Journal of Kerbala University , Vol. 7 No.4 Scientific . 2009

Design and Implementation of Email Filtering Agent by using Clustering Algorithm

تصميم وتنفيذ نظام وكيل لتصفية وترشيح البريد الالكتروني باستخدام الخوارزمية العنقودية

هدى فوزي الشهد / جامعة كربلاء
Huda_msc2006@yahoo.com

Abstract

This paper will describe the system of email filtering agent which utilize clustering algorithm that classify the messages into two lists negative and positive according to the features that extracted from each message. The agent will receive emails of its client, along with a set of examples, both similar and dissimilar, in order to develop a sense of what the user requires in his clustering. The agent has the task to group the emails into folders based on the similarity between them in the form of common subjects, common email address.. etc. . In fact, each user decides his own gold standard; in other words a user himself decides as to what he wants to be considered as similar/dissimilar. For example, someone might want his emails to be foldered based on the month of receipt whereas someone else would want the clustering to be done based on subject and content. List emails provide another such example.

To implement EMFA system, JAVA language was used. It provides a set of abstract classes defining objects that comprise the E-mail system also supporting the creation of sophisticated user interfaces.

الخلاصة //

هذا البحث يصف النظام الوكيل لتصفية البريد الالكتروني بأستخدام الخوارزمية العنقودية التي تسخدم في تقسيم الرس
الالكترونية الى مجموعتين : المجموعة الايجابية والمجموعة السلبية بالاعتماد على الخصائص المسخلصة من كل رسالة. الن
سوف يستلم الرسائل ومجموعه من الامثلة التي تساعد في تقسيم الرسائل بأستخدام الخوارزمية.النظام الوكيل سوف يقسم الرس
الالكترونية الى مجاميع بالاعتماد على التشابه بين عنوان الرسالة أو بالاعتماد على عنوان المرسل...الخ.
كل مستخدم يمكن ان يقرر الخواص المعتمدة لتقسيم الرسائل الخاصة به، مثلا البعض يريد تقسيمها على اساس الشهر والبع
الاخر يريد تقسيمها على اساس المرسل .
اللغه المستخدمة في هذا البحث هي (JAVA) لانها توفر امكانية اكثر في التعامل مع الانترنت.

1. Introduction

E-mail communication has become more available, it is being used as a fundamental communication tool by millions of people around the world. E-mail is used to organize meetings, manage virtual work teams, discuss work-related proposal, make announcements and solve problems. A user may receive tens or even hundreds of emails every day. E-mail users commonly try to manage the large amount of emails they receive by using automatic approach to processing the email which is called email filtering [1].

Email filtering is the processing of e-mail to organize it according to specified criterion. Most often this refers to the automatic processing of incoming messages, but the term also applies to the intervention of human intelligence in addition to artificial intelligence, and to outgoing emails as well as those being received. Email filtering software inputs email and for its output, it might pass the message through unchanged for delivery to the user's mailbox, it might redirect the message for delivery elsewhere, or it might even throw the message away. Some mail filters are able to edit messages during processing [2].

Agent architecture has been developed which observes a user performing tasks, and identifies features which can be used as training data by a learning algorithm. Using the learned profile, an agent can give a device to the user on dealing with new situations. The agent uses sender and recipient field of the message and the keywords from the subject field. Other information such as whether the message has been read, whether it is a reply to a previous message, etc [3].

2. Agent

An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors. A human agent has eyes, ears, other organs for sensors, and has hands, legs, mouth, and other body parts for effectors. A robotic agent substitutes cameras and infrared range finder for the sensors and various motors for the effectors. A generic agent is diagrammed in figure (1).

