Email classification and summarization: A machine learning approach

Conference Paper · January 2008

DOI: 10.1049/cp:20070271 · Source: IEEE Xplore

Taiwo Ayodele
University of Portsmouth
34 PUBLICATIONS **726** CITATIONS

SEE PROFILE

Rinat Khusainov
University of Portsmouth
78 PUBLICATIONS **712** CITATIONS

SEE PROFILE

David Ndzi
University of the West of Scotland
124 PUBLICATIONS **1,476** CITATIONS

SEE PROFILE

Email Classification and Summarization: A Machine Learning Approach

Taiwo Ayodele Rinat Khusainov David Ndzi
Department of Electronics and Computer Engineering
University of Portsmouth, United Kingdom
{taiwo.ayodele, rinat.khusainov, david.ndzi} @port.ac.uk

Keywords

Email, algorithms, email summarization, activities, Classification.

Abstract

This paper presents the design and implementation of a system to group and summarize email messages. The system uses the subject and content of email messages to classify emails based on users' activities and generate summaries of each incoming message with unsupervised learning approach. Our framework solves the problem of email overload, congestion, difficulties in prioritizing and difficulties in finding previously archived messages in the mail box.

1. Introduction

Emails are parts of everyday life. Personal computer users use emails to communicate with friends, families, e-businesses and colleagues allowing ease for communication. Emails serve as an archival tool to some people, while many users never discard messages because their information contents might be useful at a later date – for example, as a reminder of upcoming events and outstanding issues. Also, a paper by Schuff et al [1] states that "Email is widely used to synchronize real-time communication, which is inconsistent with its primary goals". Email messages are designed to be sent, accumulated in a repository and be periodically collected and read by a recipient. And because of the high volume of email received daily the mail box is easily congested. Messages range from static organizational knowledge to conversations with a broad horizon of topics. Users may find it difficult to prioritize and successfully process the contents of incoming messages. Also it may be difficult to find a previously archived message in a mail box. In this paper we propose a new effective method for managing information in email, reducing email overloads by the method of grouping emails based on users' activities, and providing summarization of emails in this project.

We propose email groupings based on users' activities where incoming mails are identified and grouped into appropriate activities and related messages are grouped in the same activity. Email messages are grouped by extracting most frequent words in the content of the message as well as comparing common words with most frequent words in the message to decide which activity the email message belongs.

We developed some techniques that allow our classifier and summarizer to extract information from email messages and build a model from extraction of most frequent and common words in email messages in order to group messages into activities. Our classifier and summarizer make use of some rules sets to group emails into activities based on their observations and set of rules that is passed unto both the classifier and the summarizer.

2. Related Work

One of the common existing methods is to manually archive messages into folders with a view of reducing the number of information objects a user must process at any given time. However, this is an insufficient solution as folder names are not necessarily a true reflection of their content and their creation and maintenance can impose a significant burden on the user [1].

There are several examples of email classifiers that attempt to sort out mails into folders, semi-automattically such as:

- Ishmail [3]: It automatically sorts email messages into folders and orders them by importance.
- Commercial email clients [4: Most popular commercial email clients like Procmail, Eudora, Mozilla Thunderbird, Microsoft Outlook and Outlook Express also support message filing according to user defined rule sets.
- IBM’s MailCat [5]: It adapts dynamically to observed users’ mail-filling habits and provides a list of three folders most likely to be appropriate for a given message.
- Magi [6]: This system records each email interaction and uses a learning algorithm to classify new messages based on the user’s prior behavior.

A rule-based system as explained by Schuff et al [1] can provide straight forward way to semi-automatic email classification and such a system requires the users to define a set of instructions for the email application to sort incoming messages into folders and order them by importance. The disadvantages of rule-based system are that they are challenging for non-technical users because writing the rules require some level of programming experience. Bifrost email classifier and prototype email management system [3] addresses this problem by letting the user to define all filtering rules with a simple graphical interface. Terry et al [4] proposed a new approach by automatically assessing incoming messages and making recommendations before emails reach the user’s inbox. This classifier assigns a priority to each message as being of either high or low importance based on its expected utility to the user. Kushmerick [2] designed a system that automatically identifies messages belonging to the same activity, an electronic commerce transaction, thereby providing a high-level view that supports the use of email as a task manager.

3. Our Solution

We have designed and developed a system that summarizes email messages and also groups emails into activities. Our proposed email summarizer and email classifier are explained below.

