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 D_{cor}, D_{CS}, D_{F1} and D_{F2}. 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

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.

Revised Manuscript Received on December 30, 2019.
* Correspondence Author

Priti Kulkarni*, Assistant professor, Symbiosis International (deemed University) Pune (Maharashtra) India.
Haridas Acharya, Professor, Allana Institute of Management, Pune (Maharashtra) India.

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].

Our paper integrate two concepts, first is thread identification and second is to control thread email sent to mass users which results into email storm.

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_{f1}, D_{f2}, D_{cor}, D_{cs} 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)={ 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.

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.

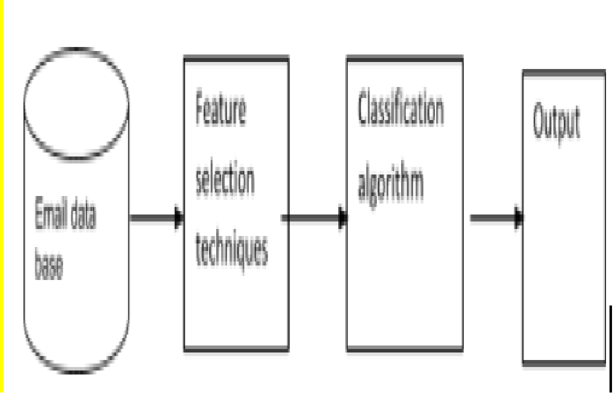

Fig.1 shows the methodology adopted in the current work

EXPERIMENTAL RESULT

The experimental result are shown in the table1, table2, table3

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

Feature selection techniques	Din (Accuracy in %)			
	No of FS	NB	DT	KNN
Chi squared	21	99.29	99.82	99.71
Correlation-based Feature Subset Selection	4	99.49	99.69	**99.88**
Information gain	20	99.37	99.65	99.74
Relief attribute evaluator	21	99.52	99.82	**99.85**
Wrapper subset eval+DT	3	99.55	**99.82**	99.76

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

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

Feature selection techniques	D_{exam} (Accuracy in %)			
	No of FS	NB	DT	KNN
Chi squared	25	99.39	100	99.49
Correlation-based Feature Subset Selection	4	100	100	100
Information gain	25	*	100	99.49
Relief attribute evaluator	**22**	*****	**100**	**99.49**
Wrapper subset eval+DT	1	100	100	100

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

Feature selection techniques	D_{adm} (Accuracy in %)			
	No of FS	NB	DT	KNN
Chi squared	20	97.75	100	98.83
Correlation-based Feature Subset Selection	3	100	100	100
Info gain	20	97.75	100	98.83
Relief attribute evaluator	**19**	98.02	100	98.02
Wrapper subset eval+DT	1	100	100	100

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.

KNN classifier shows accuracy (99.88%) with four features. All three classifiers shows 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 4 List of minimum number of Features generated

Dataset Name and Size	Feature selection technique	List of minimum number of Features generated										
		No of Features	In-Reply-To	References	Reply –To	Thread-topic	Thread-Index	List-id	Date	CC	Message id	Subject
Din, Size =6942	CBFS	4	Y	Y	Y	Y						
	WFS	3	Y	Y			Y					
Dadm, Size= 1114	CBFS	3	Y						Y	Y		
	WFS	1	Y									
Dexam, size= 3506	CBFS	4	Y	Y							Y	Y
	WFS	1	Y									

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)={f1,f2,f3,…….,fn}
//Outpu  Fmin(s)={f1,f2,…fm}  where Fmin(s) subset of F(s)
Step2:   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 2 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_{F1} 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

Dataset used	Size of dataset	correctly classified	Incorrectly classified	Accuracy in %
D_{F1}	2448	2446	02	99.91
D_{F2}	742	720	22	97.04
D_{cor}	2270	2207	63	97.22
D_{cs}	4318	4313	05	99.88

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 indicates the details. Here, 'Y' represent access right is granted and "N" indicates access right is denied.
Any reply to e-mail communication, generated by using the "Reply-all" button will be ignored or rejected.
Thread email policy to send and receive emails can be set with the following type of rights,

