

Spam Email Classification using an Adaptive Ontology

Seongwook Youn, Dennis McLeod

Department of Computer Science, University of Southern California, Los Angeles, CA. USA

Email: {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. Similar to other filters, the ontology evolves with the user requests. Hence the ontology would be customized for the user. This paper proposes to find an efficient spam email filtering method using adaptive ontology

Index Terms—spam filter, ontology, data mining, text classification, feature extraction

I. 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 the 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 [24], 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 [35] 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 [67]. 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,

This paper is based on "Efficient Spam Email Filtering using an Adaptive Ontology," by Seongwook Youn and Dennis McLeod which appeared in the Proceedings of 4th International Conference on Information Technology: New Generations (ITNG), Las Vegas, NV, April 2007. © 2007 IEEE.

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.

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 explains background and related works; Section 3 describes text mining (text classification); Section 4 introduces our idea of spam filtering using an ontology; Section 5 discusses the experimental result of the proposed framework; Section 6 concludes the paper with possible directions for future work.

II. BACKGROUND AND RELATED WORK

A. *Understanding of an Ontology*

An ontology is an explicit specification of a conceptualization. Ontologies can be taxonomic hierarchies of classes, class definitions, or subsumption relation, but need not be limited to these forms. Also, ontologies are not limited to conservative definitions. To specify a conceptualization one needs to state axioms that constrain the possible interpretations for the defined terms [29]. Ontologies play a key role in capturing domain knowledge and providing a common understanding.

Generally, ontologies consist of taxonomy, class hierarchy, domain knowledge base, and relationships between classes and instances. An ontology has different relationships depending on the schema or taxonomy builder, and it has different restrictions depending on the language used. Also, the domain, range, and cardinality are different based on ontology builder. Ontologies allow for machine-understandable semantics of data, and facilitate the search, exchange, and integration of knowledge for business-to-business (B2B) and business-to-consumer (B2C) e-commerce. By using semantic data, the usability of e-technology can be facilitated. There are several languages like extensible markup language(XML), resource description framework (RDF), RDF schema (RDFS), DAML+OIL, and OWL. Many tools have been developed for implementing metadata of ontologies using these languages. However, current tools have problems with interoperation and collaboration.

B. *Ontology Development*

Ontology tools can be applied to all stages of the ontology lifecycle including the creation, population, implementation and maintenance of ontologies [?]. An ontology can be used to support various types of knowledge management including knowledge retrieval, storage and sharing [56]. In one of the most popular definitions, an ontology is "the specification of shared knowledge" [64]. For a knowledge management system, an ontology can be regarded as the classification of knowledge. Ontologies are different from traditional keyword-based search engines in that they are metadata, able to provide the search engine with the functionality of semantic match. Ontologies are able to search more efficiently than traditional methods. Typically, an ontology consists of hierarchical description of important concepts in a domain and the descriptions of

the properties of each concept. Traditionally, ontologies are built by both highly trained knowledge engineers and domain specialists who may not be familiar with computer software. Ontology construction is a time-consuming and laborious task. Ontology tools also require users to be trained in knowledge representation and predicate logic. XML is not suited to describe machine understandable documents and interrelationships of resources in an ontology [30]. Therefore, The W3C has recommended the use of the resource description framework (RDF), RDF schema (RDFS), DAML+OIL and OWL. Since then, many tools have been developed for implementing metadata of ontologies by using RDF, RDFS, DAML+OIL and OWL.

Ontology tools have to support more expressive power and scalability with a large knowledge base, and reasoning in querying and matching. Also, they need to support the use of high-level language, modularity, visualization, etc. There are also researches and applications about dynamic web pages consisting of database reports. The research on ontology integration tasks in B2B E-Commerce is also undergoing. The infrastructure of the business documentation from the integration perspective and the identification of the integration subtasks was suggested [47]. There is research on generic e-Business model ontology for the development of tools for e-business management and IS Requirements Engineering. Based on an extensive literature review, the e-Business Model Ontology describes the logic for a business system [49].

C. *Spam Filtering*

[61] and [66] developed a algorithm to reduce the feature space without sacrificing remarkable classification accuracy, but the effectiveness was based on the quality of the training dataset. [65] demonstrated that the feasibility of the approach to find the best learning algorithm and the metadata to be used, which is a very significant contribution in email classification using Rainbow system. [5] proposed a graph based mining approach for email classification that structures/patterns can be extracted from a pre-classified email folder and the same can be used effectively for classifying incoming email messages.

Approaches to filtering junk email are considered [14] [17] [60]. [22] and [34] showed approaches to filtering emails involve the deployment of data mining techniques. [15] 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. [6] compared the performance of the Naive Bayesian filter to an alternative memory based learning approach on spam filtering. [44] addressed the problem by proposing a Word Sense Disambiguation (WSD) approach based on the intuition that word proximity in the document implies proximity also in the Hierarchical Thesauri (HT) graph. Bringing in other kinds of features, which are spam-specific features in their work, could improve the classification results [60].

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 [38]. [45] showed that the work on spam filtering using feature selection based on heuristics. [40] presented a technique to help various classifiers to improve the mining of category profiles. Upon receiving a document, the technique helps to create dynamic category profiles with respect to the document, and accordingly helps to make proper filtering and classification decisions. [65] [66] compared a cross-experiment between 14 classification methods, including decision tree, Naive Bayesian, Neural Network, linear squares fit, Rocchio. KNN is one of top performers, and it performs well in scaling up to very large and noisy classification problems.

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 Naive Bayesian, Neural Network, or Support Vector Machine (SVM) classifier.

III. EMAIL CLASSIFICATION (TEXT MINING)

A. Text Mining

Text mining is from data mining. Data mining is defined as the "the non trivial process of identifying valid, novel, potentially useful, previously unknown and ultimately understandable patterns in large databases" [23]. Most of data mining work was based solely on database or structured data. Adibi motivated data mining as "We are drowning in Data but Starving for Knowledge" [1] in his paper.

According to [31] text mining is the discovery by computer of new, previously unknown information, by automatically extracting information from different written resources. A key element is the linking together of the extracted information together to form new facts or new hypotheses to be explored further by more conventional means of experimentation." Berland and Charniak [8] used techniques similar to Hearst [32]. Mann [43] suggested the use of lexico-POS based patterns for constructing an ontology of *is-a* relations for proper nouns. He used the manually crafted pattern CN PN. He reported generating 200,000 unique descriptions of proper nouns with 60% accuracy. Moldovan and Girju [46] gave us a good overview of the various text mining techniques.

