UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

FIB

Master in Innovation and Research in Informatics

Data Science

Master Thesis

NLP Analysis of Email Interactions to Find Automation Opportunities

Author:

José Miguel Hernández Cabrera

Advisor:

Lluís Padró Cirera

Co-advisor:

Josep Carmona Vargas

Date of Defense: 20/10/2022

Abstract

Finding automatization opportunities for email interactions can have positive effects for several industries, especially in tasks such as reading, receiving, writing and responding emails, categorizing emails or even to prevent loss of productivity and financial loses by dealing with spam, or improve users' satisfaction; even improving automatic categorization systems can mitigate negative impacts on personal and organization performance [10]. Furthermore, people who work in companies spend around 28 % of their time reading and answering emails. In this project we proposed a methodology based on NLP and Unsupervised Machine Learning to look for opportunities of automation arising from recurrent email patterns found in email texts. We intent to facilitate the linguistic analysis in order to retrieve interaction patterns that can trigger automation actions. We proposed CRISP-DM methodology that lays the groundwork for detection of automatization opportunities in tasks relates. We compared the unsupervised machine learning methods K-Means, DBSCAN, and HDBSCAN with four clustering metrics applied to the Enron e-mails dataset transformed into paragraph vectors and performed several experiments with Word Mover's Distance, Euclidean Distance, L2-Norm and Cosine Similarity. Although our process yielded limited results in the detection of email interactions, we found that DBSCAN combined with Euclidean Distance was the best method among all scores. This project also contributes to the parameterization literature of said clustering algorithms as well as showing which methods, distances and scores settings are relevant for unsupervised email mining.

Key words: NLP, Cluster Analysis, Unsupervised Machine Learning, Enron Emails, Word Mover's Distance,

Contents

List of Figures

List of Tables

List of Algorithms

Listings

1 Introduction

Email messaging is one of the most important tools for any kind of industry. Up to 65 % of interactions of employees in several industries can be carried out through social technologies such as email messages, spending about 28 % of their working time responding, reading and writing emails, and consuming around 19 % of their working hours to track down important emails [20], or up to 10 % categorizing email messages [6]. There might also be detrimental situations around emails. Companies can lose a significant amount of productivity due to a lack of spam management [19].

Fortunately, the automation of email processing entails a range of benefits. For instance, an automatic email answering system can help companies to save labor force in email customer service [42]. Moreover, there are other automation opportunities such as the automatic expertise discovery using emails can improve organizational efficiency [9]. Designing systems to automate tasks of said nature may reduce time consumed and even improve the life quality [31]. There also particular industries where automation can be socially beneficial, such as healthcare, law enforcement, and education.

This project is justified for the reasons stated above. There are several tasks where automation opportunities can be implemented, namely spam detection, email categorization, contact analysis, email network property analysis, email visualization, and Automatic Email Answering, which are explained below. Hence, in this project we attempt to propose a system based on Natural Language Processing and using Unsupervised Machine Learning to detect patterns from features extracted from email representations.

In general, we compare a variety of density and partition based clustering algorithms over emails belonging the Enron data set[24], followed by the cluster quality evaluation using scores that do not require ground-truth assumptions and measure intra and inter cluster stability or differences. We conducted it with the possibility of finding emails interaction in mind. Our proposed system is based on the CRISP-DM methodology, were it considers in its various stages how we address clustering methods, clusters evaluation, distance matrices and text representation, in line with defined objectives below.

The following work is structured in several sections, which are stated as follows: Section 1 provides a brief description of the motivation for the project, what it intended to achieve, the relevance of the results, and a concise description of the findings. Then, Section 2 describes general objective as well as specific objectives. Section 3 defines what we understand as data mining, provides a brief literature review, and finishes with a definition of the methods used in the project. Section 4 details the methodology used. It starts with the establishment of the main pipeline based on the CRISP-DM methodology, followed by the explanation of our settings for K-Means, DBSCAN and HDBSCAN clustering methods. We also explain how each method is combined with Euclidean Distance, L2-Norm, Word Mover's Distance metrics as well as the Cosine Similarity. Next, we describe how we assess cluster quality with scores that do not require ground truth methods, namely Silhouette Scores, the Calinsky-Harabasz, Davies-Bouldin and Entropy; lastly, we disclose the parameters used for obtaining the text representations with doc2vec. Section 5 contains the results analysis, where we

briefly present the most relevant aspects of the clustering, as well as a short review of the emails. Section 6 introduces the discussion around obtained results. Here, we describe earlier notions mentioned in Section 3 and our insights. Lastly, Section 7 presents our conclusions and mentions future works. Unfortunately, our system was not able to detect email interactions. Nonetheless, we did found relevant parametrization settings. However, our contribution can be summarized in the parameterization of our algorithms and also in showing which methods, distances and scores are relevant for unsupervised email mining.

2 Objectives

Finding automatization opportunities for email interactions can have positive effects for several industries, especially in tasks such as reading, receiving, writing and responding emails [20], categorizing emails [6] or even to prevent loss of productivity and financial loses by dealing with spam [6], or improve users' satisfaction [31] or even improving automatic categorization systems to mitigate negative impacts on personal and organization performance [10]. Considering the foregoing, in this section we introduce the General Objectives as well as the specific objectives, which will be the backbone of the project.

2.1 General Objective

The general objective is to propose a system based on Natural Language Processing and Unsupervised Machine Learning to look for opportunities of automatization arising from recurrent email patterns found in email texts.

2.2 Specific Objectives

- Compare different clustering algorithms based on density and partition capabilities.
- Evaluate cluster quality using several indices that do not require ground-truth assumptions.
- Explore the feasibility of using unsupervised methods to group together email chains or detect interactions between emails.

3 Background

This Section contains the definitions of email mining and main tasks. It also includes a literature review around said field with applications on NLP, automatic texts and clustering techniques. We also briefly disclose the context of the Enron emails, a data set widely used in this field. Lastly, we define several methods, metrics and text representation used in reviewed literature and that are relevant for the project.

We start by describing the K-Means algorithm and a variant called K-Means++. Then, we proceed to define the density-based algorithm DBSCAN and how it uses density regions to create clusters and also detect noise. Next, we define the HDBSCAN, which combines hierarchical agglomerative methods with minimum spanning trees and density regions to identify clusters and noise points. In like manner, also Euclidean Distance, L2-Norm, Word Mover's Distance metrics as well as the Cosine Similarity which are essential for the performance of clustering algorithms. Likewise, we define the Silhouette Scores, the Calinsky-Harabasz, Davies-Bouldin and Entropy, of which none requires ground-truth information, i.e., true labels. Moreover, we compare them with methods that do require ground-truth data, which are the Homogeneity Score or Completeness Score. Lastly, we address some of the most popular text representations which are the term frequency-inverse document vector TF-IDF and the Paragraph-Vector, a neural network also known as doc2vec.

3.1 Email Mining

Email mining is a process that involves data mining techniques focused on email data. Tang, Pei, and Luk [42] surveyed the latest email mining techniques and identified six major tasks: spam detection, email categorization, contact analysis, email network property analysis, email visualization, and Automatic Email Answering, which are explained below.

- **Spam detection** refers to the detection of unsolicited bulk emails. It can be divided in detecting spam from email contents and detecting spam from email senders.
- **Email categorization** is the automatic organization of emails into different categories.
- **Contact analysis** is the identification of particular or group email contacts by analyzing contact's characteristics. [44] There are two categories: contact identification, which consists in finding email contacts with special characteristics; and contact categorization, which tries to assign email contacts into groups with common characteristics.
- **Email network property analysis** is the method that tries to detect critical properties of an email network in terms of network structure, relation strength and organizational structures.
- **Email visualization** uses visualization techniques to help users identity, retrieve, and summarize useful information contained in a large volume of emails.

- **Automatic Email Answering** tries to analyse incoming emails and reply them with an appropriate answer automatically. [7]

With regard to the analysis of emails, these are usually composed by a header and a body. The header usually contains a set of fields such as “From”, “To”, “CC”, “BCC”, “Subject” and “Date”. The way they are displayed depends on the email service providers. As to the body, it is made of unstructured text and may include graphic elements, URL links, markup tags, and attachments. To perform mining tasks, emails must apply data representation.

Data representation refers to the methods used to register email information such as sender-receiver information, subject and content. Emails can be represented by two approaches: Feature based approach and Social structure based approach. The Feature based approach uses the features of an email by text representations, usually in a vector space model [37] where an email is a vector and the extracted features will become its dimensions. The most common used vector representation models are TF-IDF and word/document embeddings. The social structure base approach attempts to extract a social network formed by emailing activities [42]. This social network is modelled as a graph, where nodes are a set of email addresses, and email interactions are edges.

Now, we proceed to review literature regarding these email mining tasks.

3.2 Literature Review

Before discussing the literature regarding email mining, we will describe briefly two important concepts for Machine Learning: Supervised Machine Learning and Unsupervised Machine Learning. The former consists of methods that process labeled data, or ground truth classes, in order to predict or classify outcomes accurately. Conversely, the latter refers to the methods where data does not have labeled class and they try to discover patterns or data groupings. In general, unsupervised machine learning algorithms are not well suited for task-oriented applications such as email classification, since they do not have the necessary labels to classify them. However, some unsupervised methods such as clustering can be used to group emails by topic, which may be helpful for organization or filtering purposes.

Regarding Email network property analysis, Agarwal et al. [1] presented one of the most important works in the field of Email Mining. It aims to predict the hierarchy of Enron employees, establishing their Enron Hierarchy Gold Standard with 158 employees and describing the immediate dominance relationships between managers and subordinates. This is relevant for email interactions since they performed experiments to find interactions between employee dominance relationships through Social Network Analysis (SNA) and NLP. They considered as the golden standard a perfect NLP system that correctly predicts an upper bound of the best performing predictions. This NLP system would require communication links between people to predict their dominance relationship. However, they preferred to compare an SNA system using an undirected weighted graph, for which they built 407.095 weighted links and 93.421 nodes.

Meanwhile Diesner, Frantz, and Carley [13] focused on the email interactions around organizational crisis. They explored the dynamics of the structure and properties of the organizational communication network, as well as the characteristics and patterns of communicative behaviour of the employees from different organizational levels. They found that during the crisis period, communication among employees became more diverse in comparison with established formal roles. Indeed, the company was not going through its best times. In December of 2000 Skilling took over the position of CEO from Lay, who then became COO. In August 2001 Skilling stepped-down, and Lay became CEO once again. Afterwards, in December 2001 Enron filed for bankruptcy. Eventually the company split into several firms, being one of them an IT focused consultancy firm named Accenture. These movements reflected the crisis that the company was facing at the time due to fraud investigations.

To achieve that, they used directed network graphs enhancing data with address-assignment ratio (i.e., the number of valid email addresses matched to an individual) to link people and not only email addresses. They also used several files to integrate with the original Enron data. To match spelling and names, they used Lehmann's Similarity algorithm. However, they found that the pattern of evolution in density networks rose as the scandal/investigation escalated when viewed by hierarchical position. Related to that, they also reported that in case of hierarchical communication, the higher rank performed more downward communication, suggesting their directive roles. In concordance, lower ranks sent out more upward communication, apparently reporting to manager-level orders.

Nonetheless, one of the disadvantages of the Enron dataset is that it does not reflect the relationships of all employees, because the sampled emails structure does not show the interactions between the 21.000 employees within their own internal network as well as external networks. However, according to Diesner, Frantz, and Carley [13], it is possible to detect organizational communication changes in both volume and nodes of communication within periods of change and crisis.

In line with communication findings, word use is correlated to the role within the organization, as suggested from Keila and Skillicorn [23]. They discovered that employees reflected in the emails their positions and relationships, as well as the changes withing the company through word usage patterns. They used a SVD matrix for terms, with rows corresponding to emails and columns to word frequency rank. No stemming was applied to their analysis. Additionally, they used a Semi-Discrete Decomposition (SDD) matrix to generate an unsupervised hierarchical classification of the emails. They concluded that there is a strong differentiation between short messages using rare words, and long messages using more typical words. Furthermore, there seems to be an effect of company role on word use patterns, as individuals with similar status and role inside the organization communicated with similar words. Also, emphasising certain words tended to bring together individuals who did not belong to the organizational environment (non-employees).

Another work related to unsupervised learning around the Enron dataset was that of Kathuria, Mukhopadhyay, and Thakur [21], in which they proposed a cohesion score to each cluster to evaluate the intra-cluster quality. Their approach used k-means and hierarchical clustering with TF-IDF vectors. The pre-processing of the email bodies was done by removing stop

words, then lemmatizing each word, and after that converting the resulting tokens into the TF-IDF. Using the elbow method, their data suggested that the optimal number of clusters is three. Moreover, the cohesion score consisted on the cosine similarity of each term for each of these three clusters. They managed to find which was the one with higher cohesion, but did not provided any insights on the content of these clusters. In fact, they mentioned that it was up to the researcher to gain insights from the clusters content. However their work is relevant for our research since we will also use the cohesion score with the applied methods.

Hermans and Murphy-Hill [17] explored the usage of Excel spreadsheets in the emails. They found that only about 10% of the emails had attached spreadsheets to share information.

Regarding the clustering evaluation without a ground truth, Wang et al. [47] used three scores to evaluate their clusters. Their intended work was to profile tourists through publishing tourist endpoints in Constance, Germany. In order to find clusters, they combined association rules along with DBSCAN and NK-means [18], which is a K-means version performing noise removal automatically that runs in $O(n^2d) + T(n)$. The goal of this algorithm is to detect outliers while calculating the k means of the dataset. Since analysing tourist intentions of traveling do not offer a readily ground truth to evaluating the patterns shown from data, most of the methods for validation in clustering data are not valid. In this sense, they used three scores to evaluate the cluster quality: Silhouettes score [35], Calinksi-Harabasz score [8] and Davies-Bouldin Score [11], which definitions are addressed in Section 4. They analysed 9.6 GB of text data and 146.6 GB of media files with the intent to find which where the Point of Interest that most tourists visited.

3.3 Enron Emails Dataset

The Enron dataset was collected and prepared by the CALO Project, and presented by Klimt and Yang [24]. It contains a set of email messages made public by the U.S. Federal Energy Regulatory Commission after fraud investigation that led to the the eventual demise of Enron Corporation. The first version contained 619.446 messages allocated in 158 user folders, most of them belonging to senior management of Enron, but over the years said number was modified by either user removal requests or cleaning. The current version contains 562.000 messages. For context, in December of 2000 Skilling took over the position of CEO from Lay, who then became COO. In August 2001 Skilling stepped-down, and Lay became CEO once again. Afterwards, in December 2001 Enron filed for bankruptcy. Eventually the company split into several firms, being one of them an IT focused consultancy firm named Accenture. This is why the majority of emails are from this period of time and it has become one of the most widely used datasets for text analysis, fraud detection and email mining.

By suggestion of Klimt and Yang [24], it has become a common practice to ignore certain folders such as "*discussion_threads*" since it seemed to be computer generated and not used directly by users, and did not provided clear information on extract actual email threads between users. Another ignored folder is the "*all_documents*" since it contains a large number of duplicate emails already present in other users' folders.

Since the objective of this work is to find automation opportunities withing email interactions, a crucial part is to find all email chains. This challenge has been addressed by others in diverse literature. Agarwal et al. [1] showed that it is difficult to get a proper structure since the emails do not provide the metadata necessary to create a proper network.

Early works tried to construct its hierarchy through the work done by Shetty and Adibi regarding the Enron ex-employees list [41].

Agarwal et al. [1] presented a gold standard that contains 158 employees, and 13.724 dominance pairs (pairs of employees where the first dominates the second in the hierarchy, not necessarily immediately). All of the employees in the hierarchy are email correspondents on the Enron email database, though obviously many are not from the core group of about 158 employees for which we have the complete inbox. The hierarchy is linked to a threaded representation of the Enron corpus using shared IDs for the employees who participated in the email conversation. The resource is available as a MongoDB database. However, they focused more on predicting Enron hierarchy rather than on interactions between emails.

3.4 Clustering Methods

There are three main types of several clustering techniques: partitioning, hierarchical and density algorithms [22]. Partition algorithms create a partition from a dataset D of n data points into a set of k clusters. These algorithms have the characteristic that k is a parameter, which in several cases is a disadvantage since sometimes it requires domain knowledge so as to get a "good" cluster. To this category also belongs K-Means, in which each cluster is represented by the gravity center of the cluster.

Hierarchical algorithms create a hierarchical decomposition of dataset D. It can be represented as a tree that iteratively splits D into smaller chunks until each chunk is of unit length. This split can be either agglomerative (bottom-up) or divisive (top-bottom). It does not require a k parameter but does need a termination condition, or *cut* to terminate the merging/division process. Here lies the disadvantage of these algorithms since it is difficult to arrive to an appropriate termination condition.

The third category addresses density to identify clusters in k-dimensional point sets. Some of the main methods are DBSCAN, HDBSCAN and OPTICS. In this project we will focus on the first two.

3.4.1 KMeans

The K-Means algorithm is a partition based clustering method widely used in clustering techniques with wide range of applications in several fields. It generates groups based on the similarity between the data points taking as a reference an Euclidean distance. It computes a set of prototypes $\mathcal{P} = \{\mu_1, \mu_2, \ldots, \mu_k\}$ representing k clusters. It performs hard partitioning,

which is the allocation of one element to one cluster. The fitted model is a set of k hyperspherical clusters where borders meet with other spheres, hence the name. The algorithm works by first randomly selecting k points from the data set (called centroids) and then assigning each data point to the cluster that has the closest centroid. The centroids are then updated to be the mean of the points in their cluster and the process is repeated until the centroids no longer change.

Formally, for each vector, the distance between data vector and each cluster is calculated:

$$d(Z_p, M_j) = \sqrt{\sum (Z_p, K_y - M_j, K_y)}$$

where Z_p is the p-th data point, M_j is the centroid of the j-th cluster.

The centroid is recalculated as follows:

$$M_j = \frac{1}{N_j} \sum Z_p, \bigtriangledown Z_p \in C_j$$

where N_j is the number of data points in cluster j.

The optimization criteria is based on minimizing the distance of each centroid of the cluster, and it performs a local search over the following distortion function.

$$Distortion = \sum_{k=1}^{K} \sum_{i \in C_k} \|x_i - \mu_k\|^2$$

The algorithm converges to a local minima, therefore the clustering depends on the initialization.

Algorithm 1: K-Means algorithm

Data: X: vector, k: integer
Result: Set of k clusters
Generate k initial prototypes ;
while *Clusters assignments changes or have not reached a number of iterations* **do**
    Reassign the examples to their nearest prototype;
    Recalculate prototypes (centroids) ;
end

This algorithm is the most used since its convergence has a low computational complexity of $O(kni)$ and it is easy to implement. However, the results are highly dependent on initialization, and it must be run several times and the k number must be selected according to the best results. Furthermore, the presence of outliers or clusters with different sizes and densities affect the results. Lastly, in some cases the hard partition might be too constrained.

K-Means++ is a modification where the initialization strategy looks to maximize the distance among the initial prototypes. This is the default algorithm for the *scikit-learn* and RAPIDS AI implementations. Its algorithm is described below:

The average complexity is given by $O(knT)$, where n is the number of samples and T is the number of iteration. The worst case complexity is given by $O(n^{\frac{k+2}{p}})$ with $n = \texttt{n\_samples}$, $p =$

Algorithm 2: K-Means++ algorithm

Data: X: vector, k: integer
Result: Set of k clusters
Choose one center uniformly among all data ;
while k *not chosen* **do**
  foreach x **do**
    compute $d(x, c)$; `/* The distance between` x `and the nearest center already chosen. */`
  end
  Using a weighted probability distribution, choose one new data point at random as a new center, where a point x is chosen with probability proportional to $d(x, c)^2$;
end
Run standard K-Means;

`n_features`. In fact, finding the optimal solution is NP-hard in an Euclidean Space and for a general number of cluster k [4] [2].

3.4.2 DBSCAN

The Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a density-based clustering algorithm proposed by Ester et al. [15]. It looks for points that are close together and groups them into clusters of points within an Epsilon-neighbourhood and points that are directly density-reachable, density-reachable, and density-connected, whereas the points that are not inside said clusters are considered noise. This is an important distinction (and advantage) over the K-Means algorithms.

Formally, let $N_\epsilon(p)$ be the epsilon-neighborhood of a point p and is defined by

$$N_\epsilon(p) = \{q \in D | d(p, q) \leq \epsilon\}$$

where D is a point dataset, $d(p, q)$ the distance function for two points, p and q. There are two types of points: those that are inside the cluster (*core points*), and those that are at the borders of the cluster (*border points*). For this to work, we require to know the value of minimum number of points (*MinPts*) that belong to that epsilon-neighborhood. In other words, for every point p in a cluster C there is a point q in C so that p is inside of the Epsilon-neighborhood of q and $N_\epsilon(p)$ contains at least *MinPts* points. This requirement takes us to the definitions of density-reachability and density-connectivity.

First of all, for a point to be a core point it must be directly density-reachable. A point p is directly density-reachable from a point q w.r.t. ϵ, *MinPts* if

1. $p \in N_\epsilon(q)$
2. $|N_\epsilon(q)| \geq$ *MinPts*

This property is symmetric for pairs of core points, but not necessarily between this kind and border points.

A point p is density-reachable from a point q w.r.t. ϵ, *MinPts* if there is a chain of points $p_1, \ldots, p_n$, where $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density-reachable from p_i. In this case, it is possible that two border points of the same cluster C might not be density-reachable from each other because the core condition does not hold for them, but there must be a core point in C for each of those points to be border. This relation is defined by the density-connectivity.

A point p is density-connected to a point q w.r.t. ϵ, *MinPts* if there is a point o such that both, p and q are density-reachable from o w.r.t. ϵ, *MinPts*.

Once we defined the density-reachability and density-connectivity, we can now define the clusters. A cluster C w.r.t. ϵ, *MinPts* is a non-empty subset of D that satisfies two conditions:

- Maximality condition: $\forall\ p, q$: if $p \in C$ and q is density-reachable from p w.r.t. ϵ, *MinPts* $\Rightarrow q \in C$
- Connectivity condition: $\forall\ p, q \in C$: p is density-connected to q w.r.t. ϵ, *MinPts*.

Lastly, to define noise we consider the points where the previous conditions are not met. Let $C_1, \ldots, C_k$ be the clusters of data set D w.r.t. ϵ, *MinPts*$_i$ $i = 1, \ldots, k$. Then, the noise is the set of points in data set D that does not belong to any cluster C_i, expressed as

$$\text{noise} = \{p \in D | \ \forall\ i : p \notin C_i\}$$

Therefore, DBSCAN will require to look for the appropriate parameters of ϵ and *MinPts* to retrieve all points that are density-reachable.

The basic pseudocode for the DBSCAN is defined in the Algorithm 3.

Algorithm 3: DBSCAN algorithm

```
Data: D: unclassified set of points, Eps: float, MinPts: integer
Result: Set of C_k clusters
ClusterId := nextId(ClusterID) ;
foreach i in D do
    Point := D.get(i) ;
    if Point.clusterid = Unclassified then
        if ExpandCluster(D, Eps, MinPts then
            ClusterID := nextId(ClusterID) ;
        end
    end
end
```

The function that makes the algorithm work is `ExpandCluster`, the pseucode of which is described in Algorithm 4. Since the Epsilon-neighborhood is expected to be much smaller than

Algorithm 4: ExpandCluster pseudocode

```
Data: D: unclassified set of points, Point: float, ClusterId: integer, Eps: float,
      MinPts: integer
Result: Boolean
seeds := D.regionQuery(Point, Eps) ;
if seeds.size < MinPts then
    D.changeClusterID(Point, Noise) ;
    return FALSE
else
    D.changeClusterIDs(seeds,ClusterID) ;          /* All points in seeds are
     density-reachable from Point */
    seeds.delete(Point) ;
    while seeds not empty do
        currentP := seeds.first() ;
        result := D.regionQuery(currentP, Eps) ;
        if result.size ≥ MinPts then
            foreach result do
                resultP := result ;
                if resultP.clusterID in (Unclassified, Noise) then
                    seeds.append(resultP) ;
                end
                D.changeClusterID(resultP, ClusterID) ;
            end
        end
        seeds.delete(currentP) ;
    end
    return True
end
```

the size of the whole data space, and since it iterates for each of the n points of the dataset, we have at most one region query. Therefore, the average run time complexity is $O(n \log n)$.

Nonetheless, it is recommended to start DBSCAN hyperparameter tuning with low ϵ. However, this presents a disadvantage as border points may be inconsistent in high dimensionality datasets [39].

