# Revisit Bayesian approaches for Spam Detection

Chun-Chao Yeh and Soun-Jan Chiang
Department of Computer Science
National Taiwan University
Keelung, Taiwan 202
Email: {ccyeh,m93570032}@mail.ntou.edu.tw

## Abstract

*Due to fast changing of spam techniques, we argue that multiple spam detection strategies should be developed to effectively against spam. Among many others, (naive) Bayesian filter is one of those being proposed. While there are some early works being done on Bayesian approaches for spam detection, it is unknown if such a simple approach is still effective to filter spam mail today. In this paper, we re-evaluate the Bayesian approach with a public spam database. We found that Bayesian approach can achieve high spam detection rate for plain-text mail. However we found, at the same time, some practical issues to use Bayesian filters, for example multimedia contents and different message encoding schemes for non-English characters, are needed to be carefully handled.*

**Keywords:** *Spam, naive Bayesian filer, spam detection.*

## 1. Introduction

Today, Internet email service, one of the most important and successful Internet applications, is threatened by spam. According to a report in 2003 [7], more than one half of Internet emails are spam. The problem seems more serious today. Without a spam filter, an Internet user might receive over one hundred mails a day and find that most of them are spam. Spam generates unnecessary traffics to Internet backbone, and as a result, wastes network bandwidth and induces addition delay to normal packets. Moreover, receiving spam is a nuisance for Internet users. It costs Internet users more time and money to download their emails from their ISPs, and additional time to find out "real" mails from lots of spam mails they downloaded [8].

In literature, many spam filters use word terms in mail context as features for classification. From a set of well-classified mails (called training samples, including both spam and non-spam mails), two sets of words are selected to present both spam and non-spam mails. Each word in both sets is associated with different degree of correlation to the classes (spam or non-spam). A binary classifier (spam or non-spam) can be constructed from the training sample using different statistical classification models such as naive Bayesian [9, 10], K-NN [11], SVM [12, 13], or boosting tree [14]. While statistics-based spam filters are popular, they are subject to be attacked. More and more anti-filtering techniques are used by spammers to elude spam-filters, especially for those based on spam/non-spam words presented in the mail context [15, 16, 17, 18].

Currently, most active researches on spam detection are those based on statistic classification schemes. Among many others, (naive) Bayesian filter is one of those being proposed. While there are some early works being done on Bayesian approaches for spam detection, it is unknown if such a simple approach is still effective to filter spam mails today. Meanwhile, most of previous works on Bayesian filter for spam detection used private collected data, and thus it is hard to know if the test data set is bias to some specific usage domains.

In this paper, we re-evaluate the Bayesian approach with a public spam database. We found that Bayesian approach can achieve high spam detection rate for plain-text mail. However we found, at the same time, some practical issues to use Bayesian filters, for example multimedia contents and different message encoding schemes for non-English characters, are needed to be carefully handled.

The paper organization is as follows. In Section 2, we describe data collection mechanisms and the data sets we used for discussions. In Section 3 we give a short introduction of naive Bayestian filter and our design. Experiment results are discussed in Section 4. Concluding remarks are given in Section 5.

## 2. Data collection and the data sets

### 2.1. Data set for spam mails

We used public spam collection as our spam data sources. All the spam mails we used in this paper are from Spam Archive[1]. Most of the mails in the Spam Archive are contributed by internet users. Anyone can contribute his/her spam to the archive. None specific checking mechanism is imposed to check if any upload spam is truly a spam or not. Nonetheless, according to our observation, most of them we collected are truly spam. All the spam mails we got from spam Archive are preprocessed before taken into our spam database. First we removed those suspected to be virus/spy mails by a commercial anti-virus package. Then, we removed those mails with improper charset or connect types. For each email we check the "CHARSET" field (character set) in the mail header [4], and we accept those emails with following character sets: ISO-8859-1, US-ASCII, ASCII, UTF-8, and windows-1252. For an email with empty CHARSET, we treat the email as with charset of US-ASCII by default, according to internet standard RFC2045. Meanwhile, we removed those emails without text parts(for example, an email with a jpeg image only). We check the "CONTENT-TYPE" field in the mail header [5]. Only the email with content-type of either text/plain or text/html are kept. For those emails with multiple-parts defined by MIME(multipurpose Internet Mail Extensions) format, we check all the parts in the email. If none of the parts with text/palin or text/html content-type, we ignore the email.

