Clustering Techniques for Email System

Meena Shewale, Pooja Patil, Pooja Kumbhar, Aishwarya Randive & Sushma Vispute
PCET’s Pimpri Chinchwad College of Engineering, Nigdi, Pune, India.

Abstract – In Human life time plays an important role; they will try to consume time as possible as they can, by considering this fact we are going to develop Email clustering system which will save the time of user by accessing emails in group according to his/her need. Clustering technique can be applied to create groups of similar objects, it has been recognized that clustering emails into meaningful groups can greatly save load to process emails. To accomplish this task we are making use of Label Induction Grouping Algorithm (LINGO). System will create LABEL (Appropriate Cluster name) of emails according to content of email message body, which will help users to access emails in easy and fast way. Finally we have presented the results which gives the accuracy of our system.

Keywords – Email Clustering, Lingo Algorithm, Single Sign-On, Voice

I. INTRODUCTION

Email is best source for communication. Email communication has came up as the most effective and popular way of communication today. People are sending and receiving many messages per day, exchanging files and information through emails. E-mail data is now becoming most frequently used (technique) written communication for many companies. Clustering is a technique of creating group of similar objects The cluster shows the similar emails exchanged between the users and finding the text similarities to cluster the users, we are using the Pattern i.e., the similar words exchanged between the users by considering the different Threshold values, here Threshold value shows the frequency of the words used.

There are many Mail Service Providers on internet today like Yahoo, MSN etc. Mails being one of the most popularly used service by all sector of life, corporate as well as personal to contact each other. And that too with no restriction on location and of course free of cost. Users have mail accounts on different mail servers. One cannot access emails from other mail servers from existing mail accounts.

In today’s world, clustering has become most important part of our lives; we do not have time to go through long list of mails every day. Instead we want a solution that will provide us result with minimum overhead. Clustering means group of interrelated documents. We feel to propose a system for Email Clustering, which will take large amount of Emails as an input and provide us group of cluster of emails that will be related to a particular topic. Hence user interested in one particular message will not have to go through a long list of Emails and will directly view his cluster and the search for appropriate message with minimum amount of time.

A. Drawbacks Of Current System

Creating so many accounts also bring many disadvantages as follow:

- Need to remember different User-Id and Passwords.
- Waste of time creating new sessions of each service providers by logging into their respective domains.
- More waste of Bandwidth and download capacity.

People cannot access mails from different mail server at the same time from a single server. We can overcome these drawbacks by using the solution provided in section V. The cluster shows the similar emails exchanged between the users and finding the text similarities to cluster the users. Hence, we are using the message content to cluster Emails. User can also interact with system using voice commands. This paper summarizes the clustering of emails which is organized as follows: In Section II, we discuss the related work. Section III presents an overview of the LINGO algorithm, then in Section IV Proposed work, Section V Results, Section VI Conclusion, and in the end references for our paper.
II. RELATED WORK

Nowadays a use of internet (www) is increased in tremendous level, because of that user is not able to find his relevant document in available solution quickly. When user fires a query in search text box of search engine, then as response of this query engine provides ranked list of document along with their priority. If the query is general, it is extremely difficult to identify the specific document which the user is interested in. The users are forced to sift through a long list of off topic documents. Moreover, internal relationships among the documents in the search result are rarely presented and are left for the user. One of the alternative approaches is to automatically group search results into thematic groups (clusters).

First introduction of web search clustering Scatter/gather [2] followed Suffix Tree Clustering (STC), implemented in the Grouper system [3] pioneered in using recurring phrases as the basis for deriving conclusions about similarity of documents. MSEEC [4] and SHOC [5] also made explicit use of words proximity in the input documents. Apart from phrases, graph-partitioning methods have been used in clustering search results [6]. When we compared Lingo algorithm with SHOC and STC we get following solution.

- SHOC does not support the language identification step where as Lingo supports language identification with snippets (cluster label).
- SHOC is totally based on suffix arrays where lingo is extended version of SHOC.
- Cluster merging is happen in SHOC lingo wont support cluster merging.
- SHOC can make use of Complete and Continuous Phrase, lingo can make use of Readable cluster label.

In this paper the main focus is on Lingo clustering algorithm, which we believe is able to capture thematic threads in a search result, that is discover groups of related documents and describe the subject of these groups in a way meaningful to a human. Lingo combines several existing methods to put special emphasis on meaningful cluster descriptions, in addition to discovering similarities among documents.

III. INTRODUCTION TO LINGO ALGORITHM

Clustering is the most important aspect in electronic communication nowadays. Because it’s necessary to group large amount of data, a particular group called “cluster”. So it becomes easy for the user to get his data as early as possible. Osinski et al. developed lingo algorithm which is a latest clustering algorithm. LINGO follows ‘description comes first’ approach it means that first the labels are created in this algorithm and then the documents are arranged under that particular label. Unlike earlier approaches where first groups are formed and then the label is created.