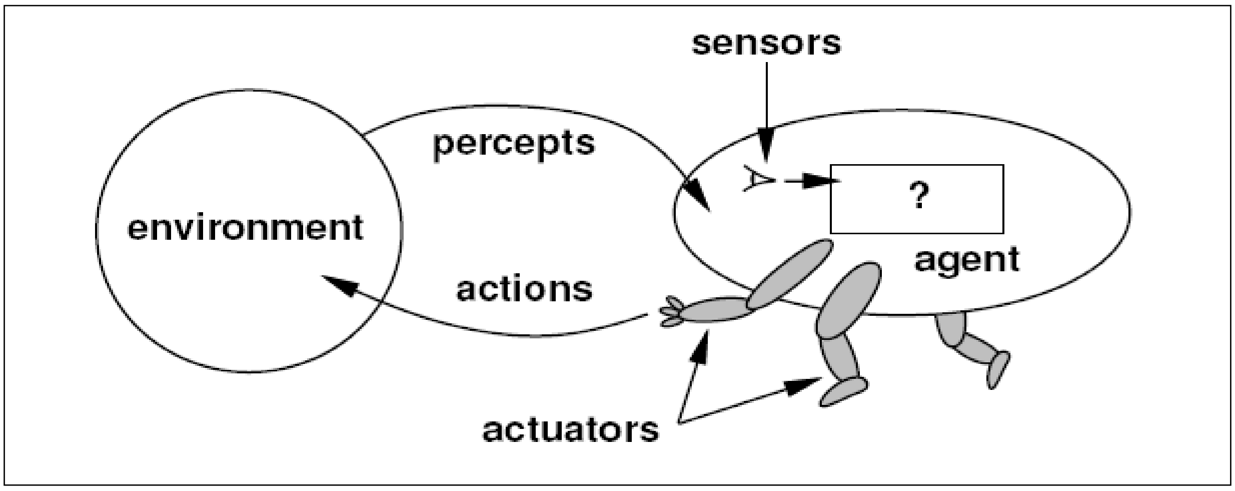


Figure (1) Agent diagram

A software agent should display some of the characteristics that are associate with human intelligence: learning, inference, adaptability, independence, creativity, etc. In other words, a software agent is a program that its users can delegate tasks to, rather than commanding it to perform the tasks[4].

Any software agent must have the following characteristics:

- It must be autonomous, which means that it can operate on its own without a direct human command. Software agents have individual internal States and goals, and act in such a manner as to meet their goals on behalf of their users.
- It must be personalized; that is, it acquires the user's interests and adapts as it evolves over time.
- It must be persistent, either run continuously or save its state, so that the user can see the agent as a stable entity and develop a trust relationship with it.

As a language, Java is ideally suited to the development of software agents. With its built-in support for HTTP communication, agent developers can easily make their agents "Web aware," without the need to write a custom HTTP implementation. This helps agent developers concentrate on the application, without being overly burdened by developing the network code [5].

2.1 Agent Approaches

Agent architectures represent the move from specification to implementation. There are two approaches of the agent:

2.1.1 Classical Approach (Deliberative Architectures)

The term "deliberative agent" seems to have derived from use of the term "deliberate agent" to mean a specific type of symbolic architecture. A deliberative agent or - agent architecture is based on a strong notion of agents and contains an explicitly represented, symbolic model of the world, and decisions (for example about what actions to perform) are made via logical (or at least pseudo-logical) reasoning, based on pattern matching and symbolic manipulation.

In classical AI, symbolic representations provide an interface between the independent information processing units. Reasoning in this systems is realized through logics, mostly second-order module logics.
The idea of deliberative agents based on purely logical reasoning is highly appealing, but there are at least two important problems to be solved:

1. The transduction problem: that of translating the real world into an accurate, adequate symbolic description, in time for that description to be useful.
2. The representation/reasoning problem: that of how to symbolically represent information about complex real world entities and processes (since computers may simulate reasoning but not the interaction with the environment), and how to get agents to reason with this information in time for the results to be useful [6]. This type of agent is used in this paper.

2.1.2 Alternative Approaches (Reactive Architectures)

A Reactive Architecture is an architecture that does not include any kind of central symbolic world model and doesn't use any complex reasoning. The problems associated with symbolic AI have led some researchers to question the viability of the whole paradigm, and to the development of what are generally known as reactive architectures [6].

