# An Automatic Email Management Approach Using Data Mining Techniques

Gunjan Soni and C.I. Ezeife*

School of Computer Science, University of Windsor, Windsor, ON, Canada
{sonig,cezeife}@uwindsor.ca

**Abstract.** Email mining provides solution to email overload problem by automatically placing emails into some meaningful and similar groups based on email subject and contents. Existing email mining systems such as BuzzTrack, do not consider the semantic similarity between email contents, and when large number of email messages are clustered to a single folder it retains the problem of email overload. The goal of this paper is to solve the problem of email overload through semantically structuring the user's email by automatically organizing email in folders and sub-folders using data mining clustering technique and extracting important terms from created folders using Apriori-based method for folder identification. This paper proposes a system named AEMS for automatic folder and sub-folder creation and later indexing the created folders. For AEMS module, a novel approach named Semantic non-parametric K-Means++ clustering is proposed for folder creation. Experiments show the effectiveness and efficiency of the proposed techniques using large volumes of email datasets.

**Keywords:** Email Mining, Email Overload, Email Management, Data Mining, Clustering, Feature Selection.

## 1 Introduction

Email is a widely used way of written communication over the internet. According to an estimate [1], the number of email messages sent daily has reached around *3.4 billion* in 2012, resulting in the evolution of the problem of email overload. Email overload [2] is a state of being completely overwhelmed of email inboxes by the amount of email one has received.

Email overload can be handled by managing email messages by summarization and automatically categorizing emails into folders. Automatic folder creation is, given a set of email messages and we need to semantically assign each message to similar groups according to the email content. Some automatic email folder creating techniques are given in [2], [3], [4] and [5]. Another solution to email overload is given by the email summarization ([6], [7]). The goal of email summarization is to provide

* This research was supported by the Natural Science and Engineering Research Council (NSERC) of Canada under an Operating grant (OGP-0194134) and a University of Windsor grant.

L. Bellatreche and M.K. Mohania (Eds.): DaWaK 2013, LNCS 8057, pp. 260–267, 2013.

concise, informative summary of email which in turn is helps to decide if the message demands immediate attention.

Both folder categorization and email summarization do not reduce email overload when email sender, subject and topic are not known. Therefore, our proposed approach overcomes this problem by folder creation based on email subject and content, sub-folder based on sender of the email and then index or view will be created in a separate web page, which contains link to the respective folder and sub-folder, and contains the summary of each folder.

### 1.1 Contributions

This paper proposes an algorithm AEMS (Automatic Email Management System) which manages emails by organizing similar emails in the folders (module 1 named AEG), then again organizes emails of each folder into subfolders (module 2 named APEG) where subfolder will contain emails sent by only a particular person and lastly creating the index, which contains name and link to the folders and sub-folders and also contains a summary annotation about the content of the respective folders.

For model AEG (Automatic Email Grouping), we have introduced document frequency based feature selection method named Associative term frequency. We also proposed novel Semantic Non-parametric K-Means++ Clustering method for folder creation, which avoids, (1) random seed selection by selecting the seed according to email weights, and (2) pre-defined number of clusters using the similarity between the email contents. Lastly, we have applied an Apriori-based folder summarization which extracts frequent patterns from the emails of respective folders useful for identification of content of folders.

## 2 Related Work

Similar to our AEG model, the work is shown in BuzzTrack [4], which used vector space model for email representation and cluster emails based on three measures: text similarity, subject similarity and people-based similarity. Next, kernel-selected email clustering [5] was introduced for email clustering. They consider the global weighting of each email subject and body for the creation of email VSM (vector space model) and then create email matrix and used an improved K-means clustering algorithm based on the lowest similarity. However, the work in [2], [3], [4] and [5] techniques are limited because if a created folder contains 2000 emails, it is hard to find an email of a specific individual whose name is not specifically known by the user.

On the other hand email summarization techniques such as: NLP and Machine Learning techniques based email summarization [6] extract the important candidate noun phrases (NPs) from the email messages and manually classify the selected NPs into those that should be or not included in the summary. These NPs are used to build training set which is then used to summarize incoming messages. Next, CWS [7] is email conversation summarizer which uses clue words to measure the importance of sentences in conversation summarization based on the clue words and sentence score

of a summary is produced. The work in [6] and [7] do not provide help to find a particular email when millions of emails are present.

