Efficient Spam Email Filtering using Adaptive Ontology

Seongwook Youn and Dennis McLeod

Computer Science Department, University of Southern California
Los Angeles, CA 90089, USA
{syoun, mcleod}@usc.edu

Abstract

Email has become one of the fastest and most economical forms of communication. However, the increase of email users has resulted in the dramatic increase of spam emails during the past few years. As spammers always try to find a way to evade existing filters, new filters need to be developed to catch spam. Ontologies allow for machine-understandable semantics of data. It is important to share information with each other for more effective spam filtering. Thus, it is necessary to build ontology and a framework for efficient email filtering. Using ontology that is specially designed to filter spam, bunch of unsolicited bulk email could be filtered out on the system. This paper proposes to find an efficient spam email filtering method using adaptive ontology

Keywords: spam, ontology, data mining, classification

1. Introduction

Email has been an efficient and popular communication mechanism as the number of Internet users increases. Therefore, email management became an important and growing problem for individuals and organizations because it is prone to misuse. The blind posting of unsolicited email messages, known as spam, is an example of misuse. Spam is commonly defined as sending of unsolicited bulk email - that is, email that was not asked for by multiple recipients. A further common definition of a spam is restricted to unsolicited commercial email, a definition that does not consider non-commercial solicitations such as political or religious pitches, even if unsolicited, as spam. Email was by far the most common form of spamming on the internet.

According to the data estimated by Ferris Research [8], spam accounts for 15% to 20% of email at U.S.-based corporate organizations. Half of users are receiving 10 or more spam emails per day while some of them are receiving up to several hundreds unsolicited emails. International Data Group [11] expected that global email traffic surges to 60 billion messages daily by 2006. It involves sending identical or nearly identical unsolicited messages to a large number of recipients. Unlike legitimate commercial email, spam is generally sent without the explicit permission of the recipients, and frequently contains various tricks to bypass email filters.

Modern computers generally come with some ability to send spam. The only necessary ingredient is the list of addresses to target. Spammers obtain email addresses by a number of means: harvesting addresses from Usenet postings, DNS listings, or Web pages; guessing common names at known domains (known as a dictionary attack); and "e-pending" or searching for email addresses corresponding to specific persons, such as residents in an area. Many spammers utilize programs called web spiders to find email addresses on web pages, although it is possible to fool the web spider by substituting the "@" symbol with another symbol, for example "#", while posting an email address. As a result, users have to waste their valuable time to delete spam emails. Moreover, because spam emails can fill up the storage space of a file server quickly, they could cause a very severe problem for many websites with thousands of users.

Currently, much work on spam email filtering has been done using the techniques such as decision trees, Naive Bayesian classifiers, neural networks, etc. To address the problem of growing volumes of unsolicited emails, many different methods for email filtering are being deployed in many commercial products. We constructed a framework for efficient email filtering using ontology. Ontologies allow for machine-understandable semantics of data, so it can be used in any system [19]. It is important to share the information with each other for more effective spam filtering. Thus, it is necessary to build ontology and a framework for efficient email filtering. Using ontology that is specially designed to filter spam, bunch of unsolicited bulk email could be filtered out on the system. This paper proposes to find an efficient spam email filtering method using ontology. We used Waikato Environment for Knowledge Analysis (Weka) explorer, and Jena to make ontology based on sample dataset.

Emails can be classified using different methods. Different people or email agents may maintain their own personal email classifiers and rules. The problem of spam filtering is not a new one and there are already a dozen

different approaches to the problem that have been implemented. The problem was more specific to areas like artificial intelligence and machine learning. Several implementations had various trade-offs, difference performance metrics, and different classification efficiencies. The techniques such as decision trees, Naive Bayesian classifiers, and Neural Networks had various classification efficiencies. The remainder of the paper is organized as follows: Section 2 describes existing related works; Section 3 introduces our idea of spam filtering using ontology; Section 4 discusses the experimental result of the framework that we proposed; Section 5 concludes the paper with possible directions for future work.

2. Related Work

Bringing in other kinds of features, which are spam-specific features in their work, could improve the classification results [17]. A good performance was obtained by reducing the classification error by discovering temporal relations in an email sequence in the form of temporal sequence patterns and embedding the discovered information into content-based learning methods [13]. [15] showed that the work on spam filtering using feature selection based on heuristics.

