Journal of Web Engineering, Vol. 7, No.2 (2008) 158-176
© Rinton Press

SEMANTIC SPAM FILTERING FROM PERSONALIZED ONTOLOGIES

VICTORIA EYHARABIDE ANALIA AMANDI

ISISTAN Research Institute, UNICEN University, Argentina
CONICET Consejo Nacional de Investigaciones Científicas y Técnicas, Argentina
{veyharab, amandi}@exa.unicen.edu.ar

Received December 7, 2007
Revised April 21, 2008

One of the biggest problems that Internet faces is the increase of email spam. The main drawback with previous anti-spam filters is that they are based only on 1) the syntactical features of words lacking semantic analysis, or 2) on what the majority of users regard as spam without considering the individual preferences of a particular user. In this paper we present a spam email filter that personalizes its filtering process using an email user profile that contains the user's preferences regarding emails. Our innovative email user profile is based not only on some common user profiling techniques but also on the knowledge contained in a domain ontology. The user profile is used to learn which spam emails (although unsolicited and large-scale sent) are interesting for the user, despite they are spam. The encouraging experimental results provide empirical evidence of the effectiveness of using an ontological approach to user profiling in an email spam filter.

Key words: Ontology, Spam Filtering
Communicated by: D. Lowe & O. Pastor

1 Introduction

Even though email spamming is widely reviled, it is disturbingly increasing. As email continues being the most popular form of communication among Internet users, the email spam is expected to grow even more. The reason for this is that spammers (as people that send spam are usually called) find in this media a fast, economical and effortless way to broadcast their advertisements. Although there have been several attempts to cope with spam, they have not reached great progress. Given the minimal costs and the broad band of audience covered, spammers continued improving their tricks to avoid spam filters.

Previous commercial and research anti-spam efforts are basically centered in machine learning algorithms such as support vector machines or naïve Bayesian classifier. However, the main drawback with these approaches is that they are based only on syntactical analysis without considering the email semantic, which refers to the meaning conveyed by its content. Therefore, in this paper we present an approach for spam filtering that takes into account the context of the emails by the use of ontologies. Context can be used to interpret emails, making a spam filter much more efficient. We show how context-awareness is a feature that allows a filter to personalize its filtering process and, as a result of this, improve its performance.

Currently, the majority of anti-spam filters consider that every spam email (unsolicited email sent in bulk) will be uninteresting for the user. However, even though an email is sent to several receivers, a particular user could be interested in reading it. For example, while a lawyer could not be interested in an email offering children book discounts, a kindergarten teacher could read it. Even more, email interests might also vary for the same user as his/her preferences and needs change over time. Therefore, a spam filter should be personalized. A way to personalize a spam filter is to build a user profile. A user profile [22] is a model that captures specific user behavior. Here, we introduce an innovative email user profile based not only on some common user profiling techniques but also on the knowledge contained in a domain ontology. The user profile is used to learn which spam emails (although unsolicited and large-scale sent) are interesting for the user, despite they are spam. As a result, our ontology-based spam filter considers not only syntactic but also semantic aspects of emails.

Our email user profile is based on two well-known techniques: association rules and ontologies. On one hand, we decided to use ontologies because they are very useful to disambiguate and also to identify the semantic categories of a particular domain. Ontologies express the main concepts and relationships in a domain in a way that is comprehensible to the user. They provide an explicit conceptualization (i.e., meta-information) that describes the semantics of the data. This enables automatic support for acquiring, maintaining and accessing information. On the other hand, association rule mining is one of the major techniques in data mining and it is perhaps the most common form of local-pattern discovery in unsupervised learning systems. The key strength of association rule mining is that it can efficiently discover the complete set of associations that exist in data. Consequently, thanks to the combination of these two techniques, we achieve a semantically-enriched email user profile.

Our user profiling algorithm has two principal building stages. In the first stage, we extract association rules from the user email data base. However, some of the obtained association rules are very similar, which led to the presumption that they might have “something” in common. Therefore, in the second stage, after pruning the redundant rules, we try to summarize those rules that express the same knowledge. In particular, using an ontology as a source of domain knowledge, for each pair of email attributes we seek if they are related to a common concept in the ontology. If that is the case, we compress the rules that contain those attributes into a more general rule in which we replace the attributes by their shared concept. The experimental results applied to several e-mail experiences databases provide empirical evidence of the effectiveness of using an ontological approach to user profiling in an email spam filter.

The rest of this paper is organized as follows. In section 2 we begin by presenting our ontology-based user profile approach by explaining in detail each of its building stages. In Section 3 we depict a case study in which our email user profile is used to personalize spam filters. Later, in Section 4 we show the results obtained from the experiments we have carried out in order to validate our approach. Then, in Section 5 we introduce some work related to this research. In particular, we discuss some rule refinement approaches, current spam filtering techniques as well as other related user-profiling approaches. Finally, in Section 6 we present our conclusions.

2 Email User Profile

To detect uninteresting emails, a personalized spam filter should first analyze the user's behavior in order to learn his/her interests and preferences regarding emails. Then, the data obtained from that analysis is gathered in a user profile. A user profile is a model containing the most important or interesting facts about the user ([22], [11]). Additionally, in this approach the user profile is enriched with ontological knowledge. Once the email user profile is acquired, a personalized spam filter can use it to adapt its filtering process accordingly.

In this section we present our email user profiling technique. We build the email user profile using the following steps:

i) **Association rule mining**: Initially, we generate association rules to identify relationships between emails and actions that the user has performed to manage them (such as read, delete, reply, forward or move a specific folder a particular email). This step is described in section 2.1.

ii) **Similar rule grouping**: Then, we group those rules containing the same user action in the rule's consequent. As a result, each group of association rules describes the emails for which the user has performed the same action (see section 2.2).

iii) **Rule filter**: Subsequently, in each group we filter redundant and not interesting rules using a) some well-known pruning algorithms [24] as we depict in section 2.3 and b) some ontology-based pruning strategies as we describe in section 2.5.

iv) **Pattern extractor**: Some of the obtained rules are very similar, which led to the presumption that they might have "something" in common. Consequently, using a domain ontology, we try to summarize those association rules that express the same knowledge. In particular, we seek if the email attribute values are related to some common concept in the ontology. If that is the case, we compress the rules into a more general rule containing the shared concept (see section 2.4).

Finally, we obtain an email user profile which is a set of ontology-enriched association rule groups. Figure 1 depicts the main components of the email user profiling technique. In the following sections, we explain in detail the principal components of our approach.

