Text Mining Agent Using Hybrid Neural Networks

Saja A. Muhammed[1*]   Laith R. Flaih[2]

1. Computer Science Department, University Of Salahaddin, Erbil, Iraq
2. Computer Science Department, Cihan University

* E-mail of the corresponding author: **sajaatta@yahoo.com**
* E-mail of the corresponding author: **d.lath1974@yahoo.com**

Abstract:

With the development of internet services and Electronic Mail communication, number of spam, advertise, or unwanted E-mails have been grown dramatically, tens of such annoying and time consuming E-mails arrives mail boxes every day and this issue becomes more critical with the availability of the internet services for children these days.

The system proposed in this paper tries to solve this problem using intelligent software agent that checks each incoming and outgoing messages and blocks unwanted messages or replace the undesired words. The proposed system also offers creating dynamic number of agents according the users desire by creating dynamic number of rules that satisfies user's requirements and objectives.

keywords: Agent, Text mining, Intelligent Agent, Electronic Mail

1. Introduction:

ne of the most popular communication means is the Electronic Mail, due to E-mails availability, flexibility, speed and cost E-mail services has developed and grown in a wide range and all around the world. Beside the original purpose that the E-mails are created for it has been used for advertising and marketing purposes by companies, also it may be used by others to broadcast viruses, illegal or harmful contents. Other times these E-mails may contain critical contents that may be not suitable for kids [1].

For all the reasons mentioned above filtering and checking E-mails has become one of the important research fields. Many E-Mail filtering and checking efforts done till today using different techniques and algorithms based on E-mail content, tags and URL or E-mail address.

Many methods have been used to solve this problem; one common method is using filtering agents. Agents are software programs designed to perform specific task, agent's task in the proposed system is to receive E-mail content and looks for unsuitable words to replace them or to block the whole message according the agent's rules and the desire of the agent's designer.

Beside checking and filtering the system allows the user to create and design new agent that meets the requirements and objectives of its designer [2][3].

2. Agent:

Computer systems that are designed to perform tasks deliberated by its designer, these systems are placed in some environments (real world for physical agents and software environments for soft bots or software agents). It performs its tasks autonomously without human intervention by sensing the environment through physical sensors or software sensors and performing action on that environment through actuators [4]. Figure 1 shows the agent in its environment.

Today, agents are used in wide diversity of applications and fields starting from the simple one that represents mail filtering to solving problems of open and complex systems some of these applications are: Industrial applications(process control and manufacturing), Commercial applications like information management(information gathering and filtering), Medical applications(health care and patient monitoring), Entertainment applications (games, interactive theater and cinema), and many other applications that are growing day after day [5].

3. Intelligent agents:

Intelligent or autonomous agents are computer systems that acts in a flexible autonomous manner which means that the agent must be able to respond to the changes that happens in its environment in timely fashion which means the agent must be reactive, beside that it must has social ability that enables the agent to communicate and interact with other agents, furthermore; the intelligent agent should be able to exhibit goal directed behavior.

Not every single designed agent is characterized by intelligent property, there are differences between an ordinary agents and the intelligent one. One of the most important differences is that ordinary agents are limited by the rules that its designer created for that particular agent to fulfill the objectives that the agent is designed for, and changes in its environment will not cause any changes in the actions that the agent perform or in the agent's rules.

For example simple reflex or table driven agents that maps percepts (environment status) to actions, for each percept there is a specified action or course of actions that must be performed by that particular agent [6].

Intelligent agent keeps track changes in the surrounding environment to update its internal state and tries to adapt to new situation and learns from its experience then generates new course of actions according or that satisfies the requirements of new situation, such agents works successfully in dynamic non deterministic environments that changes over time in which the ordinary agent if placed in will dramatically fails.(ex: utility based and learning agent as shown in figure 2)[6].
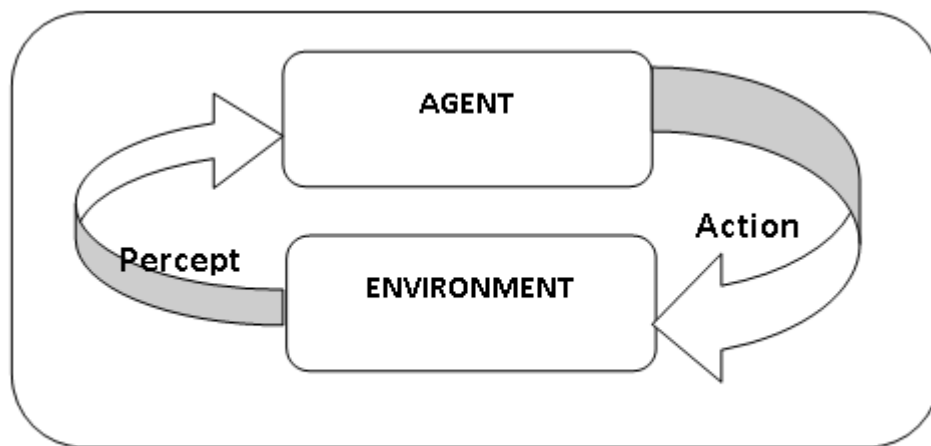

