Supporting user task based conversations via e-mail

Jyot Boparai
School of Information Technologies
University of Sydney
NSW 2006 Australia
jboparai@it.usyd.edu.au

Judy Kay
School of Information Technologies
University of Sydney
NSW 2006 Australia
judy@it.usyd.edu.au

Abstract

E-mail is commonly used for 'conversations'. These consist of a sequence of messages which deal with a common task. It would be helpful if mail clients could automatically group messages from one conversation. This would facilitate the user's processing of them as it would enable the user to establish the context of the task that is at the core of the conversation.

This paper describes IETMS, a mail client which can employ a range of approaches for this task: standard mail header elements; a TF-IDF classifier and user-lists. As a foundation for improving our understanding of the effectiveness of these mechanisms, we have performed a detailed, small-scale study involving a corpus of mail which contains a collection of conversations about an important subclass of conversations, those concerned with organising meetings. The corpus size was chosen to be comparable to the number of conversations that might run in parallel for one user who is a quite heavy e-mail user. Our study indicates the relative power of each of these as well as their combined power. It also gives insight into the value of modelling individual user's e-mail behaviours and the ways that these interact with classification mechanisms.

Keywords Document Databases, Document Workflow, Document Management, Information Retrieval

1 Introduction

E-mail is widely used for an extremely varied range of purposes. An important core of e-mail messages is the communication between people about common tasks such as organizing to meet each other, working on a joint document such as a poster or planning the details of a social event. The fundamental role of e-mail is the broad support of message-based asynchronous communication. This gives it the flexibility to support an arbitrarily diverse set of user communication. This is an important part of the reason that it has been so successful and widely adopted.

At the same time, it gives no explicit support to any one common class of tasks. This means that there are many times when a user can find it challenging to perform a task. For example, if the user receives a substantial volume of mail, it can be difficult and tedious to work systematically through all the messages related to that task. They will typically be interwoven with the mail for other sets of tasks.

Traditional e-mail clients provide two forms of support for such management of tasks: *folders* for placing related mail items together and *threading* mechanisms for recognizing mail items that may be part of an ongoing task.

In this paper we explore approaches to improving support for management of task-based e-mails by extending the power of the basic mail classification and threading. Our goal is to assist users by automatically grouping the messages belonging to a common task and by supporting the user in tracking the status of tasks.

2 Task-based e-mail conversations

In the context of e-mail, a task can be defined as an exchange of naturally chained messages [8], [1], [9], [4]. We call this a *conversation*. For example, in discussion boards, a user initially composes and sends a message about a specific topic. Other users may contribute to the discussion and the conversation continues. A conversation may involve an arbitrary number of people exchanging messages over an extended period of time. This may be just one user, where the user talks to themselves as they progress through a task. Typically, it will involve several people and considerable time may elapse between replies.

Proceedings of the 7th Australasian Document Computing Symposium,
Sydney, Australia, December 16, 2002.

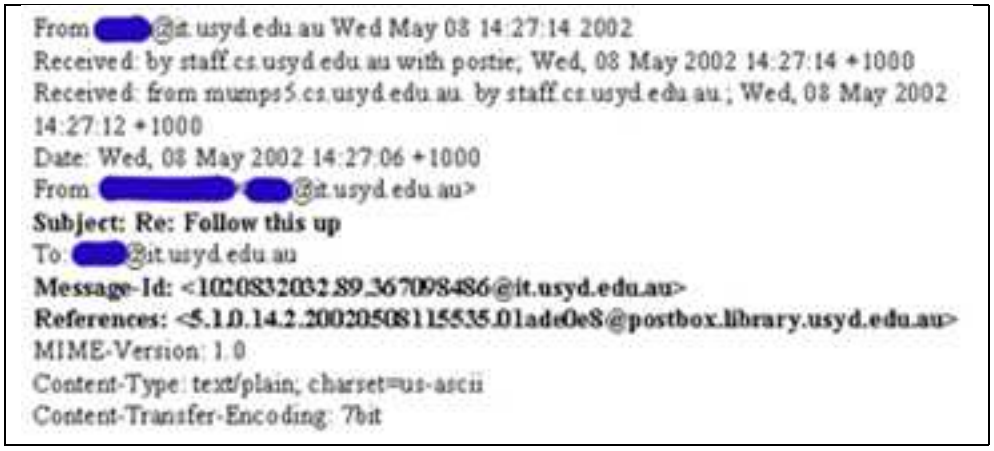

Figure 1: An example of a 'reply' message sent using a MIME-compliant e-mail client. The fields shown in bold are commonly used to thread messages in an e-mail conversation

3 Current Approaches

Current approaches make use of e-mail headers to classify e-mail messages into tasks. These headers include MIME [5, 6] headers and the headers defined in RFC2822 [7] mail standard. Another popular approach is to generate rules which can be triggered based on values of various e-mail fields (eg subject, sender, etc). When triggered, these rules automatically classify messages into folders or tasks. We now discuss these approaches and their limitations.

3.1 E-mail Headers

The mail standards RFC2822 and MIME specify headers which are recognized by most mail clients. Some are very useful for predicting which mail items are responses to an ongoing conversation. These include: *Message-Id*, *References* and *In-Reply-To*. When a person uses a MIME-compliant mail client to reply to a message, the client includes a header with the *Message-Id* of the original message in the *References* or *In-Reply-To* field of the reply message.

This is illustrated in Figure 1 which is a reply to the message with the id listed in the *References* header field. Figure 1 also illustrates the valuable role of the *Subject* line: many e-mail clients preserve the same subject in reply messages, although they may add *re:*.

