Effective Topic Modeling for Email

Hiep Hong
Department of Computer Science
San José State University
San Jose, CA 95192
hiepvanhong@yahoo.com

Teng-Sheng Moh
Department of Computer Science
San José State University
San Jose, CA 95192
teng.moh@sjsu.edu

***Abstract*— Emails have been increasingly popular and have become an indispensible tool for communication and document exchange. Because of its convenience, people use emails every day at work, at school, and for personal matters. Consequently, the number of emails people receive daily keeps on increasing, causing them to spend more time organizing the emails. People often need to classify and move email into folders so that they can go back and read them later. Most email client tools available today allow the users to filter and organize emails by defining rules on how to handle incoming emails. However, this manual process requires users to know their expected emails very well, and to make good use of these tools users need to understand how filtering rules work and how to apply them correctly. In reality, most users do not know what their incoming emails will be. The work described in this paper aims to take the burden of organizing emails away from users by using the Latent Dirichlet Allocation (LDA) [10] to automatically extract topics from emails and group them into folders of common topics. Experiments have shown that the proposed method is able to correctly group emails in appropriate topics with 77% accuracy.**

***Index Terms*—computer applications, data mining, database systems, information filtering, natural language processing, probability, random variables, text analysis, text processing**

I. INTRODUCTION

When the number of emails increases, it becomes a burden for individual users to keep up and manually organize them. Typically, when a user sees a new email, he would read its subject line and check its sender before deciding whether to read it. Depending on the priority set by the user, the email is either read right away or kept unread. For example, if the sender is someone the user knows or the subject mentions information related to some ongoing work or activities – the user would read the email right away. Otherwise, the user may keep those unread emails in the inbox or move them to appropriate folders so that he can go back and read them later. This managing sequence is effective only when the user has a small number of emails since it will become tedious and time consuming if the volume of emails is increasing. In fact, it will be overwhelming for the user to sort out and filter his emails manually in such a situation.

Having a tool that can automatically organize and group emails by topics would be of great help to email users. A user can now easily navigate through emails from topic to topic, since the title of a topic will indicate the similarity of a group of emails. In this way, the user can focus on reading emails instead of spending time on organizing emails, moving them into folders.

Such an important feature, unfortunately, has not been supported by email client tools available in the market. For example, Outlook requires users to know their incoming emails well enough to manually define filtering rules, and to handle and move emails into manually created folders. Gmail offers grouping emails into five fixed groups or tabs. No matters how many emails there are, they will have to fit into these five tabs.

The main contribution of this paper is to allow users to specify a number k, and automatically group their emails into k topics. With automatic partitioning by topics, users can read emails by the derived topics and still have the option to see all the incoming emails in the inbox. The work in this paper uses the Latent Dirichlet Allocation (LDA) [10] to automatically extract topics from emails, and partition them into corresponding topical folders. It helps users to tackle email overload where previously they have to manually classify and put emails into categorized folders to clean up their email inboxes. By extracting topics from them, similar emails can be automatically grouped and organized, freeing users from the tedious manual work.

This paper is divided into the following six sections. Section 1 introduces the email organizing problem and the motivation of this work. Section 2 provides reviews on the related work. Section 3 introduces LDA in the context of the method used in this paper. Section 4 presents the proposed method to help email users overcome the overwhelming email problem. Section 5 describes the experiment of this work that includes implementation, dataset, training method, evaluation metric, and results. Finally, Section 6 concludes this paper with future work.

II. RELATED WORK

Many researchers have been working on methods to manage overloading emails that the users are unable to catch up reading. Applying machine learning techniques, Ayodele and Shikun [1] tried to accomplish three things in email management: predict whether an email requires a reply (email prediction), group emails based on user's activities (email grouping), and summarize emails for user (email summarizer) [1]. Their goal is to save email users' time for reading email, provide a cost-effective way for managing email, and promote the efficiency of email service. The research in email management is important because it allow users to read more emails in a short time, and avoid and reduce email

congestions. It also helps users prioritize email and reduce storage space and bandwidth required for email transmission, thereby providing efficient methods to manage and organize emails.

In another work, Ayodele et al. [2] grouped emails based on a user's activities by using the Email Evolving Clustering Method (EECM), which is derived from the original Evolving Clustering Method (ECM). The EECM is a fast maximum distance-based clustering method that makes only one pass over the dataset. In the EECM method, each email is represented by a point, and each cluster has one or more points. The designated center point of each cluster moves or evolves over time as more points are added to the cluster. These center points are also called the evolved emails in the datasets. Maximum distance is calculated from the email data point to the group center. The data point is included in the group if the maximum distance is less than or equal the threshold value. A predefined function is made based on the similarity measure between email contents and the user's dictionary of favorite words found in emails. This function then determines the group to which an email belongs. Email samples come from a stream of email and the method starts with an empty set of groups. Group centers are updated as more data points are added.