2.1 Association rule generator

Initially, we extract association rules from a collection of email situations using the well-known Apriori algorithm [1]. An email situation is the email originating it and the user action over that email (such as read, delete, forward or move a specific folder a particular email). An email is represented by the email headers and a set of keywords extracted from its body. To apply rule mining, each email situation is considered as a transaction in which each email attribute is seen as an item.

Association rule mining generates rules that identify patterns in transaction data describing which events frequently occur together. According to Agrawal and Shafer in [1] association rule mining is stated as: Let $I = i_1, \ldots, i_n$ be a set of items and D be a set of transactions, each consisting of a subset X of items in *I*. An association rule is an implication of the form X ➔ Y, where $X \subseteq I$, $Y \subseteq I$, and also X and Y are disjoint itemsets. X and Y are the rule's antecedent and consequent respectively. A rule expresses that if the antecedent holds then we expect the consequent to be also satisfied.

Each rule has two metrics: support and confidence. A rule has support *s* in *D* if *s* percent of *D*'s transactions contains X ∪ Y. A rule has confidence *c* if *c* percent of *D*'s transactions that contain X also contain Y. Given a transaction database *D*, the problem of mining association rules is to find all association rules that satisfy a predefined support and confidence thresholds (known as *minsup* and *minconf* respectively).

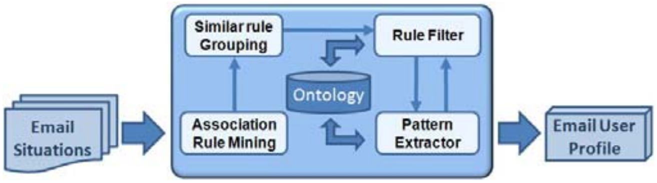

Figure 1 - Email user profiling technique

2.2 Similar rule grouping

After generating the association rules, we group those ones that refer to the same user action. We consider five kinds of user actions over incoming emails: Read, Delete, Reply, Forward and Move to a folder. As a result, each group describes the emails for which the user has performed a certain action. Particularly, inside a group the antecedent of each rule is a set of email attributes describing a certain email, and the consequent of the rule is the concrete action that the user has executed over those emails. A group is as follows:

R1: Email_attribute_set$_1$ ➔ User_action = action$_i$
R2: Email_attribute_set$_2$ ➔ User_action = action$_i$
. . .
Rn: Email_attribute_set$_n$ ➔ User_action = action$_i$

Where Email_attribute_set = {Attribute$_1$= *value$_1$*, Attribute$_2$= *value$_2$*, …, Attribute$_n$ = *value$_n$*}. For example, a possible group of association rules describing the emails that the user has read could be:

R1: FromAddress=*mgaedke@gmail.com* ∧ Subject=*ontologies* ➔ User_action=*read_email*
R2: FromAddress=*ataylor@gmail.com* ∧ Subject=*semantic_web* ➔ User_action=*read_email*
R3: Subject=*ontologies* ∧ Month=*october* ➔ User_action=*read_email*

2.3 Association rule filter

A vast amount of association rules is obtained when applying data mining techniques to a training set. Usually, these processes require some pruning and summarizing techniques in order to extract meaningful patterns.

Our approach is based on the pruning heuristics proposed by Shah and coauthors in [24]. Initially, we eliminate the redundant rules. The first pruning heuristic suggests removing those association rules whose antecedents are too specific. Consequently, given the rules A ∧ B ➔ C and A ➔ C, both rules with similar strength[a], then the first rule is redundant. For instance, suppose we discover the following two rules with similar strength:

R4: Attachment=*yes* ∧ ReceiverCant=>*5* ∧ Size=>*10MB* ➔ User_action=*delete_email*
R5: Attachment=*yes* ∧ ReceiverCant=>*5* ➔ User_action=*delete_email*

According to the previous pruning rule, the first rule (R4) is redundant because the second rule (R5) implies that the user always deletes those emails with attachments sent to more than 5 receivers without caring for the email size, as the first one says. A second pruning heuristic suggests eliminating those association rules whose consequent is too general. Therefore, given the rules A ➔ B and A ➔ B ∧ C, both rules with similar strength, then the first rule is redundant.

The existing pruning approaches developed thus far focus on the elimination of uninteresting or redundant rules. These approaches reduce the number of the discovered association rules by applying formal logics or by using some predefined constraints that describe the user target patterns. In contrast, after applying the basic pruning rules described previously, we improve our filter by carrying out an ontology-based association rule filtering process. The ontology-based filter will be explained in detail in section 2.5.

2.4 Pattern extractor

Despite the elimination of several redundant and not interesting association rules, the number of remaining rules in each group could be still too high. More rules do not mean more knowledge. In fact, some of the obtained rules are very similar, which led to the presumption that they might have "something" in common. Consequently, using a domain ontology we try to summarize those association rules that express the same knowledge. Therefore, for each pair of email attributes values, we seek if they are related to only one common instance in the ontology; given that situation, we replace them by their shared concept.

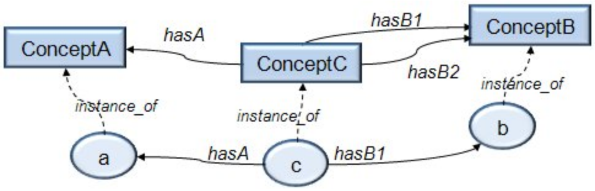

Figure 2 - Ontology example

For example, suppose we have the ontology portrayed in figure 2[b]. In the ontology the instances *a* and *b* are both related to *c* (an instance of the concept *ConceptC*) by the relationships *hasA* and *hasB1*

[a] As defined in [24], the strength of an association rule is the confidence of the rule and two rules have similar strength if for a small pre-defined value 1 > ε > 0, | *strength*(*Rule1*) – *strength*(*Rule2*) | < ε.

