An Automatic Email Mining Approach Using Semantic Non-parametric K-Means ++ Clustering

By: Gunjan Soni

Principal Advisor: Dr. Christie Ezeife
Internal Reader: Dr. Jianguo Lu
External Reader: Dr. Sévérien Nkurunziza
Chair: Dr. Dan Wu

School of Computer Science
University of Windsor

Table of Content

1. Introduction

- Email is a popular mode of internet communication and contains large percentage of important and daily information.
- According to an estimate given by (Radicati, 2011), the number of email messages sent daily has reached around *3.1 billion* in *2011*.
- Email inboxes are now filled with huge variety of voluminous messages and thus increasing the problem of "Email Overload" (Xiang, 2009) which places financial burden on companies and individuals.
- Email mining is a method for providing solution to email overload by automatically grouping emails into some meaningful and similar groups based on the email subject and content.

1. Introduction

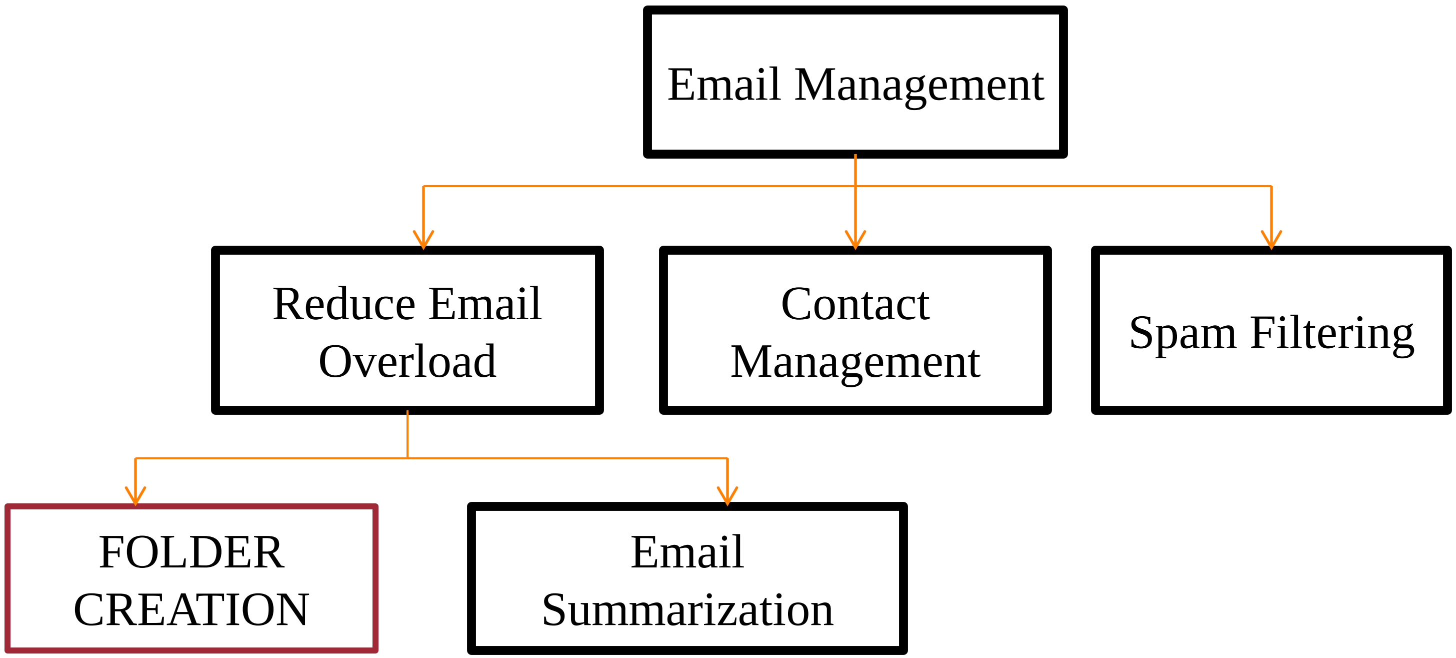

Figure 1: Categories of Email Management Tasks

1. Introduction

- Automatic folder creation can be topic oriented such as ‘appointments’, ‘personal’ and ‘entertainment’ or group oriented such as ‘courses’ and ‘project’ or people specific such as ‘John’ and ‘Mary’.
- It can be done by using data mining techniques such as CLUSTERING.
- Clustering of email is a method by which large sets of email is grouped into clusters of smaller sets of similar data.
- Clustering algorithm attempts to find natural groups of emails based on text similarity of email subject and content.
- The most popular methods for email mining are K-Means clustering and Hierarchical Agglomerative Clustering.

