bayesian Method

Bayesian filtering is a statistical approach, which was used by many researchers to build a spam filter. Paul Graham made a significant contribution to this domain in implementing and testing one of the first Bayes spam. [7] Later on, Gary Robinson added some improvement to this filter. He produced a number of alternative approaches to combining and scoring word probabilities. The architecture of the spam Bayesian system has three different parts: Tokenizing, combining and scoring, and testing.

3.1. Mathematical explanation

Bayesian e-mail filters take advantage of Bayesian theorem. Bayesian theorem, in the context of spam, says that the probability that an e-mail is spam, given that it has certain words in it, is equal to the probability of finding those certain words in spam e-mail, times the probability that any e-mail is spam, divided by the probability of finding those words in any e-mail.

Interface of the Baysian uses numerical estimate of the degree of belief in a hypothesis before evidence has been observed. Baysian interface usually relies on degree of belief or subjective probabilities. Baysian theorem adjusts probabilities given new evidence in the following,

This theorem may be summarized as

$$\text{Posterior} = \frac{\text{likelihood x prior}}{\text{normalizing constant}}$$

In words; the posterior probability is proportional to the prior probability times the likelihood. In addition the ratio, Pr(E|Ho) Pr(B) is sometimes called the standardized likelihood, so the theorem may also be

$$P(H_O|E) = \frac{P(E|H_O)\,P(H_O)}{P(E)}$$

where,

- Ho represents a hypothesis, called a null hypothesis that was inferred before new evidence, E become available
- P(Ho) is called the prior probability of Ho
- P(E|Ho) is called the conditional probability of seeing the evidence E given that the hypothesis Ho is true. It is also called the likelihood function when it is expressed as a function of Ho given E
- P(E) is called the marginal probability of E
- P(Ho|E) is called the posterior probability of Ho given E

The factor P(E|H0)/P(E) represents the impact that the evidence has on the belief in the hypothesis. If it is likely that the evidence will be observed when the hypothesis under consideration is true, then this factor will be large. Multiplying the prior probability of the hypothesis by this factor would result in a large posterior probability of the hypothesis given the evidence. Under Bayesian inference, Bayesian theorem therefore measures how much new evidence should alter a belief in a hypothesis.

Multiplying the prior probability $P(H_o)$ by the factor $P(E|H_o)/P(E)$ will never yield a probability that is greater than 1. Since P(E) is at least as great as $P(E \cap H_O)$, which equals $P(E|H_o)$.$P(H_o)$, replacing P(E) with $P(E \cap H_O)$ in the factor $P(E|H_o)/P(E)$ will yield a posterior probability of 1. Therefore the posterior probability could yield a probability greater than 1 only if P(E) were less than $P(E \cap H_O)$, which is never true.

3.2. Alternative forms of Bayes' theorem

Bayes' theorem is often blown up by nothing that

$$P(E) = P(H_O, E) + P(H_O^c, E)$$
$$P(E) = P(E|H_O)P(H_O) + P(E|H_O^c)P(H_O^c)$$

where H_O^c is the complementary event of H_o (Often called "not H_o"). The theorem can be restarted as

$$P(H_O|E) = \frac{P(E|H_O)P(H_O)}{P(E|H_O)P(H_O) + P(E|H_O^c)P(H_O^c)}$$

The Bayes' theorem in terms of odds and likelihood ratio can be explained as follows:

Bayes' theorem can also be written neatly in terms of a likelihood ratio Λ and odds O as

$$O(H_O|E) = O(H_O) \cdot \Lambda(H_O|E)$$

where

$$O(H_O|E) = \frac{P(H_O|E)}{P(H_O^c|E)}$$

are the odds of H_O^c given E

$O(H_O) = \frac{P(H_O)}{P(H_O^c)}$ are the odds of H_O^c by itself and

$$\Lambda(H_O|E) = \frac{L(H_O|E)}{L(H_O^c|E)} = \frac{P(E|H_O)}{P(E|H_O^c)}$$

is the likelihood ratio.

