Structure Abstraction Schemes for Handling Spam E-Mail

M.Manoj Kumar[1], T.Satish[2], E.V.Sandeep[3]

[1] Student, Sasi Institute of Technology and Engineering, Tadepalligudem,W.G(dt)
[2] Asst.professor, Sasi Institute of Technology and Engineering,Tadepalligudem,W.G(dt)
[3] Asst.professor, Sasi Institute of Technology and Engineering, Tadepalligudem,W.G(dt)

Abstract: SMTP Servers equipped with machine learning algorithms such as Naive Bayes, SVM models are claimed effective in handling spam e-mails, but the results appear to be farfetched considering spam emails finding their way into our inbox once in a while. So far the spammers appear to circumvent many barriers set up by spam detection system. But the vulnerability of the spammers is their message. So content-based filters are the way to stop email spam. Recently the notion of collaborative spam filtering with near-duplicate similarity matching scheme has been widely discussed and implemented. The primary idea of the similarity matching scheme for spam detection is to maintain a known spam database to block subsequent spam mails. We propose Spam Tree Abstraction Scheme, which considers e-mail layout structure to represent e-mails along with hash-based text representation for comparisons. For an optimized comparison we use Simhash. We present a procedure to generate the e-mail abstraction using HTML content in e-mail(especially anchor and img tags), and this newly devised abstraction can more effectively capture the near-duplicate phenomenon of spams. Moreover, we design a complete spam detection system that considers other spam checking criteria's besides content-based filtrations. Spam Tree Abstraction Scheme recognizes spam email effectively based primarily on content. An implementation of the proposed claim validates the results.

Keywords: Spam Emails, Near Duplicate, SimHash, Spam Trees, SAG

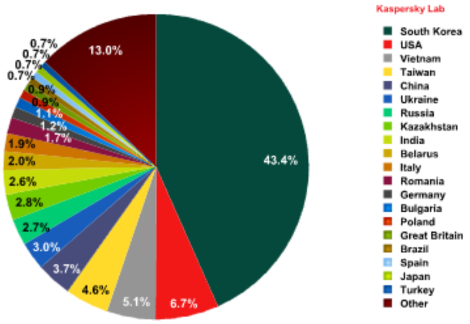
I INTRODUCTION

Spam, named after the Monty Python character, is unsolicited mail and typically consists of adverts for drugs, cheap mortgages, consumable items, pornographic websites, sex aids, “Too good to be true” scams. As of April 2013 Email Spam figures[1] estimations happens to be

- The percentage of spam in email traffic was up 2.1 percentage points compared with March and averaged 72.2%
- The percentage of phishing emails decreased by a factor of three compared with March, dropping to 0.002%
- In April, malicious files were found in 2.4% of all emails, a decrease of 1.6 percentage points

In April, although the quantity of spam emails grew very slightly – by 2.1 percentage points. The amount of ‘holiday’ spam reduced, even though spammers continued to actively exploit the Easter theme which happens to be the flavor of the season for spreading fraudulent emails and messages containing adverts for goods and services. Additionally, they tried to draw users’ attention with the names of the world political leaders and tragic events which happened recently.

Sources of spam in Europe by country.



How do spammers send mail?

A common practice of spammers is to create bulk accounts on free webmail services, to send spam or to receive e-mailed responses from potential customers. Because of the amount of mail sent by spammers, they require several e-mail accounts, and use programmable web bots to automate the creation and operation of these accounts. To try and combat this, most responsible webmail providers requires user to decipher a graphic (CAPTCHA images) to complete the registration.

Other methods include(Non content-based spamming procedures):

•Individual computers that have been infected with a virus / trojan – they connect to the Internet and download lists of email addresses and start sending out spam.

•Misconfigured email servers (open relay) - some people setup or reconfigure mail servers incorrectly and receive mail from anyone and then redeliver it - spammers love these as servers are usually on high-speed Internet connections so can send more spam quicker!

•Spammer-Friendly ISP's are are willing to take payment to setup servers and even offer to change IP addresses when those IP's get blacklisted.

•Spammers may buy mail server services from ISP's using stolen credit card details.

Since any of these approaches are practically feasible and are harder to curtail, receiver based spam handling measures gained prominence.

II RELATED WORK

Generally followed Spam Checking Criteria's for reporting mail as a spam or not can be done by monitoring the following parameters.

1)Bulk Mail to Clients

2)In Contacts ?

3)Mail Headers ?

4)Subject Headers ?

5)Message Bodies ?

6)HTML Emails ?

