ijecs
open access www.ijecs.in

International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 2 Issue 12 Dec,2013 Page No. 3538-3540

E-Mail Abstraction Scheme Using Collaborative Spam Detection Scheme

Vinod.S^1*, Insozhan.N*^{2}*, Vimal.V.R*^{3}

Computer Science & Engineering[1, 2, 3], Assistant Professor[1, 2, 3]

vinodsundaram.s@gmail.com(1), sozhanme@gmail.com(2), vimalraman2004@gmail.com(3)

Veltech Multitech Dr.Rangarajan Dr.Sakunthala Engineering College, Avadi, Chennai-600 062

Abstract- *Email communication is widely spread and essential nowadays. However, the threat of unsolicited junk emails, also known as spam, becomes more and more serious. The basic idea of the similarity matching schema for spam detection is to maintain a known spam database, formed by user feedback, to block subsequent near-duplicate spam. By achieving efficient similarity matching and reducing storage utilization, prior works mainly represent each email by a succinct abstraction derived from email content text. But, these abstractions of emails cannot fully catch the evolving nature of spam, and are thus not effective enough in near-duplicate detection. An email abstraction scheme is proposed, which considers email layout structure to represent emails. Procedure SAG(Structure Abstraction Generation) is presented to generate the email abstraction using HTML content in email, and this newly-devised abstraction can more effectively capture the near-duplicate phenomenon of spam. Moreover, we design a complete spam detection system which possesses an efficient near-duplicate matching scheme and a progressive update scheme. The progressive update scheme enables this system to keep the most up-to-date information for near-duplicate detection.*

Keywords- *SAG(Sturcture Abstraction Generation), SVMs(Support Vector Machine), TD(Time spam for Triggering deletion Handler).*

I. INTRODUCTION

Email communication is prevalent and indispensable nowadays. However, the threat of unsolicited junk emails, also known as spam, becomes more and more serious. The primary idea of the similarity matching schema for spam detection is to maintain a known spam database, formed by user feedback, to block subsequent near-duplicate spam. On purpose of achieving efficient similarity matching and reducing storage utilization, prior works mainly represent each email by a succinct abstraction derived from email content text. However, this abstraction of emails cannot fully catch the evolving nature of spam, and are thus not effective enough in near-duplicate detection.

We propose a novel email abstraction scheme, which considers email layout structure to represent emails. Procedure SAG(Structure Abstraction Generation) is presented to generate the email abstraction using HTML content in email, and this newly-devised abstraction can more effectively capture the near-duplicate phenomenon of spam. Moreover, we design a complete spam detection system which possesses an efficient near-duplicate matching scheme and a progressive update scheme. The progressive update scheme enables system to keep the most up-to-date information for near-duplicate detection.

Collaborative filtering indicates that user knowledge of what spam may subsequently appear is collected to detect following spams. Overall, there are three key points of this type of spam detection approach we have to be concerned about. First, an effective representation of email (i.e., email abstraction) is essential. Since a large set of reported spams has to be stored in the known spam database, the storage size of email abstraction should be small. Moreover, the email abstraction should capture the near-duplicate phenomenon of spams, and should avoid accidental deletion of non-spam emails (also known as hams). Second, every incoming email has to be matched with the large data-base, meaning that the near-duplicate matching process should be subsequently efficient. Finally, the latest spams have to be included instantly and successively into the database so as to effectively block subsequent near-duplicate spams.

II. CHARACTERISTICS

A. *Structure Abstraction Generation (SAG):*

Procedure SAG is to generate the email abstraction using HTML content in email. Procedure SAG 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. On purpose of accelerating the near-duplicate matching process, we reorder the tag sequence of an email abstraction in Tag Reordering Phase.

B. *SpTables and SpTrees:*

One major focus of this work is to design the innovative datastructure to facilitate the process of near-duplicate matching. SpTables and SpTrees (Sp stands for Spam) are proposed to store large amounts of the email abstractions of reported spams. The email abstractions of collected spams are maintained in the corresponding SpTrees. An email abstraction is segmented into several sub-sequences, and these sub-sequences are consecutively put into the corresponding nodes from low levels to high levels. As such, an email abstraction is stored in one path from the root node to a leaf node of SpTree, and hence the matching between a testing email and known spams is processed from root to leaf.

III. COMPLETE SPAM DETECTION SYSTEM

The system model of spam detection system is illustrated by three parameters, Tm (the maximum time span for reported spams being retained in the system), Td (the time span for triggering Deletion Handler), and Sth (the score threshold for determining spams) should be given for spam detection system. Before starting to do the spam detection, spam detection system collects feedback spams for time in advance to construct an initial database.

