Evaluating Cohesion Score with Email Clustering

Abhishek Kathuria, Devarshi Mukhopadhyay and Narina Thakur

Abstract An email has become one of the prime ways of communication for individuals or organizations and has emerged as an important research field to categorize emails and enable users for easy data segregation, topic modeling, spam detection, network analysis for investigative and analytical purposes. The paper aims to cluster the emails comprising of 500,000 emails taken from the Enron email dataset which was obtained by the Federal Energy Regulatory Commission during its investigation of Enron's collapse, based on the relevance of the words to the whole corpus. The proposed algorithm calculates the cohesion score of each cluster using intra-cluster similarity. This paper implements two unsupervised clustering algorithms for the email clustering process, namely k-means and hierarchical clustering and evaluates the cosine similarity of all the words from each cluster to evaluate the semantic similarity pervading through each cluster. The emails were clustered into three groups and the cohesion score was obtained for each cluster which measured the intra-cluster similarity. The proposed method helped in the computation of the score distribution among the clusters, as well as the intra-cluster similarity. Cluster 1 obtained the highest cohesion score among all the three clusters by attaining the cohesion score 0.1655 while using the k-means algorithm and the cohesion score of 0.2513 while using the hierarchical clustering algorithm.

Keywords Email clustering · TF-IDF · Clustering techniques · K-means · Hierarchical

A. Kathuria (✉) · D. Mukhopadhyay · N. Thakur
Bharati Vidyapeeth's College of Engineering, New Delhi, India
e-mail: abhishekkathuria40@gmail.com

D. Mukhopadhyay
e-mail: debom97@gmail.com

N. Thakur
e-mail: narinat@gmail.com

P. K. Singh et al. (eds.), *Proceedings of First International Conference on Computing, Communications, and Cyber-Security (IC4S 2019)*, Lecture Notes in Networks and Systems 121, https://doi.org/10.1007/978-981-15-3369-3_9

1 Introduction

Emails are a part of quotidian life. Emails are used in personal as well as professional life as an economical, reliable and instant way of communication. They serve as an archival tool for many people, as many users never discard some messages because of the significance of the information attached in the proximate future, for example, as a reminder of upcoming events and outstanding issues. However, systems dealing with data mining like information retrieval systems, search engines might be prone to certain unresolved ambiguities, leading to their performance degradation. Name discrimination and email clustering are two different aspects of textual classification as emails are clustered based on their contextual and semantic similarity. The growth of the Internet has led to an exponential increase in the number of digital documents being generated; hence, the analysis of the textual information within emails has become a notable field of study under the category of data mining. The two principal algorithms that are used in this paper for clustering are k-means clustering and hierarchical clustering. Hierarchical clustering is used for obtaining an in-depth analysis of the cluster as well as determining the basis of clustering for each data point, while k-means are used for an efficient and fast information retrieval clustering system. Hierarchical clustering creates clusters that contain the pretrained ordering from top to bottom. The challenges in the email clustering domains have been addressed by many researchers. Alsmadi (2015) [1] discusses various issues which arise in email clustering. Most frequently used words have been collected from the dataset using NLP techniques in these emails and inverse document frequency has been used to find the importance of these words for the document they appear in. The clustering technique used here is k-means. Generally, clustering algorithms are divided into two broad categories—hard and soft clustering methods [2]. Hard clustering algorithms differentiate between data points by specifying whether a point belongs to a cluster or not, that is, absolute assignment, whereas in soft clustering, each data point has a varying degree of membership in each cluster. Dimensionality reduction methods can be considered as a subtype of soft clustering; for documents, these include latent semantic indexing (truncated singular value decomposition on term histograms) and topic models. Other algorithms involve graph-based clustering, ontology supported clustering and order sensitive clustering.

Given a clustering technique, it can be beneficial to automatically derive human-readable labels. An email comprises numerous attributes such as sender's address, receiver's address, subject, message body, etc. Email mining is a process which is a subpart of text mining. It refers to a method of discovering insightful patterns and information from large email data. There are various applications of email mining such as email summarization, categorization, spam filtering, etc.

