2009 International Conference on Computer Engineering and Technology

Improved Bayesian Anti-Spam Filter – Implementation and Analysis on Independent Spam Corpuses

Biju Issac, Wendy Japutra Jap and Jofry Hadi Sutanto
School of Computing and Design
Swinburne University of Technology (Sarawak Campus)
Kuching, Malaysia
e-mail: bissac@swinburne.edu.my, {my.keyblade, jofrisutanto}@gmail.com

Abstract—**Spam emails are causing major resource wastage by unnecessarily flooding the network links. Though many anti-spam solutions have been implemented, the Bayesian spam score approach looks quite promising. A proposal for spam detection algorithm is presented and its implementation using Java is discussed, along with its performance test results on two independent spam corpuses – Ling-spam and Enron-spam. We use the Bayesian calculation for single keyword sets and multiple keywords sets, along with its keyword contexts to improve the spam detection and thus to get good accuracy.**

Keywords-spam mails; bayesian approach; spam corpus; keyword sets, context matching;

I. INTRODUCTION

Spam emails are getting better in its ability to break anti-spam filters and it would take a great deal to get it fully eradicated. Spammers are also becoming more innovative, so that the anti-spam research is having a great relevance these days. We are proposing a Bayesian approach to the anti-spam algorithms. First we implement a simple Bayesian filter based on single keyword sets. Then we improve that by using multiple keyword sets and assigning a higher weightage to them. Finally, we further refine the anti-spam filter by using context matching technique along with the previous steps. The keywords are mapped to a keyword context, which is a collection or set of other keywords except the keyword that is considered for mapping. This paper is organized as follows. Section 2 summarizes on existing or related works, section 3 details on the anti-spam approach, section 4 deals with the Java implementation of anti-spam algorithm, section 5 is describing further steps to improve spam detection and section 6 is the conclusion.

II. EXISTING AND RELATED WORKS

For email classification as spam or non-spam, naive bayes classification was used in several systems [1 - 4]. Chiu et al. presents an alliance-based approach to classify, discovery and exchange interesting information on spam mails. The spam filter is built based on the mixture of rough set theory, genetic algorithm and XCS (eXtended Classifier System) classifier system [5]. Sirisanyalak et al. uses an email feature extraction technique for spam detection based on artificial immune systems that extracts a set of four features that can be used as inputs to a spam detection model [6]. Dhinakaran et al. collected 400 thousand spam mails from a spam trap set up in a corporate mail server for a period of 14 months form January 2006 to February 2007, which is a sample of world wide spam traffic. Studying the characteristics of this sample helps to better understand the features of spam and spam vulnerable e-mail accounts. They believe that this analysis is highly useful to develop more efficient anti spam techniques. In their analysis they classified spam based on attachment and contents [7]. Zhou et al. explains on Good Word Attack that thwarts spam filters by appending to spam messages sets of "good" words, which are common in legitimate e-mail but rare in spam. They present a counterattack strategy that first attempts to differentiate spam from legitimate e-mail in the input space, by transforming each email into a bag of multiple segments, and subsequently applies multiple instance logistic regression on the bags. They treat each segment in the bag as an instance. An e-mail is classified as spam if at least one instance in the corresponding bag is spam, and as legitimate if all the instances in it are legitimate [8]. Gao et al. propose a system using a probabilistic boosting tree to determine whether an incoming image is a spam or not based on global image features, i.e. color and gradient orientation histograms. The system identifies spam without the need for OCR and is robust in the face of the kinds of variation found in current spam images [9]. Balakumar et al. uses ontology for Statistical based filtering: understanding the content of the email and Bayesian approach for making the classification [10]. Ali et al. investigates current approaches for blocking spam and proposes a new spam classification method by using adaptive boosting algorithm. Experiment was carried out to evaluate the results of spam filtering and the results were supporting adaptive boosting algorithm [11]. Lan et al. present a filtering mechanism applying the idea of preference ranking. This filtering mechanism will distinguish spam emails from other email on the Internet. The preference ranking gives the similarity values for nominated emails and spam emails specified by users, so that the ISP/end users can deal with spam emails at filtering points. They designed three filtering points to classify nominated emails into spam email, unsure email and legitimate email [12]. Ming et al. used a method of spam behaviour recognition filtering. The method identifies the spam according to the behaviour of mail sent, set up the model by Bayes technique, and in the mail filtering application to filter the spam by stages [13].