7)Black Listed sender Details using Centralized Repository

8)Black Listed content Details using Centralized Repository

Checking can be easily done when the repository of spam mails is small like hundreds or thousands of instances. When the size and the number of instances increasing to millions and more, it becomes impossible for human beings to check them one by one, which is complicated and error prone. Resorting to machine detectable schemes for such kind of repeatable job is desired, of which the core part is an algorithm that measures the difference between many pair of short messages, including duplicated and near duplicated ones.

III PRELIMINARIES

Since the e-mail spam problem is increasingly serious various techniques have been explored to solve the problem.

They can be categorized into the categories [2]:

a. Content-based methods

b. Non content-based methods.

Content based methods analyze e-mail content text and model this problem as a binary text classification task.

Naive Bayes[3] and Support Vector Machines

(SVMs) methods comes under this category. Naive Bayes [3] methods train a probability model using classified e-mails, and probability is assigned for each word in e-mails for making a key work as a suspicious spam keyword.

SVM [4], is a supervised learning method, which is an efficient and high performed text classification method. Markov random field model [5], logic regression [6] and nueral network [7], and certain specific features, such as URLs and images have also been taken into account for spam detection.

The other group analyzes non content information such as e-mail header, e-mail social network, and e-mail traffic [8] to filter spams.

Collecting notorious and innocent sender IP addresses or email addresses from e-mail header to create blocked list of senders and allowable mail list.

IV PROPOSED SCHEME

A. Structure Abstraction Generation

Structure Abstraction Generation [2] generates the e-mail abstraction using HTML content in e-mail. SAG [2] is composed of three major phases, Tag Extraction phase, Tag Reordering Phase, and <anchor> Appending Phase. In Tag Extraction Phase, the name of each HTML tag is extracted, and tag attributes and attribute values are eliminated. In addition, each paragraph of text without any tag embedded is transformed to <mytext/>. Since the arrangement of HTML tags are arranged in pairs, various sequential patterns of tags are contained in e-mails. In the worst case, if we consider two e-mail abstractions which have the same tag length and differ only in their last tags, the difference cannot be detected until the last tags are compared. To handle this problem, we destroy the regularity by rearranging the order of tag sequence to lower the number of tag comparisons. Note that this process ensures that the newly assigned position numbers of e-mail abstractions with the same number of tags are completely identical. In Tag Reordering phase each tag is assigned a newposition number (PN denotes for position number) with following expressions,

$$b = L1/2$$

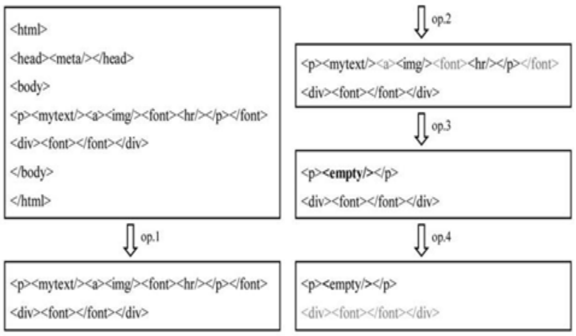
$$r= (PN_{orgi}-1) \%b$$

$$q= (PN_{orgi}-1)/b+1$$

$$PN_{new}= (b*r) + (b-q+1)$$

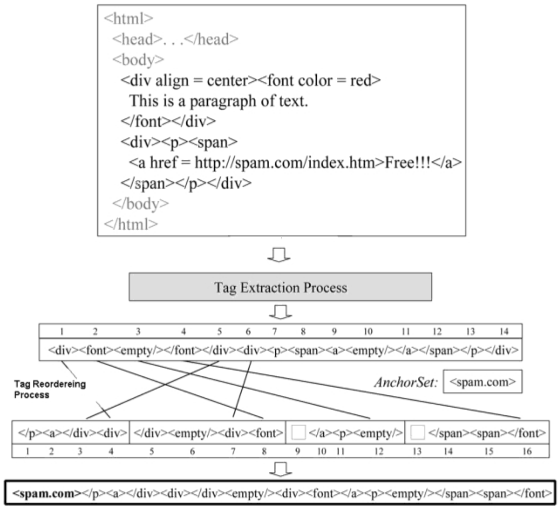
Where L is the tag length of an e-mail abstraction, and PNorig is the original position number. Variable b is the number of buckets. Variable r indicates which bucket should be placed and variable q is the number of shift counts from the end of this bucket

An example of the preprocessing step in Tag Extraction Phase of SAG.



