SPAMWallah

A Rule-Based E-mail Spam Detection System

6.871 Final Project

Buddhika N. Kottahachchi Arjun R. Narayanswamy

{buddhika, arjunrn}@mit.edu

Abstract

We present SpamWallah: a novel rule-based system that models the thought process of a human trying to identify spam by analyzing the semantic content in a message header. We further provide an extensive codified knowledge structure that can be used in determining the relevance of a message sender to a given recipient/user.

Introduction

Electronic mail is widely regarded as the most popular application on the Internet today. However, with its proliferation there has been a parallel increase in the volume of irrelevant messages users are exposed to: spam. Spam can take many forms ranging from advertisements and solicitations to jokes and chain letters – most of which are unsolicited. Users in general dislike being subjected to such messages. However, recent trends like Yahoo's change in its privacy policy [1] to allow direct marketing to its users only make it more likely for them to be the target of such messages. Given this, the need for a system by which such messages can be filtered is highly desirable. Ideally, such a system should be able to detect relevant messages and expose only those messages to the user.

In order to do this, we must first be able to accurately differentiate between relevant and irrelevant messages. There is a significant body of research focusing on this particular question. Current popular approaches to this problem include probabilistic classification using Bayesian networks [2,4], nearest neighbor models using keywords [3], and prototypes, etc. There have also been many comparisons of the merits of these approaches [5,6,7].

These approaches are well understood, customizable and do a reasonably good job of identifying Spam. However, they are all effectively pattern-matching or complicated mathematical models. Some of these need to be trained, and their performance varies dramatically with training noise and different training parameters. In fact, the training requirements may be completely unusable in a real context – consider a commercial user who is asked to train the system with a 1000 e-mail messages before using it.

Additionally, these systems do not incorporate knowledge of the semantic content of an e-mail message. They do not model how a user thinks and processes information about the problem; and therefore remain hard to configure and hard to understand. We propose SpamWallah as a system that makes some headway in a different direction.

To situate our argument, consider the example of a fictional MIT researcher - Dr. Alotta Spam – who walks into her office in the morning and sees in excess of hundred e-mail message headers in her inbox. These e-mails include conference announcements, private correspondence, student e-mails and also a lot of junk e-mail both from companies and from friends. Dr. Spam has the task of quickly sifting through her inbox and identifying which e-mails are relevant to her: and this is a task that she accomplishes with surprising speed and accuracy.

Now the question: how does she do this? Note that she has only scanned through her inbox and picked relevant messages; which means that she has not looked at message text to make this determination. This automatically means that she is not using any of the models of inference we mentioned above. Clearly she has access to some knowledge, and some ways of applying that knowledge which enables her to make this determination from the e-mail message header itself.

Our Approach – SpamWallah

Given this, we propose SpamWallah, a rule-based system that models a human user's decision process in determining if an e-mail message is spam or non-spam. SpamWallah extracts semantic content from the message headers and uses it in the context of prior interactions with the message sender and recipients to determine the relevance of an e-mail message to a recipient.

Input

SpamWallah takes in as input one e-mail message header. Particularly it needs access to those fields that humans use to make their spam determination. These are: `from:` field, `subject:` field, and the `to:` field.

Auxiliary Input
It is important for the SpamWallah system to have knowledge of the subjects and people that the user normally sees. Therefore, as auxiliary input, SpamWallah requires access to a properly configured comprehensive e-mail client. In particular SpamWallah requires access to information residing in the user e-mail client's Inbox, Sent Folder, Contacts Folder and E-Mail Subscriptions. The auxiliary input required by SpamWallah is further characterized by the following constraint:

- Auxiliary input may only include such information as may be reasonably gathered by any comprehensive e-mail client. Thus, while we expect access to the e-mail client's Inbox, Sent Folder and Contacts Folder; we do not expect the SpamWallah system to have any other outside knowledge.

Output
Two values: the confidence level for the message being spam and the confidence level for the message being non-spam.

The final objective is that both e-mail header and auxiliary input will be automatically obtained from a given e-mail client. User interaction will be kept to an absolute minimum as the system silently goes about the task of determining if a given e-mail is spam. But for demonstration purposes, the interaction between SpamWallah and an e-mail client is modeled as follows: SpamWallah asks questions and the user supplies answers from her given e-mail client.

This frees us from the burden of implementing a plug-in or an interface with an e-mail client and instead allows us to simply focus on modeling the knowledge that would be needed by such a system. Therefore we rely on the user to act as the interface between the data available on the e-mail client and SpamWallah. This is reflected in the manner we have worded some of our questions.

High-Level Intuition

Our definition of spam is conceptually based on the following chart. One axis represents the notion that the sender may be *known* or *not known*. The second axis represents the notion that the content may be *irrelevant* or *not irrelevant* to the user. Note that the relevance is defined implicitly as an absence of irrelevance; we believe this formulation capture the nature of the problem more accurately than the converse.

