Cooperative Anti-Spam System Based On Multilayer Agents

Wenxuan Shi
Laboratory of Intelligent Information Processing, University of Nankai
Weijin Road 94,Nankai District
Tianjin 300071, China
086-13920561100
shiwx@nankai.edu.cn

Maoqiang Xie
Laboratory of Intelligent Information Processing, University of Nankai
Weijin Road 94,Nankai District
Tianjin 300071, China
086-13702194492
xiemq@nankai.edu.cn

Yalou Huang
Laboratory of Intelligent Information Processing, University of Nankai
Weijin Road 94,Nankai District
Tianjin 300071, China
086-13821302626
huangyl@nankai.edu.cn

ABSTRACT
Spam is unsolicited bulk email which is extremely annoying to the recipients and the ISPs. However, most of the traditional spam filtering methods commonly neglect the bulk character of spam. This paper proposes a model of cooperative anti-spam system based on multilayer agents. We compared our model to the state-of-the-art and found that our model achieved better performance and robustness on several known corpora.

Categories and Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval *–Information filtering, Relevance feedback, Retrieval models.*

General Terms
Design, Experimentation

Keywords
Spam filtering; fingerprint; shingle; cooperative anti-spam system; multilayer agent

1. INTRODUCTION
Spam is unwanted communication intended to be delivered to an indiscriminate target, directly or indirectly, notwithstanding measures to prevent its delivery [1]. Spam is extremely annoying to the recipients and the ISPs, because of occupying much bandwidth, wasting many resources of storage and computation, more seriously, threatening the safety of internet and personal computers.

Spam filtering is an automated technique to identify spam for the purpose of preventing its delivery. For now there are many spam filtering methods to fight and block spam, such as rule-based filtering, whitelists/blacklists, challenge-response, keyword-based filtering, content-based filtering, etc. However, these methods commonly neglect the important character of spam that each email is typically sent to a vast number of recipients. Cooperative method, therefore, should resolve the problem more effectively by capturing, recording, and querying the judgments of recipients who have received the same or similar emails [2, 3].

Copyright is held by the International World Wide Web Conference Committee (IW3C2). Distribution of these papers is limited to classroom use, and personal use by others.
WWW 2011, March 28-April 1, 2011, Hyderabad, India.
ACM 978-1-4503-0637-9/11/03.

Fingerprints are short tags for large objects [4]. Calculated by a certain cryptographic hash function, we can universally identify every email with a fingerprint. Fingerprints have the property that if two of them are different then the corresponding emails are certainly different and there is only a small probability that two different emails have the same fingerprint. We can firstly extract some individual shingles or important features from an email, and then calculate a result set of fingerprints so as to discern duplicated or similar emails.

2. RELATED WORK
2.1 Email Fingerprint
The purpose of using email fingerprints, not original email or extracted summary of content, to discern spam is to reduce information storage size, raise comparison speed and protect recipients' privacy. Fingerprint generation approach can be seen as a certain collection of functions: $F = \{f : \Omega \to \{0,1\}^k\}$, where Ω is the whole set of all concerned emails and k is the length of the fingerprint. For a set of email samples S ($S \subset \Omega$), which includes n individual emails, we have such properties as:

$$|f(S)| = |S|, \quad (1)$$

$$f(A) \neq f(B) \Rightarrow A \neq B, \quad (2)$$

or

$$\Pr ob(f(A) = f(B) \mid A \neq B) \approx 0. \quad (3)$$

Robin presented a fingerprinting scheme based on arithmetic modulo an irreducible polynomial with coefficients in Z_2 [5]. Rabin's fingerprinting algorithm is fast and easy to implement, allows compounding, and comes with a mathematically precise analysis of the probability of collision. Mainstream cryptographic grade hash functions generally can also serve as high-quality fingerprint functions. A drawback of cryptographic hash algorithms such as MD5 [6] and SHA [7] is that they take longer to execute than Rabin's fingerprinting algorithm.