3.3. Process

This section mainly explains the Baysian theorem used in junk mail filtering. According to the section 3.1 and 3.2 we can summarize the Bays' theorem in junk mail detection as follows

$$Pr(spam|words) = \frac{Pr(words|spam)\ Pr(spam)}{Pr(words)}$$

Using Bayesian analysis to classify spam and non-spam was suggested by Paul Graham. A Bayesian filter takes each word in a message and looks it up in a database to see how many times that word has appeared in prior spam and non-spam messages. The Bayesian formula then lets it combine those counts into an overall probability estimate to check whether the message is spam or not [7].

Particular words have particular probabilities of occurring in spam e-mail and in legitimate e-mail. For instance, most e-mail users will frequently encounter the word Viagra in spam e-mail, but will seldom see it in other e-mail. The filter does not know these probabilities in advance, and must first be trained so that it can build them up. To train the filter, the user must manually indicate whether a new e-mail is spam or not. For all words in each training e-mail, the filter will adjust the probabilities that each word will appear in spam or legitimate e-mail 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 e-mail, 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 e-mail with a particular set of words in it belongs to which category. Each word in the e-mail contributes to the e-mail's spam probability. This contribution is called the posterior probability and is computed using Bayes' theorem. Then, the e-mail's spam probability is computed over all words in the e-mail, and if the total exceeds a certain threshold (say 95%), the filter will mark the e-mail as spam. E-mail marked as spam can then be automatically moved to a "Junk" e-mail folder, or even deleted outright.

4. Spam filtering mechanism of Baysian technology

There are several algorithms that use various modifications of Bayesian technique.

1. List every word in an incoming mail message
2. Determine the odds of each word appearing in a spam/garbage message, and
3. Use those odds as input to Bayes' Formula to determine if the message is garbage or not.

The first thing needed to do is to teach the Bayesian filter the difference between garbage and non-garbage messages. We can identify the spam or garbage e-mails from the content of e-mails. Most of spam e-mails contain certain key words. Instinctively, people know that a message containing these words or phrases is garbage/spam because of their experience in dealing with junk mail.

The Bayesian filter does not have the benefit of our years of experience, so we have to teach it what spam/garbage messages look like, and how they differ from non-garbage messages. The filter needs to introduce the garbage mail from the mail list. Whenever we show a message to the filter, it finds every word in the message and stores it (along with how many times it occurred) in a database.

Separate databases are kept for garbage and non-garbage mail messages. The filter uses a looser definition of a word than humans do – a

word (more properly called a token) can also be an IP address, a host name, an HTML tag, or a price (such as "Rs100"). However a token cannot be random strings, words less than three characters long, and numbers.
The filter scans through the message, creating a list of every word it knows about (in other words, every word in the message that's also in the token databases). In this example, the words it knows about are "prescription", "when", "today", "visit", and "your". Once the filter has the list of words it knows about, for each word it calculates the probability that the word appears in spam based on the frequency data in the token databases.

This probability value assigned to each word is commonly referred to as spamicity, and ranges from 0.0 to 1.0. A spamicity value greater than 0.5 means that a message containing a particular word is likely to be spam/garbage, while a spamicity value less than 0.5 indicates that a message containing that word is likely to be ham (non garbage/spam). 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 current circumvention technique the spam/garbage mail senders (spammers) are trying is to include obviously non-spam/garbage words in their spam/garbage messages. Any spammer had two goals in sending messages,

- The first (and obvious) goal was to see if the spammer could sneak the message past the Bayesian filter by including obviously non-spam/garbage words. In this, he failed.

- The secondary goal was to try to get the filter to start recognizing the words "congresswoman" and "soybean" as words with a high spamicity. If spam/garbage mail senders (spammers) can get the filter to assign a high spamicity to an adequate number of words that commonly appear in non-spam/garbage messages, they can render the filter useless.

This circumvention method also has limited utility for another good old-fashioned marketing reason. If spam/garbage mail senders (spammers) start including a large pile of legitimate text (such as an article from the CNN website)[11] in each message, they would confuse their target audience. A spam/garbage message advertising penile growth supplements that also contains an article about the relative value of the Euro is going to confuse the audience so much they just ignore the message.