3.1 Email Summarization

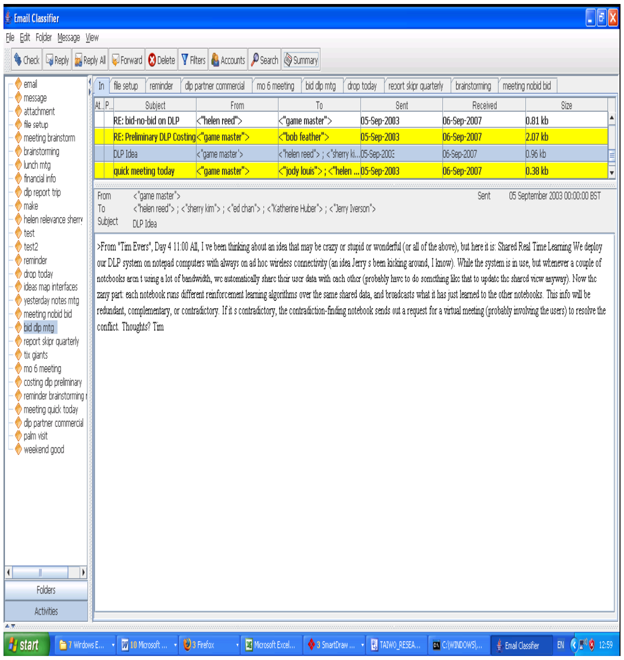
Our contribution in this area is the consideration of the highest frequencies of words in email messages, with selection of the sentences that contain the most frequent words and re-arranging these in an order that generates a good summary. The algorithm is shown in the Figure 1 below:

```
Summarization Algorithm Summary as
Input: N, M, Message    Output: Sentence list
1). Identify N most frequent words in the message
2). Select M sentences from email containing most frequent words
3). Order the selected sentences according to their occurrence in the message
4). Output the ordered sentences
```

Figure 1: Algorithm for summarization

The summarization algorithm selects sentences that have the most frequent word and arrange them in logical order to make the email message summary. Figure 2 and 3 show the output of the proposed summarizer.

Figure 2: Original Message received



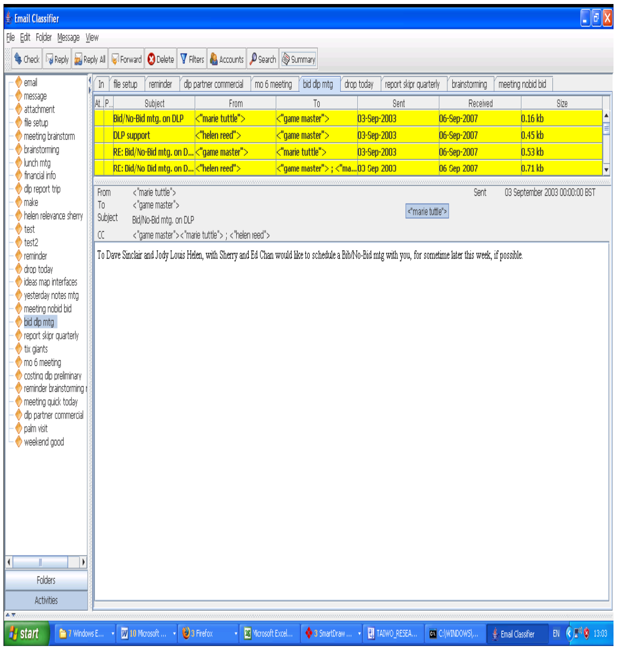
At...	P...	Subject	From	To	Sent	Received	Size
		RE: bid-no-bid on DLP	<"helen reed">	<"game master">	05-Sep-2003	06-Sep-2007	0.81 kb
		RE: Preliminary DLP Costing	<"game master">	<"bob feather">	05-Sep-2003	06-Sep-2007	2.07 kb
		DLP Idea	<"game master'>	<"helen reed"> ; <"sherry ki...	05-Sep-2003	06-Sep-2007	0.96 kb
		quick meeting today	<"game master">	<"jody louis"> ; <"helen ...	05-Sep-2003	06-Sep-2007	0.38 kb

From <"game master"> Sent 05 September 2003 00:00:00 BST
To <"helen reed"> ; <"sherry kim"> ; <"ed chan"> ; <"Katherine Huber"> ; <"Jerry Iverson">
Subject DLP Idea

>From "Tim Evers", Day 4 11:00 All, I ve been thinking about an idea that may be crazy or stupid or wonderful (or all of the above), but here it is: Shared Real Time Learning We deploy our DLP system on notepad computers with always on ad hoc wireless connectivity (an idea Jerry s been kicking around, I know). While the system is in use, but whenever a couple of notebooks aren t using a lot of bandwidth, we automatically share their user data with each other (probably have to do something like that to update the shared view anyway). Now the zany part: each notebook runs different reinforcement learning algorithms over the same shared data, and broadcasts what it has just learned to the other notebooks. This info will be redundant, complementary, or contradictory. If it s contradictory, the contradiction-finding notebook sends out a request for a virtual meeting (probably involving the users) to resolve the conflict. Thoughts? Tim

Figure 2 shows an email message received. Emails are passed to the summarizer and the summarizer makes a summary of each mail as shown in the text area. Figure 3 below shows the summary of the mail above.