Among all of the 13,353 spam mails we got from the spam archive, we have 11,786 spam mails after filtering out some of improper mails as we mentioned above. Table 1 summarizes all the spam mails we collected from the website.

### 2.2. Data set for non-spam mails

As for the corpus of normal mails (non-spam mails), we use some public web articles instead, since no public archive for normal mails is available. In this study, we use articles from American Libray Association [2](selected articles(news) on January 2004 - May 2006, 931 articles in total), Hong Kong Government News Archive [3](selected articles(news) on April 22 2006 - May 3 2006, 197 articles in total), Business Week Web site [6](selected articles(news) on January 2004 - December 2005, 931 articles in total), and two News Groups: alt.radio.talk (10 articles) and talk.politics.guns (110 articles). The choices of the above web articles are based on availability and the content attributes, so that the articles would not be restricted to some special domains only. Most of the web articles are from Business Week (accounts for 86%). Again, some preprocessing might be needed to remove those unrelated parts in the above collected web articles, for example to remove possible advertizement and banners. Totally, we have 9,244 web articles (news) to serve as non-spam mails.

## 3. Bayesian filter

We developed a Naive Bayesian filter based on Paul Graham's proposal [10]. Given a mail $M_x$, presented by a set of $k_x$ tokens, $T_x = \{t_1, t_2, ..., t_{k_x}\}$, the probability of the mail $M_x$ classified as class $c_k$ (e.g. "spam" or "non-spam") is determined by $Prob(C = c_k \mid T = T_x)$

$$= \frac{Prob(T = T_x \mid C = c_k)Prob(C = c_k)}{Prob(T = T_x)}.$$

In general, it is hard to work out $Prob(T = T_x \mid C = c_k)$. However, under Naive Bayesian assumption, we have

$$Prob(T = T_x \mid C = c_k) = \prod_i Prob(T_i = t_i \mid C = c_k),$$

in which the value of $Prob(T_i = t_i \mid C = c_k)$ is the probability of the token $t_i$ shown in a mail classified as $c_k$. Similarly we assume $Prob(T = T_x) = \prod_i Prob(T_i = t_i)$. Consequently, under Naive Bayesian assumption, we have $Prob(C = c_k \mid T = T_x)$

$$= \frac{\prod_i Prob(T_i = t_i \mid C = c_k)Prob(C = c_k)}{\prod_i Prob(T_i = t_i)}$$

$$= Prob(C = c_k) \prod_i (\frac{Prob(T_i = t_i \mid C = c_k)}{Prob(T_i = t_i)}).$$

### 3.1. Token probability database

Assume we do a random sampling over a set of mails $M$, in which there are $ngood$ of non-spam mails and $nbad$ of spam mails. For each possible token $t$ in the mails, assume $hb(t)$ spam mails have the token $t$ and $hg(t)$ non-spam mails include the token. For a mail $M_x$ randomly selected from $M$, a simple estimation of the value Prob($M_x$ includes $t \mid M_x$ is spam) can be obtained by $\frac{hb(t)}{nbad}$. Similarly, we have Prob($M_x$ includes $t \mid M_x$ is non-spam)=$\frac{hg(t)}{ngood}$. Meanwhile, by definition we have Prob($M_x$ includes $t$)=$\frac{hb(t)+hg(t)}{nbad+ngood}$.

The following are based on the procedure proposed in [10], in which we compute the value $f(t)$ as the tendency of a token $t$ likely to show up in a spam mail but less likely in a non-spam mail.

- Task: Create token probability database

| spam source | location(URL) | number of mails | number of mails selected |
|---|---|---|---|
| spam631 | ftp://spamarchieve.org/pub/archives/submit/631.r2.gz | 647 | 573 |
| spam854 | ftp://spamarchieve.org/pub/archives/submit/854.r2.gz | 114 | 104 |
| spam855 | ftp://spamarchieve.org/pub/archives/submit/855.r2.gz | 1,914 | 1,753 |
| spam856 | ftp://spamarchieve.org/pub/archives/submit/856.r2.gz | 1,863 | 1,748 |
| spam857 | ftp://spamarchieve.org/pub/archives/submit/857.r2.gz | 1,734 | 1,277 |
| spam858 | ftp://spamarchieve.org/pub/archives/submit/858.r2.gz | 1,362 | 1,159 |
| spam859 | ftp://spamarchieve.org/pub/archives/submit/859.r2.gz | 1,727 | 1,550 |
| spam860 | ftp://spamarchieve.org/pub/archives/submit/860.r2.gz | 844 | 801 |
| spam861 | ftp://spamarchieve.org/pub/archives/submit/861.r2.gz | 2,018 | 1,807 |
| spam862 | ftp://spamarchieve.org/pub/archives/submit/862.r2.gz | 1,130 | 1,014 |
| Total | | 13,353 | 11,786 |

