Journal of Engineering Science and Technology
Vol. 6, No. 1 (2011) 50 - 60
© School of Engineering, Taylor's University

INTELLIGENT E-MAIL PERSONALIZATION SYSTEM

SHANMUGASUNDARAM HARIHARAN

Department of Information Technology,
J.J. College of Engineering and Technology, Tamilnadu, India
E-mail: mailtos.hariharan@gmail.com

Abstract

In Internet era E-mail has become the most important mode of communication in every day life. E-mail offers several advantages like secure delivery, speed, cheaper cost, acknowledgement report, transparent service, and distributed environment. As spammers try to induce large amount of spam or unsolicited mails, managing these E-mails's in an efficient manner requires huge attention. This paper focus on personalizing the E-mail messages after eliminating the spam messages. The basic step starts with pre-processing the documents and classifying the contents into several folders or categories. The next step is to cluster the documents based on the relativeness they have using cosine similarity metric. This clustering approach is carried out using unsupervised method. The mail messages are the parsed through a filter that would identify the spam immediately. Studies on personalization of mails after spam identification, prioritizing the E-mail's based on the importance and summarization of were also proposed. The results were quiet promising leading to efficient spam identification providing a platform for further improvements to build a domain independent personalizer system.

Keywords: E-mail, Summarization, Prioritization, Spam, Filtering, Clustering,
Text mining, Extraction.

1. Introduction

E-mail has become part of human's life for faster communication especially among professionals, educationalists and other social networking community. Personal computer users use E-mail's to communicate with friends, families, and colleagues for faster and efficient communication. E-mail's serves as an archival tool to some people, while many users never discard messages because their information contents might be useful at a later date as a reminder of upcoming events.

Nomenclatures

Cosine(ti, tj)	Cosine value of two vectors
n	Number of sentences
Sentence score (i)	Score for the sentence *i*
Specialwt	Additional weight added to the keywords
TFi	Term frequency of the word *i*
ti	Frequency value of document *i*
tj	Frequency value of document *j*

Abbreviations

ANN	Artificial Neural Networks
AOL	American Online

Schuff et al. [1] states that “E-mail is widely used to synchronize real time communication, which is inconsistent with its primary goals”.

As E-mail becomes a popular means for communication over the Internet, the problem of receiving unsolicited and undesired E-mail’s, called spam or junk mails, severely arises. The volume of E-mail that we get is constantly growing. We are spending more and more time filtering E-mail’s and organizing them into folders in order to facilitate retrieval when necessary. The rate of unsolicited (spam) E-mail is also rapidly increasing. Several examples of E-mail classifiers that attempt to sort out mails into folders, semi automatically such as: Ishmail [2], IBM’s MailCat and Magi [3]. Works on rule based approaches [1], assessing incoming messages and making recommendations before E-mail’s reach the user’s inbox [4], systems identifying messages belonging to the same activity [5].

In the past few years, internet technology has affected our daily communication style in a radical way. The electronic mail (E-mail) concept is used extensively for communication nowadays. This technology makes it possible to communicate with many people simultaneously in a very easy and cheap way. But, many E-mail’s are received by users without their desire. Spam mail (or, junk mail or bulk mail) is the general name used to denote these types of E-mail. Spam mails are defined as electronic messages posted to thousands of recipients usually for advertisement or profit. These spam mails increases day by day; hence they have to be treated immediately. It is found that about 10% of the incoming E-mail’s to the network was spam. As a result of another study, American Online (AOL) has stated that it has received 1.8 million spam mails until precautions have been taken (NISCC Quarterly Review, January–March 2003). According to a research performed by Symantec in 2002, 63% of the people receive over 50 spam mails per week while 37% of them receive over 1000; 65% waste at least 10min to delete unnecessary spam messages on daily basis while 24% spent time over 20 min. [6].