978-0-7695-3521-0/09 $25.00 © 2009 IEEE
DOI 10.1109/ICCET.2009.170

III. ANTI-SPAM APPROACH

Bayesian filtering works on the principle that the probability of an event occurring in the future can be inferred from the previous occurrences of that event [14].

Spam emails can be processed through Bayesian filters using keywords, as widely known. Single keyword or multiple keyword combinations can be used. Along with the keywords, we propose to use keyword contexts or contexts, in short. Making a spam decision by merely using keywords cannot be that accurate. Once the keyword is checked using a context, the picture becomes clearer and a more accurate decision can be taken. Context is a set of remaining keywords that is mapped to every keyword chosen as shown in figure 1. For example, if the [keyword 1] has a context of [keyword 2, keyword 3 … keyword n], then [keyword 2] has a context of [keyword 1, keyword 3 … keyword n] etc. Generally, the keywords chosen can be uncommon or critical nouns (or combinations), along with acronyms, names etc. An exemption file list can be used during implementation.

The anti-spam algorithm can be described as follows. Accept the incoming mails and extract keywords from subject line and email contents as one-keyword (k_{1i}), two-keyword (k_{2i}), three-keyword (k_{3i}) or multi keyword sets. Form contexts C_{ij} for content keywords (k_{1i}), two-keyword (k_{2i}) and three-keyword (k_{3i}) sets. The context for any keyword is a set that contains all other keywords except itself. Thus a keyword or keyword combinations can have more than one context, as different spam can contain different sets of keyword combinations. Use the identified keywords to assign a Bayesian probability related score. The keyword contexts are compared to the set of existing keywords, to find a context matching percent (CMP).

Three approaches are discussed here –Bayesian using single keywords, Improved Bayesian with multiple keywords and Improved Bayesian with keyword context matching [14]-[15].

A. *Bayesian Approach with Single keywords*

The Bayesian probability *p(k)* for keyword *k* is given as in equation 1:

$$p(k) = \frac{s(k)}{s(k) + ns(k)} \quad -(1)$$

where, *s(k)* is the number of spam emails with keyword *k* and *ns(k)* is the number of non-spam emails with keyword *k*. The overall weighted spam score is calculated as follows. The Bayesian score for single keywords and multi-keywords are calculated and no weights are assigned to multi-keywords. The keyword scores are totaled to get the spam score for a given mail.

The Bayesian probability *p(sk)* for single keyword set *sk*,

$$p(sk) = \frac{s(sk)}{s(sk) + ns(sk)} \quad -(2)$$

where, *s(sk)* is the number of spam emails with all single keyword set *sk* and *ns(sk)* is the number of non-spam emails with all single keyword set *sk*. Similar approach is adopted for multi-keywords.

B. *Improved Bayesian Approach with Multiple keywords*

In comparison to the previous method, here weights are assigned to multiple keywords. Weights associated with one, two and three keywords (or multiple keywords) are Wk_{1i}, Wk_{2i} and Wk_{3i} respectively, where i = 1 to n (where $Wk_{1i} < Wk_{2i} < Wk_{3i}$). Spam score for one, two and three keywords are given as Sk_{1i}, Sk_{2i} and Sk_{3i} respectively, where i = 1 to n. Bayesian calculation is done with weights and keywords scores are determined, which are eventually added to get the spam score.

The Bayesian probability *p(mk)* for multi-keyword set *mk,*

$$p(mk) = \frac{s(mk)}{s(mk) + ns(mk)} \quad -(3)$$

where, *s(mk)* is the number of spam emails with all multi-keyword set *mk* and *ns(mk)* is the number of non-spam emails with all multi-keyword set mk. In the simulation done, the multiple keywords present are assigned different weights in spam score calculation as follows: Two keywords are assigned a weight of MK_WEIGHT (constant value), three keywords are assigned a weight of MK_WEIGHT*3, four keywords or more are assigned a weight of MK_WEIGHT*4. Single keywords are not assigned any weights.