[b] In all the figures that show ontologies, we represent concepts with squares and instances with circles. Dashed lines denote the relationship "instance of". For example, in figure 2 "*a*" is an instance of the concept "*ConceptA*".

respectively. Hence, there are two possible replacement cases: i) when both attributes are in the same association rule or ii) when they are in different association rules.In the first replacement case, given the rule A = *a* ∧ B = *b* ➔ User_action = *X*, we replace the attributes with their common instance and its corresponding concept resulting in the rule ConceptC = c (A, B) [ConceptC *hasA* ConceptA, ConceptC *hasB1* ConceptB] ➔ User_action = *X*. For example, suppose we discover the rule:

R6: FromName=*Martin_Gaedke* ∧ FromAddress=*mgaedke@gmail.com* ∧ Subject=*question* ➔ User_action=*replay_email*

The ontology in figure 3 depicts that the instances "Martin_Gaedke" and "mgaedke@gmail.com" are connected to the instance "Person_Martin" which is a researcher. In consequence, we summarize R6 in R7:

R7: Researcher=*Person_Martin*(FromName, FromAddress) ∧ Subject=*question*➔User_action=*replay_email*
[Researcher *hasName* Name, Researcher *hasEmailAddress* EmailAddress]

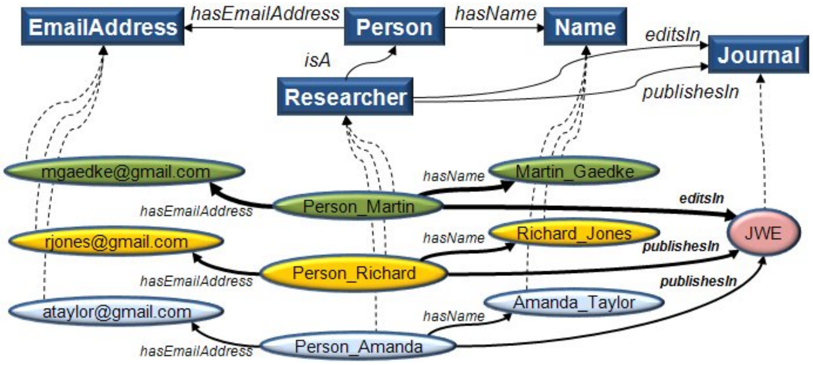

Figure 3 - Email ontology example

As it can be seen in the example, for each compressed attribute we keep track of i) the email attributes that generates it (FromName and FromAddress) and also ii) their common concepts and relationships in the ontology (Researcher *hasName* Name, Researcher *hasEmailAddress* EmailAddress). It is important to maintain the information that originates the composed attribute for two reasons: to understand the meaning and to detect the possibly inclusion of the resulting rule (see next section). In the example, the meaning of the final compressed rule is that the user replies to those emails with subject "question" sent by the researcher "Person_Martin" who has name "Martin_Gaedke" and has address "mgaedke@gmail.com".

The second replacement case is when the attributes are in different association rules. Therefore, for each pair of rules with similar strength that only differ in the two mentioned attributes, we summarize the rules into a more general rule containing the shared concept. Thus, given R8: A = *a* ∧ Y ➔ User_action = *X* and R9: B = *b* ∧ Y ➔ User_action = *X*, we combine both rules in R10: ConceptC = c (A, B) [ConceptC *hasA* ConceptA, ConceptC *hasB1* ConceptB] ∧ Y ➔ User_action = X (where Y is a set of email attributes). For instance, the following two rules (R11 and R12) can be compressed in rule R13:

R11: FromName=*Martin_Gaedke* ∧ Day=*15* ∧ Month=*May* ➔ User_action=*read_email*
R12: FromAddress=*mgaedke@gmail.com* ∧ Day=*15* ∧ Month=*May* ➔ User_action=*read_email*

R13: Researcher=*Person_Martin*(FromName,FromAddress)∧Day=*15*∧Month=*May*➔User_action=*read_email*
[Researcher *hasName* Name, Researcher *hasEmailAddress* EmailAddress]

Notice that we only join two attributes only if they have a unique instance in common. Otherwise, if they are related to several shared instances, we could not discriminate to which one they are referring to.

2.5 Ontology-based filter

After the pattern extractor stage, the iterative process returns to the filtering stage. At this point, we can explain our ontology-based pruning strategies. Since we are searching for new and previously unknown knowledge, we eliminate those rules that are already represented in the ontology. For example, if in the ontology there is a relationship between the concept A and the concept B, then the rule A ➔ B is eliminated.

Another pruning rule is the attribute inclusion finder. In this stage, we remove those rules that are included within other ones. Continuing with the pruning heuristics presented by Shah et al. [24], given two rules with similar strength, if the first rule's antecedent is included in the second one's antecedent, then the first rule is redundant. Therefore, if there are two implications (both with similar strength) of the form R14: A ➔ C and R15: B ➔ C and $B \subset A$ but $A \not\subset B$, then R15 is redundant. For example, R17 subsumes R16; consequently, R16 can be eliminated:

R16: FromAddress=*mgaedke@gmail.com* ➔ User_action=*forward_email*
R17: Researcher=*Person_Martin* (FromName, FromAddress) ➔ User_action=*forward_email*
[Researcher *hasName* Name, Researcher *hasEmailAddress* EmailAddress]

At this point it is vital to emphasize the importance of gathering the previous information of each compressed attribute. The previous situation is a clear example in which the original attributes are used to determine the inclusion of a rule. Finally, we repeat the iterative process of filtering and finding common concepts in each group of association rules until no more changes are found.

3 Case study

In this section, we present a case study in which our email user profile is used to personalize a spam filter. In particular, we illustrate through an example the improvement of an existent spam filter using the knowledge gathered in our email user profile.

Consider the situation in which the user has moved to a specific folder a new incoming email classified as spam by his/her spam filter. Suppose in that folder there are several emails that were not classified as spam by the filter. Therefore, there are two possible cases: 1) the new email was *correctly* classified as spam and the user put it in that folder by accident; or 2) the new email was *wrongly* classified as spam. Our goal is to determine which one of these two cases is true. Consequently, we will try to find out if the new email is really spam or not using the ontology-based email user profile. If we discover that the new email is somehow related to those ones in the specific folder, we can infer (with certain confidence level) that the new email was wrongly classified. If that is the case, we have detected a "false positive" case (i.e., an email wrongly classified as spam) and thereby improved the spam filter precision.

We divide the case study description in two different steps. First, we describe in detail the building process of the part of the user profile that depicts the emails contained in the folder to which the user has moved the new email. Second, we show how the wrong spam classification of the new email can be corrected using that part of the user profile.