Figure 1. An agent in its environment[3]
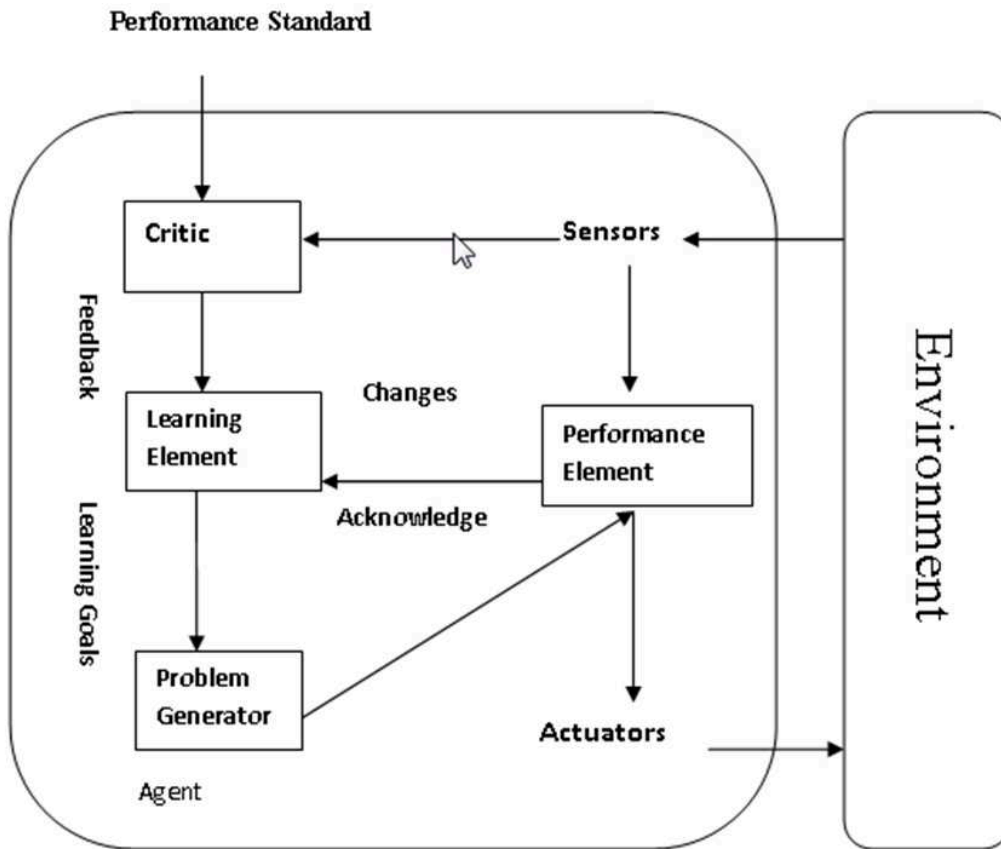
Figure 2.Learning Agent[2]