	Content not Irrelevant	Content Irrelevant
User Known	OK	?
User Not Known	?	SPAM

Given this, we can infer that a message originating from an "Unknown User" and which is "Irrelevant" to the user can be classified as SPAM (with 100% confidence) while those from "Known" senders with content that is "Relevant" can be classified as OK or non-SPAM. The gray areas in the chart describe situations where the behavior of the system is not clear. If an e-mail arrives from a hitherto unknown user, and the content is not clearly irrelevant, then the system is faced with a choice. In some situations this should lead to a spam classification (eg: Misaddressed e-mail), and in some cases it should not (eg. Departmental announcements). Alternatively, it is possible for users that we know to send us irrelevant e-mail (eg. Chain-mail). We may or may not choose to classify such e-mail as spam.

An interesting feature of this conceptualization is that we explicitly allow for the possibility of receiving irrelevant e-mail from known senders (and vice-versa). In practice though, we believe a significant portion of a user's spam determination comes from whether or not they recognize the sender. This is an attribute that has previously received minimal attention and we believe that it merits further attention. As a result a significant part of our knowledge base focuses on determining this property for a given e-mail message.

Example

Let us examine e-mail sent by an MIT departmental coordinator to a departmental jobs list. The sender regularly sends e-mail messages of which many are irrelevant and a few highly relevant. We choose this example because it shows off some the ambiguity inherent in spam determination - even human users have a hard time classifying email from this sender. It will be instructive to watch the performance of SpamWallah on this example.

Input

```
From    : Anne Hunter
To      : joblist@altoids.mit.edu
Subject : Proctor Needed - No Expertise
          Required! $12.50/hour!
```

System-User interactions(User models system interface with mail client)

```
1) When did you last store a
   message from the sender?
** within a week 1.0

2) How many messages from the
   sender have you stored?
** 0-10 1.0
```

The system is trying to collect evidence for sender relevance through the sender's presence in the user's Inbox. Checking to see if the user has any messages stored which would imply the sender had sent messages that were of some interest to the user – deduced from the fact that the user hadn't deleted the message. The closer in time to the current the higher the relevance. Furthermore the higher the occurrences of stored messages the more likely the sender is relevant to the user. In this case, the user had stored a message from Anne Hunter during the past week although he didn't have too many messages stored in his inbox. So the bias towards sender relevance will be mild.

```
3) When did you last forward a
   message from the sender?
** never 1.0
```

The system is trying to do the same as above, but with respect to messages the recipient found useful enough to forward. Since the user hadn't forwarded a message from Anne Hunter before, the system is intelligent enough to recognize that the number of messages forwarded is zero and doesn't query the user about it. This doesn't contribute towards sender relevance but mildly pushes bias towards sender irrelevance in this example.

```
4) When did you last reply a
   message from the sender?
** more than a month 1.0


5) How many messages from the
   sender have you replied?
** 0-10 1.0
```

The system is trying to do the same as above, but with respect to messages the recipient found useful enough to reply. These messages would provide the recipient with the highest value and therefore they affect the bias more. The user had responded to a message from Anne Hunter over a month ago and therefore in total it has a mild affect on the bias towards the positive direction.

```
6) Does the sender belong to a
   known domain?
** YES 1.0

7) Does the sender belong to a
   known but irrelevant domain
   (ex. Hotmail, Yahoo etc.)?
** NO 1.0

8) How many entities from this
   domain have you seen?
** 100 1.0
```

The system is trying to see if the sender is related to an entity known to the recipient. It does so by checking if the sender is from a known domain. However, the system checks if the domain is irrelevant by explicitly asking for cases like Yahoo, Hotmail etc. and implicitly by checking how many entities are known from that domain – the more entities known the less information you get about the relevance of the domain in determining sender relevance. In this case, we know that Anne Hunter is from MIT.EDU but we know well over a 100 entities from there and so to an MIT student it doesn't mean much – while this may not seem intuitive in the case of Anne Hunter, considering a random e-mail coming through the reuse@mit.edu list may help understand why this reasoning is used.

```
 9) Is the sender listed as a
    recipient in a message you
    stored?
 ** NO 1.0

10) Is the sender listed as a
    recipient in a message you
    replied?
 ** NO 1.0

11) Is the sender listed as a
    recipient in a message you
    forwarded?
 ** NO 1.0
```

The system is trying to see if the sender is related to an entity known to the recipient. In this case it checks if the sender has been a recipient of a message that the user has also received. If the user forwarded or replied it the bias will be affected more since those messages and therefore the implied value of the sender were higher than the messages that were simply stored. Anne Hunter and the user haven't received the same message.

```
12) Do you have the sender
    listed on your explicit
    address book?
 ** NO 1.0
```

Having the sender on the user's address book indicates that the recipient explicitly knows and interacts with the sender. Our user knows Anne Hunter but doesn't have her in his address book.

```
13) Does the sender belong to a
    blacklisted domain?
 ** NO 1.0

14) Is the sender explicitly
    blacklisted?
 ** NO 1.0
```

The system is checking if the sender is blacklisted. Ideally, this should have been checked initially since this is a question that we can use to infer message or user relevance with absolute certainty. While some may have blacklisted her, our user has not blacklisted the sender.

```
15) When did you last initiate a
    message to the sender?
 ** more than a month 1.0

16) How many messages have you
    initiated to the sender?
 ** 0-10 1.0
```