2.2 Collaborative Spam Filtering
According to the definition, spam is an issue about consent, not only content. So long as an unsolicited email is sent in bulk, the message is spam. Collaborative Spam filtering is a relatively newer and better approach to spam recognition. The technique is

to have many users share their judgments of what's an undesired email and what not.

Some spam filters based on collaborative approach are already available [8], such as DCC (Distributed Checksum Clearinghouse), Vipul's Razor, Pyzor and Cloudmark, etc. The DCC (Distributed Checksum Clearinghouse) is an anti-spam content filter based on distributed checksums including values that are constant across common variations in bulk messages, including "personalizations". Vipul's Razor is a distributed, collaborative, spam detection and filtering network which establishes a distributed and constantly updating catalogue of spam in propagation that is consulted by email clients to filter out known spam. Pyzor is a python implementation of Vipul's Razor, but using a different protocol. Cloudmark is an anti-spam and anti-phishing system whose solution is originally based on Vipul's Razor.

3. COOPERATIVE ANTI-SPAM SYSTEM BASED ON MULTILAYER AGENTS

In this paper, we design a cooperative model based on multilayer agents, called CMMA for short.

3.1 Multilayer Decisions

The performance of a spam filter is tightly interrelated with the adopted knowledge. That is, if a spam filter considers more sophisticated knowledge, it will achieve more veracious results, but consume more computation and storage resources. Conversely, if a spam filter implements more straightforward operations, it will run simpler and faster, but become resilient and flexible no more. For this reason, we propose a solution of multilayer decisions, as shown in Figure 1. As to those duplicated spam in bulk, the first layer based on nature-content fingerprinting will filter them immediately, and as to those seriously disguised or disturbed spam, the second layer based on statistics-content fingerprinting will discern them sophisticatedly.

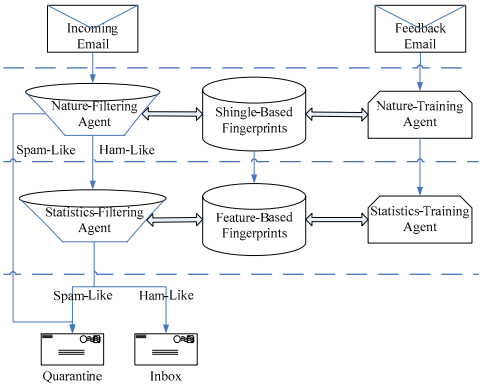

Figure 1. The solution of multilayer decision in CMMA.

3.2 Shingle Extraction

The simplest and fastest method to identify spam in bulk is to directly hash the entire content of every email to a unique fingerprint. This type of method finds exact matches by comparing the fingerprint of a coming email with the other labeled fingerprints. However, a simple hash of the entire email content is not resilient to small content changes, like an additional space added to an email, the addition or deletion of the word "the," a stem change to a term, or the replication of a sentence or paragraph.

An optimization approach is the shingling arithmetic, where one shingle means a contiguous subsequence contained in an email. Given an email E containing n words, we can regard it as a collection containing (n-w+1) shingles with the length w. For an example of 4-shingling, the string "Cooperative Anti-Spam System Based On Multilayer Agents" should be replaced by the collection of some shingles: {"Cooperative Anti-Spam System Based", "Anti-Spam System Based On", "System Based On Multilayer", "Based On Multilayer Agents"}.

Then the next step, we produce a fingerprint list for all these extracted shingles, for fast updating and querying, sorted by fingerprint value. To do this, the sketch for each email is expanded into a list of $< Fingerprint, Email\_ID >$ pairs, where the Email_ID indicates which email the fingerprint belongs to. Hence, we can calculate the similarity of two emails with the ratio of the number of fingerprints they have in common to total number of fingerprints between them.

3.3 Feature Extraction

Spammers have long been varying the individual contents of each spam to make it difficult to wholesale reject a particular message. The general notion is that if an email contains roughly many features of spam it is also spam whether or not it is a precise whole content match. In our solution, we classify email features into five categories: format features, term features, contact features, attachment features and disguised features.

