Email Thread Identification and Management

Priti Kulkarni, Haridas Acharya

Abstract: Nowadays, Email communication is use as primary communication tool in the business domain as well as in education sector. Due to massive incoming emails, overflowing inbox is one of the problems faced by email users. There are several reasons for such a situation, one of them being the unnecessary mass of thread emails. They are retained in inbox even when they are not necessary. Even if this email is deleted from inbox, the next message as thread email will hit your inbox. Wrong use of ‘reply-all’ tab adds to this situation called “Email storm”. Thread emails are often generated because of users’ careless habit to click on ‘Replyall’ button. It is almost like a reflex action on their part. This work intends to solve the problem of email storm on two fronts:

- Identification of thread emails
- Automatically controlling thread email

The three datasets Din, Dadm and Dexam from academic domain are used as training data. The experimental outcome shows that ‘In-Reply-To’, ‘References’ and additionally ‘thread-index’ are the dominant features in identifying thread emails. We have used these features to derive thread classification strategy. The developed algorithm is used to test four datasets D1a, D2a, D1b and D2b. Using this method accuracy upto 99.91% is achieved. Further, the paper also suggests access control rights strategy to control email storm. The model is proposed for controlling thread emails in education domain. The control mechanism will help system administrators to control email traffic.

Keyword: Email classification, thread, Reply email, access control, email storm

I. INTRODUCTION

A collection of messages with a common ancestor is usually called a thread. An email thread is an email conversation that starts with a root email, (the beginning of the conversation), and includes all of the subsequent replies and forwards pertaining to that original email [28]. In the educational organisation, emails are sent to a group of students, staff and faculty members, parents to inform common notices and other common information. If each receiver starts replying back to email using “Reply all” button, then it will create a huge traffic, known as ‘Email Storm’. Moreover, all receivers may not be interested in receiving all the types of emails. So to overcome this problem, it’s required to control reply to email in case of mass emails.

The work is divided into two phases; the first phase covers the identification of email header features to classify email as thread email. In the second phase, access rights matrix to control email storm with special reference to educational organisation is presented and model is suggested controlling thread emails.

II. LITERATURE REVIEW

There are some incidences indicating need of controlling thread emails. The New York University (NYU) students accidentally hit reply all button in November 2012. Due to this action, there were 39,979 subscribed addresses affected. Since students could send an email to every single student at NYU, this problem of getting reply “not to reply all” arised [23]. University College London (UCL) faced email storm of around 3000 messages which include spam messages and comments from students, using the #bellogate hashtag reached over 26000 students. A sender has sent email pretending to be the provost [26]. Next, on technical front there are different approaches for thread identification,

- a. Use of subject field [14]
- b. Use of body contents field [29][2]
- c. Use of header field [8]

When email body contents are used, emails with similar topics are grouped together and reconstructing tree structure of email conversation. Grouping of thread email can be done by parent-child relationship. The identified thread emails can be further used to form thread clusters. Thread emails are used for various purposes, conversation thread detection are useful to detect discussion in mailing list [13], topic detection [33], email summarization [9], email classification, question answering,[18][19], visualization [15] [17]. Thread grouping enhance awareness of others' contributions on a topic, and minimize lost messages by clustering related e-mail [4]. The Gmail API uses thread resources to group email replies with their original message into a single conversation or thread. This allows you to retrieve all messages in a conversation, in order, making it easier to have context for a message or to refine search results. Email Threading greatly reduces the time and complexity of reviewing emails by gathering all forwards, replies, and reply-all messages together [12]. In [29] authors have applied text matching techniques to the text of body part to detect threads effectively. But they fail to detect all conversations. In [14] Authors have group messages with the same subject and among the same group of people. But a conversation did not cover all the participants in all the messages. In [10] authors used clustering techniques to group messages by topic. Messages with the same topic are grouped by comparing subject, date, participants, and content features. Erera and Carmel [3] have used subject, date and body to apply clustering to find similar thread emails. [9,11] summarize email thread topics using topic summarization. [2,32] used semantic content, the social interactions and the timestamp to identify thread emails. To reconstruct all conversation [8] use headers base rule and topic-based heuristic from emails. Subject of message and group email with similar subject have considered [6]. Thread messages are cluster into groups [31].

Revised Manuscript Received on December 15, 2019.