3.2 Rules

Many e-mail clients allow users to define rules which automatically classify messages into folders or tasks. These rules can then be triggered, based on the values of various e-mail fields. All parts of a message may be used for this purpose including the body of a message.

Rules can be generated, automatically or manually, of the form:

if (subject *contains* "Follow this up") -> paper

to automatically classify incoming mail based on the values of subject field. In the above example, an e-mail message containing *"Follow this up"* in its subject field is classified into *paper* task.

Consider one very widely used e-mail client, Microsoft Outlook Express. It does not have any task management or task delegation support. However, it provides users with an option to group messages into threads. It also allows users to define rules which automatically classify messages based on the sender, subject and keywords present in the body of a message. Users may define these rules to automatically file incoming messages into one of the folders, where one folder represents one task. But defining these rules increases the cognitive load on the user and in some cases it may not be possible to define all the rules because of the complex nature of tasks.

Task management requires users to ensure that information relating to current tasks is grouped together. This both preserves the task context and allows users to determine the progress of an ongoing task [8], [1], [9], [4]. The threading ability of Outlook helps users in managing tasks to some extent as all the information is present in a single conversational thread.

When the option to group messages into threads is selected, the threading algorithm of Outlook Express groups messages with same subjects (after removing *'re:'* and *'fw:'*). It then uses *Message-Id* to thread the grouped messages.

3.3 Limitations

The strategies used by existing systems have some serious shortcomings for automated classification of e-mail messages into conversations. We now outline several forms of this.

The users may use the reply facility of a client to save typing an e-mail address even when the user is not actually replying to that message. If

a MIME-compliant e-mail client is used by a user, using reply feature will preserve the *Message-Id* in the reply message. As a consequence the message may be incorrectly classified.

Users tend to have conversations about multiple tasks in a single e-mail message to avoid sending multiple messages – a separate one for each topic. This introduces problems for a task management system if it classifies the message under that one task where it should, correctly, have classified it under each of the tasks discussed in the message.

Another related, but different, problem arises from drifts in conversations. Initially, the conversation may start discussing a topic but as the conversation evolves, the topic drifts. The task management system, assuming these messages belong to the same conversation, classifies them into a common task. This increases the cognitive load on the user because the information relating to one task is not readily available – they would have to go through all such messages to find the required information.

Yet another way that users can subvert task classification is by composing a fresh message as a reply to an ongoing conversation. Then the header fields (MIME and RFC2822) are not preserved. Approaches which rely on these fields to classify messages into tasks will fail when these fields are missing.

A final problem follows where the user's e-mail client is not MIME-compliant. In this case, the reply mail will not preserve MIME headers when a user chooses to reply to a message. Some of the clients add *'re:'* to the subject field of a reply message. In these cases, threading based on *Message-Id* will fail.

4 IETMS

Intelligent E-mail and Task Management System (IETMS) builds upon iems [2, 3], the intelligent e-mail sorter. IETMS supports task delegation and task management including mechanisms for defining tasks and the ability to automatically classify messages into one of these tasks. A meeting support wizard is also included to help users initiate meetings. We now present a detailed explanation of these features.