A. Background of Lingo

- Vector Space Model
- Suffix Array
- Latent Semantic Indexing And Singular Value Decomposition

Vector Space model is a technique that is useful for finding a particular weight for the relationship between a term and a document. For that “Term Frequency-Inverse Document” Frequency these formula are used. Suffix Arrays are used to detect the substring. In this technique we first find the suffixes of the string and give index to each of it, then we store these suffixes in a sorted order. LSI uses SVD to reduce dimensionality of term document matrix.

Lingo Clustering Algorithm

ALGORITHM 1 - PSEUDO-CODE OF THE LINGO ALGORITHM [7]

\[
D \leftarrow \text{input documents (or snippets)}
\]

**STEP 1: Preprocessing**

For all \( d \in D \) do

- Perform text segmentation of \( d \);
- Detect word boundaries etc.
- if language of \( d \) recognized then
  - Apply stemming and mark stop-words in \( d \);
- End if
- End for

**STEP 2: Frequent Phrase Extraction**

- Concatenate all documents;
- \( P_c \leftarrow \) discover complete phrases
- \( P_t \leftarrow p: \{ p \in P_c \wedge \text{frequency (p)} > \text{Term Frequency Threshold} \};

**STEP 3: Cluster Label Induction**

- \( A \leftarrow \) term-document matrix of terms not marked as stop words and with frequency higher than the Term Frequency Threshold;
\[ \Sigma, \ U, \ V \leftarrow \text{SVD} \ (A); \ \{\text{Product of SVD decomposition of A}\} \]
\[ k \leftarrow 0; \ \{\text{Start with zero cluster}\} \]
\[ n \leftarrow \text{rank}(A); \]
Repeat
\[ k \leftarrow k + 1; \]
\[ q \leftarrow (\Sigma_{i=1}^k \Sigma ii)/(\Sigma_{i=1}^n \Sigma ii); \]
Until \( q < \text{Candidate Label Threshold} \);
\[ P \leftarrow \text{phrase matrix for} \ P_f; \]
For all columns of \( U^T_k \) do
\[ \text{Find the largest component} \ m_i \text{ in the column}; \]
\[ \text{Add the corresponding phrase to the Cluster Label candidates set}; \]
\[ \text{LabelScore} \leftarrow m_i; \]
End for
Calculate cosine similarities between all pairs of candidate Labels;
Identify groups of labels that exceed the Label Similarity Threshold;
For all groups of similar labels do
\[ \text{Select one label with the highest score}; \]
End for

**STEP 4: Cluster Content Discovery**

For all \( L \) (Cluster Label Candidates do
Create cluster \( C \) described with \( L \);
Add to \( C \) all documents whose similarity to \( C \) exceeds the Snippet Assignment Threshold;
End For
Put all unassigned documents in the “Others” group;

**STEP 5: Final Cluster Formation**

For all clusters do
\[ \text{clusterScore} \leftarrow \text{labelScore} \times ||C||; \]
End for

**B. Pre-Processing**

This is the first phase of Lingo algorithm. Preprocessing is done in 3 phases:

- Stemming
- Tokenization

In first phase we are marking stop word e.g. a, and, the, an. Lingo is beneficial compared to other clustering as we are marking the stop words and not removing it, which helps us to understand the meaning of the phrase. In second phase we try to remove stemming words, for e.g. stem of word walking will be walk. We are removing basically the ‘ing’ of the word. In third phase we tokenize the sentences into words.

**C. Frequent Phrase Extraction**

Frequent Phrases are commonly occurring words in the document, which are useful for keeping readers attention to the topic. Frequent phrases can be extracted by using suffix array and SVD method. To be candidate label for cluster the frequent phrase should be:

- It should appear no. of times in the input documents.
- It should not begin or end with stop word.
- Phrase should be meaningful.
- Should not cross sentence boundary.

**D. Cluster Label Induction**

The cluster label induction is the main part of Lingo. The cluster label candidates that were established in the second stage are now being used. A term-document matrix \( A \) with the single terms is set up. For each term the tf-idf weights are computed and used in the matrix. Terms, which were derived from document titles, are scaled by a constant factor to justify their higher significance. A Singular Value Decomposition on the term-document matrix is performed to reduce the dimensionality. The \( U \) matrix represents the abstract concepts of the input documents. With a chosen candidate level threshold \( q \), the \( k \) is estimated: \( ||A_k||_F/||A||_F \geq q \). With this \( k \), \( U_k \) forms a matrix, which expresses the frequent single terms and the abstract concepts of the documents. Further, a new matrix \( P \) of size \( t \times (p + t) \) is constructed. \( P \)’s rows represent again all single terms (\( t \)), the columns illustrate pseudo-documents in terms of the cluster label candidates (frequent phrases (\( p \)) and single terms). The cosine similarity of the abstract concept vectors in \( U_k \) and the pseudo-documents of \( P \) can be calculated: \( M = U_k^T P \). Now the label candidates with the maximum score for each abstract concept are chosen.