Alternatively, Ester et al. [15] offer a heuristic for hyper-tuning DBSCAN without having to iteratively find the value of ϵ, which consists in finding the optimal value of epsilon by calculating the smallest slope value from Nearest Neighbors of a dataset [12]. This is done by obtaining the Nearest Neighbors, then sorting them in ascending order and then finding the value in which the change rate is minimum. The resulting number is the optimal value of epsilon, as shown in Figure 2. However, although it helps to find the best value of the Epsilon-neighborhood, now the user must look for the optimal size of neighbors along with the *MinPts*. Nonetheless, it is less expensive (Nearest Neighbors possess a time complexity of $O(knd)$).

3.4.3 HDBSCAN

HDBSCAN is a clustering algorithm that stands for Hierarchical Density-Based Spatial Clustering of Applications with Noise. Similar to DBSCAN, it is a density-based clustering algorithm, which means it looks for areas of high density when forming clusters. This makes it more robust to outliers than other clustering algorithms, such as k-means. Moreover, this method uses a minimum spanning tree over a mutual reachability distance to find the best clusters. For instance, DBSCAN works only over a pure distance matrix. In its most basic form, HDBSCAN solves the problem of the border points, not needing to find values of epsilon but to focus on minimum points. However, it is possible to still tune for ϵ if necessary.

Another parameter that can be tuned is the `min_samples`, which is the minimum number of points that a cluster must have for it to be considered dense enough to be split into two clusters.

The steps in which HDBSCAN works can be described in five stages:

1. Transform space based on density.
2. Generate a minimum spanning tree of the distance weighted undirected graph.
3. Use hierarchical clustering on the connected components.
4. Condense the resulting hierarchy according to the minimum cluster size.
5. From the condensed tree return the stable clusters.

Along with the explanations of said steps, we will use a toy dataset to illustrate each step (Figure 3.A).

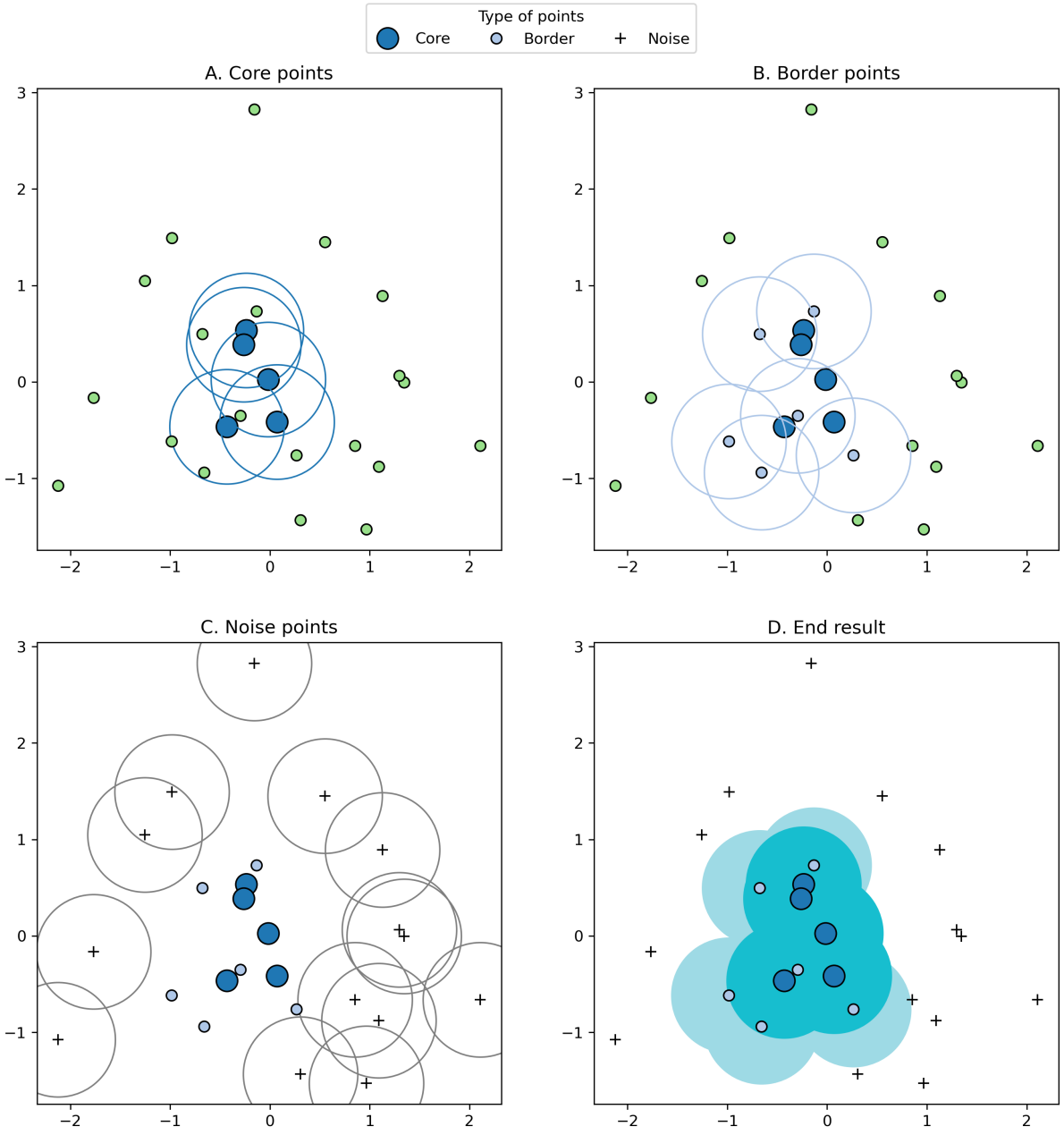


Figure 1: DBSCAN Steps - Plot A shows the core points which have at least the minimum points required inside the radius determined by the epsilon parameter; Plot B shows the border points, which are the ones that do not fulfill the minimum points requirement but are within the radius of at least one core point; Plot C shows the noise points, which are the ones that do not have enough neighbours within the epsilon radius nor are close to a core point; Plot D shows in circles all points that are within an Epsilon-neighborhood, and in crosses those that are outside the Epsilon-neighborhood. The shadows represent the final Epsilon-Neighborhood, where the darker belong to the core points and the lighter to the border points.

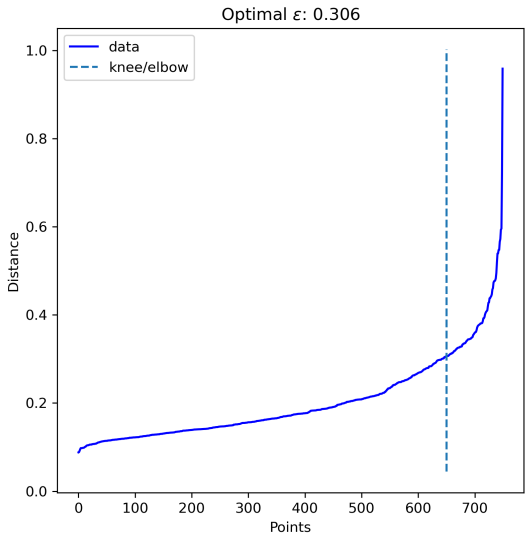


Figure 2: Heuristics for ϵ by finding the elbow point

In the first step, the principle is similar to the DBSCAN algorithm, the steps of which for determining core, border and noise points are shown in Figure 1. Furthermore, to spread points with low density, we need to define new metrics between points, a method that is called mutual reachability [14], and which is defined as

$$d_{\text{mreach}-k}(a-b) = \max\{\text{core}_k(a), \text{core}_k(b), d(a,b)\}$$

where $d(a,b)$ is the metric of choice. HDBSCAN will work with single linkage clustering to closely approximate the hierarchy levels of true density distributions of sampled points.

The following step is to consider the resulting data as an undirected graph with points as nodes and the mutual reachability distance of those points as a weighted edge. From this graph, the algorithm calculates the minimum spanning tree. Usually, Prim's algorithm and the Dual Tree Boruvka are some of the most used algorithms for this purpose. In our example, the resulting minimum spanning tree is shown in Figure 3.B.

Once the minimum spanning tree is obtained, HDBSCAN transforms it into a hierarchy of connected components, by single linkage, i.e., sort the edges of the tree by distance in ascending order and then merge them iteratively to form new clusters until there is only one cluster. Figure 3.C. illustrates the resulting robust single linkage hierarchy clustered tree.

After that, it takes the resulting hierarchy to condense the cluster tree based on the *minimum cluster size*. With this parameter, the algorithm traverses the hierarchy and at each split checks if a new created cluster has fewer points than the minimum cluster size. If it indeed possesses a smaller length, then it declares 'fall out points of a cluster' and the larger cluster retains the parent identity. Conversely, if at the split there are two clusters at least as large as the minimum cluster size, the algorithm considers the split as true and we obtain a smaller tree.

Lastly, to find the most stable clusters, HDBSCAN considers the stability as

$$\sum_{p \in \text{cluster}} (\lambda_p - \lambda_{\text{birth}})$$

where $\lambda = \frac{1}{\text{distance}}$; λ_{birth} is the value where the cluster split off and became a new cluster; λ_p is any lambda value at which the point p "fell out" of the cluster and can be between λ_{birth} and λ_{death}; the latter is the lambda value where the cluster split into smaller clusters. Figure 3.D shows in circles the fall-out points.

To obtain the stable clusters, HDBSCAN performs the following steps: it starts by declaring all leaf nodes as selected clusters. Then it calculates the stabilities of clusters. When the sum of the stabilities of the child clusters is greater than the stability of the cluster, then the cluster stability is the sum of the stabilities of the children. Alternatively, if the stability of the cluster is greater than the sum of its children, the cluster is declared to be a selected cluster and all its descendants are deselected. Lastly, when reaching the root node, the current set of selected clusters is returned as the flat cluster. Any point not selected in any cluster is considered as a noise point. The resulting clustering is shown in Figure 3.E.

Once we covered the three clustering algorithms, we can see how they work using a random generated dataset, the results of which are shown in Figure 4. Plot 4.A shows how K-Means performs hard partitioning over every point to allocate it inside a clusters, as it does not have the concept of noise. 4.B instead shows how DBSCAN defines their $N_{epsilon}(p)$ and allocates the corresponding core and border points inside it, while leaving as noise the points that are neither density-reachable nor density-connected. Lastly, although 4.C finds similar shapes as DBSCAN, the logic behind combines density points along with minimum spanning trees algorithms and hierarchical clustering, obtaining at the end stable clusters.

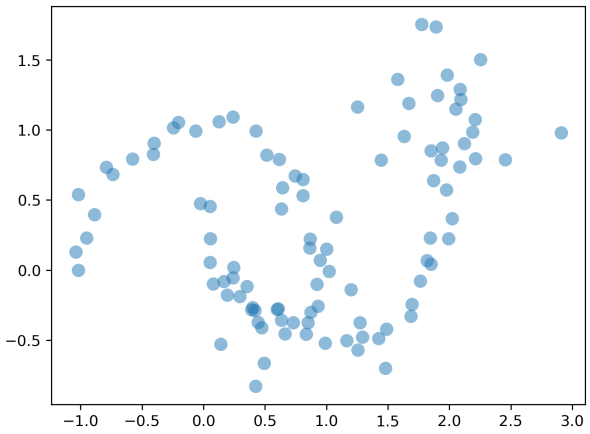
3.5 Cluster Evaluation

When making a reference to the subject of emails, it is important to note that we do not have a ground-truth over email grouping or even the shape of the embedded space. Thus, we use the following four methods to assess the quality of the obtained clusters: Silhouette Coefficient, Calinski-Harabasz Score, Davies-Bouldin Score and the Entropy. We discuss them in detail below.

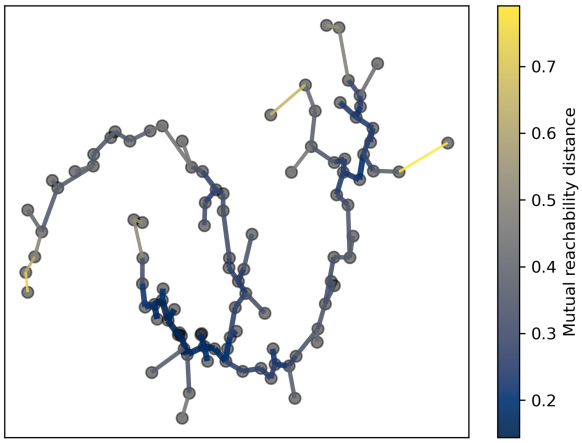
3.5.1 Silhouette Score

Proposed by Rousseeuw [35], the Silhouette Score is a technique that measures tightness and separation. The silhouettes are constructed to seek compact and clearly separated clusters. This techniques the calculated classes and a distance matrix. In the case of dissimilarities, an object i, there will be a certain value $s(i)$ and then the resulting values of i are drawn in a tile-like plot, as shown in Figure 5.

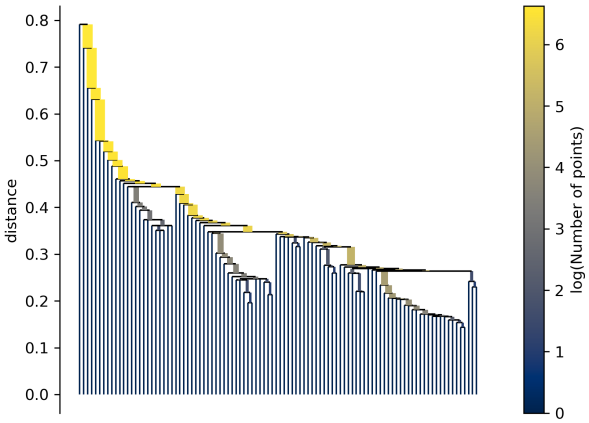
A. Initial data set.



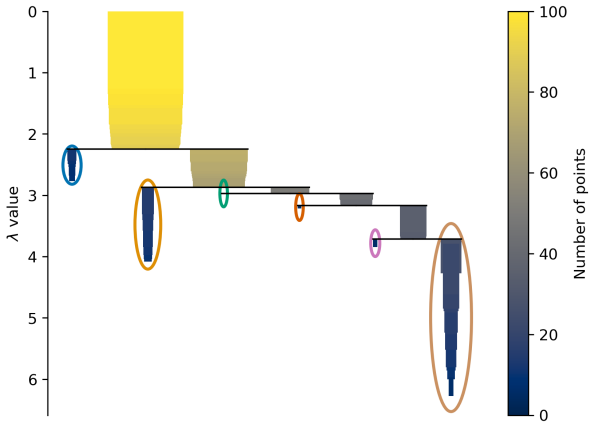
B. Minimum Spanning Tree.



C. Robust Single Linkage Hierarchy clustered tree.



D. Condensed Tree to find clusters' stability.



E. Final dataset with most stable clusters.

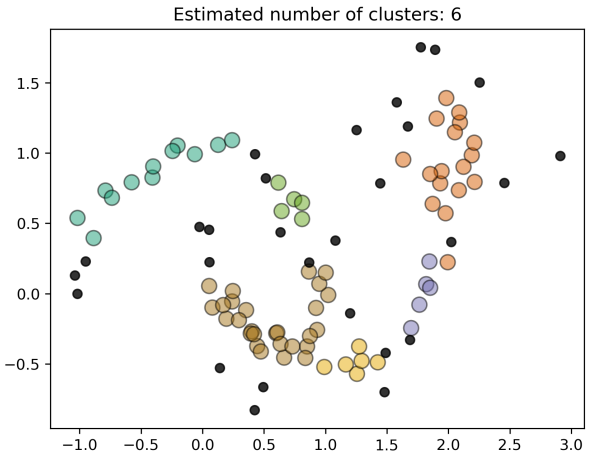


Figure 3: HDBSCAN Steps

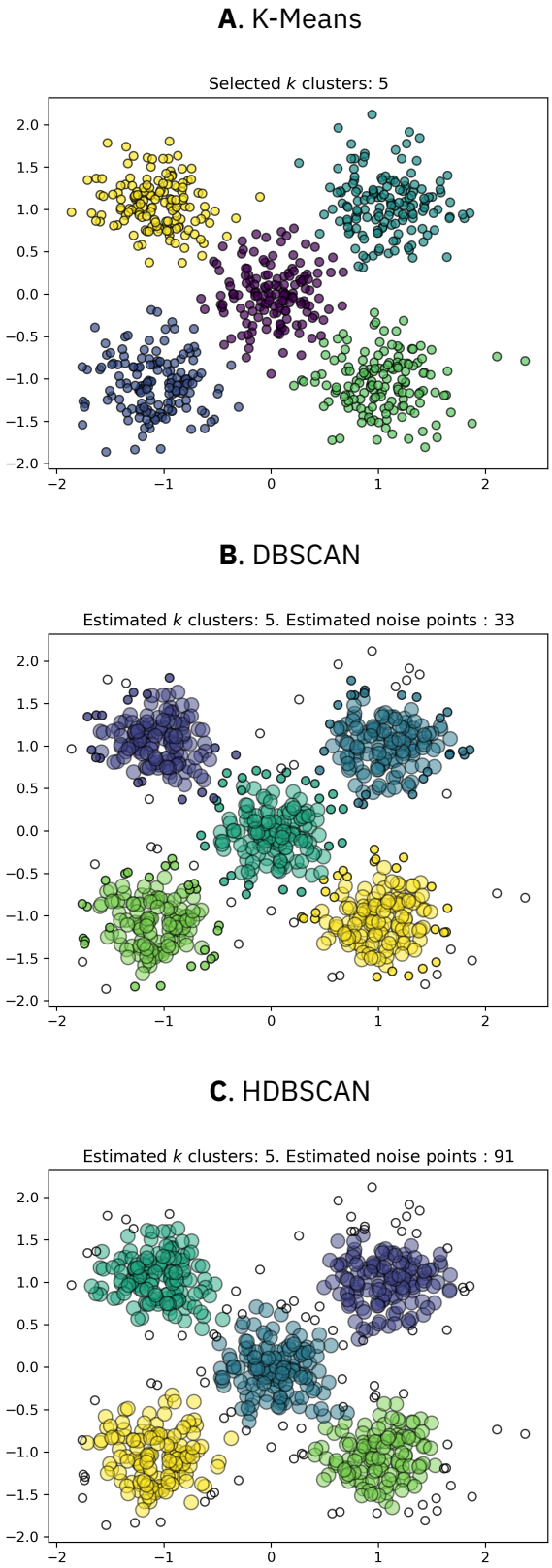
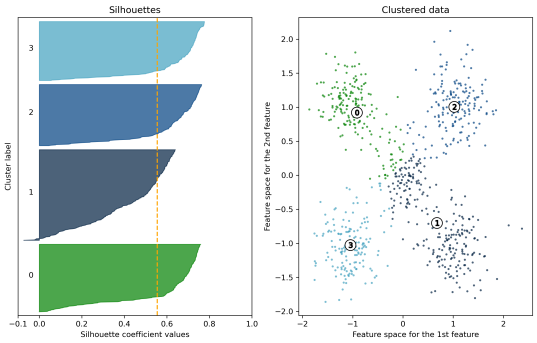
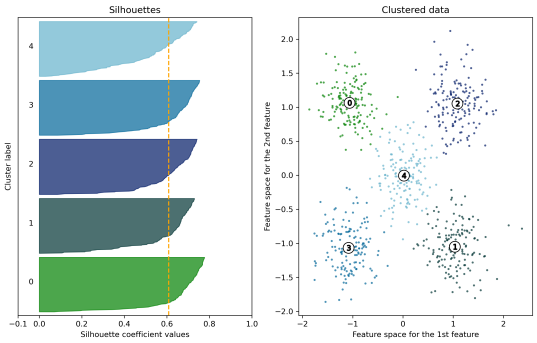


Figure 4: Comparison of clustering algorithms over a random dataset.

A. $k = 4$



B. Best Choice, $k = 5$



C. $k = 6$

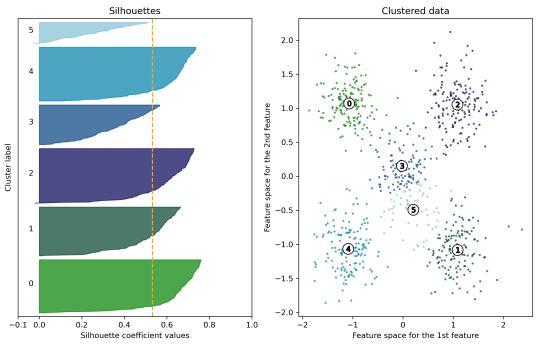


Figure 5: Silhouette analysis for K-Means.

Let i be any object in a dataset and A the cluster to which that point has being assigned. For every object different from i that is contained in cluster A, then compute

$$a(i) = \text{average dissimilarity of } i \text{ to all other objects of } A$$

and suppose there is a cluster B different from A and compute a distance

$$d(i, B) = \text{average dissimilarity of } i \text{ to all objects of } B$$

After computing $d(i, B)\ \forall C \neq A$, we select the smallest of those numbers and denote it by

$$b(i) = \min_{B \neq A} d(i, B)$$

then, $s(i)$ will be obtained as follows:

$$s(i) = \begin{cases} \frac{1-a(i)}{b(i)} & \text{if } a(i) < b(i) \\ 0 & \text{if } a(i) = b(i) \\ \frac{b(i)}{a(i)-1} & \text{if } a(i) > b(i) \end{cases}$$

or expressed as

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

In case of similarities, let $a'(i)$ and $d'(i, B)$ be the corresponding average similarities and define $b'(i)$ as

$$b(i) = \max_{B \neq A} d'(i, B)$$

Then, the corresponding values of $s(i)$ are obtained as

$$s(i) = \begin{cases} \frac{1-b'(i)}{a'(i)} & \text{if } a'(i) > b'(i) \\ 0 & \text{if } a'(i) = b'(i) \\ \frac{a'(i)}{b'(i)-1} & \text{if } a'(i) < b'(i) \end{cases}$$

In both cases, the values of $s(i)$ will be in $[-1, 1]$ for each object i. Note that when there are only two clusters, the shift from cluster 1 to cluster 2 will be $s(i)$ to $-s(i)$.

In order to interpret the dissimilarity case, one has to consider three cases of $s(i)$ values: the first one is when $s(1)$ is close to 1, it implies that dissimilarity $a(i)$ is much smaller than the smallest "between" dissimilarity $b(i)$. So, it is safe to say that i is clustered "properly". In other words, the second best-choice (another cluster denoted by C) is not nearly as close as the actual choice (A).

The second case is when $s(i)$ is close to zero. It is also known as the "intermediate case". This could mean that i could be allocated to either A or B.

The last case, when $s(i)$ is close to -1, is considered as the worst case scenario. This means that dissimilarity $a(i)$ is much larger than $b(i)$, therefore i is on average much closer to B but was allocated in A, so it means that object i was not allocated in the correct cluster. Lastly, the final score is the overall values of every silhouette value over all embeddings. That being said, hereon the Silhouette Score will be denoted as SL.

3.5.2 Calinksi-Harabasz Score

Another unsupervised method used to evaluate clusters is the Calinsky-Harabasz Score [8]. It was first defined as a method to identify clusters of points in an Euclidean Space, although it can be used with other metrics. The authors also define the method as a "variance ratio criterion" or the dendrite method. It is a measure of how well a clustering algorithm has performed. It is calculated by taking the ratio of the between-cluster dispersion to the within-cluster dispersion. The between-cluster dispersion is the sum of the squared distances between each point and the centroid of its cluster. The within-cluster dispersion is the sum of the squared distances between each point and the centroid of the entire dataset. The Calinski-Harabasz score is therefore a measure of how well clustered the data is (the lower the better) and how well separated the clusters are (the higher the better).

Between group sim of squares is defined as

$$\text{BGSS} = \sum_{k=1}^{K} n_k \|C_k - C\|^2$$

where K is the number of clusters, n_k is the number of observations in cluster k, C_k is the centroid of cluster k, and C is the baricenter of the dataset.