In Yang et al.'s method, frequent terms in the subject and body of each email were selected and assigned weights [4]. After that, they determined how to combine email subject and body terms to determine which one can better represent the email. Emails were then clustered based on the similarity of represented terms. They experimented with how to effectively select the initial **k** centroids in K-means clustering in order to achieve good result.

Dredze et al. focused on generating summary keywords for emails using topics [5]. They used an unsupervised learning approach that requires no annotated training data. They used query-document similarity and word association to describe each email message in the context of existing topics rather than simply selecting keywords based on a single message in isolation.

The selection works based on two well-known models for inferring latent topics, Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA). Together, these form four methods, LSA-doc (Query-Document Similarity + LSA), LDA-doc (Query-Document Similarity + LDA), LSA-word (Word Association + LSA) and LDA-word (Word Association + LDA). Their objective was to generate summary keywords that are not too specific or too general. In fact, the summary keywords have to be specific enough to describe one email message but common across many email messages to be associated with coherent user concepts. Therefore, they can be representative of the gist of the email, thereby allowing the user to make informed decisions about the message.

Dredze et al. used the Enron dataset, which contains 250,000 emails from 150 users with a pre-processing operation that removes common stop words and email-specific words such as "cc," "to" and "http". Stop words are words that have no meaning by themselves and do not represent the document or email that contain them. For example, "a", "the", "about", "because" and "what" are some of the common stop words. Term frequency-inverse document frequency (TF-IDF) was used as a baseline in which the nine highest scoring words for each email, according to TF-IDF, were selected. From there, automatic foldering and recipient prediction were used as two methods of evaluation.

Joty et al. worked on exploiting conversation structure in unsupervised topic segmentation for emails [6]. They grouped the sentences of an email thread into a set of coherent topical clusters with an average of 2.5 topics per thread in the BC3 email corpus. That created a prerequisite for other higher-level conversation analysis such as summarization, information extraction and ordering, information retrieval, and intelligent user interfaces. The motivation for their work was that extensive research has been conducted in topic segmentation for monologues and synchronous dialogs, but none has studied the problem of segmenting asynchronous multi-party email conversations. Models that were proved successful in monologues or dialogs might not be effective when applied to email conversations. Their work was to fulfill the missing piece for segmenting asynchronous multi-party conversations in email.

Joty et al. captured the conversation structure at the fragment level in the form of a Fragment Quotation Graph (FQG) [6]. They also applied Latent Dirichlet Allocation (LDA) together with Lexical Chain Segmenter (LCSeg) to construct a finer level structure of the underlying conversations. They performed a two-phase pilot study before carrying out the actual annotation. In the first phase, five university graduate students selected to do the annotation. The instruction manual was revised based on the students' feedback and their disagreements with the computer annotations. In the second phase, a university postdoc performed the annotation. For the actual annotation, three computer science graduates, who are native speakers of English, were selected to annotate 39 threads of the BC3 corpus. On average, it took them seven hours to annotate the whole dataset.

To evaluate their method, Joty et al. measured the local and global structural similarity between human annotations and system's output by three metrics 1-to-1, loc_k and *m*-to-1. The 1-to-1 measures the global similarity between two annotations. The loc_k measures the local agreement within a context of *k* sentences, and the *m*-to-1 measures how much the annotators agree on the general structure.

Pan et al. [7] developed the TIARA (Text Insight via Automated Responsive Analytics) system that automatically generates a time-based visual text summary that conveys key topics derived from a large collection of text. It supports a rich set of interactions that allow users to further explore the created visual summary and examine the text collection from multiple aspects.