1. Introduction

- K-Means++ Clustering (Arthur & Vassilvitskii, 2007):
  - It is a method of clustering which aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean.
  - For example,
    - Distance between 4 emails is and K=2:

Email#	**E1**	**E2**	**E3**	**E4**
E1	0	0.56	0.11	0.2
E2	0.56	0	0.13	0.082
E3	0.11	0.13	0	0.5
E4	0.2	0.082	0.5	0

Table 1: Distance between all emails

1. Introduction

- Step 1: Select initial cluster center randomly, suppose E2.
- Step 2: Now, select other cluster center where the distance is maximum

  i.e. Email E1

- Step 3: Distance is calculated from centers to other emails and emails are assigned to cluster where distance is minimum.

Email#	**E3**	**E4**
E1	0.11	0.2
E2	0.13	0.082

Table 2: Distance between emails and cluster centers

- Therefore E3 will be assigned to E1 and E4 will be assigned to E2.

1. Introduction

- Step 4: Now, mean of cluster is calculated and mean act as new cluster center
- Repeat step 3 and 4 till it converges.

◆ Limitation:

- Fixed number of clusters can make it difficult to predict what K should be
- It is sensitive to initialization.

1. Introduction

- STS - Semantic Text Similarity (Islam & Inkpen, 2008)
  - Detecting semantic similarities and differences between two sentences.
  - Problem definition:

    Given two input text segments → automatically determine a score that indicates their similarity at semantic level.

Phase 1
- String Similarity between Words

Phase 2
- Semantic Similarity between Words

Phase 3
- Overall Sentence Similarity

1. Introduction

P = "A cemetery is a place where dead people's bodies or their ashes are buried."

S = "A graveyard is an area of land, sometimes near a church, where dead people are buried."

- Step 1:

  P ={cemetery, place, where, dead, body, ash, bury}

  R ={graveyard, area, land, sometime, near, church, where, dead, bury}

  m = 7, n = 9

- Step 2: Three tokens {where, dead, burry} in P match exactly with R, therefore $\delta = 3$.

  P ={cemetery, place, body, ash}

  R ={graveyard, area, land, sometime, near, church}

1. Introduction

- Step 3: Construct 4x6 *string matching matrix* M_1.

  Consider pair (place, land) => $\eta = 5$, $\tau = 4$

  Length(LCS(place, land) = 2,

  => NLCS(place, land) = $v_1 = 2^2/(4x5) = 0.2$

  Length(MCLCS(place, land)) = 0

  => NMCLCS(place, land) = $v_2 = 0$

  Length(MCLCS(place, land)) = 2

  => NMCLCS(place, land) = $v_3 = 2^2/(4x5) = 0.2$

  $\alpha_{23} = 0.33 * (v_1 + v_2 + v_3) = 0.132$

$$M_1 = \begin{array}{c} \\ cemetery \\ place \\ body \\ ash \end{array} \begin{array}{cccccc} graveyard & area & land & sometime & near & church \\ \end{array}$$

$$M_1 = \begin{pmatrix} 0.023 & 0.021 & 0 & 0.129 & 0.052 & 0.041 \\ 0.037 & 0.083 & 0.132 & 0.017 & 0.033 & 0.022 \\ 0.018 & 0 & 0.041 & 0.021 & 0 & 0 \\ 0.024 & 0.083 & 0.055 & 0.028 & 0.055 & 0.037 \end{pmatrix}$$

M_1	graveyard	area	land	sometime	near	church
cemetery	0.023	0.021	0	0.129	0.052	0.041
place	0.037	0.083	0.132	0.017	0.033	0.022
body	0.018	0	0.041	0.021	0	0
ash	0.024	0.083	0.055	0.028	0.055	0.037

1. Introduction

- Step 4:
  - Construct 4x6 *semantic similarity matrix* M_2 using SOC-PMI method (Islam & Inkpen, 2006)
    - SOC-PMI (Second Order Co-occurrence PMI) word similarity method uses the PMI to sort lists of important neighbor words from a large dataset
    - PMI (Point wise Mutual Information) relate to the probability of two words co-occurred.