## 3 The Proposed AEMS Model

In this section, AEMS module is presented which automatically and semantically arranges email in similar groups by summarizing the content of each group and creating a view called index. The AEMS module is divided into three sub-modules which include: Automatic Email Grouping (AEG), Automatic People based Email grouping (APEG) and Indexing.

### 3.1 AEG Model – Automatic Folder Creation and Topic Detection

The input as a set of email messages directly goes to AEG where, AEG is a process of creating main folder based on similar email messages and semantic similarity measures and includes the 4 stages of (1) pre-processing, (2) feature selection, (3) clustering algorithm and (4) topic detection with details presented next.

**Step 1 and 2: Pre-processing and Feature Selection**
For pre-processing, subject line and content of the email messages are extracted from each email and stop words are removed which can reduce the size of the email to be processed. Next, we review the features by taking each term from the processed data to calculate the associative term frequency ($R_{tf}$) of a particular term x, which is the number of emails that contain the term, x. Features will be selected according to the $R_{tf}$, where $R_{tf}$ should be greater or equal to the user specified threshold, $T_s$ and $T_b$ depending on whether the term appears in subject or content of the email respectively.

$$R_{tf}(x) = (df_x * 100)/N \quad (1)$$

In equation 1, N is the total number of email messages in the dataset; $df_x$ is the total number of emails in which the term *x* appeared. Once the feature terms are selected from the email, the email vector is created by combining the feature terms and removing the duplicate terms from the vector.

**Step 3: Semantic Non-parametric K-Means++ Clustering Algorithm**
Thirdly, step of the AEG process is to apply semantic_nonparametric_Kmeans++ algorithm of Fig. 2, where the emails are clustered together according to proposed the Semantic Non-parametric K-Means++ clustering algorithm. First, select the initial cluster center by calculating the email weight as shown in semantic_nonparametric_kmeans++ algorithm of Fig. 1, step 1. The initial cluster center is the email with the maximum weight, where email weight is the total number of feature terms in the email. After this, chose all other clusters center by calculating the similarity between all emails with the initial cluster center as shown in semantic_nonparametric_Kmeans++ algorithm of Fig. 1, step 2. Chose other clusters center

using a weighted probability distribution where an email $x$ is chosen with probability proportional to $\mathrm{D}(X_{i,j})^2$ and $\mathrm{D}(X_{i,j})$ should be less or equal to $\beta$ (the optimized value of $\beta$ can overcome the problem of pre-defined cardinality of other clustering algorithms because once the $\beta$ is set it will work for all type of data and user need not to give any input such as K in K-Means++), as shown in semantic_nonparametric_Kmeans++ algorithm of Fig. 1, step 3. The similarity $\mathrm{D}(X_{i,j})$(calculated using the semantic text similarity (STS) algorithm [8], which semantically finds similarity between two emails). Once all centers are created, then form the clusters by assigning email $(X_i)$ to the cluster center $C_k$ where, similarity $(\mathrm{D}(X_{k,i}))$ is minimum in comparison to other center, as in semantic_nonparametric_Kmeans++ algorithm (Fig. 1, step 4).

**Algorithm:** semantic_nonparametric_Kmeans++(X) – {Clustering algorithm}
**Input:** Email vector set (*X)*
**Other Variables:**
$\mathrm{D}(\mathrm{C}_{\mathrm{init}},x_i)$: Decimal value indicating similarity between initial cluster center and the email $x_i$
$C_k$: Email vector of text represented as cluster centers
$\beta$: Minimum threshold value, where $0.0 \leq \beta \leq 1.0$
|T|: Total number of features terms
$E_w$: Integer value as email weight (Total number of features)
$\mathrm{X_n}$: Particular email in email vector of text
$C_{init}$: Email vector of text represented as initial cluster center
**Output:** Set of clustered email represented as grouped text
**Begin**
1. FOR each email $(X_i)$ in email vector set (X) DO
   1.1. Email weight of email $X_i$, $\mathrm{E_{wi}} = |\mathrm{T}|$ // Calculate each email weight.
   1.2 Initial cluster center, $X_j = \max(\mathrm{E_w})$ // Email having maximum weight is assigned as initial cluster center $(x_j)$.
2. FOR each email $(X_i)$ in email vector set (X) DO
   2.1 Calculate similarity $\mathrm{D}(X_{i,j})$// Calculate the similarity between initial cluster center and the each email using STS [8].
3. Choose all cluster centers $X_k$, select $X_k$with probability
$$\left(\frac{\mathrm{D}(X_{i,j})^2}{\sum_{x\in X}\mathrm{D}(X_{i,j})^2}\right) \text{ and } \mathrm{D}(X_{i,j}) \leq \beta$$
4. FOR each email $(X_i)$ in email vector set (X) DO
   4.1 Assign email $(X_i)$ to the cluster $X_k$ where, similarity $(\mathrm{D}(X_{k,i}))$ is minimum.// Cluster formation
**End**