Table 6 Email access right policy for thread email

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

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

Position holder	Send	Receive	Reply all	Forward	Access Policy type
Chancellor	Y	Y	Y	Y	TAC1
Vice Chancellor	Y	Y	Y	Y	TAC1
Dean (Academics/Faculty)	Y	Y	Y	Y	TAC1
Registrar	Y	Y	Y	Y	TAC1
Finance officer_University	Y	Y	N	Y	TAC3
Examination_University	Y	Y	Y	Y	TAC1
Research Head_University	Y	Y	Y	Y	TAC1
Admission_University	Y	Y	N	Y	TAC3
HR_University	Y	Y	Y	Y	TAC1
IT department_University	Y	Y	Y	Y	TAC1
Library_University	Y	Y	N	Y	TAC3
Director_Institute	Y	Y	Y	Y	TAC1
Deputy Director_Institute	Y	Y	Y	Y	TAC1
Head of department	Y	Y	Y	Y	TAC1
Library_Institute	Y	Y	N	Y	TAC3
Finance_Institute	Y	Y	N	Y	TAC3

Admission_institute	Y	Y	N	Y	TAC3
IT department_institute	Y	Y	Y	Y	TAC1
Administration_institute	Y	Y	N	Y	TAC3
Examination_institute	Y	Y	Y	Y	TAC1
Faculty	Y	Y	N	Y	TAC3
Student	Y	Y	N	N	TAC4

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.

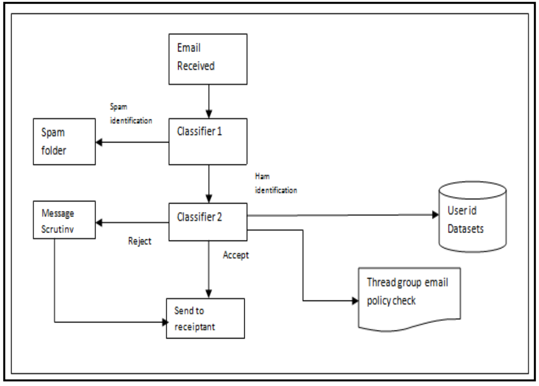

Fig 2.Proposed thread group email classification model for education enterprises

CONCLUSION AND FUTURE SCOPE

Thread identification and management of email threads are very important for smooth functioning of an educational organisation. We have identified minimum number of header features necessary to identify thread emails. The result shows that minimum one feature and maximum four features are required for optimum thread identification. The 'In-reply-To' and 'References' are two dominant features. Additionally, thread-index feature also help for thread identification. These features were used to derive rule base approach for thread identification. The algorithm is tested on four datasets showing accuracy upto 99.91%. To identify thread email automatically, the organisation can set user access control policy, specifying who can use "Reply all" button of email. The finally, email classification model showing integration of thread email policy with email classification is presented in the paper. To adopt the model it is necessary that organisation should have strong email policy which will help to control email storm.