The TIARA system consists of four main components, pre-processing, topic summarization, temporal topic segmentation and text visualization. First, the pre-processing component extracts the title and the main body text (e.g., the

body of a news story) and related metadata (e.g., the author and time information) from each document. Second, the topic summarization component uses Latent Dirichlet Allocation to automatically extract a set of topics, each of which is

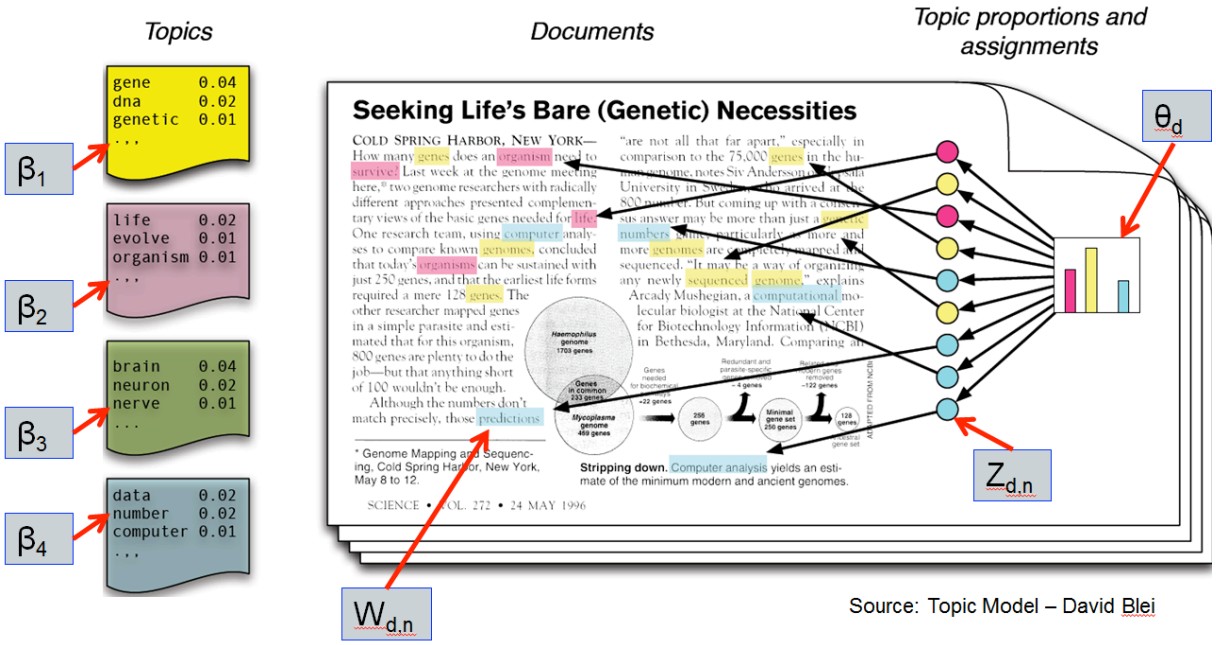

Fig.1. Example - LDA Generative Model

associated with a set of documents. Third, the temporal topic segmentation component has three sub-components, sub-topic analysis, internal determination and keyword selection. Given a derived topic (T) and a set of documents covering the topic {DT1, …, DTM}, the subtopic analysis subcomponent identifies a set of sub-topics {ST1, ..., STK} that satisfy a set of semantic, temporal, and visualization constraints. The interval determination subcomponent then computes the temporal boundary for each subtopic ST [t1, t2]. After that, the keyword selection subcomponent identifies a set of keywords to represent each derived subtopic. Finally, given the derived topics and subtopics, the text visualization component generates a time-based, interactive visual text summary. The data sets consist of more than 7000 emails over the course of two years and more than 13,000 New York Times articles in six months. The TIARA system ran LDA to derive *N* topics where *N* was set to 10 for emails and 30 for news articles. For each derived topic, it performed temporal segmentations and showed the results.

III. LDA

Latent Dirichlet Allocation or LDA introduced by David Blei et al in 2002. The method discovers the hidden themes in large text document collections such as Wikipedia articles, blogs and emails, and then it annotates the documents based on those themes. After that, all the annotations can be used to organize, summarize and search. For instance, this project uses LDA to organize emails and group them by topics. A LDA topic is a probability distribution over a collection of words. Fig.1 shows an example of four LDA topics. The purpose of a topic is to provide a theme of a collection of documents or emails. For example, a collection of corporate emails could have topics on promotion, hiring, strategy, travel and finance.

A generative model is a statistical relationship between observed and latent or unknown random variables that specifies a probabilistic procedure to generate the topics [17]. In generative model, each email is imagined to be written in a way such that the topic proportion for the email can be randomly selected given a distribution of topics. Fig.1 shows the document has three topics assigned each with a different proportion θ_d. After the topic proportion has been as assigned, each term from those topics can be randomly drawn to put in the email.