The system is trying to discover the relevance of the sender's presence in the recipients Sent Folder. Determined in a similar manner to that done with the Inbox. The user in this case had sent Anne Hunter a single e-mail over a month ago.

```
17) Does the subject line
    contain FRW and Please?
 ** NO 1.0

18) Does the subject line
    contain Free and XXX?
 ** NO 1.0
```

The system's rudimentary modeling of pattern matching techniques used to determine subject-line relevance. Used only to show where existing approaches can fit in within our framework.

```
19) Does this message belong to
    a thread which you have
    participated in (did you
    reply to a message in this
    thread)?
 ** NO 1.0

20) Does this message belong to
    a thread you have stored?
 ** NO 1.0
```

The system's using knowledge of threads to determine subject relevance. A thread that is stored has less value than a thread that was replied to and this is reflected in the way the bias is affected. Our user hadn't participated in any threads with Anne Hunter.

```
21) Is the message addressed
    just to you?
 ** NO 1.0

22) Are the majority of the
    recipients of this message
    relevant to you?
 ** NO 1.0

23) Is this message addressed to
    a group that you explicitly
    subscribed to?
 ** YES 1.0
```

The system is trying to obtain extra information about message relevance by looking at what other recipients received this message and how relevant they are to the user. Can use the component of knowledge used to determine user relevance to determine the relevance of each recipient. Not modeled explicitly in this implementation. When

our user declared Course 6 as his major, he explicitly subscribed to the joblist.

Output

```
Message Relevance : Yes (0.876)
Message Relevance : No  (0.422)
```

This implies that the message is somewhat relevant, which for the given user was a reasonable assessment.

System Knowledge

In order to achieve the functionality described above, it was necessary to build a body of knowledge in to the system. This knowledge was our formulation of what we believe is the thought process involved when human users attempt to identify Spam. As such our knowledge base was modeled as a directed graph of attributes with the root node being that of attribute MessageRelevance as shown in Figure 1. This attribute characterizes the level of relevance an e-mail message has to its recipient based on the other knowledge encoded in our system.

The manner in which UserRelevance is modeled is shown in Figure 2.