The main idea on which the clustering is based is the distance of elements. As a result of clustering, nearby elements are grouped into common sets. Through this paper, we intend to gain insights and determine important and valuable terms that occur in the message body of our dataset. This involves an in-depth data analysis and gaining valuable deductions from the message body text. The clustering will eventually help us in determining various things such as about the company from where the dataset has been taken, the important and powerful people involved, the users who communicated frequently through email about a specific topic, the main topics discussed in the company and many others.

Our approach includes clustering the emails based on the features obtained through the message body. The two most simple and less expensive clustering algorithms, namely k-means clustering and hierarchical clustering, are used, which are the most viable options for pre-clustering, as they reduce the space into disjoint smaller subspaces. The elbow method has been used in the k-means clustering algorithm for the determination of clusters. The determination of the optimal number of clusters has been a key problem for many researchers such as Mark Ming-Tso Chiang (2010) [3].

In our paper, for finding the optimal number of clusters in k-means clustering, a more precise method called the 'silhouette method' has been used. In the hierarchical clustering algorithm, the bottom-up approach for clustering called the agglomerative clustering method has been used. A dendrogram has been plotted for the determination of the number of clusters. The clusters have been determined based on the number of points intersected by the threshold line. For the determination regarding the selection of a cluster for gaining concentrated insights, a cohesion score has been calculated which determines the intra-cluster similarities.

The rest of the paper is organized as follows: Related work is discussed in short in Sect. 2, proposed cohesion-based system is described in Sect. 3, experimental setup is included in Sect. 4, the results and analysis are presented in Sect. 5, conclusion is given in Sect. 6 and the references are mentioned in the last section.

2 Related Works

A lot of research has been done on clustering [3–7] in the past decades. Collecting an archive of emails for analysis can be done for several purposes. While some of them focus on presentation reforming, others focus on investigating the efficiency of various clustering algorithms. The performance of experimented k-means clustering algorithms has been evaluated by Alsmadi (2015) [1]. Mark Ming-Tso Chiang (2010) discussed the most contentious problem in k-means clustering, that is, selecting the correct value of the number of clusters [3]. Chiang concentrated on analyzing the performance of an intelligent version of k-means, known as ik-means, which uses anomalous pattern (AP) clusters for initializing k-means clustering. The results showed that this adjusted k-means or ik-means outperformed several other methods, concerning centroid and data recovery, but overestimated the numbers of clusters in the case of small in-between cluster spreads. Andrew Lensen (2017) introduced the coherent approach consisting of three stages for selection of the optimal number of clusters, k, and performing concurrent feature selection along with clustering [8]. In the first stage, Kest, which is an estimate of k, was determined using the silhouette method. The second stage consisted of using the Kestvalue to perform a guided search by particle swarm optimization (PSO) for calculating the number of clusters. The final stage used the centroid representation to perform a localized pseudo search for fine-tuning the solution obtained in stage two. Email categorization has been addressed and worked upon by Azizpour (2018) [4]. Here, similarity measure for text processing (SMTP) has been used for clustering emails. The efficiency of similarity algorithms like Euclidean distance, cosine similarity, extended Jaccard coefficient and dice coefficient and SMTP has been compared by implementing them with a k-means clustering algorithm. For that, four clusters were created and similar emails were put into them. It was observed that the results obtained by using SMTP with the k-means clustering algorithm were better than others; M. Basavaraju (2010) used a text-based clustering approach for spam detection [9]. The datasets used here were ling spam corpus (more description). The spam detection techniques used here were based on the vector-based model. The clustering algorithm incorporated features of both k-means and the BIRCH algorithm. It was observed (result) that the k-means clustering algorithm worked best with smaller datasets while the combination of BIRCH with KNNC worked better with large datasets.

Huang (2008) used a mixed-initiative approach to hierarchically cluster emails [10]. In this approach, the hierarchies are first decided by the computer and then revised through repetitive user feedbacks to make them more meaningful and useful. An edge modification ratio is defined to compare the resulting hierarchies against a reference one. The paper concluded by stating hierarchical clustering helps a user understand the results better than flat clustering and any mixed-initiative system should acknowledge subsequent variation in user feedbacks to create effective strategies for retraining models.

Ercan G (2008) addressed the problem of text summarization, that is, forming extracts using lexical cohesion [11]. Summarization of text includes selecting the most representative sentences. Ercan employed the lexical cohesion of the text structure as a means of evaluating significant sentences. For the linear text segmentation by topic, Pérez (2010) employed clustering cohesion as a criterion for evaluation [12]. The results were then fed to their proposed incremental overlapping clustering algorithm to assign the input stream to a topic cluster.