**Table 1. Spam mails collected from the Spam Archive website.**

- Input: email corpus $M$; Token parsing function $T(.)$; weighting parameters $wb, wg$.
- Output: Value of spam indicator $f(t)$ for each token $t$ in $M$.
- Steps:
  1. Prepare two email corpus: one for spam mails, the other for non-spam mails. Assume total number of spam mails and non-spam mails are $nbad$ and $ngood$ respectively.
  2. For each mail $M_x$ in the corpus, parse mail body to get a set of tokens $T_x$.
  3. For each token $t$, count the frequency of the token shown in spam corpus(denoted as $hb(t)$), and in non-spam corpus (denoted as $hg(t)$).
  4. For each token $t$, get the probability of the token shown in spam and non-spam mails respectively:
  $$\text{Prob(M includes } t \mid \text{M is spam)} = \frac{hb(t)}{nbad}.$$
  $$\text{Prob(M includes } t \mid \text{M is non-spam)} = \frac{hg(t)}{ngood}.$$
  5. For each token $t$, calculate token probability $f(t)$
  $$= \max(0.01, \min(0.99, \frac{pb(t)}{pb(t)+bg(t)})),$$
  where $pb(t) = \min(1, wb * \frac{hb(t)}{nbad})$, $pg(t) = \min(1, wg * \frac{hg(t)}{ngood})$.

### 3.2. Detection schemes

Based on the information of $f(t)$ aforementioned, we set up a naive Bayesian filter as proposed in [10]. Detail procedure is as follows. Parameter $k$ is served as number of top tokens being selected (and ignore all the other tokens); parameter $h$ is served as threshold for classification; parameter $f_{null}$ is default token probability of $f(t)$ for a token $t$ not in the database constructed from a set of mails $M$. The three parameters are system parameters. Proper setting value of the parameters can be obtained from model training process.

- Task: Detection
- Input: incoming mail $M_x$; Token probability database $F = \{f(t) | t \text{ is a token in } M\}$; system parameters $k, h, f_{null}$.
- Output: Whether or not if the mail $M_x$ is classified as spam.
- Steps:
  1. For a new incoming mail $M_x$, parse mail body to get a set of tokens $T_x$. And let array V=empty.
  2. For each token $t$ in $T_x$, set the token probability value $v_t$ to $f(t)$ if $t \in M$; otherwise set $v_t = f_{null}$. Add the record $(t, v_t)$ to V.
  3. Let array B=empty. For each token record $(t, v_t) \in V$, count $b_t = \mid (v_t - 0.5) \mid$. Add $(b_t, t)$ to B.
  4. sort B according to the $b_t$ value in descending order. Find out top $k$ tokens with highest $b_t$ values. Assume there are $\{a_1, a_2, ..., a_k\}$. Compute $a = \prod_{1\leq i\leq k} a_i, b = \prod_{1\leq i\leq k}(1-a_i)$. Set $u = a/(a+b)$
  5. if $u > h$ then output "Spam"; else output "not-spam".

## 4. Experiment results

Among the sets of spam mails (11,786) and non-spam mails (9,244), we random selected 9,000 mails from each of them, in which 6,000 mails are form model training and the rest (3,000) are for testing. That is, our training data consists of 6,000 spam mails and another 6,000 non-spam mails; the data set for testing consists of another 6,000 mails (3,000 for each spam and non-spam mails).

## 4.1. Experiment setup

We implemented a Bayesian classifier based on the algorithms depicted in Section 3. The software is implemented in Python 2.4. And, the machine for running the classifier is IBM xSeries 346 (with two Xeon 3.2GHz CPU, 1GB RAM, 20GB Ultra 320 SCSI disk drive). The operating system for the machine is FreeBSD 6.0.

The following system parameters are fixed for all the experiments discussed, except for extra explanations. The parameter $f_{null}$ (default token probability of $f(t)$ for a token $t$ not in the database constructed from a set of mails $M$) is set to a value of 0.4. The two weighting parameters, $wb$ and $wg$, are set to 2.0 and 1.0 respectively. All the values for the above three parameters are set according to [10]. Additionally, we set up following performance matrices and notations.