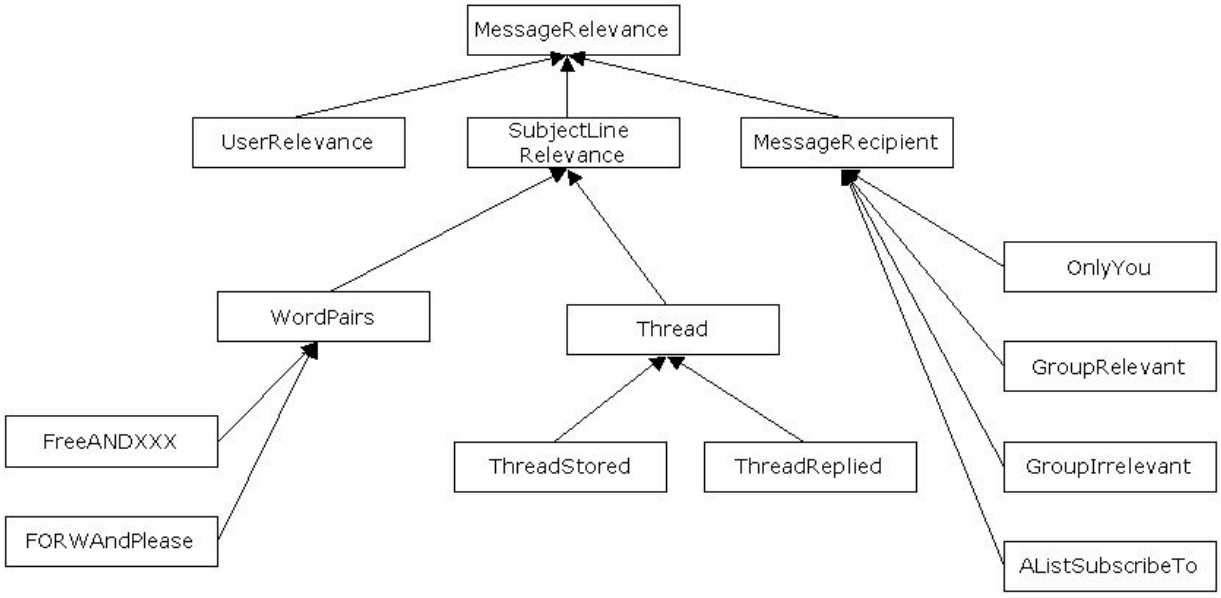

Figure 1. Top-level Knowledge Structure

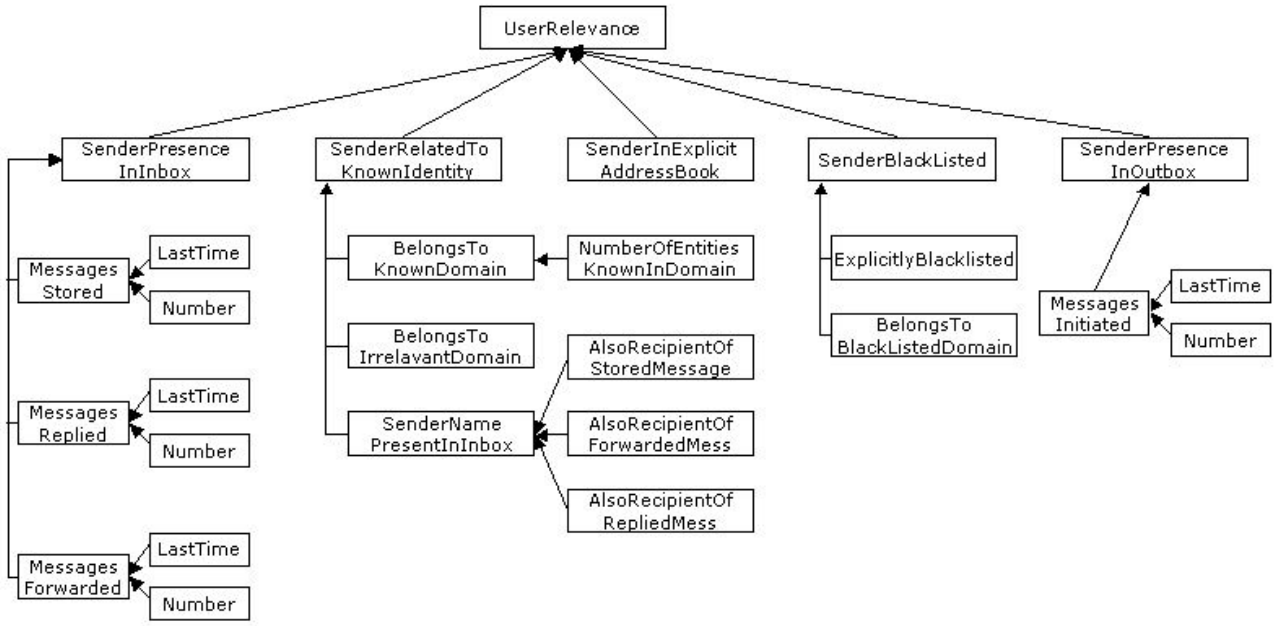

Figure 2. Knowledge structure used to infer sender/user relevance

The knowledge structure described in Figure 2 is also used to ascertain the nature of a group of recipients of the same e-mail message. A group that is relevant to the user would consist of more recipients who are relevant to you (as identified by the UserRelevance attribute) than those who aren't and the opposite would apply to a group that is irrelevant to you.

How does it work?

SpamWallah works on the initial premise that our high level intuition regarding spam is correct. Given this, we attempt to model the thought process a human user would undertake to determine where in the matrix a given message would fit. To free us from implementation issues and allow us to focus purely on the knowledge, the entire system was modeled and built using Grass – a tool that allows the modeling of chaining on rules.

While the knowledge required for this was modeled as described in Figures 1 & 2, the process of obtaining information about these attributes was achieved by setting up rules that relate these attributes. The rules used were discovered by querying a small focus group of e-mail users about their personal habits in classifying spam. This seemed like a natural representation once we found that across our focus group, users tended to gather information about these attributes by asking themselves a set of questions and chaining on the resulting answers. For example, in Figure 1 the attributes on the first and second levels are related using the following rules:

```
If
 [0] SenderRelevant = "YES"
 [1] SubjectLineRelevance = "YES"
Then
 MessageRelevance = "YES" 1.0
 MessageRelevance = "NO" 0.0

If
 [0] SenderRelevant = "YES"
 [1] SubjectLineRelevance = "NO"
Then
 MessageRelevance = "YES" 0.3
 MessageRelevance = "NO" 0.3
If
 [0] SenderRelevant = "NO"
 [1] SubjectLineRelevance = "YES"
Then
 MessageRelevance = "YES" 0.3
 MessageRelevance = "NO" 0.3

If
 [0] SenderRelevant = "NO"
 [1] SubjectLineRelevance = "NO"
Then
 MessageRelevance = "YES" 0.0
 MessageRelevance = "NO" 1.0
```

These rules capture what the members in the focus group did at the very highest level in determining if an e-mail message is spam or not.

Firstly, if the sender is known and the subject line is relevant, then the message is also likely to be relevant. Secondly, if the sender is known yet the subject line isn't relevant the user usually allows the attribute that is dominant affect the bias. This is also true for the case where the sender is unknown and yet the subject seems relevant. Thirdly, if both the sender is unknown and the subject line seems irrelevant then it is likely the message is spam. However, in such cases users also tend to look at the list of recipients to try and garner further information about the value of the message. We model this by allowing the nature or relevance of the message recipients to shift the bias in such a case.

With the entire set of rules encoding the knowledge we captured from our focus group and relating the attributes we identified in place, the system worked simply by backward chaining to the MessageRelevance attribute and collecting positive evidence both supporting the notion that that the message is relevant and also against it. The complete set of rules we used is attached as Appendix A.

Range of Problems Solved

Given adequate information, SpamWallah does a good job of classifying e-mails. We have run SpamWallah on examples of e-mails that fall into all 4 quadrants of the conceptual framework outlined in the section describing our high level intuition about Spam. The system results are reasonable in all cases.

Mail from Parent
We used an e-mail sent recently by our test user's mother as a sample. The user's mother e-mails him regularly from a Hotmail account. He stores the majority of e-mails he receives, and also replies these messages and regularly initiates new messages to her. They are rarely joint recipients of the same e-mail. Furthermore, the message under consideration is part of a thread that discusses plans for commencement.

