Clustering Emails – Enron Dataset

Adit Whorra

Ashoka University

adit.whorra_asp19@ashoka.edu.in

Abstract

Despite the evolution and growth of social networking and other communication platforms, emails continue to be the most commonly used medium for professional communication. It is estimated that an average office worker receives approximately 121 emails per day, making email clustering an important technique to efficiently communicate in the workplace. In previous attempts at clustering emails, clustering has been performed on either the TF-IDF or features extracted from the body or the header of emails like subject, date-time, bcc etc. This implementation of email clustering aims to use a combination of clustering algorithms with doc2vec (a concept that was presented in 2014 by Mikilov and in this article) as features to cluster the Enron Email Dataset which was collected and prepared by the CALO Project after the infamous Enron Scandal in October 2001.

1 Background

With the rapid growth of unstructured textual data on the internet, text mining and analysis has become one of the most important means to understand and process raw information. Clustering of text documents is a technique that helps one understand the similarity between documents within a corpus and also helps study the hierarchy within various clusters allowing an efficient organization of the documents.

1.1 Document Clustering

Traditionally, the different steps in clustering are as follows –

1. **Data pre-processing** – Given a text document, the pre-processing step commonly involves cleaning the document in order to extract the important attributes from it. Tools for pre-processing include tokenisation (breaking down the document into smaller pieces like words and symbols), noise removal (removing punctuation marks and stop words and other unnecessary attributes) and stemming/lemmatization (reducing tokens to the root form by dropping unnecessary characters, usually a suffix).

2. **Feature extraction** – Once the documents have been pre-processed and cleaned, certain exploratory features are extracted in order to identify patterns within clusters after clustering.

3. **Vector representation** – The cleaned documents are then represented in a vectorised form which becomes the input of the clustering algorithm. This vectorised form can either be a set of features extracted from the documents like average length of sentences, number of lower space characters etc. or can be the TF-IDF representation of documents.

4. **Clustering algorithm design –** An appropriate clustering algorithm is then chosen which takes the vector representations of documents as inputs and assigns a cluster label to each of the documents. The clustering algorithm is

chosen based on the number of documents, the size of the feature set and the distribution of documents in the chosen vector representation.

5. **Clustering validation and analysis –** The efficacy of the algorithm is then validated using cluster validation metrics. The clusters are then individually analysed by looking at most frequent terms and word cloud of each cluster.

2 Data

The Enron dataset was collected and prepared by the CALO Project (A Cognitive Assistant that Learns and Organizes). It contains data from about 150 users, mostly senior management of Enron, organized into folders. The corpus contains a total of about 500,000 messages. Out of the 0.5 million, 40,000 unique emails were chosen randomly due to computational limitations. Note that the algorithm was first implemented separately on 20,000 emails and then on 40,000 emails and similar clusters were observed implying the presence of a consistent cluster distributions throughout the dataset.

3 Approach

3.1 Data Pre-processing

Emails are usually very messy and may contain a lot of noise including HTML tags, links and punctuation marks. Due to this, extensive pre-processing was performed in order to clean emails before features were extracted. First, the body of the email was extracted from the email. Then, stop words were removed and the body of the email was tokenised and stemmed. This was followed by the removal of noise attributes like email IDs, punctuation marks website URLs, digits and special characters and whitespace. Certain noise words like ‘xfilename’, ‘xcc’, ‘xfrom’ were also removed.

3.2 Feature Extraction

Some of the preliminary features were extracted from the emails before the pre-processing step in order to analyse clusters later –

FEATURE	DESCRIPTION
Deleted_Email	Whether the email was deleted or not
Subject	Subject of the email
Images	Number of images attached
n_urls	Number of URLs
n_images	Number of images
n_caps	Number of capital letters
n_digits	Number of digits
n_puncts	Number of punctuations

Table 1) Extracted Features

3.3 Vector representation

Two approaches for vector representation of documents were considered. First, each document was represented in the form of its TF-IDF vector which was calculated on the pre-processed emails. However, it was observed that TF-IDF was not an accurate vector representation of emails.