Approaches to filtering junk email are considered [3], [6], [17]. [7] and [9] showed approaches to filtering emails involve the deployment of data mining techniques. [4] proposed a model based on the Neural Network (NN) to classify personal emails and the use of Principal Component Analysis (PCA) as a preprocessor of NN to reduce the data in terms of both dimensionality as well as size. [1] compared the performance of the Naïve Bayesian filter to an alternative memory based learning approach on spam filtering.

In contrast to previous approaches, ontology was used in our approach. In addition, J48 was used to classify the training dataset. Ontology created by the implementation is modular, so it could be used in another system. In our previous classification experiment, J48 showed better result than Naïve Bayesian, Neural Network, or Support Vector Machine (SVM) classifier.

3. Spam Filtering using Ontology

3.1. Approach

An assumption to create decision trees would be the intelligence behind the classification, but this was not enough because the decision tree ultimately is not a true ontology and also, querying a decision tree was also not easy. Once, we narrowed down on the type of decision tree that we going use, the next step were to create an ontology based on the classification result through J48. Resource Description Framework (RDF) which would be the form of "Subject – Object – Predicate" was used to create an ontology. Hence, our second main assumption was that we will need to map the decision tree into a formal ontology and query this ontology using our test email to be classified as spam or not. The test email is another thing we needed to consider because firstly, it is very difficult to deploy our system in such a way that it could read an incoming mail on a mail server and this would require a lot of extra work which would make the work unnecessarily complicated.

The initial step was to gather a good dataset on which the decision tree will be based. This data should consider the characteristics of spam email as well as the non-spam email. Also the attributes and the values for each type of email must be such that the decision tree based on the training data will not be biased. The dataset that we used obtained from UCI Machine Learning Lab [14]. We evaluated a number of implementations for the decision trees and decided to use the Weka explorer for implementation of J48 decision tree. The J48 tree is an implementation of the c4.5 decision tree. The tree accepts input in Attribute-Relation File Format (ARFF) format. ARFF files have two distinct sections. The first section is the header information, which is followed the data information. The Header of the ARFF file contains the name of the relation, a list of the attributes (the columns in the data), and their types.

@relation <relation-name>
@attribute <attribute-name> <datatype>
@attribute <classifier> {class1, class2,..}
@data

Each data instance is represented on a single line, with a carriage return denoting the end of the instance. Attribute values for each instance are delimited by commas. The order that was declared in the header section should be maintained (i.e. the data corresponding to the nth @attribute declaration is always the nth field of the attribute). Missing values are represented by a single question mark. The training dataset was converted to ARFF format. Based on the training dataset, a decision tree was formed. This decision tree is a type of ontology.

@relation spamchar

@attribute word_freq_make: real
@attribute word_freq_address: real
@attribute word_freq_all: real
@attribute word_freq_3d: real
@attribute word_freq_our: real
@attribute word_freq_over: real
@attribute word_freq_remove: real
@attribute word_freq_internet: real
@attribute word_freq_order: real
@attribute word_freq_mail: real

@attribute ifspam {1,0}

@data
0,0.64,0.64,0,0.32,0,0,0,0,0,0
0,0.67,0.23,0,0.17,0.6,1.6,0,1,0.9,1

The above file is a sample ARFF file where the word next to @relation is the just a name. It could be the name of the file, and name. It just signifies a header. The word next to the @attribute is the feature element on the basis of which the classification is going be done and our tree is being built. The value next to it after the ':' is its type. The last attribute in this list must be the final classifier of what we are looking for. In this case, the final classification result should be '1' if it is finally spam, otherwise, it should be '0' if it is not spam. All the leaf nodes on the classification result should be '1' or '0'. This is a rule in the ARFF file that the last attribute be the final classification result needed. After the @data, a set of values which are values of the attributes will be placed. The number of values will equal the number of attributes and the order is such that the first value in the dataset corresponds to the first attribute. i.e., here:

For the First mail:
word_freq_make is 0 and word_freq_all is 0.64
Similarly, for the Second mail:
word_freq_make is 0 and word_freq_all is 0.23

These values are calculated as follows:
100*Number of words or characters in the attribute / total number of words in the email

If you notice, in both the datasets, the last values are either 0 or 1 which means that this mail is should be classified as spam if 1 or not spam if 0.