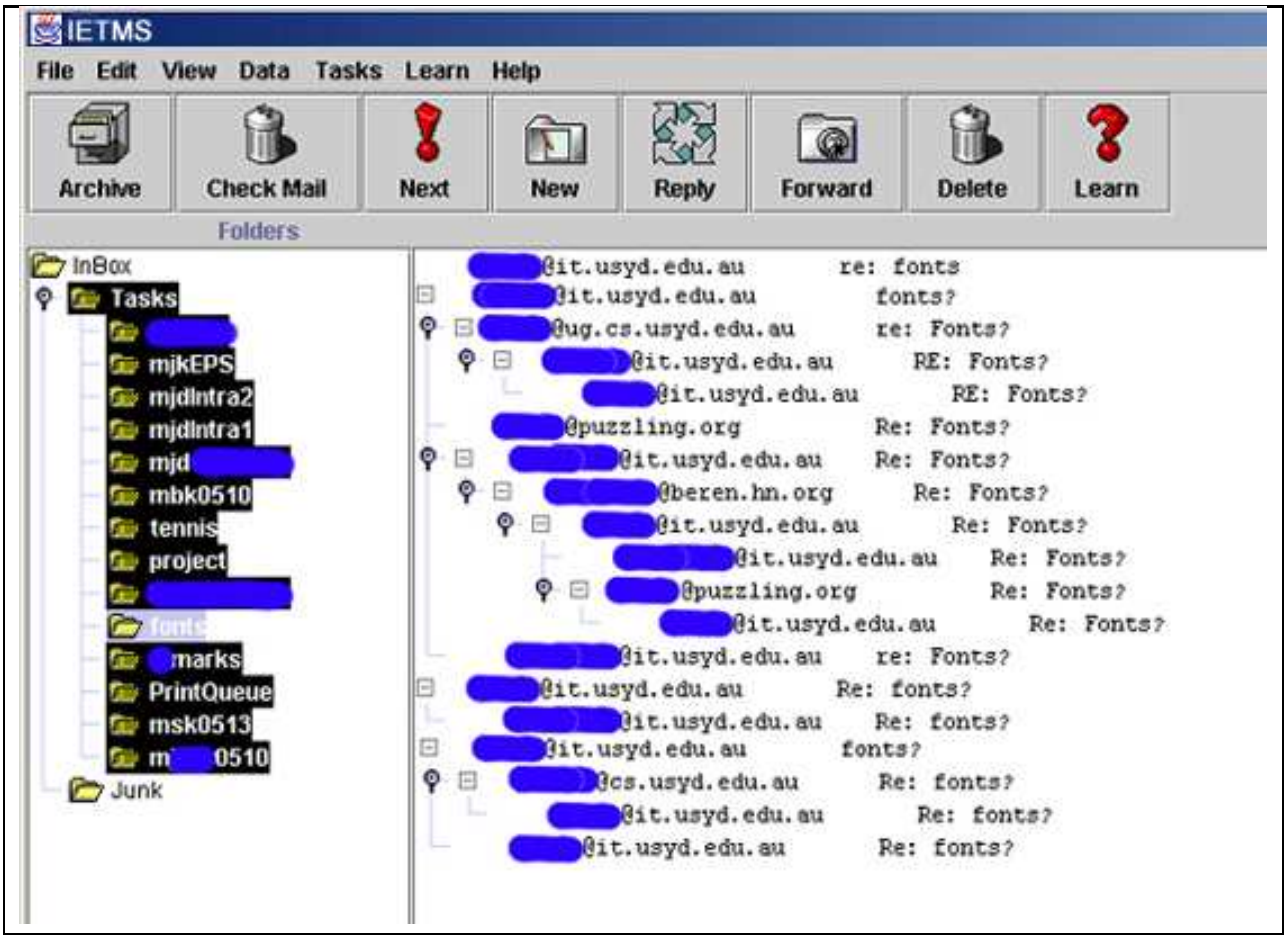

Figure 2: Tasks and Conversation threads in IETMS. Parts of the screen-shot are hidden for privacy

The interface of IETMS is similar to many mail clients, with a window for the currently selected folder, one for the currently selected mail item in that folder and one for the folder names. Unlike other clients, it separates the notion of a task from the notion of a folder. In the interface shown in Figure 2, tasks appear similar to folders (left frame) but they are colour coded to help users differentiate between a folder and a task. The interface also shows the e-mail messages belonging to *"fonts"* task grouped into threads (right frame). The reply messages are indented to help users keep track of conversations as all the conversational e-mail messages are groupped together.

An important element of the iems interface is that it can use automatically induced rules to pre-sort the inbox. IETMS extends this by distinguishing which incoming e-mail messages are part of a conversation. Then the inbox sorts the e-mails within it, placing mail predicted to be part of each conversation together and other mail is still pre-sorted into the predicted folder.

To predict whether a message belongs to a particular conversation, IETMS collects various sources of evidence. It uses the list of people involved in earlier parts of the conversation as one of the sources of evidence. The more common the list of people on two messages, the greater the likelihood they are part of the same message. When a task is initiated, a list of people involved in that task is stored. It is then used to quickly determine if an e-mail message is from one of the people involved in a task. However, the scope of this method is quite limited as people may be involved in many concurrent tasks.

As described in Section 3.1, *Message-Id*, *References* and *In-Reply-To* MIME headers provide a valuable method for classifying e-mail messages into conversations. To cater for inconsistencies in e-mail clients, the values of these fields are pre-processed before matching. Special characters (@, <, > and .) are removed because some clients insert < and > around the values of these fields while others do not. After pre-processing, the value of *References* or *In-Reply-To* fields, of a new message, is matched against the value of *Message-Id* field of task-related messages. If a match is found, the incoming message is classified into the same task as the matched message.

A related, but different, method is to classify incoming messages using subject field. The subject field of an incoming message is matched against the subject fields of task-related messages. For same reasons as MIME headers, the value of subject fields needs to be pre-processed before matching– *'Re:'* is removed along with non-alphabetical characters. When a match is found, the incoming message is classified into the same task as the matched message.

Finally, a TF-IDF text classification algorithm is used to analyze the body of the message as an additional source of evidence. This method classifies an incoming message by comparing the frequency of terms which are common in the message and the tasks but are rare otherwise. A stop list is used to remove common words (a, am, the, etc) from the body as these words do not give us any insight into the content of the message. This method may be rather limited in general since message bodies will tend to be short. However, as one of a range of evidence sources, it has some potential power.

5 Supporting conversations about Meetings

To evaluate approaches to classifying messages into conversation we decided to focus on one class of tasks. Studies of tasks have showed that *'scheduling a meeting'* is one of the key tasks accomplished using e-mail in organizations. From different types of user tasks explored by Takkinen [8] it is certainly the most interesting and the most complex.

The IETMS interface provides a specialized interface wizard for the meeting scheduling task (Figure 3). This ensures that the meeting initiator considers the standard elements of a meeting: the time, duration, place, people to invite, information to be provided to them, such as a description of the purpose of the meeting.

6 Evaluation Experiments

The success of IETMS was measured in its ability to classify e-mail messages into tasks and folders and its ability to group task related messages into threads. It involved measuring the accuracy (precision and recall) of various classification and threading algorithms. It also involved doing a user-by-user analysis to determine the significance of limitations (described in section 3.3) on classification algorithms. We now present the experimental setup and results of these experiments.