Klebanov (2008) proposed an automatic method of text analysis aimed at discovering patterns of lexical cohesion, that is, groups of words with similar meaning [13]. Political speeches were analyzed and groups of words with related meaning were extracted and compared with manual analysis of political rhetoric. Hence, the authors concluded that lexical cohesion is a viable method for quantitative and qualitative analysis of the text.

2.1 Comparison Table

The following Table 1 describes the objectives, approach, pros and cons of various state of the art methods.

Table 1 Comparison between various state-of-the-art methods

Author	Year	Objectives	Approach	Pros	Cons
Alsmadi [1]	2015	Clustering and classification of email contents	Using NGrams to develop a feature matrix	High true positives rates in almost all of the cases	Larger feature matrix had to be developed
Chiang [3]	2010	Determining the right no. of clusters in k-means	Using Intelligent k-means clustering	Outperforms other evaluation measures in centroid and data recovery	Overestimates the number of clusters
Basavaraju [9]	2010	Efficient spam mail classification using clustering techniques	Using j-means and BIRCH	High precision rates for both the clustering models	K-means works well with smaller datasets
Xie [5]	2016	Deep embedded clustering (DEC)	Propose to iteratively refine clusters with auxiliary target distribution	DEC is significantly less sensitive to the choice of hyperparameters	No method to find optimal clusters

3 Proposed Cohesion Evaluation-Based Cluster System

3.1 Problem Formulation

After preprocessing, the processed data with the message body is obtained in the form of a data frame. For each message body, the text is lemmatized and converted into lowercase. The stop words are removed by using the Natural Language Toolkit (NLTK) corpora. The term frequency-inverse document frequency matrix was generated using the term frequency-inverse document frequency (TF-IDF) vectorizer. The TF-IDF is a mathematical measure that evaluates the importance of a word to a document in a document corpus based on the frequency of the word occurring in a document and its relevance to the corpus. We used the TF-IDF vectorizer instead of TF-IDF transformer as it counts the words, finds the IDF values and calculates the TF-IDF scores simultaneously. The cleaned data was fed to the TF-IDF matrix to obtain the TF-IDF score for each word in each document. The TF-IDF score is calculated by using the formula as shown in Eq. (1).

$$w_{t,d} = \left(1 + \log_{10}\left(f_{t,d}\right)\right). \log_{10} \frac{N}{df_t} \tag{1}$$

Here, $\log_{10}\frac{N}{df_t}$ is the inverse document frequency score calculated by taking the log of N, which is the total number of documents in the corpus and f_t is the long-term frequency. The long-term frequency of a term t in d is given by $f_{t,d}$ and $w_{t,d}$ gives the TF-IDF score. Using the TF-IDF features, k-means clustering algorithm and hierarchical clustering algorithm are applied. The k-means clustering algorithm is an unsupervised clustering algorithm which determines the optimal number of clusters using the elbow method. It works iteratively by selecting a random coordinate of the cluster center and assigns the data points to a cluster. It then calculates the Euclidean distance [14] of each data point from its centroid, and based on this, it updates the data point positions as well as the cluster centers. For minimizing the within cluster sum of squares (WCSS), the formula in Eq. (2) is used:

$$\arg\ \min_S (x+a)^n = \sum_{i=1}^{k} \frac{1}{2|S_i|} \sum_{x,y\in S_i} \|x-y\|^2 \tag{2}$$

where S_i is the mean of the points, x contains the observations in a d-dimensional vector and k is the number of cluster centers. As the elbow method can sometimes obfuscate the deduction for the optimal number of clusters, silhouette analysis can prove to be a more precise method. A silhouette value always lies in the range [−1, 1] where +1 depicts that the data point under consideration is in close proximity with the assigned cluster and faraway from its neighboring cluster, whereas −1 depicts that the data point under consideration is in close proximity with its neighboring cluster and faraway from the assigned cluster. Let us consider a data point, 'x.' Let 'A' be the assigned cluster to this data point and 'B' be the neighboring cluster. Hence, the silhouette score, $S(x)$, can be given by Eq. (3).