Though many methods are available to identify & prevent such spam mails, they aren’t satisfactory. These methods are roughly grouped into two broad categories as static methods and dynamic methods. Static methods focus on spam mail identification using a predefined address list. For instance, the mail servers like ‘‘hotmail’’ allows a person to receive an E-mail only if his/her address is one of the recipient addresses; otherwise the server treats the E-mail as spam. Also there is every chance that most spam mails pass this test and some important mails are treated as spam. Some servers even collect addresses which are reported as spammers (people who send spam messages) and treat these E-mail's as spam. However, spammers are all aware of most of these methods. Hence it is essential to analyze these issues in an efficient manner.

The main focus of the paper lies in the following argument: Anti-spam legal measures are gradually being adopted, but they have had a very limited effect so far [7]. Of more direct value are anti-spam filters, software tools that attempt to block spam messages automatically [8]. Apart from blacklist of frequent spammers and list of trusted users, which can be incorporated into any anti-spam strategy, these filters have so far relied mostly on manually constructed keyword patterns. Even worse, the characteristics of spam messages (e.g. topics, frequent terms) change over time, requiring the keyword patterns to be updated periodically [9]. So our focus is mainly to identify legitimate mails precisely that pass the filtering criterion based on keyword patterns (discussed in section 3.2.3).

The paper is organized as follows. While section 1 briefs about E-mail spam filtering and management, Section 2 discusses the related research carried out. In Section 3, complete system description detailing the corpus used, modules involved in the proposed system, proposed spam filtering, clustering and summarization is discussed. Finally section 4 gives the conclusions and future improvements.

2. Related Work

Ning et al. [10] approach deals with agents based architecture that mines the textual information. In their paper two kinds of text mining agents namely USPC uncertainty sampling based probabilistic classifier) and R2L (rough relation learning) are used cooperatively, for personal E-mail filtering and management. The major setback of the authors approach is that the documents were semi structured in nature. Clark et al. [11] have presented a neural network based system for automated E-mail systems that was able to fill up the incoming E-mail messages into folders and anti-spam. From the investigations the author has found that his technique is more accurate than several other techniques. Study was also made to investigate the effects of various feature selection, weighting and normalization methods and also the portability of the anti-spam filter across different users. The proposed technique mainly deals with clustering or grouping of mails into appropriate folders, rather on e-mail filtering.

Wen and Te [12] dealt some automatic classification approaches to filter spam from legitimate E-mail's using text mining. A cluster-based classification method, called ICBC, is developed accordingly. In the first phase, ICBC clusters E-mail's in each given class into several groups, and an equal number of features (keywords) are extracted from each group to manifest the features in the minority class. In the second phase, ICBC with an incremental learning mechanism is adopted to accommodate the changes of the environment in a fast and low-cost manner.

Enrico and Karl [13] integrated user-defined folder structures with classification schemes that have been automatically derived from the E-mail content. This technique allows automating the process of evolving and optimizing directory structures without sacrificing knowledge captured in manually created folder structures. A prototype is also demonstrated that would analyze the feasibility and utility and finally they were evaluated. The authors have addressed important practical problems and provided a relevant study on the application of various techniques for maintaining application specific ontologies. The major setback of this approach is that the application requires strong knowledge specific ontologies. Similarly, Levent Ozgur et al. [6] approach is based on Artificial Neural Networks (ANN) and Bayesian Networks. In their approach the authors developed algorithms that are user-specific and adapt themselves with the characteristics of the incoming E-mail's. Their algorithms have two main components. The first one deals with the morphology of the words and the second one classifies the E-mail's by using the roots of the words extracted by the morphological analysis. Two ANN structures, single layer perceptron and multi-layer perceptron, are considered and the inputs to the networks are determined using binary model and probabilistic model. Similarly, for Bayesian classification, three different approaches are employed: binary model, probabilistic model, and advanced probabilistic model. However our work doesn't focus on language processing techniques which are found to be domain dependent in nature.

Mizuno et al [14] analyzed a novel approach to detect fault-prone modules in a way that the source code modules are considered as text files and these were applied to the spam filter directly. Several experimental applications using source code repositories of Java based open source developments were conducted in their investigations and the result of experiments shows that their approach was able to classify more than 75% of software modules correctly.