Pasca [52] suggested a technique for extracting glosses present for nodes present in WordNet from a corpus of 500 million webpages. He also proposed a clustering technique to group together nuggets that have a very high overlap. According to his paper, 60% of the WordNet nodes had at least one gloss extracted from the web corpus. In [53], Pasca used a pattern-based technique to extract *is-a* relations from a corpus of 500 webpages. Etzioni et al.

[18] [19] [20] extracts instance-concept relations from a huge web corpus. To perform the task, a combination of pattern learning, subclass extraction and list extraction was used. Ciaramita and Johnson [13] use a fixed set of 26 semantic labels to perform *is-a* supersense tagging.

Caraballo [10] used a clustering technique to extract hyponym relations from newspaper corpus. Similar method was also used by Pantel and Ravichandran [51]. They used Clustering by Committee (CBC) algorithm [50] to extract clusters of nouns belonging to the same class. Cederberg and Widdows [11] use Latent Semantic Analysis and noun co-ordination (co-occurrence) technique to extract Hyponyms from a newspaper corpus. They reported 58% accuracy using their approach. Snow et al. [62] exploited both pattern-based and rich co-occurrence features to extract *is-a* relations from text, but the technique was not web-scalable.

B. Feature Selection

Given the rampant amount of textual data these days, it is becoming increasingly important to be able to extract domain-specific semantic content from some texts. Such semantic knowledge, in the form of ontologies, can facilitate integration of information from various sources. Additionally, ontologies enable the visualization and maintenance of knowledge. However, in most cases, ontology building is still conducted by hand. It is time-consuming, error-prone, and labor-intensive. Moreover, manual ontology building has a critical weakness, in that the ontology usually reflects the inherent knowledge and biases of its creator, which may not be shared across people. If the ontology were created (semi-)automatically, then such problems will be significantly reduced. Therefore it would be very desirable to have a (semi or fully) automatic method for acquiring a domain ontology.

One of the early attempts at ontology learning was by Faure and Nedellec [21], who proposed applying two techniques from the field of Natural Language Processing (NLP), namely verb-subcategorization and noun-clustering for ontology learning. Kietz *et al.* [36] developed a method for semi-automatic ontology acquisition for a corporate intranet (e.g., insurance company). To organize a concept hierarchy for the target ontology, a number of heuristics were used. While constructing an ontology, a human domain expert was expected to be on hand to intervene in this process by comparing the resulting ontology with a reference ontology.

Navigli et al. [48] made use of techniques from Information Retrieval and Machine Learning to resolve ambiguity in the meaning of words and their semantic relationships, which is crucial to building a domain ontology. The performance of their method was evaluated with respect to a number of web pages on travel. Other techniques from Machine Learning and Information Retrieval for building ontologies have been outlined in [42]. However, the majority of this work has tried to learn ontologies for relatively constrained domains. To date, there has been relatively little work on trying to construct ontologies

for an open domain. Furthermore, *tfidf* is typically used to determine words for the domain ontology concepts. Since *tfidf* purely reflects the frequency-based importance of words, it cannot capture dependencies, such as those between a concept in the domain and the words that correspond to that concept. Text learning techniques, such as statistical feature selection methods, have proven to be useful in extracting more informative words from a given text for a given text learning task. However, there have been few studies that empirically examine the value of text learning techniques to extract a set of candidate words for concept words in an ontology for ontology learning.

To use of existing feature selection methods for the extraction of a set of good-candidate words for concept words in an ontology, we use a number of existing feature selection methods to identify sets of candidate concept words. These sets are then evaluated with respect to manually created domain ontologies [7]. Feature selection generally refers to the way of selecting a set of features which is more informative in executing a given machine learning task while removing irrelevant or redundant features. This process ultimately leads to the reduction of dimensionality of the original feature space, but the selected feature set should contain sufficient or more reliable information about the original data set. For the text domain, this will be formulated into the problem of identifying the most informative word features within a set of documents for a given text learning task. Feature selection methods have relied heavily on the analysis of the characteristics of a given data set through statistical or information-theoretical measures. For text learning tasks, for example, they primarily count on the vocabulary-specific characteristics of given textual data set to identify good word features. Although the statistics itself does not care about the meaning of text, these methods have been proved to be useful for text learning tasks (e.g., classification and clustering). In our study, *tfidf* was considered as a feature selection mechanism.

C. Email Classification (Text Mining)

Generally, the main tool for email management is text classification. A classifier is a system that classifies texts into the discrete sets of predefined categories. For the email classification, incoming messages will be classified as spam or legitimate using classification methods.

1) *Neural Network (NN)*: Classification method using a NN was used for email filtering long time ago. Generally, the classification procedure using the NN consists of three steps, data preprocessing, data training, and testing. The data preprocessing refers to the feature selection. Feature selection is the way of selecting a set of features which is more informative in the task while removing irrelevant or redundant features. For the text domain, feature selection process will be formulated into the problem of identifying the most relevant word features within a set of text documents for a given text learning task. For the data training, the selected features from the data preprocessing step were fed into the NN, and

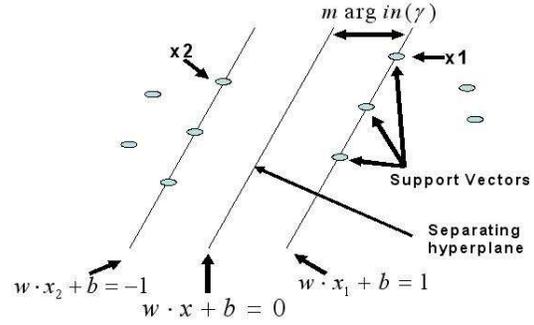


Figure 1. Support Vector Machine (SVM)

an email classifier was generated through the NN. For the testing, the email classifier was used to verify the efficiency of NN. In the experiment, an error BP (Back Propagation) algorithm was used.