$$M_2 = \begin{array}{c} \\ cemetery \\ place \\ body \\ ash \end{array} \begin{array}{cccccc} graveyard & area & land & sometime & near & church \\ \end{array}$$

$M_2 =$	graveyard	area	land	sometime	near	church
cemetery	0.986	0	0.390	0.195	0.542	0.856
place	0	0.413	0.276	0.149	0	0
body	0.465	0	0.363	0.122	0.063	0.088
ash	0.796	0	0.213	0.238	0.395	0.211

1. Introduction

- Step 5:

  Construct 4x6 *joint matrix* M, assign equal weight factors ψ = φ = 0.5 (determined heuristically)

$$M = (\psi * M_1) + (\varphi * M_2)$$

area land sometime near church

land sometime church church

$M = body \; (0.039 \quad \mathbf{0.071} \quad 0.044) \; 0.044$

~~0.124~~

ρ ={ 0.50} 0.248} 0.225} 0.071}

1. Introduction

$$C = \sum_{i=1}^{|\rho|} \rho_i = (0.50 + 0.248 + 0.225 + 0.071) = 1.049$$

- Step 6:

$$S(P,R) = \frac{(\delta + C) \times (m+n)}{2mn}$$

$$= ((3 + 1.049) \times 16)/126$$

$$= 0.514$$

δ →	No. common terms in P and S
C →	Summation of relevant terms
m →	No. of terms in P
n →	No. of terms in S

1. Introduction

- BuzzTrack (Cselle, Albrecht, & Wattenhofer, 2007) is a popular tool to reduce email overload by automatic folder creation using clustering.