**Fig. 1.** Algorithm for Semantic Non-parametric K-Means++ Clustering

**Step 4: Topic Detection and Folder Creation**
Next, the folders are created by topic detection. The subject term with the highest $R_{tf}$ in the whole cluster, $C_k$; is considered as a folder name.

### 3.2 APEG Model – An Automatic Sub-folder Creation

APEG is a process for creating the sub-folders. Sub-folder creation is based on email sender ID and contains the emails from that specific person in the respective folder. The whole process of APEG model is divided into two steps:

*Step 1:* Once the folders are created from the AEG model they serve as input for APEG model. So, firstly email ID of the sender is extracted from email message.

*Step 2:* Then some comparisons are made as follows:

*a.* If a sub-folder exists with the sender information, then that respective email message is moved to that sub-folder.

*b.* Else a new sub-folder is created with the name of the sender and email is then moved into that folder.

### 3.3 Indexing

Lastly, create index, which is a view of the hierarchical folder with links to emails and contains summary annotation of each folder. The output of APEG serves as input for indexing. Here, Apriori algorithm [10] is applied to the folder data to extract important terms which helps identify the content of folders. The whole process of indexing is divided in two steps (repeat following steps for all folders created):

*Step 1:* Feature terms of subject and content of each email from the folder are extracted using associated term frequency explained in section 3.1.

*Step 2:* Apply Apriori algorithm to the feature terms to extract the terms that are important for summary.

*Step 3:* Index/View is created as HTML web page which is a hierarchical representation of folders from AEG model, sub-folders from APEG model and containing link to each individual email. Additionally, summary of folder is contained.

## 4 Application Example for AEMS Module on Sample Data

Example 1: Given a user, *u* email inbox with say 5 emails from 2 senders, sender-1 and sender-2. Create topic folders F, sub-folders of sender (SF) and index *i* containing links to those F and SF (small size of file chosen only for illustration purposes). Consider thresholds $T_s = 30\%$, $T_b = 50\%$ and $\beta = 0.2$.

Solution 1: Five emails are input to the AEG model of AEMS system.

*Step1*: The subject and content of the email are extracted and all special characters, punctuation and stop words are removed, according to the section 3.1(Pre-processing) and these simple emails with 2-term subjects and up to 3-term contents are as follows:

Email X1 (sender1) – Subject: {T1, T2} and Content: {T1, T3, T5}
Email X2 (sender2) – Subject: {T1, T3} and Content: {T1, T4, T3}
Email X3 (sender1) – Subject: {T5, T2} and Content: {T5, T4}
Email X4 (sender2) – Subject: {T6, T1} and Content: {T6, T7, T2}
Email X5 (sender1) – Subject: {T4, T3} and Content: {T7, T3, T4}

*Step 2*: we need to find $R_{tf}$ (document frequency of term) for each term and select only those terms with $R_{tf}$ greater or equal to threshold of $T_s = 30\%$, $T_b = 50\%$, according to the section 3.1. The $R_{tf}$ for subject and content are given below respectively.

$R_{tf}$(s) = {(T1, 3) (T2, 2) (T3, 2) (T4, 1) (T5, 1) (T6, 1) (T7, 0)}.
$R_{tf}$(b) = {(T1, 2) (T2, 1) (T3, 4) (T4, 3) (T5, 2) (T6, 1) (T7, 1)}.

