Non-ML Anti-Spamming: A Role Based Solution

Anthony Y. Fu, Email: anthony@cs.cityu.edu.hk

WebPage: http://www.cs.cityu.edu.hk/~anthony

Department of Computer Science, City University of Hong Kong

Hong Kong SAR

1. Introduction

Anti-Spam has become a hot topic in email research and email services. It is very annoying at receiving many emails with unwanted and even disgusting contents. Former researchers have done a lot of researches using machine learning (ML) or machine learning related techniques. The major techniques include Bayesian Based Classifier, Support Vector Machine, Neural Network, Decisions Trees, Centroid-Based Classifier, Document Space Density Based Recognition, Latent Semantic Indexing, and Logic Programming. All of these methods have achieved high spam recognition precision and recall percentages. Some excellent techniques have even been implemented into tools and used by millions of people around the world. But none of them can stop spam without any mistake. The reason that machine learning techniques can not fully stop spam is very clear. People are using mathematical methods, computers and algorithms to simulate the functions of human brain. Unfortunately, human brain is too complicated that machines can not imitate it perfectly. The authors have investigated almost all of the anti-spam techniques, but can not see any existing technique achieving anti-spam much well. Then maybe the non-machine-learning anti-spam techniques should be investigated. In this paper, we will introduce one of these kinds of techniques. The contribution of this paper is that, we will introduce a non-ML anti-spamming method, a role based technique, to highly reduce the severe spam problem. And hopefully, this method could play an important role in anti-spamming society.

2. Definitions

For easy discussion, several definitions will be introduced in this section. Suppose there is an email sender *Alice* whose email address is *alice@A.com*, and an email receiver *Bob* whose email address is *bob@B.com*. Alice will send *Bob* an email. Alice could be a legal sender or spammer, and *Bob* will receive the email from *Alice*, in case server *B.com* does not recognize the emails as a spam.

Friend List (*FL*) and Black List (*BL*)

Each email account has a friend list *FL* and black list *BL*. They are both email lists which can record email addresses. *FL* records the email addresses that people wish to contact with, and *BL* contains the email addresses that people refuse to contact with. There is a rule that $FL \cap BL = \phi$, and if one email address both exists in *FL* and *BL*, then the one in *FL* will be deleted.

Certificate Authentication (*CA*)

Certificate Authentication (*CA*) should be used to guarantee an email with the parameter *sender_field=alice@A.com* in the email header is sent by *Alice*. It is well known by people that most current email systems allow the email senders to revise the *sender_field* freely, and many spammers are abusing this email system mechanism to send unwanted emails to their victims. The usage of *CA* will help *Bob* be sure about that nobody else could fake an email with *sender_field=alice@A.com* to him unless *Alice* sends it herself. If *Bob* receives an email with no *CA*, he has the right to choose to request the *CA* from *Alice* or just reject this email. The request on *CA* could be sent in an email form for fewer changes of the email systems we are using.

Question Server

A *Question Server* should be installed for each Email Server. A *Question Server* is an application to assist an *Email Server* to make confirmation that the email sender has a very high probability to be a human rather than a robot. To achieve this, the questions asked by the *Question Server* should be easy for human being to answer but difficult for robots to answer. And for better to leave the current email protocol untouched, the question can be asked and answered in an email with a simple tag based form. E.g. *<question>What is the result of "1+1"?</question>*. And the correct answer can be : *<ans>2</ans>*.

Communication Request Email (*CRE*)

Communication Request Email (CRE) is an email for Bob to confirm the following operation after both the *sender_field* and non-robot property of the email is guaranteed. It will be sent to the inbox of *bob@B.com*. When *Bob* checks his email, he will find this *CRE* and chose to add *alice@A.com* to *FL* or *BL*, to delete this *CRE*, or to ask *B.com* to send the email from *alice@A.com* to his inbox without doing any other thing.

3. Anti-Spamming

We will discuss the whole process in our anti-spam model. Initially, the *FL* and *BL* of *bob@B.com* are both empty. There is an option for Bob to choose: he wants to use anti-spam function or not, and we suppose *Bob* has chosen anti-spam function. Now *Bob* receives an email with *sender_field=alice@A.com*. We will have 6 cases to discuss, and each case can be represented by a triple *<Has_CA, In_FL, In_BL>*, where *Has_CA ∈{False, True}*, and *<In_FL, In_BL> ∈{<False, False>, <False, True>, <True, False> }*. For each case, Email Server B.com will react respectively.

Case1, <*False, False, False*>. In this case, the email does not have *CA*, so this email is not guarantied to be sent by *Alice*. And *alice@A.com* is not in *FL* and *BL* of *bob@B.com*, so *B.com* should not reject this email. Thus *Alice* will be asked a question by *Question Server* of *B.com* through email. If the question is successfully answered, and again, *send_field= alice@A.com*, then *B.com* will believe that this email is sent from *alice@A.com* and *Alice* is not a robot. Then a *CRE* will be sent to the inbox of *bob@B.com*. When *Bob* checks his email, he will find this *CRE* and chose to add *alice@A.com* to *FL* or *BL*, to delete this *CRE*, or to ask *B.com* to send the email from *alice@A.com* to his inbox without doing any other thing.