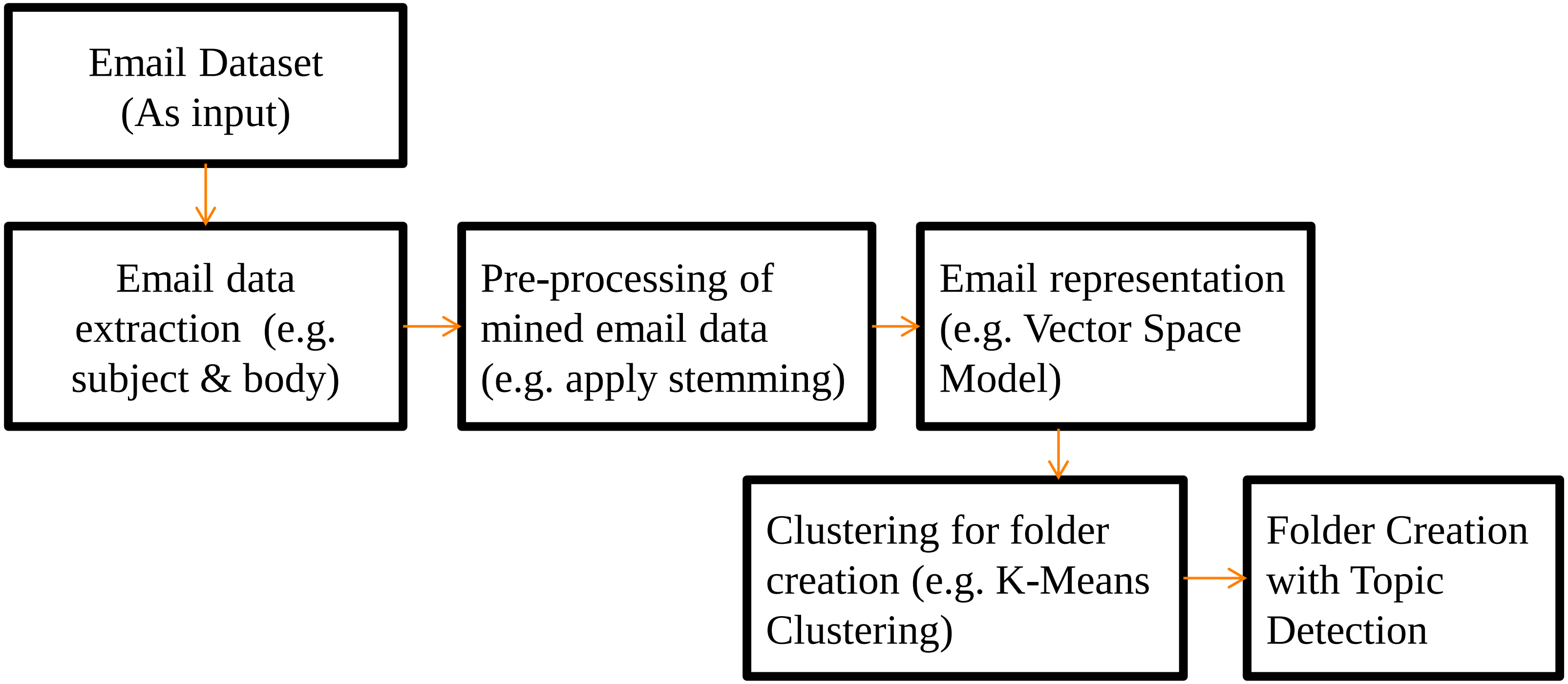

Figure 2: Automatic Folder Creation by Email Clustering

1. Introduction

- Each email is then represented as vector using vector space model.

  For example, email vector, E is given as:

  E = {(1, assign), (2, students), (5, attach)}

  where, in element (1, assign)

  weight assign to term 'assign' — term from email content

  Here, weights are assigned using the following formula:

$$w_{i,j} = \begin{cases} (1 + \log(tf_{i,j}))log\left(\frac{N}{df_i}\right) & if\ tf_{i,j} \geq 1 \\ 0 & if tf_{i,j} = 0 \end{cases}$$

  Where, $tf_{i,j}$ → Frequency of term t_i in email j

  N → Total number of emails in dataset

  df_i → Total number of emails in which the term appeared.

1. Introduction

- Clustering:
  - Firstly, finding text similarity based on Cosine similarity algorithm.
  - Secondly, finding subject similarity calculating the overlap between the set of words S_i, S_j in the subject lines of two emails.

    $$sim_{subject}(m_i, m_j) = 2|S_i \cap S_j|/|S_i| + |S_j|$$

    For example, we have two subject sets of two different emails:

    S_i = {'hello', 'assignment', 'professor', 'exam', 'score', 'grade'}.
    S_j = {'hello', 'student', 'exam', 'car', 'grade', 'school'}.

    So, the subject similarity will be $(2 * 3)/(6 + 7) = 0.461$.
  - If this score is below a clustering threshold for all existing clusters, the email is mapped to a new cluster else it is mapped to closest cluster.

1. Introduction

- Lastly, topic is detected from the cluster. Term with highest weight is selected as a topic
- Limitation:
  - Feature selection is not taken into consideration.
  - Since using the Vector Space Model, therefore there is a
    - loss of correlation and context of each term which are important in grouping the document and
    - it is inefficient for sentence representation because the vector representing the sentence does contain many null value.

2. Related Work

- Automatic Clustering E-Mail Management System - ACEMS (Schuff, Turetken, & D'Arcy, 2006).
  - They introduced the concept of multi-attribute and multi-weight and extends the application of hierarchical clustering to the domain of email.
  - Limitations is that there is no provision for relocation of emails that are incorrectly grouped and no feature selection considered.
- Automatic Nonparametric Text Clustering Algorithm (Xiang, 2009):
  - Proposed an automatic email clustering system, underpinned by a new nonparametric text clustering algorithm which does not require any predefined input parameters (k)
  - Limitation is this method greatly depends on the length of the vector to be compared and no feature selection.

2. Related Work

- Kernel-selected email clustering algorithm (Yang, Luo, Yin, & Liu, 2010):
  - Preprocess the emails and construct the email VSM(vector space model) by combining the body and subject.
  - Then adopt the advanced k-means algorithm to cluster the emails and design a kernel-selected algorithm based on the lowest similarity.
  - Limitation
    - Based on the vector space model,
    - Based on random seed selection, and
    - No feature selection

3. Thesis Problem

- Given an user (*u*) email inbox, we need to create topic folders (F) based on similarity of email content, sub-folders of sender (SF) and index (*i*) containing links to those F and SF, we need to find
  - How to identify the feature terms for clustering which can best represent the content of document.
  - How to form the clusters for folder creation without any pre-defined parameters (such as, *K* in K-Means++ clustering).
  - How to cluster the emails semantically.
  - How to create sub-folders (SF) based on sender of email.
  - How to index and link the folders created.

4. Thesis Contribution

- Proposed AEMS (Automatic Email Management System) model consisting of three sub-modules:
  - AEG (Automatic Email Grouping) model which manages email by organizing similar email in the topic folders (F).
  - APEG (Automatic People Email Grouping) model which organizes emails into subfolder (SF) which contain emails sent by a particular person.
  - Proposed method for index (i) creation, which contain name and link to the folders and sub-folders.
- Introduced document frequency based feature selection method named Associative term frequency for clustering in AEG model.