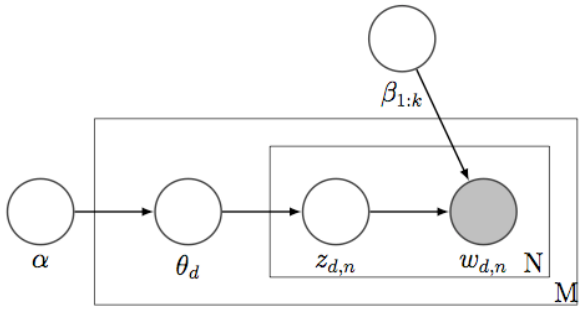

Fig.2. LDA Graphics Model

α: proportions parameter that determines the number of topics for each email.
β: topic distribution that is a collection of words each with a probability.
θ: per-email topic proportion. For example, an email may have three topics, 20% of the words in the email belongs to topic 1, 30% of them belongs to topic 2, and 50% of them belongs to topic 3.
z: per-word topic assignment since a word exists in every topic distribution with a different probability. For example, the word "gene" may have 4% probability in the topic distribution β_1 but it may have 10% probability in the topic distribution β_2.
w: observed word of the email used for learning the topics.
N: number of words in the email.
n: the n-th word in the email.
M: number of emails in the dataset.
d: the email d-th in the dataset.
K: number of topics in the dataset.

LDA provides a generative model that describes how the emails in a dataset were created [17]. Fig.2. shows the graphical mode of LDA where nodes represent random variables and edges indicate dependence between them. Shaded node is observed or known, and un-shaded nodes are latent or unknown. Plates represent replicated variables. In the context of this paper, the dataset is a collection of M emails each of which is a collection of N words. The LDA generative model describes how each email obtains its words. Initially, the number of topic distributions, K, is assumed to be known. For instance, a collection of corporate emails may have five topics such as promotion, hiring, strategy, travel and finance. Each topic distribution is a multinomial that contains V elements where V is number of terms in the email corpus. Let β_i be the multinomial for the i-th topic, where the size of β_i is V, $|\beta_i| = V$. Given these distributions, the LDA generative process proceeds as follows:

For each email:
(i) Randomly choose a per-email topic proportion (a multinomial of length K)
(ii) For each word in the email:
  a. Per-word topic assignment - probabilistically draw one of the K topics from the distribution over topics obtained in step (i), say topic β_i
  b. Word selection - probabilistically draw one of the V words from the multinomial β_i

The generative model indicates that each email contains multiple topics in different proportion. For example, one corporate email may have two topics with 40% of the words belongs to the topic on promotion and 60% of the words belongs to the topic on hiring.

The generative process assumes that the K topic distributions are known, but only words in the emails are known in reality. Therefore, the next step after the generative mode is posterior inference in which the K topic distributions must be learned by a statistical inference algorithm, collapsed Gibbs sampling. Each iteration of Gibbs sampling proceeds by drawing a sample for z, per-word topic assignment, for each word in each email. Given a sample of z, the per-email topic proportion, θ, for each email can be estimated. At the end of all the Gibbs sampling iterations, each email has a certain probability for each topic in the K topic distributions.

IV. PROPOSED METHOD

[1] and [2] proposed a method to group email based on a user's activities whereas this paper allows the user to group emails based on latent or unknown topics of the emails that will be derived by learning the LDA model. Moreover, it is appropriate for emails of common topic to be in the same folder. Compared to a user's activities, topics of emails are more generic giving the users a new way to read emails. Moreover, the EECM method in [2] requires the user to manually annotate a dictionary of favorite words found in the emails, whereas the user only needs to specify the number of topics when grouping emails by topics.

The method used in [4] converts each email into a vector space model from which the similarity of emails can be calculated using the cosine similarity measure. The drawback of the vector space model is that it suffers from curse of dimensionality resulting in large space and time complexity. In machine learning, when each data has high-dimensional features and each feature has multiple values, the trained model will become spare. That will result in less accurate classification even though it requires more space to store the data and more time to train to the model.

The limitation of [5] is that each email is summarized into nine keywords. As a result, the content of a long email may not be sufficiently represented. The work in this paper overcomes that issue by using all the remaining keywords after preprocessing each email before using them to train the LDA model.

The method in [6] is very specific and may not cover the general case where emails are not part of conversions such as notification and confirmation emails where replies are not accepted. The work in this paper considers all emails since they may share common topics.

The work in this paper is most related to the work in [7] which uses LDA to derive topics for emails. However, instead of showing topic temporal segmentation as a result, the work in the paper helps users to organize their growing inboxes by grouping emails into folders of common topic so that they can focus on reading emails.