Case2, <*False, False, True*>. In this case, the email does not have a *CA*, and the email address is in *BL* of *bob@B.com*. Thus *B.com* will not accept this email, and a rejection email will be sent to *alice@A.com*.

Case3, <*False, True, False*>. In this case, *alice@A.com* is in the *FL* of *bob@B.com*, but this email does not have a *CA*. Although the email address is in *FL*, it does not guarantee this email is not faked by somebody else. So the *Question Server* will send a question email to *alice@A.com*. Whether the answer email correctly solve the question or not, the email will be sent to the inbox of *bob@B.com*.

Case4, <*True, False, False*>. In this case, this email has a *CA*. This means that this email is sent from *alice@A.com*. But it is not guaranteed that this email is not sent by a robot. So the *Question Server* will send a question email to *alice@A.com*. Only if there is a reply and the reply is correct, a *CRE* will be sent to the inbox of *bob@B.com*. When *Bob* checks his email, he will find this *CRE* and chose to add *alice@A.com* to *FL* or *BL*, to delete this *CRE*, or to ask *B.com* to send the email from *alice@A.com* to his inbox without doing any other thing.

Case5, <*True, False, True*>. In this case, although this email has a *CA*, the email address is in *BL* of *bob@B.com*. Thus *B.com* will not accept this email, and a rejection email will be sent to *alice@A.com*.

Case6, <*True, True, False*>. In this case, this email has a *CA*, and the email address is in *FL* of *bob@B.com*. Thus *B.com* will accept this email directly.

So far, the new coming email for *Bob* could be handled according to the above cases. These cases can be presented in a table, as shown in Table1.

Has_CA	*In_FL*	*In_BL*	The processes in *B.com*
False	*False*	*False*	Send question email to *alice@A.com*. If answer is right, then send a *CRE* to *bob@B.com*, otherwise reject the email.
False	*False*	*True*	Reject the email
False	*True*	*False*	Send question email to *alice@A.com*. Whether answer is right or not, this email will be accepted.
True	*False*	*False*	Send question email to *alice@A.com*. If answer is right, then send a *CRE* to *bob@B.com*.
True	*False*	*True*	Reject this email.
True	*True*	*False*	Accept this email

Table1. Anti-Spamming Reaction Table for *B.com*

According to the above discussion, spammers will not have chance to send junk emails to other people. The question "What if *Alice* applies many email accounts and sends many conversation requests to *Bob*?" could be successfully solved. The reason is, even if *Alice* applies many email accounts to send requests to *Bob*, *Alice* should answer many questions form *Question Server*, and

that means to answer millions of questions. And we have an assumption that these questions must be answered by human, so spammers will spend a lot of man-hours to do it.

Another problem should be mentioned is that "what will happen in case a teacher wants to broadcast an email to his 1000 students, should the teacher answer 1000 questions for accomplishing the conversation request process?". The answer will be very simple. Firstly, in most cases, broadcasting emails are in-system-emails. E.g. a teacher in *T@cityu.edu.hk* can broadcast emails to students using *S@cityu.edu.hk*. In this case, system administrator can setup the system to allow teachers to broadcast emails to students or teachers, but student can not do this. (System administrator can control the friend list). Secondly, in some cases, broadcasting emails are sent to other domains. E.g. a professor is organizing a conference, and he wants to broadcast emails to attendees. In this case, attendees should have known the consequences of NOT adding this important email address in his friend list in advance; otherwise he will only receive a *CRE* asking him to add this important email to his friend list, if the broadcasted email includes a *CA* of this professor.

4. Conclusion and Future Work

As machine learning techniques in anti-spamming have been discussed a lot by former researchers, in this paper, we have introduced a new anti-spam approach with role based method. Authors have used a very simple email communication model, *Alice* and *Bob*, and the concepts, *FL*, *BL*, *CA*, *Question Server*, and *CRE* to achieve effective anti-spamming result.

Although the paper is discussing about a non-ML technique, ML is a very hopeful tool for anti-spamming. And maybe it can be combined with the method introduced in this paper to achieve better effect. E.g. the email sent by *Alice* will be preprocessed using Naive Bayesian

classifier. This email will be put into the inbox of *bob@B.com* directly iff the email is approximately 100% recognized as a ham by Naive Bayesian classifier. Finally, we hope this work could play an important role in the anti-spamming society in the future.

Acknowledgement

I would like to thank Professor Xiaotie Deng, who has led me to this interesting anti-spamming topic. Thank you Dr. Xiaojian Tian and Dr. C.K. Poon, your valuable discussion with me always inspire to good ideas. Thanks to Tommy G. Yang for revising my clumsy manuscript. And thank you all researchers working in anti-spamming field, hope we can stop the annoying spam problem in the near future.