SpamWallah returns (0.997, 0.474)[1] for MessageRelevance. The SenderRelevance in this case has values (0.948, 0.113)[1] while SubjectLineRelevance returns (0.994,0.482)[1]. Therefore, the system identifies the sender to be extremely relevant and we also get a bias towards relevance in the message content inferred from the subject line. Since both these values put the message on the top left quadrant of our conceptual framework. SpamWallah has correctly identified it as a relevant piece of mail for the recipient.

Spam mail from Parent
In this case we use the same two individuals mentioned in the case above. However, the message we consider is a chain letter forwarded to the user by his mother.

SpamWallah returned values (0.661, 0.612)[1] for MessageRelevance. The SenderRelevance in this case still returns (0.948, 0.113)[1] while SubjectLineRelevance returns (0.190, 0.840)[1]. This puts the message in the top right quadrant of our conceptual framework and therefore we see that SpamWallah has identified the mail correctly again. Since both biases are very strong we end up being unable to classify one way or the other.

Blatant Spam
Lets consider an adult entertainment e-mail advertisement sent to a mailing list our test user is unaware of. The message originates from an e-mail address and domain the user has never seen before. However, the sender nor the domain from where the message is originating is not black listed.In this case SpamWallah returns (0.484, 0.945)[1] for MessageRelevance, (0.200, 0.735)[1] for SenderRelevance and (0.190, 0.840)[1] for SubjectLineRelevance. It is apparent that SpamWallah has again classified the message correctly by position it in the bottom right quadrant and identifying it as Spam.

Bursar's Announcement
For our final case we consider an e-mail message sent to our user from the bursar's office regarding a dispute over a charge appearing on the user's account statement. The user has never received a message from the bursar's office before.

[1] Confidence level for (positive evidence for, positive evidence against)

Here SpamWallah returned (0.745, 0.355)[1] for MessageRelevance, (0.200, 0.735)[1] for SenderRelevance and (0.510, 0.000)[1] for SubjectLineRelevance. Here again the system does a good job by placing this message in the bottom left quadrant.

The system may potentially be extended to interface with a commercial anit-viru program to automatically detect and delete e-mail viruses

System Limitations

Firstly, the SpamWallah system is critically dependent on having access to a comprehensive e-mail client. This means that SpamWallah cannot exist as a stand-alone system. Additionally, SpamWallah contains some components that require training. Therefore, SpamWallah cannot be implemented as a drop-in system either.

Secondly, the system is heavily reliant on message headers and since message headers are vulnerable to spoofing, our system would fail in the event we had to analyze messages with spoofed headers. Furthermore, some users tend to leave the subject line blank and this robs the system of a valuable source of information.

Thirdly, while this representation captured sufficient knowledge to let us bias the attributes we chose in the correct direction – towards Spam or non-Spam – we found it very difficult to associate a magnitude of Spamness or non-Spamess with the final confidence level we got. Our confidence levels were based on a relative scale. For example in our model a sender whose e-mail message had been replied by the recipient had a higher confidence level with regards to relevance than one whose e-mail message had simply been stored. While we are confident this is a correct assumption, the actual value we gave for this confidence was somewhat arbitrary with the only requirement being that they conformed to the relative scale. The issue is that one cannot determine with perfect confidence what these values are. Therefore, we believe that the magnitude associated with MessageRelevance is somewhat inaccurate. This was also observable by the fact that a message sent by an author's mother was only found to be 52.3% relevant while a blatant spam advertisement was only found to be -46.1% relevant (ie. 46.1% irrelevant).

Finally, we also found that users tend to follow different paths of reasoning based on the confidence levels of certain attributes. For example, if the UserRelevance was low and the MessageRelevance was also low but not extensively so, the user would move to the MessageRecipient attribute. We found this to be difficult to model in GRASS since it does not allow us to condition based on the confidence level of the premise. Therefore, the level of detail in our model of the reasoning process is somewhat lacking and using a tool that would have allowed this facility would have helped us greatly.

Conclusion

SpamWallah represents a rule-based approach to the task of building an intelligent spam filter. Initial results show the approach to be relatively simple, require less training, and to do a good job of identifying spam. Simplicity is due to the fact that the number of rules in the system is small, and because these rules correspond closely with human approaches to the same problem. We believe that spam determination is a problem in which modeling the human decision process via rules gives significant returns with even a simple set of rules. In fact, we were surprised at the compact knowledge structure of the problem domain. A few well-chosen rules coupled with a large amount of personal data enabled SpamWallah to make reliable guesses.

The critical issue was in discovering the correct questions and how to ask them. Using a focus group helped significantly in this area. The fact that we were able to observe significant similarities in the thought process indicated that we were onto something that was worthy of further investigation. Preliminary results show that the methodology is viable, and that there exists an opportunity to build a fully-fledged commercial system on these principles.