4. The Proposed System:

In this section we convey the components of the system proposed, the block diagram below shows the main components of the system as shown below in figure 3.
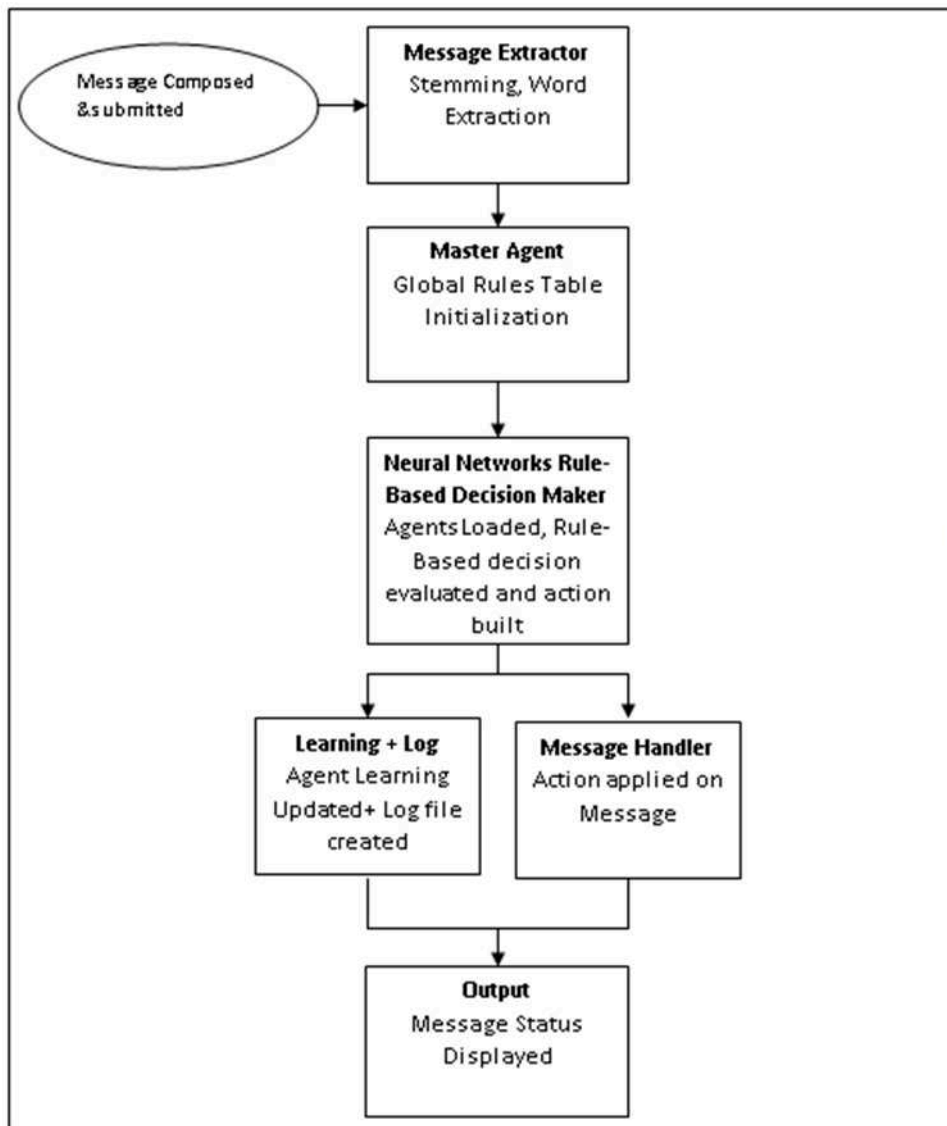
Figure 3. Text Minig Agent Block Diagram

After the message has been composed by the user and submitted to the system. Message Extractor will handle the task of message decomposing and analyzing, it checks the submitted message starting from its subject and identifies its beginning and the end of the message then it will be read word by word, the Message Extractor will continue the task of message stemming by reducing each word to its original word for example: playing, played, player, played will be reduced to its root or original word which is "play" and so on, then it starts word extraction and retrieves table or array that contains the set of words that the message is composed of.

The extracted words and the list of agents which consists of the set of all agents will be delivered to the Master Agent which in turn initializes the global rules and according to these rules the master agent will perform the task of distributing the incoming E-mails to the most appropriate available agents after determining the active agent from the non ones. The user can invoke or use more than one agent at a time.

After that the Master Agent has determined the active agent set, organized and distributed the incoming E-mails the Neural Network Rule-Based Decision Maker will initialize the local rules for each agent which represents the core of each agent, when an agent receives an incoming E-mail from the master agent it depends on these rules to decide wither to block, filter out (replacing specific undesired words) or simply pass it to its destination. This action will be done by matching the extracted word table against these local rules if the rules match or simply if

the message contains specific block or undesired words then appropriate action will be performed on that message.

When the Neural Network Rule-Based Decision Maker finishes its task and the action to be performed on the delivered E-mail has been decided by the chosen agent, the Learning rate for that agent will be increased with each block or word replacing action, then a log file will be created, the file size and content will be updated continuously with each taken action by that agent and the new replaced or blocked word will be added to the file which in result will decreases the time needed for decision making in future.

The final component which is the Message Handler will specify the action applied on the message wither blocking or filtering. According to the relevant agent's rules the message will either be blocked or specific words will be replaced and message status will be displayed as Output.