Since a lot of the emails were mass-forwarded emails sent to all employees (emails in which except name of employee, everything is same), the TF-IDF representations ended up grouping same emails together instead of similar ones hence defeating the purpose of email clustering. Furthermore, it was also seen that the presence of a small amount of noise (employee names, employee IDs) even after the cleaning caused the TF-IDF to give weightage to unnecessary features that adversely affected the clustering. Therefore, given these issues with using TF-IDF, an alternate approach was considered.

A doc2vec model was trained on the pre-processed emails and then the same model was used to calculate the vector representation of each of the documents. The documents were represented such that documents with the same context were closer together in vector space. This solved both the problems discussed above. The mass-forwarded emails along with the emails that occur in similar contexts occurred together in the vector space resulting in clusters that contained similar documents that occur in the same context. The presence of noise was also taken care of as the noise attributes also occurred in certain contexts which allowed emails containing similar noise patterns also to have similar representations and hence be grouped together.

3.4 Clustering Algorithm Design

Approach 1

Since the document clusters could be of arbitrary shapes and densities, the first approach was to use DBScan on the doc2scan vector representations of the documents. DBScan works in the following way –

1. **Parameters:** The input parameters are the radius of the cluster (Eps) and minimum required points inside the cluster (MinPts).
2. **Working:** The datapoints are classified into three categories – core points, border points and noise points (See figure below)

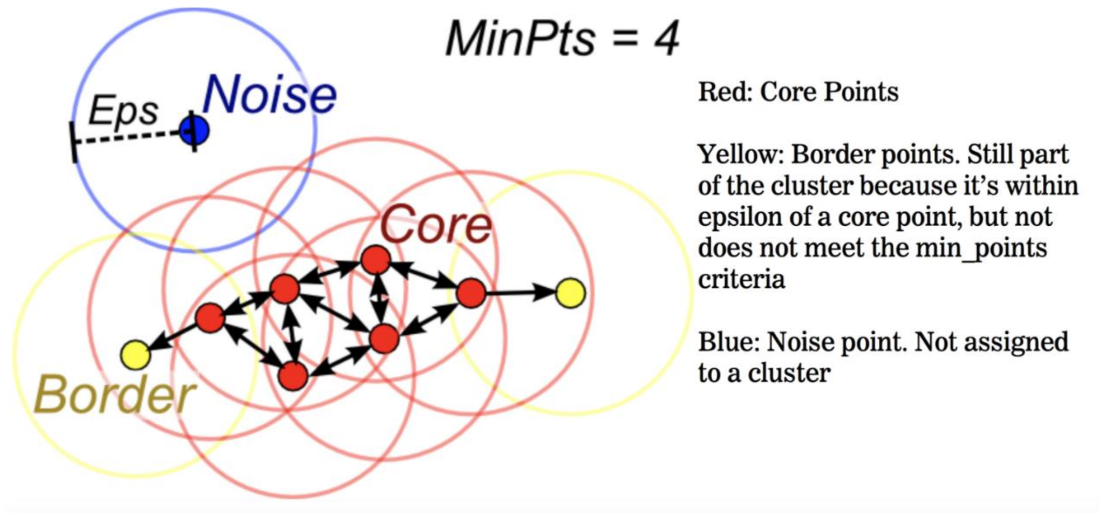
MinPts = 4
Eps
Noise
Core
Border
Red: Core Points
Yellow: Border points. Still part of the cluster because it's within epsilon of a core point, but not does not meet the min_points criteria
Blue: Noise point. Not assigned to a cluster

Figure 1) DBScan

Each core point forms a cluster together with the points that are reachable within its Eps radius. Two points are considered “directly density-reachable” if one of the points is a core point and the other point is within its Eps radius. A point which is within the Eps radius of a core point but has less than MinPts number of points that are directly-density reachable from it forms the boundary of the cluster as the cluster cannot be expanded from that point. Points that are not core points and are reachable from any other points are the noise points. In this way, DBScan is able to cluster datapoints of arbitrary shapes and distributions.

3. **Parameter Estimation:** If the Eps value chosen is too small, a large part of the data will not be clustered and will be considered noise. Whereas, if it is too large, clusters will merge together, and majority of data points will be clustered together. Therefore, Eps is chosen based on the distance of the dataset and can be estimated using a k-distance graph. MinPts, on the other hand, is usually taken as the natural log of the number of data points to be clustered (as described here).