Figure 3: Summary of Original message



At...	P...	Subject	From	To	Sent	Received	Size
		Bid/No-Bid mtg. on DLP	<"marie tuttle">	<"game master">	03-Sep-2003	06-Sep-2007	0.16 kb
		DLP support	<"helen reed">	<"game master">	03-Sep-2003	06-Sep-2007	0.45 kb
		RE: Bid/No-Bid mtg. on D...	<"game master">	<"marie tuttle">	03-Sep-2003	06-Sep-2007	0.53 kb
		RE: Bid/No Bid mtg. on D...	<"helen reed">	<"game master"> ; <"ma...	03-Sep-2003	06-Sep-2007	0.71 kb

From <"marie tuttle"> Sent 03 September 2003 00:00:00 BST
To <"game master">
Subject Bid/No-Bid mtg. on DLP
CC <"game master"><"marie tuttle"> ; <"helen reed">

To Dave Sinclair and Jody Louis Helen, with Sherry and Ed Chan would like to schedule a Bib/No-Bid mtg with you, for sometime later this week, if possible.

3.2 Email Classification

The classifier learns from the set of rules that are passed unto it and based on the level of similarity in the email content and the subject, our classifier decides the appropriate activity the message should belong. Our classifier also extracts common words (repeated words) in the subject of the email as well as the content of the mail. We use the stop words to prevent the algorithm to count unnecessary words like "the, a, in, at". Our algorithm explains more in figure 4 below.

Figure 4: **Classification Algorithm**

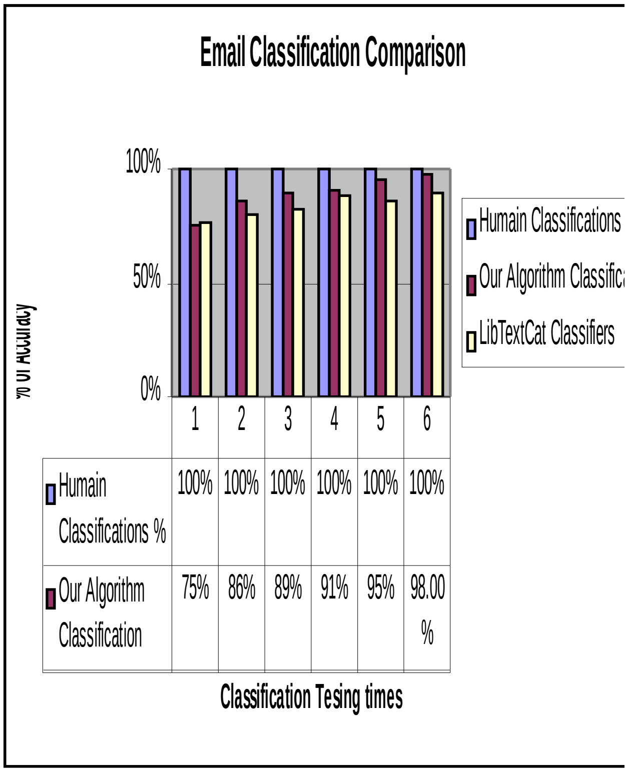
```
For every message M
Let FW = N most frequent words in the message
        Iterate over all activities and for each activity AC
                Let AFW = common words in activity AC
                If (FW = AFW then
mark the activity AC
update the message activity as AC
create a rule that states // machine learning
        for each message received that has some words like FW
        AC is the activity for this message
 Else create a new activity
End
```

When emails are received by the email client, these are passed to the classifier and then the classifier groups the emails into activities that users perform. In Figure 3, these activities are shown at the left hand side. So, if the same email belongs to the same activities, they tend to form a structured thread and be grouped into the activities based on the content and subject of the email message.

4. Evaluation

We evaluate our classification algorithm's performance by comparing performance of human participants as well as other email classification software (LibTextCat[1]). The results above show improvement in our proposed classification algorithm as we evaluate the accuracy of correct and incorrect classification as well as the activity that is created, if the emails are being grouped into the correct activities using human classification as the gold standard. Our output in Figure 5 shows that LibTextCat[2] classification accuracy for the sixth classification tests is 90% while our propose classifier obtain 98% accuracy. This indicates that our proposed classifier gives a very good performance. Figure 5 shows more output results.