The intra-cluster dispersion, or the within group sum of squares (WGSS) is expressed as

$$\text{WGSS}_k = \sum_{i=1}^{n_k} \|X_{ik} - C_k\|^2$$

where X_{ik} is the i-th observation of cluster k. Then, the sum of the individual within group sums of squares for each cluster k is:

$$WGSS = \sum_{k=1}^{K} \text{WGSS}_k$$

Finally, to calculate the CH score, we calculate the sum of inter-cluster dispersion and the sum of the intra-cluster dispersion for all clusters. Let $\bar{d}^2$ denote the general mean of all $\frac{n(n-1)}{2}$ squared distance d_{ij}^2 and $\bar{d}_g^2$ that of the $\frac{n_g(n_g-1)}{2}$ squared distances within the g-th group ($g = 1, 2, \ldots, k$), then from WGSS, and BGSS, we obtain

$$CHS = \frac{\text{BGSS}/K - 1}{\text{WGSS}/N - K} = \frac{\bar{d}^2 \frac{n-k}{k-1} A_k}{\bar{d}^2 - A_k}$$

where A_k is a weighted mean of the differences between the general and the within-group mean squared distances.

To choose the best number of clusters, Caliński and Harabasz [8, p. 12] suggest that it should be the k number for which the CHS is an absolute or local maximum, or at least the one that presents a rapid increase. Hereon, the Calinski-Harabasz Score will be identified with CH.

3.5.3 Davies-Bouldin Score

The Davies and Bouldin [11] score (or index) measures the cluster separation based on the distance between vectors and the dispersion of the obtained clusters. It offers the advantage of being computationally feasible for large data sets, requires little to no user interaction and is applicable to hierarchical data sets. It does not depend on a particular clustering algorithm and it is used to validate data partitions regardless of how they where obtained. The index is built on a vector of clusters and a metric that holds the following properties:

- $d(X_i, X_j) \geq 0 \quad \forall\, X_i, X_j \in E_p$
- $d(X_i, X_j) = 0 \quad \iff \quad X_i = X_j$
- $d(X_i, X_j) = d(X_j, X_i) \quad \forall\, X_i, X_j \in E_p$
- $d(X_i, X_j) \leq d(X_i, X_k) + d(X_k, X_j) \quad \forall\, X_i, X_j, X_k \in E_p$

where C is a cluster that has members $X_1, X_2, \ldots, X_m \in E_p$, E_p is p-dimensional Euclidean Space, and $d()$ is a distance function or metric. Next, a dispersion measure has the following measures:

- $S(X_1, X_2, \ldots, X_m) \geq 0$
- $S(X_1, X_2, \ldots, X_m) = 0 \iff X_i = X_j \quad \forall\, X_i, X_j \in C$

where M_{ij} is the distance between vectors which are chosen as characteristic of clusters i and j, and S_i and S_j are the dispersion of clusters i and j, respectively.

Then, we can define the function $R(S_i, S_j, M_{ij})$ as a general cluster separation measure that allows to compute the average similarity of each cluster with its most similar cluster, and holds the following properties:

- $R(S_i, S_j, M_{ij}) \geq 0$
- $R(S_i, S_j, M_{ij}) = R(S_j, S_i, M_{ji})$
- $R(S_i, S_j, M_{ij}) \iff S_i = S_j = 0$
- $S_j = S_k \wedge M_{ij} < M_{ik} \Rightarrow R(S_i, S_j, M_ij) > R(S_i, S_k, M_{ik})$
- $M_{ij} = M_{ik} \wedge S_j > S_k \Rightarrow R(S_i, S_k, M_{ij}) > R(S_i, S_k, M_{ik})$

Then, the cluster separation measure that satisfies the previous properties is expressed as

$$R_{ij} \equiv \frac{S_i + Sj}{M_{ij}}$$

Lastly, the Davies-Bouldin score is defined as

$$\bar{R} \equiv \frac{1}{N} \sum_{i=1}^{N} R_i$$

where R_i is equivalent to the maximum of R_{ij} $i \neq j$. In other words, $\bar{R}$ is the system-wide average of the similarity measures of each cluster with its most similar cluster, meaning that the "best" number of clusters will be that which minimizes $\bar{R}$.

This can be demonstrated with the following distance function, dispersion measure and characteristic vector:

$$S_i = \left\{ \frac{1}{T_i} \sum_{j=1}^{T_i} |X_j - A_i|^q \right\}^{\frac{1}{q}}$$

where T_i is the number of vectors in cluster i. A_i is the centroid of cluster i.

Next, we define M_{ij} as the Minkowski metric of the centroids of clusters i, and j, expressed as

$$M_{i,j} = \left(\sum_{k=1}^{N} |a_{ki} - a_{kj}|^p \right)^{\frac{1}{p}}$$

where a_{ki} is the k-th component of the n-dimensional vector a_i which is the centroid of cluster i.

When $p = 1$, M_{ij} reduces to the Manhattan Distance. Moreover, when $p = 2$, M_{ij} is the Euclidean distance between centroids. S_j is the q-th root of the q-th moment of the points of cluster i about their mean. If $q = 1$, S_i becomes the average Euclidean distance of vectors in cluster i to the centroid of cluster i. If $q = 2$, S_i is the standard deviation of the distance between of samples in a cluster to the respective cluster center. If $p = q = 2$, then R_{ij} is the reciprocal to the Fisher similarity measure [16] for clusters i and j. Hereon, the Davies-Bouldin Score will appear as DB.

3.5.4 Entropy

Since we do not have a ground truth and there is no certainty on how emails are structured and neither we know the shape of the embedded space, we need to measure the uncertainty of the cluster structure. To do this, we measure our cluster stability with Entropy.

Following Shannon [40] definition, let $C = \{c_1, c_2, \ldots, c_k\}$ be the set of clusters, then we define the entropy of a cluster as

$$H(c) = -\sum_{k \in K} P(c_k) \log P(c_k)$$

where k is the classification of set K classifications, and $P(c_k)$ is the probability of an observation being classified as k in cluster c.

In consequence, the total entropy of the obtained clusters would be:

$$H(C) = \sum_{c \in C} H(C) \frac{N_c}{N}$$

This measure has been used extensively in cluster quality measure to evaluate how stable a cluster system is. The lower the entropy, the more stable is the cluster. It also used as a starting points for several methods, including Homogeneity and Completeness Scores.

3.5.5 Homogeneity and Completeness Scores

The Homogeneity and Completeness Scores were formulated by Rosenberg and Hirschberg [34]. They start from the idea that external validation is necessary by establishing *a priori* determined ground-truth class labels. Once having these labels, it should be possible to quantify how imperfect the clustering solution is. In fact, cluster evaluation measure should fulfill two criteria: Homogeneity and Completeness. The former concept refers that a cluster is homogeneous when it contains member of a single class. The latter means that a cluster is complete if all member of a given class are assigned to the same cluster. To measure this, they designed the V-Measure, or "Validity" measure. It is based on external cluster entropy, and is computed as the harmonic mean of distinct homogeneity and completeness scores in a similar way as it is done with precision and recall into the F-Measure [45].

Formally, let N be data points of a a given data set that is partitioned into a set of classes $C = \{c_i | i \ldots, n\}$ and a set of clusters $K = \{k_i | 1, \ldots, m\}$. Then, let A be a contingency table which represents a cluster solution given by a clustering algorithm. In this sense, the homogeneity score is defined as

$$h = 1 - \frac{H(C|K)}{H(C)}$$

Where $H(C|K)$ is defined as

$$H(C|K) = -\sum_{c=1}^{|C|} \frac{\sum_{k=1}^{|K|} a_{c,k}}{n} \cdot \log \left(\frac{\sum_{k=1}^{|K|} a_{c,k}}{n} \right)$$

and the entropy of classes are defined as

$$H(C) = -\sum_{c=1}^{|C|} \frac{a_c}{n} \cdot \log \left(\frac{a_c}{n} \right)$$

where a_c, a_k are the numbers of points belonging to class c and cluster k respectively, and $a_{c,k}$ the points from class c assigned to cluster k. Therefore, the V-measure is defined as

$$v = 2 \cdot \frac{h \cdot c}{h + c}.$$

Homogeneity and Completeness run in opposite directions. Increasing Homogeneity deceases completeness and vice-versa. However, this measure is not useful when there is no way to obtain a ground truth.

3.6 Distances

So far we have discussed the inner workings of the clustering algorithms and how to evaluate their respective clusters. Now, we will describe the metrics and distances that we used for the email exploration.

3.6.1 Euclidean Distance

The Euclidean distance is the departing metric for several methods, including the clustering algorithms and quality evaluation scores previously described. This is due in order to be able to measure the distance between two points using the length of a segment between said points.

Indeed, let p and q two points on a real line, then the distance between them in high dimensions is

$$d(p, q) = \sqrt{(p - q)^2}$$

As we stated before, the Euclidean distance is extensively used for several methods since it possesses all defining properties of metric space:

- It is symmetric: $\forall\ p, q \Rightarrow d(p, q) = d(q, p)$.
- It is positive: $d(p, q) \geq 0$ with equality $\iff p = q$.
- It obeys triangle inequality: $d(p, q) \leq d(p, q) + d(q, r)$.

Furthermore, there are multiple distances implemented in plenty of programming modules that include the Euclidean Distance. In particular, the Minkowski distance generalizes the Euclidean Distance along with the Manhattan Distance. Formally, let $X = (x_1, \ldots, x_n)$ and $Y = (y_1, \ldots, y_n) \in \mathcal{R}^n$, the Minkowski distance of order $p \in \mathcal{Z}$ is defined as

$$d(X, Y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}}$$

in which cases where $p = 2$ is equivalent to the Euclidean Distance and $p = 1$ to the Manhattan Distance.

3.6.2 Cosine similarity and L2-Norm

The cosine similarity is widely used for information retrieval and text matching, specially when working with term frequency vectors. It offers the advantage of being a fast distance computation since it can be used on sparse vectors and saves one vector subtraction.

Let **u** and **v** be two document vectors, then the cosine similarity is defined as

$$\cos(\theta) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \cdot \|\mathbf{v}\|} \in [-1, 1]$$

where -1 means that the components of the two vectors are exactly opposite, whereas 1 means they are exactly the same.

Nonetheless, a problematic property of the cosine similarity is that it reflects a relative comparison of individual vector dimensions:

$$\forall\, a, V \Rightarrow \text{max}(S(V, aV))$$

Hence it is more appropriate for frequency than absolute values. It can also be expressed in terms of the Euclidean distance as

$$d(\mathbf{u}, \mathbf{v}) = \frac{\|\mathbf{u} - \mathbf{v}\|^2}{2} \text{ when } \|\mathbf{u}\|^2 = \|\mathbf{v}\|^2 = 1$$

In this sense, the cosine distance is half of the squared Euclidean distance and it does not satisfy the triangle inequality property required to be a metric. An alternative to solve this caveat is to work with angular distances or obtaining the L_2 normalisation of vectors. Still, cosine similarity exists in an Euclidean space. However, the cosine similarity is preferred in several works related to word embeddings and document embeddings [33].

3.6.3 Word Mover's Distance

The Word Mover's Distance (WMD) is a widely used metrics in the field of text analysis research, as it not only finds the similarity of words, but also considers the underlying geometry [38], i.e. semantically similar words are allocated in a similar space. It has now become one of the persistent tools for NLP and for other several fields [50]. The seminal work of WMD, done by Kusner et al. [25], performed several tests with kNN classifiers for document classification.

Before continuing the discussion of the WMD, it is important to define where did the distance come from. It is a special case of the Wasserstein Distance, or Earth Mover's Distance [36] [49]. The main idea behind this type of distance is to calculate the cost of "transporting" one word embedding to another. A more similar word would be less costly than those that are dissimilar. For this, it considers the n-grammed Bag-of-Words (**nBOW**) representation, the *Word travel cost* and its *document distance*.

Regarding the **nBOW** representation, let vector **d** be a point on the $n-1$ dimensional simplex of word distributions. Then consider two documents that although have different unique words that lie in different regions of this simple, said words are semantically close. Then, after preprocessing, the resulting nBOW vectors **d** and **d**′ are close to maximum simplex distance, but their true distance is shorter.

Since we want to consider the semantic closeness between word pairs into the document distance metric, we use the Euclidean distance in the word embedding space. Indeed, calculating the distance between word i and word j becomes

$$c(i, j) = \|\mathbf{x}_i - \mathbf{x}_j\|_2$$

which is identified as the associated *cost* of moving one word to another. A nice property of this costs is that since $c(i,)$ is metric, the WMD is also a metric.

Naturally, this travel cost between two words will be extrapolated to distances between two documents. Let $\mathbf{d}$ and $\mathbf{d}'$ be the nBOW representation of two text documents. Considering that $\mathbf{T} \in \mathcal{R}^{n \times n}$ is a sparse flow matrix where $\mathbf{T}_{ij} \geq 0$ means how much of a word i in $\mathbf{d}$ is transported to word j in $\mathbf{d}'$. This outgoing flow will be denoted as $\sum_j \mathbf{T}_{ij} = d_i$. Similarly, the amount of incoming flow to word j is denoted by $\sum_i \mathbf{T}_{ij} = \mathbf{d}'_j$. Therefore, the distance between this two documents is the minimum weighted cumulative cost required to move all words from $\mathbf{d}$ to $\mathbf{d}'$, which is defined as

$$\sum_{ij} \mathbf{T}_{ij} c(i,j).$$

Once we define the cost, the WMD will try to solve the following linear program

$$\begin{aligned} \min_{\mathbf{T} \geq 0} \quad & \sum_{i,j=1}^{n} \mathbf{T}_{i,j} c(i,j) \\ \text{subject to:} \quad & \sum_{j=1}^{n} \mathbf{T}_{ij} = d_i \quad \forall i \in \{1,\dots,n\} \\ & \sum_{i=1}^{n} \mathbf{T}_{ij} = d'_j \quad \forall j \in \{1,\dots,n\} \end{aligned}$$

However, a major drawback is its high time complexity of $O(p^3 \log p)$. This becomes prohibitive for large sizes of p.

As an alternative to this linear program, there is the word centroid distance, which is based on Rubner, Tomasi, and Guibas [36] where the centroid distance must be lower bound of WMD between documents $\mathbf{d}, \mathbf{d}'$, defining the distance as

$$\sum_{i,j=1}^{n} \mathbf{T}_{ij}\, c(i,j) = \|\mathbf{X}\mathbf{d} - \mathbf{X}\mathbf{d}'\|_2$$

where each document is represented by its weighted average word vector. It has a runtime complexity of $O(dp)$.

Another approach to reduce the original complexity is to relax the linear program constraints. The WCD is not tight [25] but can be tighten by removing one of the two constrains. For instance, removing the second constrain, the linear problem would become

$$\begin{aligned} \min_{\mathbf{T} \geq 0} \quad & \sum_{i,j=1}^{n} \mathbf{T}_{i,j} c(i,j) \\ \text{subject to:} \quad & \sum_{j=1}^{n} \mathbf{T}_{ij} = d_i \quad \forall i \in \{1,\dots,n\} \end{aligned}$$

Then, the relaxed problem will find the optimal solution in $\mathbf{T}^*$, defined as

$$\mathbf{T}^*_{ij} = \begin{cases} d_i & \text{if } j = \arg\min_j c(i,j) \\ 0 & \text{otherwise} \end{cases}$$

Then, in the optimality $\mathbf{T}^*$ yields a minimum objective value, which is showed as

$$\sum_j \mathbf{T}_{ij} c(i,j) \geq \sum_j \mathbf{T}^*_{ij} c(i,j)$$

Here it is only necessary to identify any word i with the closest word $j^* = \arg\min_j c(i,j)$, i.e., the nearest neighbor search in the Euclidean space of the word embedding. Therefore, the computation relies on pairwise distance matrices to obtain faster computations that can be reduced to quadratic complexity instead of cubic [3].

3.7 Document Vectors

In this Subsection we explore the concepts of TF-IDF and doc2Vec vectors to transform Enron emails into document vectors.

3.7.1 TF-IDF

TF-IDF is short for term frequency-inverse document frequency, and it is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as a weighting factor in information retrieval and text mining. The TF-IDF value increases proportionally to the number of times a word appears in the document and is offset by the frequency of the word in the corpus.

The TF-IDF has two principles:

- The greater the frequency of t in d, the greater the weight it should have.
- The more frequent t is in the whole collection, the lower the weight it has in the documents.

Therefore, formally, a document is a vector of weights

$$d = [W_{d,1}, \ldots, w_{d_i}, \ldots, w_{d,T}]$$

Each weight is a product of two terms:

$$w_{d,i} = \mathsf{tf}_{d,i} \cdot \mathsf{idf}_i$$

So, the term frequency of term tf is

$$\mathsf{tf}_{d,i} = \frac{\mathsf{f}_{d,i}}{\max_j \mathsf{f}_{d,i}}$$

where $\mathsf{f}_{d,j}$ is the frequency of t_j in d.

And the inverse of the document frequency idf is

$$\mathsf{idf}_i = \log_2 \frac{D}{\mathsf{df}_i}$$

where D is the number of documents and df_i is the number of documents containing term t_i.

Once calculated the TF-IDF, it is possible to measure how similar the document vectors are using the cosine similarity.

3.7.2 Paragraph Vectors

Proposed by Le and Mikolov [27], a paragraph vector (also known as doc2vec and document embeddings) is a neural network trained to predict words in a document given a context. In turn, the context is provided by other words in the document. The training data is a collection of documents, each with a unique id. The network is trained to predict the next word in the document, given the previous words. This text representation is recommended for cases when there are little to no labeled data.

Doc2vec is built on top of word2vec, a word embeddings vector [30], and constructs representations of input sequences of variable length, applicable to texts of any length (sentences, paragraphs, and documents). It does not require tuning word weighting functions nor rely on parse trees.

Originally, word2vec tries to predict a word through other words that appear in the context. But in doc2vec, every word is mapped to unique vector, which is represented as a column in a matrix W. Then, the column is indexed by a position of the word in the vocabulary. After that, it obtains the features for prediction of the next word in a sentence by concatenation or summation of the vectors.

In the case of word2vec, the definition of the framework is as follows. Let $w_1, \ldots, w_T$ a sequence of training words, then to maximize the average log probability, the vector model uses

$$\frac{1}{T}\sum_{t=k}^{T-k} \log P\left(w_t \,|\, w_{t-k}, \ldots, w_{t+k}\right)$$

Then a multiclass classifier performs the prediction. For instance, typically the framework uses softmax, which is expressed as

$$P(w_t \,|\, w_{t-k}, \ldots, w_{t+k}) = \frac{e^{y_{w_t}}}{\sum_i e^{y_i}}$$

where y_i is un-normalized log probability for each output of word i, which is expressed as

$$y = b + Uh(w_t \,|\, w_{t-k}, \ldots, w_{t+k}; W)$$

where U, b are softmax parameter, and h is a concatenation or average of word vectors extracted from W [27].

Regarding doc2vec, there are two versions: **distributed memory** (PV-DM) and **distributed bag-of-words** (PV-DBOW). In the first version, what changes is that h is constructed from W and D, i.e., the D matrix is added. It is called distributed memory because it acts as a memory that recalls what is missing from the current context or topic of the paragraph. The training is done by stochastic gradient descent with backpropagation. It has two stages:

- Training stage: trains to get W word vectors, the softmax weights and paragraph vectors D on already observed paragraphs;

- Inference stage: obtain paragraph vectors D for new paragraphs by adding more columns in D, and gradient descending on D while fixing the softmax weights, and then uses D to make a prediction with another standard classifier.

It also inherits the semantics of words and takes into consideration the word order, similar to an n-gram model but without losing generalization capacity. A basic model is shown in Figure 6. A caveat of this model is that it needs to store softmax weights and word vectors.

The second version consists of ignoring the word context in the input and forcing the model to predict words iterating over the stochastic gradient descent and sampling a text window and forming a classification task given the PV. This is why it is called Distributed Bag of Words. It offers the advantage of storing less data as it only needs to store the softmax weights without the paragraph vectors. This is illustrated in Figure 7. However, according to Le and Mikolov [27], the PV-DBOW does not perform as good as the PV-DM.

One of the most used implementations of the Paragraph Vector is the Gensim's doc2vec module [33]. It supports both PV-DM and PV-DBOW, having as default the latter. Currently, it only implements hierarchical softmax and negative sampling [30] [29].

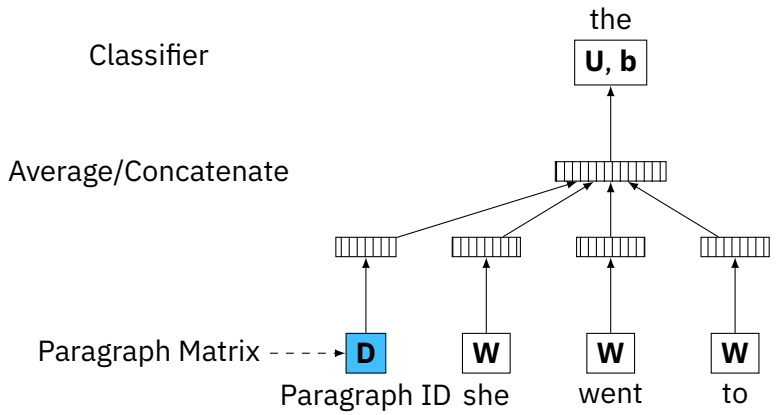


Figure 6: Paragraph Vector - Distributed Memory (PV-DM) model

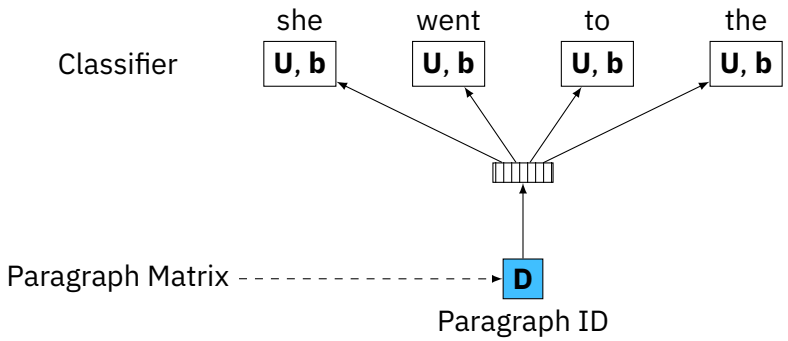


Figure 7: Paragraph Vector - Distributed Bag of Words (PV-DBOW) model

4 Methodology

In this Section we describe the process method used for the project, as well as a description on how we address clustering methods, clusters evaluation, distance matrices and text representation inside the described project, in line with defined objectives. For starters, the process is based on the CRISP-DM Methodology and establish that we will focus in some of the stages of the process. Then, we explain how we implemented the clustering methods K-Means, DBSCAN and HDBSCAN; cluster metrics, which are Silhouette Scores, the Calinsky-Harabasz, Davies-Bouldin and Entropy; also Euclidean Distance, L2-Norm, Word Mover's Distance metrics as well as the Cosine Similarity; lastly, the text representations.

4.1 CRISP-DM Methodology

According to this project objectives, we will follow a feature based approach by performing K-Means, DBSCAN and HDBSCAN over the four data splits with four distances (euclidean, cosine, l2 and WMD), and lastly evaluate the obtained cluster quality with the four scores. For this project, we use the *scikit-learn* [32] implementations for the Calinski-Harabasz and Davies-Bouldin scores, and the RAPIDS's *cuML* [43] implementations for the Silhouette score and the entropy. Since the general objective is to find automation opportunities, it is considered that for a successful deployment the pipeline of the model should follow a CRISP-DM methodology [48]. Figure 8 portrays the basic CRoss Industry Standard Process for Data Mining. The center of this methodology is the data collection of emails, in this case Enron emails. The CRISP-DM is composed by six phases:

- **Business understanding**: Grasp what the business needs regarding emails are (folder classification? ticket generation? automatic answers?).
- **Data understanding**: It is important to have a wide knowledge over how emails are going to be collected, describe their properties and metadata, as well as the capacity to explore relationship among other emails or users, along with the assessment of data quality (which in case of the Enron emails is a major concern).
- **Data preparation**: Create subsets of data, pre-process for later text representations, discard erroneous values, in addition to creating the correspondent attributes, and integrate external data and format appropriately.
- **Modeling**: Select the appropriate models, obtain the corresponding distance matrices, build the model and assess which model is to prevail.
- **Evaluation**: More than a model assessment, it is about the appraisal of success criteria, review the process and the findings to determine following steps. Here lies the stage were the CRISP-DM becomes iterative.
- **Deployment**: Develop and document a deployment plan, monitor and establish a maintenance plan, produce reports and review the project.

An important consideration when detecting automation opportunities is having a clear business understanding. However, since we follow a broad approach, we will focus on data understanding, data preparation and modeling. The rest of the stages are not in the focus of this project.