Results

Due to the dense distribution of documents, it was observed that clustering using DBScan resulted in the formation of a few local clusters but caused most of the data points to be clustered in the same cluster (at the optimal value of the parameters).

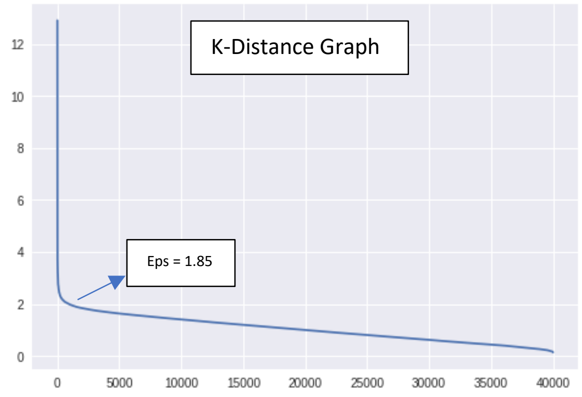

Figure 2a) DBScan K-Distance Graph

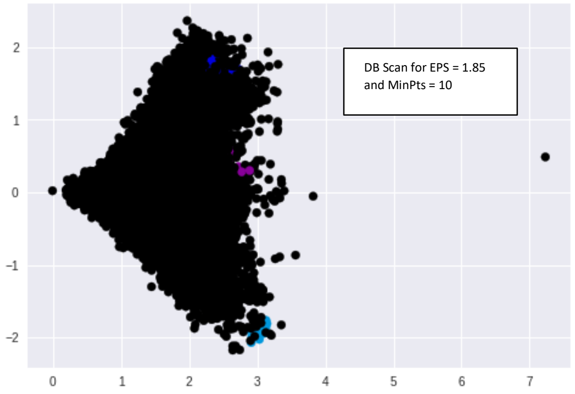

Figure 2b) DBScan Clustering Result

As seen in the above figure, some local clusters were formed but most points were clustered in the same cluster. Therefore, it was observed that DBScan would be useful to identify local clusters but not to identify broad clusters in the dataset. Also note that since the documents are represented by their doc2vec vector, similar documents should extremely close to each other and a density reachability approach might cause documents far away from each other also to be clustered together. In this case, as the distribution of documents is very dense, a similar occurrence is observed.

Approach 2

Given the density of the documents, an alternate approach was to first make broad document clusters using KMeans and then identify local sub-clusters within these clusters using DBScan. KMeans resulted in the formation of well-separated dense clusters, each containing emails of similar contexts. Further application of DBScan within the clusters resulted in the identification of local sub-clusters within clusters.

Results and Cluster Analysis

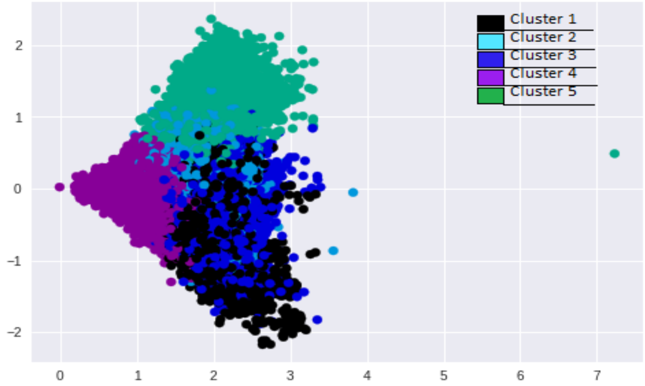

Figure 3) KMeans Clustering Result

The number of clusters was chosen to be 5 as it gave the maximum Silhouette Score for the clustering. The cluster analysis is as follows -