2) *Support Vector Machine (SVM)*: SVMs are a relatively new learning process influenced highly by advances in statistical learning theory. SVMs have led to a growing number of applications in image classification and handwriting recognition. Before the discovery of SVMs, machines were not very successful in learning and generalization tasks, with many problems being impossible to solve. SVMs are very effective in a wide range of bioinformatic problems. SVMs learn by example. Each example consists of a m number of data points (x_1, \dots, x_m) followed by a label, which in the two class classification we will consider later, will be +1 or -1. -1 representing one state and 1 representing another. The two classes are then separated by an optimum hyperplane, illustrated in figure 1, minimizing the distance between the closest +1 and -1 points, which are known as support vectors. The right hand side of the separating hyperplane represents the +1 class and the left hand side represents the -1 class. This classification divides two separate classes, which are generated from training examples. The overall aim is to generalize well to test data. This is obtained by introducing a separating hyperplane, which must maximize the margin (γ) between the two classes, this is known as the optimum separating hyperplane

Lets consider the above classification task with data points $x_i, i=1, \dots, m$, with corresponding labels $y_i = \pm 1$, with the following decision function:

$$f(x) = \text{sign}(w \cdot x + b)$$

By considering the support vectors x_1 and x_2 , defining a canonical hyperplane, maximizing the margin, adding Lagrange multipliers, which are maximized with respect to :

$$W(\alpha) = \sum_{i=1}^m \alpha_i - \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$

$$\left(\sum_{i=1}^m \alpha_i y_i = 0, \alpha_i \geq 0 \right)$$

3) *Naive Bayesian Classifier (NB)* : Naive Bayesian classifier is based on Bayes' theorem and the theorem of total probability. The probability that a document d with vector $\vec{x} = \langle x_1, \dots, x_n \rangle$ belongs to category c is

$$P(C = c | \vec{X} = \vec{x}) = \frac{P(C=c) \cdot P(\vec{X}=\vec{x}|C=c)}{\prod_{k \in \{spam, legit\}} P(C=k) \cdot P(\vec{X}=\vec{x}|C=k)}$$

However, the possible values of \vec{X} are too many and there are also data sparseness problems. Hence, Naive Bayesian classifier assumes that X_1, \dots, X_n are conditionally independent given the category C . Therefore, in practice, the probability that a document d with vector $\vec{x} = \langle x_1, \dots, x_n \rangle$ belongs to category c is

$$P(C = c | \vec{X} = \vec{x}) = \frac{P(C=c) \cdot \prod_{i=1}^n P(X_i=x_i|C=c)}{\prod_{k \in \{spam, legit\}} P(C=k) \cdot \prod_{i=1}^n P(X_i=x_i|C=k)}$$

$P(X_i|C)$ and $P(C)$ are easy to obtain from the frequencies of the training datasets. So far, a lot of researches showed that the Naive Bayesian classifier is surprisingly effective.

4) *J48 Classifier (J48)* : J48 classifier is a simple C4.5 decision tree for classification. It creates a binary tree.

D. Result Evaluation

In this section, four classification methods (Neural Network, Support Vector Machine classifier, Naive Bayesian classifier, and J48 classifier) were evaluated the effects based on different datasets and different features. Finally, the best classification method was obtained from the training datasets. 4500 emails were used as a training datasets. 38.1% of datasets were spam and 61.9% were legitimate email. To evaluate the classifiers on training datasets, we defined an accuracy measure as follows.

$$Accuracy (\%) = \frac{CorrectlyClassifiedEmails}{TotalEmails} \cdot 100$$

Also, Precision and Recall were used as the metrics for evaluating the performance of each email classification approach.

$$Recall = \frac{N_{ii}}{N}, Precision = \frac{N_{ii}}{N_i}$$

$$N = \#OfTotalInterestingEmails$$

$$N_i = \#OfEmailsClassifiedAsInteresting$$

$$N_{ii} = \#OfInterestingEmailsClassifiedAsInteresting$$

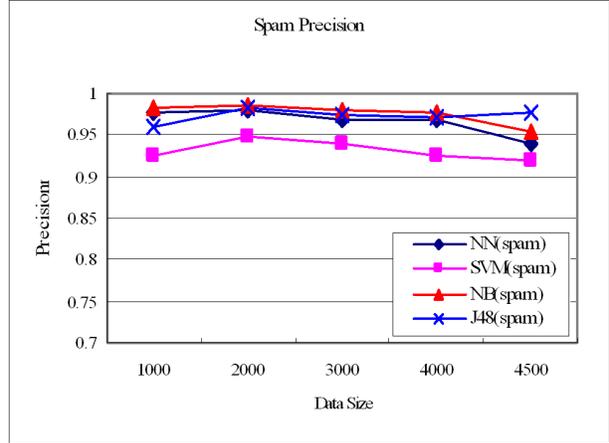


Figure 2. Spam precision based on data size

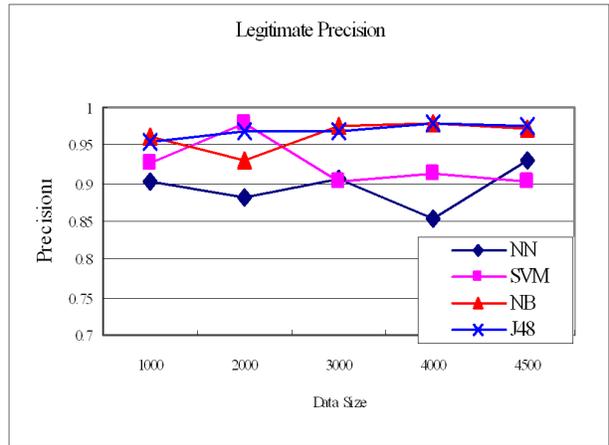


Figure 3. Legitimate precision based on data size

1) *Effect of datasets on performance* : An experiment measuring the performance against the size of datasets was conducted using datasets of different sizes listed in Table I. The experiment was performed with 55 features from *tfidf*. For example, in case of 1000 datasets, Accuracy was 95.80% using J48 classifier. A few observations can be made from this experiment. As shown on Table I, the average of correct classification rate for both J48 and NB was over 95%. Size of datasets was not an important factor in measuring precision and recall. The results show that the performance of classification was not stable.

For four different classification methods, precision of spam mail was shown in Figure 2, likewise, precision of legitimate mail was shown in Figure 3. As shown on Figure 2, 3, 4, and 5, the precision and recall curves of J48 and NB classification were better than the ones of NN and SVM. Also, the average precision and recall for both J48 and NB was over 95%.

In Figure 5, legitimate recall values were sharply decreased at the data size 2000. The increase of spam mail in the training datasets between 1000 and 2000 result in a sharp decrease of legitimate recall values for all classifiers.