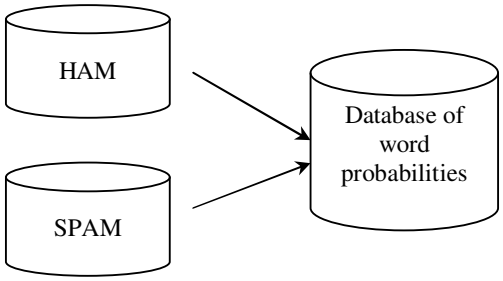

Figure 02: Creating a word database for the filter

Before mail can be filtered using this method, the user needs to generate a database with words and tokens collected from a sample of spam/garbage mail and valid mail (referred to as 'ham').

A probability value is then assigned to each word or token; the probability is based on calculations that take into account how often that word occurs in spam/garbage as opposed to legitimate mail (ham). This is done by analyzing the users' outbound mail and by analyzing known spam/garbage.

When we create the ham database the Baysian method does not require an initial learning period, it has 2 major flaws:

1. The ham data file is publicly available and can thus be hacked by professional spam/garbage mail senders (spammers) and therefore bypassed. If the ham data file is unique to your company/organization, then hacking the ham data file is useless. For example, there are hacks available to bypass the Microsoft Outlook 2003 or Exchange Server spam/garbage filter.
2. Such a ham data file is a general one, and thus not tailored to your company/organization, it cannot be as effective and you will suffer from noticeably higher false positives.

Besides ham mail, the Bayesian filter also relies on a spam/garbage data file. This spam/garbage data file must include a large sample of known spam/garbage and must be constantly updated with the latest spam/garbage by the anti-spam/garbage software. This will ensure that the Bayesian filter is aware of the latest spam/garbage tricks, resulting in a high spam/garbage detection rate

(note: this is achieved once the required initial two-week learning period is over).

When actual filtering is working, once the ham and spam/garbage databases have been created, the word probabilities can be calculated and the filter is ready for use. When a new mail arrives, it is broken down into words and the most relevant words – i.e., those that are most significant in identifying whether the mail is spam/garbage or not – are singled out. From these words, the Bayesian filter calculates the probability of the new message being spam/garbage or not. If the probability is greater than a threshold, say 0.9, then the message is classified as spam/garbage. This Bayesian approach to spam/garbage is highly effective – a May 2003 BBC article reported that spam/garbage detection rates of over 99.7% can be achieved with a very low number of false positives.

5. Applications in E-mail filtering with Baysian technology

The Bayesian technique is widely used in many technological areas. Image processing [12], Microscopic image analyzing, medical research [13], Detecting Speech Recognition Errors [14] are the mostly used areas in this technique. This review paper is trying to concern the Baysian technique in the junk mail filtering area.

5.1. SpamAssassin

SpamAssassin is a rules-based filter written in Perl. It was used for a while but spammers rapidly figured out how to get around each new rule. So it was becoming less and less effective [15]. In version 2.5 the developers added Bayesian learning to address that problem. Besides, since it is still in Perl, it is difficult to maintain and is slow.

5.2. Bogofilter

Bogofilter was one of the first Bayesian filters. Originally by über-hacker Eric S. Raymond, it's written in good old-fashioned C and runs fast [16]. If it has a weakness, it is being little too conservative about rating things as spam.

5.3. Quick Spam Filter (QSF)

QSF is a more recent Bayesian filter. It is also written in C and is even smaller than bogofilter. The scores it generates seem to skew somewhat higher than bogofilter's, to the point where it gives a lot of false positives [17].

5.4. Bayesian Mail Filter (BMF)

BMF is another option. It is very small - only 4600 lines of code, 110 KB. It is quite fast. In addition to SourceForge any person can find it in the FreeBSD ports tree [18].

5.5. iFile

ifile collects statistics on the occurrences of words in mail documents that have been filed/refiled, and uses that to determine a "best guess" of where new mail should best be filed. Some researches have done quite a lot of work to tune it to provide decent performance.