4.1. Message Composer:

When the user composes a new message and submitted to the system or send it to another system user the Message Composer checks the submitted message for its emptiness, if the message was empty it will wait for the next submitted message otherwise the message will pass through a **Preprocesses Phase** that is performed by the **Message Cleaner** which composes the below processing steps:

4.1.1. Lexical analysis:

It's the process of converting the stream of characters into a stream of words. The objective of lexical text analysis is to identify the words in text document, where the stream of characters is given as input to the compiler it scans that from left to right and produces a stream of words.

4.1.2 String tokenize:

Here each word is represented as token. After tokenizing each word it becomes easy to identify words in that plain text. Preprocessing technique is implemented to decrease the storage space of document.

4.1.3 Stop word elimination:

In this we are going to define a group of stop words in a list such as (the, which, it, at, in and etc.), then we remove those words which can decrease the space of the document to store in a database. If we take prepositions like to, the, and, into, etc. these words doesn't belong to the undesired word set that the system is designed to eliminate or replace. That's why these words have been removed to enhance the search and decision process time.

4.1.4 Stemming:

Stemming is a technique that is used to reduce the words to its grammatical root. This kind of retrieving data or information is applications of data mining. For example Words like comparing, compare, compared, compares have been rooted to the word compare so that it will useful during retrieving information. Stemming process is applied to eliminate suffixes like ed, ing, ily etc. by removing these suffixes we reduced the number of words in each document and the space needed by each document to be stored, space reducing is not the only benefit of stemming process but it's also necessary to reduce the document complexity and information retrieval more efficient in text analysis.

After the preprocessing phase the message will be checked again for its validity, if it wasn't valid, the message will be sent back to the Message Cleaner until it becomes clean and valid, then the word extractor starts the process of word extraction to extract individual words and finally it produces words array that will be sent to the Master Agent. Figure 4 shows the block digram of Message Composer.
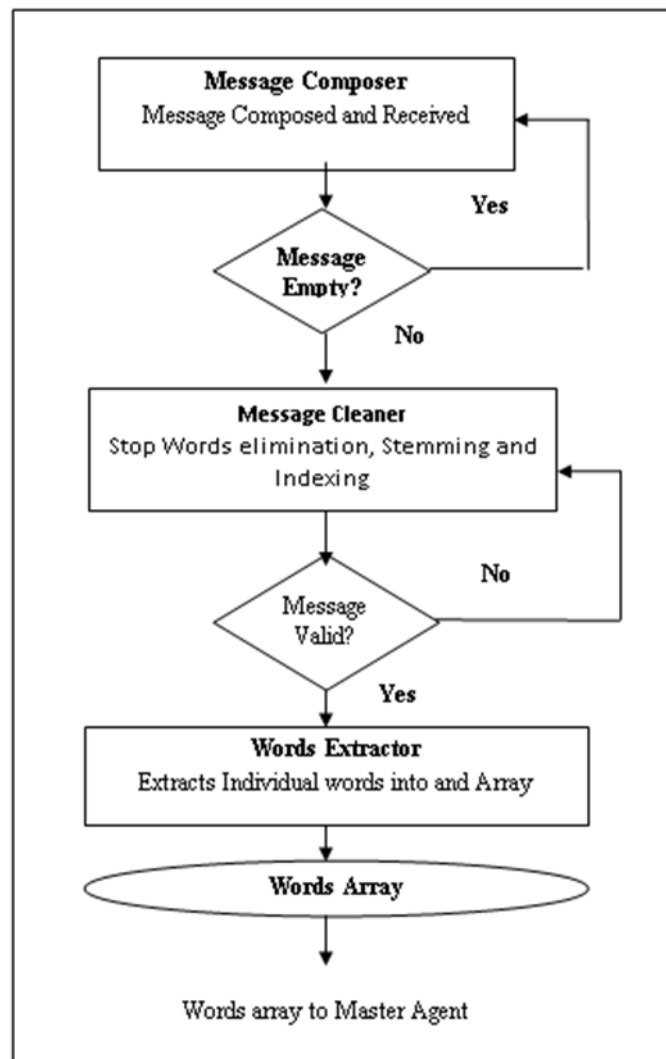
Figure 4: Message Composer Diagram