Term {T1, T2, T3, and T4} is taken as feature term because there $R_{tf}$ are greater or equal to the pre-defined threshold. Therefore the email vector by the selected feature terms will be: Email X1 with {T1, T2, T3}; Email X2 with {T1, T3, T4}; Email X3 with {T2, T4}; Email X4 with {T1, T2} and Email X5 with {T3, T4}

*Step 3*: Now we will cluster the email with Semantic Non-parametric K-Means++ clustering algorithm according to section 3.1. For this we need to follow steps below:

1. Find email containing the maximum weight ($\max(X_w)$). Here, $\max(X_w) = 3$ which is the email weight for X1, X2 and X3, obtained by calculating the total number of feature terms in email vector. Thus, first initial cluster center will be X1.
2. Calculate the similarity $D(X_{1,j})$ between pairs of emails $X_1$ and $X_j$ to choose the next cluster centers (we chose STS [8] to find the similarity between emails). To compute the similarity between two emails $X_1$ and $X_j$, we need to find the common terms in the two emails and place in the set C and delete theses common terms from both emails as in $X_1 = X_1$ – set C and $X_j = X_j$ – set C. Then, calculate the string similarity between pairs of terms in $X_1$ and $X_j$ using the average of 3 common similarity measures. Next, compute semantic similarity of pairs of terms of $X_1$ and $X_j$ using SOC-PMI [12] (uses point-wise mutual information to sort lists of important neighbor words of the two target words. Then, consider the words which are common in both lists and aggregate their PMI values (from the opposite list) to calculate the relative semantic similarity) before computing the joint similarity matrix (M) of the two matrices for string and semantic similarities of the emails. Now, extract the maximum value from M and delete corresponding row and column (repeat till M become empty) and store the summation of extracted values in variable (say, S). Lastly, similarity score $D(X_{1,j})$ is calculated using equation 8

$$D(X_{1,j}) = ((S + \delta) * (m + n))/2mn \quad (8)$$

Here, m and n are the total number of terms in email $X_1$ and $X_j$ respectively and $\delta$ is the total number of terms in set C.

The next center is chosen if its $D(X_{1,j}) \leq \beta$ and similarity proportional to $D(X_{1,j})^2$. Here, $\beta$ is the pre-defined threshold value.

3. Again calculate the similarity of each email with each cluster center and assign that to the cluster where, the similarity is maximum. Suppose, the clusters formed are: C1 – {X1, X2, X4} and C2 – {X3, X5}.
4. Using these clusters, two folders are created and choose subject term as folder name, which have maximum $R_{tf}$. Therefore two folders created are: Folder T3 containing email X1, X2 and X4, and Folder T4 containing email X3 and X5.

*Step 4*: These two folders will be the input of the APEG model and APEG will create sub-folders according to the senders' email ID. Here, folder T3 will contain two sub-folders from sender-1 and sender-2, where sub-folder from sender-1 will contain email X1 and sub-folder from sender-2 will contain email X2 and X4. Similarity, for folder T4 will contain one sub-folder from sender-1 containing email X3 and X5.

*Step 5*: Finally, one index file will be created with links to the folders and sub-folders according to section 3.3 and also, contain email summary is created by applying Apriori algorithm.

# 5 Experimental Results

The AEMS module is implemented in Java and Eclipse is used as a development IDE. The hardware configuration to run the experiments is 3GB RAM, intel core i3 CPU, 2.34 GHz and 32-bit windows-7 operating system. To test our approach, we used publically available 20-Newsgroup collections [10]. It is a collection of 20,000 email messages, divided into 20 different newsgroups. The 20-Newsgroup data comes in, one file per email message containing email logs.

## 5.1 Evaluation Criteria

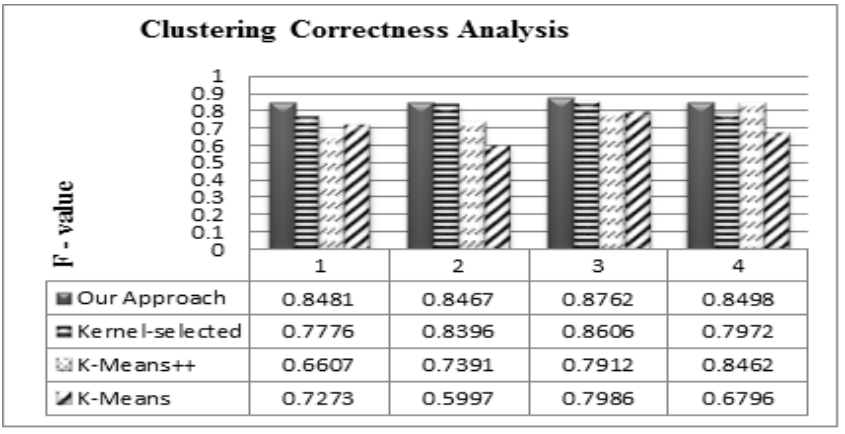
We used the F-value measure to evaluate the clustering quality and its formula is defined in equation 8:

$$F - Value, F(i,j) = 2 * P(i,j) * R(i,j)/P(i,j) + R(i,j) \quad (9)$$

where, $Precision, P(i,j) = n_{ij}/n_j$ , and $Recall, R(i,j) = n_{ij}/n_i$
$n_i$ is the number of clusters which human has labeled, $n_j$ is the number of emails with clustering algorithms, and $n_{ij}$ is the number of emails clustered correctly.