Figure 3: A meeting wizard to help user initiate a meeting

Task Number	Number of Messages	Number of Users	Initiated	Completed
1	84	8	20 Feb 2002	24 Oct 2002
2	30	4	4 Apr 2002	10 Apr 2002
3	6	2	19 Mar 2002	19 Mar 2002
4	34	5	4 May 2002	6 May 2002
5	18	5	8 May 2002	3 Jul 2002
6	6	3	8 May 2002	10 May 2002
7	3	3	10 May 2002	10 May 2002
8	5	3	10 May 2002	13 May 2002
9	27	4	10 May 2002	18 Jun 2002
10	16	6	11 May 2002	20 May 2002
11	18	12	22 May 2002	20 Jun 2002
12	37	7	14 Aug 2002	29 Aug 2002
13	25	11	22 Sep 2002	24 Sep 2002
14	13	5	24 Sep 2002	28 Sep 2002
Total:	322	78		

Table 1: Details of the tasks in testing data

6.1 Experimental Design

We were able to make use of a small collection of e-mail. These all involved the task of organizing meetings within a tertiary institution. The messages were part of a study and protocol analysis of the way that a smart personal assistant might manage the organisation of meetings. Accordingly, a person took the role of organising a series of

meetings on behalf of other individuals who needed meetings organised: mailing the participants, collating replies, reviewing plans and going back to the meeting sponsor. All individuals involved were informed that their mail would be studied in order to improve our understanding of how to build a smart personal assistant for organising meetings. The collection contained 322 task-related e-mail messages involving 50 users. The messages were classified into 14 tasks. The smallest task contained only 3 messages where the largest task contained 84 messages (over 25% of total messages). Table 1 summarizes each of the tasks.

Some conclusions could be drawn about the nature of task related e-mails by analyzing the data given in Table 1. Firstly, the table illustrates that some tasks overlapped each other. There were a number of ongoing tasks during May 2002 to Jul 2002 period. Interestingly, Task 1 overlapped all the other tasks. Secondly, the length of tasks from initiation to completion varied significantly. The length of tasks 4, 6 and 12 was just 2 days where the length of Task 1 was over 8 months. Thirdly, some tasks did not have a great deal of communication between participants. In task 7 all three e-mail messages were sent by three different participants, where as, in task 1, over 80 messages were exchanged between eight participants.

6.2 Results

A 'Sliding Window' tester was implemented with varying training window size and a fixed testing window of size 1. Given the nature of conversational e-mail messages, list of e-mail messages [1..N] is chronologically ordered. So we chose $j < N$ and trained on the set [1..j] in order to be able to test on the (j+1)th message. This process was repeated as the size of training set was increased by 1 and the test window was shifted.

In Figure 4, accuracy of a MIME-header based classification algorithm is presented. The average precision of this algorithm was very high (94.39%) as it classified a message if and only if its *References* or *In-Reply-To* fields referred to *Message-Id* field of one of the pre-classified messages. Only 3 messages were ever incorrectly classified (1 in task 3 and 2 in task 14). These features were due to users employing a single e-mail message to respond to multiple tasks.

On the other hand, average recall was very low (only 35.23%) as it missed a lot of messages due of limitations described in section 3.3. For example, two of the participants (users 2 and 8) in Tasks 1 and 4 used a non-MIME compliant e-mail client. As a result the recall for these tasks was about 6% because the mail client did not copy the *Message-Id* of the original message in *References* or *In-Reply-To* fields of the reply message.