TABLE I.
CLASSIFICATION RESULTS BASED ON DATA SIZE. (WITH 55 FEATURES)

Data Size	NN	SVM	Naive Bayesian	J48
1000	93.50%	92.70%	97.20%	95.80%
2000	97.15%	95.00%	98.15%	98.25%
3000	94.17%	92.40%	97.83%	97.27%
4000	89.60%	91.93%	97.75%	97.63%
4500	93.40%	90.87%	96.47%	97.56%

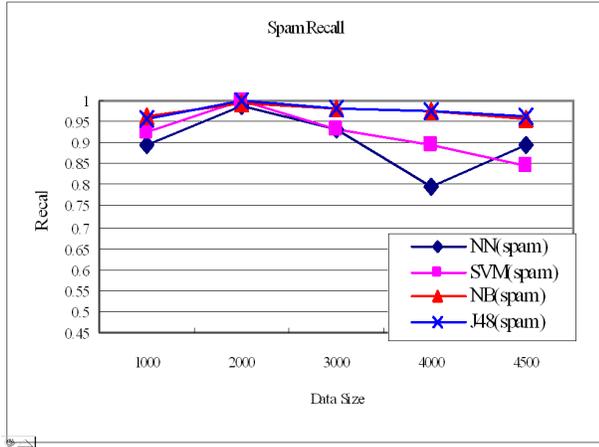


Figure 4. Spam recall based on data size

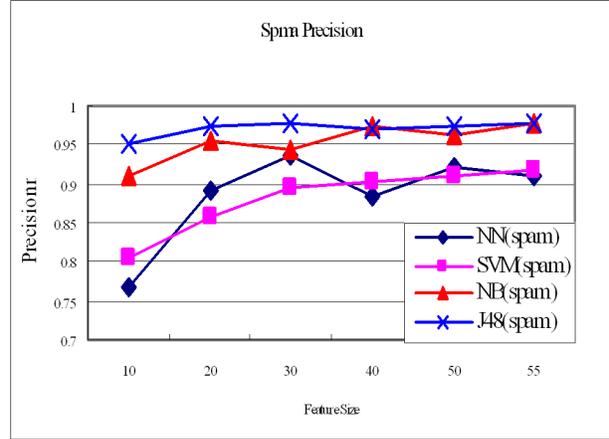


Figure 6. Spam precision based on feature size

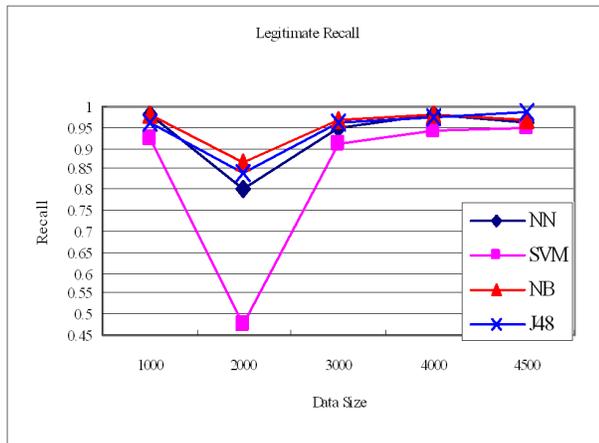


Figure 5. Legitimate recall based on data size

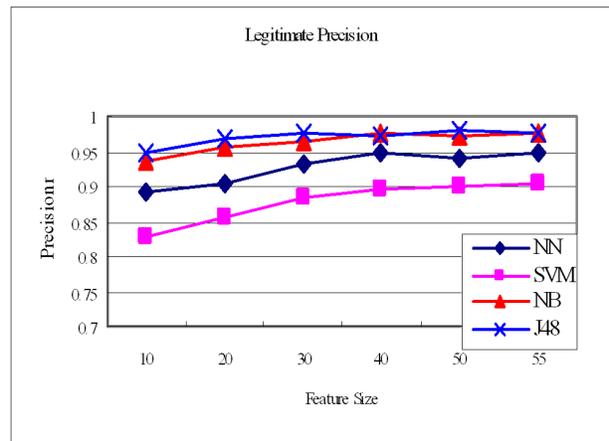


Figure 7. Legitimate precision based on feature size

2) *Effect of feature size on performance* : The other experiment measuring the performance against the size of datasets was conducted using different features listed in Table II. 4500 email datasets was used for the experiment. For example, in case of 10 features, Accuracy was 94.84% using J48 classifier. The most frequent words in spam mail were selected as features. Generally, the result of classification was increased for all classification methods according the feature size increased.

As shown in Figure 6, 7, 8, and 9, good classification result order in the experiment was J48, NB, NN, and SVM for all cases (spam precision, legitimate precision, spam recall, and legitimate recall). The overall precision and recall for email classification increase and become

stable according to the increase of the number of feature. Gradually, the Accuracy increase and finally saturated with the increased feature size. As shown in Figure 6 and 7, J48 classifier provided the precision over 95% for every feature size irrespective of spam or legitimate. Also, J48 classifier supported over 97% of classification accuracy for more than 30 feature size. For the recall, J48 and NB showed better result than NN and SVM for both spam and legitimate mail, but J48 was a little bit better than NB.

TABLE II.
CLASSIFICATION RESULTS BASED ON FEATURE SIZE.

Feature Size	NN	SVM	Naive Bayesian	J48
10	83.60%	81.91%	92.42%	94.84%
20	89.87%	85.73%	95.60%	96.91%
30	93.31%	88.87%	95.64%	97.56%
40	92.13%	89.93%	97.49%	97.13%
50	93.18%	90.27%	96.84%	97.67%
55	93.10%	90.84%	97.64%	97.56%

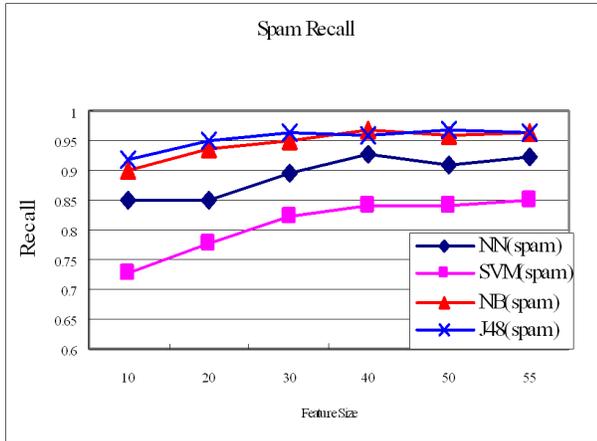


Figure 8. Spam recall based on feature size

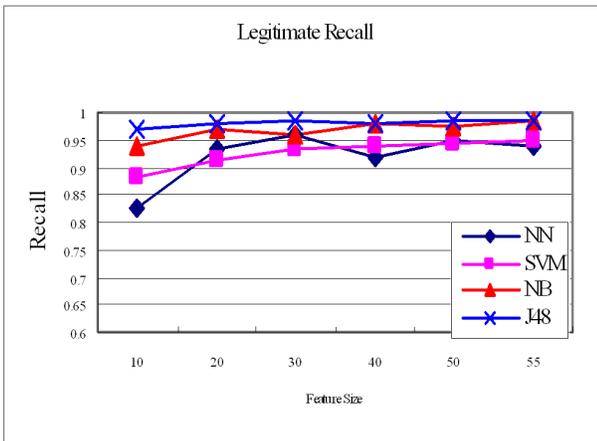


Figure 9. Legitimate recall based on feature size

IV. SPAM FILTERING USING AN ONTOLOGY

A. 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 was 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. 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.

