Automatic Classification for Unlabeled Email Messages into Folders

Fucheng Zhu *

College of Arts and Sciences, Boston University, MA, United States

* Corresponding Author Email: rockfz@bu.edu

Abstract. Imagine returning from an excused absence because of Covid-19 or any force majeure alike, and having to immediately face 300+ unread emails; getting overwhelmed by emails has become part of office workers' daily routine. Numerous pieces of research have shown effective methods to categorize email messages, detect potential harassment, and even automatically send a reply. But still, email is an interesting type of text to analyze and gives rise to many challenges. First discussing the challenge in the problem, this paper aims to research, study, and propose a method that can deal with a specific challenge: making folders out of income email messages and then classifying emails automatically. By cooperating basic methods, techniques, and algorithms, an intuitive program is developed that can perform the task with the given public email dataset. The method is then expected to raise prospects for future investigations and improvements in performance and robustness.

Keywords: Email Classification, Unsupervised Learning, Natural Language Processing, Vector Space Model.

1. Introduction

Since its invention in the 70s, electronic mail has grown in user size and popularity. Despite the appearance of alternative platforms that perform remote communication and social networking tasks in our daily life, email is still of the top choices for business communication use. Nowadays, people in college and all kinds of working spaces receive around 100 emails daily. Processing information from emails is an essential part of everyday work. As the popularity of email keeps growing, the time spent on handling email messages will only increase. In reality, not all emails from a user's inbox have the same significance level. They can vary greatly from nonpersonal and noncritical spam emails to the most important work email from the manager. Thus, preprocessing emails into categories increases working efficiency and shortens the time someone spends dealing with emails [1]. This type of classification problem receives growing interest as a field of text mining and machine learning studies.

Email messages as unstructured text consist of a subject, user names, and body content. Such text parts allow many possibilities for natural language processing and text mining techniques to take a role in conducting experimentation and analysis. Most of the research in classification problems includes the bag-of-word model, stemming, and stop word list choice to construct a method to explore the potential functions of email texts. Email has determined fields, such as sender and receiver, which are also useful in creating functional folders in the email management system. Such fields contribute the most to problems like personalized email prioritization and social network groups. This type of study utilized classification and clustering techniques to gain insights into the predictive power of using private information to solve the email overload problem. Also, the relationship between the subject, user name, and the email message is considered in many research studies to analyze and construct social network models [2]. Other than those, studies based on email contents aim to build models that have the predictive power to classify documents into predestined folders. This paper's main text body to analyze is the email message. From the rich contents of the message itself, we can extract concepts and patterns to develop methods that perform various information retrieval tasks.

Nowadays, machine learning methods targeting email messages can be used in a wide range of applications. An example is using the Naïve Bayesian classifier to classify messages into spam or

non-threat [3]. This method is designed to have the predictive power to put emails into classifications, whose accuracy can be directly labeled by the rate at which the email is correctly classified. Such classification methods have a substantial social impact for spam spread directly related to harassment with no good purposes.

For most email-using situations, one challenge is realizing and tackling the classifying problem's unlabeled nature. It is known for many practical and effective classifiers that the training set required to obtain a satisfying performance score is relatively huge. In general, hundreds or more training examples are needed to output accurate enough results. Such classifiers often require hand-labeled training set with laborious data preprocessing work. In real-life workplaces, the email inbox may not have pre-determined classes, and the time needed to genera enough email data may take weeks. Then, the training set becomes only the raw messages of the email. In this research, the labels for each email are determined by the unsupervised clustering algorithm. Either word with the highest frequencies will be treated as the most critical message in a document.

In this paper, an unsupervised email classification method that separates unlabeled email messages into explainable discrete clusters is designed and tested. The experiment is conducted using the Enron Corpus, a public email, and natural language processing, data mining, and statistical technique to develop a system that separates email messages into automatically chosen folders.

2. Methodology

2.1. Data

This research used the Enron corpus to perform an email classification task. The raw dataset contains over 600,000 emails belonging to 158 users [4]. This dataset was made public during the legal investigation concerning the Enron corporation.