Ayodele et al. [15] have designed and implemented a system to group and summarize E-mail messages. The authors proposed system uses the subject and content of E-mail messages to classify E-mail's based on users' activities and generate summaries of each incoming message with unsupervised learning approach. Also the problem of E-mail overload, congestion, difficulties in prioritizing and difficulties in finding previously archived messages in the mail box were solved using their approach. Taking all these issues in mind, this paper focus on building up an intelligent e-mail personalizer which solely focus on spam filtering, clustering and summarization of E-mails.

3. Experimental Setup

This section briefs the complete system description, corpus used and the details of each tasks associated with the proposed system.

3.1. Corpus used

The corpus used for our work was collected from at different time spans ranging over a period of 6 months. Table 1 shows the statistics on the number of spam, sent mails and inbox mails were collected from different E-mail addresses.

Table 1. Statistics of the Corpus Used for Experimenting.

E-Mail server	**No. of mail users**	**Training documents**			**Test documents**		
		Inbox mails	Sent mails	Spam mails	Inbox mails	Sent mails	Spam mails
Rediffmail	100	1400	200	350	150	25	40
Yahoo	75	300	95	100	50	25	20
Gmail	85	1250	190	290	150	50	50

3.2. System description

The proposed system architecture is shown in Fig. 1. Each of these modules is discussed in detail in sub sections shortly. Prioritization and Visualizer alone were not explained in detail, as this is not the main concern in this paper. The pseudo algorithm for Intelligent E-mail Personalizer is shown in Fig. 2. The sequence of steps involved is listed below.

- Process the mail messages
- Identify for spam mails
- Extraction of information
- Cluster the contents bases on similarity
- Summarize the contents

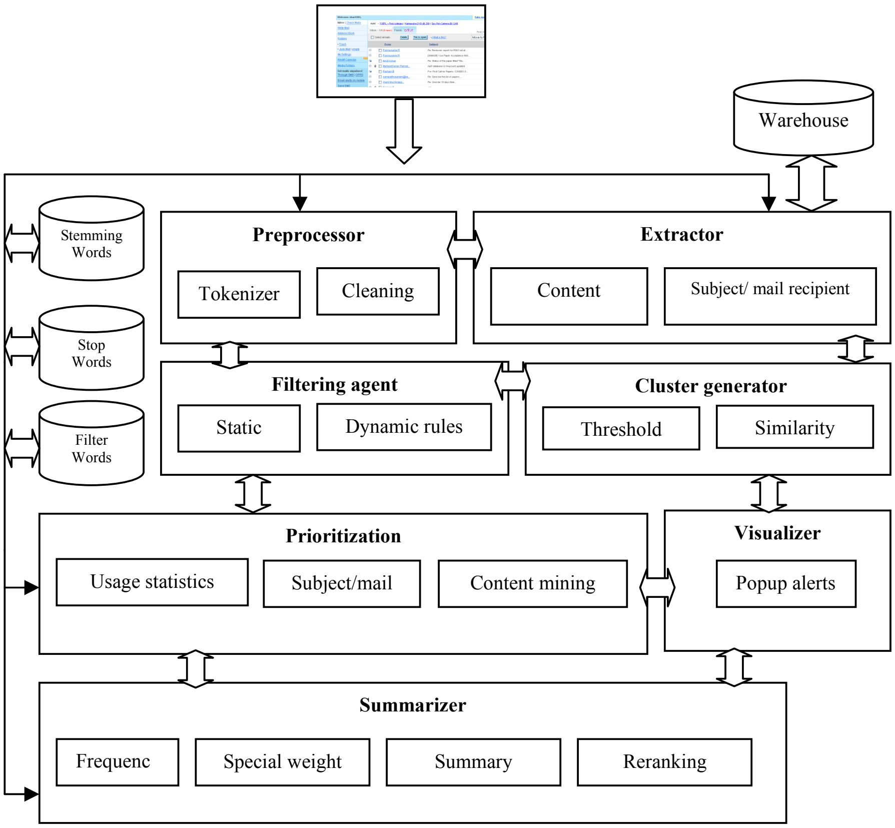


Fig. 1. System Architecture of the Proposed Intelligent E-mail Personalizer System.