4. Thesis Contribution

- Proposed Semantic Non-parametric K-Means++ Clustering for AEG model which,
  - Selects the initial seed according to the email weight,
  - Decides the cardinality according to the similarity between the email content, and
  - Semantically cluster formation.

5. Proposed AEMS Model

- AEMS model mines data from the email and cluster email in the specific group and sub-group, of similar email and person respectively and produce index.
- The method for building AEMS model is divided into three modules
  - Automatic Email Grouping (AEG);
  - Automatic People based Email Grouping (APEG) and
  - Indexing.
- The process flow is shown in Figure 3:

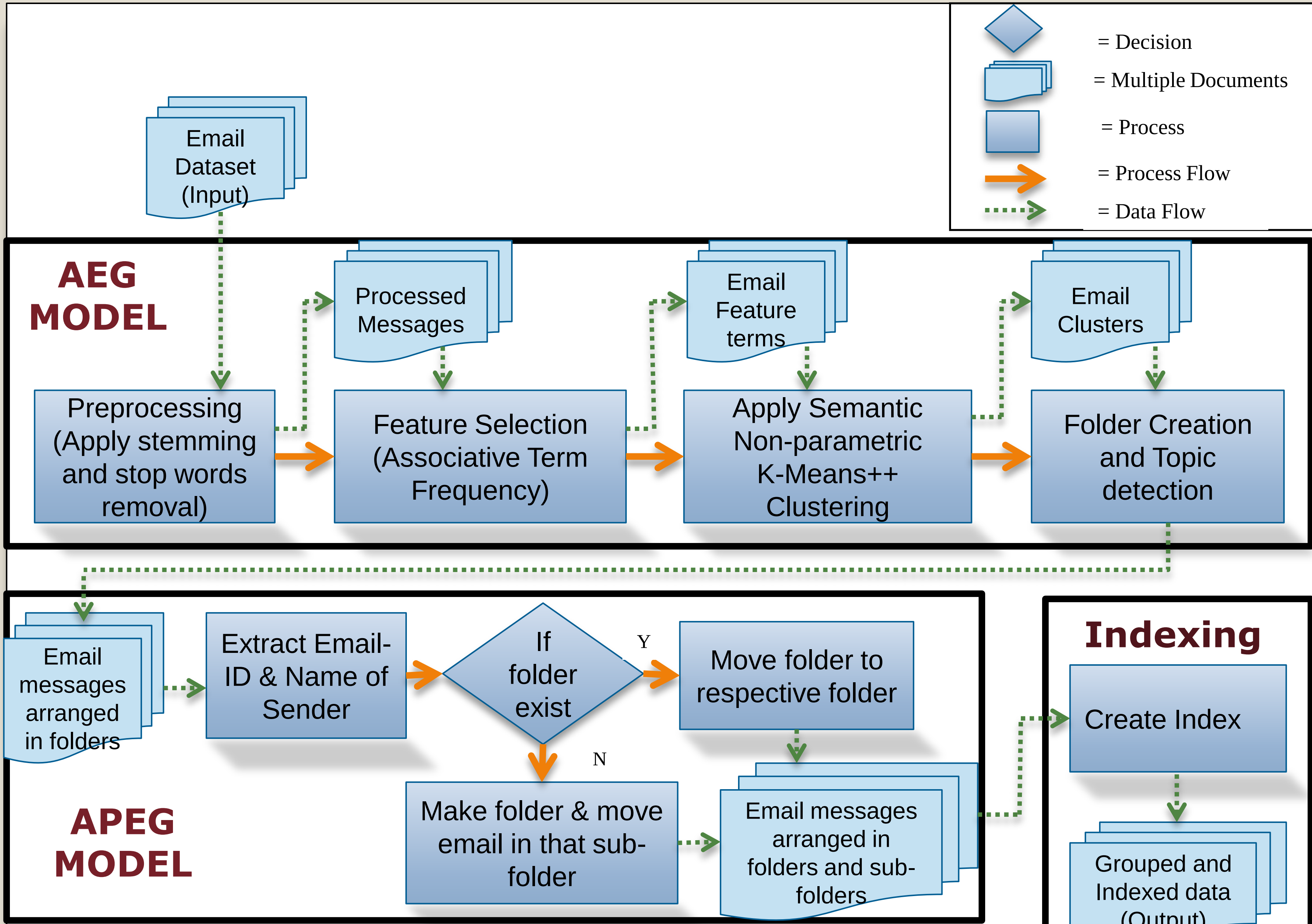

Figure 3: AEMS Model

5. Proposed AEG System

- AEG system is a process of creating topic based folder based on similar email messages.
- Step 1: Input -
  - Raw emails from inbox of user.
- Step 2: Extract email subject and content -
  - For example, in email E1