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

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. 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: $100 * \text{Number of words or characters in the attribute} / \text{total number of words in the email}$

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.

B. Objective

The training datasets are the set of emails that gives us a classification result. The test data is actually the email that 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, and will sometimes merge with the training data. The training datasets were used as input to J48 classification. To do that, the training datasets should be modified as a compatible input format. To query the test email in Jena, an ontology should be created based on the classification result. To create ontology, an ontology language is required. Resource Description

Framework (RDF) was used to create an ontology. The classification result of RDF format was inputted to Jena, and inputted RDF was deployed through Jena, finally, an ontology was created. An 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 legitimate.

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 legitimate. The system updates periodically the datasets with the emails classified as spam when user spam report is requested. Then, increased training datasets are inputted to Weka to get a new classification result. Through this procedure, the number of ontology will be increased. Finally, this spam filtering ontology will be customized for each user.

Customized ontology filter would be different with each other depending on each user's background, preference, hobby, etc. That means one email might be spam for person A, but not for person B. The ontology evolves periodically and adaptively. The input to the system is mainly the training datasets 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 spam or legitimate. For the training datasets which we used, several feature selection algorithms including Naive Bayesian, Neural Network, SVM, and J48 were tested, then J48 and Naive Bayesian classifier showed the good performance on the training email datasets [68]. The classification results through Weka need to be converted to an ontology. The classification result which we obtained through J48 decision tree was mapped into RDF format. 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 datasets. 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 spam or legitimate. Queries using the obtained ontology were processed again through Jena. The output obtained after querying was the decision that the new email is spam or legitimate. [69].

C. Architecture and Implementation

Figure 10 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 To query the test

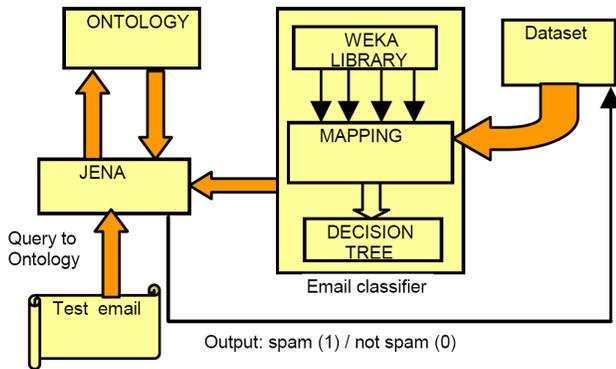


Figure 10. Filtering Architecture

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.

Figure 11 shows how we choose the J48 classification filter, which uses the simple c4.5 decision tree for classification. Figure 11 shows that word "remove" was selected

```

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 11. Part of J48 classification result

```

=== 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            0.9284 %
Root relative squared error        33.0585 %
Total Number of Instances         4601

```

Figure 12. Summary of classification result

as a root node by J48 classification.

Figure 12 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 13 shows the classification result using J48. Whole result is so big, so figure 13 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.

Figure 14 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 15 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 16 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 17 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.

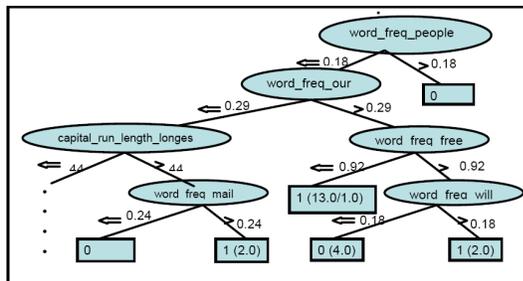


Figure 13. Visualized result of J48 classification

```
<?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:freqseq_0>char_freq_0</cd:freqseq_0>
<cd:freqqr_0>word_freq_hp</cd:freqqr_0>
</rdf:Description>
<rdf:Description
rdf:about="http://www.spamfilter.fake/spam/char_freq_0">
<rdfs:subClassOf rdf:resource="char_freq_0"/>
<cd:freqseq_0.055>word_freq_000</cd:freqseq_0.055>
<cd:freqqr_0.055>word_freq_hp</cd:freqqr_0.055>
</rdf:Description>
<rdf:Description
rdf:about="http://www.spamfilter.fake/spam/word_freq_000">
<rdfs:subClassOf rdf:resource="char_freq_0"/>
<cd:freqseq_0.25>char_freq_1</cd:freqseq_0.25>
<cd:freqqr_0.25>word_freq_re</cd:freqqr_0.25>
</rdf:Description>
```

Figure 14. RDF file based J48 classification result

Algorithm 1 is a pseudo code for ontology filter creation.

V. RESULTS AND DISCUSSION

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.

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 200 emails that we feed in and got 192 correctly classified. This is 96% accuracy. Then we increased the number

Figure 15. W3C RDF Validation Service

Number	Subject	Predicate	Object
1	http://usc.edu/spam/word_freq_hare_2	http://www.w3.org/2000/01/rdf-schema#rdf:type	http://www.w3.org/2000/01/rdf-schema#Class
2	http://usc.edu/spam/word_freq_hare_2	http://usc.edu/spam#freqseq_0.39625	"word_freq_remove_110"
3	http://usc.edu/spam/word_freq_hare_2	http://usc.edu/spam#freqqr_0.39625	"word_freq_remove_111"
4	http://usc.edu/spam/word_freq_you_16	http://www.w3.org/2000/01/rdf-schema#rdf:type	http://www.w3.org/2000/01/rdf-schema#Class
5	http://usc.edu/spam/word_freq_you_16	http://usc.edu/spam#freqseq_0.497612	"word_freq_remove_151"
6	http://usc.edu/spam/word_freq_you_16	http://usc.edu/spam#freqqr_0.497612	"word_freq_remove_1510"
7	http://usc.edu/spam/word_freq_ve_4	http://www.w3.org/2000/01/rdf-schema#rdf:type	http://www.w3.org/2000/01/rdf-schema#Class
8	http://usc.edu/spam/word_freq_ve_4	http://usc.edu/spam#freqseq_1.470889	"word_freq_remove_111"
9	http://usc.edu/spam/word_freq_ve_4	http://usc.edu/spam#freqqr_1.470889	"word_freq_remove_111"
10	http://usc.edu/spam/word_freq_ht_7	http://www.w3.org/2000/01/rdf-schema#rdf:type	http://www.w3.org/2000/01/rdf-schema#Class
11	http://usc.edu/spam/word_freq_ht_7	http://usc.edu/spam#freqseq_1.06393	"word_freq_remove_61"
12	http://usc.edu/spam/word_freq_ht_7	http://usc.edu/spam#freqqr_1.06393	"word_freq_remove_610"

Figure 16. Triplets of RDF data model

of email to a 400 and got 385 classified. This increased the accuracy to 96.25%. Finally, we feed in 600 emails and got 581 classified correctly which is a good 96.83% 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.

VI. CONCLUSION

A customized ontology filter was evolved based on specific user's background. Hence, as expected, better spam filtering rate can be achieved using a customized ontology filter which is adaptive and scalable. Text-oriented email datasets are adapted, but the same idea can be applicable to other classification or clustering tasks.

The important objective of the paper is 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 generic in nature. This idea will have great advantage on systems ahead. The classification accuracy can be improved initially by pruning the tree and using better classification algorithms, more number and better classifiers or feature elements, etc. These are bigger issues in the machine learning and artificial intelligence domain which are not primary

TABLE III.
CLASSIFICATION RESULT WITH TRAINING DATASETS

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

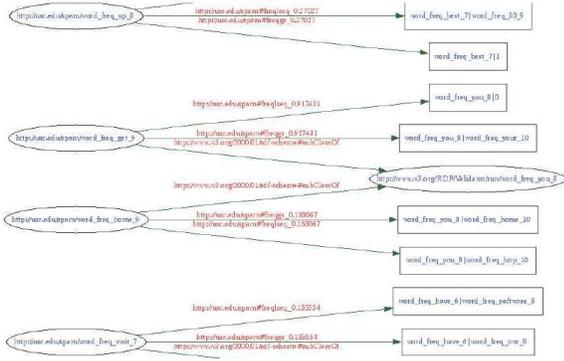


Figure 17. RDF data model

Algorithm 1 Ontology filter pseudo code