C. *Improved Bayesian with Keyword-Context Approach:*

To further improve the accuracy, we add the keyword context score to the improved Bayesian score. Spam score for one, two and three keywords with corresponding keyword contexts are Skc_{1i}, Skc_{2i} and Skc_{3i} respectively, where i = 1 to n. This score is calculated with respect to the matches spam mail keywords contexts find in the existing database of keywords. For example, consider a keyword [viagra] that has a context of [word1, word2, word3, word4] in a mail received. Matching percentage can be given as *x*% for keyword context match. If two words match out of four, then matching percentage would be 50%. The keyword context score (Skc_{ij}) would be a function of this matching percentage. This spam score for keyword-context pairs can have a greater contribution in the overall score. This is effected by W_1 and W_2, where W_1 is the weight (say, 70%) associated with keyword score and W_2 (say, 30%) is associated with keyword-context score component in equation 4. These values can be fine-tuned for best results. Weights associated with contexts that corresponds to one, two and three keywords are Wkc_{1i}, Wkc_{2i} and Wkc_{3i} respectively, where i = 1 to n (where $Wkc_{1i} < Wkc_{2i} < Wkc_{3i}$).

The Total Spam Score = Total weighted Bayesian score for all keywords found + Total weighted score based on matching percent for all keyword-contexts found, corresponding to all keywords. That can be mathematically expressed as in equation 4:

$$S_{total} = \sum_{i=1;j=1}^{i=n;j=n} W_1(Sk_{ij} \times Wk_{ij}) + W_2(Skc_{ij} \times Wkc_{ij}) \quad -(4)$$

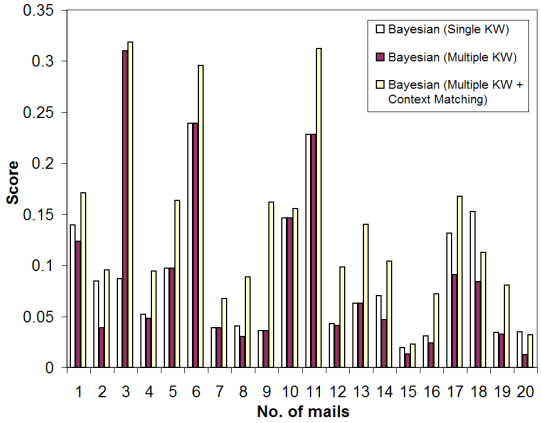

Figure 1. The graph showing the spam scores for emails during testing session on Ling-Spam corpus.

For each keyword, the corresponding contexts are formed. The presence of spam keyword itself doesn't guarantee a good spam score, but keywords with contexts if present, can give a good spam score. Threshold and weight factors should be fine tuned in different stages [15]-[16].

IV. Implementation And Analysis

The implementation program was written in Java and the software once developed was trained and tested using two public spam corpuses – Ling-spam Corpus (small size) and Enron-spam Corpus (big size) as found in [17].

Ling-spam corpus is a mixture of 481 spam messages and 2412 messages sent via the Linguist list, a moderated (hence, spam-free) list about the profession and science of linguistics. Attachments, HTML tags, and duplicate spam messages received on the same day are not included. The corpus contains 10 directories with a combination of non-spam and spam mails amounting to 2893 total mails. Enron-spam corpus contains preprocessed and raw forms of Enron-Spam datasets, amounting to 33716 total messages. The "preprocessed" directory contains the messages in preprocessed format. Attachments, HTML tags, and duplicate spam messages received on the same day are not included. The "raw" directory contains the messages in their original form. Spam messages in non-Latin encodings, ham messages sent by the owners of the mailboxes to themselves (sender in "To:", "Cc:", or "Bcc" field), and a handful of virus-infected messages have been removed, but no other modification has been made. The corpus is arranged into 6 directories that contains a combination of non-spam and spam messages.

In the Ling-spam corpus used (under bare directory), it contained contains 10 subdirectories (part1, ... part10). These correspond to the 10 partitions of the corpus that were used in the experiment. The 9 parts (part1 to part 9) were used for training and one part was used for testing (part 10). Later, all possible combinations of folders were used – nine for training and one for testing. Each one of the 10 subdirectories contains spam and legitimate messages, one message in each file. In Enron corpus, it was organized into 6 folders. Each time five folders are used for training and the remaining one was used for testing. In our implementation, we extracted only the first 100 keywords from all the mails for spam score analysis. Figure 1 shows the scores during Ling-spam testing.