Suppose the user has moved the new email to a certain folder called "Number 7". Initially, we generated association rules from the emails gathered in that mentioned folder. Then, after filtering the association rules generated, we obtained the following group of rules describing those emails:

R18: ToName=Sarah_Wilson ➔ User_action=*Move_to_folder7*
R19: ToAddress=swilson@gmail.com ➔ User_action=*Move_to_folder7*

R20: FromName=*Martin_Gaedke* ➔ User_action=*Move_to_folder7*
R21: FromAddress=*mgaedke@gmail.com* ➔ User_action=*Move_to_folder7*

R22: FromName=Richard_Jones ➔ User_action=*Move_to_folder7*
R23: FromAddress=rjones@gmail.com ➔ User_action=*Move_to_folder7*

R24: FromName=Amanda_Taylor ➔ User_action=*Move_to_folder7*
R25: FromAddress=ataylor@gmail.com ➔ User_action=*Move_to_folder7*

Analyzing these rules, we can see that they are very similar. They only differ in one attribute. Although, we have eight different rules, we cannot discover too much knowledge because all of them are expressing almost the same. Consequently, we search in the ontology depicted in figure 3 if they have concepts in common. As a result, based on the summarizing process presented in section 2, we can summarize each pair of rules obtaining:

R26: Researcher=*Person_Sarah* (ToName, ToAddress) ➔ User_action=*Move_to_folder7*
[Researcher *hasName* Name, Researcher *hasEmailAddress* EmailAddress]

R27: Researcher=*Person_Martin* (FromName, FromAddress) ➔ User_action=*Move_to_folder7*
[Researcher *hasName* Name, Researcher *hasEmailAddress* EmailAddress]

R28: Researcher=*Person_Richard* (FromName, FromAddress) ➔ User_action=*Move_to_folder7*
[Researcher *hasName* Name, Researcher *hasEmailAddress* EmailAddress]

R29: Researcher=*Person_Amanda* (FromName, FromAddress) ➔ User_action=*Move_to_folder7*
[Researcher *hasName* Name, Researcher *hasEmailAddress* EmailAddress]

Having a closer look at the resulting rules, we notice that we obtain four different researchers in the rule's antecedent. Therefore, our goal now is to find what those persons have in common according to the ontology. Firstly, we check if the attributes are derived from the same original attributes. Thus, the researcher in R26 (Person_Sarah) cannot be summarized as it is derived from the attributes “ToName” and “ToAddress” whereas the others come from “FromName” and “FromAddress”. Secondly, we check if the other three rules (R27, R28 and R29) share common concepts and relationships in the ontology. Hence, we discover that the other three persons are related to “JWE” which is an instance of a “Journal”. However, “Person_Martin” is related by the relationship “editsIn” whereas “Person_Richard” and “Person_Amanda” are related by the relationship “publishesIn”. Consequently, we compress rules R27, R28 and R29 in only one rule with two possible ontological paths. At the end, the group looks as follows:

R26: Researcher = *Person_Sarah* (ToName, ToAddress) ➔ User_action=*Move_to_folder7*

[Researcher *hasName* Name, Researcher *hasEmailAddress* EmailAddress]

R30: Journal=*JWE* (Researcher,FromName,FromAddress)➔User_action=*Move_to_folder7*

[[Researcher *hasName* Name, Researcher *hasEmailAddress* EmailAddress, Researcher ***publishesIn*** Journal]

[[Researcher *hasName* Name, Researcher *hasEmailAddress* EmailAddress, Researcher ***editsIn*** Journal]

Notice that the group size was considerably reduced (eight rules at the beginning and only two rules at the end). Initially, it was difficult to understand at first glance what the rules were expressing. However, at the end, we discovered that the user moves to folder “Number 7” those emails that i) are sent to the researcher “Person_Sarah” or ii) are from those researchers who publish or edit in the journal “JWE”.

Consider now that the new incoming email offers discounts in the subscription of the “JWE” Journal. As the user has moved that email to folder “Number 7”, according to the rules R26 and R30 in the user profile, we search if that new email is related to the researcher “Person_Sarah” or the journal “JWE”. Imagine that the word “JWE” appears in the email subject as well as in the email body. Consequently, we can infer (with certain confidence level) that the new email has a connection with those ones in folder “Number 7” and therefore, the new email was not spam. Although the spam filter could wrongly consider the new email as spam (because it is offering discounts); as it is related with the mentioned journal it might be interesting for the user. As a result, under the guidance of the user profile, we can recover an email that it was going to be ignored or deleted.

At this point it is important to notice that we are searching in the ontology if the attributes are related to some common unique instance, without considering the meaning of the relationships that related them. That is to say, in an ontology there may be either positive or negative relationships. In all the examples presented in this paper, the ontologies contain positive relationships. But an ontology might also state negative relationships; for instance, that a “*Researcher*” “*isDeniedFromPublishIn*” a “*Journal*”. However, our approach is based on the user’s action over emails and not on the ontology. The ontology is only used to refine the rules in the user profile. For the purpose of our work, the positiveness or negativeness of a relationship is not important. The positive or negative action of the user with an email is what matters. For example, in the case study presented here, we have positive evidence that the user is interested in the email (because he/she has moved it to a folder with other emails which are not spam). Based in that evidence, we can infer that the new email was not spam, and therefore, we correct the wrong classification of the filter.

4 Experimental Results

We have carried out two different experiments to validate our email user profile in spam filtering. First, we compared the performance of existent spam filters with and without using our email user profile. Second, we analyzed our technique ability to learn user's email preferences regarding the amount of data available.

4.1 Experimental data

We collected data from a group of 37 users. All participants were researchers of the computer science department at UNICEN University. For each user, we obtained a dataset composed by 500 email situations. Each email situation was composed by an *email*, the *email's spam classification* (i.e. spam or not spam) and the *user's action* to manage it. An email is represented by the email headers (such as the receiver's email address, sender's email address, subject or size) and a set of keywords extracted from its body. Some numerical and time varying variables were averaged, like for example the email's creation or reception time.

The email's spam classification is given by the spam filter that the user is currently using. All users are protected with MailScanner[c], an email security and anti-spam package that runs on UNICEN university mail servers. MailScanner provides spam detection, using public open-relay databases and SpamAssassin[d] which is an open source spam detector package. SpamAssassin employs a variety of mechanisms for detecting spam including header and text analysis, Bayesian filtering, DNS blocklists, and collaborative filtering databases. This spam filter uses a combined score to determine if a given message is spam. If that score exceeds a certain threshold, the system automatically adds the label "[POTENTIAL SPAM]" in the email subject. In addition, some users are using existing commercial spam filters based on naïve Bayesian classifiers without personalized filtering process on their individual desktop PCs. The user's actions considered were read, delete, replay, forward and move to a specific folder. On average, 41,7% of dataset were spam and 58,3% we legitimate email. We built the ontology used in the filtering process with knowledge obtained from the researchers' personal web pages. The ontology contains user's information such as their personal data, research interests, publications or projects in which they participate.