```

1: // Initialize variables
2: set training dataset  $d$  to  $d_1, \dots, d_n$ 
3: set test dataset  $t$  to  $t_1, \dots, t_p$ 
4: set normalized values  $v$  to  $v_1, \dots, v_m$ 
5:
6:  $Feature(f: f_1, \dots, f_m) \leftarrow tfidf(d)$ ;
7:
8: foreach( $f: f_1, \dots, f_m$ ) {
9:   foreach( $d: d_1, \dots, d_n$ ) {
10:    ( $n: n_1, \dots, n_m$ )  $\leftarrow Normalize(f, d)$ ;
11:   }
12: }
13: foreach( $n: n_1, \dots, n_m$ ) {
14: result  $\leftarrow J48(n, d)$ ;
15: }
16:
17:  $Ontology() \leftarrow Jena(RdfConversion(result))$ ;
18:
19: foreach( $t: t_1, \dots, t_p$ ) {
20:   if( $Ontology(t_i == 1)$ ) then
21:     decision = SPAM;
22:   else then
23:     decision = LEGITIMATE;
24: }

```

concerns but helped in better classification after all. 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 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 was mainly to make J48 classification outputs to RDF and fed it into 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.

Experimental results in this paper are based on the default settings. Extensive experiments with different settings are applicable in Weka. Moreover, different algorithms which are not included in Weka can be tested. Also, experiments with various feature selection techniques should be compared. We implemented an adaptive ontology as spam filter based on classification result. Then, this ontology is evolved and customized based on user's report when a user requests spam report. By creating a spam filter in the form of ontology, a filter will be user-customized, scalable, and modularized, so it can be embedded to many other systems.

REFERENCES

- [1] Adibi, J., and Shen, W. Data Mining Techniques and Applications in Medicine Tutorial, 1999.
- [2] Agichtein, E., and Luis Gravano, L. Querying Text Databases for Efficient Information Extraction. *proceedings of the 19th IEEE International Conference on Data Engineering (ICDE03)*, Bangalore, India, 2003.
- [3] Albrecht, K., Burri, N., and Wattenhofer, R. Spamato - An Extendable Spam Filter System. *Proceedings of 4th Conference on Email and Anti-Spam (CEAS05)*, Mountain View, CA 2005
- [4] Allemang, D., Polikoff, I., and Hodgson, R. Enterprise Architecture Reference Modeling in OWL/RDF. *Proceedings of 4th International Semantic Web Conference (ISWC05)*, Galway, Ireland, 844-857, 2005.
- [5] Aery, M., and Chakravarthy, S. eMailSift: Email Classification Based on Structure and Content. *Proceedings of The 5th IEEE International Conference on Data Mining (ICDM05)*, Clearwater Beach, FL, 18-25. 2005
- [6] Androustopoulos, I., Paliouras, G., Karkaletsis, V., Sakkis, G., Spyropoulos, C., and Stamatopoulos, P. Learning to Filter Spam E-Mail: A Comparison of a Naive Bayesian and a Memory-Based Approach. *The Computing Research Repository (CoRR)*, cs.CL/0009009, 2000.
- [7] Ankolekar, A., Seo, Y., and Sycara, K. Investigating semantic knowledge for text learning. *Workshop on Semantic Web of 26th Annual International ACM SIGIR Conference*, Toronto, Canada, 2003.

- [8] Berland, M., and Charniak, E. Finding Parts in Very Large Corpora. *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL99)*, College Park, MD, 1999.
- [9] Bishr, Y., Pundt, H., and Ruther, C. Design of a Semantic Mapper Based on a Case Study from Transportation. *Proceedings of 2nd International Conference of Interoperating Geographic Information Systems (INTEROP99)*, Zurich, Switzerland, 203-215, 1999.
- [10] Caraballo, S. Automatic construction of a hypernym-labeled noun hierarchy from text. *Proceedings of the Association for Computational Linguistics (ACL99)*, College Park, MD, 20-26, 1999.
- [11] Cederberg, S., and Widdows, D. Using LSA and Noun Coordination Information to Improve the Precision and Recall of Automatic Hyponymy Extraction. *Proceedings of Conference on Computational Natural Language Learning (CoNLL-2003)*, Edmunton, Canada, 118, 2003.
- [12] Chhabra, S., Yerazunis, W., and Siefkes, C. Spam Filtering using a Markov Random Field Model with Variable Weighting Schemas. *Proceedings of 4th IEEE International Conference on Data Mining (ICDM04)*, Brighton, UK, 347-350, 2004
- [13] Ciaranita, M. and Johnson, M. Supersense Tagging of Unknown Nouns in WordNet. *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP 2003)*, Sapporo, Japan, 2003.
- [14] Cohen, W. Learning rules that classify e-mail. *Proceedings of the AAAI Spring Symposium on Machine Learning in Information Access*, Palo Alto, CA, 1996.
- [15] Cui, B., Mondal, A., Shen, J., Cong, G., and Tan, K. On Effective E-mail Classification via Neural Networks. *Proceedings of the 16th International Conference on Database and Expert Systems Applications (DEXA05)*, Copenhagen, Denmark, 85-94, 2005.
- [16] Crawford, E., Koprinska, I., and Patrick, J. Phrases and Feature Selection in E-Mail Classification. *Proceedings of the 9th Australasian Document Computing Symposium (ADCS04)*, Melbourne, Australia, 59-62, 2004.
- [17] Diao, Y., Lu, H., and Wu, D. A comparative study of classification based personal e-mail filtering. *Proceedings of the 4th Pacific-Asia Conference of Knowledge Discovery and Data Mining (PAKDD00)*, Kyoto, Japan, 2000.
- [18] Etzioni, O., Cafarella, M., Downey, D., Popescu, A., Shaked, T., Soderland, S., Weld, D., and Yates, A. Unsupervised Named-Entity Extraction from the Web: An Experimental Study. *Proceedings of International Conference on Artificial Intelligence (ICAI05)*, Las Vegas, NV, 2005.