This paper uses the LDA [10] to model topics for emails since LDA requires no manual annotations of the dataset like other methods as shown in Table I. that compares methods used in the related work with LDA. As a result, it will save users a lot of time because they only need to specify the number of topics when they organize and analyze their emails. In other words, emails can be partitioned into a number of groups each of which is represented by an LDA topic represented by a set of words. These groups of emails can also be interpreted as folders of emails with common LDA topic or similar content. This method is extremely helpful for users who have a high volume of emails in their inboxes because

manually reading and classifying each email and then organizing it into a folder would be a tedious and unpractical task. With the LDA model, users can understand the general content of their emails better and can focus on reading emails based on a particular topic. One very useful feature of the LDA method is its capability of handling high volume of documents or emails since that was one the motivations of the authors [10].

TABLE I. COMPARISON OF EMAIL GROUPING METHODS

Method	**Operation**	**Number of topics**	**Number of passes over dataset**	**Manual annotations**
EECM [2]	Automatic	Variable	One	Yes
K-means [4]	Automatic	Variable	Multiple	Yes
FQG [6]	Semi-automatic	Variable	Multiple	Yes
LDA	Automatic	Variable	One	No

Most email processing features today are able to handle high volume emails. However, the email-filtering feature still requires manual configuration, and the email-grouping feature is limited by a fixed number defined by the developer. For example, the Outlook application requires the users to know their incoming emails well enough to manually define filtering rules to handle and move emails into manually created folders. Gmail offers grouping emails into five fixed groups or tabs. No matters how many emails there are, they will have to fit into these five tabs. TABLE II shows the comparison and contrast on the important email processing features between current applications used by many users in the market and the proposed method.

TABLE II. COMPARISON AND CONTRAST OF FEATURES

Email Processing Features	**Current tools**	**Proposed method**
Handle high volume of emails	Yes	Yes
No manual configuration required for filtering	No	Yes
Group emails by any number of categories	No	Yes

Fig.3 shows the flow diagram for the proposed method in which the user can specify the value of n topics from the graphic user interface. The backend component will run the LDA model to produce the result.

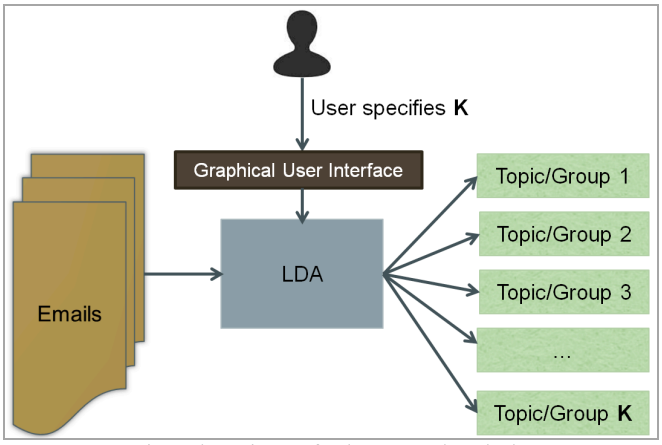

Fig.3. Flow Diagram for the Proposed Method

V. EXPERIMENT

A. Implementation

For the purpose of experiments, a Java application shown in **Error! Reference source not found.** was developed for this paper. The application incorporates the Mallet toolkit [15] to perform LDA modeling on emails. It has the graphical user interface to show various features provided by Mallet as well as preprocessing features that specifically handle importing emails before learning the LDA model. It provides the interface to the optional Mallet training feature where the number of threads can be specified for parallel training. This feature improves the training performance when the data set has tens of thousands of emails. The application also has the option to turn on hyper-parameter optimization, which allows the LDA model to better fit the data by allowing some topics to be more prominent than other topics. The application has three main tabs, preprocessing, training and post-processing which can be optional.