- SN: number of spam mails being misclassified as non-spam mails.
- NS: number of non-spam mails being misclassified as spam mails.
- SS: number of spam mails being correctly classified.
- NN: number of non-spam mails being correctly classified.

## 4.2. Model training

To find proper values for the two system parameters, $k$ (number of significant feature selected) and $h$ (the classification threshold), we used a fraction of our collected data (12,000 mails, including 6,000 spma mails and 6,000 non-spam mails), as training data. To better utilize our data, we performed 5-fold cross validation procedure in the model training. Among all the 12,000 mails, we used four-fifth (9,600 mails) to generate token database (the $f(t)$ value for all possible tokens in the 9,600 mails). And, the rest one-fifth data (2,400 mails, including each of 1,200 spam and non-spam mails) were used for validation.

First, we fixed the parameter $k$ (number of significant features) to 15, as suggested in [10]. With the 5-fold cross validation procedure working on the 12,000 training data, we got 5 different sets of results as shown in Table 2, which shows the results over different threshold $h$. Due to space limitation, only four of them are presented. For each run, 9,600 mails (4,800 spam and non-spam mails each) are used to generate the token probability database; the rest, 2,400 mails (1,200 spam and non-spam mails each), are used as incoming mails for detection. Table 3 presents both average and standard deviation of the results over the five different runs, which shows high degree of consensus among different sets of training data. It is worthy of noting that the error to misclassified a non-spam mail as spam is more critical than the error to misclassified a spam mail as non-spam. That is, the value of $NS$ should be as low as possible while the value of $SN$ is more tolerable. The results show that no non-spam mail is misclassified with threshold $h$ set to larger than or equal to $0.1$. However, more than $1.14\%$(136.4 out of 1200)of false negative (misclassified spam mails as not spam) could arise under the same condition ($h \geq 0.1$). On the other hand, with $h$ set to as low as $10^{-7}$ false negative could be reduced to $0.32\%$ (38.8 out of 1200) at the cost of false positive being risen to $0.23\%$ (2.8 out of 1200).

## 4.3. Results from test procedure

Finally, we used another set of 6,000 mails (3,000 for each of the spam and non-spam mails) to test the performance of the spam filter. The 6,000 mails is totally different from those mails we used in model training. The spam filter is set with a fixed value of 15 for the parameter $k$, according to our observation in model training. The results are presented in Figure 1, in which we show the spam indicator value (that is the value "$u$" in Step 4 of the detection algorithm depicted in Section 3) of each of the 6,000 test mails. A mail will be classified as "spam" if the spam indicator of the mail is larger than the specified threshold $h$; otherwise it is classified as "non-spam". In Figure 1, we sort the indicator values of all the spam mails in ascending order, while sorting the indicator values of all the non-spam mails in descending order. Clearly, the results show most of spam mails with spam indicator larger than $0.1$, while none of non-spam mails with a indicator value as large as $0.1$. Figure 2 highlights the results for those spam mails with 500 smallest indicator values and those non-spam mails with 500 largest indicator values. The results show that among all the 6,000 test mails no non-spam mails is with spam indicator value larger than 0.03. That is, if we choose a threshold of 0.03 for the experiment, none of non-spam mail will be misclassified (100% of non-spam recall) and the false negative rate is less than 16.6%(500 out of 3000)(or, 83.4% of non-spam precision).

## 4.4. Discussion

While statistics-based spam filters, specially those using word-terms as feature vectors, are popular, something are worthy of noting. First, the effectiveness of the statistics-based filters is quite sensitive to the quality of training sam-