Cluster 1

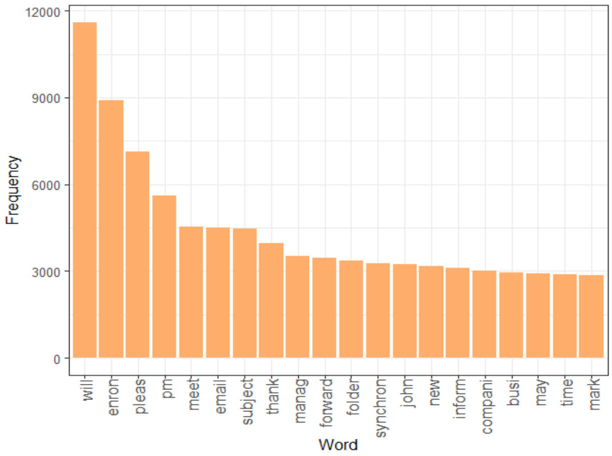

Figure 4a) Cluster 1 Word Frequency Table

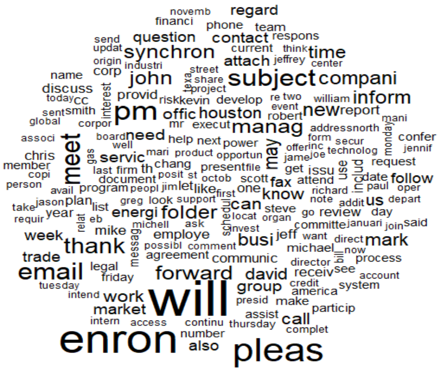

Figure 5b) Cluster 1 Word Cloud

Cluster 1 seems to contain emails exchanged between employees within the company. The most frequent words include “meet”, “pm”, “thank”, “time”, indicating conversations regarding fixing up meetings at specific times. The average words per email in this cluster is 91 which is the lowest amongst all the clusters. Furthermore, 35 percent of the emails in this cluster are replies which indicates personal conversation. The following is an example email -

> *“Subject: Private lesson on Sunday*
> *Body: Helen-Is a stand-by time available for Sunday afternoon around 2 pm? The lesson would be for my daughter Kali. Thanks, Susan Pereira ”*

Cluster 2

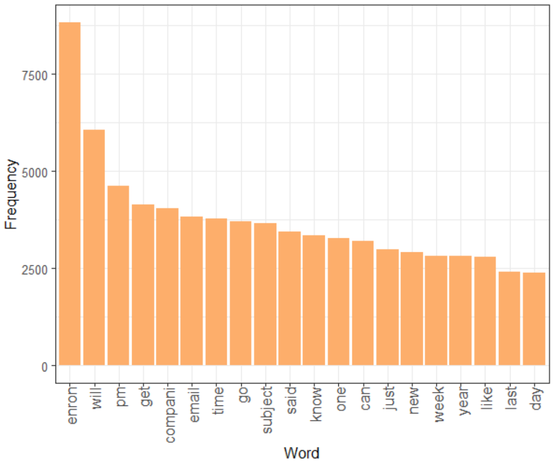

Figure 5a) Cluster 2 Word Frequency Table

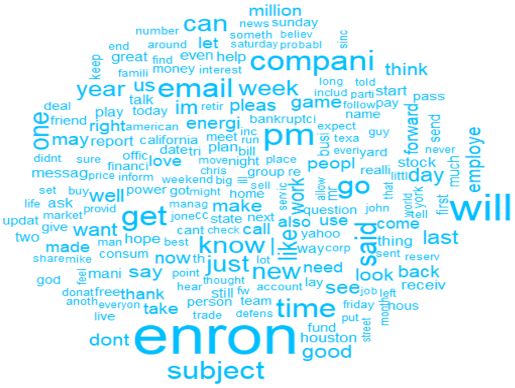

Figure 5b) Cluster 2 Word Cloud

Cluster 2 is similar to cluster 1 and also contains mostly personal conversations between employees. The average words per cluster are 340 so these emails might be longer but mostly have the same distribution of words as in cluster 1. As observed in the graph above, cluster 1 and 2 are overlapping as well.