$$S(x) = \frac{M(x) - N(x)}{\max(M(x) - N(x))} \tag{3}$$

where $M(x)$ is the mean distance of the point 'x' with respect to all the data points in the assigned cluster. And $N(x)$ is the mean distance of the point 'x' with respect to all the data points in the neighboring cluster. Hierarchical clustering creates clusters which consist of a predetermined ordering from top to bottom. In this paper, agglomerative method is used for hierarchical clustering. It is a bottom-up approach where each observation is assigned to its own cluster and each data point is considered as a separate cluster. The distance between each cluster is calculated using Ward's method and two similar clusters are combined together. This process is continued until there is only one cluster left. The Ward's method for calculating distance is given by the following formula:

$$\Delta(A,B) = \sum_{i\in A\cup B} \left\|\overrightarrow{x_i} - \vec{m}_{A\cup B}\right\|^2 - \sum_{i\in A}\left\|\overrightarrow{x_i} - \vec{m}_A\right\|^2 - \sum_{i\in B}\left\|\overrightarrow{x_i} - \vec{m}_B\right\|^2 \tag{4}$$

$$= \frac{n_a n_b}{n_a + n_b}\left\|\overrightarrow{m_A} - \vec{m}_B\right\|^2 \tag{5}$$

In Eqs. (4) and (5), m_j is the cluster center of j and n_j is the number of points inside the cluster. Δ refers to the cost of combination of A and B. Basically, the sum of squares is zero in the beginning and grows gradually as the clusters are merged. This method is responsible for keeping this growth as minimal as possible. The cohesion score [15] is calculated by using the cosine similarity formula [16] as discussed in Eq. (6):

$$\cos\theta = \frac{d_2 \cdot q}{\|d_2\| \|q\|} \tag{6}$$

In Eq. (6), d_2 and q are the TF-IDF vectors and θ is the angle between the vectors. The obtained cohesion score identifies the intra-cluster similarity between the clusters.

3.2 Architecture

The message body of each email (obtained from the ‘body’ column of the data frame) was processed before constructing a TF-IDF matrix, that is, stop words were removed, and each word was lemmatized. These processed messages were used to construct a TF-IDF matrix [17]. Figure 1 clearly depicts that this matrix was fed to the k-means and hierarchical clustering functions to obtain three clusters. The words from each cluster were extracted and their pairwise cosine similarity was used as a measure of cohesion within the clusters.

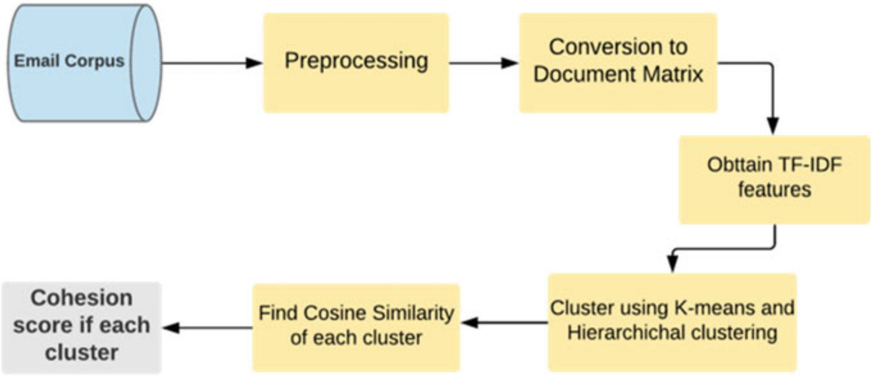
Email Corpus
Preprocessing
Conversion to Document Matrix
Obtain TF-IDF features
Cluster using K-means and Hierarchichal clustering
Find Cosine Similarity of each cluster
Cohesion score if each cluster

Fig. 1 Proposed cohesion evaluation-based cluster system

4 Experimental Setup

4.1 Dataset

The Enron email dataset was collected and prepared by the Cognitive Assistant that Learns and Organizes (CALO) project. It contains data from about 150 users, mostly senior management of Enron, organized into folders [18]. The corpus contains a total of about 0.5 M messages. The original data included approximately 500,000 emails generated by employees of the Enron Corporation. These emails were read as a.csv file, where the data was split into three columns, namely index, message id and raw message. This raw message contained all the fields present in an email format. From this raw message, the message body was obtained; hence, a data frame of 500,000 fields was constructed with the columns as ‘body,’ ‘to’ and ‘from.’ It is on the column of ‘body,’ that is, the message body of each email that the cluster analysis was implemented and evaluated.