3.2. Architecture and Implementation

Figure 1 shows our framework to filter spam. The training dataset is the set of email that gives us a classification result. The test data is actually the email will run through our system which we test to see if classified correctly as spam or not. This will be an ongoing test process and so, the test data is not finite because of the learning procedure, the test data will sometimes merge with the training data. The training dataset was used as input to J48 classification. To do that, the training dataset should be modified as a compatible input format. After J48 classification procedure, classification result was created.

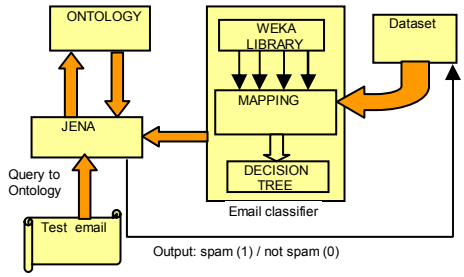

Figure 1. Filtering Architecture

To query the test email in Jena, an ontology should be created based on the classification result. To create ontology, an ontology language was required. RDF was used to create an ontology. The classification result in the form of RDF file format was inputted to Jena, and inputted RDF was deployed through Jena, finally, an ontology was created. Ontology generated in the form of RDF data model is the base on which the incoming mail is checked for its legitimacy. Depending upon the assertions that we can conclude from the outputs of Jena, the email can be defined as spam or otherwise. The email is actually the email in the format that Jena will take in (i.e. in a CSV format) and will run through the ontology that will result in spam or not spam.

The input to the system mainly is the training dataset and then the test email. The test email is the first set of emails that the system will classify and learn and after a certain time, the system will take a variety of emails as input to be filtered as a spam or not. The training dataset which we used, which had classification values for features on the basis of which the decision tree will classify, will first be given to get the same. The classification results need to be converted to an ontology. The decision result which we obtained J48 classification was mapped into RDF file. This was given as an input to Jena which then mapped the ontology for us. This ontology enabled us to decide the way different headers and the data inside the email are linked based upon the word frequencies of each words or characters in the dataset. The mapping also enabled us to obtain assertions about the legitimacy and non-legitimacy of the emails. The next part was using this ontology to decide whether a new email is a spam or not. This required querying of the obtained ontology which was again done through Jena. The output obtained after querying was the decision that the new email is a spam or not.

The primary way where user can let the system know would be through a GUI or a command line input with a simple 'yes' or 'no'. This would all be a part of a full fledged working system as opposed to our prototype which is a basic research model.

```
word_freq_remove: > 0
|   word_freq_hp: <= 0.19
|   |   word_freq_edu: <= 0.08
|   |   |   word_freq_1999: <= 0.25: 1 (716.0/17.0)
|   |   |   word_freq_1999: > 0.25
|   |   |   |   word_freq_george: <= 0.08: 1 (31.0)
|   |   |   |   word_freq_george: > 0.08: 0 (3.0)
|   |   word_freq_edu: > 0.08
|   |   |   word_freq_000: <= 0.1: 0 (7.0/1.0)
|   |   |   word_freq_000: > 0.1: 1 (20.0)
|   word_freq_hp: > 0.19
|   |   word_freq_our: <= 0.3: 0 (16.0/1.0)
|   |   word_freq_our: > 0.3
|   |   |   capital_run_length_average: <= 2.689: 0 (3.0/1.0)
|   |   |   capital_run_length_average: > 2.689: 1 (11.0)
```

Figure 2. Part of J48 classification result

Figure 2 shows how we choose the J48 classification filter, which uses the simple c4.5 decision tree for classification. Figure 2 shows that word "remove" was selected as a root node by J48 classification.

```
=== Summary ===

Correctly Classified Instances        4471       97.1745 %
Incorrectly Classified Instances       130        2.8255 %
Kappa statistic                          0.9406
Mean absolute error                      0.0522
Root mean squared error                  0.1615
Relative absolute error                  9.9284 %
Root relative squared error             33.0585 %
Total Number of Instances             4601
```

Figure 3. Summary of classification result

Figure 3 shows the classification result including precision, recall. The confusion matrix which shows the number of elements classified correctly and incorrectly as the percentage of classification.