4.2.   Agent Loader:

The Agent Loader populates rules table for agents then it starts to count the rules if there were no rules it invokes the Agent Loader again until rules count is greater than zero. The populated rules by the Agent Loader will be filtered by the Rule Filter to eliminate rules from inactive agents. Set of relevant agents will be initialized by the Agents Loader; this set will be delivered with the set of extracted words to the Neural Network Decision Maker to decide which relevant agent will handle the task of filtering the incoming E-mail. Figure 5 shows the block diagram of Agent Loader.
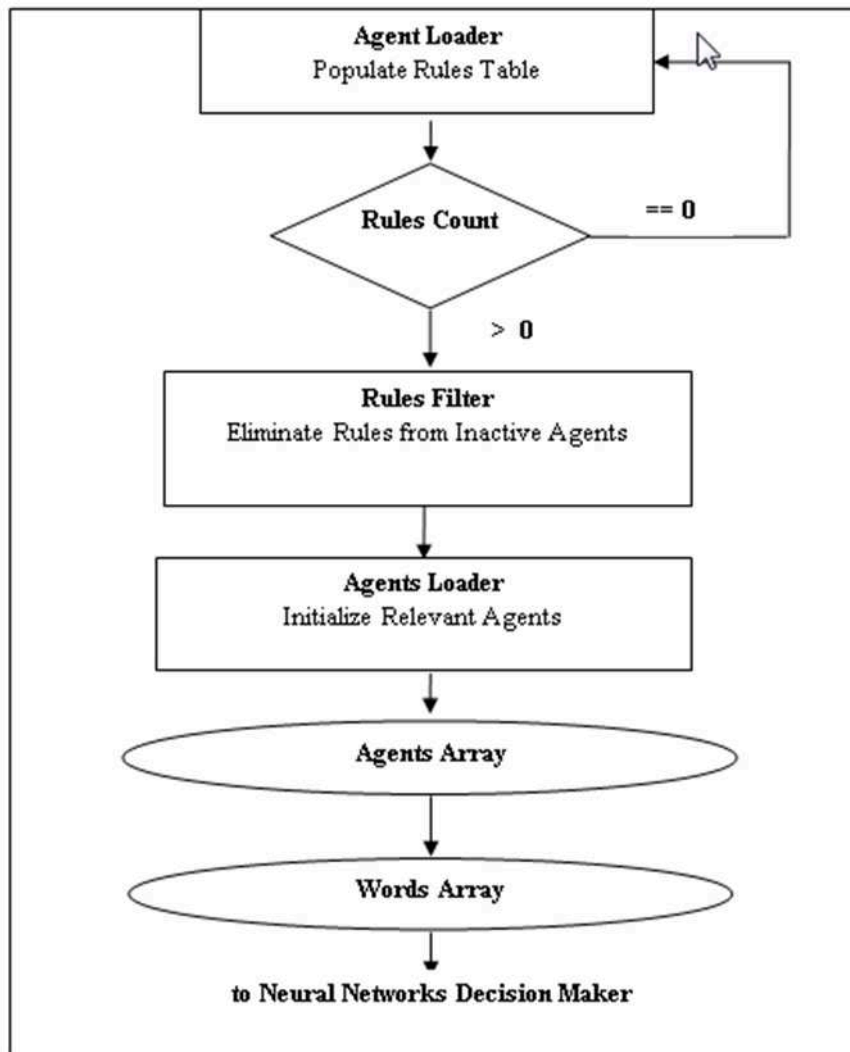
Figure 5: Agent Loader Diagram

4.3. Block Rules Scanner:

After the creation of the agents array that contains the set of active and relevant agents and words array that contains the set of extract words of the submitted message. The Block Rule Scanner will scan for the occurrences of message blocking rules and checks the message to find out if it has been blocked or not. If the message blocked then a log file will be generated and in the same time a block message will be sent to both sender and receiver of the message to notify them about the blocked message and specify the reasons behind this blocking which is the occurrences of unwanted words in the message body, while if the message has not been blocked the filter rule will be invoked to scan the message for the occurrences of message blocking rules which in turn checks wither the message has been filtered or not.