1) Format Features: We often find some white noises and striking terms in spam. In the front case, there are some HTML concealments, such as font-size is less than 0, the forecolor value is equal to the backcolor value, existing some invisible span or div, etc. In the later case, spammers always attract recipients using some evident contrast, such as "COLOR=FF0000", subject line with all uppercase, etc.

2) Term Features: Decompose email content into individual terms and select the most important top k terms that could represent the email content by a certain weight based method, such as IDF (inverse document frequency), IG (information gain), MI (mutual information), etc.

3) Contact Features: Spammers often remain some dubious contact information in spam to draw in recipients, such as a website URL, a phone number, an IM address, etc.

4) Attachment Features: Extract every attachment or image as the input parameter of fingerprint function. Thus we can reduce attachments' size to that of short fingerprints, in addition, block virus dissemination and be immune to vicious content interfering.

5) Disguised Features: We often find such disguises, for example, using "c0ck" instead of "cock", anchor tags whose href attribute does not match the displayable text inside the anchor, etc.

These features extracted from an email turn out to be as good an indicator of spam as any content-based feature. Then next step, we produce a fingerprint list for all these extracted features, for fast updating and querying, sorted by fingerprint value. To do this, the sketch for each email is expanded into a list of $< Fingerprint, Indicator\_Score, Category\_ID >$ triplets, where the Category_ID indicates which category the fingerprint belongs to, and the Indicator_Score indicates how "spammy" the fingerprint is.

3.4 Fingerprint Generation

Fingerprinting an email rather than transmitting the entire email content protects the privacy of sender and recipient and dramatically reduces the cost associated with transmitting, storing, and processing feedback.

In our solution of multilayer decisions, the size of shingle extracted in the first layer (nature-content fingerprinting) is generally larger than that of feature extracted in the second layer (statistics-content fingerprinting). So we choose different hash function as fingerprint generator for each layer.

The first layer uses a 40-bits fingerprint function, based on Rabin fingerprints [5], which is fast to calculate hash value. We use the ordered shingles extracted from an email as input parameters and get $< Fingerprint, Email\_ID >$ pairs as output result. In our solution, shingle-based fingerprinting has some good properties as follows:

$$f(A+B) = f(A) + f(B), \quad (4)$$

where A and B are different shingles, $f(*)$ is the fingerprint function, and

$$f(concat(A,B)) = f(concat(f(A),B)), \quad (5)$$

where $concat(*,*)$ is the concatenation of two shingles.

The second layer uses the SHA-1 hash function [7], which designed by the NSA (National Security Agency) and published by the NIST (National Institute of Science and Technology). We use the classified features as input parameters and get $< \text{Fingerprint}, \text{Indicator\_Score}, \text{Category\_ID} >$ triplets as output result, where the Category_ID indicates which category the fingerprint belongs to, and the Indicator_Score will be assigned a value in the phase of model training.

3.5 Model Training

Even though it is an obvious idea that to accumulate a giant corpus including both spam and ham for spam filtering, we might meet with enormous difficulties of how to collect individual emails, how to mark their labels (spam or ham), and how to assure personal information safety. Considering these difficulties, it may be a good solution to have a cooperatively maintained list of encrypted fingerprints according to a pattern of decentralisation and personalisation.

In practice, in order to reduce the cost of deployment and operation, meanwhile, to improve the efficiency of model training, we employ a cooperative anti-spam model (see Figure 2) based on integrated decision-making of multi-recipient collaboration and multi-honeypot collaboration.

In model training phase of CMMA, when a training agent receives an email, spam or ham, from recipients or honeypots, the agent extracts shingles and features, firstly. Then input these shingles or features into a certain fingerprint generator to receive the output of fingerprints. Finally, use these fingerprints to update some shared databases of fingerprints.