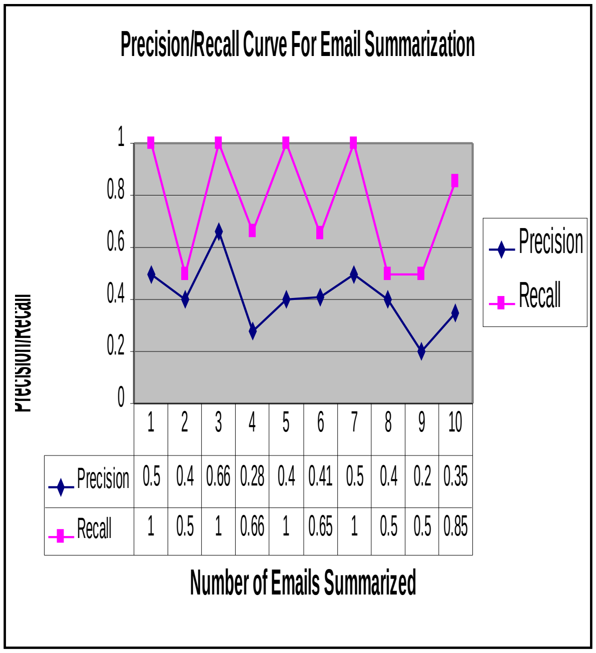
Figure 5: Email Grouping Result



So, comparing to other techniques in [1, 8, 9], this is more suitable for real time email client system because of the efficiency and good performances.

We also evaluate our summarization algorithm's performance by comparing performance of human participants as well as other email summarizers' software (Ss summarizer[3]). Figure 6 below explain more details

Figure 6: Email Summarization



To measure the quality of the email summaries, human summaries are used as references. We evaluate our proposed email summarizer against summaries from human participants and well as against Ss summarizer's software[4] as shown in Figure 6. The comparison is performed by using information retrieval metrics of precision and recall: a person is asked to select sentences that seem to best convey the meaning of the email message to be summarized and then the sentences selected automatically by a system are evaluated against the human selections. Recall is the fraction of sentences chosen by the person that were also correctly identified by the system

$$\text{Recall} = \frac{|\text{system-human choice overlap}|}{|\text{sentences chosen by human}|}$$

[1] http://software.wise-guys.nl/libtextcat/

[2] http://software.wise-guys.nl/libtextcat/

[3] http://www.kryltech.com/summarizer.htm

[4] http://www.kryltech.com/summarizer.htm

$$\text{Precision} = \frac{|\text{system-human choice overlap}|}{|\text{sentences chosen by system}|}$$

The summarization algorithm has been found to work well for high volume of emails. Figure 6 shows from the summaries of 100 emails (Showing 10 emails results) that we obtain a recall score between 50% and 100% difference in the qualities of the email summaries while precision score is between 20% and 70% as shown by our evaluation in Figure 6 above. In light of these observations, our output result shows that recall is more beneficial and generates better email summaries. In this paper recall measures the overlap with already observes sentences choice.

5. Conclusions

We have presented an overview of the proposed solutions to extract important words in email messages to provide a better summary than simply running the unprocessed message with a machine learning approach. This is another better way of generating useful summaries thus far. Our system also would be able to group emails messages into user's activities and provide a mechanism for emails that needs attention.

References

[1]. D. Schuff, O. Turetke, D. Croson, F 2007, ‘Managing Email Overload: Solutions and Future Challenges’, *IEEE Computer Society, vol. 40, No. 2, pp*. 31-36.

[2]. N. Kushmerick, T. Lau, 2005, ‘*Automated Email Activity Management: An Unsupervised learning Approach’, Proceedings of 10th International Conference on Intelligent User Interfaces*, ACM Press, pp. 67-74.

[3]. J. Helfman, C. Isbell, 1995, ‘Ishmail: Immediate Identification of Important Information’, AT&T Labs.

[4]. G. Boone, 1998, ‘Concept Features in Re: Agent, An Intelligent Email Agent’, *Proceedings of 2nd International Conference on autonomous agents,* ACM Press, pp.141-148.

[5]. R.B. Segal, J.O. Kephart, 2002, ‘MailCat: An Intelligent Assistant for Organizing Email’, *Proceedings of* 3^{rd} *Annual Conference on Autonomous Agent*, ACM Press, pp. 276-282. [6]. T. Payne, P. Edwards, 1997, ‘Interface Agents that learn: An Investigation of Learning Issues in a Mail Interface’, Applied Artificial Intelligence, vol. 11, no. 1, pp1-32.

[7]. L. Zhou, E Hovy, 2005, “On the Summarization of Dynamically Introduced Information: Online Discussions and Blogs”,`In *Proceedings of AAAI-2006 Spring Symposium on Computational Approaches to Analyzing Weblogs*, Stanford, CA.

[8]. D. Lam, S. Rohall, C. Schmandt, M. Stern, F 2002, ‘Exploiting Email Structure to improve Summarization’, A Collaborative User Experience Technical Report (TR2002- 02), IBM Watson Research Center.

[9]. S. Whittaker, C. Sider, 1996, ‘Email overload: exploring personal information management of email’, *CHI ‘96’*, pp.276- 283. ACM Press.