    Subject: “Assignment”

    Content: “Hi all 60-510 students, Please find assignment #5 attached”
- Step 3: Pre-processing -
  - Remove stop words, such as ‘a’, ‘but’, ‘the’
  - Apply stemming algorithm
    - For example, words assignment, assigning, assigned will be converted to assign.

5. Proposed AEG System

For example,

Subject: ‘assign’

Content: ‘students please find assign attach’

- Step 4: Feature selection -
  - Calculate the associative term frequency ($R_{tf}(x)$) of a particular term x, which is the percentage of emails that contains the term, x.
  - Term x is a feature, if $R_{tf}(x) \geq T_s$ (if term appear in subject) or$R_{tf}(x) \geq T_b$ depending (if term appear in content)

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

Where, df_x → Total number of emails in which the term *x* appeared.

N → Total number of email messages in the dataset

5. Proposed AEG System

- For example,
  - If term ‘assignment’ from subject appears in 5 emails out of 50 emails
    - Then, R_{tf} (assignment) = 10.
  - If $T_s = 5$, then ‘assignment’ will be a feature term, because

    R_{tf} (assignment) $\geq T_s$.

- Step 5: Semantic Non-parametric K-Mean++ clustering
- First seed selection:
  - Email with the maximum weight is selected as the first cluster center where email weight is considered as the total number of feature terms in that email.

5. Proposed AEG System

For example,

Consider a set of 6 email vectors {E1, E2, E3, E4, E5, E6}:

E1 – {Assignment, Student, Please, Attached, Try}
E2 – {Assignment, Student, Please, Attached, University}
E3 – {Assignment, Please, Attached}
E4 – {Appointment, Meet, University, Windsor}
E5 – {Thesis, Defense, Please, Attached}
E6 – {Appointment, Windsor, Meet}

E_{W1}	5
E_{W2}	5
E_{W3}	3
E_{W4}	4
E_{W5}	4
E_{W6}	3

So, here E1 is selected as the first cluster center.

- Cluster centers:
  - Calculate the similarity, $D(x_{i,j})$ (using STS coefficient) between all emails with the initial cluster center.

5. Proposed AEG System

- Choose other cluster centers x_j if
  - $D(x_{i,j}) \leq \beta$, where i is existing cluster center and j is other email and
  - $\forall i, \sum D(x_{i,j})$ is minimum.

For example, let similarity between email E1 and other emails are

Email	E2	E3	E4	E5	E6
E1	0.86	0.74	0.11	0.15	0.21

Suppose, $\beta = 0.2$, the next cluster center will be E4.

5. Proposed AEG System

- Next, let similarity between E1 and E4 with other emails is:

Email	E2	E3	E5	E6
E1	0.86	0.74	0.15	0.21
E4	0.13	0.23	0.15	0.73

  - Other cluster center will be E5 since its summation of similarity (0.15+0.15 = 0.30) with E1 and E4 is minimum and is less than β
  - Thus, there will be three cluster center, E1, E4 and E5

- Since STS find semantic similarity therefore it adds **SEMANTIC** to the clusters.
- There is no pre-defined parameters such as K (Number of cluster) is taken from the user, so the algorithm is **NON-PARAMETRIC.**

5. Proposed AEG System

- Cluster formation –
  - The cluster will be formed by finding the similarity between emails and the cluster centers.
  - With the maximum similarity, that email will be assigned to the respective cluster.