3.2.1. Pre-processing:

The first basic step is to preprocess and extracting the relevant contents from different mail recipients**.** E-mail source may be of different formats and styles. By format and style we mean the language of the mail message. However our focus would on English language only and hence it is monolingual. The following tasks are carried out in this phase as shown below. Figure 2 presents the code snippet for preprocessing task.

a. Elimination of special characters,
b. Converting all uppercase letters to lowercase
c. Eliminating all non-letter characters
d. Removal of stop words [16]
e. Applying porter stemming algorithm [17].

```
preprocess( )
begin
remove special characters from the raw text;
identify unwanted words using IR corpus; /* eliminate unwanted words
stemming ( );        /* obtain the morphological root of each term
for each word, calculate the frequency of occurrence;
end
```

Fig. 2. Pseudo Code Snippet for Preprocessing Task.

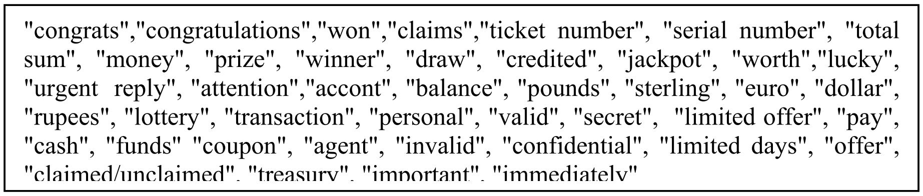
3.2.2. Extraction:

E-mail messages have subject, from/to mail identifiers, and content irrespective of mail servers, by default. Once the mail messages are classified into different categories, extraction identifies the components of the mail messages. To classify E-mail's, first a data set containing examples of spam mails and normal mails is compiled (done manually).The extracted information are stored in a separate warehouse, which is retrieved for further tasks. Pre-processed tokens are then passed to filtering agent, whose function is given on next section. The extraction task is done based on three different levels namely based on email's subject (subject level), mail recipient information (mail recipient level) and based on the contents (content level).

Each approach has its own significance. The first approach (subject level) reflects the importance (or) in turn tends the users' attention in reading through the mail immediately. Mail recipient approach is slightly inferior to the previous task, where in the user expects the mail from well-known recipients. However the first two tasks are not efficient and sometimes lead to false spam identification. The third task is most important, as it analyzes the contents of the entire document. Results were projected in appropriate Tables 3 and 4.

3.2.3. Filtering:

Filtering is done based on the list of filter words analyzed or mined by manual examination from the training corpus. These filter words are stored in filter databases for providing assistance in the process of filtering the terms or contents or mails. A sample of collected words is shown in Fig. 3. The pseudo code snippet for filtering is given in Fig. 4.



"congrats","congratulations","won","claims","ticket number", "serial number", "total sum", "money", "prize", "winner", "draw", "credited", "jackpot", "worth","lucky", "urgent reply", "attention","accont", "balance", "pounds", "sterling", "euro", "dollar", "rupees", "lottery", "transaction", "personal", "valid", "secret", "limited offer", "pay", "cash", "funds" "coupon", "agent", "invalid", "confidential", "limited days", "offer", "claimed/unclaimed", "treasury", "important", "immediately"

Fig. 3. Sample of Filter Words Identified in Training Corpus.

```
filter( )
begin
        extract tokens from the file
        if  tokens = = filter words
                mark as spam      /* mark as spam if match occurs
        else
                mark as ordinary mails
end
```

Fig. 4. Pseudo Code Snippet for Filtering.