4.2 Evaluation Measure

When TF-IDF transformation is applied to a text corpus, real-valued vectors of the features (words) for each data point (document) are obtained. Cosine similarity is a common measure to measure cohesion within clusters in text mining [19]. Although the extent to which the quality of clustering is judged is usually determined by comparing how tightly packed a cluster is, and the distance of that cluster from other clusters, it proves to be a trivial and naïve method for analyzing the clustering. A more sophisticated and precise method, for analyzing the quality of clustering, called the cohesion analysis, is discussed in this paper. This cohesion analysis shows the intra-cluster similarity by using cosine similarity. The cosine similarity can be thought of as a similarity measure which calculates the dot product between two nonzero vectors. This cosine score varies between 1 and zero, with zero implying that the vectors are perpendicular, that is, highly unrelated and one implying perfect correlation or practically speaking, they are the same vectors. Hence, all the words from each cluster were extracted, and for each cluster, a matrix was constructed and the cosine similarity score for each pair of the word was calculated. Cosine similarity is a widely used metric where the vector magnitude is not important; as this usually happens while we are working with the textual data which is represented by the word counts. Hence, an average score across this matrix was calculated. This is the cohesion score for the cluster.

5 Result Analysis and Discussion

This section deals with the implementation and evaluation of the clustering algorithms. In this paper, the Enron email dataset containing 500,000 emails was collected. From every email, the message body was extracted and a data frame of these message bodies was created using Pandas [20]. Removal of stop words is done along with lemmatizing. Lemmatizing is the process of reducing each word to its lemma [21]. Next, the email bodies are converted into a document-term matrix using TF-IDF. The term frequency-inverse document frequency (TF-IDF) is the mathematical statistic that is intended to reflect how important a word is to a document in a collection or corpus. The next step is writing a function to get the TF-IDF features out of all the emails. Using document vectors, clustering is done using k-means and hierarchical clustering algorithm. K-means assigns each data point to a different cluster, ensuring that the distance between that cluster center and the data point is minimum compared to the distance with other cluster centers [22]. For finding the number of centroids, the 'elbow method' is used. In this, the sum of squared error (SSE) value is calculated for different values of k (that is, number of clusters) by clustering the dataset following each value of k [23]. The point on the graph where a 'hinge' occurs is considered to be the optimal value of k. Figure 2 shows the elbow method for k-means algorithm. Thus, by looking at the graph, the total number of clusters can be either 2 or 3.

In order to find the optimal number of clusters, silhouette score is used. As the number of cluster centers given by the elbow method lies in the range of [2, 9], we will use n_clusters for determining the optimal number of cluster centers by using the silhouette scores, where n_clusters is a number in the range [2, 9]. The following silhouette scores were obtained for each cluster which is shown in Table 2.

It can be inferred from Table 2 that the value of n_clusters as 4 and 5 are a bad pick as they have the least average silhouette score for the given set of data. The average

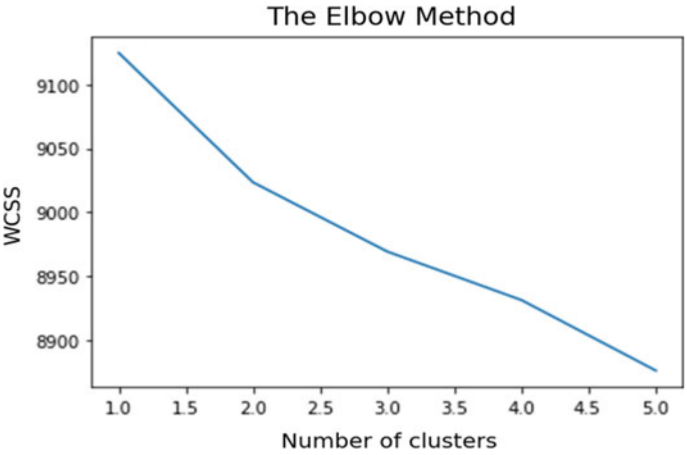

Fig. 2 Elbow method

Table 2 Average silhouette scores

Number of clusters	Average silhouette score
n_clusters = 2	0.65
n_clusters = 3	0.74
n_clusters = 4	0.58
n_clusters = 5	0.41

sihouette score of n-clusters = 2 is 0.65 which is greater than that of n_cluster value of 4 and 5 but is still less than the average silhouette score of n_clusters = 3. Hence, Table 2 clearly depicts that the average silhouette score for n_clusters value of 3, is the highest. Hence, 3 is the optimal number of clusters. After training, the following three clusters are obtained, which are shown in Fig. 3.