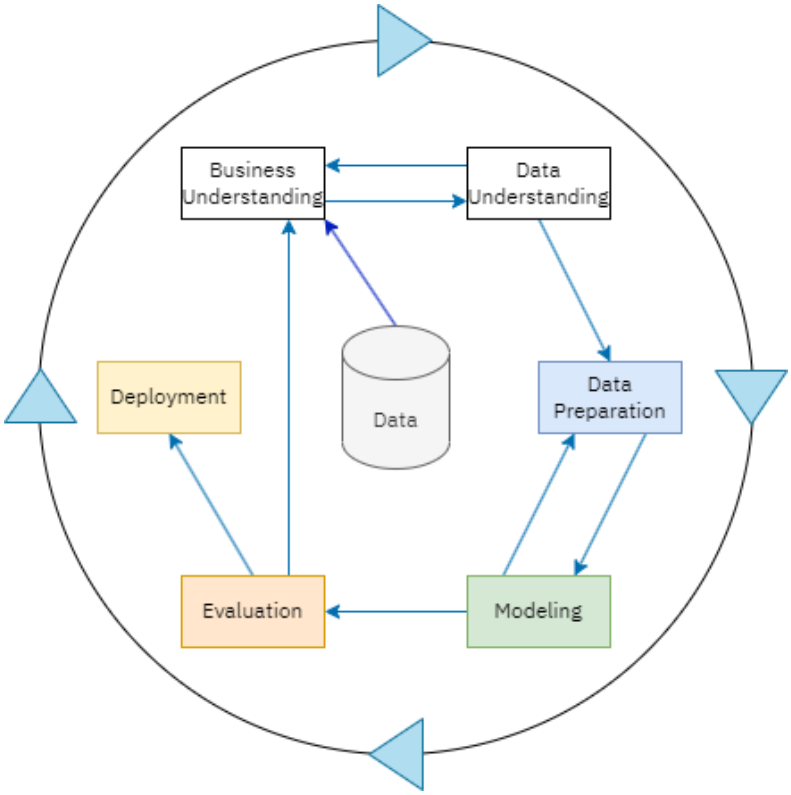


Figure 8: CRISP-DM methodology for an email automation.

Figure 9 displays two stages: data preparation and modeling. In the first stage, we identify email chains. Then, we clean the data, tokenize it and remove noise data. After that, we transform the obtained data into text representation, which in our case are doc2vec embedded documents. We finish the stage by splitting data according to email chain lengths. Continuing with the following stage, first we obtain the distance metrics with euclidean, l2 and WMD metrics as well as cosine similarity. After that, we perform three clustering methods: K-Means, DBSCAN, and HDBSCAN. It is important to note that in the case of DBSCAN, we apply a Heuristic method to adjust the Epsilon-Neighborhood by calculating the slope change of Nearest Neighbors. For each of the algorithms we select a range of values and iterate over each correspondent parameters. Afterwards, given that we do not have a ground-truth for the each email, we use the following scores that do not require true-labels: Silhouette Coefficient, Calinski-Harabasz Score, David-Bouldin Score and the Entropy. We explore visually a small sample of emails depending on the characteristics of the best results that we obtained to see if the method does arrive to a meaningful output. Once we finish the process, we proceed to the evaluation stage to see if it is necessary to review the process and results and start the overall process again. Each of these steps are detailed in Section 5. We now describe theory behind each of these steps below.

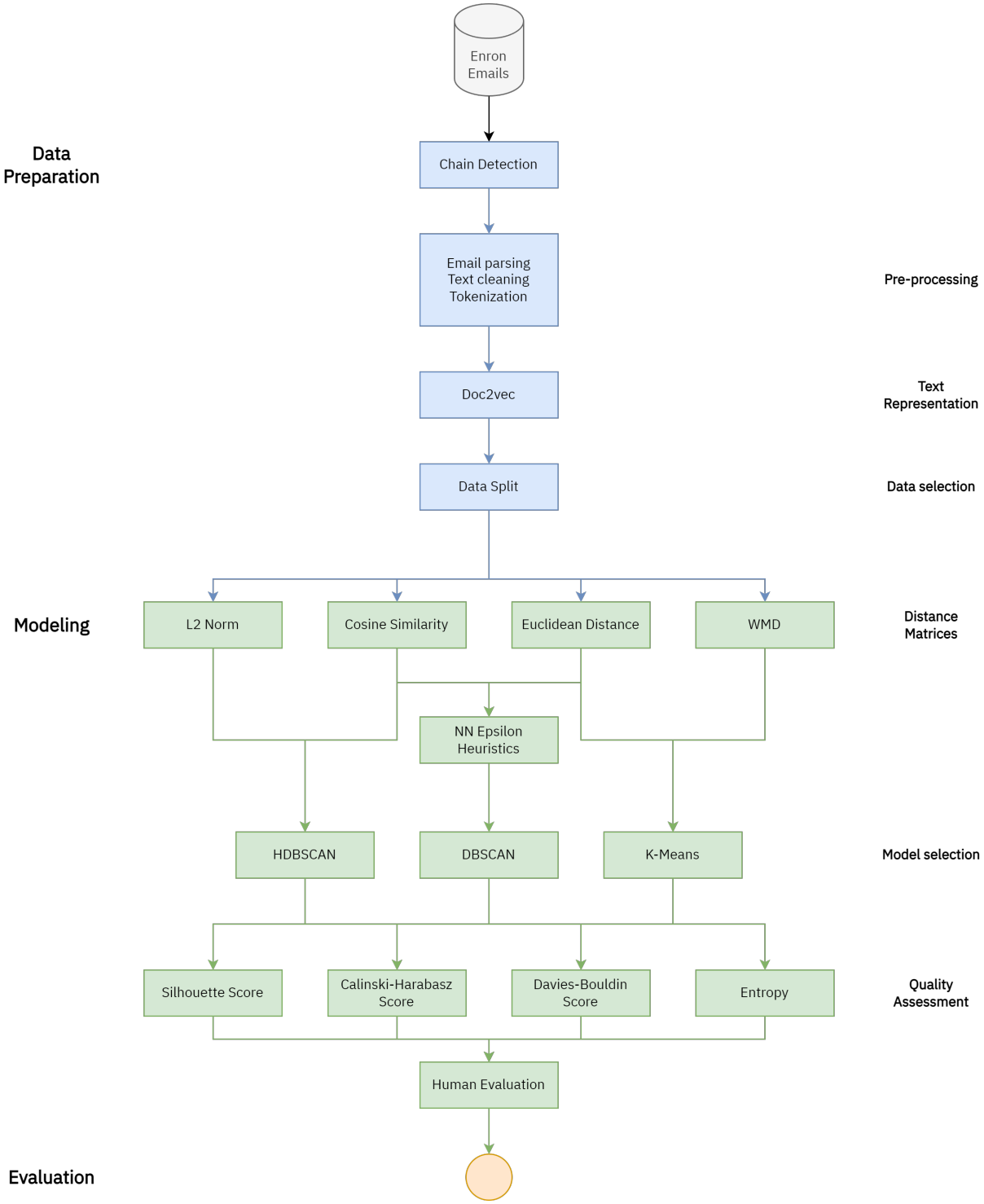


Figure 9: Data Preparation and Modeling Pipeline

4.2 Development

Once we defined our process methodology, we explain in this subsection what we did in each of the stages of the pipeline. We start with Data Preparation and Chain Detection, where we describe how we cleaned data and constructed the email chains. Next, how we tokenized and prepared data for text representation. Then, we explain the parameters used to train the paragraph vectors. And the subsequent parts deal with description of parameters used for the clustering methods, the distance matrices and clustering evaluation scores. We finish this subsection with a disclosure on human evaluation necessary to review if any process or pattern was detected.

4.2.1 Data Preparation and Chain Detection

Earlier in Section 3, we mentioned that the way the Enron Emails are organized present several cases of duplicity. Our own approach to deal with this duplicity was to parse date to timestamp and then create a key on subject, and all users involved. Finally, remove all duplicates using timestamp, keys and body of the email.

Before detecting the chains, we need to investigate the structure of the Enron Emails. We identify seven parts, and illustrate them in Figure 10. There we can locate Metadata, message body, users and subjects, envelop data or computer generated data, chain syntax of the email and attached documents that are no longer included in the original repository. In this work we paid special attention to the message body, the users and subject and the chain inside the message. These chains result problematic since they not always follow the metadata structure of the main email and sometimes they come from external sources that are difficult to follow. Indeed, we detected that there is a large amount of emails that contain

```
----Original Message----
```

in which we find untraceable chains or even spam. So, for the sake of simplicity we decided to eliminate all content below the identified chain and proceeded to create our own chain definition, also described in said Section.

The last step was to tokenize each email and remove non alphanumeric characters to later proceed to the Text Representation. To clean email message from unwanted characters we applied the following regular expression:

```
'_{4,}.*|\n{3,}|<[^>]*>|-{4,}(.*)(\d{2}:\d{2}:\d{2})\s*(PM|AM)'
```

The steps for calculating the chains and their respective chain length was as follows:

1. All unique users involved for each email were allocated in a single array and sorted alphabetically.
2. A timestamp in seconds was calculated from the “date” field obtained after parsing the email. This decision bears some inconveniences such as parsing from inconsistencies

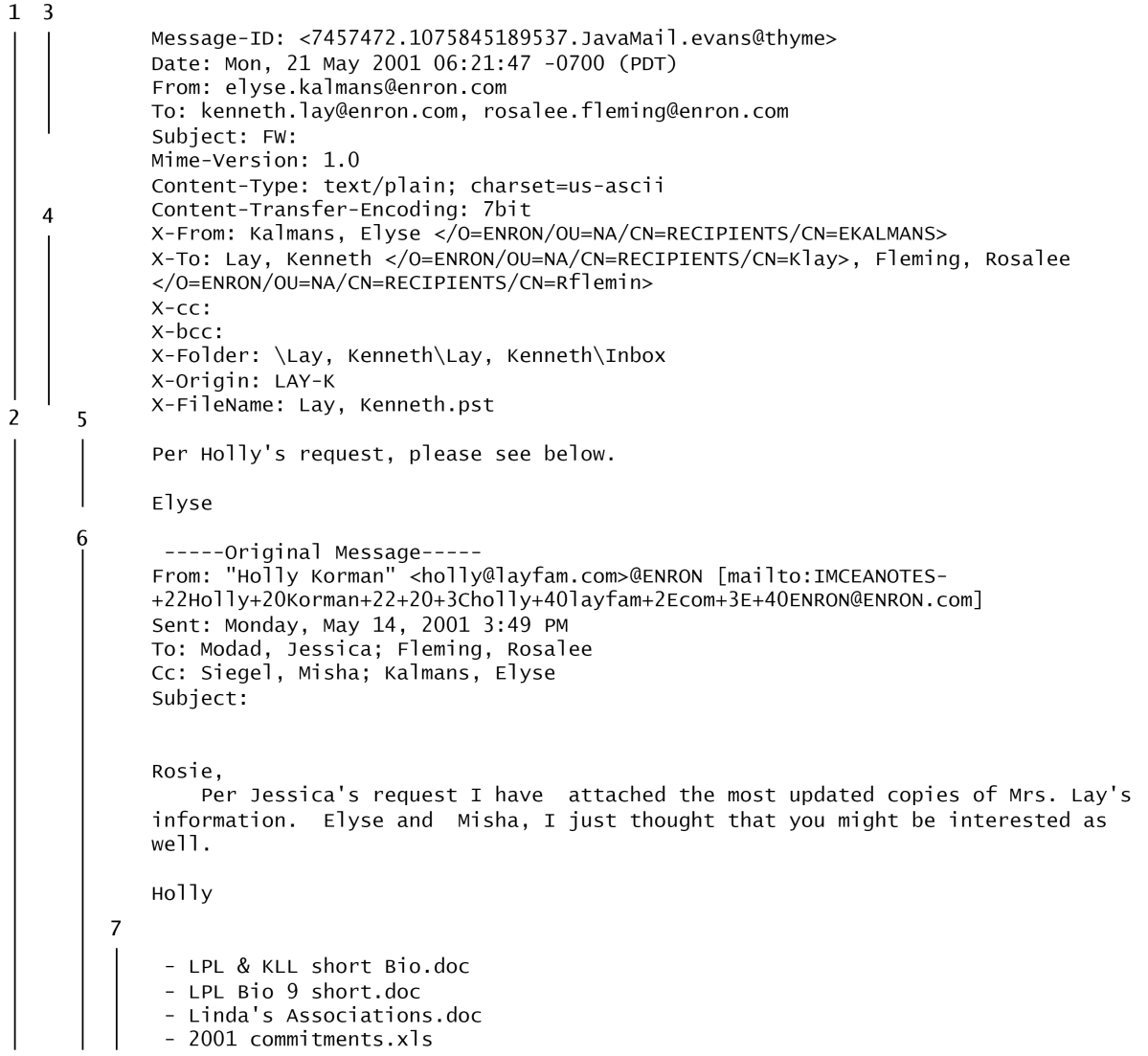


Figure 10: Parts of Enron Emails - 1. Metadata, 2. Message Body, 3. Users and Subject, 4. Computer-generated metadata, 5. Most recent content written by the sender, 6. Chain inside message, 7. Attached documents not included in the dataset.

from source (there are parsers that appear to belong to prior the 80 and others beyond 2020). However, for the sake of simplicity, they were left as is.

3. All emails were sorted by timestamp.
4. A key was created by concatenating Subject and sorted users. For this key, several patterns were removed from the subject, such as `RE`, `FWD`, `FW`. For this task, original Subjects were kept for later steps.
5. To consider a chain, the key with sender plus ordered subjects were allocated to a dictionary, assigning them a unique id.
    - If the original subject do not contain the `RE:` prefix and there is no email with a prior timestamp than that email, it is considered as the initial email from the chain, assigning it a unique ID. This is registered as false in a boolean column to register if the email is a reply.
    - if the original subject does contain a `RE:`, it will be considered as the same chain from the previous timestamped email with the same subject and users. It will be allocated for the same chain id as the previous corresponding email.
    - if the email contains the same subject and users as others emails but does not contain the `RE:`, it will be considered as a new chain. Here we assume that are independent chains, although we recognise that it can be a case where they belong to a sub-chain. For the purposes of this work, we assume that those emails are independent.
6. After obtaining different email chain ids, the number of emails containing the same chain id is considered as the length of id, thus storing it in an array.
7. For analysis purposes and due to hardware constraints, the final dataset was divided in five groups, which are detailed in the following sections, particularly in Table 2.

After obtaining chains, Table 1 shows the distributions of chains with their email chains. For instance, the first row shows that there are 203.172 emails with email chain of 1 single email. Next, there are 12.009 email chains with a length of 2 emails, and so on. Lastly, there are 192 threads with more than ten emails. Here are concentrated 5.332 emails, of which the largest chains have an email length of 798, 290, 233, 123 and 122.

Once eliminated duplicates and detected the chains, the our final data set is composed by 251.068 messages, 119.647 users, from which there are 19.993 unique senders. The distribution of senders and number of chains can be seen in Figure 11.

4.2.2 Pre-processing

As discussed in Section 3, a major challenge when working with the Enron emails is that they have a difficult format to work with. Although metadata is consistent, starting the time format and the way some recipients are reported.

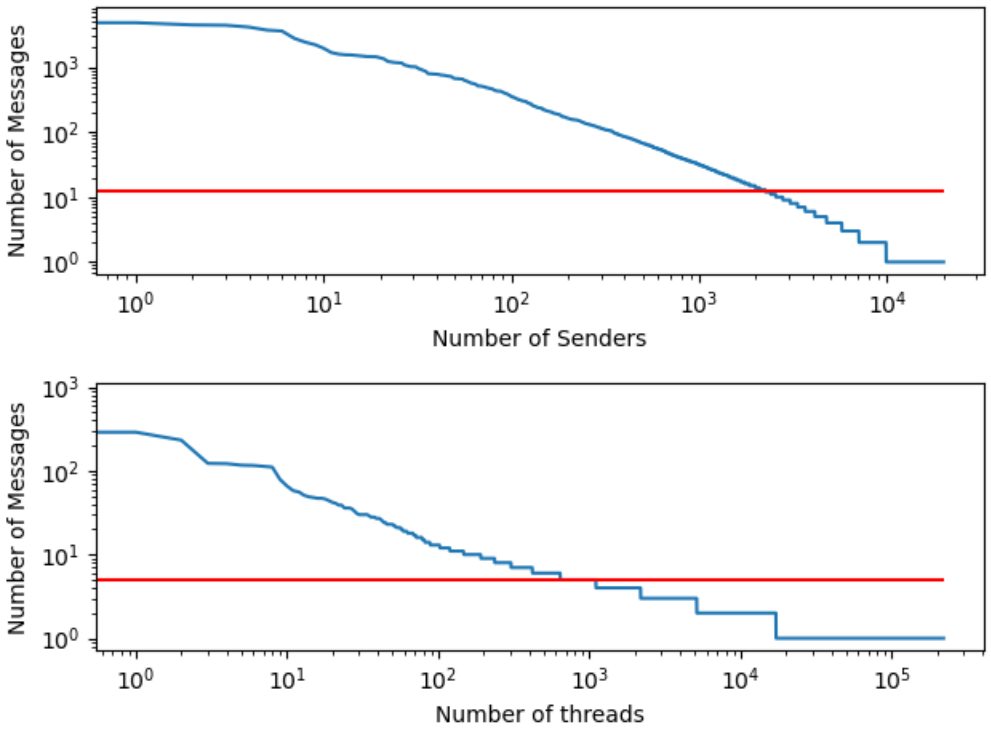


Figure 11: Email Characteristics of Senders and number of Chains per recipient.

Table 1: Distribution of chains length

Chain Length	Number of chains
1	203.172
2	12.009
3	2.946
4	1.077
5	463
6	222
7	116
8	67
9	45
>10	192

That being said, the way in which we approached the issue was by parsing the emails and store it in array-like data-structure. Then, we addressed four important parts: subject, the senders, the recipients and the body of the email. How we dealt with the first three was described previously in the chain detection. As per the body of the message, we removed noisy data via regular expressions.

We refer to noisy data all repetitive contents, tables and numbers that seemed like flat tables or even spreadsheets. Also, there were emails with raw HTML content, as well as whole databases.

The last step was to tokenize each email and remove non alphanumeric characters to later proceed to the Text Representation. To clean email message from unwanted characters we applied the following regular expression:

```
'_{4,}.*|\n{3,}|<[^>]*>|-{4,}(.*)(\d{2}:\d{2}:\d{2})\s*(PM|AM)'
```

4.2.3 Text Representation

Proceeding with analysis, each message was tokenized with lowercase, removing those with only digits and keeping only alphanumeric characters to later calculate doc2vec vectors. These embedded documents were processed immediately after the first stage of pre-processing.

It is worth noting that a common practice in literature is calculating word and document embeddings with 300 vectors. However, there are also other works that report stability with only 50 vectors. So to test both scenarios, two versions of the doc2vec embeddings with both of the dimensions.

Moreover, the rest of parameters chosen to train the paragraph vector were the following:

- Window size of 15.
- Minimum count of 1.
- 20 train epochs.
- Alpha of $0,25$.
- Threshold of alpah at 1.0e-5.
- Enabled version being PV-DBOW.

Regarding the number of train epochs, Le and Mikolov [27] recommend a number between 10 and 20 epochs. We decided to choose 20 iterations to simplify the research. Additionally, we used the PV-DBOW since the size of our data is large and memory is limited.

Another important step that we undertook was keeping the same index of data for each vector created and this same index was inherited to the labels obtained in the Modeling Stage. This was done to be able to match the obtained label to the actual email.

With respect of TF-IDF, one of the main challenges faced in this project was the limited available hardware that can perform complex computations such as the one for the WMD, and in addition to the extensive literature that used TF-IDF, we decided to focus the research over the Paragraph Vectors.

4.2.4 Data selection

The final part of the data preparation stage, described in Section 4, involves splitting the whole data set in manageable parts. Given the chain length sizes with different orders of magnitude, the data sets were split in five groups according to their chain length, and whose cardinality are shown in Table 2. Several messages also contain empty strings, for which they are considered as missing values and not counted. After removing said empty messages, we obtain the final cardinality of each group that is going to be analysed. Notice that out of all groups, the chain length of 1 messages contain the largest amount of empty messages with 3.4 %. Moreover, the group of chain lengths greater than 10 is considered a group apart since it contains the several interactions along time, so we explore if these chains are considered as a group.

Table 2: Data set groups by chain length and their respective proportions with raw messages and cleaned messages

Group	**Raw emails**	**Without empty messages**	**% of empty messages**
Chain length $= 1$	203.172	196.212	3.4
Chain length $= 2$	24.018	23.841	0.7
Chain length $= 3$	8.838	8.769	0.8
$10 >$ Chain length ≥ 4	9.708	9.676	0.3
Chain length ≥ 10	5.332	5.324	0.2

4.2.5 Distance Matrices

As mentioned in Section 4, we used GPU acceleration with the Team [43] library. In Particular, the module `cuML` offers several implementations based on the Pedregosa et al. [32] library but enabled for NVIDIA GPU. We used `pairwise_distances` to calculate the Euclidean distance, cosine similarity and the L2-norm metric.

In the case of the Word Mover's Distance, it required a different approach. The Gensim library does have an integrated method for a objects of class `word2vec` and `doc2vec` that calculates the WMD based on Kusner et al. [25]. However, this method is only available for CPU, and since the runtime complexity is cubic, it made prohibitive to perform several tests. Another implementation is the Wasserstein Distance of the Scipy library [46], where one can make a custom implementation of the WMD, however was also discarded since it was also only available for CPU. Another implementation called `dist_matrix` was done by Baird and Sparks [5], which uses the Numba library [26], enabling computations on GPU. The `dist_matrix` is capable to calculate a weighted Wasserstein distance from a 10.000 $\times$ 10.000 matrix in about 15

seconds. In our case, following the Kusner et al. [25] method, calculated first an Euclidean matrix and then used the `dist_matrix` to obtain the final matrix.

4.2.6 Clustering Methods

The `cuML` [43] module also provide the implementations of K-Means, DBSCAN and HDBSCAN enabled for GPU. The three versions were used with pre-computed distances since this accelerates calculation time.

About K-Means, the default initialization algorithm is the K-Means++ version, whose pseudocode 2 is explained in 4. Also, for simplicity purposes, we also selected the default 300 iterations before convergence. The only parameter to be tuned is the k clusters, which at first was tested in a range of $[2, 100]$, but after several tests we downed to $[2, 50]$, and, as we show later, the best results were in a range of $[2, 5]$.

In the case of DBSCAN, we had to tune for three parameters: the epsilon value, minimum cluster size (named `min_samples` in the implementation) and the distance. With respect to the Epsilon neighborhood, we used two approaches: The first was by selecting the `eps` parameter close to the default values, which are 0,5, around a range of [0,20, 1,00] (with two decimals) following the recommendations of Schubert et al. [39] on using small number of `eps` for better results. The second approach is using the Nearest Neighbors' Heuristics. First, we calculate the NN for using also `NearestNeighbors` class of cuML, iterating the parameter `n_neighbors` in a range of $[2, 30]$. Then for each NN, we sorted the values and obtained the slope and the resulting array of values is used to iterate over DBSCAN. In the case of the `min_samples`, we first iterated over values in a range of $[2, 101]$ but for values of > 15 there was found consistently in a single cluster. At the end we run our tests with a range of $[2, 15]$.

In the case of HDBSCAN, we iterate over three parameters: `min_samples` and `min_clt_size`, and distance. Although the advantage of HDBSCAN is that we do not need to tune for the Epsilon-neighborhood parameter, as explained in Section 4, we did wanted to test `min_samples` to check how conservative we must be with the clustering size. First, we used a range of $[2, 20]$, but then we reduced it to $[2, 10]$. On the other hand, for the `min_clt_size` we kept a range of $[2, 25]$

Lastly, for the distance parameter, the cuML HDBSCAN version is available to work only the Euclidean distance[1] amd does not allow pre-calculated metrics. To solve this problem, the PyPI module `hdbscan`, developed by McInnes, Healy, and Astels [28], was used. This module is only enabled for CPU but offers to implement core paralleling, so we set the `workers` parameter to the value that allows us to use all available cores.

[1] Although in the official documentation states that it is possible to calculate HDBSCAN with pre-computed distances. At the moment of writing this report, the issue was reported and a other metrics are still not available until further notice: `https://github.com/rapidsai/cuml/issues/4475#issuecomment-1011536398`.

4.2.7 Cluster Evaluation

The final stage of the modeling involves the quality assessment of the resulting clusters. Both the Silhouette Score and the Entropy are available in the cuML module[2]. In the case of the other two methods, it was used the implementations from Scikit-Learn.