TABLE I. Comparison Table for Ling-Spam Corpus

Bayesian with single keyword		Bayesian with multiple keywords		Bayesian with multiple keywords and context matching		Train and Test folders
False +ve %	False -ve %	False +ve %	False -ve %	False +ve %	False -ve %	
16.53	0	12.81	0	12.40	0	2-10 and 1
15.35	2.08	6.64	2.08	6.64	0	1, 3-10 and 2
12.86	6.25	8.71	6.25	7.05	≈ 6.25	1-2, 4-10 and 3
13.69	0	7.05	0	6.22	0	1-3, 5-10 and 4
7.02	0	1.65	0	1.24	0	1-4, 6-10 and 5
2.90	4.17	1.66	2.08	1.24	≈ 2.08	1-5, 7-10 and 6
11.62	0	6.22	0	4.56	≈ 4.17	1-6, 8-10 and 7
23.24	0	17.01	0	14.11	0	1-7, 9-10 and 8
8.71	0	5.39	0	4.56	0	1-8, 10 and 9
9.09	8.16	2.07	8.16	2.07	≈ 8.16	1-9 and 10
12.10 (avg)	**2.07** (avg)	**6.92** (avg)	**1.86** (avg)	**6.01** (avg)	**2.07** (avg)	

The average number of training and testing mails used in each of the 10 runs in Ling-spam corpus were as follows:

No. of Training Non-Spam mail = 2171

No. of Training Spam mail = 432

No. of Testing Non-Spam mail = 242

No. of Testing Spam mail = 49

The spam thresholds set were as follows: Bayesian with single keywords (0.15), Bayesian with multiple keywords

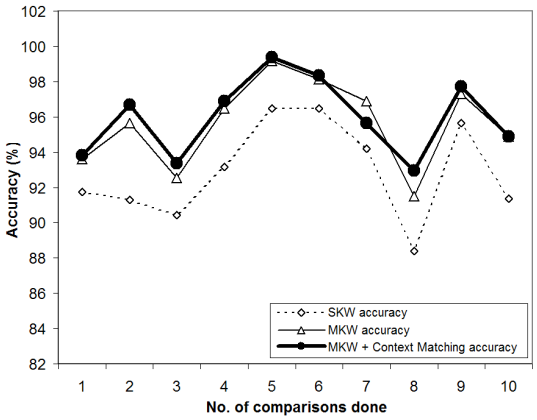

Figure 2. The graph for Ling-spam corpus showing the spam score accuracy for the three approaches (single keyword, multiple keyword, multiple keyword with context matching)

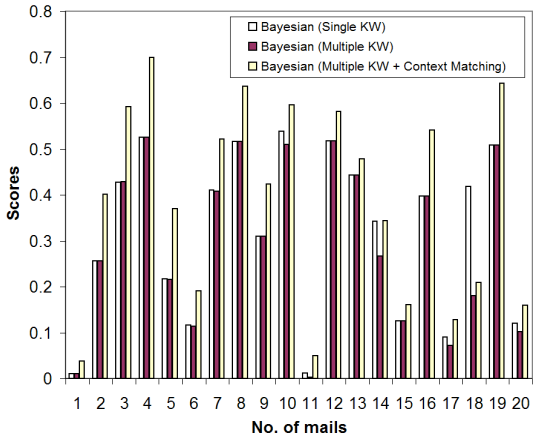

Figure 3. The graph showing the spam scores for emails during testing session on Enron spam corpus.

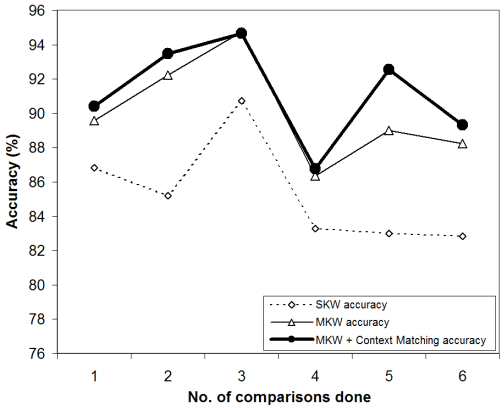

Figure 4. The graph for Enron corpus showing the spam score accuracy for the three approaches (single keyword, multiple keyword, multiple keyword with context matching)

(0.15) and Bayesian with multiple keywords and context matching (0.24). Table I shows the comparison of all possible combinations on folders in Ling-spam. Thus the average spam detection accuracy was around 96%. The accuracy graphs for all approaches are shown in figure 2.