3.Email Agent

Electronic mail provides an essential communication media for people all over the world. Many people are overloaded with a large number of emails on a regular basis. Users today want the ability to automatically prioritize and organize their e-mail, and in the future, they would like to do even more automatically, such as addressing mail by organizational function rather than by person.

Intelligent agents can facilitate all these functions by allowing mail handling rules to be specified ahead of time, and letting intelligent agents operate on behalf of the user according to those rules. Usually it is also possible (or at least it will be) to have agents deduce these rules by observing a user's behavior and trying to find patterns in it [7].

Messages are exchanged between hosts using the SMTP (Simple Mail Transfer Protocol) with software programs called mail transport agents. Users can download their messages from servers with standard protocols such as the POP (Post Office Protocol).IMAP (Internet Message Access Protocol) provides the user more capabilities for retaining e-mail on the server and for organizing it in folders on the server [2].

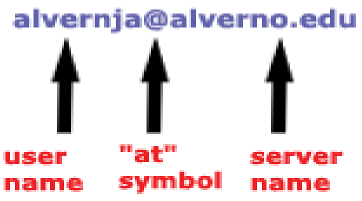
3.1 Email Message Format

Internet e-mail messages consist of two major components:
Headers: Message summary, sender, receiver, and other information about the e-mail.
Body: The message itself, usually containing a signature block at the end.
The headers usually have at least four fields [8]:
1.From: The e-mail address of the sender of the message.
2. To: The e-mail address of the receiver of the message.
3. Subject: A brief summary of the contents of the message.
4. Date: The local time and date when the message was originally sent.

An email address is unique, just like a Post Office street, city, state and zip address. Email addresses have two parts:

- The user name
- The email server or host address

The host address is similar to a post office address. When you send mail to someone in another city, the address on the envelope is read and that piece of mail is directed to a post office for delivery. That is the purpose of the host or email server. The user name is separated from the host address by the @ (pronounced "at") sign. Each user name is unique to a particular host address [2]. For example Jane A. Alverno's email address at Alverno would be:



Other common header fields include:

1. Cc: Carbon copy (because typewriters use carbon paper to make copies of letters). It's contains the addresses of others who are to receive the message, though the content of the message may not be directed at them.

2. Bcc: Blind carbon copy (the recipient of this copy will know who was in the **To** field, but the recipients cannot see who is on the Bcc: list).

It contains addresses of recipients of the message whose addresses are not to be revealed to other recipients of the message [8].

3. Received: Tracking information generated by mail servers that have previously handled a message.

4. Content-Type: Information about how the message has to be displayed, usually a MIME type [9].

4. Email Filtering

The term "mail filtering" is used in different contexts and requires some discussion before the meaning can be cleared. Mail filters, or, sets of rules that users put together to file incoming e-mail into different mailboxes or folders. It is call personal mail filtering because it pertains directly to a single person's organizational preferences. As e-mail use has grown, some regularity has come about the sort of e-mail that appears in users' mail boxes. In particular, un-solicited e-mail, such as "make money fast" schemes, chain letters and porn advertisements, is becoming all too common. Filtering out such unwanted trash is known as junk mail filtering [10].

There are three types of filtering [11]:

- ***Cognitive Filtering:*** this characterizes a message by the contents and meaning of the message. Participants in the survey looked for certain keywords or phrases to classify messages.
- ***Social Filtering:*** this complements the cognitive approach by concentrating on the personal and organizational interrelationships between sender and receiver. For example, more attention may be given to messages from a superior such as a supervisor.
- ***Economic Filtering:*** this is based on a cost-benefit assessment of a message, such as the length of a message.

Social filtering is used in this paper.

5. Clustering algorithm

The objective of the project is to build a module for an office agent that clusters emails into folders. The primary question that arises is whether an agent can do the above since such a clustering (as any other clustering operation) requires a metric of similarity/dissimilarity as the centric operator. However, there is no gold standard available in such a scenario. In fact, each user decides his own gold standard; in other words a user himself decides as to what he wants to be considered as similar/dissimilar. For example, someone might want his emails to be foldered based on the month of receipt whereas someone else would want the clustering to be done based on subject and content. List emails provide another such example.