Filtering agent focus on two types of rules namely static and dynamic rules. Static rules are those which block the mails by sending a request to the server. The server in turn blocks such mail recipients, on the other hand mails that pas through this filter may be blocked by users after they receive them. If is trivial and time consuming to mark such mails as spam, without knowing what content they have. Such documents are parsed through filter words database, so as to identify unsolicited mails.

Our focus is on identifying candidate messages that are to be marked as spam. We solve this issue using dynamic rules designed using mined content. We use combinational rules to identify such unwanted mails. We even categorize some filter words as context sensitive words. For example if subject or content contains the word “urgent”, “immediate”, users tend to open such mails at first. To have a feel of differentiation between these words, three examples were illustrated in Table 2. Few examples are illustrated to show the filtering criterion. Consider the sentence “Answer the questions and win gifts”. Here answer, questions, win and gifts are identified as valid tokens, of which gifts is a filter word. The scores assigned for the example is -47 (1+1+1-50). The scores for each sentence were assigned by the system (automatically), and then filtering was carried out. From the scores obtained, it is inferred that the sentence is to be neglected or the context is totally unimportant. Considering the category 2, the score would be 52. Hence this context is deemed to be more important. So the filtering is dependent on the context.

So the rule is designed such a way that if the tokenized word is filter word and context sensitive, then we go for finding out the some additional filter words. Each content in the document is rated based on the weights assigned by the system. Weights are obtained by Higher weights are signed to category 2, while lower weights for category 1&3. Then based on the final weights, the message is determined as useful mail or vice versa. Table 3 shows the accuracy of the system in identifying the spam mails. Accuracy of the system is defined as the number of correctly identified spam mails divided by the expected number of spam mails. From the experiments, it is inferred that accuracy achieved was 64.5%, 51.8% and 92% respectively for each of the approaches discussed previously in section 3.2.2. Hence it is clearly inferred that the spam mails were detected effectively by analyzing the entire content rather than focusing on the subject and mail recipients. Accuracy is defined as the no. of spam mails identified to number of spam mails to be identified correctly.

Table 2. Sample Rules Illustrating the Context.

Rule category	Subject	Sample Content	Context	Score
1	Urgent-reply immediately	U have won Rs 1 crore…	Fun	-50
2	Urgent!	Need O+ ve blood to save a life..	Emergency	+50
3	Urgent attention	Answer the questions and win gifts	Fun	-50

Accuracy = No. of mails correctly identified as spam mails/ Expected no. of spam mails.

Table 3. Spam Filtering for the Test Documents in Table 1.

Mail server	Approach focused	# of spam mails to be identified	# of spam mail identified	Accuracy
Rediff	Subject	40	25	0.625
	Mail id	40	27	0.675
	Content	40	38	0.950
Yahoo	Subject	20	13	0.650
	Mail id	20	8	0.400
	Content	20	17	0.850
Gmail	Subject	50	33	0.660
	Mail id	50	29	0.580
	Content	50	48	0.960

3.2.4. Clustering:

The content of each E-mail's were analyzed, with each word is represented as a vector model. These words are represented with a vector whose each element corresponds to a particular word and indicates whether that word occurs or not in the text or the number of times it occurs. There are wide ranges of similarity metrics available that reflects the importance of content like cosine, dice, hellinger, overlap and jaccard measure [18]. Cosine measure is most popularly used to reflect the originality of the content [19]. If t_i and t_j are frequency values of document 1 and 2 respectively, expression (1) measures the cosine relativeness among the two documents.

$$Cosine(ti,tj) = \sum_{i=1}^{n} t_i t_j \Big/ \sqrt{\sum_{i=1}^{n} t_i^2 \sum_{i=1}^{n} t_j^2} \qquad (1)$$

Under clustering, we investigate three types of clustering approaches. Such clustering approach has been successful earlier for clustering the text contents (in case of summarization) effectively [20]. Table 4 shows the results illustrating the three types of clustering approaches (discussed in Section 3.2.2) for various thresholds. From the values, it is inferred that threshold of 0.20 is optimal to cluster the features effectively.