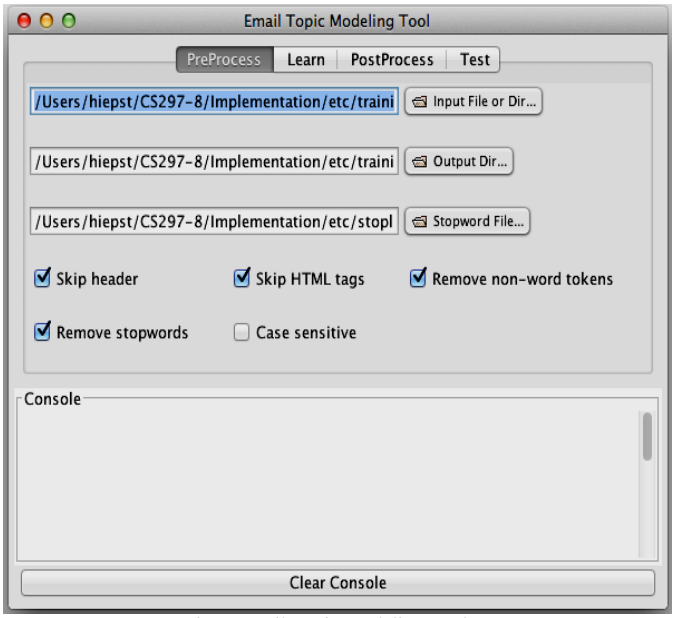

Fig.4. Email Topic Modeling Tool

First, the preprocessing tab contains features that preprocess emails during importing to create a Mallet data structure, which is a vector of tokens before passing to training. Its features include skip header, skip HTML tag, remove non-word tokens, remove stop words, and case sensitive. The skip-header feature removes the header of the email so that the body message is used for training. The skip HTML tag feature removes HTML tags that exist in some HTML-formatted emails. The remove-non-word-tokens feature filters emails further by removing tokens that are non-alphabetic. The remove stop words feature removes all stop words from emails. A predefined list of non-domain-specific stop words is contained in a text file, which is specified in the preprocessing tab. To generalize the Enron email data set, Enron-domain-specific terms such as Enron and EES (Enron Energy Service) are considered to be stop words included in the predefined list. The case-sensitive feature provides the ability to differentiate uppercase and lowercase tokens. The experiments were performed under the case insensitive condition, which treats uppercase and lowercase tokens alike.

Second, the training tab's features include the number of topics, the number of iterations, the number of words with highest probability to be displayed in the result, and the topic proportion threshold. The number of topics feature allows setting the number of topics required for training the LDA model. This is the key feature for training and it depends on the type and volume of the email dataset. Moreover, it depends on the user's perspective of how general he wants the topics to be. For instance, the Enron data set may have many specific topics, but at a higher level of generalization, the number of topics may be much lower. Another training feature is the number of iterations the training process will sample the imported dataset. The number of top words feature allows changing the number of top words in a topic to be printed to the training result output.

The topic proportion threshold feature allows specifying the threshold of likelihood that an email has a topic that is included in the result. For example, when the number of topics is specified as five, each email in the dataset will have a probability for each topic. If the likelihood is below the topic proportion threshold, then it will be removed from the list of topic for a particular email.

Third, the post-processing tab implements the semi-automatic labeling feature for LDA topics, which contains a list of words or tokens. Post-processing can be optional because it depends on the user's preference whether he wants to see either the original LDA topic containing a set of terms or a single-term label. In the latter case, a list of three-topic-words-to-a-label is manually pre-defined and added to a text file required for post processing. For every three words in a topic, a label is manually defined and stored in the text file. For example, [transaction, account, report] may be labeled as "finance". When the post processing operation runs, it will automatically match the top three words in a trained topic with available labels. Since the LDA model assumes the bag-of-words condition, any order of these three words can be matched. In fact, each topic can be considered as a bag that contains three topic words, which can be shuffled in any way without affecting the meaning of the corresponding topic. If no match is found, a new entry can be added to the label list to help minimize mismatch in the next post processing. This also demonstrates the use case in which email users can label a LDA topic that seems most appropriate for them.

B. Dataset

The experiments in this paper utilized the Enron email dataset that has about 500k emails from 150 users, and was originally made public by the federal government during Enron scandal investigation [16]. It is the only real email dataset with large enough number of emails to be used for machine learning. It was preprocessed so that all attachments were removed from the emails.

Email folders from each Enron user are considered and randomly selected as the input data for training the LDA model. For the evaluation purpose, the number of emails in each folder is set to 20. Therefore, for every experiment run, the training data set size is 100 emails in five different randomly selected folders. For example, in one of the experiments, five email folders were randomly selected from three Enron users, baughman-d, beck-s and davis-d. The folders selected were baughman-d/power/legal_agreements, beck-s/commercialization, beck-s/congratulations, beck-s/recruiting and davis-d/finanial_operations. Each email is then manually read to make sure its content agrees with the folder label. Otherwise, it will be replaced with a different email whose content does agree with the folder label. This benchmark is to ensure that all emails in the same folder should share the same label, and those emails can be compared with the trained LDA model to see if they have the same topic.