After performing these steps, the cosine similarity is calculated for all the terms using the TF-IDF scores of each term for every cluster. For hierarchical clustering [24, 25], the maximum difference between the Euclidian distance of the level of each pair of group is chosen. Then, a horizontal line is passed in the middle of the maximum difference of that Euclidean distance. The number of points the line cuts gives the number of cluster centers. From the dendrogram [26] plotted in Fig. 4, the number of cluster centers is given as three.

The cosine similarity was calculated for all the terms for each cluster using the TF-IDF scores of each term. Cosine similarity is a common technique used to measure the cohesion within the clusters in the field of data mining [18]. Cosine similarity is a measure of similarity between two nonzero vectors of an inner product space that measures the cosine of the angle between them as in (7). The cosine of 0° is 1, and it is less than 1 for any other angle.

To find the cosine distance of one email and all the others, one just needs to compute the dot products of the first vector with all of the others as TF-IDF vectors

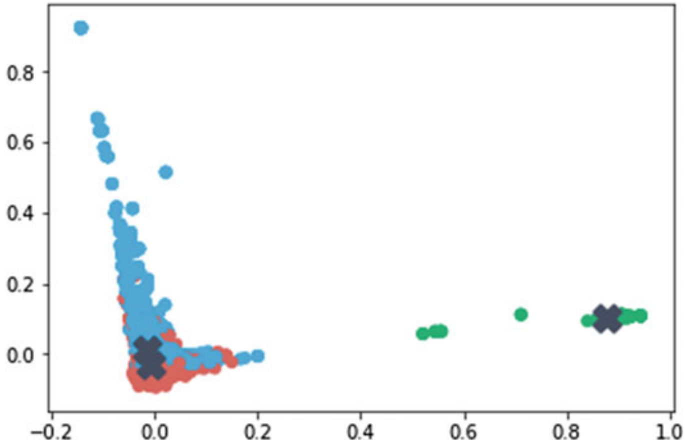

Fig. 3 Clusters

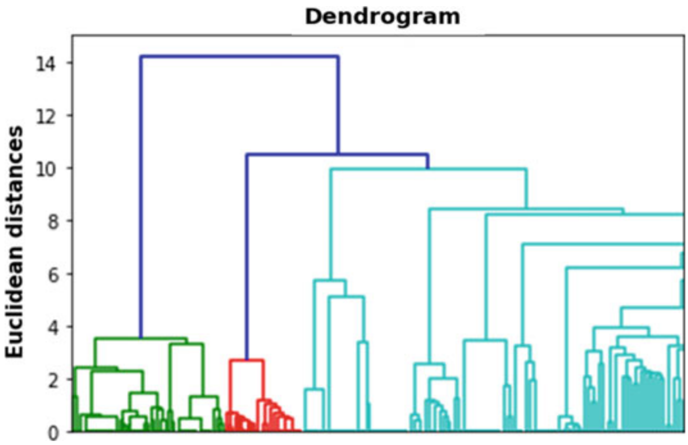

Fig. 4 Dendrogram

are already row normalized. Scikit-learn already provide pairwise metrics (also known as kernels in machine learning parlance) that work both for dense and sparse representations of vector collections [27]. In this case, the dot product is also known as the linear kernel [18]. The cohesion score of the three clusters is given in Table 3.

The cohesion scores given in Table 3 depict that the majority of the information and relevant insights can be extracted through cluster 1 as it has the highest cohesion score. By using the k-means algorithm, the cohesion score of cluster 1 is 0.1655, which is the highest as compared to the cohesion scores of 0.1011 and 0.1252 for the cluster 0 and cluster 2, respectively. For the hierarchical clustering algorithm, the least cohesion score of 0.1421 is obtained by cluster 2 while the highest score is obtained by cluster 1 which corresponds to 0.2513. Moreover, it can also be observed from Table 3 that the cohesion score obtained by cluster 0 is 0.1900. All these observations further strengthen the relevancy of the textual content and help in obtaining dependable and precise insights by choosing a specific cluster. Hence, it can be inferred that all the three clusters can be utilized for getting the holistic insights and specific chosen clusters can be used to gain pertinent and relevant insights.