References

[1] Seeking Profits, Internet Companies Alter Privacy Policy – Saul Hansell. New York Times, April 11, 2002.
[2] M. Sahami, S. Dumais, D. Heckerman, E. Horvitz. "A Bayesian approach to filtering junk email," Proc. AAAI Workshop on Learning for Text Categorization, 1998.
[3] W. W. Cohen. Learning rules that classify e-mail. In Papers from the AAAI Spring Symposium on Machine Learning in Information Access, pages 18--25, 1996.
[4] D.Lin, P. Pantel. SpamCop: A Spam Classification & Organization Program, 1998
[5] M. J. Pazzani. Representation of electronic mail filtering profiles: a user study. In Proc. 2000 int. conf. Intelligent user interfaces (IUI'00), pages 202--206, New Orleans, LA, 2000.
[6] I.Androutsopoulos et al. Learning to Filter Spam E-Mail: A Comparison of Naïve Bayesian and a Memory-Based Approach.
[7] J.M.G Hidalgo et al. Combining Text and Heuristics for Cost-Sensitive Spam Filtering, 2000.

Determining Sender Relevance

Rule 0
```
If
 [0] SenderBlackListed = "YES"
Then
 SenderRelevant = "NO" 1.0
 SenderRelevant = "YES" 0.0
```

Rule 1
```
If
 [0] SenderBlackListed = "NO"
Then
 SenderRelevant = "YES" 0.2
```

Rule 10
```
If
 [0] SenderPresenceInInbox = "YES"
Then
 SenderRelevant = "YES" 0.6
```

Rule 11
```
If
 [0] SenderPresenceInInbox = "NO"
Then
 SenderRelevant = "NO" 0.4
```

Rule 12
```
If
 [0] SenderRelatedToKnownIdentity = "YES"
Then
 SenderRelevant = "YES" 0.6
```

Rule 14
```
If
 [0] SenderInAddressBook = "YES"
Then
 SenderRelevant = "YES" 0.6
```

Rule 15
```
If
 [0] SenderInAddressBook = "NO"
Then
 SenderRelevant = "NO" 0.4
```

Rule 18
```
If
 [0] SenderPresenceInOutbox = "YES"
Then
 SenderRelevant = "YES" 0.6
```

Rule 19
```
If
 [0] SenderPresenceInOutbox = "NO"
Then
 SenderRelevant = "NO" 0.4
```

Determining Relevance of Sender Presence in User Inbox
Through properties of the messages stored in the Inbox

Rule 20
```
If
 [0] SenderMessagesStored = "YES"
Then
 SenderPresenceInInbox = "YES" 0.6
```

Rule 21
If
 [0] TimeLastMessageStored = "within a week"
Then
 SenderMessagesStored = "YES" 0.9
 SenderMessagesStored = "NO" 0.1

Rule 22
If
 [0] TimeLastMessageStored = "within a month"
Then
 SenderMessagesStored = "YES" 0.7
 SenderMessagesStored = "NO" 0.2

Rule 23
If
 [0] TimeLastMessageStored = "more than a month"
Then
 SenderMessagesStored = "YES" 0.5
 SenderMessagesStored = "NO" 0.3

Rule 24
If
 [0] TimeLastMessageStored = "never"
Then
 NumOfMsgsStored = "None" 1.0
 SenderMessagesStored = "NO" 1.0
 SenderMessagesStored = "YES" 0.0

Rule 25
If
 [0] NumOfMsgsStored = ">50"
Then
 SenderMessagesStored = "YES" 0.9
 SenderMessagesStored = "NO" 0.1

Rule 26
If
 [0] NumOfMsgsStored = "10-50"
Then
 SenderMessagesStored = "YES" 0.7
 SenderMessagesStored = "NO" 0.2

Rule 27
If
 [0] NumOfMsgsStored = "0-10"
Then
 SenderMessagesStored = "YES" 0.5
 SenderMessagesStored = "NO" 0.3

Rule 28
If
 [0] SenderMessagesStored = "NO"
Then
 SenderPresenceInInbox = "NO" 0.4

Determining Relevance of Sender Presence in User Inbox
Through properties of the messages forwarded in the Inbox

Rule 30
If
 [0] SenderMessagesForwarded = "YES"
Then
 SenderPresenceInInbox = "YES" 0.8

Rule 31
If
 [0] TimeLastMessageForwarded = "within a week"
Then
 SenderMessagesForwarded = "YES" 0.9
 SenderMessagesForwarded = "NO" 0.1

Rule 32
If
 [0] TimeLastMessageForwarded = "within a month"
Then
 SenderMessagesForwarded = "YES" 0.7
 SenderMessagesForwarded = "NO" 0.2

Rule 33
If
 [0] TimeLastMessageForwarded = "more than a month"
Then
 SenderMessagesForwarded = "YES" 0.5
 SenderMessagesForwarded = "NO" 0.3

Rule 34
If
 [0] TimeLastMessageForwarded = "never"
Then
 NumOfMsgsForwarded = "None" 1.0
 SenderMessagesForwarded = "NO" 1.0
 SenderMessagesForwarded = "YES" 0.0

Rule 35
If
 [0] NumOfMsgsForwarded = ">50"
Then
 SenderMessagesForwarded = "YES" 0.9
 SenderMessagesForwarded = "NO" 0.1

Rule 36
If
 [0] NumOfMsgsForwarded = "0-50"
Then
 SenderMessagesForwarded = "YES" 0.7
 SenderMessagesForwarded = "NO" 0.2