Priti Kulkarni, Assistant Professor, Symbiosis International (deemed University) Pune (Maharashtra) India.
Haridas Acharya, Professor, Alama Institute of Management, Pune (Maharashtra) India.
Our paper integrates two concepts, first is thread identification and second is to control thread email sent to mass users which results into email storm.

III. DATA COLLECTION

The separate program in python was developed to extract emails from inbox. A training datasets from education domain $D_{in}$ (size=6942), $D_{adm}$ (size=1114), $D_{exam}$ (size=3507) were designed based on domain knowledge. The four datasets $D_{in}$, $D_{adm}$, $D_{exam}$ were used for testing purposes. The five feature selection techniques and three classifiers are applied on datasets. Following feature set $F(s)$ is used to find most significant features.

$$F(s) = \{\text{Authentication-Results, bcc, cc, Content-Disposition, Content-Type, Date, DKIM-Signature, From, In-Reply-To, List-Archive, List-Help, List-ID, List-Owner, List-Post, List-Software, List-Subscribe, List-Unsubscribe, Mailing-List, Message-ID, Precedence, Received, Received-SPF, References, Reply-To, Resent-bcc, Resent-cc, Resent-Date, Resent-From, Resent-Message-ID, Resent-Reply-To, Resent-To, Return-Path, Subject, Thread-Index, Thread-Topic, To, X-Mailer}\}$$

The feature set of 37 features are used for further analysis.

IV. EXPERIMENTAL SETUP

The data mining tool Weka has been used for applying machine learning techniques. The feature selection techniques namely Information Gain, Chi-squared, relief, correlation based, wrapper feature selection were applied on datasets. The resulting set of features used as input to classifier Naïve Bayes, Decision tree and KNN to find effect on accuracy on five datasets. All runs of experiment are carried out using 10 fold cross validation techniques to test the datasets.

Table 1- No of features Vis-à-Vis accuracy of classifier on Din dataset

<table>
<thead>
<tr>
<th>Feature selection techniques</th>
<th>$D_{in}$ (Accuracy in %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No of FS</td>
<td>NB</td>
</tr>
<tr>
<td>Chi squared</td>
<td>21</td>
</tr>
<tr>
<td>Correlation-based Feature Subset Selection</td>
<td>4</td>
</tr>
<tr>
<td>Information gain</td>
<td>20</td>
</tr>
<tr>
<td>Relief attribute evaluator</td>
<td>21</td>
</tr>
<tr>
<td>Wrapper subset eval+DT</td>
<td>3</td>
</tr>
</tbody>
</table>

Where, NB=Naïve Bayes, DT=Decision tree, KNN=K-nearest neighbour

Table 2-No of features Vis-à-Vis accuracy of classifier on Dexam dataset

<table>
<thead>
<tr>
<th>Feature selection techniques</th>
<th>$D_{exam}$ (Accuracy in %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No of FS</td>
<td>NB</td>
</tr>
<tr>
<td>Chi squared</td>
<td>25</td>
</tr>
<tr>
<td>Correlation-based Feature Subset Selection</td>
<td>4</td>
</tr>
<tr>
<td>Information gain</td>
<td>25</td>
</tr>
<tr>
<td>Relief attribute evaluator</td>
<td>22</td>
</tr>
<tr>
<td>Wrapper subset eval+DT</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 3 No of features Vis-à-Vis accuracy of classifier on Dadm dataset

<table>
<thead>
<tr>
<th>Feature selection techniques</th>
<th>$D_{adm}$ (Accuracy in %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No of FS</td>
<td>NB</td>
</tr>
<tr>
<td>Chi squared</td>
<td>20</td>
</tr>
<tr>
<td>Correlation-based Feature Subset Selection</td>
<td>3</td>
</tr>
<tr>
<td>Info gain</td>
<td>20</td>
</tr>
<tr>
<td>Relief attribute evaluator</td>
<td>19</td>
</tr>
<tr>
<td>Wrapper subset eval+DT</td>
<td>1</td>
</tr>
</tbody>
</table>

Wrapper feature selection generates a minimum one feature for $D_{adm}$ & $D_{exam}$ dataset and three features for $D_{in}$ dataset. These features include, In-Reply-To, References and Thread index. But wrapper method requires classifier as input. So the number of features resulted are depend on the input classifier. As shown in table 1, these three features with the decision tree classifier show accuracy of 99.82%. Correlation-based feature subset selection technique generated four features.