First, an effective representation of e-mail (i.e., e-mail abstraction) is essential. Since a large set of reported spams has to be stored in the known spam database, the storage size of e-mail abstraction should be small. Moreover, the email abstraction should capture the near-duplicate phenomenon of spams, and should avoid accidental deletion of nonspam e-mails (also known as hams). Second, every incoming e-mail has to be matched with the large database, meaning that the near-duplicate matching process should be substantially efficient. Finally, the latest spams have to be included instantly and successively into the database so as to effectively block subsequent near-duplicate spams.

Procedure flow of Structure Abstraction Generation


<html>
<head>. . .</head>
<body>
<div align = center><font color = red>
This is a paragraph of text.
</font></div>
<div><p><span>
<a href = http://spam.com/index.htm>Free!!!</a>
</span></p></div>
</body>
</html>
Tag Extraction Process
1 2 3 4 5 6 7 8 9 10 11 12 13 14
<div><font><empty/></font></div><div><p><span><a><empty/></a></span></p></div>
Tag Reordereing Process
AnchorSet:
<spam.com>
</p><a></div><div>
</div><empty/><div><font>
</a><p><empty/>
</span><span></font>
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
<spam.com></p><a></div><div></div><empty/><div><font></a><p><empty/></span><span></font>

B. Design of Spam Tree:

SP tree [2] is a data structure to facilitate the process of near-duplicate matching. SpTable and SpTrees (sp stands for spam) are proposed to store large amounts of the e-mail abstractions of reported spams. Several SpTrees are the kernel of the database, and the e-mail abstractions of collected spams are maintained in the corresponding SpTrees. According to near duplicate definition, two e-mail abstractions are possible to be near-duplicate only when the numbers of their tags are identical. For efficient matching Sp Trees are designed to be binary trees. The branch direction of each SpTree is determined by a binary hash function. If the first tag of a subsequence is a start tag (e.g.,<div>), this subsequence will be placed into the left child node. A subsequence whose first tag is an end tag (e.g., </div>) will be placed into the right child node. Since most HTML tags are in pairs and the proposed e-mail abstraction is reordered in SAG, subsequences are expected to be uniformly distributed. Moreover, on level i of each SpTree (with the root on level 0), each node stores subsequences whose tag lengths are equal to 2i. For instance, as shown in Fig, the subsequence <spam:com> is placed into level 0, the subsequence </p><a> (whose tag length is 21) is placed into level 1, and so forth.

C. Spam Content Comparison Using BY SIMHASH

Charikar’s SimHash[9], actually, is a fingerprinting technique that produces a compact representation of the objects may be documents or images. So, it allows for various processing, once applied to original data sets, to be done on the compact sketches, a much smaller and well formatted (fixed length) space. With documents, SimHash works as follows: a Web document is converted into a set of features, each feature tagged with its weight. Then, we transform such a high dimensional vector into an f bit - fingerprint where f is quite small compared with the original dimensionality.

The calculation of the hash is performed in the following way:

a. Document is spitted into tokens (words for example) or super-tokens (word tuples)

b. Each token is represented by its hash value; a traditional hash function is used

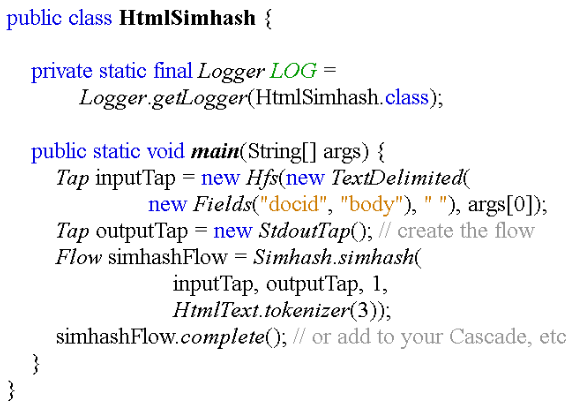
c. Weights are associated with tokens

d. A vector V of integers is initialized to 0, length of the vector corresponds to the desired hash size in bits

e. In a cycle for all token's hash values (h), vector V is updated: i^{th} element is decreased by token's weight if the i^{th} bit of the hash h is 0, otherwise i^{th} element is increased by token's weight if the i^{th} bit of the hash h is 1

f. Finally, signs of elements of V correspond to the bits of the final fingerprint

Sample program to show how SimHash works:



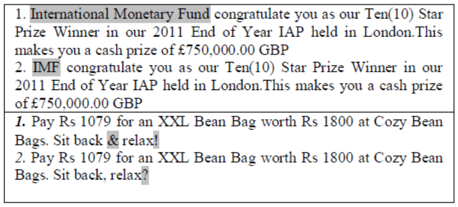
```
public class HtmlSimhash {

  private static final Logger LOG =
        Logger.getLogger(HtmlSimhash.class);

  public static void main(String[] args) {
    Tap inputTap = new Hfs(new TextDelimited(
              new Fields("docid", "body"), " "), args[0]);
    Tap outputTap = new StdoutTap(); // create the flow
    Flow simhashFlow = Simhash.simhash(
                    inputTap, outputTap, 1,
                    HtmlText.tokenizer(3));
    simhashFlow.complete(); // or add to your Cascade, etc
  }
}
```

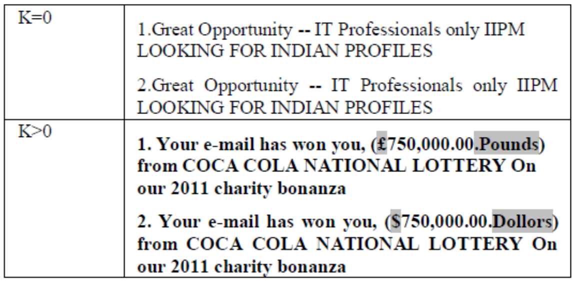
V PERFORMANCE

In this paper, we show that SimHash is indeed Effective and efficient in detecting both duplicate (with k = 0) and near-duplicate (with k > 0) (see the two typical examples in TABLE II.) Among large short message repository. However, we also notice that due to the born feature of short messages, k = 3 may not be an Ideal parameter. For k = 2 is enough to detect the one-character difference, but k has to be 5 to detect the same pair of messages with two-character difference. Besides, with the same one-character difference, short messages require larger k for effective detection. This may be explained by an observation, that the same difference, e.g. having one different character on the same position of two spam messages, would be more influential to short text than to long text.



1. International Monetary Fund congratulate you as our Ten(10) Star Prize Winner in our 2011 End of Year IAP held in London.This makes you a cash prize of £750,000.00 GBP 2. IMF congratulate you as our Ten(10) Star Prize Winner in our 2011 End of Year IAP held in London.This makes you a cash prize of £750,000.00 GBP
1. Pay Rs 1079 for an XXL Bean Bag worth Rs 1800 at Cozy Bean Bags. Sit back & relax! 2. Pay Rs 1079 for an XXL Bean Bag worth Rs 1800 at Cozy Bean Bags. Sit back, relax?

Table 1. Typical near-duplicates of spam mails with differences highlighted in grey



K=0	1.Great Opportunity -- IT Professionals only IIPM LOOKING FOR INDIAN PROFILES 2.Great Opportunity -- IT Professionals only IIPM LOOKING FOR INDIAN PROFILES
K>0	**1. Your e-mail has won you, (£750,000.00.Pounds) from COCA COLA NATIONAL LOTTERY On our 2011 charity bonanza** **2. Your e-mail has won you, ($750,000.00.Dollors) from COCA COLA NATIONAL LOTTERY On our 2011 charity bonanza**

Table 2. Example: detect duplicate with k =0 and near-duplicate with k >0 (with differences highlighted in gray)

This is a paper focusing on discussing Solution for real application. Firstly, we demonstrate a series of practical values of SimHash-based approach by experiments and our experience. Secondly, we point out that k = 3 may be suitable for near duplicated spam mail detection, but obviously not suitable for short messages. Thirdly, we propose one empirical choice, k = 5, as applied on our Online short message search.

ADVANTAGES AND DISADVANTAGES OF SIMHASH

SimHash has several advantages for application based on our experience:

a. Transforming into a standard fingerprint makes it applicable for different media content, no matter text, video or audio;

b. Fingerprinting provides compact representation, which not only reduces the storage space greatly? but allows for quicker comparison and search.

c. Similar content has similar SimHash code, which permits easier distance function to be? Determined for application.

d. It is applicable for both duplicate and near duplicate Detection, with k = 0 and k > 0 respectively.

e. Similar processing time for different setting of k if via the proposed divide-and-search mentioned above, and this is valuable for practice since we are able to detect more near duplicates with no extra cost.

f. The search procedure of similar encoded objects is easily to be implemented in distributed environment based on our implementation experience.

g. From the point of software engineering view, this procedure may be implemented into standard module and be re-used on similar applications, except that the applicants may determine the related parameters themselves.