The average number of training and testing mails used in each of the 6 runs in Enron-spam corpus were as follows:

No. of Training Non-Spam mail = 12533
No. of Training Spam mail = 15671
No. of Testing Non-Spam mail = 4012
No. of Testing Spam mail = 1500

Figure 3 shows the scores during Enron-spam corpus testing. The spam thresholds set were as follows: Bayesian with single keywords (0.57), Bayesian with multiple keywords (0.59) and Bayesian with multiple keywords and context matching (0.70). Table II shows the comparison of all possible combinations on folders on Enron.

TABLE II. COMPARISON TABLE FOR ENRON-SPAM CORPUS

Bayesian with single keyword		Bayesian with multiple keywords		Bayesian with multiple keywords and context matching		Train and Test folders
False +ve %	False -ve %	False +ve %	False -ve %	False +ve %	False -ve %	
9.01	17.33	6.43	14.40	5.39	13.80	2-6 and 1
16.37	13.23	10.41	5.15	11.42	1.60	1, 3-6 and 2
9.72	8.80	5.38	5.13	5.96	4.73	1-2, 4-6 and 3
3.33	30.07	2.27	25.04	1.87	24.62	1-3, 5-6 and 4
2.80	31.18	2.00	20.02	1.20	13.71	1-4, 6 and 5
5.93	28.4	3.67	19.87	3.13	18.22	1-5 and 6
7.86 (avg)	**21.50** (avg)	**5.03** (avg)	**14.94** (avg)	**4.83** (avg)	**12.78** (avg)	

The false positives (non-spam as spam) and false negatives (spam as non-spam) percentage is lesser for the third category, that we proposed. Thus the average spam detection accuracy was around 92%. The accuracy graphs for all three approaches are shown in figure 4.

V. STEPS TO FURTHER IMPROVE SPAM DETECTION

Some other additional steps that can improve the overall spam detection capability can be added as follows:

1. Check for any embedded hyperlinks within the email text, with the centralized hyperlink blacklist. Stamp it as spam, if the link is found in hyperlink blacklist. This single step if positive, can override other spam score calculations.
2. The user software interface can have a "Report Spam" option, to report the anti spam server software, on the status of the new incoming emails. This ensures automatic on-going training in real time. The reported spam details are used for training and fed to database, once minimum *n* users have reported it as spam.
3. Special characters (like $, -, *, digits 1-9, ', ", -#, etc.) introduced by spammers to confuse spam filters can be extracted/removed or replaced (say, 0 with o) from keywords to improve filtering.
4. Growing White List and Black Lists can be maintained as a local (or global) online repository that could be checked for existing spam signatures. Implement a white-list, which is a list of "fully permitted" email addresses. Black-listed email addresses will also be ranked based on how many people reported it as spam or phishing addresses.
5. Securing of SMTP Server is another option. SMTP servers from registered static IP address only should be allowed. It should support SMTP user authentication and be standardized to work only in this way. No SMTP relays should be allowed. SMTP servers should

not be allowed to run from a dynamic IP address, as spammers could run their own SMTP servers from dial-up connections. Optionally, digital signatures can be gradually made mandatory in emailing systems so that sender identity cannot be forged. This will prevent further email messages with spoofed sender addresses as such emails would be rejected. Only a valid sender can now send emails.

6. Implementing Grey Listing is a good option too. The Grey listing approach proposed by Harris [18] looks at three pieces of information that form a signature – the IP address of the host attempting the delivery, the envelope sender address and the envelope recipient address. If the receiving side has never seen this signature, the email would be rejected for the first time and it would become a bounced email. It would be allowed in only a second time (when the sender resends), after a delay of 25 minutes to 4 hours. Generally, this would stop spam emails to a great extent, since spammers may not resend (most of the time) their emails with the same signature.
7. Matching DNS names can improve the scenario. The web links in spam emails are also checked for veracity with the original organizations web domain, through a DNS query. If it is a concocted website link and a domain, the link can immediately be notified to the user and the central server database can be updated with the details. For example, consider a spam email with Citibank details, asking the user to click a web link to update Citibank account details. The first 2 octets in IP address of Citibank in decimal dot notation is 192.193 and this can be checked with the forged domain's IP address.
8. Email authentication can ensure that message is sent by the intended person who is the sender of the mail. The attacker normally forges the return address and would send email from a similar-looking domain to that of an original domain. There are different approaches proposed for email authentication, as of now. Return address forgery can be tackled by Sender-ID and SPF by checking DNS records to ensure whether the IP address of the sending MTA (Mail Transfer Agent) is an authorized sender. Domain level cryptographic signatures can also be used to provide authentication through Domain keys by cross-checking the DNS record. Cryptographically signed emails can be a good option especially if signing becomes a normal way of sending emails.