Rule 37
If
 [0] NumOfMsgsForwarded = "0-10"
Then
 SenderMessagesForwarded = "YES" 0.5
 SenderMessagesForwarded = "NO" 0.3

Rule 38
If
 [0] SenderMessagesForwarded = "NO"
Then
 SenderPresenceInInbox = "NO" 0.2

Determining Relevance of Sender Presence in User Inbox
Through properties of the messages replied in the Inbox

Rule 40
If
 [0] SenderMessagesReplied = "YES"
Then
 SenderPresenceInInbox = "YES" 0.9

Rule 41
If
 [0] TimeLastMessageReplied = "within a week"
Then
 SenderMessagesReplied = "YES" 0.9
 SenderMessagesReplied = "NO" 0.1

Rule 42
If
 [0] TimeLastMessageReplied = "within a month"
Then
 SenderMessagesReplied = "YES" 0.7
 SenderMessagesReplied = "NO" 0.2

Rule 43
If
 [0] TimeLastMessageReplied = "more than a month"
Then
 SenderMessagesReplied = "YES" 0.5
 SenderMessagesReplied = "NO" 0.3

Rule 44
If
 [0] TimeLastMessageReplied = "never"
Then
 NumOfMsgsReplied = "None" 1.0
 SenderMessagesReplied = "NO" 1.0
 SenderMessagesReplied = "YES" 0.0

Rule 45
If
 [0] NumOfMsgsReplied = ">50"
Then
 SenderMessagesReplied = "YES" 0.9
 SenderMessagesReplied = "NO" 0.1

Rule 46
If
 [0] NumOfMsgsReplied = "10-50"
Then
 SenderMessagesReplied = "YES" 0.7
 SenderMessagesReplied = "NO" 0.2

Rule 47
If
 [0] NumOfMsgsReplied = "0-10"
Then
 SenderMessagesReplied = "YES" 0.5

Rule 48
If
 [0] SenderMessagesReplied = "NO"
Then
 SenderPresenceInInbox = "NO" 0.3

Determining if the sender is related to an entity known to the recipient/user

Rule 50
If
 [0] SenderBelongsToKnownDomain = "YES"
 [1] SenderBelongsToIrrelevantDomain = "NO"
 [2] NumberOfEntitiesKnownInDomain > 25
Then
 SenderRelatedToKnownIdentity = "YES" 0.0
 SenderRelatedToKnownIdentity = "NO" 0.0

Rule 51
If
 [0] SenderRecipientOfStoredMessage = "YES"
Then
 SenderRelatedToKnownIdentity = "YES" 0.8
 SenderRelatedToKnownIdentity = "NO" 0.0

Rule 52
If
 [0] SenderRecipientOfRepliedMessage = "YES"
Then
 SenderRelatedToKnownIdentity = "YES" 0.6
 SenderRelatedToKnownIdentity = "NO" 0.0

Rule 53
If
 [0] SenderRecipientOfForwadedMessage = "YES"
Then
 SenderRelatedToKnownIdentity = "YES" 0.6
 SenderRelatedToKnownIdentity = "NO" 0.0

Rule 54
If
 [0] SenderBelongsToKnownDomain = "YES"
 [1] SenderBelongsToIrrelevantDomain = "NO"
 [2] NumberOfEntitiesKnownInDomain >= 10
 [3] NumberOfEntitiesKnownInDomain <= 25
Then
 SenderRelatedToKnownIdentity = "YES" 0.6
 SenderRelatedToKnownIdentity = "NO" 0.0

Rule 55
If
 [0] SenderBelongsToKnownDomain = "YES"
 [1] SenderBelongsToIrrelevantDomain = "NO"
 [2] NumberOfEntitiesKnownInDomain < 10
 [3] NumberOfEntitiesKnownInDomain >= 0
Then
 SenderRelatedToKnownIdentity = "YES" 0.8
 SenderRelatedToKnownIdentity = "NO" 0.0

Rule 56
If
 [0] SenderBelongsToKnownDomain <> "YES" | SenderBelongsToIrrelevantDomain <> "NO"
 [1] SenderRecipientOfStoredMessage <> "YES"
 [2] SenderRecipientOfForwadedMessage <> "YES"
 [3] SenderRecipientOfRepliedMessage <> "YES"
Then
 SenderRelatedToKnownIdentity = "NO" 0.3
 SenderRelatedToKnownIdentity = "YES" 0.1

Determining if the Sender is Black listed

Rule 60
If
 [0] SenderAddressInBlacklistedDomain = "YES"
Then
 SenderBlackListed = "YES" 1.0
 SenderBlackListed = "NO" 0.0

Rule 61
If
 [0] SenderAddressExplicitlyBlacklisted = "YES"
Then
 SenderBlackListed = "YES" 1.0
 SenderBlackListed = "NO" 0.0

Rule 62
If
 [0] SenderAddressExplicitlyBlacklisted = "NO"
 [1] SenderAddressInBlacklistedDomain = "NO"
Then
 SenderBlackListed = "NO" 1.0
 SenderBlackListed = "YES" 0.0