As input the agent shall receive emails of its client, along with a set of examples, both similar and dissimilar, in order to develop a sense of what the user requires in his clustering. The agent has the task to group the emails into folders based on the similarity between them in the form of common subjects, common email threads etc. The goal is to design a learning strategy for the agent to learn how to cluster [12].

Clustering is the algorithm of organizing objects into groups whose members are similar in some way. The designing of clustering algorithm is to choose how to classify emails. It represents each email using both header features that include the subject line, email addresses, and date of sending message.

A clustering algorithm creates a set of rules for each class in a concept. Each rule covers many examples of the class in question (*positive* examples), yet selects examples of other classes (*negative* examples). According to these examples, the messages will be filled into two lists, *Negative* and *Positive* lists. The *Negative* list contains messages that have the date which is specified by the user under specific condition and the *Positive* list contains messages that will be filtered by using filtering architecture after extracting other features from it, see figure (2) and algorithm (1) illustrates the steps of clustering algorithm.

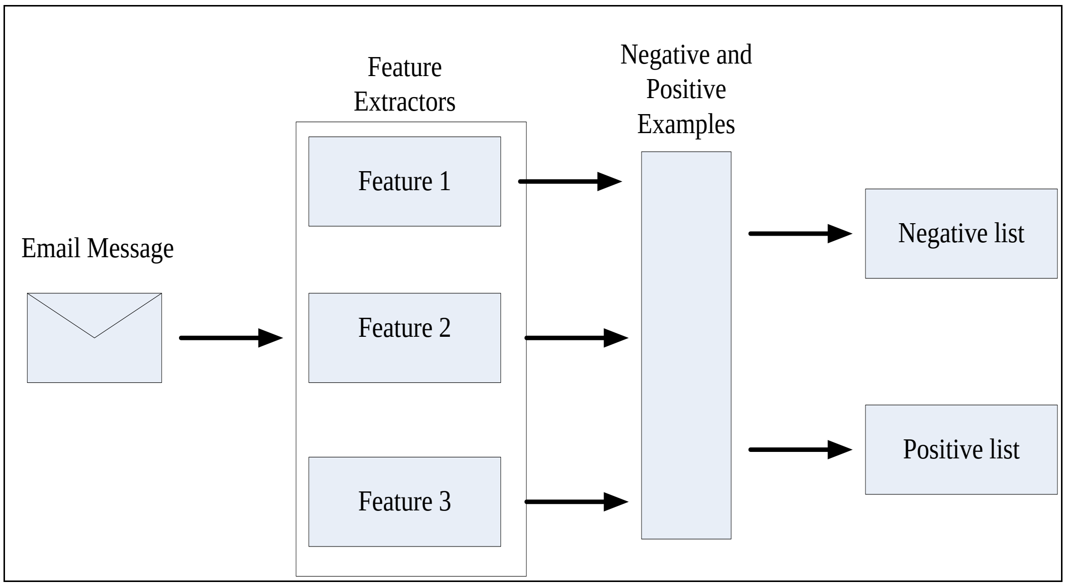


Figure (2) Clustering architecture

The agent should be able to classify the incoming email message into folders (*negative* and *positive*) according to the features that were extracted from the message headers. The header contains information about who was the sender, time and date of sending, subject of the message, recipients, etc. The functions *getFrom*, *getSubject* and *getDate* are used to extract the features: *From* field, *Subject* field and *Date* field from the headers, algorithm (2) illustrates the steps of features extraction.