Table 4. Clustering E-mail Messages at Specified Threshold.

			t = 0.2			*t* =0.3			*t* =0.4		
E-Mail Server	**Feature Used**	**# of legitimate mails**	**#mail clusters to be formed**	**# mails clusters identified**	**Accuracy**	**#mail clusters to be formed**	**# mails clusters identified**	**Accuracy**	**#mail clusters to be formed**	**# mails clusters identified**	**Accuracy**
Rediff	Subject	150	12	4	0.333	12	3	0.250	12	3	0.250
	Mail id	150	12	7	0.583	12	6	0.500	12	5	0.416
	Content	150	12	11	0.916	12	9	0.750	12	7	0.583
Yahoo	Subject	50	6	2	0.333	6	2	0.333	6	2	0.333
	Mail id	50	6	2	0.333	6	1	0.166	6	2	0.333
	Content	50	6	6	1.000	6	4	0.666	6	3	0.333
Gmail	Subject	150	13	4	0.307	13	3	0.231	13	3	0.231
	Mail id	150	15	7	0.466	15	6	0.400	15	5	0.333
	Content	150	15	14	0.933	15	12	0.800	15	10	0.666

Clustering the document sets based on the similarity exiting between the documents. When the content is too large, then it is proved form our previous results that top frequency terms is enough to cluster the contents effectively for threshold of 0.20. Also the top frequency based clustering would cluster the documents in lesser time span compared to other methods investigated [20]. It is inferred from Table 4 that threshold of 0.2 is optimal to identify the spam mails correctly. As the threshold increases the number of outliers also increases, leading to lesser accuracy. Accuracy is defined again as the no. of clusters identified correctly to the expected no. of clusters. The code snippet for clustering task is given in Fig. 5.

```
Cluster ( )
begin
for each valid mail          /* mails that are not identified as spam mail
similarity();                /* measure the overlap among the content
group them based on similarity value; / * cluster them based on similarity
end
```

Fig. 5. Pseudo Code Snippet for Clustering Task.

3.2.5. Summarization:

Having clustered the contents, now we propose an algorithm to summarize the contents. Summarization by extraction involves scoring the sentences and picking up the most important sentences. The following steps were adopted to summarize the contents:

a. Calculation of frequency weights of each token.
b. Scoring the terms in the document.
c. Ranking of documents based on weights.
d. Reproducing the results based on the user requirements.

Extract summaries are first produced for each document in the cluster and then sentences from these summaries are considered for inclusion in the summary of the multi document cluster. Sentences within each document are ranked depending on the sentence weights using term frequency approach [21]. For generating weight for each sentence we adopted extraction method, where the frequency of terms in each document, special weights to subject are considered. Finally for forming summary, sentences that are important in each cluster are considered for inclusion in the summary. We pick up sentences from the cluster up to the user specified compression rate. Sentence score has been obtained by adding Term Frequency. This is given in expression as shown below:

$$Sentence\ score_i = TF_i \times special_{wt} \tag{2}$$

where TF_i is the Term Frequency, Special weight denotes the additional weights for any word that matches the subject of e-mails.

4. Conclusion and Future Improvements

In this paper, we investigated several integrated approach on E-mail personalization, to identify the spam mails efficiently. Also we have made some studies pertaining to clustering the E-mails based on their contents and summarizing the contents. For each of these tasks pseudo algorithms were also given. The proposed spam detection technique was able to detect spam mails at an accuracy of 92% based on the contents of the emails. It is also inferred that the subject contained in each mail and the recipient would not be a key attribute to determine the mails as spam or vice versa. Now work is on progress to add some additional metrics to improve the spam mail detection ratio. Having identified such spam mails, documents were clustered effectively. From experiments it is found that a threshold of 0.20 is optimal. Then finally a summarization mechanism is explored further.