Determining the relevance of sender preference in the Outbox

Rule 70
If
 [0] TimeLastMessageInitiated = "within a week"
Then
 SenderPresenceInOutbox = "YES" 0.9
 SenderPresenceInOutbox = "NO" 0.0

Rule 71
If
 [0] TimeLastMessageInitiated = "within a month"
Then
 SenderPresenceInOutbox = "YES" 0.7
 SenderPresenceInOutbox = "NO" 0.2

Rule 72
If
 [0] TimeLastMessageInitiated = "more than a month"
Then
 SenderPresenceInOutbox = "YES" 0.5
 SenderPresenceInOutbox = "NO" 0.4

Rule 73
If
 [0] TimeLastMessageInitiated = "never"
Then
 NumOfMsgsInitiated = "none" 1.0
 SenderPresenceInOutbox = "NO" 1.0
 SenderPresenceInOutbox = "YES" 0.0

Rule 75
If
 [0] NumOfMsgsInitiated = "0-10"
Then
 SenderPresenceInOutbox = "YES" 0.5
 SenderPresenceInOutbox = "NO" 0.4

Rule 76
If
 [0] NumOfMsgsInitiated = "10-50"
Then
 SenderPresenceInOutbox = "YES" 0.7
 SenderPresenceInOutbox = "NO" 0.2

Rule 77
If
 [0] NumOfMsgsInitiated = ">50"
Then
 SenderPresenceInOutbox = "YES" 0.9
 SenderPresenceInOutbox = "NO" 0.0

Determining subject line relevance

Rule 80
If
 [0] ForwAndPlease = "YES" | FreeAndXXX = "YES"

```
Then
 SubjectLineRelevance = "YES" 0.1
 SubjectLineRelevance = "NO" 0.6
```

Rule 81
```
If
 [0] BelongsToARepliedThread = "YES"
Then
 SubjectLineRelevance = "NO" 0.1
 SubjectLineRelevance = "YES" 0.8
```

Rule 82
```
If
 [0] BelongsToAStoredThread = "YES"
Then
 SubjectLineRelevance = "YES" 0.6
 SubjectLineRelevance = "NO" 0.2
```

Rule 83
```
If
 [0] FreeAndXXX = "NO"
 [1] ForwAndPlease = "NO"
 [2] BelongsToARepliedThread = "NO"
 [3] BelongsToAStoredThread = "NO"
Then
 SubjectLineRelevance = "YES" 0.3
 SubjectLineRelevance = "NO" 0.0
```

Determining Message Relevance

Rule 91
```
If
 [0] SenderRelevant = "YES"
 [1] SubjectLineRelevance = "YES"
Then
 MessageRelevance = "YES" 1.0
 MessageRelevance = "NO" 0.0
```

Rule 92
```
If
 [0] SubjectLineRelevance = "YES"
 [1] SenderRelevant = "NO"
Then
 MessageRelevance = "YES" 0.3
 MessageRelevance = "NO" 0.3
```

Rule 93
```
If
 [0] SubjectLineRelevance = "NO"
 [1] SenderRelevant = "YES"
Then
 MessageRelevance = "YES" 0.3
 MessageRelevance = "NO" 0.3
```

Rule 94
```
If
 [0] SenderRelevant = "NO"
 [1] SubjectLineRelevance = "NO"
Then
 MessageRelevance = "NO" 1.0
 MessageRelevance = "YES" 0.0
```

Rule 95
```
If
 [0] SenderRelevant = "NO"
 [1] SubjectLineRelevance = "YES"
```

```
 [2] MessageRecipient = "Just you"
Then
 MessageRelevance = "YES" 0.5
 MessageRelevance = "NO" 0.1
```

Rule 96

```
If
 [0] SenderRelevant = "NO"
 [1] SubjectLineRelevance = "YES"
 [2] MessageRecipient = "A group relevant to you"
Then
 MessageRelevance = "YES" 0.4
 MessageRelevance = "NO" 0.2
```

Rule 97

```
If
 [0] SenderRelevant = "NO"
 [1] SubjectLineRelevance = "YES"
 [2] MessageRecipient = "A list you knowingly belong to"
Then
 MessageRelevance = "YES" 0.3
 MessageRelevance = "NO" 0.2
```

Rule 98

```
If
 [0] SenderRelevant = "NO"
 [1] SubjectLineRelevance = "YES"
 [2] MessageRecipient = "A list or group that is not relevant to you"
Then
 MessageRelevance = "NO" 0.7
 MessageRelevance = "YES" 0.2
```

Determining relevance of other message recipients

Rule 100

```
If
 [0] MessageRecipientIsJustYou = "YES"
Then
 MessageRecipient = "Just you" 1.0
```

Rule 101

```
If
 [0] MessageRecipientsMostlyRelevant = "YES"
Then
 MessageRecipient = "A group relevant to you" 1.0
```

Rule 102

```
If
 [0] MessageRecipientDueToExplicitSubscription = "YES"
Then
 MessageRecipient = "A list you knowingly belong to" 1.0
```

Rule 103

```
If
 [0] MessageRecipientDueToExplicitSubscription <> "YES"
 [1] MessageRecipientIsJustYou <> "YES"
 [2] MessageRecipientsMostlyRelevant <> "YES"
Then
 MessageRecipient = "A list or group that is not relevant to you" 1.0
```