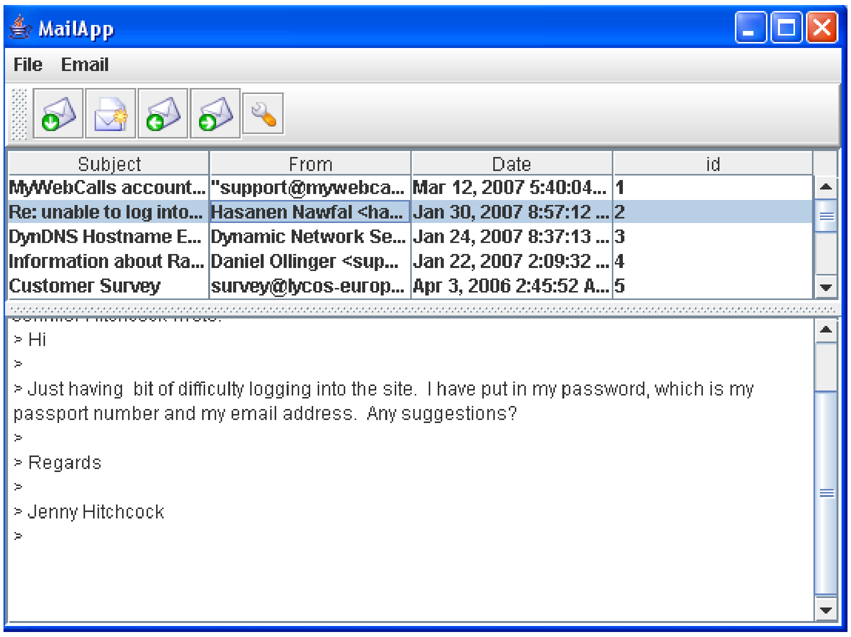
```
Algorithm (1) Clustering
Input: Message Date, Subject and From
Output: negative and positive lists
Begin of clustering algorithm
Step1: Define two arrays: positive and negative.
Step2: Fetch the Date,From and Subject fields from arrays.
Step3: Split the day, month and year from the Date field.
Step4: Check the month with a specific value that the user was entered.
Step5: If the year is less than the predefine value then consider the message as negative and save it in the negative array.
Step6: If the month is larger than the value then consider the message aspositive and save it in the positive array.
Step7:Check the From field if the address is in the list or not .
Step8:Check the Subject field if it correct or not.
Step9: Increment the counter.
Step10: Check the counter with the length of array, if it is less than the number of messages in the array, If  No then fetch another fields, If Yes then close the array.
End of clustering algorithm
```

```
Algorithm (2) Feature extraction
Input: Message
Output: Date, Subject and From
Beginof feature algorithm
Step1: Define three arrays for the features that extracted from messages.
Step2: Open Inbox for read _write folder.
Step3: Save the number of messages in a counter.
Step4: Fetch a message from Inbox.
Step5: Extract the Subject field, From field and Date field from the header of the message and save them in the arrays.
Step6: Increment the counter.
Step7: Check the counter if it is less than the number of messages in the Inbox, If  No then fetch another message, If Yes then close the Inbox.
End of feature algorithm
```

6. Implementation of EMFA system

EMFA system is designed with many frames to enable the agent to filter the messages stored in the Inbox. The first frame that appears on the screen is the main frame that contains the inbox and the messages header. The header of the messages is divided into four fields: *Subject*, *From*, *Date* and *ID* (that is a unique number associated with each message).

When any message is selected, the body of the message appears in the same frame as shown in figure (3).



Subject	From	Date	id
MyWebCalls account...	"support@mywebca...	Mar 12, 2007 5:40:04...	1
Re: unable to log into...	Hasanen Nawfal <ha...	Jan 30, 2007 8:57:12 ...	2
DynDNS Hostname E...	Dynamic Network Se...	Jan 24, 2007 8:37:13 ...	3
Information about Ra...	Daniel Ollinger <sup...	Jan 22, 2007 2:09:32 ...	4
Customer Survey	survey@lycos-europ...	Apr 3, 2006 2:45:52 A...	5

```
> Hi
>
> Just having  bit of difficulty logging into the site.  I have put in my password, which is my
passport number and my email address.  Any suggestions?
>
> Regards
>
> Jenny Hitchcock
>
```