Several folders are removed from the original corpus to develop a method that aims to classify email messages in a meaningful way automatically. For example, the system-created folders like "discussion thread" and the folder named "all messages" are removed since they contain duplicate messages appearing in other folders.

During each experiment, the training set is chosen to be a subset of the entire corpus comprising 15,000 randomly sampled emails. The data was cleaned by removing all emails with the empty sender, receiver, or content and extracting the email content. Also, since the method is developed to tackle text document, any email message with only URL or HTML format content are neglected. After cleaning, each experiment has over 14,000 valid plain text email messages.

2.2. Tokenization and vector space model

To start applying the k-means algorithm, the experiment uses the technique to convert the document as vectors of weighted word frequency. The construction process has the following characteristics: the case of the word is ignored, and all words in the entire corpus are extracted. The non-content-bearing stop words are eliminated. For each word, its frequency over all words is calculated [5].

This research utilized the term frequency-inverse document frequency (tf-idf) as the measurement for the significance of each word in the entire corpus. Tf-idf is a statistical representation of the tokens that shows how relevant the term is to the entire document.

The tf-idf model is the product of term frequency and inverse document frequency. The term frequency measures the time of appearance of the specific word over the entire number of words in the corpus. The inverse document frequency assigns weights to different keywords. In the document, a word that is too frequent is assigned to a lower score, and a word with a smaller number of appearances is assigned to a higher score [6].

For example, if the word "Enron" appears 5000 times in the corpus with 60,000 unique words, then "Enron" is assigned with a tf score of 500 / 60,000 = 0.008. And it appears in 500 documents out

of 3,000 documents; it is assigned with an idf score of log(3000/500) = 0.778. Then, “Enron” has a tf-idf score of 0.008 * 0.778 = 0.006.

Moreover, if a word appears too often or too few times in all documents, it can be considered meaningless in the context of text mining.

2.3. Stop word list

To gain meaningful interpretations of the clustered results from the unsupervised algorithm, a stop word list is necessary to trim the unimportant information. The construction of the stop word list for experiments conducted on the email dataset can be a complex decision-making process. Three primary reasons contribute to the complexity, as presented below:

(1) Emails are mainly unstructured messages, making it hard to define a specific beginning, body paragraph, or ending to create meaningful contexts. Email is designed not to have a very long structure compared to news articles and research papers; most emails have only a few sentences or paragraphs.

(2) Email messages are less carefully constructed than other published articles or research papers. It is often the case that the user sends personal messages that contain badly cased words, misspellings, or grammatical errors. Also, using emoticons as a favored communication tool specified to email users and words that only make sense in a particular context is different from other types of literature.

(3) Email can be in other formats other than plain text. In the dataset of choice, HTML and rich text formats has non-negligible occurrences.

For the experiment conducted in this research, the stop word list is constructed with consideration to avoid all possible causes of deviating from making meaningful interpretations to the clusters and further making automatic classification possible [7].

The stop word list consists of the typical English stop word list with 432 words, including determinants, conjunctions, pronouns, and prepositions [8]. Some frequent verbs are trimmed, such as 'do,' 'can,' etc. To cut off more irrelevant information, words like "recipient" and "original" are considered stop words because they only provide limited information—the original address where the email is forwarded or replied. Words like "hou," "com," "ect," "enron,"and "ees" are considered to be less informative. In this dataset, they represent the domain of the email address of the Enron company located in Houston. Moreover, "mail," "please," and "thanks" are trimmed off. In most of the emails, they appear in the context of non-content-bearing words. Last, the integer list from "0" to "60" is excluded because they appear in email messages as the timestamp in which the original message is forwarded or replied.

Also, the words that appear in less than five emails or more than 50 % of the files are considered insignificant and trimmed off.

2.4. Unsupervised learning algorithm

Due to the unlabeled nature of the email classification problem, unsupervised machine learning algorithm can first perform the task of clustering data set in the VSM. Here, the K-means algorithm is employed to generate the clusters of the email messages to recognize the general pattern in which the email contents are structured. A cluster of emails is considered to have the most relevant messages, and thus is designed to go into similar folders. Each cluster's top features represent the words most representative of the email contents [9].