REFERENCES

1. The Internet Society. RFC 2822 – Internet Message Format. Available at http://www.faqs.org/rfcs/rfc2822.html. 2001
2. Domeniconi, G., Semertzidis, K., Moro, G., Lopez, V., Kotoulas, S., & Daly, E. M. (2016, July). Identifying Conversational Message Threads by Integrating Classification and Data Clustering. In International Conference on Data Management Technologies and Applications (pp. 25-46). Springer, Cham.
3. Erera S and Carmel D. Conversation detection in email systems. In: Proceedings of European conference on information retrieval (ECIR'08), 2008, pp. 498–505
4. Steven L. Rohall and Dan Gruen, ReMail: A Reinvented Email Prototype, CSCW'02
5. Stephen W and Kathy M in Proceeding COLING '04 Proceedings of the 20th international conference on Computational Linguistics Article No. 549
6. Wu, Y., and Oard, D. W. Indexing Emails and Email Threads for Retrieval. In Proceedings of SIGIR 2005, (Salvador,Brazil. 2005).
7. Zhu, W., Song, M. and Allen, R. B. TREC 2005 Enterprise Track Results from Drexel. In Proceedings of the TREC 2005
8. Wang X, Xu M, Zheng N and Chen M. Email conversations reconstruction based on messages threading for multi-person. In: Proceedings of international workshop on education technology and training and international workshop on geoscience and remote sensing (ETTANDGRS'08), 2008, pp. 676–680.
9. Rambow, O., Shrestha, L., Chen, J., & Lauridsen, C. (2004, May). Summarizing email threads. In *Proceedings of HLT-NAACL 2004: Short Papers*(pp. 105-108). Association for Computational Linguistics.
10. Cselle, G., Albrecht, K., & Wattenhofer, R. (2007, January). BuzzTrack: topic detection and tracking in email. In Proceedings of the 12th international conference on Intelligent user interfaces (pp. 190-197). ACM.
11. Zajic, D. M., Dorr, B. J., & Lin, J. (2008). Single-document and multi-document summarization techniques for email threads using sentence compression. Information Processing & Management, 44(4), 1600-1610
12. Jacob P, https://people.dsv.su.se/~jpalme/ietf/message-threading.html
13. Kolla, M., & Vechtomova, O. (2007, July). Retrieval of discussions from enterprise mailing lists. In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 881-882). ACM.)
14. Klimt, B. and Yang, Y. Introducing the Enron Corpus. In Proceedings of the CEAS 2004 (Mountain View, CA. 2004)
15. Kerr B. Thread arcs: An email thread visualization. In: Proceedings of the Ninth annual IEEE conference on Information visualization (INFOVIS'03), 2003, pp. 211–218.
16. Lewis DD and Knowles KA. Threading electronic mail: A preliminary study. Information Processing and Management 1997; 33(2): 209–217
17. Perer A and Shneiderman B. Beyond threads: Identifying discussions in email archives. In: Proceedings of the eleventh annual IEEE symposium on information visualization (InfoVis 2005), 2005, pp. 41–42.
18. Ding S, Cong G, Lin C and Zhu X. Using conditional random fields to extract contexts and answers of questions from online forums. In: Proceedings of the Association for Computational Linguistics (ACL), 2008, pp. 710–718.
19. Hong L and Davison B. A classification-based approach to question answering in discussion boards. In: Proceedings of the 32th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'09), 2009, pp.171–178.
20. Kelly Fiveash, 19 Sep 2013 retrieved from https://www.theregister.co.uk/2013/09/19/cisco_reply_all_email_wastes_tons_of_man_hours/

Retrieval Number: B2992129219/2019©BEIESP
DOI: 10.35940/ijeat.B2992.129219

21. Reply-all E-mail storm hits State Department, 2007 retrieved from https://web.archive.org/web/20090201072613
22. /http://www.boston.com/news/nation/washington/articles/2009/01/11/reply_all_e_mail_storm_hits_state_department
23. Casey Chan,2012 retrived from https://gizmodo.com/5963774/heres-what-happens-when-40000-college-students-realize-they-can-e-mail-all-40000-people-at-once
24. Graham Cluley,2016,"The NHS suffered a massive email storm today DON'T REPLY WHATEVER YOU DO", Retrieved 21 April 2018. https://www.grahamcluley.com/nhs-suffered-massive-email-storm-today/
25. https://hbr.org/2012/02/stop-email-overload-1
26. Wakefield, Lawrence (9 October 2014). "#Bellogate trends after pranksters target UCL students' email". The Guardian. Retrieved 11 May 2018. from https://www.theguardian.com/education/2014/oct/09/-sp-bellogate-ucl-students-email-addresses-leaked-14
27. Perlberg, Steven, 26 August 2015, "Reuters Employees Bombarded With Reply-All Email Catastrophe". Retrieved 13 April 2018. https://blogs.wsj.com/cmo/2015/08/26/reuters-employees-bombarded-with-reply-all-email-catastrophe/
28. Thomas Jackson, Ray Dawson, Darren Wilson, (2001) "The cost of email interruption", Journal of Systems and Information Technology, Vol. 5 Iss: 1, pp.81 – 92
29. Email threading, https://help.relativity.com/9.3/Content/Relativity/Analytics/Email_threading.htm
30. Yeh, J. Y., & Harnly, A. (2006). Email thread reassembly using similarity matching.
31. Bouguettaya, A., Yu, Q., Liu, X., Zhou, X., Song, A.: Efficient agglomerative hierarchical clustering. Expert Syst. Appl.42, 2785–2797 (2015)
32. Zhao, Q. and Mitra, P. (2007). Event detection and visu-alization for social text streams. In ICWSM, Boulder,Colorado, USA, March 26-28, 2007
33. Cowan-Sharp, J. (2009). A study of topic and topic change in conversational threads. Naval Postgraduate School Monterey Ca Dept of Computer Science.

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 30years 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.