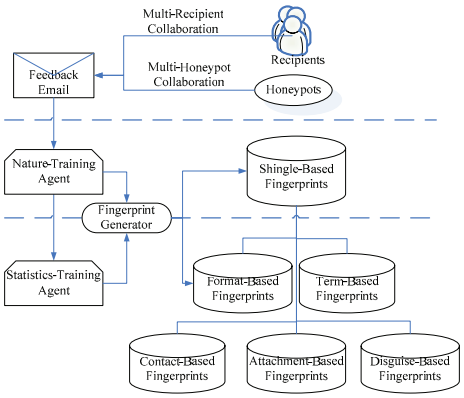

Figure 2. Model training in CMMA.

3.5.1 Training Agent

There are two kinds of training agents in CMMA, i.e. nature-training agent and statistics-training agent.

The nature-training agent is applied to deal with shingle-based fingerprints by extracting shingles from the nature of email content and maintaining the count of each feedback email. These feedback emails from recipients and honeypots may be labeled or not, therefore called semi-supervised training. Assume input $(T, S, label)$ where T is the set of all training emails, $S \subset T$ and $label : S \to \{spam, ham\}$, that is, label is defined for only a subset of the training examples. The nature-training agent will get the output result of $< Fingerprint, Email\_ID >$ pairs and $< Email\_ID, Indicator\_Score, Duplicate\_Count >$ triplets, where the Indicator_Score is calculated according to the labeled emails, and the Duplicate_Count is added up according to the count of all the feedback emails.

The statistics-training agent is applied to deal with feature-based fingerprints by extracting, selecting and weighting features according to statistical analysis. The statistics-training agent will get the output result of $< \text{Fingerprint}, \text{Indicator\_Score}, \text{Category\_ID} >$ triplets, where the Category_ID indicates which category the fingerprint belongs to, and the Indicator_Score indicates how "spammy" the fingerprint is according to a certain feature weighting algorithm.

3.5.2 Database of Fingerprints

From a technical standpoint, it is convenient to build and maintain a sharing fingerprint database with the ability of storing, updating and querying. In connection with statistics-content fingerprinting, we classify email features into five categories: format features, term features, contact features, attachment features and disguised features, therefore as to the database of fingerprints, we produce five types of databases: format-based fingerprints, term-based fingerprints, contact-based fingerprints, attachment-based fingerprints and disguise-based fingerprints (see Figure 2).

3.5.3 Multi-Recipient Collaboration

The nature of spam is that each message is typically sent to a vast number of recipients. Multi-recipient collaboration permits the

anti-spam system to capture, record, and query recipients' feedback of their judgments. One vantage is that the system can provide recipients with adequate right to decide whether their emails are unwanted messages or not in a distributed way. Another vantage of multi-recipient collaboration is that we can observe the concept change of what is spam over person and over time.

3.5.4 Multi-Honeypot Collaboration

Honeypot is a spam trap could be deployed in a distributed way to decoy spammers. The feedbacks submitted by distributed honeypots are passed to training agents, which generate fingerprints for them and update these fingerprints' Indicator_Score in fingerprint database. Multi-honeypot collaboration is very effective because the emails collected by honeypots are almost assured spam..

3.6 Spam Detection

Cooperative spam filtering, in which the decision-making result is used not only to detect spam for incoming emails, but also to provide useful statistical information to fingerprint database, promises to make spam detection easier and faster with shared knowledge. As to those duplicated spam in bulk, the nature-content fingerprinting will filter them immediately by comparing shingle-based fingerprints and then calculating the similarity. As to the seriously disguised or disturbed spam which has escaped the first layer based on nature-content fingerprinting, the statistics-content fingerprinting will discern them more sophisticatedly by matching feature-based fingerprints and then calculating the combined score (see Figure 3).

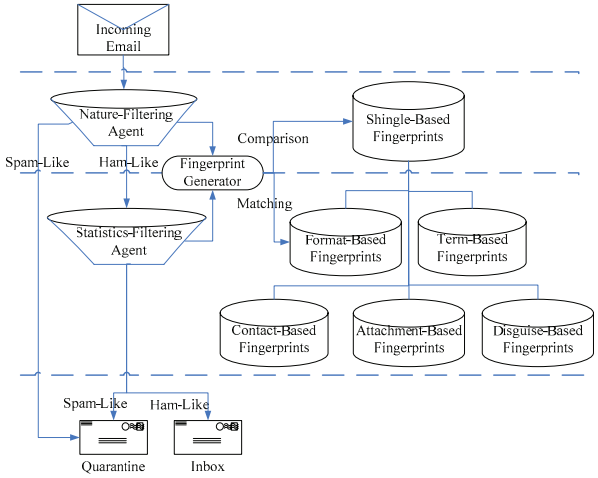

Figure 3. Spam detection in CMMA.