The idea is to collect a dictionary of statistics on the number of occurrences of words in messages filed to each folder. Incoming messages are compared to the dictionary, and are filed to the folder to which they have the highest degree of correspondence. When messages are refiled (due to being misclassified by the filter), the dictionary is revised [23]. Words that are not commonly used are eliminated, so that the dictionary does not get too large.

ifile uses naive Bayesian filtering as a statistical approach to direct messages to MH folders to which they have the highest degree of correspondence.

The "naive" assumption that is made is that correlations need only be done on the basis of individual word occurrences, that is, we count the number of times that the word "stop" is used, and do not consider combinations of words (e.g. "stop it" or "stop and go").

The weakness of this scheme is that while it is wonderful at "discrimination," that is, classifying dissimilar documents into different groups, it has nothing that makes it good at aggregating several kinds of dissimilar documents into a single group.

5.6. dbacl

A digramic Bayesian filter, is not restricted to just spam and non-spam. This mail filter will classify a message into one of many categories [19].

Another junk mail filters are also available like SPASTIC, SpamProbe. These all are based on the Bayesian Technique.

6. The need of Bayesian technique

There are some experiment results of Baysian filters and non-Baysian filters. Each program was installed according to its documentation. For the filters that required training, the training set data was supplied. Each filter was taken in turn and executed once for each e-mail in the spam and legitimate sets and the classification it gives was recorded.

The standard metrics for text classification are recall and precision. Spam classified as non-spam is known as a false negative. Non-spam classified as spam is known as a false positive. Precision is the percentage of messages that were classified as spam that actually are spam. High precision is essential to prevent the messages we want to read being classified as spam. A low precision indicates that there are many false negatives. Recall is the percentage of actual spam messages that were classified as spam messages. High recall is necessary in order to prevent our inbox filling with spam. A low recall indicates that there are many false positives. According to the experiment by using the several types of Bayesian mail filters [25].

According to the experiments Bayesian algorithm is compared with bag-valued features against the RIPPER rule-learning algorithm in different e-mail classification tasks. In learning a user's foldering preferences and learning to detect spam, the Bayesian filter substantially outperformed RIPPER in classification accuracy.

6.1. Advantages of Bayesian filter

The Bayesian method takes the whole message into account. It recognizes keywords that identify spam/garbage, but it also recognizes words that denote valid mail. Bayesian filtering is a much more intelligent approach because it examines all aspects of a message, as opposed to keyword checking that classifies a mail as spam/garbage on the basis of a single word.

A Bayesian filter is constantly self-adapting - By learning from new spam/garbage and new valid outbound mails, the Bayesian filter evolves and adapts to new spam/garbage techniques. For example, when spam/garbage mail senders (spammers) started using "f-r-e-e" instead of "free" they succeeded in evading keyword checking until "f-r-e-e" was also included in the keyword database. On the other hand, the Bayesian filter automatically notices such tactics; in fact if the word "f-r-e-e" is found, it is an even better spam/garbage indicator, since it is unlikely to occur in a ham mail.

The Bayesian technique is sensitive to the user - It learns the e-mail habits of the company/organization and understands that, for example, the word 'mortgage' might indicate spam/garbage if the company/organization running the filter is, say, a car dealership, whereas it would not indicate it as spam/garbage if the company/organization is a financial institution dealing with mortgages.
The Bayesian method is multi-lingual and international - A Bayesian anti-spam/garbage filter, being adaptive, can be used for any language required. Most keyword lists are available in English only and are therefore quite useless in non English-speaking regions.

A Bayesian filter is difficult to fool, as opposed to a keyword filter - An advanced spammer who wants to trick a Bayesian filter can either use fewer words that usually indicate spam/garbage (such as free, Viagra, etc), or more words that generally indicate valid mail (such as a valid contact name, etc).

7. Conclusion

The Bayesian filters, after training, offer better recall than the heuristic filters. Catching a higher proportion of spam is clearly good, since that is the reason people use them. With insufficient training, however, the Bayesian filters perform poorly in comparison with SpamAssassin in terms of recall. But some of Bayesian filters work very poorly (Quick Spam) compared with other Bayesian filters.