- [19] Etzioni, O., Cafarella, M., Downey, D., Popescu, A., Shaked, T., Soderland, S., Weld, D., and Yates, A. Methods for Domain-Independent Information Extraction from the Web: An Experimental Comparison. *Proceedings of 9th National Conference on Artificial Intelligence (AAAI04)*, San Jose, CA, 2004.
- [20] Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A., Shaked, T., Soderland, S., Weld, D., and Yates, A. Web-scale Information Extraction in KnowItAll. *Proceedings of 13th International World Wide Web Conference (WWW04)*, New York, NY, 2004.
- [21] Faure, D., and Nedellec, C. A corpus-based conceptual clustering method for verb frames and ontology acquisition. *Proceedings of LREC Workshop on adapting lexical and corpus resources to sublanguages and applications*, Granada, Spain, 1998.
- [22] Fawcett, T. 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)*, Washington, DC, 2003.
- [23] Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R. *Advances in Knowledge Discovery and Data Mining*. M.I.T. Press, 1996.
- [24] Ferris Research. Spam Control: Problems & Opportunities, 2003.
- [25] Fetterly, D., Manasse, M., and Najork, M. Spam, damn spam, and statistics. *In Proceedings of the Seventh International Workshop on the Web and Databases (WebDB)*, Paris, France, 2004.
- [26] Fleischman, M., Hovy, E., and Echihabi, A. Offline Strategies for Online Question Answering: Answering Questions Before They Are Asked. *Proceedings of the 41st Meeting of the Association for Computational Linguistics (ACL03)*. Sapporo, Japan, 2003
- [27] Gee, K. Using latent semantic indexing to filter spam. *Proceedings of the 18th ACM Symposium on Applied Computing, Data Mining Track (SAC03)*, Melbourne, FL, 2003.
- [28] Girju, R., Badulescu, A., and Moldovan, D. Learning Semantic Constraints for the Automatic Discovery of Part-Whole Relations. *Proceedings of Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, Edmonton, Canada, 2003.
- [29] Gruber, T. What is an Ontology? <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>.
- [30] Gunther, O. *Environment information systems*. Springer, 1998
- [31] Hearst, M. What is Text Mining? <http://www.sims.berkeley.edu/hearst/textmining.html>, 2003.
- [32] Hearst, M. Automatic Acquisition of Hyponyms from Large Text Corpora. *Proceedings of the 14th International Conference on Computational Linguistics (COLING92)*, Nantes, France. 1992.
- [33] Henzinger, M., Motwani, R., and Silverstein, C. Challenges in web search engines. *Proceedings of the 25th Annual International ACM SIGIR Conference SIGIR Forum*, Tampere, Finland, 36(2), 2002.
- [34] Hotho, A., Staab, S., and Stumme, G. Ontologies Improve Text Document Clustering. *Proceedings of 3rd IEEE International Conference on Data Mining (ICDM03)*, Melbourne, FL, 541-544, 2003.
- [35] International Data Group. Worldwide email usage 2002 - 2006: Know what's coming your way, 2002.
- [36] Kietz, J., Maedche, A., and Volz, R. A method for semi-automatic ontology acquisition from a corporate intranet. *Workshop on Ontologies and Text of 12th International Conference on Knowledge Engineering and Knowledge Management (EKAW00) Workshop on Ontologies and Text*, Juan-les-Pins, France, 2000.
- [37] Kiritchenko, S., and Matwin, S. Email classification with co-training. *Proceedings of workshop of the Center for Advanced Studies on Collaborative Research (CASCON01)*, Ontario, Canada, 2001.
- [38] Kiritchenko, S., Matwin, S., and Abu-Hakima, S. Email Classification with Temporal Features. *Proceedings of the International Intelligent Information Systems (IIS04)*, Zakopane, Poland, 523-533, 2004.
- [39] Li, L., and Tan, C. Improving OCR Text Categorization Accuracy with Electronic Abstracts. *Proceedings of 2nd International Conference on Document Image Analysis for Libraries (DIAL06)*, Lyon, France, 82-87, 2006.
- [40] Liu, R. Dynamic Category Profiling for Text Filtering and Classification. *Proceedings of The 10th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD06)*, Singapore, 255-264, 2006.
- [41] Machine Learning Lab. in Information and Computer Science, University of California at Irvine. <http://www.ics.uci.edu/mlearn/MLSummary.html>

- [42] Maedche, A., and Staab, S. Ontology learning for the semantic web. *IEEE Intelligent Systems* 16(2): 72-79, 2001.
- [43] Mann, G. Fine-Grained Proper Noun Ontologies for Question Answering *Proceedings of SemaNet: Building and Using Semantic Networks*, Taipei, Taiwan, 2002.
- [44] Mavroudis, D., Tsatsaronis, G., Vazirgiannis, M., Theobald, M., and Weikum, G. Word Sense Disambiguation for Exploiting Hierarchical Thesauri in Text Classification. *Proceedings of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD05)*, Porto, Portugal, 181-192, 2005.
- [45] Meyer, T., and Whateley, B. SpamBayes: Effective open-source, Bayesian based, email classification system. *Proceedings of the 1st Conference of Email and Anti-Spam (CEAS04)*, Mountain View, CA, 2004.
- [46] Moldovan, D., and Girju, R. Knowledge Discovery from Text. *Tutorial of the 41st Meeting of the Association for Computational Linguistics (ACL03)*. Sapporo, Japan, 2003
- [47] Monostori, L., Vancza, J., and Ali, M. Ontology integration tasks in Business-to-Business E-Commerce. *Proceedings of the 14th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE01)*, Budapest, Hungary, 2003.
- [48] Navigli, R., Velardi, P., and Gangemi, A. Ontology learnig and its application to automated terminology translation. *IEEE Intelligent Systems* 18(1): 22-31, 2003.