If the message has been filtered then the filtered words will be replaced and a log file will be generated finally the message will be sent as a final step otherwise the message will be sent directly for not containing any filtered or illegal words and with changes. Figure 6 shows the block diagram of "Block" Rule Scanner.
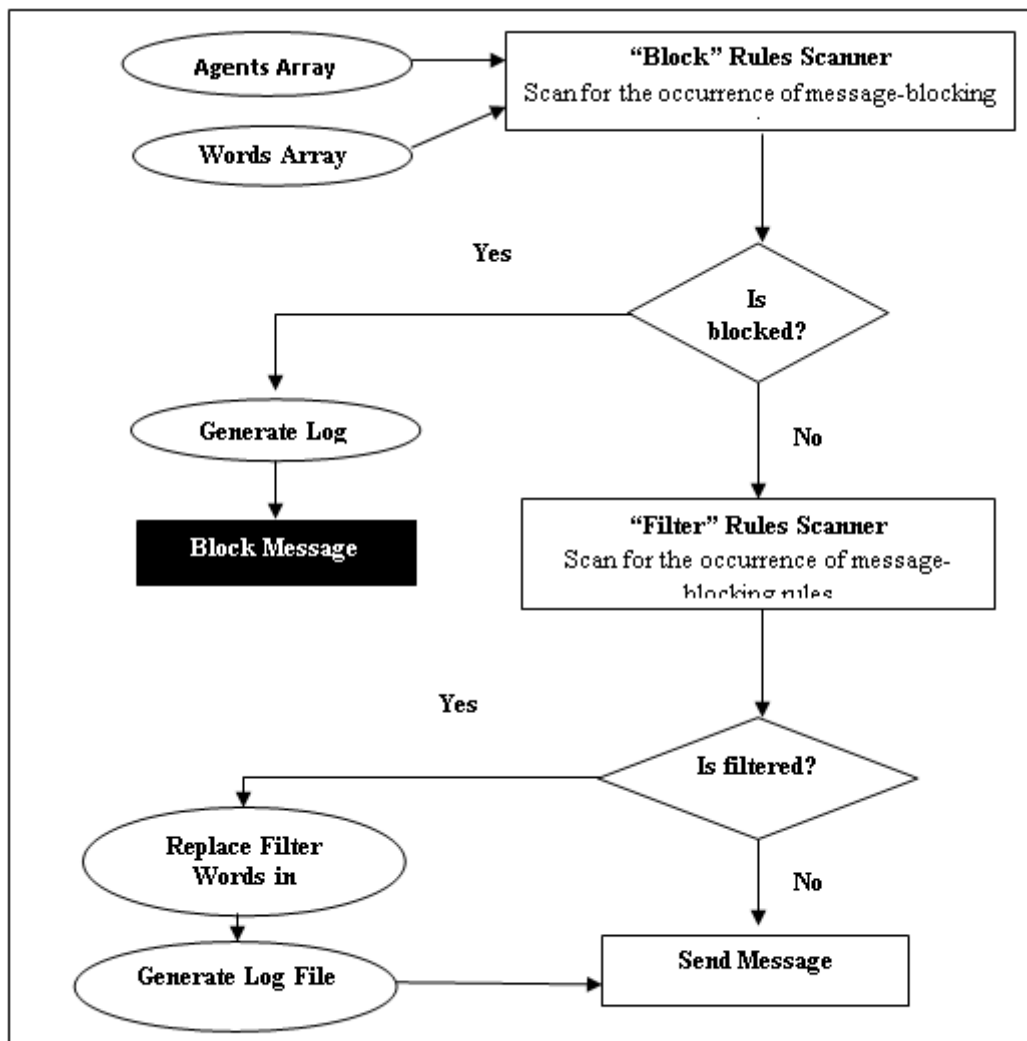
Figure 6:"Block" Rule Scanner Diagram