The Bayesian filter outperforms by far the keyword-based filter, even with very small training corpora.

As future work, we can plan to implement alternative anti-spam filters, based on other machine learning algorithms. Also the filter can include the foddering mechanism to check the spam if user need it. The server side mail filters are most effective, because many users can be protected from the filters. More than one filter can be activated in a server as a set of layers, and it is more effective than one filter

because if one filter fails another one can be successful.

8. References

[1]. L. Pelletier, J. Almhana, V. Choulakian, Adaptive Filtering of SPAM, Proceedings of the Second Annual Conference on Communication Networks and Services Research (CNSR'2004)

[2]. Mingjun Lan, Wanlei Zhou, Spam Filtering based on Preference Ranking, Proceedings of the 2005 The Fifth International Conference on Computer and Information Technology (CIT'05)

[3]. M Sahami, S. Dumais, D. Heckerman, and E. Horvitz, A Bayesian approach to filtering junk e-mail, AAAI'98 Workshop. Learning/or Text Categorization, Madison, WI, July 27, 1998.

[4]. W. W. Cohen, Learning rules that classify e-mail, In Proceedings of the 1996 AAAI Spring Symposium on Machine Learning in Information Access.

[5]. Spam statistics, http://bloodgate.com/spams/stats.html

[6]. Paul Graham, Better Bayesian filtering, January 2003,

[7]. http://paulgraham.com/better.html

[8]. Paul Graham, A Plan for Spam, August 2002, http://paulgraham.com/spam.html

[9]. The Coalition Against Unsolicted Commercial E-mail, www.cauce.org

[10]. X. Carreras and L. Andm, Boosting trees for anti-spam e-mail filtering, In Proceedings of RANLP-2001, 4th International Conference on Recent Advances in Natural Language Processing., 2001.

[11]. L. Pelletier, J. Almhana, and V. Choulakian, Adaptive Filtering of SPAM, Proceedings of the Second Annual Conference on Communication Networks and Services Research (CNSR'04), 2004.

[12]. CNN 2005, http://www.cnn.com

[13]. Nobuo Kumagai Masayoshi Aritsugi . On Applying an Image Processing Technique to Detecting Spam/garbages. Department of Computer Science, Faculty of Engineering, Gunma University, 1–5–1 Tenjin-cho, Kiryu 376-8515, Japan. 2005

[14]. S B Tan. Introduction to Bayesian Methods for Medical Research. Division of Clinical Trials and Epidemiological Sciences, National Cancer Centre. 2001

[15]. Lina Zhou1, Jinjuan Feng1, Andrew Sears1, Yongmei Shi2. Applying the Naïve Bayes Classifier to Assist Users in Detecting Speech Recognition Errors. 1Information Systems Department, UMBC, Baltimore, MD 21250, 2Computer Science and Electrical Engineering Department, UMBC, Baltimore, MD 21250. 2005

[16]. SpamAssassin, http://spamassassin.apache.org/

[17]. Bogofilter, http://bogofilter.sourceforge.net/

[18]. Quick Spam Filter (QSF), http://www.ivarch.com/programs/qsf/

[19]. Bayesian Mail Filter (BMF), http://sourceforge.net/projects/bmf/

[20]. dbacl, http://www.lbreyer.com/gpl.html

[21]. SPASTIC, http://spastic.sourceforge.net/index.html

[22]. SpamProbe, http://spamprobe.sourceforge.net/

[23]. http://en.wikipedia.org

[24]. Jason D. M. Rennie, ifile: An Application of Machine Learning to E-mail Filtering, Artificial Intelligence Lab, Massachusetts Institute of Technology

[25]. Spamfilters, http://freshmeat.net/articles/view/964/

[26]. I. Androutsopouios, J. Koutsias, K.V. Chandrinos, and D. Spyropoulos. An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal e-mail messages. In Proceedings of the 23rd A CM SIGIR Annual Conference, pages 160-167, 2000.

[27]. Jefferson Provost, Naïve-Bayes vs. Rule-Learning in classification of e-mail, Deprtment of Computer sciences, The University of Texas at Austin