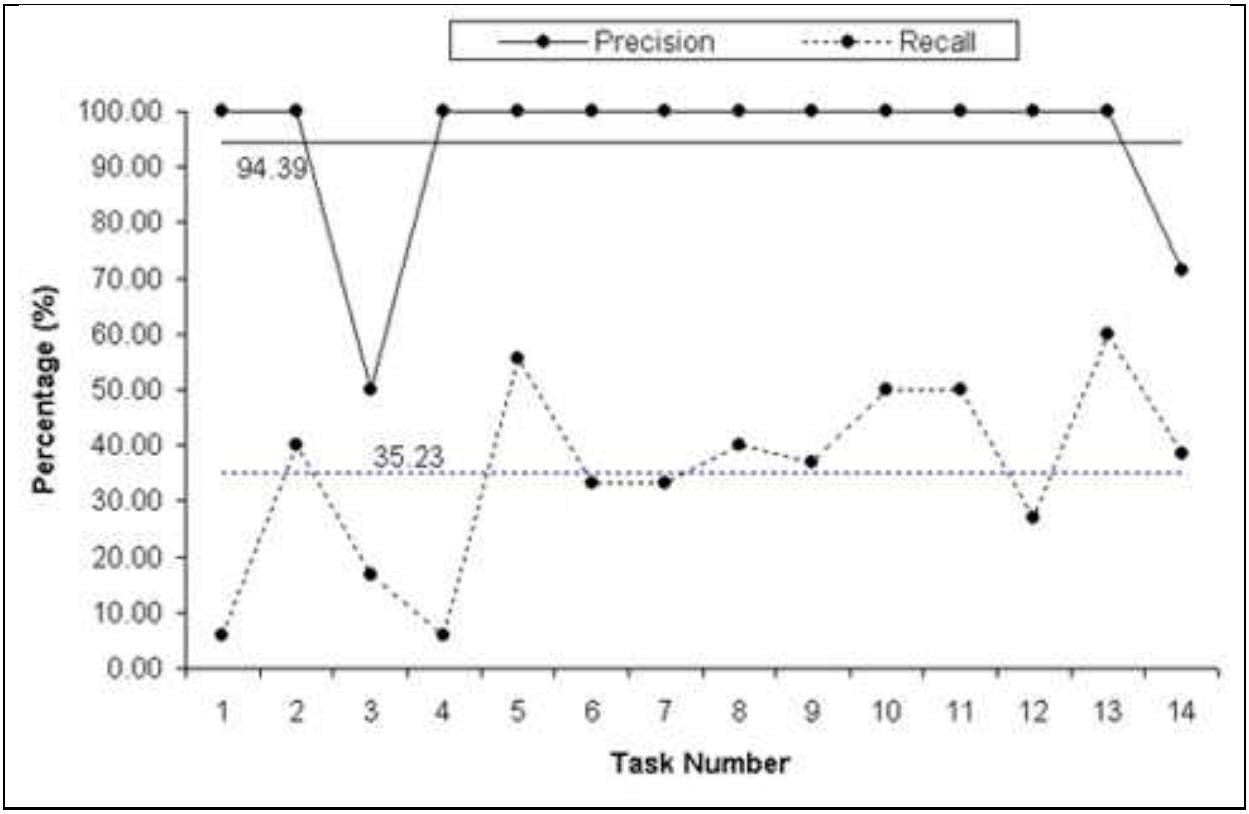

Figure 4: Accuracy of Message-Id based Classifier

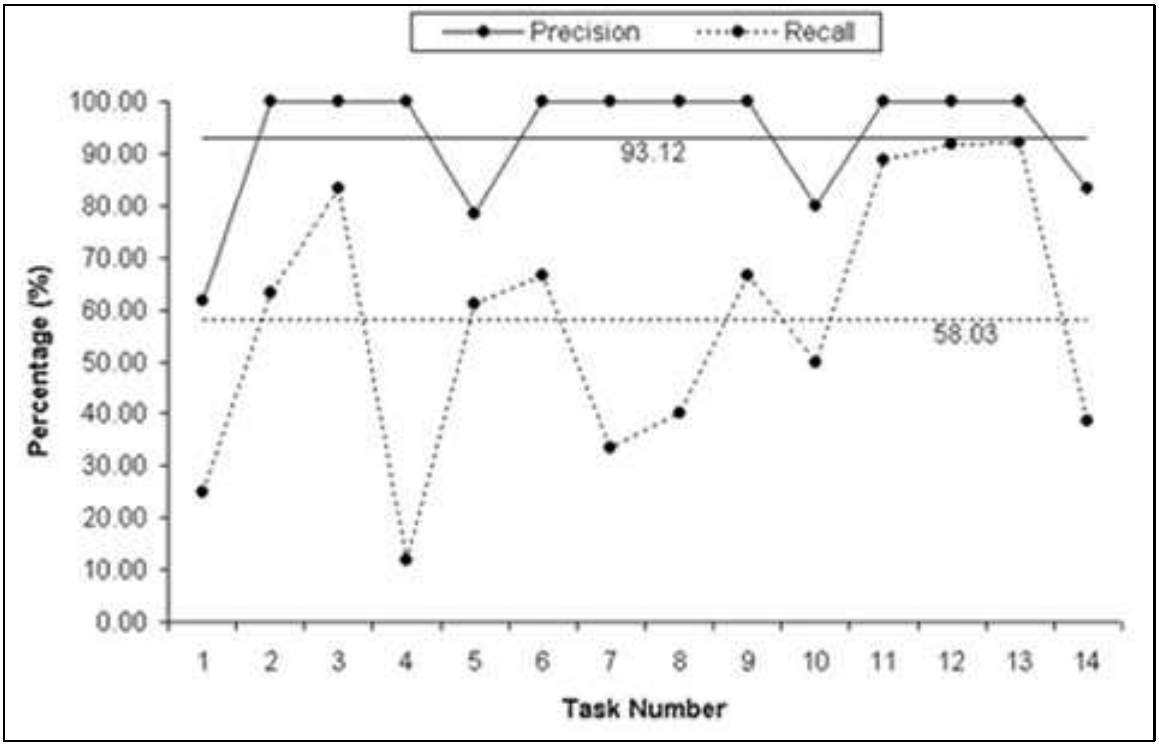

Figure 5: Accuracy of Subject based Classifier

Figure 5 presents the accuracy of a Subject based algorithm. The average precision of this algorithm was high (93.12%) as it classified a message if its subject, after removing *'re:'*, matches the subject of a pre-classified message. In tasks 5 and 10 a total of 5 messages were incorrectly classified; task 5 was a meeting about 'School's Intranet site' and task 10 was a follow-up meeting on the same topic, so both tasks had messages with common subjects. Messages in task 1 had some generalized subjects (eg *'meeting'*) and messages belonging to other categories with same subject were incorrectly classified in task 1.