- [49] Osterwalder, A., Parent, C., and Pigneur, Y. Setting up an Ontology of Business Models. *Proceedings of 16th International Conference on Advanced Information Systems Engineering (CAiSE03) Workshops (3)*, Riga, Latvia, 319-324, 2004.
- [50] Pantel, P., and Lin, D. Discovering Word Senses from Text. *Proceedings of 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD02)*, Edmonton, Canada, 613-619, 2002.
- [51] Pantel, P., and Ravichandran, D. Automatically Labelling Semantic Classes. *Proceedings of Human Language Technology conference North American chapter of the Association for Computational Linguistics annual meeting (NAACL/HLT04)*, Boston, MA, 2004.
- [52] Pasca, M. Finding Instance Names and Alternate Glosses on the Web: Word-Net Reloaded. *Proceedings of 6th International Conference of Computational Linguistics and Intelligent Text Processing (CICLing05)*, 280-292, Mexico City, Mexico, 2005.
- [53] Pasca, M. Acquisition of categorized named entities for web search. *Proceedings of the 13th ACM Conference on Information and Knowledge Management (CIKM-04)*, Washington, D.C, 2004.
- [54] Perkins, A. The classification of search engine spam. <http://www.ebrandmanagement.com/whitepapers/spam-classification/>.
- [55] Pu, C., Webb, S., Kolesnikov, O., Wenke, L., and Lipton, R. Towards the integration of diverse spam filtering techniques. *Proceedings of the IEEE International Conference on Granular Computing (GrC06)*, Atlanta, GA, 17-20, 2006
- [56] Pundt, H., and Bishr, Y. Domain ontologies for data sharing: An example from environmental monitoring using field GIS. *Computer & Geosciences*, 28, 98-102, 1999
- [57] Raghavan, P. The Changing Face of Web Search. *Proceedings of 10th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD06)*, Singapore, 2006.
- [58] Robert von Behren, J., Czerwinski, S., Joseph, A., Brewer, E., and Kubiawicz, J. NinjaMail: The Design of a High-Performance Clustered, Distributed E-Mail System. *Proceedings of Workshop of the 29th International Conference on Parallel Processing (ICPP00)*, Toronto, Canada, 2000.
- [59] Sahami, M., Mittal, V., Baluja, S., and Rowley, H. The happy searcher: Challenges in web information retrieval. *In Trends in Artificial Intelligence, 8th Pacific Rim International Conference on Artificial Intelligence (PRICAI)*, Guilin, China, 2004.
- [60] Sahami, M., Dumais, S., Heckerman, D., and Horvitz, E. A Bayesian Approach to Filtering Junk E-Mail. *Proceedings of the AAAI Workshop on Learning for Text Categorization*, Madison, WI, 1998.
- [61] Shankar, S., and Karypis, G. Weight adjustment schemes for a centroid based classifier. *Computer Science Technical Report TR00-035*, 2000.
- [62] Snow, R., Jurafsky, D., and Ng, A. Learning syntactic patterns for automatic hypernym discovery. *Proceedings of Neural Information Processing*, Vancouver, Canada, 2004.
- [63] Taghva, K., Borsack, J., Coombs, J., Condit, A., Lumos, S., and Nartker, T. Ontology-based Classification of Email. *Proceedings of the International Symposium on Information Technology (ITCC03)*, Las Vegas, NV, 194-198, 2003.
- [64] Waterson, A., and Preece, A. Verifying ontological commitment in knowledge-based systems. *Knowledge-Based Systems*. 12(1-2): 45-54, 1999.
- [65] Yang, J., Chalasani, V., and Park, S. Intelligent Email Categorization Based on Textual Information and Metadata. *IEICE TRANS. INF. & SYST.*, VOL. E82, NO.1 JANUARY 1999
- [66] Yang Y., and Pedersen, J. A Comparative Study on Feature Selection in Text Categorization. *Proceedings of the Fourteenth International Conference on Machine Learning (ICML97)*, Nashville, TN, 412-420, 1997.
- [67] Youn, S., and McLeod, D. Ontology Development Tools for Ontology-Based Knowledge Management. *Encyclopedia of E-Commerce, E-Government and Mobile Commerce*, Idea Group Inc., 2006.
- [68] Youn, S., and McLeod, D. A Comparative Study for Email Classification. *Proceedings of International Joint Conferences on Computer, Information, System Sciences, and Engineering (CISSE06)*, Bridgeport, CT, December, 2006
- [69] Youn, S., and McLeod, D. Efficient Spam Email Filtering using an Adaptive Ontology. *Proceedings of 4th International Conference on Information Technology: New Generations (ITNG07)*, Las Vegas, NV, April, 2007.

Seongwook Youn is currently a Ph.D. candidate in Computer Science Department, University of Southern California. He received the bachelor' degree in computer science from Sogang University, Seoul, Korea, in 1997 and MS degree from University of Southern California in 2003. He was a lecturer at Sogang University, Seoul, Korea, in 1999. His research interests include data classification, data mining, spam filtering, semantic web, and ontology.

Dennis McLeod is currently Professor of Computer Science at the University of Southern California, and Director of the Semantic Information Research Laboratory at USC. He received his Ph.D., M.S., and B.S. degrees in Computer Science and Electrical Engineering from MIT. Dr. McLeod has published widely in the areas of data and knowledge base systems, federated databases, database models and design, and ontologies. His current research focuses on dynamic ontologies, user-customized information access, database semantic heterogeneity resolution and interoperation; personalized information management environments; information management environments for geoscience and homeland security information, crisis management decision support systems, and privacy and trust in information systems.