2.5. Cosine similarity score

The cosine similarity gives the semantic relationship between two vectorized documents. In this paper, this metric calculates the similarity between the text documents by considering the cosine value. The similarity between the two document vectors is [10]:

$$sim(A,B) = \frac{A \cdot B}{||A|| \cdot ||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} {A_i}^2} \sqrt{\sum_{i=1}^{n} {B_i}^2}} \quad (1)$$

2.6. Experiment design

2.6.1 Data extraction

In this experiment, each time, around 14,000 email messages contribute to 460,000 total words randomly selected from over 600,000 emails. The email dataset is parsed into a data frame that contains the information of the email and the body message. During the extraction process, invalid emails with an empty sender or receiver or with a format other than plain text are discarded.

2.6.2 Vectorization and construction of VSM

The data is then tokenized into a vector space using the tf-idf score of each word over all documents. Stemming and stopping word list is then applied to trim unnecessary words in the vector space model.

2.6.3 Apply k-means algorithm

Then, apply the k-means algorithm with initial parameters of k = 5 to the document-term matrix representation of the email messages. The parameter is chosen as a default. More evaluations can be done using different number of clusters. Here, different data points are separated in clusters where the distance between the document vector and the centroids is minimized. This experiment assumes that the distance reflects the similarity between emails. Therefore, the overall similarity within each cluster is the greatest.

2.6.4 Generating folders and assign email

The folder names are generated by selecting the top 3 features from each cluster, without duplicate. The folder names are transformed and normalized into the same vector space as the email message document vectors. Find three folders with the highest cosine similarity score for each email and assign the email to the folders accordingly.

3. Results

The data is cleaned by removing emails with an empty sender, user, or content and messages in formats other than plain text. A total of 14,037 emails remained in the training corpus. The corpus is then tokenized. Each word in the corpus represents a feature. Then, the document-term matrix is constructed by calculating the tf-idf score of each term. Table 1 below shows the top ten significant terms generated using the document-term matrix and their scores.

Table 1. Top ten features in the document term matrix

Feature	Score	feature	score
solberg	0.234748	davis	0.193490
slinger	0.234593	geir	0.172246
guzman	0.230731	harrison	0.157406
pete	0.227819	causholli	0.155915
ryan	0.215304	leaf	0.155605

Then, applying the k-means algorithm separates the emails into five distinctive clusters. The VSM provides a way to show the five clusters in a scatter plot with dimension reduction that only maintains the top 2 principal components. Figure 1 shows the scatter plot form of the clusters.

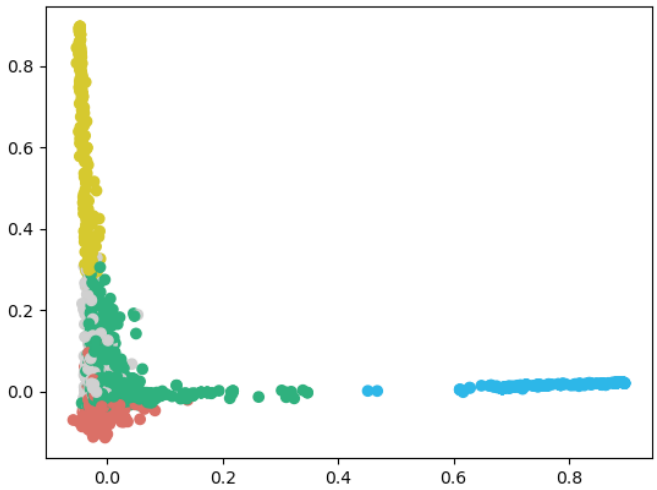

Figure 1. five clusters of email messages in VSM after applying the k-means algorithm for 100 iterations

Each cluster should be interpreted as a different set of emails belonging to different folders. The top features generated from each cluster are shown in Figure 2.