VI. CHALLENGES TO DETECT SPAM E-MAILS

Now a day, spammers are becoming more and more sophisticated. They are finding ways to trick people into thinking that their unsolicited junk messages are worth the time you spend reading them. Some users may understand it as a spam and sends it to spam box but some users consider it as worthy and opens it. We specify some of the rules for specifying a mail as a spam mail A. It is placed in Spam Folder: Sometimes we unknowingly categorize a legitimate email as spam, and emails from certain websites end up in the spam folder. We must deal with issue on a case-by-case basis to determine whether the mail is a legitimate or garbage into your inbox. B. By seeing

Email Address: Legitimate companies send emails through a server based out of their company website like support@ companyname.com. If we have a long string of numbers in front of the @ sign or the name of a free email service before the .com or any other domain, we need to question the legitimacy of the email. C. Content of the mail: Sometimes mails may be consisting of content which tells us to do something with in a period of time like hours or days and it may consist of links that may be leading us to some other website. Most companies tell you what to do, but they never direct you to where to do it with a link. Mails contains spelling mistakes purposefully have the chance to be a spammer. Spammers don't care enough about the actual messages they're sending to take the time to make them make sense. D. Spam's ask for personnel Information: Legitimate institutions never ask for personal information in an email. They don't need to ask you for your personal information anyway because they usually have it on hand. So, if you get an email that asks you for any personal information, no matter how legitimate it might seem, delete it right away. Personal information is only meant to be entered in secure, encrypted forms, not emails where anyone and everyone can get their hands on your information. E. By Seeing Greeting in the mail: When you receive a genuine email, the sender addresses you directly, using either your first or last name. If you receive an email where they refer to you as a ―Valued Customer‖ or as a member of some company, its spam. Senders of your genuine emails want to get your attention, so they always address you directly.

VII CONCLUSION AND FUTUREWORK

Compared to the existing methods in prior research, in this paper, we use an innovative customized tree data structure called SpTrees, to store large amounts of the e-mail abstractions of reported spams. To achieve efficient matching with balanced tree structure, SpTrees are designed to be binary trees. The branch direction of each SpTree is determined by a binary hash function of known spam words. The improvement is limited since we map each subsequence in a node of an SpTree to a hash value. Therefore, the subsequences that have some prefix tags in common still can be differentiated with one comparison. In this paper, Instead of mapping each subsequence in a node of an SpTree to a hash value using a binary hash function we propose to replace it with a special hash function, namely Simhash. The advantage of this over other hash functions is that it sets a minimum on the number of members that the two sets must share in order to match. This mitigates the effect of extremely common set members on data clusters. SimHash based approach is Fast, Flexible, Customizable (HtmlSimhash), Scalable and is Google patented that validate its efficiency. As it happens to be prevention is better than cure, so is usage of new age bot defeating procedures deployed at mail service providers can decrease the load on detection mechanisms which can interesting future research.

VIII REFERENCES

[1] For Spam Statistics (http://www.securelist.com/en/analysis/204792293/Spam_in_April_2013)

[2]. Chi-Yao Tseng, Pin-Chieh Sung, and Ming-Syan Chen―Cosdes: A Collaborative Spam DetectionSystem with a Novel E-Mail Abstraction Scheme,IEEE transactions on knowledge and data engineering, vol. 23, no. 5, may 2011

[3]. V. Metsis, I. Androutsopoulos, and G. Paliouras, ―Spam Filtering with Naive Bayes—Which Naive Bayes? Proc. Third Conf. Email and Anti-Spam (CEAS), 2006.

[4].E. Blanzieri and A. Bryl, ―Evaluation of the Highest Probability SVM Nearest Neighbor Classifier with Variable Relative Error Cost, Proc. Fourth Conf. Email and Anti-Spam (CEAS), 2007.

[5].S. Chhabra, W.S. Yerazunis, and C. Siefkes, ―Spam Filtering Using a Markov Random Field Model with Variable Weighting Schemas, Proc. Fourth IEEE Int'l Conf. Data Mining (ICDM), pp. 347-350, 2004.

[6].M.-T. Chang, W.-T. Yih, and C. Meek, ―Partitioned Logistic Regression for Spam Filtering,‖ Proc. 14th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data mining (KDD), pp. 97-105, 2008.

[7].A.C. Cosoi, ―A False Positive Safe Neural Network; The Followers of the Anatrim Waves, Proc. MIT Spam Conf., 2008.

[8].R. Clayton, ―Email Traffic: A Quantitative Snapshot,Proc. of the Fourth Conf. Email and Anti-Spam (CEAS), 2007

[9].M. S. Charikar. Similarity estimation techniques from Rounding

[10]. Algorithms. In Proc. 34th Annual ACM Symposium on Theory of Computing, pages 380-388. ACM, 2002.