3.6.1 Filtering Agent

There are two kinds of filtering agents in CMMA, i.e. nature-filtering agent and statistics-filtering agent. If the queried fingerprint exists in the fingerprint database, the filtering agent can do the similarity calculation or the combined score calculation. Or else if the queried fingerprint is not in the fingerprint database, the filtering agent can register the new fingerprint with recipient's judgment in the fingerprint database for the learning of the filtering agent.

The nature-filtering agent is applied to deal with shingle-based fingerprints by comparing fingerprints and then calculating the similarity of two emails. The comparison of two fingerprints sets allows us to calculate a percentage of overlap between two emails' shingles. Assume that m_i and m_j are two emails to be compared, we calculate the similarity of them as:

$$r(m_i, m_j) = \frac{|\{f(m_i)\} \cap \{f(m_j)\}|}{|\{f(m_i)\} \cup \{f(m_j)\}|}, \quad (6)$$

where $f(m)$ is the function of fingerprint generation. Alternatively, we also can calculate the similarity as:

$$r(m_i, m_j) = \frac{2*|\{f(m_i)\} \cap \{f(m_j)\}|}{|\{f(m_i)\}| + |\{f(m_j)\}|}. \quad (7)$$

Then we compare $r(m_i, m_j)$ with a certain threshold t, and if $r(m_i, m_j) > t$, we will regard the two emails as the same. Finally, by querying the triplets of $< Email\_ID, Indicator\_Score, Duplicate\_Count >$ generated in the training phase from fingerprint database, we can make the following decision: if the Indicator_Score exceeds a certain threshold t_i or the Duplicate_Count exceeds a certain threshold t_d, the nature-filtering agent will classify the email to spam.

The statistics-filtering agent is applied to deal with feature-based fingerprints by matching identical fingerprints generated from each category. Then, by querying the triplets of $< \text{Fingerprint}, \text{Indicator\_Score}, \text{Category\_ID} >$ generated in the training phase from fingerprint database, we can calculate the combined decision-making score as follows:

$$C(m) = \sum_{i=1}^{k} Score(f_i(m)), \quad (8)$$

where $f_i(m)$ is the i th fingerprint of email m, $Score(f_i(m))$ is the Indicator_Score of $f_i(m)$. Finally, we can make the following decision: if $C(m)$ exceeds a certain threshold t, the statistics-filtering agent will classify the email to spam.

4. EXPERIMENTS

In order to examine the performance of CMMA, we implement two groups of experiments. In our experiments, we adopt the evaluation method of ROC curve to compare our results to that of SpamAssassin and Vipul's Razor. The ROC curve is the representation of the tradeoffs between FPR (false positive rate) and FNR (false negative rate), where FPR is the proportion of ham identified as spam, and FNR is the proportion of spam identified as ham. For clarity, the percentage of $(1 - AUC)\%$ often be report for evaluation, where the AUC is the area under the ROC curve.

We performed two groups of experiments to compare the performance and robustness of SpamAssassin, Vipul's Razor and CMMA. Table 1 summarizes the comparison results of two groups of experiments.

TABLE 1. Results of Two Groups of Experiments

Corpora	(1-AUC)%		
	SpamAssassin	Vipul's Razor	CMMA
Ling-Spam	8.89	4.49	3.57
PU3	5.25	3.01	2.45

We performed the first group of experiments on the Ling-Spam corpus [9], which includes 481 spam messages and 2412 ham messages. To simulate the real-world of spam in bulk, we generated multi-duplicate of spam, i.e. 2-duplicate, 4-duplicate, 8-duplicate, 16-duplicate, and 32-duplicate. Then mixed these duplicated spam into the original corpus. The comparison results are shown in Figure 4.