C. Training the LDA Model

For every set of five randomly selected of folders, ten rounds of training the LDA model are performed. After the

folders are selected, the pre-processing step is performed before training step runs. Only body messages are used for training. Email headers, non-alphabetical words and stop words are removed during the pre-processing operation. The basic requirement for training is that the number of topics specified for training must be greater than two ($t >= 2$). The number of topics is set to five for comparing with the five folders. Twenty-five hundred iterations of Gibbs sampling are performed during the training and the topic proportion threshold is set to 0.02. Experimental results show that the threshold below 0.02 does not yield correct classifications on the topics that emails may have. The resulted LDA model consists of five topics and each contains a list of words. For evaluation purposes, the top five words in each topic are printed in the output to both the application console and the text file. Each email has a probability for each of the five topics.

D. Validation

Each LDA topic has more 100 words each with different probability. The order of these words ranges from the highest to the lowest probability. Only the first five words with highest probabilities are selected to represent the LDA topic, the rest of the words are ignored.

10-fold cross validation was used with a total of 100 emails which were split into 60% for training and 40% for testing.

E. Evaluation Metric

The trained LDA model on the Enron data set is evaluated based on the following accuracy definition.

- Primary main topic is the topic that majority of emails in the same folder has as the one with the highest probability among a list of K topics.
- Topical threshold is the percentage difference between two probabilities of two different topics.
- Secondary main topic is the topic with the highest probability for some emails but it is not the topic that majority of emails in the same folder has.
- Emails with the primary main topic are counted toward the total of correct classifications.
- For emails that have the secondary main topic as the first and the primary main topic as the second in the list of K topics ordered from the highest probability to the lowest probability, if their probabilities are within 5% difference then the emails are also counted toward the total of correct classifications.

Fig.5 shows an example of the training result of one email folder. Number of topics is set to be five since five email folders are used for training. The training result consists of a list of emails. Each has five pairs of topic and the corresponding probability that the email has that particular topic. The percentages are sorted in the descending order from left to right. Each email is considered to have the topic with the highest probability. Among the five topics, the major topic is the one that a majority of emails in the same folder has. For example, **Error! Reference source not found.** shows that the major topic is three. If an email has topic t_1 with the highest percentage and the major topic t_2 with the second highest percentage, which is within 5% of the first one, then that email is also considered to have the major topic t_2. In **Error! Reference source not found.**, the email "congratulations16", highlighted in green, has both topic 4 and 3 with 21.8%, so it is considered to have the major topic 3. The email "congratulations9", highlighted in green, has topic 2 with 26.7% as the highest and topic 3 with 25.5%. Since the percentages are within 5%, email "congratulations9" is considered to have the major topic 3. On the other hand, the email "congratulations3", highlighted in yellow, has topic 2 and 3 with the probability more than 5% different and so it does not have the major topic 3.

The reason for selecting the second topic for the major topic count is that each LDA topic associates with a number of emails. Based on observation of the selected emails used for training, one email may associate with or have more than one topic. When two topics are only 5% different in the likelihood, chances are the email can fit in either topic and it still makes sense. In general, email contents are very variable since they do not fall into specific domains such as medical and astronomy.

F. Results

The average accuracy is 77% based on the majority topic in each folder. The training results show that majority of emails in the same folder have the same LDA topic. This implies that the trained LDA model correctly groups the emails similar to the Enron email user. Moreover, words in the LDA topics correctly represent the original folder labels. For example, most emails in "recruiting" folder have the topic [scholarship mba time bus dinner]. It was verified that emails in the "recruiting" folder talks about recruiting scholarship recipients including MBA students and that they were having the reception dinner with those scholarship recipients.

	topic	%	topic	%	topic	%	topic	%	topic	%
ns/congratulations10	3	23.53	2	23.53	0	19.12	4	17.65	1	16.18
ns/congratulations11	0	27.59	4	18.97	3	18.97	2	17.24	1	17.24
ns/congratulations12	3	21.82	4	20.00	1	20.00	0	20.00	2	18.18
ns/congratulations13	3	26.32	0	21.05	4	17.54	2	17.54	1	17.54
ns/congratulations14	3	25.37	0	20.90	4	19.40	1	17.91	2	16.42
ns/congratulations15	3	22.64	2	20.75	4	18.87	1	18.87	0	18.87
ns/congratulations16	4	21.82	3	21.82	0	20.00	2	18.18	1	18.18
ns/congratulations17	3	22.22	4	20.37	2	20.37	1	18.52	0	18.52
ns/congratulations18	3	28.13	4	20.31	0	18.75	1	17.19	2	15.63
ns/congratulations19	3	23.64	4	21.82	2	18.18	1	18.18	0	18.18
ns/congratulations2	3	85.54	4	5.40	2	3.31	1	3.14	0	2.61
ns/congratulations20	3	22.64	0	20.75	4	18.87	2	18.87	1	18.87
ns/congratulations3	2	33.10	3	22.54	0	20.42	4	11.97	1	11.97
ns/congratulations4	3	24.56	4	22.81	2	17.54	1	17.54	0	17.54
ns/congratulations5	3	24.07	2	20.37	4	18.52	1	18.52	0	18.52
ns/congratulations6	2	23.08	4	19.23	3	19.23	1	19.23	0	19.23
ns/congratulations7	3	73.96	4	8.58	1	7.10	0	5.62	2	4.73
ns/congratulations8	3	25.35	0	23.94	2	18.31	4	16.90	1	15.49
ns/congratulations9	2	26.74	3	25.58	1	16.28	0	16.28	4	15.12