Fig.1 shows the methodology adopted in the current work

V. EXPERIMENTAL RESULT

The experimental result are shown in the table1, table2, table3
KNN classifier shows accuracy (99.88%) with four features. All three classifiers show optimum performance with 4 features, In-Reply-To, References, Reply-To, Thread topic. For dataset, $D_{adm}$ and $D_{exam}$ the decision tree classifier performed well with chi-squared as feature selection technique giving an accuracy of 100%. The performance of a decision tree is constant with a maximum of 25 features as well with minimum 1 feature.

Following Table 4 shows the comparison of minimum features generated by feature selection techniques on four datasets. It is observed from the above table that, minimum 1 to 4 numbers of features are generated by all feature selection techniques. Among all, In-reply-to and Reference are common features generated by all feature selection techniques. The result of the above experiment is verified with actual dataset to find relationships among the features. According to our dataset Din, three features In-reply-To, Reference, Thread-Index are the most significant features for thread classification. Reply-to specifies where the sender replies to go, overrides from field.

<table>
<thead>
<tr>
<th>Dataset Name and Size</th>
<th>Feature selection technique</th>
<th>List of minimum number of Features generated</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No of Features</td>
<td>In-Reply-To</td>
</tr>
<tr>
<td>Din, Size=6942</td>
<td>CBFS</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>WFS</td>
<td>3</td>
</tr>
<tr>
<td>Dadm, Size=1114</td>
<td>CBFS</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>WFS</td>
<td>1</td>
</tr>
<tr>
<td>Dexam, size=3506</td>
<td>CBFS</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>WFS</td>
<td>1</td>
</tr>
</tbody>
</table>

### VI. PROPOSED APPROACH FOR THREAD EMAIL IDENTIFICATION

As explained in the section 4 (see table 4) feature selection techniques generates subset of features. It is observed that three features are most dominant features for thread email classification. With study of these features and its presence in our dataset, following algorithm is implemented in python programming language.

**Step 1:**  Step 1: Extract email header features from email $F(s) = \{f_1, f_2, f_3, \ldots, f_m\}$

```
// Output: Fmin(s) = \{f_1, f_2, f_3, \ldots, f_m\} where Fmin(s) subset of F(s)
```

**Step 2:**  Step 2: Check if any of the following conditions is true

i. Check if ‘reference’ feature is not empty and thread-index is empty
ii. Check if thread-index is not empty and In-Reply-To is not empty
iii. Check if ‘In-Reply-To’ is not empty and reference is equal to empty
iv. Check if ‘In-Reply-To’ is empty and reference is not equal to empty and thread-index is not empty

**Step 3:**  Step 3 is true, classify email as thread

**Step 4:**  If step 2 is false, classify email as No-Thread

**Step 5:**  End

The algorithm is tested by using four datasets. When the above approach is applied on a $D_{tri}$ testing dataset, it provides 99.91% accuracy with three features. The results are shown in table 5. The obtained classification result is cross verified with actual email record in the datasets. Table 5 shows the accuracy of algorithm on four datasets. The accuracy of minimum 97.04% is achieved. When feature selection techniques are used on $D_{CS}$, its result shows “In-Reply-To” and “references” as the minimum number of features, derived from a wrapper feature selection technique.

### Table 5 Accuracy of Algorithm on four datasets

<table>
<thead>
<tr>
<th>Dataset used</th>
<th>Size of dataset</th>
<th>correctly classified</th>
<th>Incorrectly classified</th>
<th>Accuracy in %</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_{tri}$</td>
<td>2448</td>
<td>2446</td>
<td>02</td>
<td>99.91</td>
</tr>
<tr>
<td>$D_{r2}$</td>
<td>742</td>
<td>720</td>
<td>22</td>
<td>97.04</td>
</tr>
<tr>
<td>$D_{cs}$</td>
<td>2270</td>
<td>2207</td>
<td>63</td>
<td>97.22</td>
</tr>
<tr>
<td>$D_{as}$</td>
<td>4318</td>
<td>4313</td>
<td>05</td>
<td>99.88</td>
</tr>
</tbody>
</table>
VII. PROPOSED EMAIL POLICY TO CONTROL EMAIL STORM

Email storm occurred when multiple members in the distribution list starts replying at the same time to the entire list. To prevent this situation we have proposed here access rights according to various stakeholders in educational organisation.