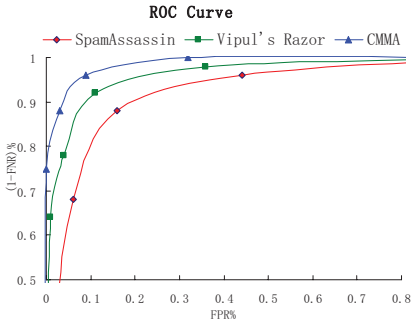

Figure 4. Results on Ling-Spam corpus.

We performed the second group of experiments on the PU3 [10] corpus, which consists of 2313 ham messages and 1826 spam messages. To simulate the real-world of spam in bulk, we generated multi-duplicate of spam as that in the first group of experiments. The comparison results are shown in Figure 5.

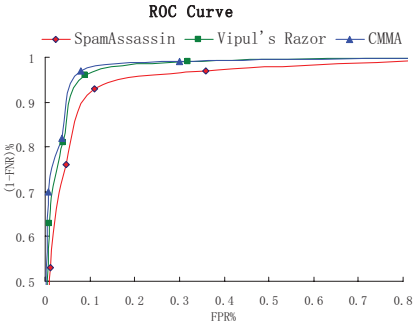

Figure 5. Results on PU3 corpus.

5. CONCLUSIONS AND FUTURE WORK

This paper proposes a model of cooperative anti-spam system based on multilayer agents using email fingerprints. Experiments indicate the model has better performance and robustness than the state-of-the-art on Ling-Spam corpus and PU3 corpus. However, the corpora in our experiment were duplicated by hand, which may be different pattern from that in practice. Another future work is to build the model into the real-world network to measure and test the performance of our solution.

6. ACKNOWLEDGMENTS

We wish to thank the open-source project: jASEN [11], depending on which we can implement our model into the plugin-based framework of spam filtering.

7. REFERENCES

[1] Gordon V. Cormack. 2007. Spam Filtering: A Systematic Review. *Foundations and Trends in Information Retrieval*. Vol. 1 no. 4, April 2007.

[2] Kang Li, Zhenyu Zhong, Lakshmish Ramaswamy. 2009. Privacy-Aware Collaborative Spam Filtering. In *IEEE Transactions on Parallel and Distributed Systems*. Vol. 20, no. 5, 2009.

[3] Jason J. Jung. 2009. Towards Collaborative Spam Filtering Based on Collective Intelligence. In *Proceedings of the First Asian Conference on Intelligent Information and Database Systems* (April 2009). ACIIDS '09. Washington, DC.

[4] A. Z. Broder. 1993. Some applications of Rabin's fingerprinting method. In *Sequences II: Methods in Communications, Security, and Computer Science*. New York, NY: Springer-Verlag, 1993, 143-152.

[5] M. O. Rabin. 1981. *Fingerprinting by random polynomials. Center for Research in Computing Technology*. Harvard University, Report TR-15-81, 1981.

[6] RFC 1321, section 3.4. *Step 4. Process Message in 16-Word Blocks*. Page 5.

[7] Secure Hash Standard, U.S. Department of Commerce/National Institute of Standards and Technology, FIPS PUB 180-1, April 1995.

[8] Stason.org. 2006. *Anti-SPAM Techniques: Collaborative Content Filtering.* http://stason.org/articles/technology/email/junk-mail/collaborative_content_filtering.html.

[9] Ion Androutsopoulos, John Koutsias, et al. 2000. An Evaluation of Naive Bayesian Anti-Spam Filtering. In *The 11th European Conference on Machine Learning*. ECML '00. Barcelona, Spain, 9-17.

[10] I. Androutsopoulos, G. Paliouras, and E. Michelakis. 2004. *Learning to filter unsolicited commercial e-mail.* Technical Report 2004/2, NCSR "Demokritos".

[11] Jasen.org. 2010. The java Anti Spam Engine. http://www.jasen.org/.