After evaluating and comparing results, a human reviewing is necessary since we do not have a ground-truth and, therefore, it is not possible to test-validate through other traditional methods such as Completeness Score or Homogeneity Score.

[2] At the moment of writing this report, the stable version was 22.08 and the Silhouette implementation required that labels should be in integer type of 32 bits, otherwise it would not compile.

5 Results Analysis

This section provides a thorough explication of the results obtained after performing the steps of the data preparation and modeling stages. The explanation is done following the order of the data splits. It is important to mention that, although in Section 4 we explained how SL is a graphical representation of the cluster quality, it is customary to apply a dimensionality reduction technique to show how the vectors are distributed. After performing Principal Component Analysis (PCA) over the document vectors, we noticed that the explained variance of the first two components barely explained more than 1 % of data.

Another important aspect to be taken into account regarding DBSCAN heuristics is that the corresponding plots shown in Appendix B only revealed the L2-Norms results. This is due to the fact that with other distances the algorithm only finds one cluster, and in those cases the cluster evaluation was not performed.

Ultimately, following the processes showed in Figure 9, at the end of each section there is a brief analysis on some relevant aspects of the obtained clusters. It is not meant to be an exhaustive revision, as the amount of emails and nuances about employees' interactions would be worth of future research. Notwithstanding, in adherence to the project objectives, we show cases that we believe represent best automation opportunities.

5.1 Chains of Length 2

Given that this subset contains a large number of emails, performing HDBSCAN with the precomputed metrics not belonging to the `cuML` module was not possible. The same scenario happened with DBSCAN, which only the Euclidean distance was able to run successfully.

In regard to the results, Table 3 displays the best clustering results for each method by score. The best results for all scores were obtained with DBSCAN. Moreover, all methods in all scores suggested the best clustering choice is 2, with the exceptions of K-Means were the DB score suggested 20 clusters. In all of the scores except the Entropy, the best distance was the Euclidean and vectors with 300 dimensions. Additionally, two important points should be noted. We also detected that there are 314 emails allocated in one cluster, and 5 in another, whereas the rest of the 23510 were classified as noise. Secondly, the corresponding value of `Eps` parameter in all the best scores was around 0.9, which is smaller than the those values obtained with the heuristic knee values. Furthermore, the most stable cluster was 2.

Aside from the best results, it is worth to note that for K-Means and for scores SL, CH and the Entropy the behaviour is as expected, since the quality of clusters increases as the number of clusters decreases. However, the DB Score has some parameters that seem to make better clusters as the number of clusters increases.

With regard to DBSCAN, when considering the SL, CH and DB scores, we detected that the euclidean distance provides the best results. In the case of the Entropy, the WMD performs

Table 3: Best Scores for Chains of Length 2, focusing by score.

Score	Method	Distance	Dim	k (% Noise)	SL	CH	DB	Entropy	Eps	MinPts	Min Clt Size	Min Samples	NN
SL	DBSCAN	Euclidean	300	**2 (0.982)**	**0.974**	3260.839	0.149	0.102	0.630	2.000	-	-	-
	HDBSCAN	Euclidean	300	2 (0.977)	0.743	85.182	0.489	0.216	-	-	3.000	25.000	-
	KMeans	-	300	2	0.151	1.694	0.741	0.002	-	-	-	-	-
CH	DBSCAN	Euclidean	50	**2 (0.982)**	0.971	**4488.032**	0.080	0.103	0.560	2.000	-	-	-
	HDBSCAN	L2 Norm	300	2 (0.973)	0.720	236.593	0.525	0.454	-	-	4.000	22.000	-
	KMeans	-	50	2	0.043	130.748	5.806	0.615	-	-	-	-	-
DB	DBSCAN	Euclidean	50	**2 (0.982)**	0.971	4488.032	**0.080**	0.103	0.550	2.000	-	-	-
	HDBSCAN	Euclidean	300	2 (0.977)	0.743	85.182	0.489	0.216	-	-	3.000	25.000	-
	KMeans	-	300	3	0.146	1.791	0.726	0.004	-	-	-	-	-
Entropy	KMeans	-	300	**2**	0.151	1.694	0.741	**0.002**	-	-	-	-	-
	DBSCAN	Euclidean	300	2 (0.011)	0.370	7.192	0.783	0.003	5.436	2.000	-	-	30.000
	HDBSCAN	Cosine	50	2 (0.146)	0.044	2.010	1.646	0.004	-	-	2.000	34.000	-

slightly better than the rest. Conversely, the WMD obtains the worst values for the DB. Furthermore, focusing on the `Eps` parameter, the values obtained with the heuristics do not have good results at all. Moreover, regarding `Min Points`, we do not perceive a clear pattern, except for number > 10 the overall quality seems to decrease. Lastly, the case of HDBSCAN is rather straight forward. Of all scores, the 50 dimension vectors obtained the best results. The size of Minimum samples seem not to have a strong effect in the quality on the clusters, nor the Minimum Clusters Sizes.

We now proceed to inspect some emails and their respective clusters. For instance, the chain 209486, the emails of which were not placed in the same cluster. The clustered email corresponds to the DBSCAN labels with the highest SL score. The opening lines of the email seem to indicate that it is a sales message, however the second part shows a clustered email that contains a closure message. This incomplete chain exemplifies an opportunity to automate email responses. It is worth mentioning that we tried to track down the intermediate email interactions, but the dataset only contained these two emails. This chain can be consulted in Appendix C.3. After performing a survey on the obtained clusters, it was not possible to detect a clear interaction pattern.

5.2 Chains of Length 3

With regard to the subset of email chains with a length of three, it is clear that the best score values are also achieved with DBSCAN, vectors of 50 dimensions and all suggesting that 2 clusters is the best option. However, in Table 4 we can appreciate some variety regarding distances among the different scores. For instance, SL best value is achieved with the WMD metric, having two orders of magnitude larger values than any of the other scores. Same difference in size of score value happens with the CH, although in this case it is achieved with the Euclidean distance. In the case of the DB, WMD is once again the best option. It is important to note that for SL and DB, the DBSCAN classified 8766 emails in one cluster and 3 in another. This contrasts with the performance of the other scores, in which for CH allocated only 126 emails in a cluster, and 4 in another, while with the Entropy, the algorithm grouped 7905 emails in a cluster and a single email in another.

If we consider the hypertuning parameters results, we find that the 300 dimensional vectors

Table 4: Best Scores for Chains of Length 3, focusing by score.

Score	Method	Distance	Dim	k (% Noise)	SL	CH	DB	Entropy	Eps	MinPts	Min Clt Size	Min Samples
SL	DBSCAN	WMD	50	**2 (0.000)**	**12.712**	1.057	0.003	-	0.270	2.000	-	-
	HDBSCAN	L2 Norm	300	2 (0.984)	0.832	30.547	0.538	0.076	-	-	2.000	6.000
	KMeans	-	300	2	0.125	235.633	5.859	0.678	-	-	-	-
CH	DBSCAN	Euclidean	50	**2 (0.985)**	0.967	**4375.641**	0.140	0.137	0.790	4.000	-	-
	KMeans	-	300	2	0.125	235.633	5.859	0.678	-	-	-	-
	HDBSCAN	L2 Norm	300	4 (0.952)	0.376	180.775	0.896	0.134	-	-	2.000	5.000
DB	DBSCAN	WMD	50	**2 (0.000)**	12.712	1.057	**0.003**	-	0.250	2.000	-	-
	HDBSCAN	L2 Norm	300	2 (0.984)	0.832	30.547	0.538	0.076	-	-	2.000	6.000
	KMeans	-	300	19	0.057	29.396	2.523	1.361	-	-	-	-
Entropy	DBSCAN	Cosine	50	**2 (0.098)**	0.019	1.178	0.907	**0.001**	0.430	15.000	-	-
	HDBSCAN	Cosine	50	2 (0.047)	0.043	1.708	1.726	0.002	-	-	2.000	33.000
	KMeans	-	300	2	0.125	235.633	5.859	0.678	-	-	-	-

almost have better results than their 50 vectors counterparts in K-Means, but then again, according to SL, CH and Entropy, 2 clusters are detected as the better options.

The performance of DBSCAN differs from each score. It is important to highlight that for SL, the best value is the vector with a WMD and 50 dimensions, as noted previously. However, the plot suggests that this behavior is more like an outlier rather than a trend. In fact, without that data point, the Euclidean Distances points achieve better SL scores. The same scenario happens with the DB and Entropy scores. As to the CH scores, it is clear that Euclidean distances with 50 vectors dominate completely the scores. In contrast, the vectors with cosine similarity are consistently worse than every other metric. Regarding the parameters, the heuristics method for epsilon shows performs worse than using smalls values of `eps`, and also the smaller the `MinPts`, the better. However, the Entropy shows that some data points with small value of `MinPts` have clusters with more unstable structure.

Next, for HDBSCAN, we identified that the Euclidean distance with 300 dimension vectors performs better in all scores, closely followed by the L2-Norm and WMD. With respect to the parameters, it is worth stressing that the higher minimum samples, the better, getting the best values between 25 and 35. Conversely, the best Minimum clusters sizes are close to 2.

When we analysed the clustered emails, we detected an interesting behavior of what we suspect is the WMD effect. The chain 071164 corresponds to the labels obtained with DBSCAN and WMD with the highest SL score. The topic seems to be related to the interest of a Bolivian energy company to work with Enron and the reaction of the employees to this situation. The three messages of the chain belong to the Epsilon-Neighborhood but two were allocated in cluster 0 and the other in cluster 1. Notice that the latter is an email that references a web page post with sentences in Spanish; while the other emails are conversations around that message. Here are two main takes. The first one is that a potential detection of a topic of interest and a correctly automated system could create a task or generate an internal ticket whenever it detects this kind of behaviour. Needless to say, it is difficult to achieve it with the current clustering information, but the possibility does exist. These email chains can be consulted in Appendix C.4. Besides that, we also found no clear pattern of interaction.

5.3 Chains of Length greater or equal to 4, less than 10

DBSCAN dominates in other subsets, followed by HDBSCAN and K-Means, respectively, as reported in Table 6. Also, for all top scores, DBSCAN suggests 2 cores as the best option. A difference in behaviour with respect to the other subsets is that in this case the Euclidean Distance is the most dominant value in all metrics, with the exception the Cosine Similarity in the Entropy. Also, the best values were obtained with vectors of 50 dimensions. However, when assessing the distribution of clusters, we can see that the majority allocated around 139 emails in one cluster and 2 in another, leaving the rest of the 9535 emails outside the Epsilon-neighborhood. Indeed, for SL, CH and DB scores only 95 emails were allocated in a cluster. The exception is the Entropy with only 60 emails detected as noise. However, in all cases the difference in cluster size is noticeable.

Table 5: Best Scores for Chains of Length greater or equal to 4 and less than 10, focusing by score.

Score	Method	Distance	Dim	k (% Noise)	SL	CH	DB	Entropy	Eps	MinPts	Min Clt Size	Min Samples
SL	DBSCAN	WMD	300	**1**	**2.000**	-	-	-	0.200	15.000	-	-
	HDBSCAN	Euclidean	300	2 (0.985)	0.952	595.956	0.445	0.074	-	-	2.000	29.000
	KMeans	-	300	2	0.131	250.298	5.987	0.673	-	-	-	-
CH	DBSCAN	Euclidean	50	**2 (0.985)**	0.965	**8049.610**	0.121	0.211	0.670	5.000	-	-
	HDBSCAN	Euclidean	300	2 (0.985)	0.952	595.956	0.445	0.074	-	-	2.000	28.000
	KMeans	-	50	2	0.026	255.963	6.042	0.692	-	-	-	-
DB	DBSCAN	Euclidean	50	**2 (0.985)**	0.971	4648.157	**0.078**	0.074	0.500	2.000	-	-
	HDBSCAN	Euclidean	300	2 (0.985)	0.945	261.492	0.335	0.074	-	-	2.000	25.000
	KMeans	-	300	18	0.061	43.332	2.736	1.354	-	-	-	-
Entropy	DBSCAN	Cosine	50	**2 (0.029)**	0.058	0.829	1.081	**0.001**	0.430	7.000	-	-
	HDBSCAN	Cosine	50	2 (0.018)	0.053	1.831	1.679	0.002	-	-	2.000	26.000
	KMeans	-	300	2	0.131	250.298	5.987	0.673	-	-	-	-

Switching the attention toward parameters iterations, we detected that K-Means consistently performs better with small number of clusters in all of the scores. With respect to number of dimensions, for SL, DB and Entropy, the 300 dimensions perform better than 50 dimensions. The reverse happens with the CH score, where the 50 dimensions perform better.

As to the DBSCAN algorithm, we can appreciate how dominant the Euclidean distance is for every score. Cosine similarity has the worst outcomes in SL and Entropy, whereas in the DB score the WMD obtains the worst values. Regarding `Eps` we see again that the heuristics methods have a mediocre performance. The best results were around 0.5. In the case of the `MinPts`, there is a clear trend for better results for small values.

Regarding HDBSCAN, both the Euclidean distance and the L2-Norm are the best performing metrics than the others with 300 dimension vectors. Conversely, the WMD performs worse in SL, CH and DB scores, specially with 50 dimensions vectors. Focusing on the Minimum Clusters Size parameter, the small values have the best clusters, in concordance to the final k results. And in the case of minimum samples, we find again that minimum samples between 25 and 34 neighbors perform better.

Regarding visual inspection, we put special attention to chain 218660. Here, DBSCAN failed to detect a whole interaction of size 9, where it only allocated the email with index 193405 to cluster 0. This message contains a reaction to a previous answer but seem to fail to detect

that interaction. Indeed, cluster 0 has 122 emails, for which only one chain possess a complete chain, but unfortunately possess only monosyllabic words. Said email chain is located in Appendix C.8. In this case, neither we find a clear pattern of interaction.

5.4 Chains of Length greater or equal to 10

This subset contains the largest length of values, going up to more than 700 emails in chain. Nonetheless, all the algorithms suggest that these emails can be allocated in 2 clusters. Table 6 exhibits that DBSCAN is once again the dominant method with the exception of K-Means with the entropy. Furthermore, for scores CH and DB, the 50 dimension vectors perform better, whereas for SL and Entropy, the 300 dimension vectors. Also, the best distance was the Euclidean distance. In terms of cluster distribution, the DBSCAN algorithm allocated 93 emails in one cluster, 2 in a second cluster and the rest of the 5262 emails were classified as noise in the SL, CH, and DB cases. Conversely, the lowest Entropy value was obtained with KMeans, for which the algorithm allocated 5323 emails in one clusters and a single email in another cluster.

Table 6: Best Scores for Chains of Length greater or equal to 10, focusing by score.

Score	Method	Distance	Dim	k (% Noise)	SL	CH	DB	Entropy	Eps	MinPts	Min Clt Size	Min Samples	NN
SL	DBSCAN	Euclidean	300	**2 (0.982)**	**0.974**	3260.839	0.149	0.102	0.630	2.000	-	-	-
	HDBSCAN	Euclidean	300	2 (0.977)	0.743	85.182	0.489	0.216	-	-	3.000	25.000	-
	KMeans	-	300	2	0.151	1.694	0.741	0.002	-	-	-	-	-
CH	DBSCAN	Euclidean	50	**2 (0.982)**	0.971	**4488.032**	0.080	0.103	0.560	2.000	-	-	-
	HDBSCAN	L2 Norm	300	2 (0.973)	0.720	236.593	0.525	0.454	-	-	4.000	22.000	-
	KMeans	-	50	2	0.043	130.748	5.806	0.615	-	-	-	-	-
DB	DBSCAN	Euclidean	50	**2 (0.982)**	0.971	4488.032	**0.080**	0.103	0.550	2.000	-	-	-
	HDBSCAN	Euclidean	300	2 (0.977)	0.743	85.182	0.489	0.216	-	-	3.000	25.000	-
	KMeans	-	300	3	0.146	1.791	0.726	0.004	-	-	-	-	-
Entropy	KMeans	-	300	**2**	0.151	1.694	0.741	**0.002**	-	-	-	-	-
	DBSCAN	Euclidean	300	2 (0.011)	0.370	7.192	0.783	0.003	5.436	2.000	-	-	30.000
	HDBSCAN	Cosine	50	2 (0.146)	0.044	2.010	1.646	0.004	-	-	2.000	34.000	-

As to choosing the best parameters, we found the same trend as the other subsets: best clusters are obtained with smaller number of clusters. Furthermore, save for the CH score, where the 50 dimension vector performs consistently better than the 300 dimensions, for the others scores it happens the other way around.

With regard to DBSCAN, we observed that Euclidean distance is better for every score with 50 dimensional vectors. Yet again, the performance with Cosine similarity obtains the worst values. Notice that this time the points with WMD do not appear and this is due because only one single cluster or none was detected, therefore we did not perform cluster comparison. We can also appreciate the consistent pattern of the performance of `Eps` obtained with heuristics. It is mediocre at best, but generally stable. Also, the smaller neighbors established for `MinPts`, the better.

Finally, with respect to HDBSCAN, the L2-Norm distances with 300 dimensional vectors have the best performance for SL, CH and DB but are somewhat unstable, according to the Entropy. It is also marked how the WMD distance vectors obtain the overall worst performance, although they tend to achieve stable vectors. Moreover, the best minimum Cluster values are

between 2 and 10, according to the SL, CH and DB scores. Furthermore, the better clusters are obtained in a range of 20 to 30 Minimum Samples.

Focusing on the clustered emails, the resulting emails contained mostly few words. According to the labels obtained, the parameters of DBSCAN with the best SL scored are showed in Table 6. We found that for the cluster composed by two emails, those two emails contains a message with a simple `ok` sent as an answer to another email. And said answers belong to different email chains. In fact, the chain with index 131203 belongs to the largest chain length with 798 emails in which users `mike.maggi@enron.com` and `michelle.nelson@enron.com` interchanged several messages with private conversations as if one would expect to see in a modern messaging social network. Moreover, in the largest cluster, there are 25 messages from the chain 131203 and 6 from chain 122336, both existing in the other cluster. These emails can be consulted in Appendix C. Regrettably, none of the clustered emails contained a discernible interaction pattern.

6 Discussion

In Section 5 we described and analysed the results from our proposed pipeline described in Section 4 and illustrated in Figure 9. Hereafter, we present our findings according to said results.

First of all, parting from our methodology, we found that DBSCAN dominated in almost every metric, clustering methods with Euclidean distances performed better according to the selected scores and the preferred size of clusters in all datasets was $k = 2$. Conversely, the Cosine Similarity performed worse than any other metric. WMD did not provided either the best results in terms of cluster quality, and the L2-Norm came close to the Euclidean distance (specially in the case of HDBSCAN), but rarely outperformed the Euclidean distance. This comes as a surprise since L2-Norm is equivalent to the Euclidean Distance. Nonetheless, considering L2-Norm with vectors **x** and **y**

$$||\mathbf{x}||_2 = ||\mathbf{y}||_2 = 1,$$

we know that the squared Euclidean distance is proportional to the cosine distance:

$$\begin{aligned} ||\mathbf{x} - \mathbf{y}||_2^2 &= (\mathbf{x} - \mathbf{y})^\top (\mathbf{x} - \mathbf{y}) \\ &= \mathbf{x}^\top \mathbf{x} - 2\mathbf{x}^\top \mathbf{y} + \mathbf{y}^\top \mathbf{y} \\ &= 2 - 2\mathbf{x}^\top \mathbf{y} \\ &= 2 - 2\cos \angle(\mathbf{x}, \mathbf{y}) \end{aligned}$$

In this sense, even though they are equivalent, we can deduct that observed differences were due to squaring. Furthermore, Cosine's poor performance may be explained by its lack of triangle inequality property, addressed in Section 4.

Notwithstanding, despite WMD's poor performance, during the visual inspection we realised that this metric does detect a humanly-expected relationship of emails, and is even able to detect foreign languages. This opens the possibility that an naive approach in the sense of looking only for cluster quality buries valuable relevant information to the user and the eventual automated system.

Turning our attention to the scores, it is difficult to decide if one score should prevail over another when selecting the best parameter combination. Ultimately, the scores take into consideration different aspects from clusters, which were explained in Section 4. However, while analysing the clustering results, we realised that entropy functioned more like an auxiliary measure rather than a quality measure per se. With regard to the other methods, in the majority of cases HDBSCAN came close to DBSCAN performance, and K-Means did not perform as good as the other methods.

With respect to the visual inspection, the majority of resulting clusters are related to short answers to either formal emails or to personal conversations. There was also a disproportionate allocation of emails in the obtained labels. Notwithstanding, automation opportunities were detected: the most achievable might be a system to respond another emails automatically as if it were for closure. Other options include automatic answering messages, task allocation

or ticket generation. There are companies that are already implementing this kind of tasks. For now, the pipeline developed in this project should be a first step to develop eventually a system that provides automated tasks. Unfortunately, no process was detected with this methodology. In none of the visual inspections of the emails belonging to each data split were found any indication that the clusters detected an existing defined process.

Lastly, working with the Enron emails presented several challenges. In this regard, Listing 1 shows an example of emails of chain length of 2, obtained with DBSCAN with the best SL score. This small chain has a clustered email and a noise email. The latter does not have a consistent formatting and and if we were to apply the same treatment as in the development Section 5, we would lose this information. Most of the reviewed works provided little information on how to deal with the data impurities. One might argue that while pre-processing the emails and converting them into text-representation vectors, one case reduce nise due to Zipfs' Law. However, we consider that in the case of Enron emails, the standard pre-processing is not enough. In fact, until another complete email dataset with state-of-the-art format standars is released, there is a strong necessity to create a methodology to properly pre-process this data set. In this respect, Listing 7 also shows an email with several differences in formats that limit the extraction of information.

Listing 1: Content of the shortest cluster from best SL result of Chain Length greater than 10

```
<Cluster:  0>, Chain ID: 117527, Index: 3599
Date: 1999-11-22 15:37:00
Subject: Re: Neil Mayer
Sender: richard.sanders@enron.com, Recipient:  julia.murray@enron.com
Message: I'll be happy to schedule him. What do you think?


<Cluster:  -1>, Chain ID: 117527, Index: 3804
Date: 1999-12-03 14:27:00
Subject: Re: Neil Mayer
Sender: richard.sanders@enron.com, Recipient:  julia.murray@enron.com
Message: FYI
---------------------- Forwarded by Richard B Sanders/HOU/ECT on 12/03/99
01:25 PM --------------------------

        Enron North America Corp.

        From:  Legal Temp 1                           12/03/99 01:23 PM


Sent by: Mark E Haedicke
To: Richard B Sanders/HOU/ECT@ECT
cc:
Subject: Re: Neil Mayer

Ok to talk to him, I just don't want to get his expectations up.  Mark
```

7 Conclusions

In this project we attempted to propose a system based on document embeddings and three clustering methods to look for opportunities of automation that surges from recurrent patterns in emails. One of the main motivations for this line of research is that employees in companies spend around 28 % of their time reading and answering emails [20, p. 46]. Finding automation opportunities can reduce budget spending, increase job performance and improve welfare.

To pursue our objective, we decided to follow a feature based approach in terms of email mining. We also compared said clustering algorithms combining distance matrices and splitting data into manageable partitions, performed cluster evaluation quality using several scores that do not require ground-truth assumptions, and explored the feasibility of using unsupervised methods to group together email chains or detect interactions between emails.

Specifically, we described the three clustering methods, K-Means, DBSCAN and HDBSCAN, and outlined the advantages and disadvantages of each method. We also defined the distance matrices, the document text representation through Paragraph Vectors (also known as doc2vec), and obtained distance matrices such as the Word Mover's distance. We also proposed a system based on the CRISP-DM methodology to integrate industry's best practices into our developed process pipeline.

Unfortunately, with respect to feasibility, our results suggested that the proposed system is not capable of detecting a clear process from the analyzed emails. We did find that all used scores coincide that two clusters were the best grouping that the clustering algorithms could get. While doing visual inspection, we also noticed that in cases with clustering using WMD there were group emails written in different languages.

Resuming the tasks that comprises the email mining field, our results were more approachable for Automatic Email Answering, since most of the clustered emails seemed to be short answers to another incoming message. However, following the idea of Bickel and Scheffer [7], it is necessary to establish ground truth labels in order to make an effective automatic system.

Another possibility for future research is to combine Unsupervised Machine Learning with a social network base approach. Specially with HDBSCAN since the method already performs graph-based operations, it should be possible to find patterns in the sense of detecting automatically organizational structure or group contacts.

As a final note, Enron emails dataset offers a rich opportunity to study emails and message interaction. However it suffers from inconsistencies and impurities that affect the investigation. Considering that there is a Hierarchical Gold Standard, there should also be a Golden Standard for Enron pre-processing and extraction.