Three major modules, Abstraction Generation Module, Database Maintenance Module, and Spam Detection Module, is included in this system. Abstraction Generation Module, each email is converted to an email abstraction by Structure Abstraction Generator with procedure SAG. Three types of action handlers, Deletion Handler, Insertion Handler, and Error Report Handler, are involved in Database Maintenance Module. Note that although the term "database" is used, the collection of reported spams can be essentially stored in main memory to facilitate the process of matching. In addition, Matching Handler in Spam Detection Module takes charge of determining results.

A. *Matching Handler:*

Matching Handler is the most significant procedure to achieve efficient matching between every testing email and the known spam database to detect whether the email is spam or not.

B. *Insertion Handler:*

Initially, the corresponding SpTree is found in SpTable according to the tag length of the inserted spam, and is assigned as the root of this SpTree. Then we iteratively insert the sub-sequences of the email abstraction along the path from root to leaf. Then, the node is assigned as the corresponding child node based on the type of the next tag. If the next tag is a start (end) tag, is assigned as the left (right) child node.

C. *Error Report Handler:*

When receiving a misclassified ham as an input to the Error Report Handler. We find the corresponding SpTree and do the matching process as the same in Matching Handler. For the spams matched with the reported misclassified ham, we reset of these spams as to avoid subsequent misclassification incurred by the identical group of spams. In addition, the reputation scores of reporters who cause the false positive error are halved to prevent continuous attacks by specific users.

D. *Deletion Handler:*

To delete obsolete spams, for every Td (the time span for triggering Deletion Handler), Deletion Handler traverses each SpTree in order to visit all nodes in SpTrees. If the existing time exceeds Tm, it will be viewed as outdated and be deleted from this node. As such, all obsolete spams are removed from the known spam database after Deletion Handler is processed.

IV. EXISTING SYSTEM

Based on what features of emails are being used, previous works on spam detection can be generally classified into three categories: (a) content- based methods, (b) non-content-based methods, and (c) others. Initially, researchers analyze email content text and model this problem as a binary text classification task. Representatives of this category are Naïve Bayes and Support Vector Machines (SVMs) methods. one major disadvantage is that it is cost prohibitive for large-scale applications to constantly re- train these methods with the latest information to adapt to the rapid evolving nature of spams. The spam detection of these methods on the email corpus with various languages has been less studied yet.

The other group attempts to exploit non-content information such as email header, email social network, and email traffic to filter spams. Collecting notorious and innocent sender addresses (or IP addresses) from email header to create black list and white list is a commonly applied method initially. Since email header can be altered by spammers to conceal the identity, the main drawback of these methods is the hardness of correctly identifying each user.

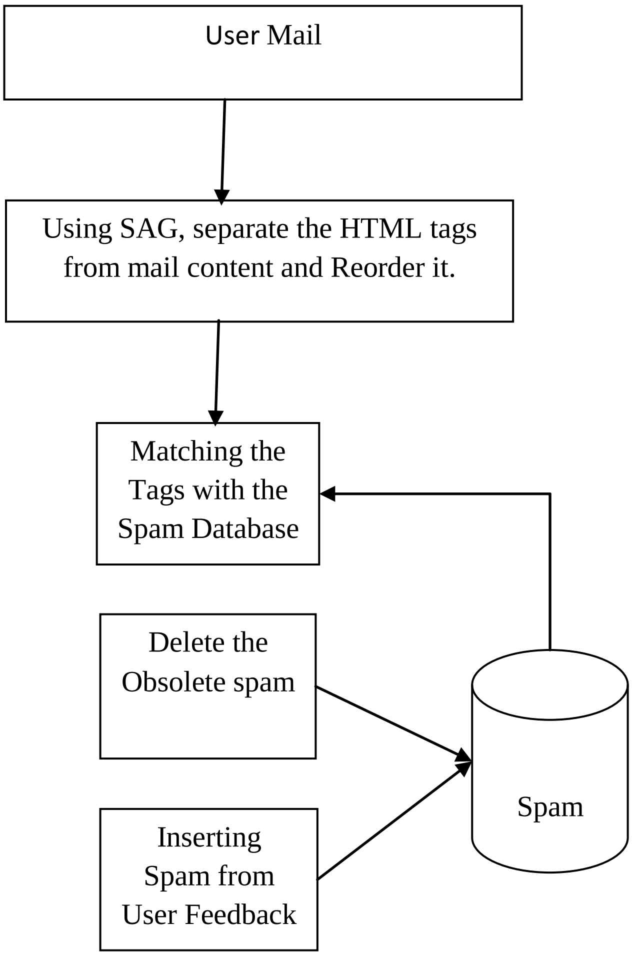
V. PROPOSED SYSTEM

In the field of collaborative spam filtering by near-duplicate detection, a superior email abstraction scheme is required to more certainly catch the evolving nature of spams. Compared to the existing methods in prior research, in this paper, we explore a more sophisticated and robust email abstraction scheme, which considers email layout structure to represent emails. The specific procedure SAG is proposed to generate the email abstraction using HTML content in email, and this newly-devised abstraction can more effectively capture the near-duplicate phenomenon of spams. Moreover, a complete spam detection system has been designed to efficiently process the near-duplicate matching and to progressively update the known spam database. Consequently, the most up-to-date information can be invariably kept to block subsequent near-duplicate spams. Since we ignore the semantics of the text, the proposed abstraction scheme is inherently applicable to emails in all languages.