Throughout the paper we have made some investigations on the E-mail content which has text as its basic element. We have also focused only on English language only. The paper adopts popular extraction based approach rather than not focusing on the abstraction mechanism. In future studies, we plan to improve the algorithms by taking the attachments of the E-mail’s (pictures, text files, etc.) into consideration. We also like to extend the proposed system to multi lingual context. We focus our attention on mining the E-mail contents using linguistic features and measure the system behavior.

References

1. Schuff, D.; Turetken, O.; D’Arcy, J.; and Croson, D. (2007). Managing email overload: solutions and future challenges. *IEEE Computer Society*, 40(2), 31-36.

2. Helfman, J.I.; and Isbell, C.L. (1995). Ishmail: Immediate identification of important information. *AT&T Labs*.
3. Payne, T.; and Edwards, P. (1997). Interface agents that learn: An investigation of learning issues in a mail interface. *Applied Artificial Intelligence*, 11(1), 1-32.
4. Boone, G. (1998). Concept features in Re: agent, an intelligent email agent. *Proceedings of 2nd International Conference on Autonomous Agents, ACM Press*, 141-148.
5. Kushmerick. N.; and Lau, T. (2005). Automated email activity management: An unsupervised learning approach. *In: Proceedings of 10th International Conference on Intelligent User Interfaces, ACM Press*, 67-74.
6. Ozgur, L.; Gungor, T.; and Gurgen, F. (2004). Adaptive anti-spam filtering for agglutinative languages: A special case for Turkish. *Pattern Recognition Letters*, 25(16), 1819–1831.
7. http://www.cauce.org, http://spam.abuse.net, http://www.junkemail.org (last accessed 11.07.2010).
8. http://www.tucows.com (last accessed 11.07.2010).
9. Cranor, L.F.; and LaMacchia, B.A. (1998). Spam! *Communications of the ACM*, 41(8), 74–83.
10. Zhong, N.; Matsunaga, T.; and Liu, C. (2002). A text mining agents based architecture for personal e-mail filtering and management. *Intelligent Data Engineering and Automated Learning — IDEAL 2002. LNCS* 2412, 329-336.
11. Clark, J.; Koprinska, I.; and Poon, J. (2003). A neural network based approach to automated e-mail classification. *Proceedings of Web Intelligence, Proceedings. IEEE/WIC International Conference*, 702- 705.
12. Hsiao, W.F.; and Chang, T.M. (2008). An incremental cluster-based approach to spam filtering. *Expert Systems with Applications*, 34(3), 1599-1608.
13. Giacoletto, E.; and Aberer, K. (2003). Automatic expansion of manual email classifications based on text analysis. *LNCS,* 2888, 785 -802.
14. Mizuno, O.; Ikami, S.; Nakaichi, S.; and Kikuno, T. (2007). Spam filter based approach for finding fault-prone software modules. *Fourth International Workshop on Mining Software Repositories*, *ICSE Workshops MSR '07*, 1-4.
15. Ayodele, T.; Khusainov, R.; and Ndzi, D. (2007). Email classification and summarization: A machine learning approach. *IET Conference on Wireless, Mobile and Sensor Networks (CCWMSN07)*, 805-808.
16. http://www.lextek.com/manuals/onix/stopwords1.html (last accessed 19.08.2010).
17. Porter, M.F. (1980). An algorithm for suffix stripping. *Program*, 14(3), 130−137.
18. http://www.dcs.shef.ac.uk/~sam/stringmetrics.html (last accessed 14.08.2010).
19. Berry, M.W.; and Browne, M. (2006). *Lecture notes in data mining*. World Scientific Publishing, Singapore.
20. Maruthamuthu, P.; Maheedharan, R.; Kirubakaran, G.; and Hariharan, S. (2009). Experiments on clustering and multi-document summarization. *The IUP Journal of Computer Sciences*, *ICFAI Publications*, 3(2), 64-73.
21. Hariharan, S.; and Srinivasan, R. (2008). Investigations in single document summarization by extraction method. *International Conference on Computing, Communication, and Networking, ICCCN'08*, 1-5.