Fig.5. Example - Training Result of One Email Folder

VI. CONCLUSION AND FUTURE WORK

This paper has introduced a new way to organize the high volume of emails. By modeling topics on emails, a general summary view can be constructed. Each topic associates with a number of emails. By looking at each topic, the user can have a general indication of the contents of the associated emails. The contribution of this paper is to free users from manually defining rules to filter emails. It also helps them to avoid the limitation of manual email grouping by allowing them to specify any number of topics or folders they want the application to automatically group emails. From there, the users can read emails by the derived topics and still have the option to see all incoming emails in the inbox. Future work can enhance the preprocessing step by incorporating the named-entity recognition NER method into the application, this way people names may be automatically removed from emails instead of specifying them as stop words in the current implementation. In addition, the accuracy of the LDA model may be improved with additional post-processing that can correct any misclassifications in the current round of training, which can then be used for the next round. This would be very useful, since users can correct misclassified emails based on their personal judgments and understanding of their emails.

ACKNOWLEDGMENT

The first author would like to thank Dr. Robert Chun and Prof. Ron Mak, both at San José State University, for their constructive reviews and inputs that improve the quality of the paper.

REFERENCES

[1] Ayodele, Taiwo, and Shikun Zhou. "Applying Machine learning Algorithms for Email Management." *Pervasive Computing and Applications, 2008. ICPCA 2008. Third International Conference on*. Vol. 1. IEEE, 2008.

[2] Ayodele, Taiwo, Shikun Zhou, and R. Khusainov. "Evolving email clustering method for email grouping: A machine learning approach." *Applications of Digital Information and Web Technologies, 2009. ICADIWT'09. Second International Conference on the*. IEEE, 2009.

[3] Davies, David L.; Bouldin, Donald W., "A Cluster Separation Measure," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* , vol.PAMI-1, no.2, pp.224,227, April 1979.

[4] Huijie Yang; Junyong Luo; Meijuan Yin; Yan Liu, "Automatically Detecting Personal Topics by Clustering Emails," *Education Technology and Computer Science (ETCS), 2010 Second International Workshop on* , vol.3, no., pp.91,94, 6-7 March 2010.

[5] Dredze, Mark, et al. "Generating summary keywords for emails using topics." *Proceedings of the 13th international conference on Intelligent user interfaces*. ACM, 2008.

[6] Joty, Shafiq, et al. "Exploiting conversation structure in unsupervised topic segmentation for emails." *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2010.

[7] Pan, Shimei, et al. "Optimizing temporal topic segmentation for intelligent text visualization." *Proceedings of the 2013 international conference on Intelligent user interfaces*. ACM, 2013.

[8] Dey, Lipika, et al. "Email Analytics for Activity Management and Insight Discovery." *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences on*. Vol. 1. IEEE, 2013.

[9] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." *the Journal of machine Learning research* 3 (2003): 993-1022.

[10] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." *Advances in neural information processing systems* 1 (2002): 601-608.

[11] Hoffman, Matthew D., David M. Blei, and Francis R. Bach. "Online Learning for Latent Dirichlet Allocation." *NIPS*. Vol. 2. No. 3. 2010.

[12] Porteous, Ian, et al. "Fast collapsed gibbs sampling for latent dirichlet allocation." *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008.

[13] Canini, Kevin R., Lei Shi, and Thomas L. Griffiths. "Online inference of topics with latent Dirichlet allocation." *International conference on artificial intelligence and statistics*. 2009.

[14] Yang, Tao, and Dongwon Lee. "On handling textual errors in latent document modeling." *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*. ACM, 2013.

[15] McCallum, Andrew Kachites. "MALLET: A Machine Learning for Language Toolkit." http://mallet.cs.umass.edu. 2002.

[16] "Enron Email Dataset." https://www.cs.cmu.edu/~./enron/. 2009.

[17] Reed, Colorado. "Latent Dirichlet Allocation: Towards a Deeper Understanding." 2012.