4.2 Personalized Spam Filtering

In the first experiment, as we are interested in whether our email user profile has positive impact on the spam email filtering, we compare for each user the performance of his/her current spam filter in two different ways: i) using and ii) without using our proposed email user profile. The experiment is conducted in two different steps. First, we build an individual email user profile for each user involved in the experiment (using the method described in section 2). Second, we personalize the results of his/her spam filter using that email user profile. To generate association rules, we used WEKA [28], which is a well-known data mining tool. We empirically determine the parameters used to run the algorithms (confidence and support as described in Section 2.1): minconf = 0.8, minsup = 1/N, where N is the number of instances in the dataset. Then, we implemented our approach in JAVA to group and

[c] http://www.mailscanner.info/
[d] http://spamassassin.apache.org/

filter the rules as we mentioned before. The ontology used in the filtering process was implemented in OWL.

We aim at retrieving emails interesting for a user that were wrongly classified as spam by his/her filter. Therefore, two possible measure of effectiveness are precision and recall. The metrics are defined as follows:

$$\text{Precision} = \frac{\text{Number of interesting spam-emails retrieved}}{\text{Number of retrieved spam-emails}}$$

$$\text{Recall} = \frac{\text{Number of interesting spam-emails retrieved}}{\text{Number of interesting spam-emails}}$$

With *precision* we measure if **only** interesting spam-emails are retrieved (i.e., it is the percentage of retrieved spam-emails that are really interesting for the user), whereas with *recall* we measure if **all** interesting spam-emails are retrieved (i.e., it is the percentage of interesting spam-emails retrieved). Retrieving every spam-email would yield maximum recall but poor precision, while retrieving no spam-emails would yield maximum precision but poor recall. The goal is to maximize both concepts at the same time.

	Original Spam Filter		**Enhanced Spam Filter**	
User	Precision	Recall	Precision	Recall
User 1	0,67	0,58	0,87	0,63
User 2	0,74	0,61	0,81	0,67
…	…	…	…	…
User 37	0,72	0,66	0,86	0,69
Average	**0,71**	**0,65**	**0,84**	**0,68**

Table 1 – Precision and recall of both approaches

Initially, we calculated precision and recall for each user in his/her spam filter. Later, we recalculated both metrics for each user but this time enhancing his/her spam filter with our email user profile, as shown in table 1. In particular, for each user we compared the number of interesting spam-emails retrieved against a) the total number of retrieved spam-emails and b) the total number of interesting spam-emails for the user. We consider that a spam-email is interesting for the user if it was classified as spam by his/her filter but the user gives spam-contradicting evidence over it. In this approach, as we do not inquire the user for explicit feedback, we infer that a user is interested in a spam-email when he/she neither deletes it nor moves it to the spam dump folder. Therefore, spam-supporting evidences are email deletions and email moves to the spam dump folder; otherwise, they are spam-contradicting evidences.

In order to compare the original spam-filters against the enhanced spam-filters using our email user profile, we have created separate precision and recall graphs. Figure 4 depicts the first experimental results. While figure 4 (a) shows the precision values, figure 4 (b) shows the recall values for both approaches. To make results comparable, the values shown were obtained by averaging the precision and recall for the different data sets belonging to the users.

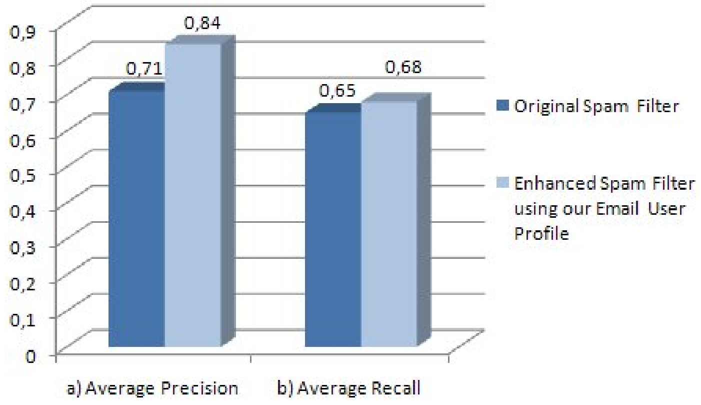

Figure 4 - Average precision and average recall of both approaches

The experimental results show improvements when using the email user profile for spam filtering. As displayed in Figure 4 (a), the enhanced spam filters out performs the original ones from 0,71 to 0,84 better in average precision. Therefore, our approach has a 13% increase in average precision when compared to the other approach. On average, without using our email user profile, only about 70% percent of interesting-spam emails were retrieved. In addition, as we can see in Figure 4(b), recall has also improved. The spam filters using our email user profile had a higher recall (0,68) than the original spam filters (0,65). However, the average recall values of both approaches are very close. The retrieval of interesting wrongly-classified spam emails by using the email user profile produced only an overall recall increase of 0,03.

Some observations can be made from this experiment. Overall, the results of the experiments show that spam filtering combined with personalized user profile achieve a very good performance in both precision and recall. However, the improvements obtained were not as significant as we expected. We have two possible explanations for this result. First, the reduced size of the ontology used in the experiments did not allow the detection of useful semantic relationships among attributes. Second, the data set was not big enough to supply representative samples for the user's behavior. Therefore, we performed a second experiment analyzing how precision and recall vary regarding the amount of data available.

4.3 Email User Profile Adaptation

In the second experiment, we randomly divided the original dataset into smaller datasets differing in 50 email situations. For each dataset, we rebuilt the email user profile for each user measuring precision and recall after each trial. In each dataset, 2/3 of the email situations were used for building his/her email user profile and the rest was used for testing. Figure 5 portrays the experimental results obtained by averaging precision and recall over all the datasets described above. As figure 5 (a) shows, there is a trend of more accurate precision when using our email user profile, especially when more data is available. When focusing on recall (figure 5 (b)), it was found that it tends to maintain as data increases.