8 References

[1] Apoorv Agarwal et al. “A comprehensive gold standard for the enron organizational hierarchy”. In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 2012, pp. 161–165.

[2] Daniel Aloise et al. “NP-hardness of Euclidean sum-of-squares clustering”. In: *Machine learning* 75.2 (2009), pp. 245–248.

[3] Alexandr Andoni and Piotr Indyk. “Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions”. In: *Communications of the ACM* 51.1 (2008), pp. 117–122.

[4] David Arthur and Sergei Vassilvitskii. “How slow is the k-means method?” In: *Proceedings of the twenty-second annual symposium on Computational geometry*. 2006, pp. 144–153.

[5] Sterling Baird and Taylor Sparks. *Distance Matrices*. 2021. url: https://github.com/sparks-baird/dist-matrix.

[6] Victoria Bellotti et al. “Quality versus quantity: E-mail-centric task management and its relation with overload”. In: *Human–Computer Interaction* 20.1-2 (2005), pp. 89–138.

[7] Steffen Bickel and Tobias Scheffer. “Learning from message pairs for automatic email answering”. In: *European Conference on Machine Learning*. Springer. 2004, pp. 87–98.

[8] T. Caliński and J Harabasz. “A dendrite method for cluster analysis”. In: *Communications in Statistics* 3.1 (1974), pp. 1–27. doi: 10.1080/03610927408827101. eprint: https://www.tandfonline.com/doi/pdf/10.1080/03610927408827101. url: https://www.tandfonline.com/doi/abs/10.1080/03610927408827101.

[9] Christopher S Campbell et al. “Expertise identification using email communications”. In: *Proceedings of the twelfth international conference on Information and knowledge management*. 2003, pp. 528–531.

[10] Laura A Dabbish and Robert E Kraut. “Email overload at work: An analysis of factors associated with email strain”. In: *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*. 2006, pp. 431–440.

[11] David L. Davies and Donald W. Bouldin. “A Cluster Separation Measure”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-1.2 (1979), pp. 224–227. doi: 10.1109/TPAMI.1979.4766909.

[12] “Determination of Optimal Epsilon (Eps) Value on DBSCAN Algorithm to Clustering Data on Peatland Hotspots in Sumatra”. In: *IOP Conference Series: Earth and Environmental Science* 31 (Jan. 2016), p. 012012. doi: 10.1088/1755-1315/31/1/012012. url: https://doi.org/10.1088/1755-1315/31/1/012012.

[13] Jana Diesner, Terrill L Frantz, and Kathleen M Carley. “Communication networks from the Enron email corpus “It’s always about the people. Enron is no different””. In: *Computational & Mathematical Organization Theory* 11.3 (2005), pp. 201–228.

[14] Justin Eldridge, Mikhail Belkin, and Yusu Wang. “Beyond hartigan consistency: Merge distortion metric for hierarchical clustering”. In: *Conference on Learning Theory*. PMLR. 2015, pp. 588–606.

[15] Martin Ester et al. “A density-based algorithm for discovering clusters in large spatial databases with noise.” In: *kdd*. Vol. 96. 34. 1996, pp. 226–231.

[16] Ronald A Fisher. “The use of multiple measurements in taxonomic problems”. In: *Annals of Eugenics* 7.2 (1936), pp. 179–188. doi: https://doi.org/10.1111/j.1469-1809.1936.tb02137.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1469-1809.1936.tb02137.x. url: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-1809.1936.tb02137.x.

[17] Felienne Hermans and Emerson Murphy-Hill. “Enron’s spreadsheets and related emails: A dataset and analysis”. In: *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*. Vol. 2. IEEE. 2015, pp. 7–16.

[18] Sungjin Im et al. “Fast noise removal for k-means clustering”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 456–466.

[19] Nucleus Research Inc. *SPAM: The Repeat Offender*. 2003. url: https://nucleusresearch.com/wp-content/uploads/2018/05/h22-Spam-the-repeat-offender.360.pdf.

[20] Mckinsey Global Institute. *The social economy: Unlocking value and productivity through social technologies*. 2012. url: https://www.mckinsey.com/~/media/mckinsey/industries/technology%5C%20media%5C%20and%5C%20telecommunications/high%5C%20tech/our%5C%20insights/the%5C%20social%5C%20economy/mgi_the_social_economy_full_report.pdf.

[21] Abhishek Kathuria, Devarshi Mukhopadhyay, and Narina Thakur. “Evaluating cohesion score with email clustering”. In: *Proceedings of First International Conference on Computing, Communications, and Cyber-Security (IC4S 2019)*. Springer. 2020, pp. 107–119.

[22] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, 2009.

[23] Parambir S Keila and David B Skillicorn. “Structure in the Enron email dataset”. In: *Computational & Mathematical Organization Theory* 11.3 (2005), pp. 183–199.

[24] Bryan Klimt and Yiming Yang. “The Enron Corpus: A New Dataset for Email Classification Research”. In: *Machine Learning: ECML 2004*. Ed. by Jean-François Boulicaut et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 217–226. isbn: 978-3-540-30115-8.

[25] Matt J. Kusner et al. “From Word Embeddings to Document Distances”. In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*. ICML’15. Lille, France: JMLR.org, 2015, pp. 957–966.

[26] Siu Kwan Lam, Antoine Pitrou, and Stanley Seibert. “Numba: A LLVM-Based Python JIT Compiler”. In: *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*. LLVM ’15. Austin, Texas: Association for Computing Machinery, 2015. isbn: 9781450340052. doi: 10.1145/2833157.2833162. url: https://doi.org/10.1145/2833157.2833162.

[27] Quoc Le and Tomas Mikolov. “Distributed representations of sentences and documents”. In: *International conference on machine learning*. PMLR. 2014, pp. 1188–1196.

[28] Leland McInnes, John Healy, and Steve Astels. “hdbscan: Hierarchical density based clustering”. In: *The Journal of Open Source Software* 2.11 (Mar. 2017). doi: 10.21105/joss.00205. url: https://doi.org/10.21105%2Fjoss.00205.

[29] Tomas Mikolov et al. “Distributed representations of words and phrases and their compositionality”. In: *Advances in neural information processing systems* 26 (2013).

[30] Tomas Mikolov et al. “Efficient estimation of word representations in vector space”. In: *arXiv preprint arXiv:1301.3781* (2013).

[31] David Ndumiyana, Munyaradzi Magomelo, and Lucy Sakala. “Spam detection using a neural network classifier”. In: (2013).

[32] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[33] Radim Řehůřek and Petr Sojka. “Software Framework for Topic Modelling with Large Corpora”. English. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. http://is.muni.cz/publication/884893/en. Valletta, Malta: ELRA, May 2010, pp. 45–50.

[34] Andrew Rosenberg and Julia Hirschberg. “V-measure: A conditional entropy-based external cluster evaluation measure”. In: *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*. 2007, pp. 410–420.

[35] Peter J. Rousseeuw. “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis”. In: *Journal of Computational and Applied Mathematics* 20 (1987), pp. 53–65. issn: 0377-0427. doi: https://doi.org/10.1016/0377-0427(87)90125-7. url: https://www.sciencedirect.com/science/article/pii/0377042787901257.

[36] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. “A metric for distributions with applications to image databases”. In: *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*. IEEE. 1998, pp. 59–66.

[37] Gerard Salton and Michael J McGill. *Introduction to modern information retrieval*. mcgraw-hill, 1983.

[38] Ryoma Sato, Makoto Yamada, and Hisashi Kashima. *Re-evaluating Word Mover’s Distance*. 2021. doi: 10.48550/ARXIV.2105.14403. url: https://arxiv.org/abs/2105.14403.

[39] Erich Schubert et al. “DBSCAN revisited, revisited: why and how you should (still) use DBSCAN”. In: *ACM Transactions on Database Systems (TODS)* 42.3 (2017), pp. 1–21.

[40] C. E. Shannon. “A mathematical theory of communication”. In: *The Bell System Technical Journal* 27.3 (1948), pp. 379–423. doi: 10.1002/j.1538-7305.1948.tb01338.x.

[41] Jitesh Shetty and Jaffar Adibi. “Ex employee status report”. In: *Retrieved November* 4 (2004).

[42] Guanting Tang, Jian Pei, and Wo-Shun Luk. “Email mining: Tasks, common techniques, and tools”. In: *Knowledge and Information Systems* 41 (Oct. 2013). doi: 10.1007/s10115-013-0658-2.

[43] RAPIDS Development Team. *RAPIDS: Collection of Libraries for End to End GPU Data Science*. 2018. url: https://rapids.ai.

[44] Athena Vakali. "Web Data Management Practices: Emerging Techniques and Technologies: Emerging Techniques and Technologies". In: (2006).

[45] C Van Rijsbergen. "Information retrieval: theory and practice". In: *Proceedings of the Joint IBM/University of Newcastle upon Tyne Seminar on Data Base Systems*. Vol. 79. 1979.

[46] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. doi: 10.1038/s41592-019-0686-2.

[47] Zehui Wang et al. "Analysis of Instagram Users' Movement Pattern by Cluster Analysis and Association Rule Mining". In: *ENTER22 e-Tourism Conference*. Springer. 2022, pp. 97–109.

[48] Rüdiger Wirth. "CRISP-DM: Towards a standard process model for data mining". In: *Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining*. 2000, pp. 29–39.

[49] Laurence A Wolsey and George L Nemhauser. *Integer and combinatorial optimization*. Vol. 55. John Wiley & Sons, 1999.

[50] Hongteng Xu et al. "Distilled Wasserstein Learning for Word Embedding and Topic Modeling". In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. NIPS'18. Montréal, Canada: Curran Associates Inc., 2018, pp. 1723–1732.

A Cluster Distribution

This Appendix presents the cluster distribution of emails according to the best scores obtained for each of the clusters in all four data splits.

Table 7: Chains of Length 2 Emails Distribution by Score and Method.

	Counts											
Score	CH			DB			Entropy			SL		
k	DBSCAN	HDBSCAN	K-Means	DBSCAN	HDBSCAN	K-Means	DBSCAN	HDBSCAN	K-Means	DBSCAN	HDBSCAN	K-Means
-1	9529	9535		9535	9534		278	179		2	9535	
0	139	2	4639	139	140	5	9397	9495	3868	9674	2	3868
1	8	139	5037	2	2	4621	1	2	5808		139	5808
2						357						
3						338						
4						1						
5						9						
6						7						
7						2370						
8						1789						
9						2						
10						6						
11						1						
12						8						
13						5						
14						1						
15						10						
16						145						
17						1						

Table 8: Chains of Length 3 Emails Distribution by Score and Method.

	Counts											
Score	CH			DB			Entropy			SL		
k	DBSCAN	HDBSCAN	K-Means	DBSCAN	HDBSCAN	K-Means	DBSCAN	HDBSCAN	K-Means	DBSCAN	HDBSCAN	K-Means
-1	8639	8344			8632		863	411			8632	
0	126	6	3615	8766	135	1	7905	8356	3615	8766	135	3615
1	4	2	5154	3	2	4168	1	2	5154	3	2	5154
2		415				1						
3		2				1						
4						1						
5						3						
6						557						
7						3						
8						3						
9						2						
10						1700						
11						2						
12						2						
13						1						
14						1812						
15						1						
16						1						
17						3						
18						507						

Table 9: Chains of Length greater or equal to 4 and less than 10 Emails Distribution by Score and Method.

	Counts											
Score	CH			DB			Entropy			SL		
k	DBSCAN	HDBSCAN	K-Means	DBSCAN	HDBSCAN	K-Means	DBSCAN	HDBSCAN	K-Means	DBSCAN	HDBSCAN	K-Means
-1	9529	9535		9535	9534		278	179		2	9535	
0	139	2	4639	139	140	5	9397	9495	3868	9674	2	3868
1	8	139	5037	2	2	4621	1	2	5808		139	5808
2						357						
3						338						
4						1						
5						9						
6						7						
7						2370						
8						1789						
9						2						
10						6						
11						1						
12						8						
13						5						
14						1						
15						10						
16						145						
17						1						

Table 10: Chains of Length greater or equal to 10 Emails Distribution by Score and Method.

	Counts											
Score	CH			DB			Entropy			SL		
k	DBSCAN	HDBSCAN	K-Means	DBSCAN	HDBSCAN	K-Means	DBSCAN	HDBSCAN	K-Means	DBSCAN	HDBSCAN	K-Means
-1	5230	5182		5230	5199		60	777		5229	5199	
0	92	118	3698	92	7	5322	5262	4545	1	93	7	1
1	2	24	1626	2	118	1	2	2	5323	2	118	5323
2						1						

B Best Scores

This Appendix contains the Figures related to the best results divided by each of the data splits, showing their respective parameters.