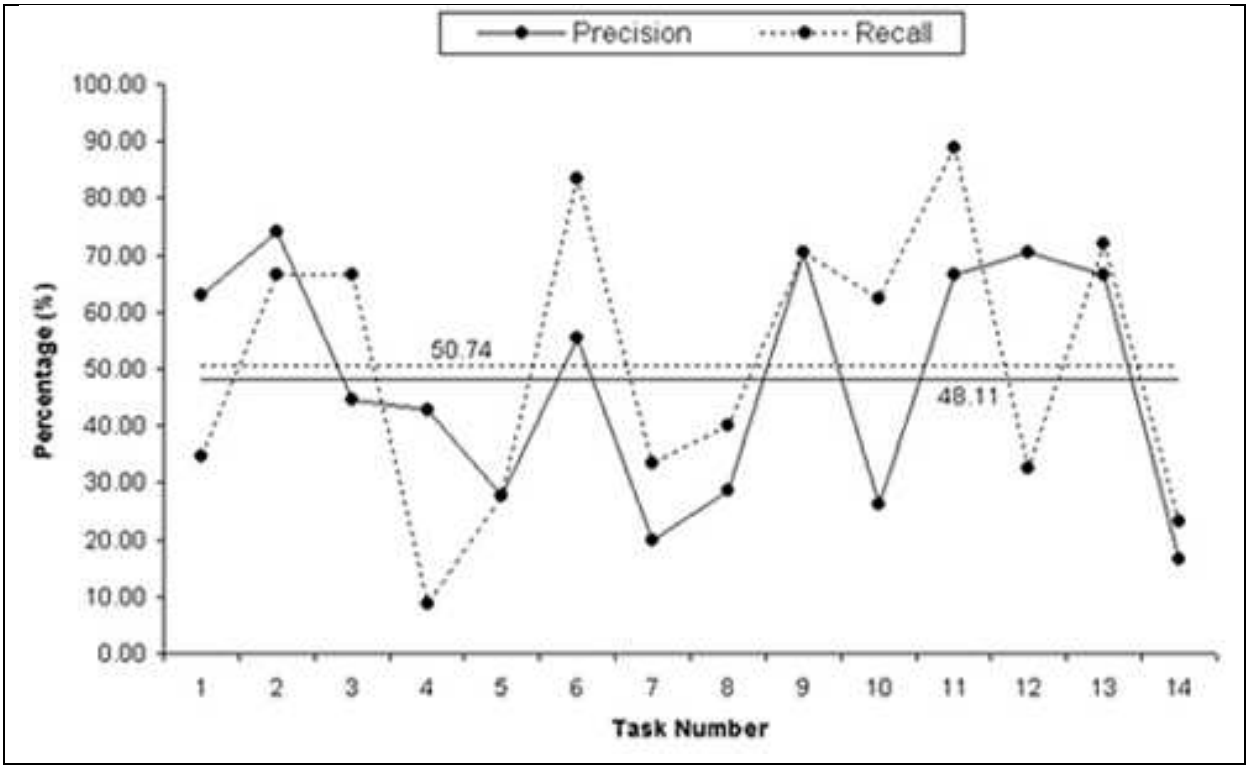

Figure 6: Accuracy of TF-IDF based Classifier

The average recall of 58.03% for Subject based algorithm was an improvement on MIME-header based algorithm, but it still missed a few messages. The reason for this was that users manually edited the subject line of an e-mail message before sending it. In some cases, users sent a fresh message rather than using the reply feature of the e-mail client–the subject line was changed when typing the fresh message. The recall for task 4 was very low (only 11.76%) because messages were one-way interactions where the initiator sent e-mail messages to inform other participants about various facts about the meeting. Task 1 involved many interactions consisting of only a few messages (only 2 or 3 messages). All these interactions had different values for subject fields and thus the Subject-based algorithm was not able to classify them correctly.

The accuracy figures of a TF-IDF based algorithm, which analyzed the body of messages, are presented in Figure 6. The average precision of this algorithm was quite low (48.11%). The biggest problem was that some of the tasks shared common terms which caused this algorithm to classify messages incorrectly. As stated above, tasks 5 and 10 shared a common subject which meant that messages belonging to these tasks contained similar terms (eg 'intranet') which were rare among other tasks. As a result, the precision of about 26% was achieved in both cases. Similar problems also caused the precision of tasks 7 and 14 to be quite low (about 18%).

The TF-IDF based algorithm worked best for tasks 11 and 13. In both cases, the e-mail messages contained distinct terms which were not common in any of the other tasks ('ADCS' was the dominating term for task 11 and 'tennis' for task 13). The body of a message may contain content relevant to multiple tasks as people may employ a single e-mail message to reply to multiple messages. Surprisingly, using TF-IDF algorithm the recall for task 1 was highest as compared with recall for the same task using other algorithms proving that this method had some potential power in classifying task-related messages.

MIME-header and Subject based algorithms classified messages with high precision but they tend to have a fairly low recall values. On the other hand, the TF-IDF based algorithm classified messages with moderate precision and recall values but it improved the accuracy for some tasks (eg tasks 1 and 6). In order to combine the strengths of these three algorithms a hybrid algorithm was implemented. This hybrid algorithm, firstly, attempted to classify messages using MIME-header and Subject based algorithms. If a suitable task was not found, it then used the TF-IDF algorithm. The accuracy of this algorithm is presented in Figure 7. The accuracy figures for some tasks, which showed poor precision and recall values (eg tasks 4) with other algorithms, improved dramatically.