|  | R1 |  |  |  | R2 |  |  |  | R3 |  |  |  | R4 |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $h$ | SS | SN | NS | NN | SS | SN | NS | NN | SS | SN | NS | NN | SS | SN | NS | NN |
| 0.9 | 1027 | 173 | 0 | 1200 | 984 | 216 | 0 | 1200 | 996 | 204 | 0 | 1200 | 1025 | 175 | 0 | 1200 |
| $10^{-1}$ | 1081 | 119 | 0 | 1200 | 1050 | 150 | 0 | 1200 | 1058 | 142 | 0 | 1200 | 1079 | 121 | 0 | 1200 |
| $10^{-2}$ | 1097 | 103 | 0 | 1200 | 1073 | 127 | 0 | 1200 | 1078 | 122 | 0 | 1200 | 1097 | 103 | 0 | 1200 |
| $10^{-3}$ | 1145 | 55 | 3 | 1197 | 1114 | 86 | 1 | 1199 | 1116 | 84 | 1 | 1199 | 1132 | 68 | 2 | 1198 |
| $10^{-4}$ | 1159 | 41 | 3 | 1197 | 1133 | 67 | 1 | 1199 | 1127 | 73 | 1 | 1199 | 1147 | 53 | 2 | 1198 |
| $10^{-5}$ | 1163 | 37 | 3 | 1197 | 1142 | 58 | 1 | 1199 | 1136 | 64 | 1 | 1199 | 1153 | 47 | 2 | 1198 |
| $10^{-6}$ | 1169 | 31 | 6 | 1194 | 1154 | 46 | 2 | 1198 | 1147 | 53 | 1 | 1199 | 1157 | 43 | 3 | 1197 |
| $10^{-7}$ | 1173 | 27 | 6 | 1194 | 1158 | 42 | 2 | 1198 | 1154 | 46 | 1 | 1199 | 1163 | 37 | 3 | 1197 |

**Table 2. Detection results of four different runs(total number of spam and non-spam mails are 1,200 for each, parameter $k$=15).**

|  | average |  |  |  | stdev |  |  |  |
|---|---|---|---|---|---|---|---|---|
| $h$ | SS | SN | NS | NN | SS | SN | NS | NN |
| 0.9 | 1005.4 | 194.6 | 0.0 | 1200.0 | 19.4 | 19.4 | 0.0 | 0.0 |
| $10^{-1}$ | 1063.6 | 136.4 | 0.0 | 1200.0 | 15.3 | 15.3 | 0.0 | 0.0 |
| $10^{-2}$ | 1083.0 | 117.0 | 0.2 | 1199.8 | 13.1 | 13.1 | 0.4 | 0.4 |
| $10^{-3}$ | 1125.6 | 74.4 | 1.6 | 1198.4 | 12.9 | 12.9 | 0.9 | 0.9 |
| $10^{-4}$ | 1140.6 | 59.4 | 1.6 | 1198.4 | 12.6 | 12.6 | 0.9 | 0.9 |
| $10^{-5}$ | 1147.2 | 52.8 | 1.6 | 1198.4 | 10.8 | 10.8 | 0.9 | 0.9 |
| $10^{-6}$ | 1155.6 | 44.4 | 2.8 | 1197.2 | 8.4 | 8.4 | 1.9 | 1.9 |
| $10^{-7}$ | 1161.2 | 38.8 | 2.8 | 1197.2 | 7.3 | 7.3 | 1.9 | 1.9 |

**Table 3. Average and standard deviation of the detection results (total number of spam and non-spam mails are 1,200 for each, parameter $k$=15).**

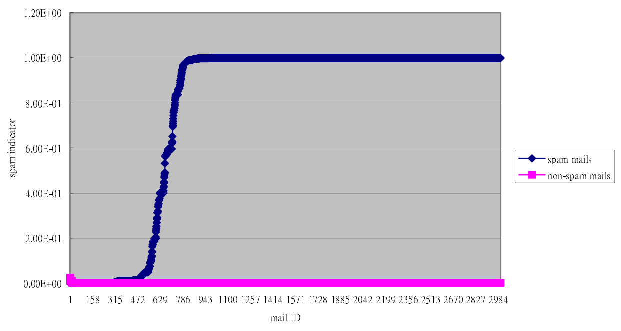

**Figure 1. Test results (total mails =6,000, including 3,000 spam and another 3,000 non-spam).**

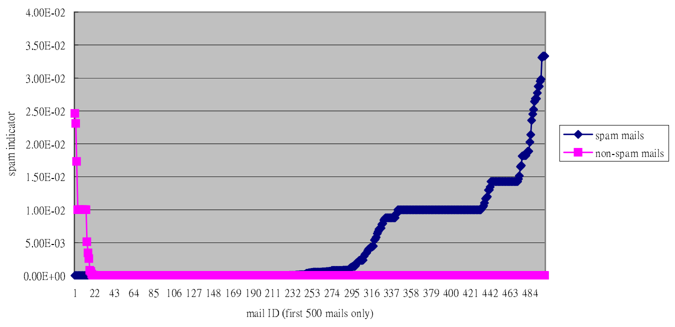

**Figure 2. Test results for the first 500 mails with highest/lowest spam indicators.**

ples and training processes. Currently, most of email systems support MIME format (Multipurpose Internet Mail Extensions). An MIME message can be represented with different combinations of content types/subtypes, character sets, and content transfer encodings. The flexibility enables same "message" in terms of semantics to be represented in many different ways, which jeopardizes the effectiveness of those spam filters using simple statistical rules and feature selection schemes.