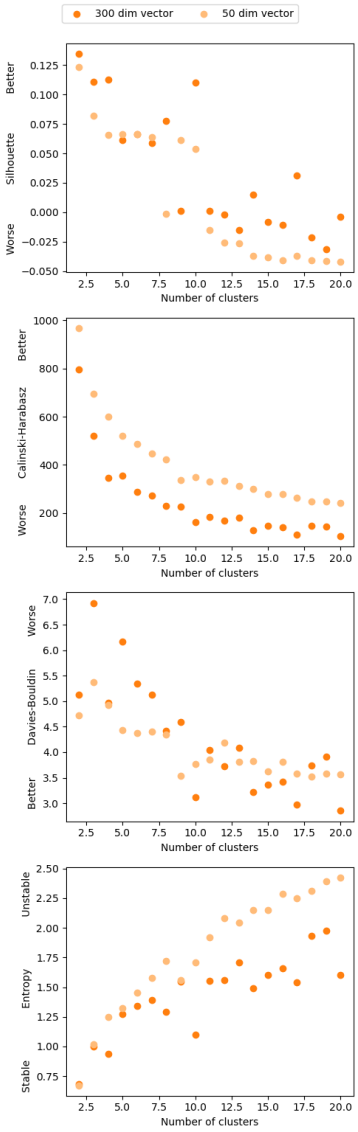


Figure 12: Score results for emails of Length Chain of 2 using K Means

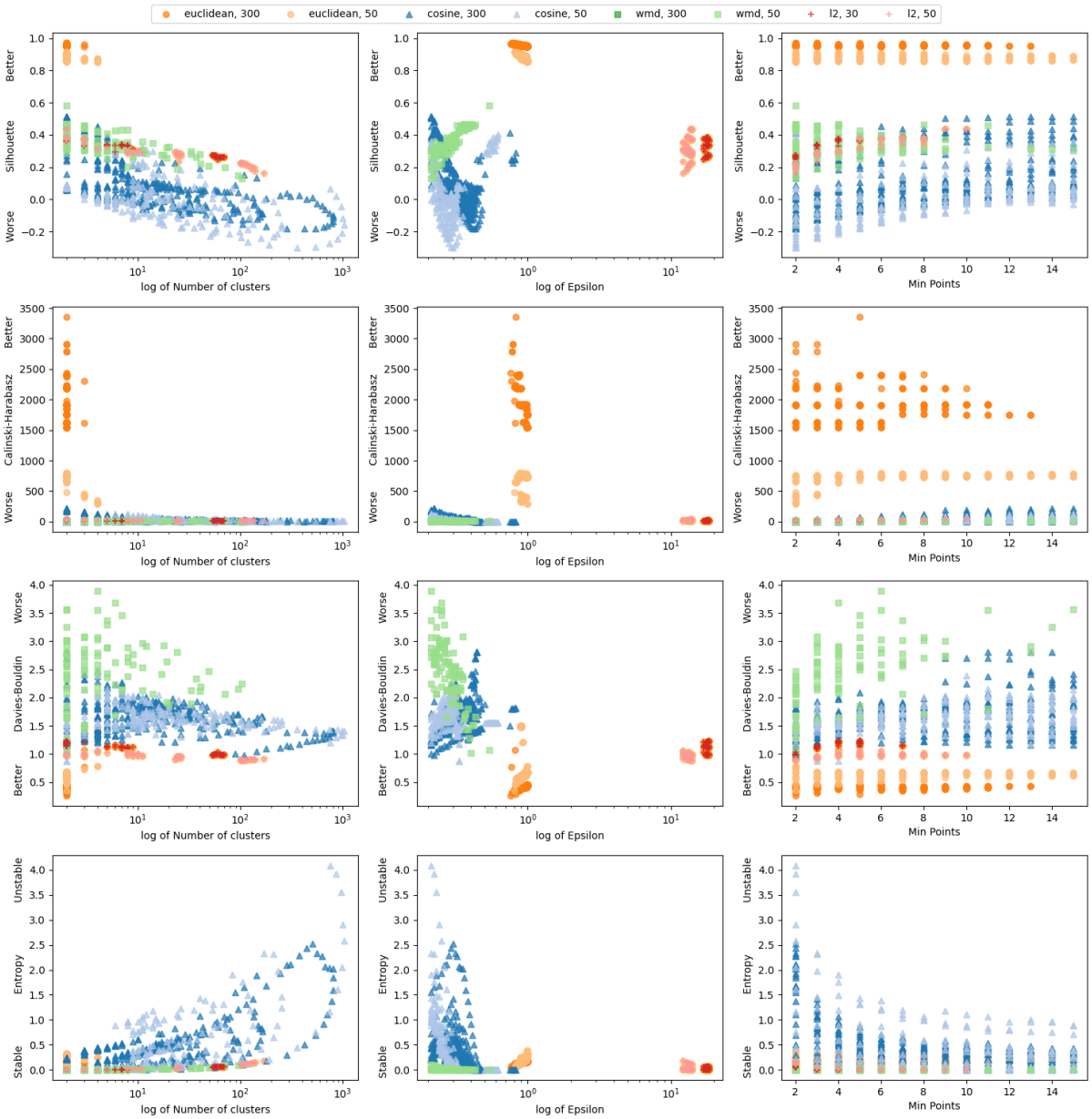


Figure 13: Score results for emails of Length Chain of 2 using DBSCAN

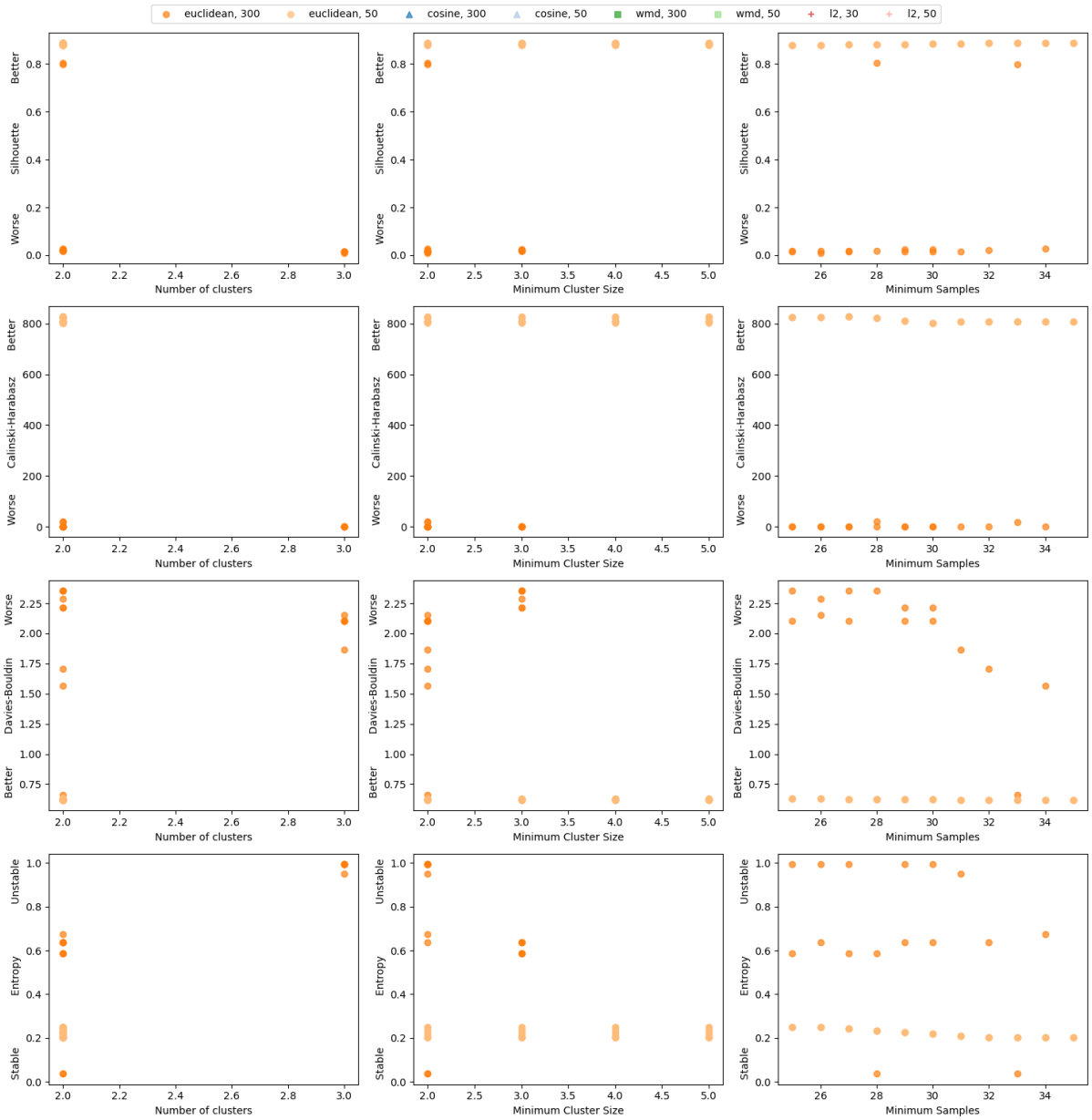


Figure 14: Score results for emails of Length Chain of 2 using HDBSCAN

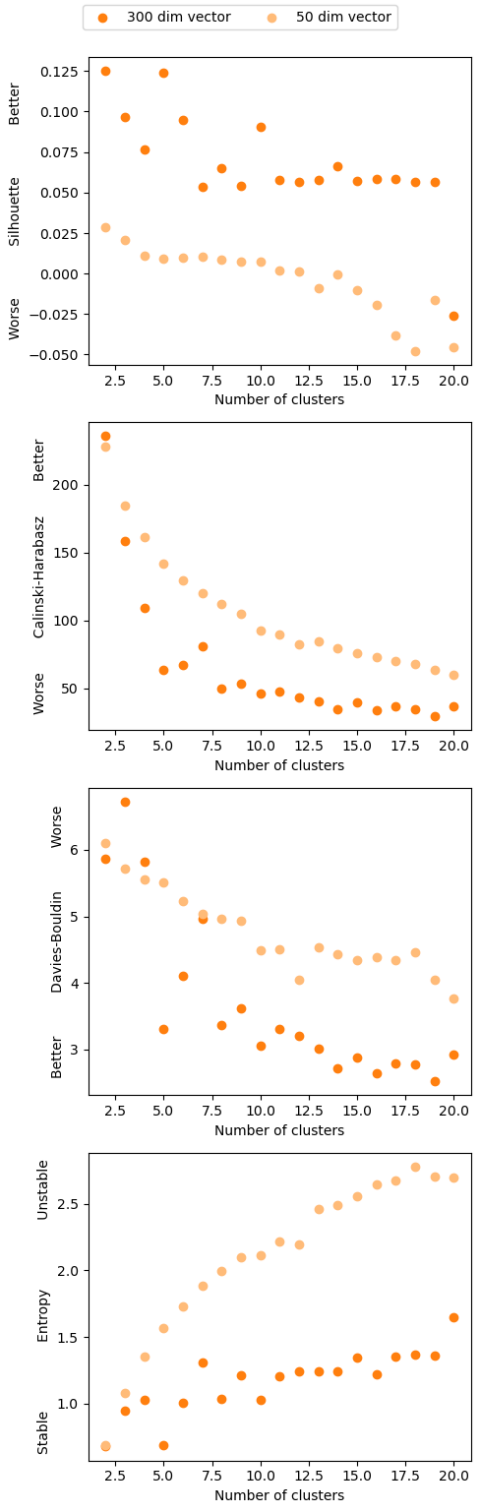


Figure 15: Score results for emails of Length Chain of 3 using K Means

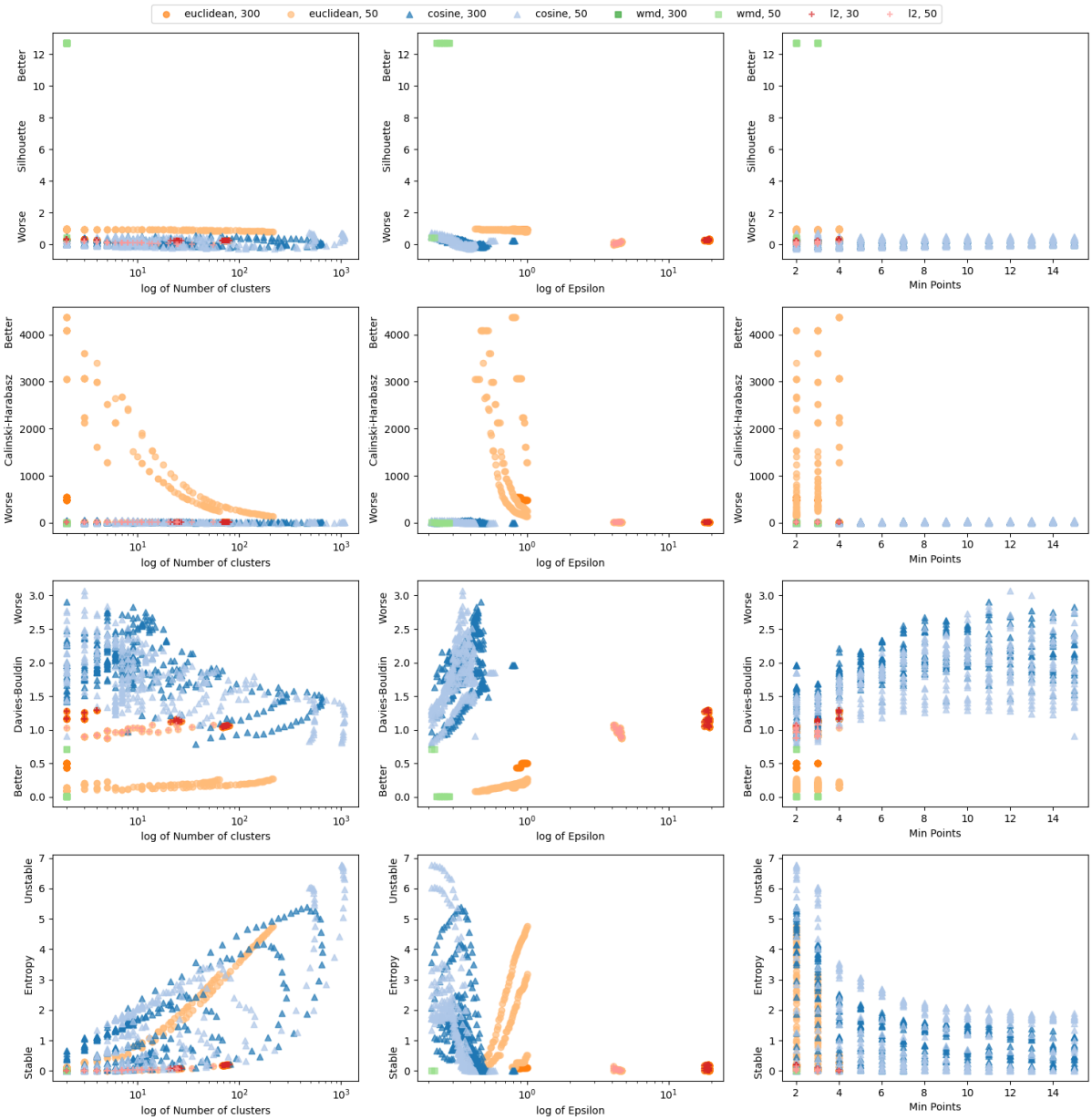


Figure 16: Score results for emails of Length Chain of 3 using DBSCAN

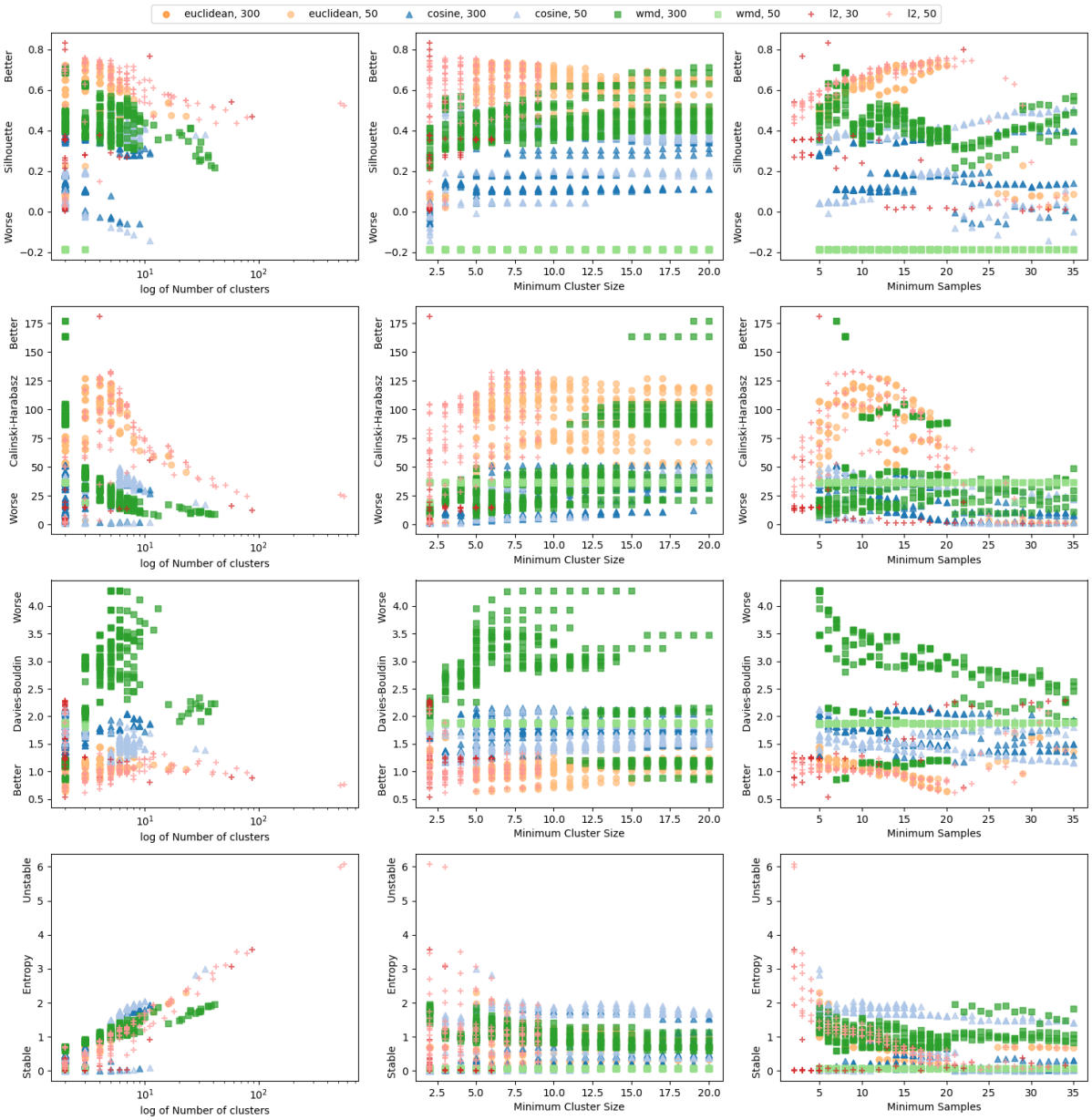


Figure 17: Score results for emails of Length Chain of 3 using HDBSCAN

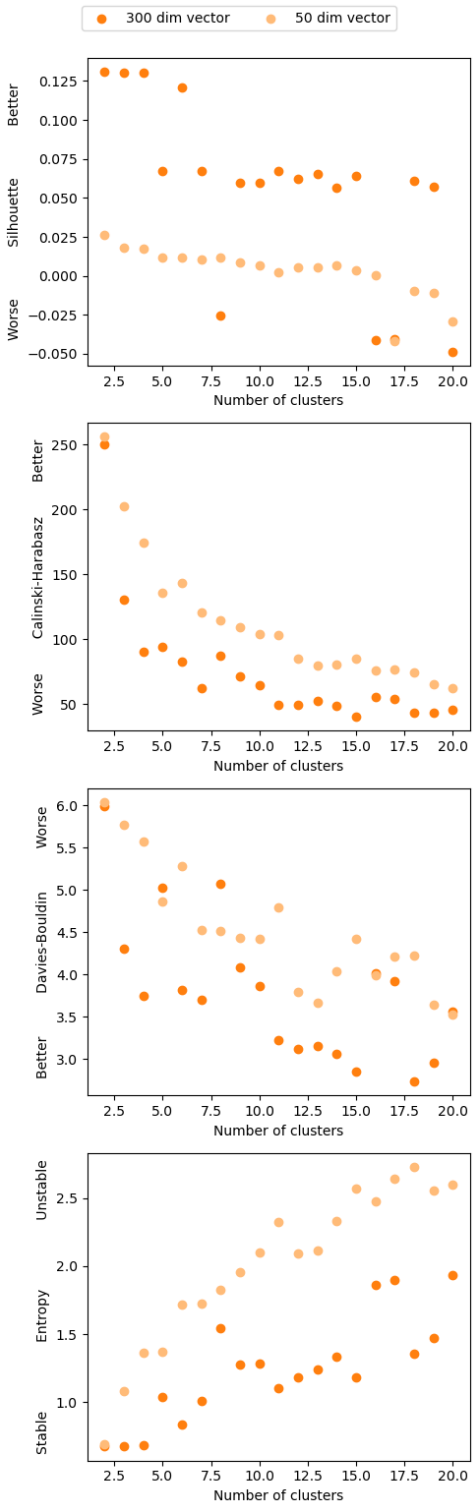


Figure 18: Score results for emails of Length Chain greater than 4 and less than 10 using K Means

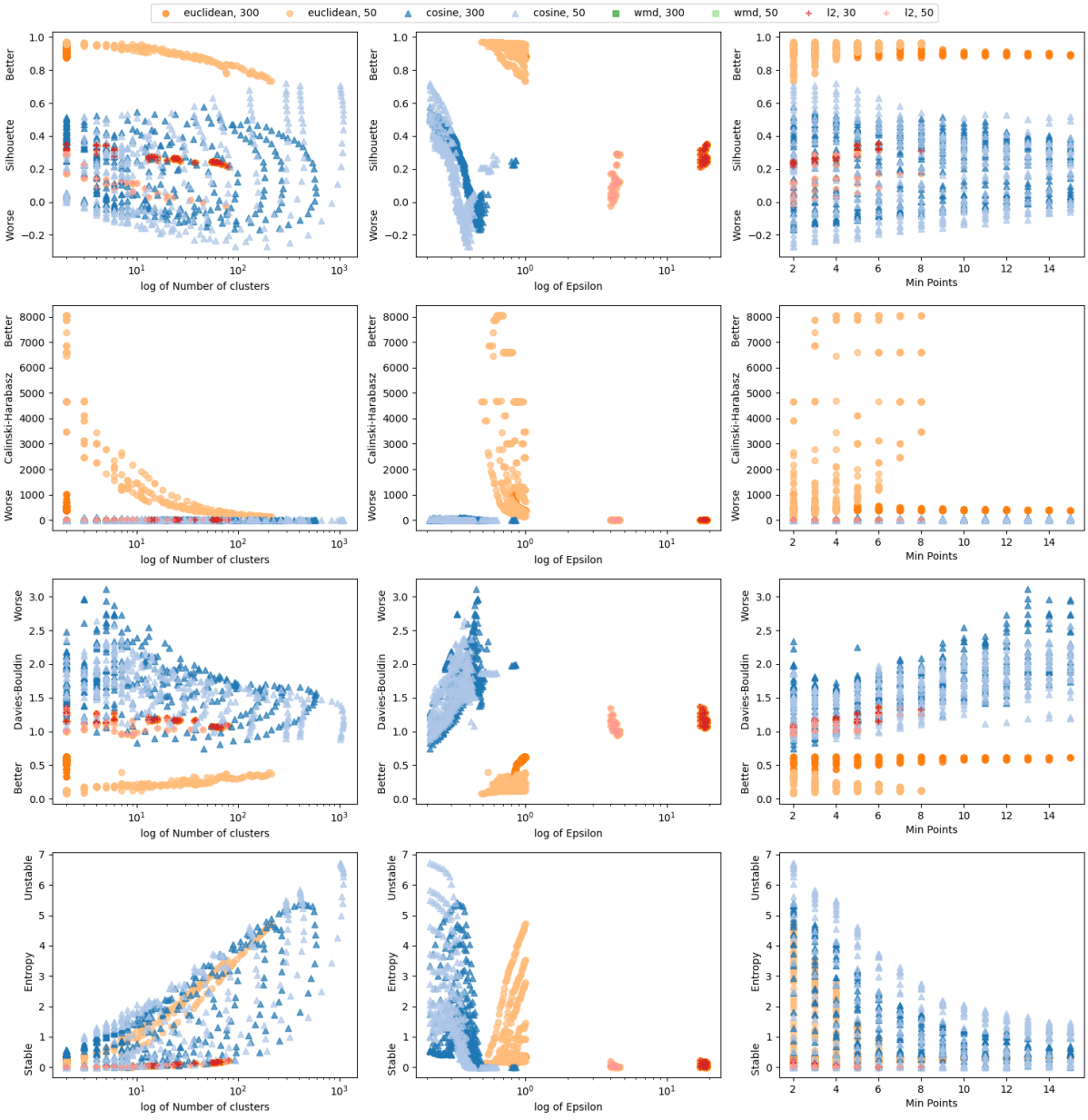


Figure 19: Score results for emails of Length Chain greater than 4 and less than 10 using DBSCAN

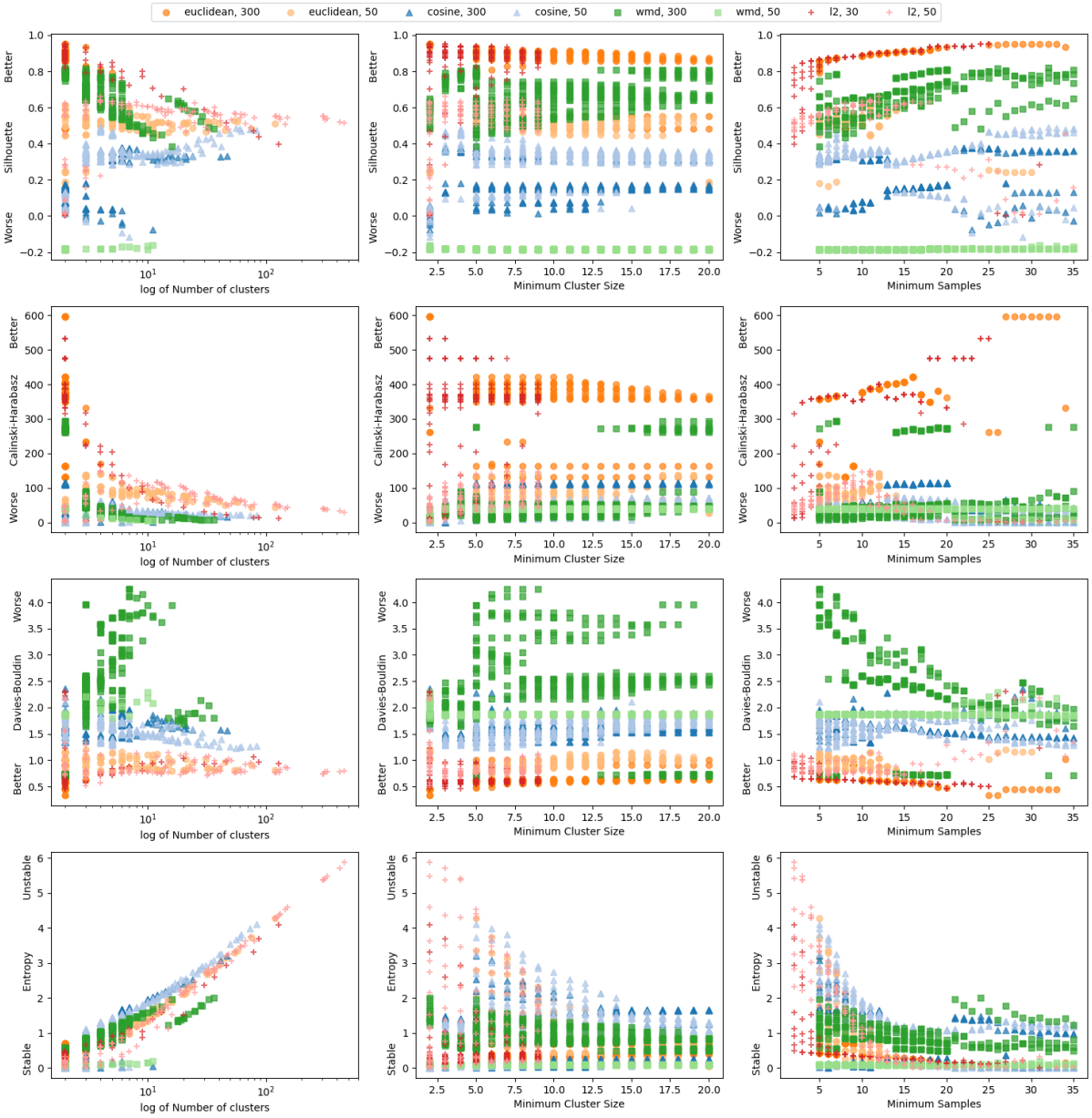


Figure 20: Score results for emails of Length Chain greater than 4 and less than 10 using HDBSCAN

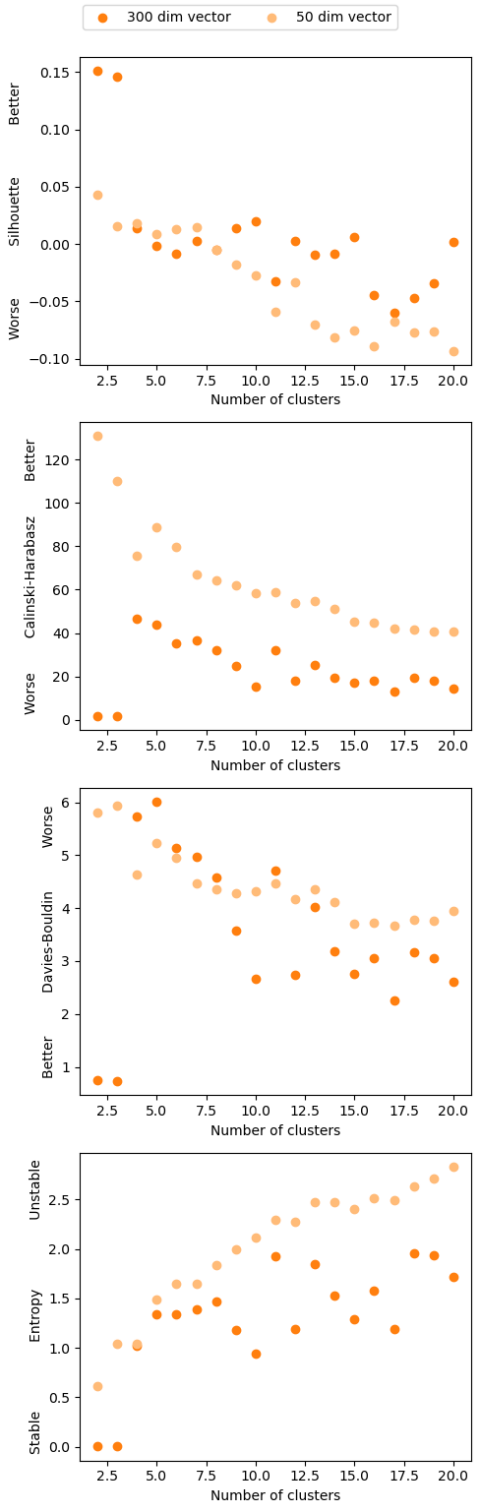


Figure 21: Score results for emails of Length Chain greater or equal to 10 using K-Means

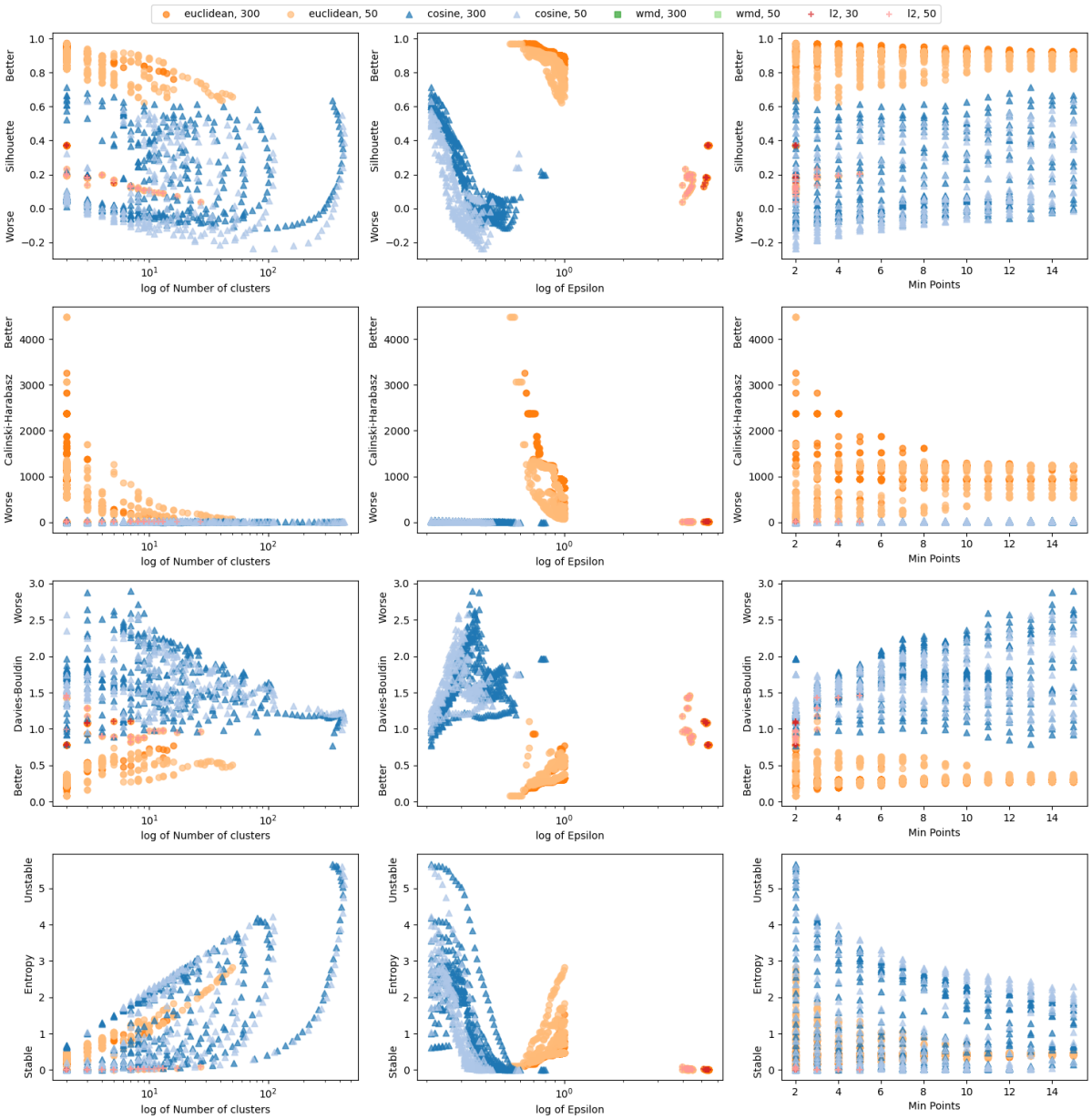


Figure 22: Score results for emails of Length Chain greater than 10 using DBSCAN

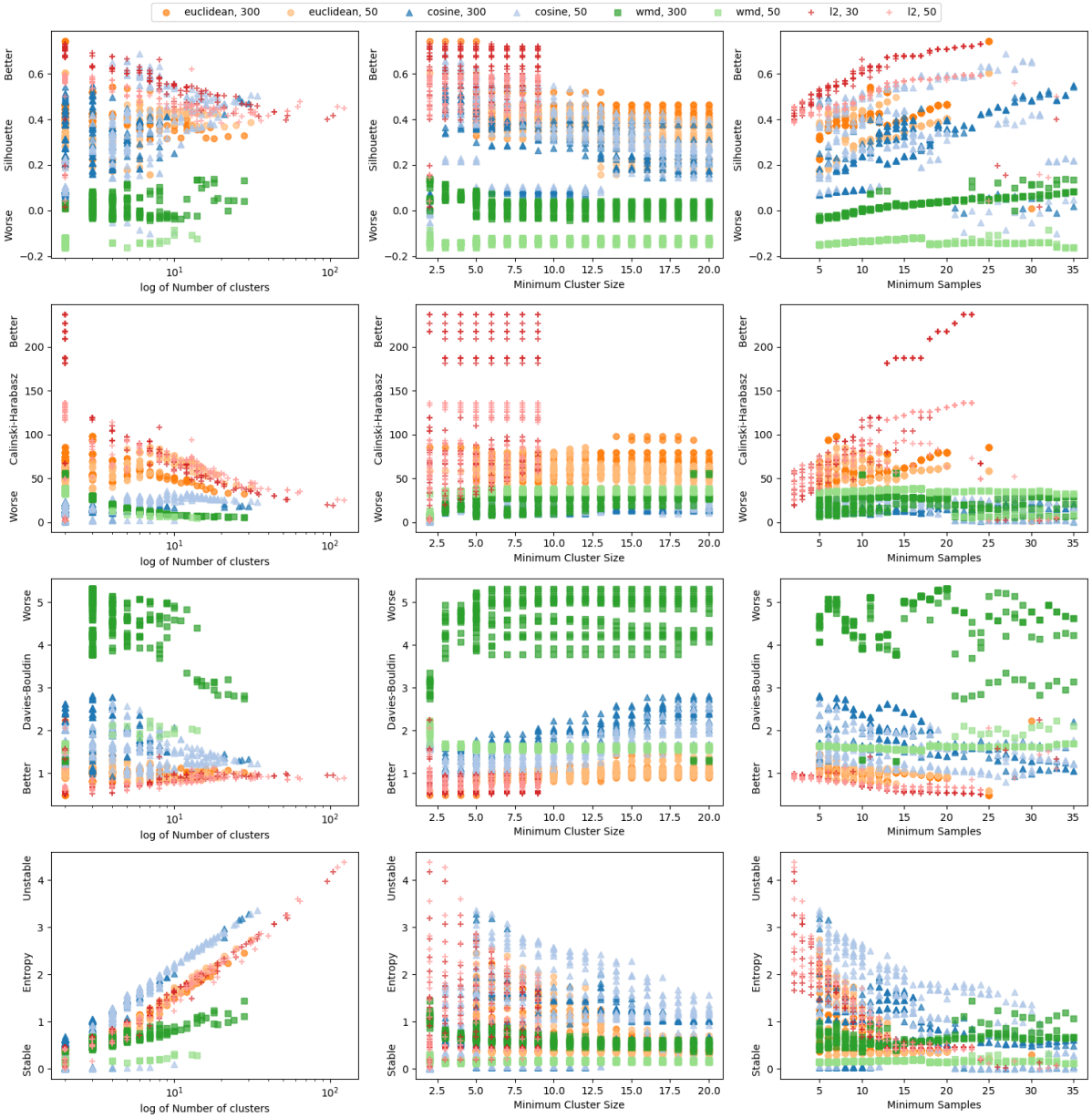


Figure 23: Score results for emails of Length Chain greater than 10 using HDBSCAN

C Emails Samples

This Appendix contains samples of email chains that are taken as example for how the algorithms labeled this emails. We can observe some cases were all emails were not allocated in a single chain. There are other cases were there were located in a single group foreign language emails.