Table 6 Email access right policy for thread email

<table>
<thead>
<tr>
<th>Thread access control type</th>
<th>Send</th>
<th>Receive</th>
<th>Reply all</th>
<th>Forward</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TAC1</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>All Access rights, can be given to top positional heads</td>
</tr>
<tr>
<td>TAC2</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Send, receive and reply email but do not forwards</td>
</tr>
<tr>
<td>TAC3</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Send, receive and forward but reply all is disable</td>
</tr>
<tr>
<td>TAC4</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Emails can send and receive but do not forward it and reply all is disable</td>
</tr>
<tr>
<td>TAC5</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>People who just designated to send emails. Email Receiving address may be different from sending address</td>
</tr>
<tr>
<td>TAC6</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Access rights only to sent and forward email.</td>
</tr>
<tr>
<td>TAC7</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Access rights to just receive email, reply and forward it. These rights can be possible if email sending account and receiving account are different.</td>
</tr>
<tr>
<td>TAC8</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Access rights to just receive email and forward it. These rights can be possible if email sending email account and receiving account are different.</td>
</tr>
</tbody>
</table>

Following table shows an example of how access policy rights can be implemented in the educational organization at various job roles. A sample set of control is listed in the table below; one can set access rights according to individual need.

Table 7 Thread access right policy for education domain

<table>
<thead>
<tr>
<th>Position holder</th>
<th>Send</th>
<th>Receive</th>
<th>Reply all</th>
<th>Forward</th>
<th>Access Policy type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chancellor</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>TAC1</td>
</tr>
<tr>
<td>Vice Chancellor</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>TAC1</td>
</tr>
<tr>
<td>Dean (Academics/Faculty)</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>TAC1</td>
</tr>
<tr>
<td>Registrar</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>TAC1</td>
</tr>
<tr>
<td>Finance officer_University</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>TAC3</td>
</tr>
<tr>
<td>Examination_University</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>TAC1</td>
</tr>
<tr>
<td>Research Head_University</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>TAC1</td>
</tr>
<tr>
<td>Admission_University</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>TAC3</td>
</tr>
<tr>
<td>HR_University</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>TAC1</td>
</tr>
<tr>
<td>IT department_University</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>TAC1</td>
</tr>
<tr>
<td>Library_University</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>TAC3</td>
</tr>
<tr>
<td>Director_Institute</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>TAC1</td>
</tr>
<tr>
<td>Deputy Director_Institute</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>TAC1</td>
</tr>
<tr>
<td>Head of department</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>TAC1</td>
</tr>
<tr>
<td>Library_Institute</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>TAC3</td>
</tr>
<tr>
<td>Finance_Institute</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>TAC3</td>
</tr>
</tbody>
</table>
A. Proposed Model for thread Classification

The email policy explained in section 5 is integrated with email classification. The fig.2 shows proposed email classification model.

**Working of model:**
1. When an email hits the email server of the organisation first it will be classified as spam or ham
2. Spam email will be quarantined and send it to spam folder.
3. At the next level, Email sender (From) and receiver (To) features will be checked and verified against email policy set as described in table 7. It will classify email as “Accept” or “Reject” as per policy criteria. If email is rejected, email will be sent to the concerned authority for scrutiny.
4. If email is accepted, it will be directly sent to the respective recipient of email.

Fig 2. Proposed thread group email classification model for education enterprises

**REFERENCES**

4. Steven L. Rohall and Dan Gruen, ReMail: A Reinvented Email Prototype, CSCW’02
20. Kelly Fiveash, 19 Sep 2013 retrieved from https://www.theregister.co.uk/2013/09/19/cisco_reply_all_email_waste_s_tons_of_man_hours/
25. https://hbr.org/2012/02/stop-email-overload-1

AUTHORS PROFILE

Priti Kulkarni, Assistant professor at Symbiosis International (deemed University). She has 17+ years of teaching experience. Her research areas are Data mining, text mining and computer network. She has published various research papers in the national and international journal.

Haridas Acharya, Professor at Allana Institute of Management, Pune. He has more than 30 years of experience. He has expertise in the area of Agro Information Systems and data Analytics. He is research Guide for Phd students. He has published various research papers in national and international journals in the area of data mining, NLP, Design of Experiments, Information systems and Engineering computations.