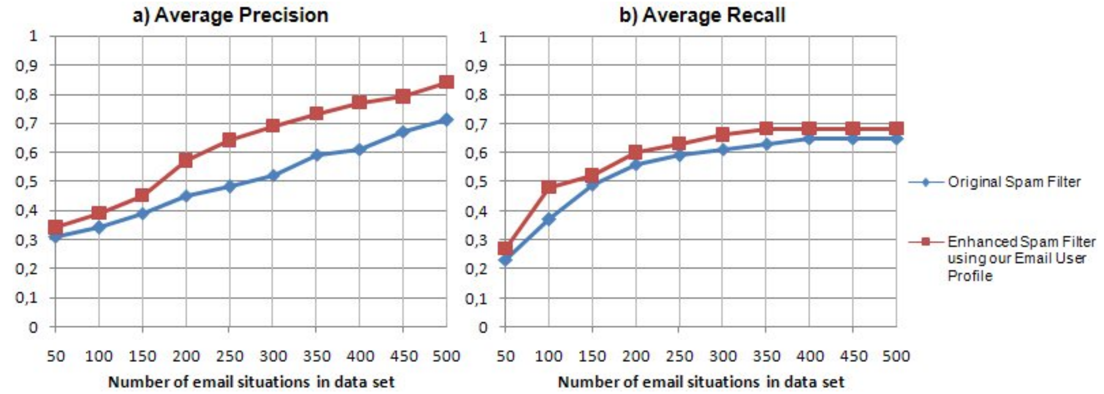

Figure 5 - Comparative performance of email filtering with and without personalization (a) Average precision with different number of email situations. (b) Average recall with different number of email situations

4.4 Discussion

These experiments results show the advantages of our approach. First, the significant increment in precision reveals that our approach is good at retrieving false positives (a legitimate email termed as spam). These experiments support our initial suspicion that a great amount of emails were wrongly classified as spam by conventional spam filters and that by using personalized user profiles we can improve spam filter precision.

Second, recall did not improve as foreseen due to the lack of information in the ontology, as a result of which the email user profile did not learn certain user preferences. Nevertheless, the small increment in recall is also reasonable considering the small size of the training set. Recall is expected to improve as more email situations are added to the dataset and more knowledge is added to the ontology. Consequently, although general improvements pay off on average; as future work, it would be possible to enhance our approach by augmenting the data available and the ontology size, leading to an even finer personalization.

Third, unlike most profiling approaches, as our technique is based on association rules and ontologies, it can be easily comprehended for users. One advantage of both techniques is that they are easily understood by humans, a fundamental characteristic when developing user profiles. From this point of view, our email user profile contrasts with traditional user profiles approaches, which are illegible and merely passive data repositories. Our combination of association rules and ontologies enables the user not only to visualize the rules within the user profile, but also to understand them, revise them and update them afterwards. As we gather for each new compressed attribute the two original attributes and their common relationships in the ontology that originated that compression (see section 2.4), it is easy for the user to follow the intermediate steps that generate that new rule. This provides a trace of the rule evolution to the user so that he/she may be able to interpret the logic behind a certain rule compression; without such information the email user profile will turn into a “black-box” which is not appreciated by users. As future work, by profiting this human-readable characteristic of ontologies, we plan to give the user the possibility of visualizing his/her email user profile in order to modify it and incorporate new personal information to the ontology. Nowadays, given their powerful knowledge representation formalism and associated inference mechanisms, ontologies are emerging as a natural choice for the next generation of user profiles.

Fourth, after analyzing the results obtained from the comparison of both approaches in the previous experiments, we find that:

i) There are cases filtered by the enhanced spam filters which are not filtered by the original ones. The reason for that is that the original spam filters made only a syntactical analysis of words, without considering the email semantic. Consequently, they cannot discover semantic relations between different emails. Therefore, when they detect an email that is not spam, for example, they cannot determine its semantic similarity to other incoming emails to decide if the new ones are spam or not.

ii) All the cases filtered by the original spam filters are also filtered by the enhanced spam filters. We did not build new spam filters from scratch. On the contrary, our aim was to enrich the user's actual spam filter with semantic. Based on the results of the original spam filters, our approach tries to improve those results using a personalized email user profile. Therefore, our approach obtains better results (or the same in the worst case) than the original spam filters.

iii) There are some cases in which our approach fails. The possible personalization improvements achieved by our approach depend on the knowledge gathered in the ontology. Hence, the size and accuracy of the ontology influence directly the performance of our approach. From an ontology with only a few concepts that are wrongly or poorly related, we cannot learn too much. Consequently, as future work we plan to augment the amount of data collected from users; as well as, to enrich the knowledge gathered in the ontology.

Finally, we want to emphasize that enhancing a spam filter with semantic knowledge can bring a much better result. The experiment encouraging results show the usefulness of incorporating semantics and user's preferences in spam email filtering. Since the original spam filter does not use ontological knowledge, the variability seen in the precision and recall figures can be attributed to the lack of semantics. As shown in figures 4 and 5, the results are better when additional semantic analysis is done rather than when only considering syntactical analysis. As a conclusion of the experiments we believe that the proposed method is significant since they can improve the effectiveness of the classification process for email spam filtering.

5 Related work

In this section we describe several works in different research areas that are related to our proposal. First, we analyze some related works in rule refinement, which is the technique used to build our user profile. Second, we present some related work in spam filtering that is the application field in which we test our approach.

5.1 Related work in Rule Refinement

Rule-based systems are widespread and have been successfully employed in many domains ([7], [14], [19], [9], [30], [31]). However, wrong and redundant rules may exist in a rule base. Therefore, rule refinement is crucial for enhancing the efficacy and efficiency of utilizing a rule base. Several research and commercial approaches have been proposed for detecting and eliminating redundant and inconsistent rules. The first rule refinement system was TEIRESIAS [5] which has been designed for the acquisition of new inference rules. Other well-known pioneer systems for rule refinement are SEEK [23] and SEEK2 [9] which are rule-based expert systems for the diagnosis of rheumatological diseases. These approaches compare the expert system conclusions against an available data base of clinical cases with known diagnoses. The comparison generates rule performance statistics for each rule in order to suggest rule refinements for the correction of misdiagnosed cases.

Other approaches have also addressed the problem of rule refinement. For example, the work proposed by Brisoux et al. in [3] is a partial instantiation schema that exports local search to first-order knowledge bases. Another approach is the paper described in [33] which proposes two approaches for refining a rule base: one is to remove implication redundant rules by using the closure of literal set and the other is to remove abstraction redundant rules by using rule abstraction. Based on Zhuge previous work [32], the proposed approach can be used to refine inheritance rules between components [31] and to refine rules in Knowledge Grid [30].

There are also other works that aim at refining a rule base using machine learning techniques, like for example neural networks or case-based reasoning. Some approaches have demonstrated that neural networks are able to perform rule refinement ([8], [7], [27]); that is, once rules have been inserted into the network, they can be verified and even corrected. For instance, Tresp et al. in [27] demonstrate how a set of rules can be incorporated into a neural network of normalized basis functions with the aim of minimizing the number of rules and the number of conjuncts. After training, the refined rules are extracted from the network and analyzed. Another related work is described in [8], which trains a recurrent neural network to recognize a known non-trivial, randomly generated regular grammar. The authors consider the individual transitions between DFA (deterministic finite-state automata) states as rules. By comparing the rules extracted from the trained networks in the form of a DFA with the prior knowledge, the validity of the rules was established.