Listing 2: Content of the shortest cluster from best DB result of Chain of Length 2

```
# Example 1
<Cluster: 1>, Chain ID: 037831, Index: 5471
Date: 2000-01-11 11:13:00
Subject: Congrats
Sender: vince.kaminski@enron.com, Recipient: laura.luce@enron.com
Message: Laura,

Congratulations. Well deserved.

Vince


<Cluster: -1>, Chain ID: 037831, Index: 5480
Date: 2000-01-11 11:50:00
Subject: Re: Congrats
Sender: laura.luce@enron.com, Recipient: vince.kaminski@enron.com
Message: Vince,

You beat me to the congrats.  The surprise was that I already believed you
were a Managing Director, so a long overdue congratulations to you.

LauraVince J Kaminski@ECT
01/11/2000 10:13 AM
To: Laura Luce/HOU/ECT@ECT
cc:
Subject: Congrats

Laura,

Congratulations. Well deserved.

Vince

# Example 2
<Cluster: 1>, Chain ID: 150121, Index: 178636
Date: 2001-10-23 18:05:03
Subject: RE: QF Presentation
Sender: jeff.dasovich@enron.com, Recipient: michael.etringer@enron.com
Message: thanks.

<Cluster: 1>, Chain ID: 150121, Index: 179525
Date: 2001-10-24 00:23:47
Subject: FW: QF Presentation
Sender: jeff.dasovich@enron.com, Recipient: michael.etringer@enron.com
Message: FYI.
```

```
<Cluster: 1>, Chain ID: 044392, Index: 182632
Date: 2001-10-25 19:39:52
Subject: RE: Dave Dronet's Birthday
Sender: vladi.pimenov@enron.com, Recipient: esdronet@hastingsmail.alief.isd.tenet.
    edu
Message: We're in!!!!

<Cluster: 1>, Chain ID: 044392, Index: 182952
Date: 2001-10-25 21:27:41
Subject: RE: Dave Dronet's Birthday
Sender: esdronet@hastingsmail.alief.isd.tenet.edu, Recipient: vladi.pimenov@enron.
    com
Message: Excellent!

>
```

Listing 3: Content of the shortest cluster from best SL result of Chain of Length 2

```
<Cluster: -1>, Chain ID: 209486, Index: 782
Date: 1999-06-03 17:41:00
Subject: derivatives documentation software
Sender: tana.jones@enron.com, Recipient: ian.howells@documentum.com
Message: I have been given the name of your company by a consultant we have hired
    to
advise us on sofware systems to manage our physical and financial
confirmation process and ISDA Master Agreements.  I have also seen
information on your company in the January 1999 issue of Risk Magazine.  I am
a Senior Legal Specialist in the Legal Dept. and am looking at software
systems that can help us manage our documentation needs.

If you are unfamiliar with our company,  Enron is the largest integrated
marketer of energy in the United States.  We have a Web Page located at
www.enron.com.  Enron Capital & Trade Resources Corp. is the marketing
affiliate of our parent, Enron Corp.

I would like to obtain marketing material about the services you provide, and
after I have a chance to look at the information, would like to talk to you
or a representative from you company in some depth.  I am located in Houston,
Texas and my phone number is (713) 853-3399.  Thank your for your attention
herewith.

<Cluster: 0>, Chain ID: 209486, Index: 784
Date: 1999-06-04 09:49:00
Subject: RE: derivatives documentation software
Sender: tana.jones@enron.com, Recipient: ian.howells@documentum.com
Message: Thank you for your help.  I look forward to hearing from your U.S. person.
```

Listing 4: Language related email from best SL result of Chain of Length 3 and WMD

```
<Cluster: 0>, Chain ID: 071164, Index: 173224
Date: 2001-10-18 17:35:57
Subject: RE: FYI - LNG Terminal in California - using Bolivina gas
Sender: richard.shapiro@enron.com, Recipient: jose.bestard@enron.com
Message: What's our potential involvement?
```

<Cluster: 1>, Chain ID: 071164, Index: 173501
Date: 2001-10-18 19:52:41
Subject: FYI - LNG Terminal in California - using Bolivina gas
Sender: jose.bestard@enron.com, Recipient: richard.shapiro@enron.com
Message: This news item involves a project to export Bolivian gas through Chile or Peru to California. Proyecto Pacific LNG ser? presentado en California

http://energypress.com/cgi-bin/npublisher/extras/viewnews.cgi?category=1&id=1003264703

Este lunes 15 de octubre, una delegaci?n boliviana estar? en California para realizar el mismo trabajo que hizo hace semanas atr?s, con M?xico: presentar el proyecto oficialmente a las autoridades californianas y a las empresas petroleras privadas para su consideraci?n.

Los representantes de California Power y de la Compa??a Sempra Energy est?n esperando a la delegaci?n boliviana que dirige el ministro de Desarrollo Econ?mico, Carlos Kempff. Se sumar? tambi?n el secretario del Estado de California, Bill Jones interesado en escuchar la propuesta nacional para este mega-emprendimiento que costar? alrededor de 6 millones de d?la! re! s en su fase inicial, desde el pr?ximo a?o.

"Pese a los conflictos b?licos, los proyectos grandes de nuestro pa?s siguen adelante. Estar? esta semana en California para sostener una reuni?n importante para poder venderle nuestro gas a esta regi?n que est? atravesando una crisis energ?tica y que precisa comprar gas del exterior", explic? Carlos Kempff antes de su partida a Los Estados Unidos. Su agenda incluye otras reuniones de tipo comercial en otros estados norteamericanos.

Con la visita al estado de California, el proyecto Pacific LNG, el gobierno concluir? su fase de aproximaci?n a los mercados potenciales que tiene el gas boliviano para dar paso al inter?s y la decisi?n de Estados Unidos y M?xico, con probabilidades positivas. El gobierno sabe que no es el ?nico interesado en la exportaci?n de este energ?tico hacia los pa?ses del norte (M?xico y EE.UU.). Tras de ?l, est?n otros ofertantes que esperan la apertura de estos grandes mercados. El plan de factibilidad para la realizaci?n de Pacific LNG de las empresas privadas en Bolivia sigue adelante. Ellos esperan que en los primeros meses del pr?ximo a?o, haya ya una real dimensi?n de lo que puedan realizar para iniciar la construcci?n de gasoductos licuefacci?n del gas para el transporte mar?timo hasta las riberas de ambos pa?ses demandantes.

<Cluster: 0>, Chain ID: 071164, Index: 173766
Date: 2001-10-18 22:05:48
Subject: RE: FYI - LNG Terminal in California - using Bolivina gas
Sender: jose.bestard@enron.com, Recipient: richard.shapiro@enron.com
Message: Rick.

We have NOT pursued this aspect from South America. I do not know if there is anyone in North America looking at this, at least from the perspective of competitive sourcing to California. This project is in gestation and will take a few years of development, if it is launched. British Gas is one of the movers in Bolivia, because they DO NOT want to remain dependent exclusively on to the Brazilian market (and the hold of Petrobras).

```
JoseFrom:       Richard Shapiro/ENRON@enronXgate on 10/18/2001 10:35 AM
To:     Jose Bestard/ENRON_DEVELOPMENT@ENRON_DEVELOPMENT
cc:

Subject:        RE: FYI - LNG Terminal in California - using Bolivina gas

What's our potential involvement?
```

Listing 5: Email in foreign language

```
##
Labelled from best SL result of Chain of Length 3 with WMD
##
<Cluster: 0>, Chain ID: 131320, Index: 195238
Date: 2001-11-12 17:37:30
Subject: RE:
Sender: nshand@condenast.co.uk, Recipient: vladi.pimenov@enron.com
Message: Uzh mog bi i cherkanut paru slov o poezdke na Rodinu!!!

<Cluster: 1>, Chain ID: 131320, Index: 195269
Date: 2001-11-12 17:58:02
Subject: RE:
Sender: vladi.pimenov@enron.com, Recipient: nshand@condenast.co.uk
Message: nu znachit delo bilo tak:

priexal, potomkazhdiy den' nosilsia po vsey Moscve, so vsemi nado vstretitsia,
    kazhdiy vecher s kem-nibud' poiti vipit' i zakusit'. Ne uspel dazhe vse dela
    sdelat' (pasport pomeniat', naprimer).
S Taney dazhe ne udalos' uvidetsia.
Xodili odin raz v Tarasa Bul'bu, posideli tam kak sleduet, potom eshe raz u
    Grishenki bili i vse.
Pogoda v Moscve gadkaia, letet' daleko (nazad bol'she sutok dobiralis'). V obshem,
    v blizhayshem budushem tuda vriad li poedu.

Vot takie dela.

Sorry chto ne pisal, kak tol'ko priexal tut nachalsia v kompanii durdom, chut' za 2
     nedeli ne obankrotilis'. Zakonchilos' vse tem, chto nashu kompaniu kupila v
    piatnitsu drugaia kompania. Tak chto vse zhdut sokrasheniy, posmotrim kak eto
    vse budet.

Vot takie dela.

Davay tozhe pishi, ne zatiagivay.

bratik Vladik.

<Cluster: 1>, Chain ID: 131320, Index: 195312
Date: 2001-11-12 18:26:43
Subject: RE:
Sender: vladi.pimenov@enron.com, Recipient: nshand@condenast.co.uk
Message: V obshem ,vse po poriadku.

Kogda priexal, uzhe na vtoroy takoe bilo vpechatlenie, chto nikuda i ne uezhal, vse
     blin tozhe samoe, vse tol'ko rabotu pomeniali, a tak v printsipe odna i ta zhe
```

```
    fignia. U menia stol'ko za poslednie 2.5 goda proizoshlo, chto I ozhidal chto i
    u vsex takie zhe ogromnie izmenenia. No vse v tselom po staromu, chto v
    printsipe grustno. Narod poxozhe zaela obidennost' i rutinka.
No v tselom situatsia poluche chem I uezhal, s rabotoy i den'gami I ponial problem
    bol'shix net, chto ne mozhet ne radovat'.

S Mashkoy i Azarovoy I kak bi voobshe i ne planiroval vstrechat'sia, s Taney xotel,
     no ne poluchilos', I ey zvonil paru raz, nikto ne otvechal.
S Shipitsinoy tol'ko raz posle priezda razgovarival, u nee vrode vse normal'no,
    mozhet pereedet na zimu vo Floridu, tam teplee.

Naschet ostavat'sia ili uezhat' : plan takoy - xotelos' bi eshe minimum neskol'ko
    let zdes' pobit', esli vse budet normal'no s rabotoy. Esli uvoliat i prilichnuiu
     rabotu naiti ne udastsia, to togda bez osibix slez i napriagov mozhno exat' v
    Moscvu. Naschet poiska raboti eshe rano govorit', posmotrim kogda vse tozhno
    opredelitsia.
Pogoda vse-taki eto ochen' vazhno - v Moscve takaia gadost' ,a  u nas tut +20 i vse
     v shortax. Moskovskaia pogoda - eto to chto menia napriagaet bol'she vsego, bol
    'she vsiakix ekonomicheskix i drugix problem, a v ostal'nom mne kazhetsia mozhno
     exat' nazad i zashibat' tam prilichnuiu babku.

DAvay, napishi eshe che-nibud'..
```

Listing 6: Emails from best DB result of Chain of Length 3

```
<Cluster: 0>, Chain ID: 000395, Index: 69681
Date: 2001-01-04 16:39:00
Subject: Re: #486435
Sender: kate.symes@enron.com, Recipient: kimberly.allen@enron.com
Message: Matt Motley and Mike Swerzbin both said this trade should be desk-to-desk
    and
has nothing to do with Bonneville. Could you send me a copy of the confirm
letter? They'd like to look at it to try and figure out what trade BPA is
referring to. If it's a hard copy, you can fax it to 503-464-3740. Thanks a
lot.

Kate


        From:  Kimberly Allen                           01/04/2001 03:16 PM


To: Kate Symes/PDX/ECT@ECT
cc:

Subject: #486435

Hey Kate! I have a trade that Motley did and it shows as a desk to desk
trade. But BPA sent me a confirmation on the trade. Is this suppose to be
desk to desk or should the counterparty actually be BPA?

Thanks,

Kimberly Indelicato
```

```
<Cluster: 0>, Chain ID: 000395, Index: 69727
Date: 2001-01-04 18:16:00
Subject: #486435
Sender: kimberly.allen@enron.com, Recipient: kate.symes@enron.com
Message: Hey Kate! I have a trade that Motley did and it shows as a desk to desk
trade. But BPA sent me a confirmation on the trade. Is this suppose to be
desk to desk or should the counterparty actually be BPA?

Thanks,

Kimberly Indelicato

<Cluster: 0>, Chain ID: 000395, Index: 69847
Date: 2001-01-05 10:05:00
Subject: Re: #486435
Sender: kimberly.allen@enron.com, Recipient: kate.symes@enron.com
Message: Just faxed the BPA confirm to you. Just let me know.

Thanks,

KI    Kate Symes                01/04/2001 05:39 PM

To: Kimberly Allen/HOU/ECT@ECT
cc:
Subject: Re: #486435

Matt Motley and Mike Swerzbin both said this trade should be desk-to-desk and
has nothing to do with Bonneville. Could you send me a copy of the confirm
letter? They'd like to look at it to try and figure out what trade BPA is
referring to. If it's a hard copy, you can fax it to 503-464-3740. Thanks a
lot.

Kate


        From:  Kimberly Allen                              01/04/2001 03:16 PM


To: Kate Symes/PDX/ECT@ECT
cc:

Subject: #486435

Hey Kate! I have a trade that Motley did and it shows as a desk to desk
trade. But BPA sent me a confirmation on the trade. Is this suppose to be
desk to desk or should the counterparty actually be BPA?

Thanks,

Kimberly Indelicato
```

Listing 7: Example of an email with severe impurities

```
<Cluster: -1>, Chain ID: 167107, Index: 138964
Date: 2001-07-11 12:45:00
```

```
Subject: Re: Sher Shops Alternative Edison Bailout Plan
Sender: drothrock@cmta.net, Recipient: jeff.dasovich@enron.com
Message: worse for SCE and generators, who have to eat the small guy share of the
undercollection
between them. No transmission sale.

D

Jeff.Dasovich@enron.com wrote:

> better or worse than ours?
>
>
>                     Dorothy
>                     Rothrock             To:     Jeff.Dasovich@enron.com
>                                          ta.net>             Subject:     Re:
    Sher Shops
Alternative Edison
>                                          Bailout Plan
>                     07/11/2001
>                     12:20 PM
>
>
>
> let me know if delaney doesn't send to you...
>
> d
>
> Jeff.Dasovich@enron.com wrote:
>
> > Thanks.  415.782.7854.  Better or worse than ours?
> >
> >
> >                       Dorothy
> >                       Rothrock             To:     Jeff.Dasovich@enron.com
> >                        "'Barbara Barkovich
> >                       ta.net>              (E-mail)'"
> , "Dominic
> >                                            DiMare (E-mail)"
> ,
> >                       07/11/2001           "'John Fielder (E-mail)'"
> ,
> >                       11:54 AM             "'Phil Isenberg (E-mail)'"
> ,
> >                                            "'Jeff Dasovich (E-mail)'"
> ,
> >                                            "'Keith McCrea (E-mail)'"
> ,
> >                                            "'Linda Sherif (E-mail)'"
> ,
> >                                            "'Linda Sherif (E-mail 2)'"
> ,
> >                                            "'Gary Schoonyan (E-mail)'"
> ,
> >                                            "'John White (E-mail)'"
```

```
> ,
> >                                                dhunter@s-k-w.com,
> Rick.Simpson@asm.ca.gov
> >                                                Subject:     Re: Sher Shops
> Alternative Edison
> >                                                Bailout Plan
> >
> >
> > I have the plan.....who wants it? send your fax number (and $10 for
> > shipping
> > and handling....just kidding)
> >
> > D
> >
> > Jeff.Dasovich@enron.com wrote:
> >
> > > Folks:  Please see highlighted sections.  Anyone seen Byron's plan?
> Know
> > > where it's headed, etc.?
> > >
> > > Best,
> > > Jeff
[...]
```

Listing 8: Emails from best CH result of Chain of greater or equal to 4 and less than 10

```
<Cluster: -1>, Chain ID: 218660, Index: 193402
Date: 2001-11-07 22:47:58
Subject: RE: thinking of you
Sender: jason.wolfe@enron.com, Recipient: eellwanger@triumphboats.com
Message: Around North Carolina? no

<Cluster: -1>, Chain ID: 218660, Index: 193403
Date: 2001-11-07 22:48:15
Subject: RE: thinking of you
Sender: eellwanger@triumphboats.com, Recipient: jason.wolfe@enron.com
Message: Around Houston, ya dill wacker.

<Cluster: 0>, Chain ID: 218660, Index: 193405
Date: 2001-11-07 22:49:43
Subject: RE: thinking of you
Sender: jason.wolfe@enron.com, Recipient: eellwanger@triumphboats.com
Message: really? why?

<Cluster: -1>, Chain ID: 218660, Index: 193411
Date: 2001-11-07 22:54:40
Subject: RE: thinking of you
Sender: jason.wolfe@enron.com, Recipient: eellwanger@triumphboats.com
Message: whatever. just let me know when you aren't dicking around

<Cluster: -1>, Chain ID: 218660, Index: 194055
Date: 2001-11-08 21:47:42
Subject: RE: thinking of you
Sender: eellwanger@triumphboats.com, Recipient: jason.wolfe@enron.com
Message: Do you still have a job? Tara is hearing some pretty nasty stuff about
```

```
Enron. She's pretty worried that they are gonna renig on their offer.

<Cluster: -1>, Chain ID: 218660, Index: 194075
Date: 2001-11-08 22:02:57
Subject: RE: thinking of you
Sender: jason.wolfe@enron.com, Recipient: eellwanger@triumphboats.com
Message: If there is an Enron next week, she will probably get an offer. Unless we
    are owned by Dynegy.
Then I don't know. Either way, don't quit cleaning boats yet.

<Cluster: -1>, Chain ID: 218660, Index: 194373
Date: 2001-11-09 15:32:15
Subject: RE: thinking of you
Sender: eellwanger@triumphboats.com, Recipient: jason.wolfe@enron.com
Message: So, are people in your area getting nervous? Is everybody looking for jobs
at other companies? There are only 3 companies left to interview with
through Duke for Tara. Under normal circumstances she could just contact a
bunch of other energy companies in Houston and probably get as good an offer
from one of them, but with thousands of Enron employees flooding the market,
who knows. Tara found out from a friend of hers at UT that a company called
Peabody in St. Louis is willing to hire anybody with experience at Enron.
Sweet Jesus, I could get season tickets!

<Cluster: -1>, Chain ID: 218660, Index: 194476
Date: 2001-11-09 17:13:08
Subject: RE: thinking of you
Sender: jason.wolfe@enron.com, Recipient: eellwanger@triumphboats.com
Message: According to CNBC, the Dynegy deal is close, with an announcement today.
I'm not really nervous; Enron's strength is its wholesale trading dept.
I'm not sure about non-core business and corporate hacks, though.
We'll see how it all shakes out - it's really hard to believe it has come to this.
Jason

<Cluster: -1>, Chain ID: 218660, Index: 194561
Date: 2001-11-09 18:19:24
Subject: RE: thinking of you
Sender: jason.wolfe@enron.com, Recipient: eellwanger@triumphboats.com
Message: I'm eating Droubi's middle eastern cuisine right now, thinking of the
    times
we used to meet there for lunch. I miss those days.
```

Listing 9: Content of the shortest cluster from best SL result of Chain Length greater than 10

```
<Cluster: 1>, Chain ID: 131203 Index: 224024
Date: 2002-01-07 21:54:47
Subject: RE:
Sender: mike.maggi@enron.com , Recipient:  michelle.nelson@enron.com
Message: ok


<Cluster: 1>, Chain ID: 122336 Index: 229806
Date: 2002-01-17 17:36:36
Subject: RE:
Sender: mike.maggi@enron.com , Recipient:  amanda.rybarski@enron.com
Message: ok
```

Listing 10: Content example of Chain from the best SL result of Chain Length greater than 10

```
<Cluster: 0>, Chain ID: 131203 Index: 200565
Date: 2001-11-19 15:41:01
Subject: RE:
Sender: mike.maggi@enron.com , Recipient:  michelle.nelson@enron.com
Message: terrible, yours?

<Cluster: 0>, Chain ID: 131203 Index: 200584
Date: 2001-11-19 15:49:12
Subject: RE:
Sender: michelle.nelson@enron.com , Recipient:  mike.maggi@enron.com
Message: good.

# Example 2
<Cluster: 0>, Chain ID: 122336 Index: 209652
Date: 2001-11-27 20:42:17
Subject: RE:
Sender: amanda.rybarski@enron.com , Recipient:  mike.maggi@enron.com
Message: PERFECT...Black Lab???

# Example 3
<Cluster: 0>, Chain ID: 122161 Index: 65626
Date: 2000-12-14 09:14:00
Subject: Re:
Sender: jeffrey.shankman@enron.com , Recipient:  alexandra.saler@enron.com
Message: oh.....my.....god.....
```