VI. CONCLUSION

The anti-spam implementation in Java and the subsequent analysis on two independent spam corpuses (Ling-spam and Enron-spam) proved that Bayesian approach taking into account multiple keywords and keyword contexts looks very promising. The idea is very practical and can be implemented with much promise.

REFERENCES

[1] S. Kiritchenko and S. Matwin, "Email classification with co-training," in the Centre for Advanced Studies on Collaborative Research, Toronto, Ontario, Canada, 2001, pp.1-8.

[2] K. J. Chan and J. Poon, "Co-training with a single natural feature set applied to email classification," Proc. of IEEE International Conference on Web Intelligence, 2004.

[3] K. Schneider, "A comparison of event models for naive bayes anti-spam e-mail filtering", Proc. of 11th Conference of the European Chapter of the Association for Computational Linguistics, 2003.

[4] I. Androutsopoulos, G. Paliouras, V. Karkaletsis, G. Sakkis, C. Spyropoulos, and P. Stamatopoulos, "Learning to filter spam e-mail: A comparison of a naive bayesian and a memory-based approach," Proc. of 4th PKDD's Workshop on Machine Learning and Textual Information Access, 2000.

[5] Y. Chiu , C. Chen, B. Jeng and H. Lin, "An Alliance-based Anti-Spam Approach", Proc. of Third International Conference on Natural Computation, 2007, pp.203-207.

[6] B. Sirisanyalak and O. Sornil, An artificial immunity-based spam detection system, Proc. of IEEE Congress on Evolutionary Computation, 2007, pp.3392-3398.

[7] C. Dhinakaran, J. K. Lee and D. Nagamalai, "An Empirical Study of Spam and Spam Vulnerable email Accounts", Proc. of Future generation communication and networking, 2007, pp.408-413.

[8] Y. Zhou, Z. Jorgensen and M. Inge, "Combating Good Word Attacks on Statistical Spam Filters with Multiple Instance Learning", Proc. of 19th IEEE International Conference on Tools with Artificial Intelligence, 2007, pp.298-305.

[9] Y. Gao M. Yang X. Zhao B. Pardo, Y. W. Pappas and T. N. Choudhary., "Image spam hunter", Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing, 2008, pp.1765-1768

[10] M. Balakumar and V. Vaidehi, "Ontology based classification and categorization of email", Proc. of Conference on Signal Processing, Communications and Networking, 2008, pp.199-202.

[11] S. Ali and Y. Xiang, "Spam Classification Using Adaptive Boosting Algorithm", Proc. of 6th IEEE/ACIS International Conference on Computer and Information Science, 2007, pp.972 - 976.

[12] M. Lan and W. Zhou, "Spam filtering based on preference ranking", Proc. of Fifth International Conference on Computer and Information Technology, 2005 pp.223-227.

[13] L. Ming, L. Yunchun and L. Wei, "Spam Filtering by Stages", In Proc. of International Conference on Convergence Information Technology, 2007, pp. 2209-2213.

[14] P. Graham, "Better Bayesian Filtering", 2003. Retrieved 2 May 2005, [Online]: http://www.paulgraham.com/ better.html

[15] B. Issac and V. Raman, "Implementation of Spam Detection on Regular and Image based Emails - A Case Study using Spam Corpus", Proc. of MMU International Symposium on Information and Communication Technologies, 2006, pp.431-436.

[16] I. Androutsopoulos, J. Koutsias, K.V. Chandrinos, G. Paliouras and C.D. Spyropoulos, "An Evaluation of Naive Bayesian Anti-Spam Filtering". Proc. of Workshop on Machine Learning in the New Information Age, 11th European Conference on Machine Learning, Barcelona, Spain, pp. 9-17, 2000.

[17] Internet Content Filtering Group, Spam Corpora, [Online]: http://www.iit.demokritos.gr/skel/i-config/

[18] E. Harris, The Next Step in the Spam Control War: Greylisting, 2004. [Online]:http://projects.puremagic.com/greylisting/whitepaper.html