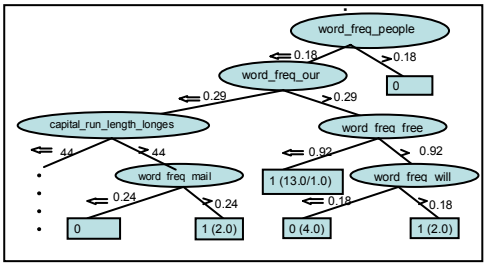

Figure 4. Classification result using J48

Figure 4 shows the classification result using J48. Whole result is so big, so figure 4 is just a part of it. According to the figure 5, if the normalized value of word "people" is greater than 0.18, email is classified as legitimate, otherwise, the system will check the normalized value of word "our". Finally, if the normalized value of word "mail" is greater than 0.24, then the email is classified as spam. Ontology using RDF was created based on the classification result.

```
<?xml version="1.0"?><rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:cd="http://www.spamfilter.fake/spam#">
<rdf:Description
rdf:about="http://www.spamfilter.fake/spam/word_freq_remove">
<rdfs:subClassOf rdf:resource="word_freq_remove"/>
<cd:freqlseq_0>char_freq_$</cd:freqlseq_0>
<cd:freqgr_0>word_freq_hp</cd:freqgr_0>
</rdf:Description>
<rdf:Description
rdf:about="http://www.spamfilter.fake/spam/char_freq_$">
<rdfs:subClassOf rdf:resource="word_freq_remove"/>
<cd:freqlseq_0.055>word_freq_000</cd:freqlseq_0.055>
<cd:freqgr_0.055>word_freq_hp</cd:freqgr_0.055>
</rdf:Description>
<rdf:Description
rdf:about="http://www.spamfilter.fake/spam/word_freq_000">
<rdfs:subClassOf rdf:resource="char_freq_$"/>
<cd:freqlseq_0.25>char_freq_!</cd:freqlseq_0.25>
<cd:freqgr_0.25>word_freq_re</cd:freqgr_0.25>
</rdf:Description>
```

Figure 5. RDF file of J48 classification result

Figure 5 shows the RDF file created based on J48 classification result. The RDF file was used as an input to Jena to create an ontology which will be used to check if the test email is spam or not.

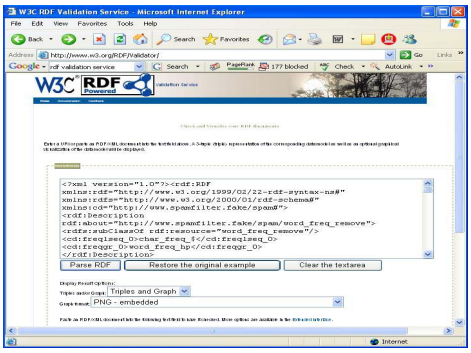

Figure 6. W3C RDF Validation Services

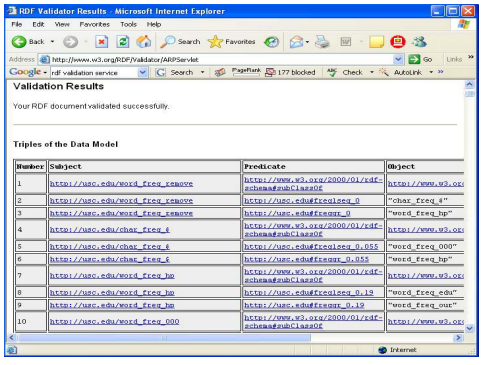

Figure 7. Triplets of RDF data model

Figure 6 shows RDF validation services. W3C RDF validation services help us to check whether the RDF schema which we are going to give as input to Jena is syntactically correct or not.

Because the RDF file based on the classification result using J48 was created by us, and should be compatible with Jena, the validation procedure for syntax validation was required. Figure 7 also shows the database of Subject-Predicate-Object model we got after inputting the RDF file into Jena. This ontology model is also produced in Jena.