5. System Implementation:

In this section we will specify how the system is implemented, figure 7 shows the main page of the system in which consists of:
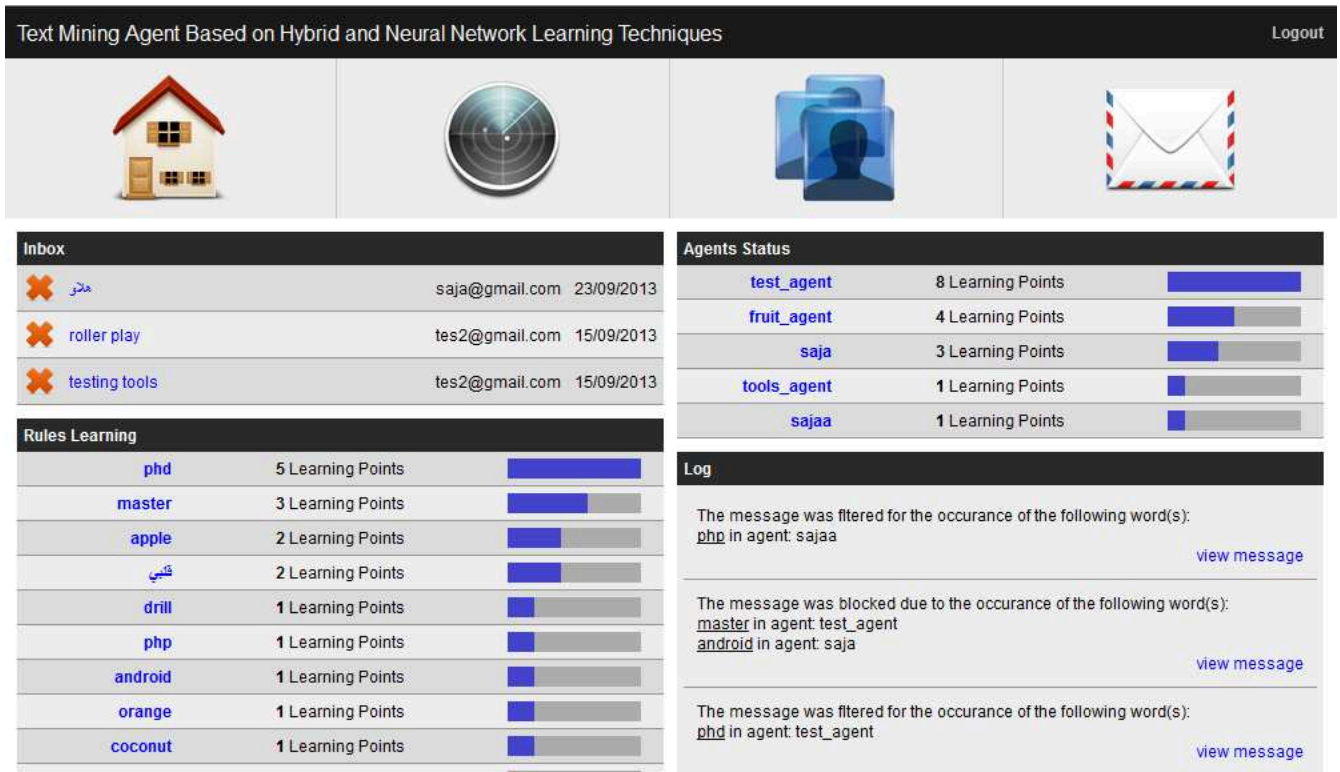
Figure 1. The Main System GUI

5.1. Home:

The main or the home page is the first page that opens when we start system running, and it represents the main GUI that allows the user to communicate with the system and it has been programmed using PHP and some Java Script codes, it is composed of the following divisions:

5.1.1. Top panel:

Allow the user to navigate the system through choices of Home, Configure Mail Agent, User Accounts and Send/Receive Mails, each choice will be explained in details in the next section figure 8 below shows the Top Panel division.

Figure 2. Top Panel

5.1.2. Inbox:

This division is similar to the inbox of any mail switching engine it shows the received messages with the details of the message as shown in the figure 9 below the message roller play has been sent by test2@gmail.com in 15/9/2013.

Figure 3. Inbox Division

5.1.3. Rules Learning:

Each agent may has one or more rules each time a rule matches with one or more E-mail contents the learning rate for that rule of the specific agent that handled filtering task will be increased. Rules learning division shows the learning rate for each rule and sort them in descending order as shown in figure 10.

Figure 4. Rules Learning Division

5.1.4. Agents Status:

Figure 11 shows the Agent Status division in which the status of the agents are shown and also shows the learning rate for each agent, Rules Learning shows the learning rate for the rules of different agents , while Agent Status shows the learning rate for different agents which increases with each invoking for agents by the master agent to handle filtering task and the new filtered word will be added to the attached file.
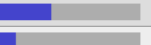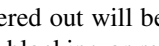

Figure 5. Agent Status

5.1.5. Log:

In this division (figure 12) all the received and sent messages that has been filtered out will be shown with their status (blocked/ filtered), it also shows the reason of blocking/filtering with the blocking or replaced words, and it enables the logged in user to view the message in new page.
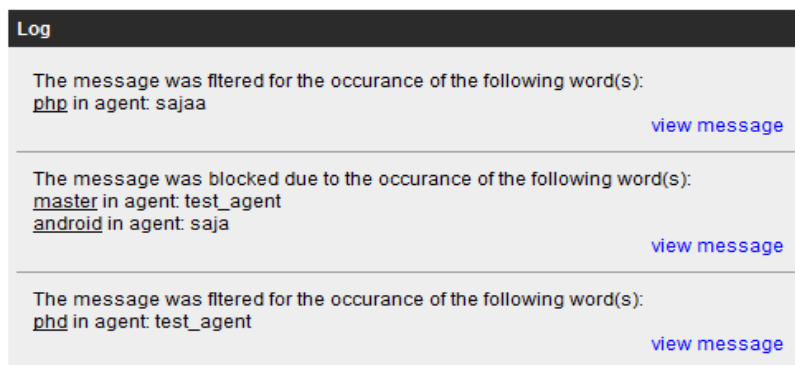
Figure 6. Log Division

5.2. Configure Mail Agent:

When the user needs to create a new agent, add a new rule to an existing agent or to delete an existing agent or rule of a specific agent then Configure Mail Agent allow the logged in user to perform one or more of these actions depending on user's requirements.