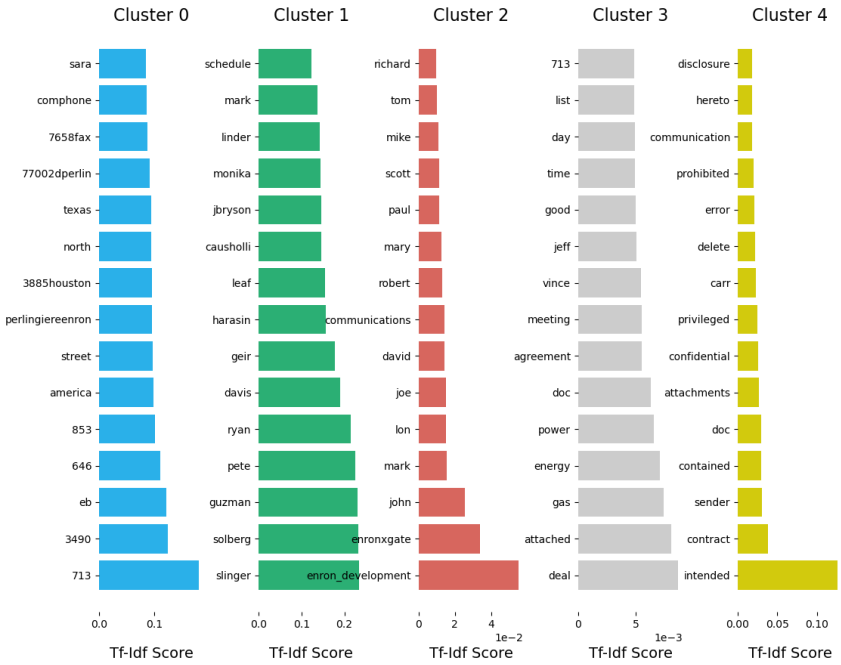

Figure 2. 15 most significant words in each cluster of emails

The top 3 most significant words are selected from each cluster as the folders' names. The first cluster's most frequent terms are person's names. It provides insight into one category of emails that, whether introduces the person or contains personal information like a business card. Cluster 1 and 2 contains emails that has a general topic of deals or energy, as the Enron company has business in the commodity and energy. Cluster 3 has shown a tendency to have documents or other files attached with it. The last cluster, after looking into the email messages, it found to have the contact numbers, including phone number and fax numbers starting with 713 as it is the area number Houston in the 90s.

The cosine scores between the document vector and the normalized folder label vector are recorded for each email in the training set. Each email is put into the folders according to the three highest scores that the email is assigned. Figure 3 shows an excerpt of all 12 folders.

```
# of folders 12
Folder label: 713
email11186->email7059->email11837->email9043->email8171->email13959->email1747->email11436->email13540->email10291->...
Folder label: 3490
email3466->email1786->email7072->email807->email4022->email13946->email12804->email7062->email10325->email3371->...
Folder label: slinger
email11898->email9505->email10945->email10308->email4719->email67->email12342->email10323->email12875->email2008->...
Folder label: solberg
email9815->email7891->email12342->email6240->email12151->email2008->email6708->email9505->email11340->email6717->...
Folder label: guzman
email6067->email768->email6717->email7901->email6165->email3252->email6987->email2493->email11266->email10945->...
Folder label: john
email12548->email2006->email2449->email5539->email12602->email5036->email7355->email3433->email6985->email3109->...
Folder label: deal
email7584->email8844->email6093->email4754->email2168->email1178->email7008->email1313->email2825->email9527->...
Folder label: attached
email4153->email3812->email9506->email12750->email1867->email294->email3647->email5621->email1559->email4166->...
Folder label: gas
email6270->email11826->email9719->email1388->email8625->email7735->email12239->email14120->email6543->email9543->...
Folder label: intended
email12576->email13438->email13437->email13476->email7940->email5192->email12623->email734->email12468->email1849->...
Folder label: contract
email7037->email9281->email5773->email9208->email1745->email7581->email6067->email14168->email5743->email8766->...
Folder label: sender
email5463->email8421->email2590->email6834->email12263->email7731->email12909->email4612->email8329->email12954->...
```