**E. Cluster Content Discovery**

The cluster content discovery is carried out with the Vector Space Model. The input documents are
compared with the cluster labels in the same way as before: The columns of a new matrix \( Q \) each represent one of the received cluster label, so that \( C = Q^T A \) (\( A \) is still the original term-document matrix) is a matrix where the elements \( c_{ij} \) represent the relationship of document \( j \) with cluster \( i \). If \( c_{ij} \) of document \( j \) passes the next control parameter, the snippet assignment threshold, it is added to the corresponding cluster \( i \). An artificial cluster labeled “Others” is created and contains all documents which previously could not be allocated to an existing cluster.

F. Final Cluster Formation

The final cluster formation is to refine the final display of the results. A simple method is used to sort the clusters preferring well-described, larger clusters over smaller and noisy arranged according to the scoring function \( C_{score} = \text{label score} \times \text{number of documents in cluster} \ C \).

IV. PROPOSED WORK

The proposed work of the system gives us an idea about how system works. The proposed system is Email Clustering System which includes single Sign-In and extra feature added is voice enabled commands. In proposed system third party server forms clusters according to the content of the emails that are available in Inbox. It is desktop based application in which it will first fetch the emails from inbox and then forms cluster, to form the clusters we are using Lingo algorithm.

![Fig. 1 Detail Architecture Diagram](image)

In this architecture we have user interface which will have option to log in to the system. The user will have to give his details regarding his current ID and passwords present on the third party server. The Application then store this all information and allocate a user ID and password to the user. This user ID and password will then always be used by the user to access all the mails through single sign-on facility. These Emails are given as input to our clustering system which then performs all the clustering steps, and as a output we get all the clustered mails present on our system. Hence we have used lingo algorithm to cluster the emails and single sign-on facility is provided. Extra feature that is provided to our system is voice enabled functions which can help handicapped people.

V. RESULTS

There are two parameters based on which our systems performance is based Precision and Recall. Let us consider \( D \) be the set of documents, set \( A \) of documents was retrieved for users query is the set of documents that are relevant to documents present in \( D \). \( R_A \) be the intersection of \( R \) and \( A \).

Definition — Precision is the fraction of the retrieved documents which is relevant:

\[
\text{Precision} = \frac{|R_A|}{|A|}
\]

Definition — Recall is the fraction of the relevant documents which has been retrieved:

\[
\text{Recall} = \frac{|R_A|}{|R|}
\]

Consider a dataset for email system which contains 100 emails regarding different topics. Data is fetched from the email server to the database and the lingo algorithm works on these emails content. Hence we present result of our system. We calculated the Precision and Recall measures for each and every cluster shown by our system.

| TABLE I |
| RESULTS FOR LINGO CLUSTERING ALGORITHM |
| Clusters | Results |
| flower | Precision | Recall |
| 89% | 75% |
| water | 88% | 67% |
| cricket | 90% | 82% |

Hence, these results show the Lingo algorithm gives us the best accuracy for clustering set of documents.

VI. CONCLUSION

This paper concludes that an email clustering approach is proposed to show text similarities. The proposed technique shows the email attributes and how the text similarities are used to cluster the users. The mails will be finally in a clustered format so as to
minimize the job of searching for the users. Hence this will also reduce the consumption and make browsing user friendly. The future scope of the work could be incorporating the similarity of the email attachments etc. for the more accurate clustering of the emails. The other direction of the proposed work could be applying the proposed email similarity function for the more email mining operations like thread summarization, automatic answering, and applying the same Techniques for other Email datasets for participating all the attributes of the emails and achieving more accurate results. Large number of mails can be clustered and handled simultaneously by the server depending on the network load. Also the Account holder doesn’t get disturbed while the clustering is in progress. So it saves a large amount of time of administrator and also the space on the server.

VII. REFERENCES

[1] Yiming Yang, Shinjae Yoo, Frank Lin, and Carnegie Mellon University Il-Chul Moon, Korea Advanced Institute of Science and Technology, “Personalized Email Prioritization Based on Content and Social Network Analysis”, IEEE INTELLIGENT SYSTEMS Published by the IEEE Computer Society.


[8] Stanislaw Osinski and Dawid Weiss, Poznan University,”A concept Driven Algorithm For Clustering Search Result”,2005 IEEE.