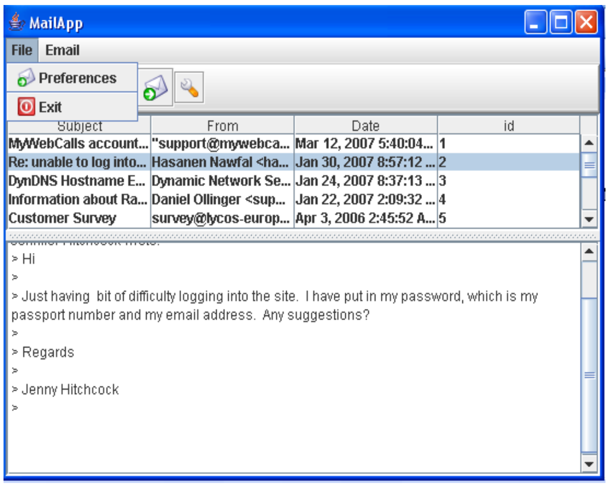
Figure (3) EMFA system

6.1 EMFA frame

The EMFA frame contains a menu of two options *File* and *Email*, and those options are illustrated below:

1-**File** option: This option contains two items, *Preferences* and *Exit* as shown in figure (4).

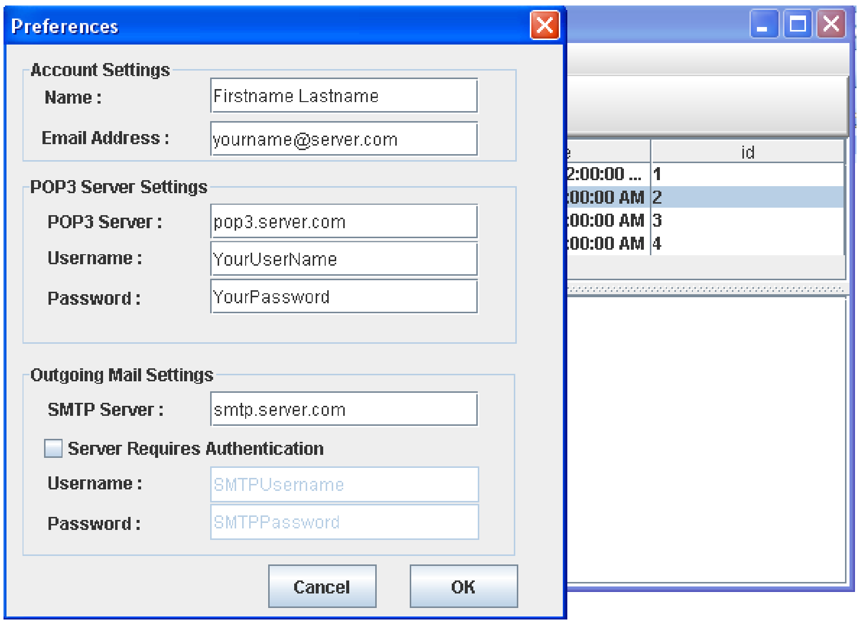
- **Preferences**: It contains three fields, *Account Settings*, *POP3 Server Settings* and *Outgoing mail Setting*. These fields should be filled correctly so that the user can have access to his account.
- The *Account Settings* field should be filled with the name and the Email address of the user. Then the username and password must be entered in the *POP3 Server Settings* fields. If the server requires authentication, the username and password of server must be entered in the *Outgoing Mail Settings* as shown in figure (5).
- **Exit**: this button ends the execution of the monitoring system.