Figure 3. The top ten emails from each folder, sorted by index

4. Conclusion

In general, the email classification problem has profound effects in the modern workplace. It differs from typical text classification problems for email's unlabeled nature, subjective structure, and non-monolithic means of determining the classes' names. The Enron Corpus provides the public with a convenient way to conduct experiments on this unique text. Taking advantage of it, and using several NLP and machine learning techniques such as tf-idf models, stop word list, clustering, and vector similarity measurements, this paper shows the challenges of unlabeled email classification problems and proposes an intuitive method to explore the possibility of generating automatic folders and classifying emails based on them. The method has a complete pipeline to split the data, extract and clean email messages, tokenize documents, and show the results of emails in each cluster. For the experiment, efforts are made to give back a meaningful result of several specific folders and classified emails. Still, more evaluations are needed to prove the effectiveness and performance of this method. More test cases considering different dataset size and algorithm parameter is desired to show the robustness.

The experiment is designed to go through all the necessary steps to reach the final goal of classifying emails into folders. The algorithm can be improved in various ways to achieve better accuracy. For example, NGram can be applied to substitute the current process of choosing the most frequent words from the VSM. Despite NGram’s structure that causes the tokenized set to be huge, it is more suitable for smaller documents like an email message. Instead of normalized folder name vectors and considering cosine similarities, the WordNet class can be used to measure the similarity. Besides that, the method also provides possibilities to corporate with other algorithms, such as co-training, to achieve higher accuracy, which further adds to the potential for future applications.

In the end, despite the unlabeled nature of email messages, it is desirable that pre-determined folders created from the email servers, such as inbox, set, or the “starred" folder of emails, can cooperate with the method to better analyze the email content. Email systems can also in this way categorize email based on group and individual email separately. Combined with other tools like the intelligent patronization method based on private information from users, classifying unlabeled emails has more undiscovered pragmatic powers that can make a change in future email managing systems.

References

[1] G. Mujtaba, L. Shuib, R. G. Raj, N. Majeed and M. A. Al-Garadi. Email Classification Research Trends: Review and Open Issues [J]. IEEE Access, 2017, 5, 9044-9064, 10.1109/ACCESS.2017.2702187.

[2] Shinjae Yoo, Yiming Yang, Frank Lin, Il-Chul Moon. Mining Social Networks for Personalized Email Prioritization [B]. ACM, 2009. 10.1145/1557019.1557124

[3] Hesham Altwaijry, Saeed Algarny. Bayesian-based intrusion detection system [J]. Journal of King Saud University - Computer and Information Sciences, 2012 24(1): 1-6.

[4] Klimt, B., Yang, Y. The Enron Corpus: A New Dataset for Email Classification Research. Machine Learning: ECML 2004. 3201. https://doi.org/10.1007/978-3-540-30115-8_22.

[5] Dhillon, I.S., Fan, J., Guan, Y. Efficient Clustering of Very Large Document Collections. Data Mining for Scientific and Engineering Applications. Massive Computing, 2001, 2.

[6] Ramos, Juan. Using tf-idf to determine word relevance in document queries [J]. Proceedings of the first instructional conference on machine learning, 2003, 242(1).

[7] Rathi Dinesh, Michael B. Twidale. Ditch the Smileys: Customizing a Stopword List for Email-Based Data [J]. CAIS, 2013. https://doi.org/10.29173/cais394.

[8] Ljiljana Dolamic, Jacques Savoy. When stopword lists make the difference [J]. Journal of the American Society for Information Science and Technology, 2010, 61(1): 1532-2882.

[9] Izzat Alsmadi, Ikdam Alhama. Clustering and classification of email contents [J], Journal of King Saud University - Computer and Information Sciences, 2015, (27)1, 46-57, ISSN 1319-1578.

[10] Faisal Rahutomo, Kitasuka Teruaki, Masayoshi Aritsugi. Semantic cosine similarity. ICAST, 2012, 4(1).