Email	E2	E3	E6
E1	0.86	0.74	0.21
E4	0.13	0.23	0.73
E5	0.21	0.27	0.32

Table 3: Similarity between all emails

For example, Cluster

C1 – {E1, E2, E3}
C2 – {E4, E6}
C3 – {E5}

5. Proposed AEG System

- Step 6: Folder creation and topic detection -

- Using these clusters, folders are created.
- Subject term having maximum R_{tf} in there respective cluster, will be chosen as folder name.
  For example,

Folder Name	**Content**
Assignment	E1, E2, E3
Appointment	E4, E6
Defense	E5

Table 4: Cluster Formed

5. Proposed APEG System

- APEG system is a process for creating the sub-folders based on email sender ID and contains the emails from that specific person in the respective folder created by AEG system. The algorithm works as follows:

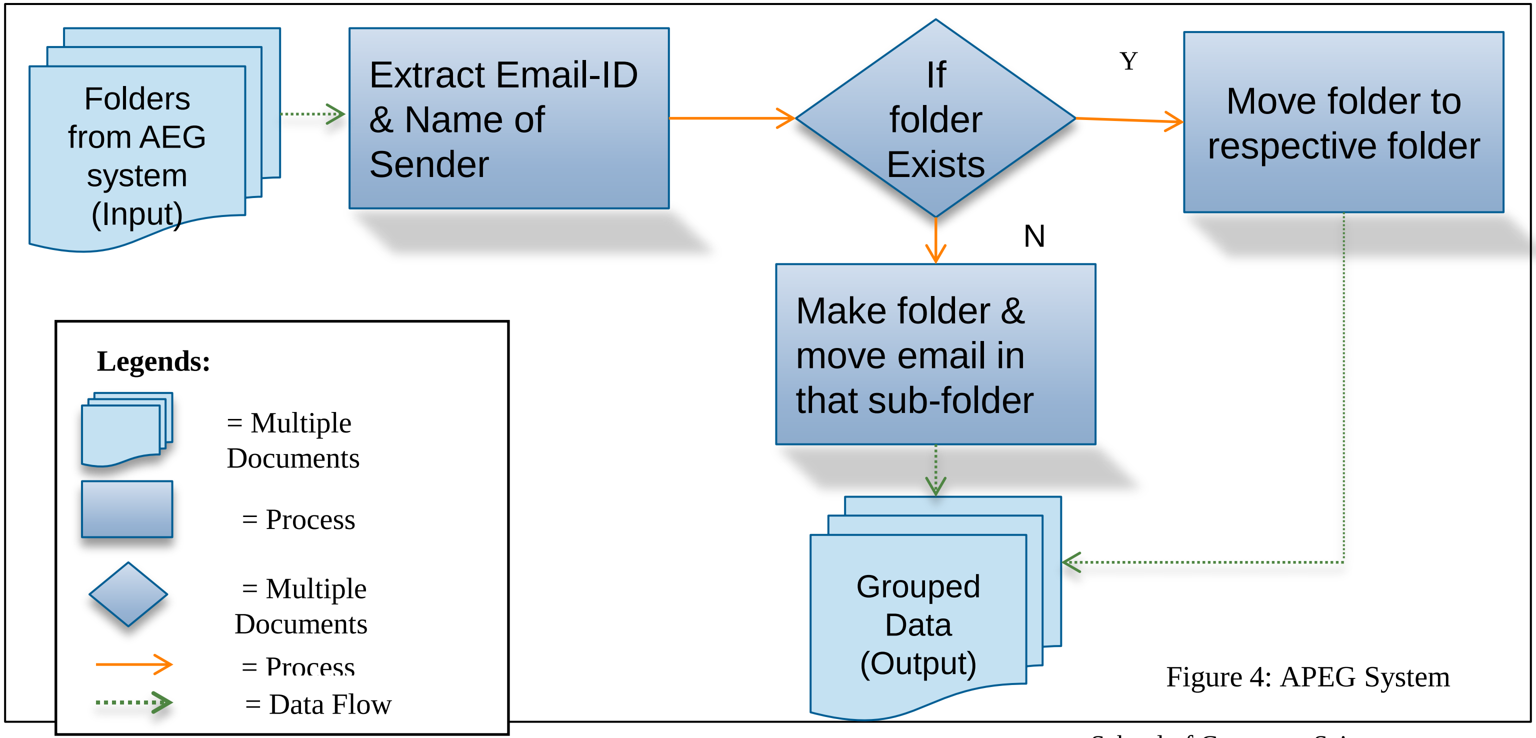

Figure 4: APEG System

5. Proposed APEG System

For example,

- From email E1 in folder “Assignment”, sender Email-ID (david12@gmail.com) and name (David) is extracted.
- Since there is no folder named “David” therefore, a folder is created in folder “Assignment” and email E1 is moved to that folder.
- Email E2 is taken and sender email ID (david12@gmail.com) is extracted and name (David).
- Since sub-folder named “David” already exists, therefore email E2 is moved to that folder. Therefore sub-folder David will contain 2 email messages E1 and E2.