Other approaches suggest rule refinements using case-based reasoning. The basic idea of case-based reasoning is to solve a problem by using similar case solutions retrievable from a well-maintained case base. Knauf et al. ([19], [18], [17]) present a test case–based methodology for validation of rule based expert systems. The main idea of their refinement technique is to find rules that are “guilty” in the system and to replace them by rules that received “better marks” from the experts. Another similar approach is the one presented by Kelbassa in [14] which discusses the adaptation problem of engineering applications and presents a global case-based approach to the optimal refinement of expert system rule bases.

We can conclude that although there have been considerable efforts in selecting optimal rule refinements, the current state of the art in rule base validation and refinement reveals that there is no generic validation interface and no optimal rule trace refinement [15]. As it is ascertain by several authors ([13], [15], [19], [30]), refinement heuristics are suboptimal for cases with multiple refinement problems and that there is a need for higher order refinement heuristics for coping with this problem. In addition, some authors [10] affirm that there are several open questions about rule refinement: “Are the new rules introduced as a result of the retranslation process acceptable from the semantic point of view?" Unfortunately, classical rule refinement approaches lead to rules that might reflect reality fairly well, but are not readable or interpretable by domain experts [19]. Even worse, these refinement systems might construct rules that reflect the examples correctly, but are wrong with respect to the causal connection they express [15]. Therefore, it is vital to consider the semantic of the rule in the rule refinement process in order to achieve refined rules that are interpretable by humans. In consequence, in this paper we present a rule refinement approach based on ontologies.

Finally, some other works have also combined association rules with ontologies. On one hand, some approaches describe how to improve association rule mining using ontologies. The improvement is obtained by incorporating prior ontological knowledge to direct the association search. Chen and colleagues [4], for instance, introduce an implementation of ontologies with association rule mining for the purpose of finding generalized rules with high support. Another example it is the work presented by Shen et al. [25] which use ontology and semantic web techniques to improve semantic retrieval for association rules. On the other hand, other approaches try to improve ontologies using association rules. These heuristics use the information discovered by association rules to help ontological developments. Among these proposals, is the one presented by Madche and Staab [21] which combine ontologies and association rules to semi-automatically construct ontologies. In other work, Song and colleagues in [26] found that using a combination of association rules with ontologies and information retrieval techniques is effective in semantic query expansion.

In summary, previous works combine ontologies with association rules although with many different purposes. While some approaches improve the association rule mining process using ontologies, other approaches enrich ontologies using the associations discovered by the rules. However, the combination of the techniques mentioned before is innovative in user-profiling since, as far as we know, it has never been applied before for user profile construction.

5.2 Related work in spam filtering

A rich literature on spam filtering techniques exists in the Web field. Heymann and colleagues [12], for example, provide a good survey on approaches which fight spam on social web sites. However, in spite of the benefits spam filtering process can provide, current spam filters also have some disadvantages. Their main weakness is that they operate without a lexicon and then ignore word meaning, leading to a number of semantic errors. In addition, they generally consider neither word variations nor synonyms, what makes it difficult to compare semantically-equivalent spam emails.

Several previous research approaches have demonstrated the effectiveness of using ontologies for supporting the user's behavior discovery process. Among these approaches, there is the one presented by Garofalakis and colleagues in [6], which addresses a new web site log mining analysis tool enhanced with semantic knowledge. Consequently, due to the ontological benefits for mining analysis, some authors [2] argue that using the email semantics as an additional classification parameter might result in improved performance of the spam filter. Some earlier works [4] tried to take advantage of ontologies to incorporate semantic. However, they were restricted only to the use of the "is_a" hierarchy without considering relationships among entities.

Only some recent related works ([2], [29], [20], [16]) have appeared trying to improve spam filters exploiting the full potential of ontologies. Among these approaches, there is the one presented by Brewer and colleagues in [2] which addresses a technique for spam filtering that uses semantics along with the syntax of an email message. Even though they also use an ontology of user's interests, their purpose is to build a new spam filter; whereas ours is to improve existent spam filters by retrieving those spam emails that might be interesting for the user. Another approach that filters spam emails using an adaptive ontology is [29]. Initially, Youn and McLeod created a decision tree from an email database. Then, they map the decision tree into a formal ontology and query that ontology to classify emails as spam or not. Although, this work aims at proposing an efficient spam filter, it is still a research model and it is still at an inception phase.

In another work [20], an email-centric personal intelligent assistant called ECPIA provides Web-based environment to support the processing of emails. This paper describes an agent-based system, which provides ontology-based email management and user's behavior analysis in his/her past emails. In this work, the authors use an ontology to: i) store background knowledge of the user and his/her emails; and ii) combine ontology-based filtering agents for blocking spam. The approach presented by Kim et al. [16] develops a user's preference ontology and then they use it to filter new incoming spam emails. The authors collected user's preference information and email responses to train an association and classification mining system. Later, they translate the rules they got from data mining into axioms to specify predefined relationships in the ontology.

Although these ontology-based approaches also benefit from the incorporation of semantics in the spam filtering process, their work differ from ours in one fundamental aspect: their final purpose is to create an ontology; in contrast, we use an ontology as initial knowledge to improve the email user profile construction. However, regardless how the ontology is used, all experimental results have shown encouraging evidence of the benefit in using ontologies to fight against spam. Nevertheless, as some authors argue [2], using semantics in spam filtering presents still many challenges for future work such as building the ontology, relationship discovery or relevancy scoring.

6 Conclusions

Our work points out an innovative anti-spam approach using an email user profile enriched with ontological knowledge. A first outcome of our proposal is a more effective way to filter spam by adding to the classical syntactical analysis some extra semantic knowledge provided by the ontology. In addition, by combining association rules and domain ontologies, we obtain smaller and more specific email user profiles. Our approach show better quality in terms of comprehensibility for the association rules derived from ontology-based optimization method. However, the amount of useful summarized rules discovered depends on the complexity of the ontology and the number of attributes per rule. Also, our ontology-based email user profile allows new user's email preferences inferred without the need for direct user's querying. The encouraging experimental results provide evidence as to the effectiveness of using an ontological approach for user profiling in an email spam filter. Therefore, we are convinced that more work should be done in the area of user profile personalization to improve existent email spam filters.