## 5.2 Study on Cluster Performance

We compared our clustering approach with standard K-means [9], K-Means++ [11] and Kernel-selected clustering [5], to show its efficiency of cluster correctness.




| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Our Approach | 0.8481 | 0.8467 | 0.8762 | 0.8498 |
| Kernel-selected | 0.7776 | 0.8396 | 0.8606 | 0.7972 |
| K-Means++ | 0.6607 | 0.7391 | 0.7912 | 0.8462 |
| K-Means | 0.7273 | 0.5997 | 0.7986 | 0.6796 |

**Fig. 2.** F-value comparison of clustering Algorithm

We choose four folders from the 20-NewsGroup data set. These four folders consist of 1000 email messages each and results are evaluated terms of F-Value. Now, when experimenting data with Kernel-selected email clustering method and our proposed clustering algorithm, we have taken $\beta$ = 0.5. We can observe from Fig. 2 that our approach performs better than the standard K-Means, K-Means++ and Kernel-selected clustering approach. Since the average of the F-Value when threshold is

taken as zero comes out to be 0.8552 for our clustering approaches whereas for Kernel-selected clustering method comes to be 0.8187.

## 6 Conclusions and Future Work

This paper proposed an Automatic Email Management System (AEMS) which clusters emails into meaningful groups and extract important feature words for identification of each folder. For AEMS, we proposed a novel feature selection based clustering approach. Future work could be that AEMS module, do not handle the processing of incoming messages; therefore, a method can be developed to immediately process incoming messages using classification methods. Additionally, some recommendation system can be built based on the emails logs for deletion of unused email.

## References

1. Radicati, S., Hoang, Q.: Email statistics report, 2012-2016. The Radicati Group, Inc., London (2012)
2. Xiang, Y.: Managing Email Overload with an Automatic Nonparametric Clustering Approach. The Journal of Supercomputing 48(3), 227–242 (2009)
3. Schuff, D., Turetken, O., D'Arcy, J.: A multi-attribute, multi-weight clustering approach to managing "e-mail overload". Decision Support Systems, 1350–1365 (2006)
4. Cselle, G., Albrecht, K., Wattenhofer, R.: BuzzTrack: topic detection and tracking in email. In: Proceedings of the 12th International Conference on Intelligent User Interfaces (IUI 2007), New York, NY, USA (2007)
5. Yang, H., Luo, J., Yin, M., Liu, Y.: Automatically Detecting Personal Topics by Clustering Emails. In: 2010 Second International Workshop on IEEE Education Technology and Computer Science (ETCS), pp. 91–94 (2010)
6. Muresan, S., Tzoukermann, E., Klavans, J.L.: Combining linguistic and machine learning techniques for email summarization. In: Proceedings of the 2001 Workshop on Computational Natural Language Learning (2001)
7. Carenini, G., Ng, R.T., Zhou, X.: Summarizing email conversations with clue words. In: Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada (2007)
8. Islam, A., Inkpen, D.: Semantic text similarity using corpus-based word similarity and string similarity. ACM Transactions on Knowledge Discovery from Data (TKDD) 2, 10 (2008)
9. Han, J., Kamber, M.: Data mining: concepts and techniques. Morgan Kaufmann (2006)
10. Lang, K.: Newsweeder: Learning to filter netnews. In: Proceedings of the Twelfth International Conference on Machine Learning (1995)
11. Arthur, D., Vassilvitskii, S.: k-means++: the advantages of careful seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1027–1035. ACM (2007)
12. Islam, A., Inkpen, D.: Second order co-occurrence PMI for determining the semantic similarity of words. In: Proceedings of the International Conference on Language Resources and Evaluation, Genoa, Italy, pp. 1033–1038 (2006)