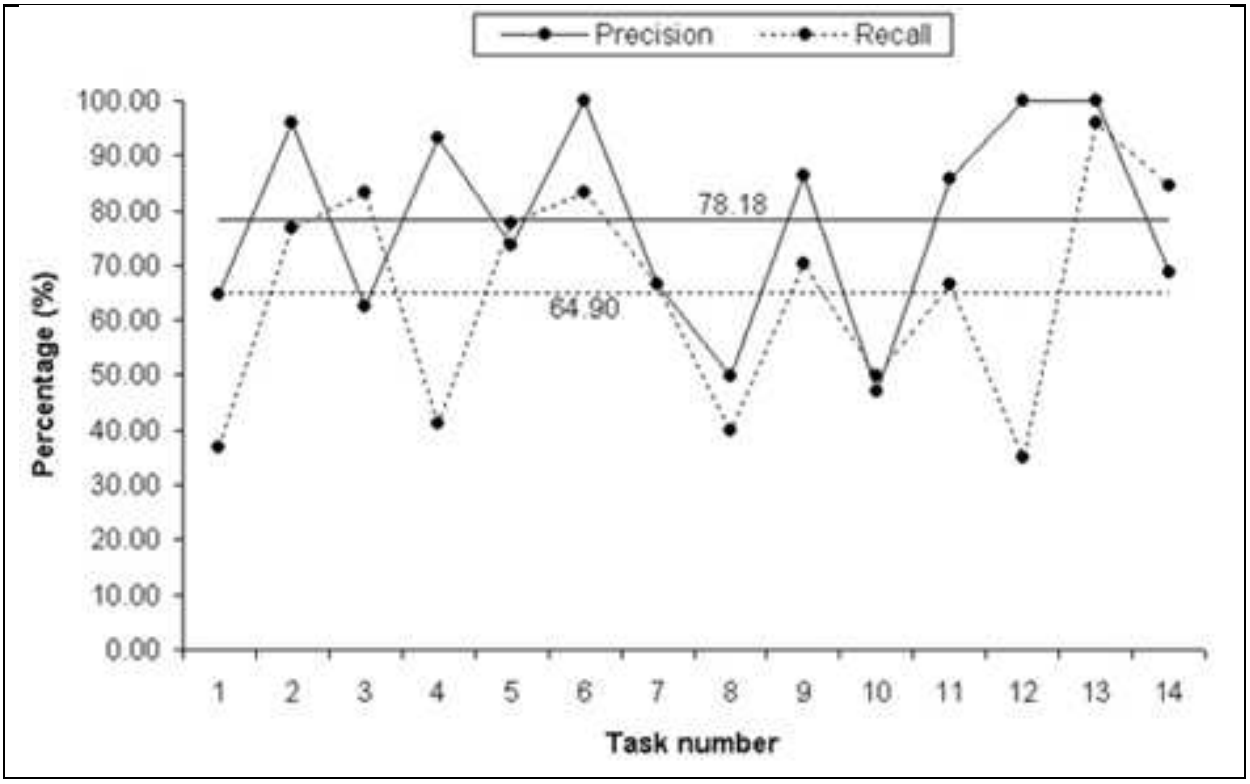

Figure 7: Accuracy of Hybrid Classifier

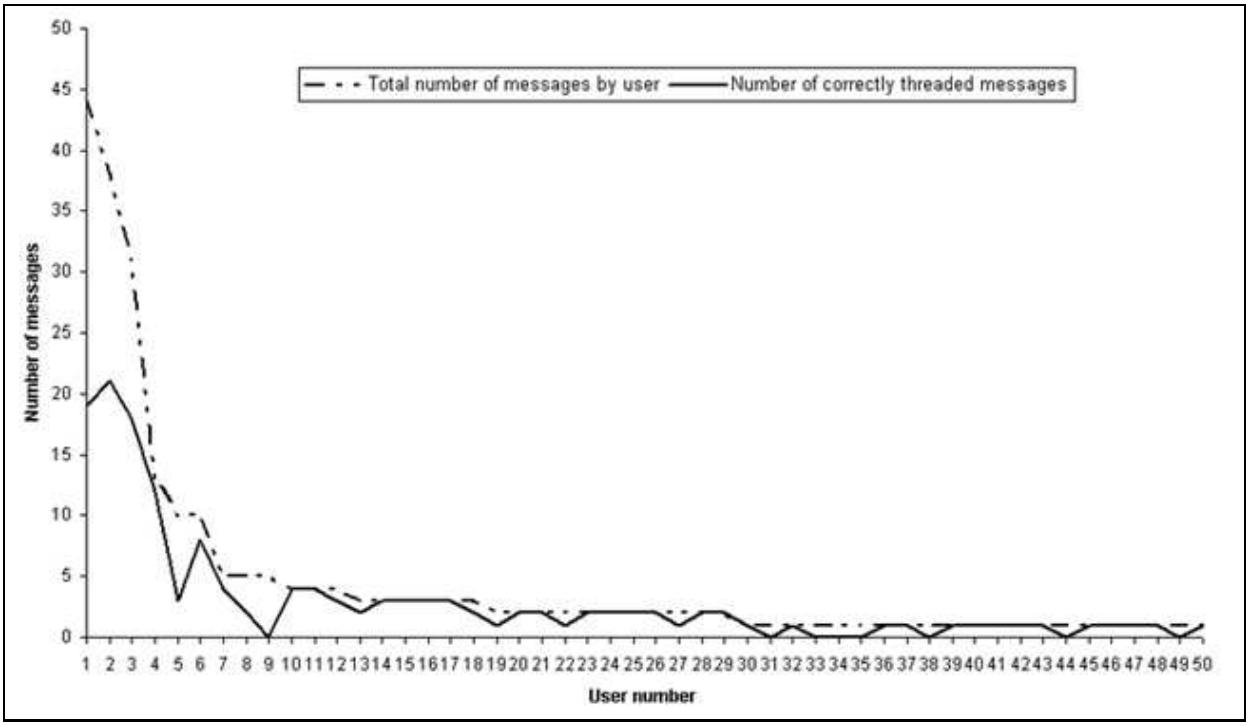

Figure 8: Number of messages sent by a user and number of messages correctly threaded

The results of a user-by-user analysis, using the Hybrid algorithm, are presented in Figure 8. The purpose of this analysis was to explore the potential value of modelling the e-mail addreses which were poorly handled by the Hybrid algorithm. In a deployed system it would be useful to be able to flag messages that the system believed it had poor classification powers. The main problems for the hybrid algorithm follow from user behaviour and the user's choice of mail client. This suggests that marked number of individual users would be responsible for the problems in use of the hybrid algorithm.

User 1 initiated all the tasks and as a result many messages were incorrectly classified for them. Interestingly, users 2 and 8 used non-MIME compliant e-mail clients which meant that MIME-headers could not be used to classify the message sent by them. However, the accuracy figures of 55% and 40% for users 2 and 8 respectively suggested that other parts of the message (subject or body) were able to classify them correctly some of the time. Hence, some of the limitations described in Section 3.3 were oercome by the hybrid algorithm.

The same 'Sliding window' tester was then used with a constant sized training window which was moved through the list of chronologically ordered messages. Figure 9 presents the accuracy figures of the hybrid algorithm with a training window of size 19. In other words, in first window messages numbered [1..19] were used for training and message numbered 20 was used for testing. This process was repeated for all the windows. The average precision of 87.50% and average recall of 91.67% were achieved. But as shown in the figure, precision and recall of most windows were 100% highlighting the fact that users participated in tasks for a short period of time and typical tasks only involved a moderate number of messages (19).