References

1. Agrawal, R., & Shafer, J. (1996). Parallel Mining of Association Rules. IEEE Transactions on Knowledge and Data Engineering , 8 (6), 962--969.
2. Brewer, D., Thirumalai, S., Gomadam, K., & Li, K. (2006). Towards an Ontology Driven Spam Filter. ICDEW '06: Proceedings of the 22nd International Conference on Data Engineering Workshops (ICDE 2006) (p. 79). Washington, DC, USA: IEEE Computer Society.

3. Brisoux, L., Gregoire, E., & Sais, L. (2001). Checking depth-limited consistency and inconsistency in knowledge-based systems. International Journal of Intelligent Systems , 16 (3), 319 - 331.
4. Chen, X., Zhou, X., Scherl, R., & Geller, J. (2003). Using an Interest Ontology for Improved Support in Rule Mining. DaWaK: Data Warehousing and Knowledge Discovery, 5th International Conference, (pp. 320-329). Prague, Czech Republic.
5. Davis, R., & Lenat, D. (1982). Knowledge Based Systems in Artificial Intelligence. New York: McGraw Hill Int. Book Company.
6. Giannakoudi,, J., & Sakkopoulos, E. (2007). An Integrated Technique for Web Site Usage Semantic Analysis: the Organ System. Journal of Web Engineering (JWE) , 6 (2), 261-280.
7. Giles, C., & Omlib, C. (1992). Inserting rules into recurrent neural networks. Neural Networks for Signal Processing [1992] II., Proceedings of the 1992 IEEE-SP Workshop , 13-22.
8. Giles, C., & Omlin, C. (1993). Rule refinement with recurrent neural networks. IEEE International Conference on Neural Networks , 2, 801-806.
9. Ginsberg, A. (1988). Automatic refinement of expert system knowledge bases. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
10. Ginsberg, A. (1990). Theory Reduction, Theory Revision, and Retranslation. AAAI, (pp. 777-782).
11. Godoy, D., & Amandi, A. (2005). User Profiling for Web Page Filtering. IEEE Internet Computing , 9 (4), 56-64.
12. Heymann, P., Koutrika, G., & Garcia-Molina, H. (2007). Fighting Spam on Social Web Sites: A Survey of Approaches and Future Challenges. IEEE Internet Computing , 11 (6), 36-45.
13. Kelbassa, H.-W. (2002). Context Refinement - Investigating the Rule Refinement Completeness of SEEK/SEEK2. ECAI, (pp. 205-209).
14. Kelbassa, H.-W. (2003). Optimal Case-Based Refinement of Adaptation Rule Bases for Engineering Design. ICCBR, (pp. 201-215).
15. Kelbassa, H.-W., & Knauf, R. (2003). The Rule Retranslation Problem and the Validation Interface. Proceedings of the Sixteenth International Florida Artificial Intelligence Research Society Conference , 213-217.
16. Kim, J., Dou, D., Liu, H., & Kwak, D. (2007). Constructing a User Preference Ontology for Anti-spam Mail Systems. Canadian Conference on AI, (pp. 272-283). Montreal, Canada.
17. Knauf, R., Gonzalez, A., & Abel, T. (2002). A framework for validation of rule-based systems. IEEE Transactions on Systems, Man, and Cybernetics, Part B , 32 (3), 281-295.
18. Knauf, R., Philippow, I., & Gonzalez, A. (2000). Towards validation and refinement of rule-based systems. journal of experiment and theoretical artificial intelligence , 12 (4), 421-431.
19. Knauf, R., Philippow, I., Gonzalez, A., Jantke, K., & Salecker, D. (2002). System Refinement in Practice - Using a Formal Method to Modify Real-Life Knowledge. Proceedings of the Fifteenth International Florida Artificial Intelligence Research Society Conference, (pp. 216-220).
20. Li, W., Zhong, N., & Liu, C. (2006). ECPIA: An Email-Centric Personal Intelligent Assistant. Rough Sets and Knowledge Technology, First International Conference, RSKT 2006, (pp. 502-509). Chongquing, China.
21. Maedche, A., & Staab, S. (2001). Ontology Learning for the Semantic Web. IEEE Intelligent Systems , 16 (2), 72--79.
22. Pazzani, M., & Billsus, D. (1997). Learning and Revising User Profiles: The Identification of Interesting Web Sites. Machine Learning , 27 (3), 313--331.
23. Politakis, P. (1998). Empirical Analysis for Expert Systems. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
24. Shah, D., Lakshmanan, L., Ramamritham, K., & Sudarshan, S. (1999). Interestingness and Pruning of Mined Patterns. ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery.
25. Shen, B., Yao, M., Wu, Z., Zhang, Y., & Yi, W. (2006). Ontology-based Association Rules Retrieval using Protege Tools. ICDMW '06: Proceedings of the Sixth IEEE International Conference on Data Mining - Workshops (pp. 765--769). Washington, DC, USA: IEEE Computer Society.
26. Song, M., Song, I.-Y., Hu, X., & Allen, R. (2005). Semantic Query Expansion Combining Association Rules with Ontologies and Information Retrieval Techniques. Data Warehousing and Knowledge Discovery. 3589, pp. 326-335. Springer-Verlag Berlin Heidelberg 2005.
27. Tresp, V., Hollatz, J., & Ahmad, S. (1993). Network Structuring and Training Using Rule-Based Knowledge. Advances in Neural Information Processing Systems 5, [NIPS Conference] (pp. 871--878). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
28. Witten, I., & Frank, E. (2005). Data Mining: practical machine learning tools and techniques.
29. Youn, S., & McLeod, D. (2007). Efficient Spam Email Filtering using Adaptive Ontology. ITNG '07: Proceedings of the International Conference on Information Technology (pp. 249--254). Washington, DC, USA: IEEE Computer Society.

30. Zhuge, H. (2002). A Knowledge Grid Model and Platform for Global Knowledge Sharing. Expert Systems with Applications , 22 (4), 313-320.
31. Zhuge, H. (1998). Inheritance rules for flexible model retrieval. Decision Support Systems , 22 (4), 379--390.
32. Zhuge, H. (1995). Research on Object Analogical Reasoning. Journal of Software , 6, 5260.
33. Zhuge, H., Sun, Y., & Guo, W. (2003). Theory and algorithm for rule base refinement. IEA/AIE'2003: Proceedings of the 16th international conference on Developments in applied artificial intelligence (pp. 187--196). Springer Springer Verlag Inc.