Subject	From	Date	id
MyWebCalls account...	"support@mywebca...	Mar 12, 2007 5:40:04...	1
Re: unable to log into...	Hasanen Nawfal <ha...	Jan 30, 2007 8:57:12 ...	2
DynDNS Hostname E...	Dynamic Network Se...	Jan 24, 2007 8:37:13 ...	3
Information about Ra...	Daniel Ollinger <sup...	Jan 22, 2007 2:09:32 ...	4
Customer Survey	survey@lycos-europ...	Apr 3, 2006 2:45:52 A...	5

```
> Hi
>
> Just having  bit of difficulty logging into the site.  I have put in my password, which is my
passport number and my email address.  Any suggestions?
>
> Regards
>
> Jenny Hitchcock
>
```

Figure (4) EMFA frame



Account Settings

Name :	Firstname Lastname
Email Address :	yourname@server.com

POP3 Server Settings

POP3 Server :	pop3.server.com
Username :	YourUserName
Password :	YourPassword

Outgoing Mail Settings

SMTP Server :	smtp.server.com

☐ Server Requires Authentication

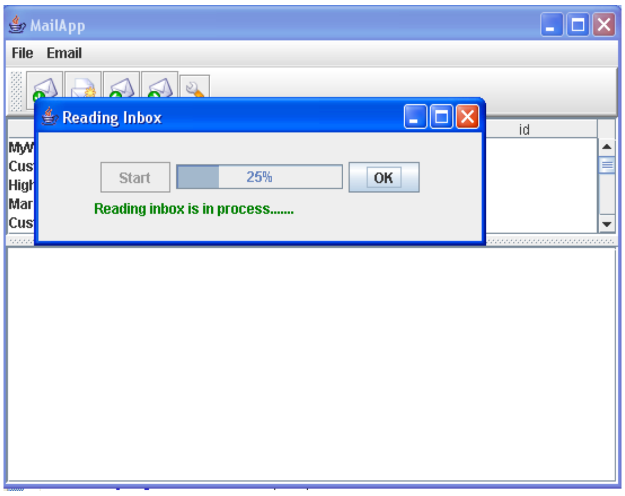
Username :	SMTPUsername
Password :	SMTPPassword

Figure (5) Preference frame

2- **Email** option: this option contains many items, Compose, Reply, Forward, Delete and Agent.

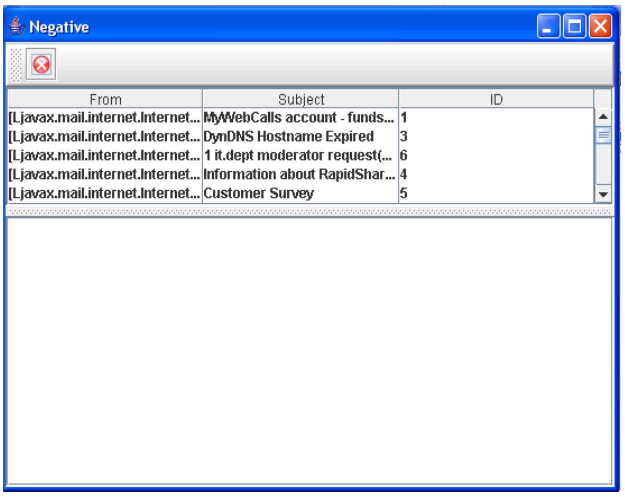
When an agent button is clicked, then the program will be run automatically. A reading Inbox frame will appear that contains the progress bar, Start and OK buttons. If start button is pressed, then progress bar will indicate the progress of reading the messages in the inbox one by one as shown in figure (6).
When the reading inbox process is done, a new report window will appear reporting the number of messages in the inbox, then agent will apply clustering algorithm on all of the messages the agent will extract the features from messages such as *Subject*, *From* and *Date*.

When extraction is finished, messages will be divided into two lists, Positive list and Negative list according to the extracted features. These lists will be shown in figure (7) and figure (8).