VI. SYSTEM DESIGN

The system model complete spam detection and the algorithmic form is outlined below. Initially, three parameters,(the maximum time span for reported spams being retained in the system),(the time span for triggering Deletion Handler), and(the score threshold for determining spams) should be given for spam detection. Before starting to do the spam detection, spam detection collects feedback spams for time in advance to construct an initial database. Three major modules, Abstraction Generation Module, Database

Maintenance Module, and Spam Detection Module, are included in spam detection. With regard to Abstraction Generation Module, each email is converted to an email abstraction by Structure Abstraction Generator with procedure SAG. Three types of action handlers, Deletion Handler, Insertion Handler, and Error Report Handler, are involved in Database Maintenance Module. Note that although the term "database" is used, the collection of reported spams can be essentially stored in main memory to facilitate the process of matching. In addition, Matching Handler in Spam Detection Module takes charge of determining results. There are three types of emails, reported spam, testing email, and misclassified ham. When receiving a reported spam, Insertion Handler adds the email abstraction of this spam into the database except that the reputation score of this reporter is too low. Whenever a new testing email arrives, Matching Handler performs the near-duplicate detection with collected spams to do the judgment. Meanwhile, if a testing email is classified as a spam, this email will be viewed as a reported spam and be added into the database. Moreover, Error Report Handler copes with feedback misclassified hams and adjusts by Degrading the reputation of related reporters to prevent malicious attacks. For every, Deletion Handler is triggered to delete obsolete spams which exist over time.


User Mail
Using SAG, separate the HTML tags from mail content and Reorder it.
Matching the Tags with the Spam Database
Delete the Obsolete spam
Inserting Spam from User Feedback
Spam

System model of complete spam detection system

The main functionalities of deleting outdated spams are not only to alleviate the overhead of the server, but to reduce the risk of accidental deletion of hams. Due to the evolving nature of spams, it is inappropriate to utilize old spams to filter current ones. Overall, this paper is self-adjusting and retains the most up-to-date spam for near-duplicate detection.

VII. CONCLUSION

In the field of collaborative spam filtering by near-duplicate detection, a superior email abstraction scheme is required to more certainly catch the evolving nature of spams. Compared to the existing methods in prior research, in this paper, we explore a more sophisticated and robust email abstraction scheme, which considers email layout structure to represent emails. The specific procedure SAG is proposed to generate the email abstraction using HTML content in email, and this newly devised abstraction can more effectively capture the near-duplicate phenomenon of spams. Moreover, a complete spam detection system has been designed to efficiently process the near-duplicate matching and to progressively update the known spam database. Consequently, the most up-to-date information can be invariably kept to block subsequent near-duplicate spams. In the experimental results, we show that complete spam detection system significantly outperforms competitive approaches, which indicates the feasibility of spam detection in real world application.

REFERENCES

[1] E. Blanzieri and A. Bryl. Evaluation of the highest probability svm nearest neighbor classifier with variable relative error cost.*Proc. of the 4th Conference on Email and Anti-Spam (CEAS)*, 2007.

[2] M.-T. Chang, W.-T. Yih, and C. Meek. Partitioned logistic regression for spam filtering. *Proc. of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data mining (KDD)*, pages 97–105, 2008.

[3] S. Chhabra, W. S. Yerazunis, and C. Siefkes. Spam filtering using a markov random field model with variable weighting schemas. *Proc. of the 4th IEEE International Conference on Data Mining (ICDM)*, pages 347–350, 2004.

[4] P.-A. Chirita, J. Diederich, and W. Nejdl. Mailrank: Using ranking for spam detection. *Proc. of the 14th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 373–380,2005.

[5] R. Clayton. Email traffic: A quantitative snapshot. *Proc. of the 4th Conference on Email and Anti-Spam (CEAS)*, 2007.

[6] A. C. COSOI. A false positive safe neural network the followers of the anatrim waves. *Proc. of the MIT Spam Conference*, 2008.

[7] E. Damiani, S. D. C. di Vimercati, S. Paraboschi, and P. Samarati.An open digest-based technique for spam detection. *Proc. of the 2004 International Workshop on Security in Parallel and Distributed Systems*, pages 559–564, 2004.

[8] E. Damiani, S. D. C. di Vimercati, S. Paraboschi, and P. Samarati.P2p-based collaborative spam detection and filtering. *Proc. of the 4th IEEE International Conference on Peer-to-Peer Computing*, pages 176–183, 2004.