7 Conclusion and Discussion

The goal of this paper was to explore the support for task management using e-mail. This involved using rule-based and machine learning algorithms to automatically classify and prioritize messages into tasks and folders.

A thorough evaluation of these techniques was carried out using a corpus of mail which contained a collection of conversations about meetings. The results of this evaluation showed that while keeping the attractive properties of e-mail, including:

- easy of use,
- wide acceptance, and
- asynchronous nature

new and improved uses of e-mail, *e.g. task management*, could be encouraged by developing specialized applications on top of, otherwise, unstructured e-mail domain.

We also improved our understanding of the prevalence of user behaviours that thwart such applications. The value of modelling individual user's e-mail behaviours was also explored, as was the value of adapting classification mechanisms to changes in these behaviours.

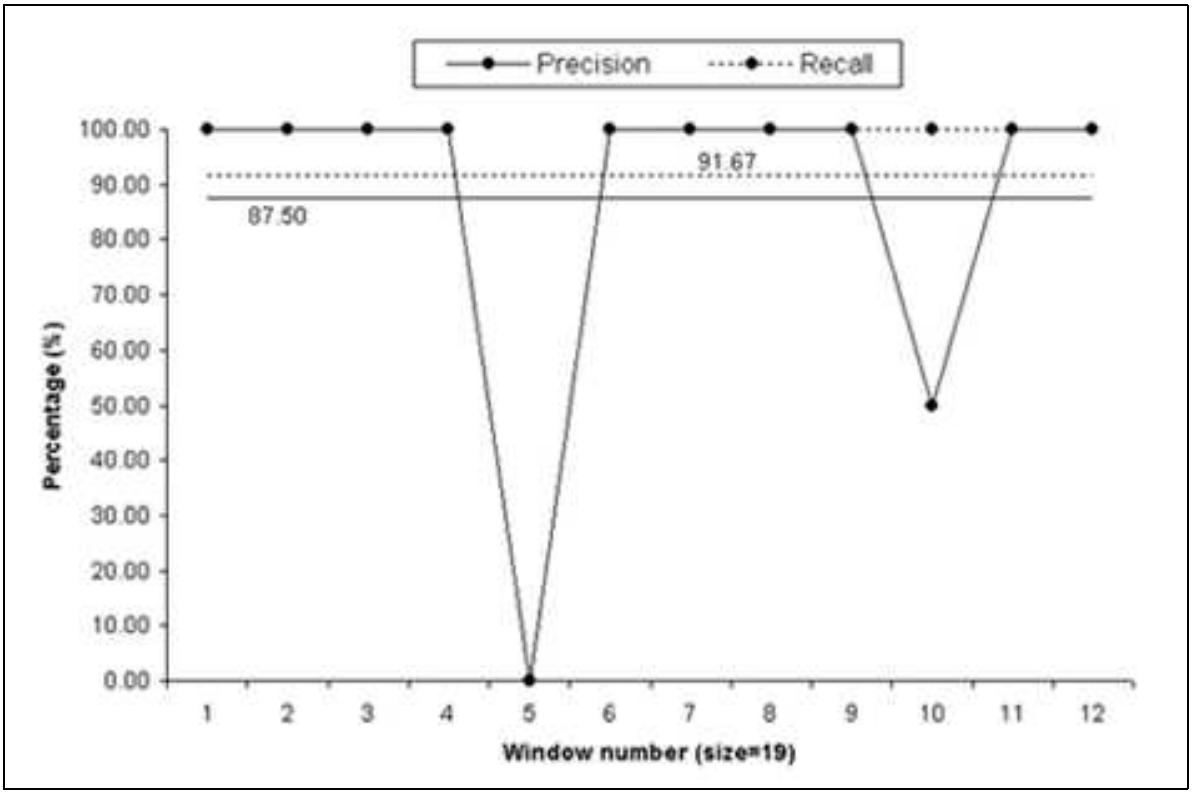

Figure 9: Accuracy for a constant sized window

8 Acknowledgement

We would like to acknowledge Sarah Kummerfeld, Josiah Poon and Kalina Yacef for allowing us to use the data set, compiled for a separate project.

References

[1] D. Comer and L. Peterson. Conversation-based mail. *ACM*, Volume 4, Number 4, pages 299–319, November 1989.

[2] Kay J. Crawford E. and McCreath E. Automatic induction of rules for e-mail classification. In *Proceedings of the Sixth Australiasian Document Computing Symposium*, December 2001.

[3] Kay J. Crawford E. and McCreath E. An intelligent interface for sorting electronic mail. *A short paper in IUI'02*, January 2002.

[4] S. Fleming and A. Kilgour. Electronic mail: Case study in task-oriented restructuring of application domain. In *Proceeding of IEEE: Computers and Digital Techniques*, number 2, pages 65–71, March 1994.

[5] Freed N. and Borenstein N. Rfc2046 multipurpose internet mail extensions (mime) part one: Format of internet message bodies, November 1996.

[6] Freed N. and Borenstein N. Rfc2046 multipurpose internet mail extensions (mime) part two: Media types, November 1996.

[7] Resnick P. Rfc2822 internet message format, April 2001.

[8] J. Takkinen. *From Information Management to Task Management in Electronic Mail.* Ph.D. thesis, Department of Computer and Information Science, Linkopings universitet, SE-581 83 Linkoping, Sweden, 2002.

[9] J. Takkinen and N. Shahmehri. Coordination in message-based environments: Restructuring e-mail to accomplish tasks. In *Proceedings of the CTIS98, the International Workshop on Coordination Technologies for Information Systems*, pages 566–571, August 1998.