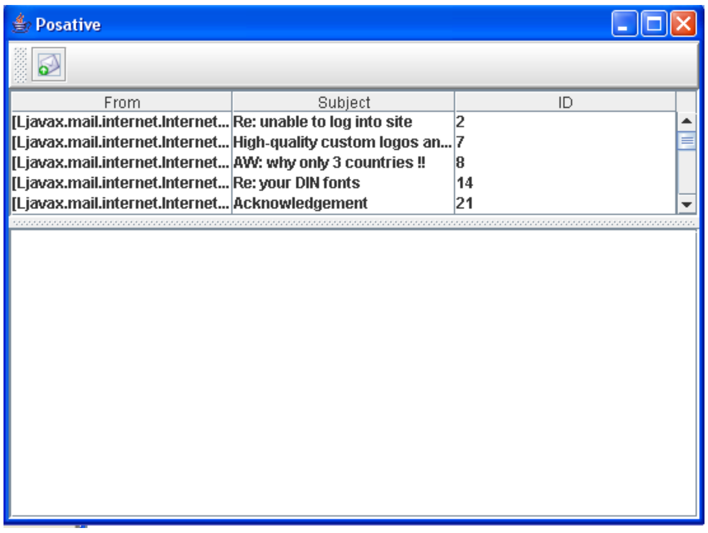
Reading inbox is in process.......

Figure (6) Reading Inbox frame



From	Subject	ID
[Ljavax.mail.internet.Internet...	MyWebCalls account - funds...	1
[Ljavax.mail.internet.Internet...	DynDNS Hostname Expired	3
[Ljavax.mail.internet.Internet...	1 it.dept moderator request(...	6
[Ljavax.mail.internet.Internet...	Information about RapidShar...	4
[Ljavax.mail.internet.Internet...	Customer Survey	5

Figure (7) Negative Lists



From	Subject	ID
[Ljavax.mail.internet.Internet...	Re: unable to log into site	2
[Ljavax.mail.internet.Internet...	High-quality custom logos an...	7
[Ljavax.mail.internet.Internet...	AW: why only 3 countries !!	8
[Ljavax.mail.internet.Internet...	Re: your DIN fonts	14
[Ljavax.mail.internet.Internet...	Acknowledgement	21

Figure (8) Positive Lists

7. Conclusions and Future Work

The EMFA system makes it easier for the user to use the E-mail rather than the normal system. The agent will assist the user in opening the inbox, siplitting the messages into two groups: negative and posative by using the features that extracted from the messages that will facilitate the comparison between messages' header and the predefined conditions that created by the user. In the future, the suggestion to dealing with inbox with a large number of messages it is recommended to use fuzzy to make an efficient classification for the messages so the agent will work in a better way in such cases and the agent may perform extra tasks in sorting the incoming messages by distributing them into folders with different categories such as folders for messages containing pictures or attachment document…etc.Can use the messages into posative array to replay and forword or deleting the messages in a negative array.

8.Reference

[1] Shen, D., Zhang, B. and Chen, Z., “Adding Semantics to Email Clustering”, Sixth International Conference on Data Mining, 2006.

[2] Wikipedia, “Email”, The free encyclopedia, last modified 20-May-2007.

[3] Payne, T., “Experience with Rule Induction and K-nearest Neighbor Methods for Interface Agents that learn”, IEEE, Transformations on Knowledge and Date Engineering, Vol.9, No.2, 1997.

[4] Hui, Y., “Keyphrase-Based Information Sharing In Multi- Agent System”, Master thesis, 2000.

[5] David, R. and Reilly, M., “JAVA Netwark Programing and Distributed Computing”, 2002. "http://www.davidreilly.com/jnpbook"

[6] Wooldridge, M. and Jennings, N., R.,” Intelligent Agents: Theory and Practice”, Research supported by UK Science and engineering Research ,1998.

[7] Hermans, B., “Intelligent Software Agents on the Internet”, First Monday Vol.2, No. 3, 1997.

[8] Resnick, P.,”Internet Message Format”, RFC2822, Network working group, 2001.

[9] Crocker, D., H., “Standard for the format of ARPA Netwark Text Message (1), RFC 733, 1997.

[10] Rennie, J., D., “ifile: An Application of Machine Learning to E-MailFiltering”, Text Mining Workshop, 2000.

[11] Payne, T.,” Learning Email Filtering Rules with Magi”, A mail agent Interface, Master Thesis 1994.

[12] Saha, M. and Wadhera, G., “Learning a Distance Metric to cluster E-Mails”, 2003.