## 5. Concluding remarks

Naive Bayesian filter is one of those being widely use spam filter schemes. While there are some early works being done on Bayesian approaches for spam detection, it is unknown if such a simple approach is still effective to filter spam mail today. In this paper, we re-evaluate the Bayesian approach with a public spam database. We collected 11,786 spam mails and 9,244 web articles from different domains as sources of non-spam mails. Based on our data, we found that Bayesian approach can achieve high spam detection rate for plain-text mail. With proper setting of threshold, results from our experiments show that the spam filter can achieve 100% of non-spam recall at the cost of the false negative rate of 16.6%(in other word, 83.4% of non-spam precision).

While many spam detection schemes have been developed, spam systems become more sophisticate to against spam filters at the same time. To successfully send their spam message to Internet users, spammers would change their spam behaviors to avoid being detected. Very likely, each spam detection technique has its weakness. It seems not possible to design a detection technique to against all possible spam tricks. At least, the argument is correct today as we have not seen such an omnipotent technique existed. Combining different strategies to form an effective detection system sounds more practical.

## Acknowledgement

This research was supported in part by National Science Council of Taiwan under the grand NSC96-2221-E-019-009, and by National Taiwan Ocean University under the grand NTOU-RD961-05-02-01-01.

## References

[1] http://www.spamarchive.org/

[2] http://www.ala.org/

[3] http://www.info.gov.hk/isd/news/index.htm/

[4] charset set. http://www.iana.org/assignments/character-sets/

[5] Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies. http://www.ietf.org/rfc/rfc2045.txt

[6] http://www.businessweek.com/

[7] Weinstein, L.: Inside risks: Spam wars. Communication of ACM, Vol. 46, No. 8 (2003) 136–136.

[8] Corbato, F.J.: On computer system challenges. Journal of ACM, vol. 50, No. 1 (2003) 30–31.

[9] Sahami, M., Dumaisy, S., Heckermany, D., Horvitzy, E.: A Bayesian approach to filtering junk E-Mail. In Proc. Of AAAI Workshop on Learning for Text Categorization, July 1998, Madison, Wisconsin, (1998) 55–62.

[10] Graham, P.: A plan for spam. Aug 2002, available at http://www.paulgraham.com/spam.html.

[11] Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Sakkis, G., Spyropoulos, C., Stamatopoulos, P.: Learning to filter spam e-mail: A comparison of a naive bayesian and a memorybased approach. In Proc. of the PKDD workshop on Machine Learning and Textual Information Access, (2000) 1–13.

[12] Tong, S., Koller, D.: Support vector machine active learning with applications to text classification. Journal of Machine Learning Research, vol. 2, (2001) 45–66.

[13] Drucker, H., Wu, D., Vapnik, V.N.: Support vector machines for spam categorization. IEEE Trans. on Neural Networks, Vol. 10, No. 5,(1999) 1048–1054.

[14] Carreras, X., Marquez, L.,: Boosting trees for anti-Spam email filtering. In Proc. of Euro Conference on Recent Advances in Natural Language Processing (RANLP 2001), Sep. 2001.

[15] Hulten, G., Penta A., Seshadrinathan G., Mishra, M.:Trends in spam products and methods. In Proc. of First Conference on Email and Anti-Spam (CEAS), 2004.

[16] Machlis, S.: Uh-oh: spam's getting more sophisticated. Computerworld, Jan 17 2003, available at http://www.computerworld.com.

[17] Graham-Cumming,J.: How to beat an adaptive spam filter. In Proc. of MIT Spam Conference, 2004.

[18] Wittel, G.L., Wu, S.F.: On attacking statistical spam filters. In Proc. of First Conference on Email and Anti-Spam (CEAS), 2004.

[19] Distributed Checksum Clearinghouse (DCC). Available at: http://www.rhyolite.com/anti-spam/dcc/.