Figure 8 shows the RDF data model or ontology model. This model is obtained from the W3C validation schema. This ontology is obtained in Jena in memory and not displayed directly. But it can be showed using the graphics property of the Jena.

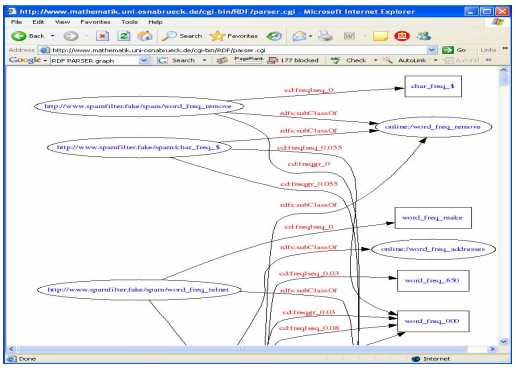

Figure 8. RDF data model (Ontology)

4. Results

About 4600 emails were used as an initial dataset. 39.4% of dataset were spam and 60.6% were legitimate email. J48 was used to classify the dataset in Weka explorer. 97.17% of emails were classified correctly and 2.73% were classified incorrectly. In the case of spam, precision was 0.976, recall was 0.952, and F-Measure was 0.964. In the case of legitimate, precision was 0.969, recall was 0.985, and F-measure was 0.977. Like the above, based on J48 classification result, ontology was created in RDF format using Jena. The ontology created using the RDF file was used to check input email through Jena.

Class	TP rate	FP rate	Precision	Recall	F-measure
spam	0.952	0.015	0.976	0.952	0.964
legitimate	0.985	0.048	0.969	0.985	0.977

Table 1. Classification result of training dataset

The result was generated after we consider the word frequencies of various words inside the email and then querying our ontology data model for these word frequencies. If the value we get after comparing all the word frequencies of the email words is '0' then the result was that the email was not spam and if the value is '1' then the result is that the email is spam. The result may have False Positives (A legitimate mail termed as not spam) or False Negatives (spam email termed as not spam). This case, in future, can be handled by updating the decision tree and hence the ontology model in Jena based upon the decision tree. The updated ontology will then be queried next time we check for the legitimacy of a new email. The experiment we conducted initially consisted of 100 emails that we fed in and got 94 correctly classified. This is 94% accuracy. Then we increased the number of email to a 150 and got 143 classified. This increased the accuracy to 95.3%. Finally, we fed in 200 emails and got 192 classified correctly which is a good 96% accuracy. By creating an ontology as a modularized filter, the ontology could be used in most of Semantic Web, or to correlate with other Semantic applications. This ontology also could be increased adaptively, so it is scalable.

5. Conclusion and Future Work

Our experiment here is still at an inception phase where the model is still learning. The accuracy of the decision tree was approximately 97.17% which was quite good at this stage. Our system gave an accuracy of 96%, so we can conclude not a large loss from the work which is an idea and an attempt at aiding ontology based classification and filtering. The important objective of the paper was to use an ontology to help classifying emails and it was successfully implemented. Learning motivation was that this approach has been taken and opens up a whole new aspect of email classification on the semantic web. Also, this approach fits into any system because they are so generic in nature. This idea will have great advantage on systems ahead. As mentioned above, the classification accuracy can be increased initially by pruning the tree and using better classification algorithms, more number and better classifiers or feature elements, etc. These are issues more in the machine learning and artificial intelligence domain which are not primary concerns but helped in better classification after all.

The work is still a research model and the accuracy can be improved later. Moreover, ontologies play a key role here as after that email gets classified through the ontology we created, and more work can be done in the area of creating intelligent ontologies and ontologies that can be used in certain areas of decision making, etc. The ontologies were created in Jena and this is just one aspect of ontology creation. There are other various and maybe better techniques that would have created ontologies without Jena or in some format that is more flexible and open to intelligence. This paper, as mentioned earlier, is more research-oriented and involved testing particular interfacing and checking for feasibility of classification of email through ontologies. The challenge we faced was mainly to make J48 classification outputs to RDF and gave it to Jena, i.e. interfacing two independent systems and creating a prototype that actually uses this

information that flows from one system to another to get certain desired input. In our case, it was classification of email. The only aspect of this work that is evolutionary and can be worked upon in the future is the fact that the email we use is in a particular Comma Separated Values (CSV) format. This is a requirement for Jena. Therefore, future work can be to create a system that takes a normal email (i.e. in HTML parsed text format) or text format itself to be given to the ontology – which again could be created using alternate methods. To obtain better result, we need to classify the training dataset using Neural Network, Naïve Bayesian Classifier, SVM, etc. Also, if the ontology increases adaptively, then the rate of correctly classified data will be increased.