5.3. User Account:

This choice allows us to define a new user, to define a new user, the system will ask for an existing, trusted user Id for example Yahoo, Gmail, Hotmail or any other trusted account Id, then the system will randomly generate a log in or user entered password after checking the validity of the entered user account Id. It also shows the current system users.

5.4. Send/ Receive Mail:

Choosing this will move the logged in user to a new page in which enables the user to compose, send, and view received mails with the status for each sent/received mail, before sending a new composed message it checks the emptiness of message's subject and body, if it one or both are empty it shows error message and asks the user to add a subject to the sent message.

6. Conclusion and future work:

Spam or junk mails have become a serious problem, many techniques have been used to solve this problem, the proposed in this paper is solving spam problem using intelligent agent that performs filtering based on E-mail content and also it allow the system user to create different number of agents, each agent has its own set of rules, these rules can be modified to meet the objectives of the user that may change over time. The system proposed in this paper can be used in future with other filtering types or techniques as a complete multi layer filtering system.

7. References:

Lazurca C., Leon F. (2010). "An E-mail Filtering Agent Based On Support Vector Machines". Bulletin of the Polytechnic Institute of IASI, Posted by(Technical University "Gheorghe Asachi" Iasi),section(Automation and Computers).

Christina V., Karpagavalli S. & Suganya G. (2010). "A Study On Email Spam Filtering Techniques", International Journal of Computer Applications 12(1), 0975-8887.

Nosseir A., Nagati K. |& Taj-Eddin I. (2013). "Intelligent Word-Based Filter Detection Using Multi-Neural Networks", International Journal of Computer Applications 10(1), 1694-0784.

Wooldridge M. (2002). "Intelligent Agent: The Key Concepts", University of Liverpool, Computer Science Department. UK.

Jennings N. and Wooldrige M., "Applications of Intelligent Agents", Queen Mary & Westfield College, University Of London.

Russell S., Norving P. (2003). "Artificial Intelligence a Modern Approach". (2nd ed.). USA: Prentice Hall.

Jerey M. Bradshaw, (1997). " Software agents". AAAI Press . USA.

Qadir K., Mahdi, Q., & Flaih, L. (2010). "Web Page Classification Using Neural Networks With CPBF". University of Salahaddin:Kurdistan Region Of Iraq.

Akram s., Al-Hashimi A. & Al-Khafagi H. (2006). "Automatic Web Text Classification Using Data Mining". Iraqi Commission for Computer and Informatics. Iraq.

Wooldridge M. (2002). "An Introduction to Multi Agent Systems". University of Liverpool, Computer Science Department. UK.

This academic article was published by The International Institute for Science, Technology and Education (IISTE). The IISTE is a pioneer in the Open Access Publishing service based in the U.S. and Europe. The aim of the institute is Accelerating Global Knowledge Sharing.

More information about the publisher can be found in the IISTE's homepage: http://www.iiste.org

CALL FOR JOURNAL PAPERS

The IISTE is currently hosting more than 30 peer-reviewed academic journals and collaborating with academic institutions around the world. There's no deadline for submission. **Prospective authors of IISTE journals can find the submission instruction on the following page:** http://www.iiste.org/journals/ The IISTE editorial team promises to the review and publish all the qualified submissions in a **fast** manner. All the journals articles are available online to the readers all over the world without financial, legal, or technical barriers other than those inseparable from gaining access to the internet itself. Printed version of the journals is also available upon request of readers and authors.

MORE RESOURCES

Book publication information: http://www.iiste.org/book/

Recent conferences: http://www.iiste.org/conference/

IISTE Knowledge Sharing Partners

EBSCO, Index Copernicus, Ulrich's Periodicals Directory, JournalTOCS, PKP Open Archives Harvester, Bielefeld Academic Search Engine, Elektronische Zeitschriftenbibliothek EZB, Open J-Gate, OCLC WorldCat, Universe Digtial Library , NewJour, Google Scholar