Cluster 3

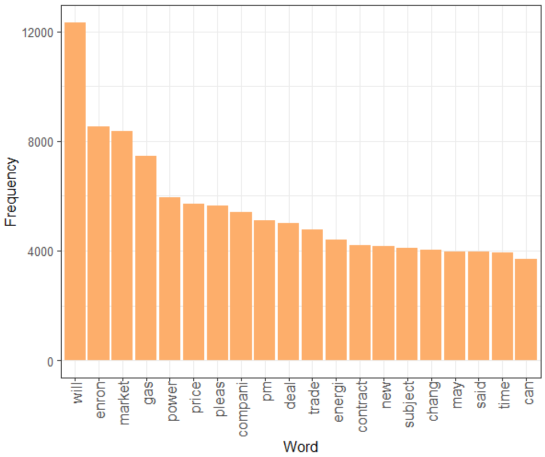

Figure 6a) Cluster 3 Word Frequency Table

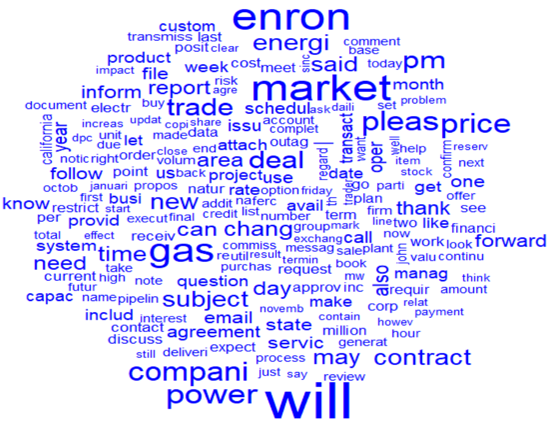

Figure 7b) Cluster 3 Word Cloud

Cluster 3 contains terms like “market”, “prices”, “trade”, “deal” and “gas” indicating that this cluster could contain emails regarding the company’s business and professional relationships with customers. Majority of them talk about the company’s finances and its trade deals with customers. The following is an example email –

“Subject: Re: equistar meter # 1552

Body: The new fixed price deal with Equistar market is #365013. We just did a deal for the rest of the month for 10,000/d at meter # 1552 QE-1 @ $4.355 .... can you let me and Robert Lloyd know what the sitara # is? Thanks"

Cluster 4

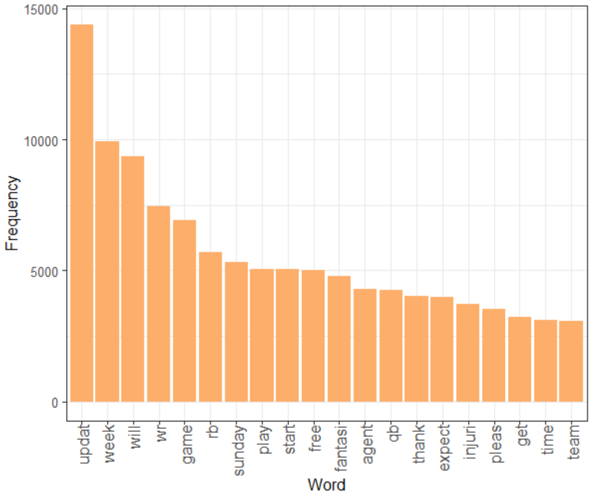

Figure 7a) Cluster 4 Word Frequency Table

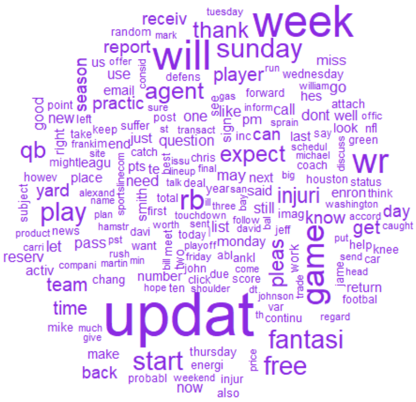

Figure 7b) Cluster 4 Word Cloud

It can be inferred from the most frequent terms in cluster 4 that it contains emails that talk about NFL Fantasy Football. It contains terms like “rb” and “qb” which are abbreviations of “Right Back” and “Quarter Back” respectively. It also contains terms like “update”, “week” and “game” which constitute very common Fantasy Football lingo. The cluster contains both personal emails exchanged amongst employees regarding Fantasy Football as well as weekly Fantasy Football reports and newsletters. The following are example emails –

“Subject: NFL.COM FANTASY FOOTBALL

Body: Fantasy Football Newsletter October 19, 2001Fantasy Sports NFL.com Welcome to another edition of the 2001 Fantasy Football Newsletter! The Fantasy Football newsletter will arrive in your e-mail inbox at the end of every week…”

“Subject: RE:

Body: Boy, we both had a crappy fantasy performance this week…." ”

Cluster 5

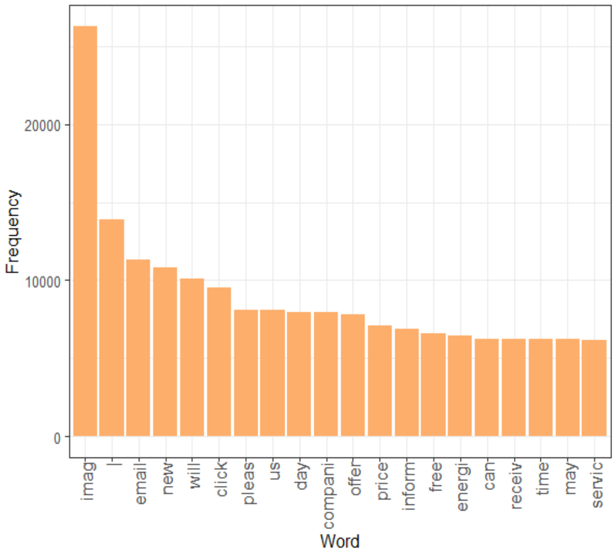

Figure 8a) Cluster 5 Word Frequency Table

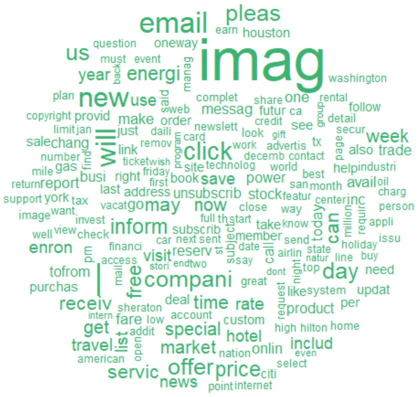

Figure 8b) Cluster 5 Word Cloud

It is evident that cluster 5 contains spam and offer emails received by the employees of the company. The most frequent words include “click”, “offer” and “free” which indicate emails containing offer deals and click baits. On an average, these emails contain 6 images and 10 URLs. The average number of capital letters is 600 and more than 60 percent of these emails were deleted by employees. All these characteristics can be attributed to the category to spam emails. Here is an example email –

> “*Subject: You Won't Believe These Rates!*
>
> *Body: [IMAGE] [IMAGE] Check out the great rates from Discover(R). See below to learn how you can get a subscription to AlumniAccess FREE. CCGBLU00001066 Discover is back in the AlumniAccess Sponsor Program! If you would like to get a free AlumniAccess membership when you qualify for a Discover card, click here to sign up through our AlumniAccess subscription registration and choose the \"Get AlumniAccess Free\" option. If you would like to stop receiving our Deal of the Week, please change your email settings on HighSchoolAlumni.com.*”

Further Analysis

Further application of DBScan within the clusters showed results similar to the initial DBScan clustering—DBScan was able to identify local sub-clusters within the clusters but mostly merged together most of the emails in one sub-cluster. This approach brought out an important trade-off between identifying local clusters and identifying broader clusters. While KMeans showed good results in the identification of broad clusters, DBScan allowed the discovery of local sub-clusters within the clusters. This process of clustering and sub-clustering can be repeated to find local and broad clusters at any granularity. Therefore, the extent of clustering depends on the granularity to which one wants to cluster.

4 Conclusion and Future Work

The analysis and clustering of the Enron Email dataset through the approach described above resulted in the formation of well-defined clusters that contain emails of similar contexts. Compared to a previous implementation of the same, the above approach has resulted in much better results. Along with a more rigorous pre-processing of the emails as compared to the previous attempt, the approach described above uses doc2vec instead of TF-IDF as input features for the clustering algorithm. Even though TF-IDF may seem like a good option, it is not able to capture entire context of the email like doc2vec does. Doc2vec improves the results substantially as the doc2vec model is able to deal with noise attributes much better than TF-IDF.

As discussed above, the extent of clustering depends on the granularity at which one wants to the clusters to be. One can continue to find sub-clusters within clusters and go deeper to find more and more compact and unambiguous clusters. However due to computational and time constraints, this aspect could not be explored further in the Enron Dataset.