6. Acknowledgement

This research has been funded in part by the Integrated Media Systems Center, a National Science Foundation Engineering Research Center, Cooperative Agreement No. EEC-9529152.

References

[1] 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", *CoRR cs.CL/0009009*, 2000.

[2] K. Choi, C. Lee, and P. Rhee, "Document Ontology Based personalized Filtering System", *Proceedings of the ACM Multimedia*, 2000.

[3] W. Cohen, "Learning rules that classify e-mail", *Proceedings of the AAAI Spring Symposium on Machine Learning in Information Access*, 1996.

[4] B. Cui, A. Mondal, J. Shen, G. Cong, and K. Tan, "On Effective E-mail Classification via Neural Networks", *Proceedings of the 16th International Conference on Database and Expert Systems Applications (DEXA05),* 2005, pp. 85-94.

[5] E. Crawford, I. Koprinska, and J. Patrick, "Phrases and Feature Selection in E-Mail Classification", *Proceedings of the 9th Australasian Document Computing Symposium (ADCS04*), 2004, pp. 59-62.

[6] Y. Diao, H. Lu, and D. Wu, "A comparative study of classification based personal e-mail filtering", *Proceedings of the 4th Pacific-Asia Conference of Knowledge Discovery and Data Mining (PAKDD00*), 2000.

[7] T. Fawcett, "in vivo spam filtering: A challenge problem for data mining", *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining Explorations. vol.5 no.2 (KDD03)*, 2003.

[8] Ferris Research, "Spam Control: Problems & Opportunities", 2003.

[9] K. Gee, "Using latent semantic indexing to filter spam", *Proceedings of the 18th ACM Symposium on Applied Computing, Data Mining Track (SAC03*), 2003.

[10] A. Hotho, S. Staab, and G. Stumme, "Ontologies Improve Text Document Clustering", *Proceedings of 3rd IEEE International Conference on Data Mining (ICDM03)*, 2003, pp. 541-544.

[11] International Data Group, "Worldwide email usage 2002 - 2006: Know what's coming your way", 2002.

[12] S. Kiritchenko, and S. Matwin, "Email classification with co-training", *Proceedings of workshop of the Center for Advanced Studies on Collaborative Research (CASCON01*), 2001.

[13] S. Kiritchenko, S. Matwin, and S. Abu-Hakima, "Email Classification with Temporal Features", *Proceedings of the International Intelligent Information Systems (IIS04),* 2004, pp. 523-533.

[14] Machine Learning Lab. in Information and Computer Science, University of California at Irvine, http://www.ics.uci.edu/~mlearn/MLSummary.html

[15] T. Meyer, and B. Whateley, "SpamBayes: Effective open-source, Bayesian based, email classification system" *Proceedings of the 1st Conference of Email and Anti-Spam (CEAS04*), 2004.

[16] J. Robert von Behren, S. Czerwinski, A. Joseph, E. Brewer, and J. Kubiatowicz, "NinjaMail: The Design of a High-Performance Clustered, Distributed E-Mail System", *Proceedings of Workshop of the 29th International Conference on Parallel Processing (ICPP00),* 2000.

[17] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A Bayesian Approach to Filtering Junk E-Mail", *Proceedings of the AAAI Workshop on Learning for Text Categorization*, 1998.

[18] K. Taghva, J. Borsack, J. Coombs, A. Condit, S. Lumos, and T. Nartker, "Ontology-based Classification of Email, *symposium of ITCC*, 2003, pp. 194-198.

[19] S. Youn, and D. McLeod, "Ontology Development Tools for Ontology-Based Knowledge Management", *Encyclopedia of E-Commerce, E-Government and Mobile Commerce*, *Idea Group Inc.,* 2006.