5. Indexing

- Lastly, these folders and sub-folders serves as input to indexing method.
- In indexing a separate html file is creates named “Email Index”, which contain the folders name and links to that respective folder.
- For example, the output will be:

Email Index

Appointment
    John (1)
    Sonig (1)
Assignment
    David (2)
    Richi(1)
Defense
    David (1)

Figure 5: Email Index

6. Work Done so far

- Implementation:
  - For AEG system
    - Downloading the emails in text file
    - Pre-processing
    - Feature Selection
    - STS similarity coefficient
  - Implemented whole APEG system

7. Experiments

- Experimental Setup:
  - The proposed algorithm is implemented using open source technologies, Java.
  - Algorithm is applied on Enron email dataset used for the purpose of research in email management, which contains more than 200K messages belonging to 158 users.
  - In this experiments we used inbox folders of “bass-e” and “germany-c” of the Enron email dataset, which consists of 310 and 326 email messages respectively.
  - The hardware configuration to run the experiments used is 3GB RAM, intel core i3 CPU, 2.34 GHz and 32-bit windows-7 operating system.

7. Experiments

- Implemented K-Means++ clustering to test the working of feature selection.
- Evaluation Criterion:
  - The clustering performance of the proposed technique is calculated by using the Davies-Bouldin (DB) index coefficient.
  - The formula given is:

$$DB = 1/n \sum_{i=1}^{n} max_{i \neq j}\left((\sigma_i + \sigma_j)/d(c_i, c_j)\right)$$

Where, *n* is the number of clusters,
C_x is the centroid of cluster *x*,
σ_x is the average distance of all elements in cluster x to centroid C_x, and
$d(C_i, C_j)$ is the distance between centroids C_i and C_j.

7. Results

- Results:
  - Table below demonstrates the number of features selected when using different threshold and corresponding DB-index using K-Means++ clustering on experimenting dataset.

Threshold		Bass-e		Germany-c	
T_s	T_b	No. of Features	DB-Index	No. of Features	DB-Index
0	0	9952	0.8773	7341	0.9359
1	5	779	0.9132	772	0.9617
5	15	72	0.8993	44	0.9883

Table 5: No. of features selected for different threshold

  - Here it is also observed that when the number of features reduced from 9952 to 72 it does not make much significant changes on DB-index for cluster correctness.

8. TimeLine

Thesis Plan													
Deliverables		December 2012			January 2013				February 2013				March 2013
		1W	2W	3W	1W	2W	3W	4W	1W	2W	3W	4W	1W
Solution Design	2W	■	■										
Implementation	4W			■	■	■	■	■					
Experiments	2W							■	■				
Thesis Report	4W							■	■	■	■	■	
Thesis Defence	1W												■

I am planning to defend my thesis around March 11, 2013.

9. Reference

1. Cselle, G., Albrecht, K., & Wattenhofer, R. (2007). BuzzTrack: topic detection and tracking in email . *In Proceedings of the 12th international conference on Intelligent user interfaces (IUI '07)* (pp. 190-197). New York, USA,: ACM.
2. Islam, A., & Inkpen, D. (2006, May). Second order co-occurrence PMI for determining the semantic similarity of words. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2006)* (pp. 1033-1038).
3. Islam, A., & Inkpen, D. (2008). Semantic text similarity using corpus-based word similarity and string similarity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *2*(2), 10.
4. Markov, Z., & Larose, D. T. (2007). *Data mining the Web: uncovering patterns in Web content, structure, and usage*. Wiley-Interscience.
5. Schuff, D., Turetken, O., & D'Arcy, J. (2006). A multi-attribute, multi-weight clustering approach to managing e-mail overload. *Decision Support Systems, 42*, 1350-1365.

8. The Radicati, S. (2011). *Email statistics report, 2011-2015.*
9. Xiang, Y. (2009). Managing Email Overload with an Automatic Nonparametric Clustering Approach. *The Journal of Supercomputing*, *48*(3), 227-242.
10. Yang, H., Luo, J., Yin, M., & Liu, Y. (2010). Automatically Detecting Personal Topics by Clustering Emails. *Education Technology and Computer Science (ETCS), 2010 Second International Workshop on IEEE*, 91-94.

THANK YOU