Table 3 Cohesion score for each cluster

Clustering algorithms	Cohesion score		
	Cluster 0	Cluster 1	Cluster 2
K-means	0.1011	0.1655	0.1252
Hierarchical	0.1900	0.2513	0.1421

6 Conclusion

Email clustering utilizes several natural language processing and data mining activities such as: text parsing, stemming, classification, clustering, etc. There are many reasons for carrying out clustering whether in real time or historical. This may include reasons such as: spam detection, subject or folder classification, information extraction, etc. The clustering of this email dataset revealed quite a few interesting aspects. One of them was the score distribution among the clusters as well as the similarity of the clusters obtained by the two different clustering algorithms. The cohesion score helped in concentration of analysis to a specific cluster for better insights. Further investigation could explain as to the reasons for this skewed score distribution as well as the resulting conclusions from this similarity of clusters. Soft clustering methods like fuzzy clustering could be utilized to determine the degree of membership each data point has to each cluster, opening up further avenues of investigation.

References

1. Alsmadi, I., Alhami, I.: Clustering and classification of email contents. J. King Saud Univ.-Comput. Inf. Sci. **27**(1), 46–57 (2015)
2. Chen, M.: Soft clustering for very large data sets. Comput. Sci Netw Secur. J **17**(11), 102–108 (2017)
3. Chiang, M.M.-T., Mirkin, B.: Intelligent choice of the number of clusters in k-means clustering: an experimental study with different cluster spreads. J. Classif. **27**, 3–40 (2010)
4. Azizpour, S., Giesecke, K., Schwenkler, G.: Exploring the sources of default clustering. J. Financ. Econ. **129**(1), 154–183 (2018)
5. Xie, J., Girshick, R., Farhadi, A.: Unsupervised deep embedding for clustering analysis. In: International Conference on Machine Learning, pp 478–487 (2016)
6. Nayak, P., Devulapalli, A.: A fuzzy logic-based clustering algorithm for WSN to extend the network lifetime. IEEE Sens J **16**(1), 137–144 (2015)
7. Ferrari, D.G., De Castro, L.N.: Clustering algorithm selection by meta-learning systems: a new distance-based problem characterization and ranking combination methods. Inf. Sci. **301**, 181–194 (2015)
8. Lensen, A., Xue, B., Zhang, M.: Using particle swarm optimisation and the silhouette metric to estimate the number of clusters, select features, and perform clustering. In: European Conference on the Applications of Evolutionary Computation, pp. 538–554. Springer, Cham (2017)
9. Basavaraju, M., Prabhakar, D.R.: A novel method of spam mail detection using text based clustering approach. Int. J. Comput. Appl. **5**(4), 15–25 (2010)
10. Huang, Y., Mitchell, T.M.: Exploring hierarchical user feedback in email clustering. Email **8**, 36–41 (2008)
11. Ercan, G., Cicekli, I.: Lexical cohesion based topic modeling for summarization. In: International Conference on Intelligent Text Processing and Computational Linguistics, pp. 582–592. Springer, Berlin (2018)
12. Klebanov, B.B., Diermeier, D., Beigman, E.: Lexical cohesion analysis of political speech. Polit. Anal. **16**(4), 447–463 (2008)
13. Pérez, R.A., Pagola, J.E.M.: Text segmentation by clustering cohesion. In: Iberoamerican Congress on Pattern Recognition, pp. 261–268. Springer, Berlin (2010)

14. Behrens, T., Schmidt, K., Viscarra Rossel, R.A., Gries, P., Scholten, T., MacMillan, R.A.: Spatial modelling with euclidean distance fields and machine learning. Eur. J. Soil Sci. **69**(5), 757–770 (2018)
15. Rathee, A., Chhabra, J.K.: Improving cohesion of a software system by performing usage pattern based clustering. Procedia Comput. Sci. **125**, 740–746 (2018)
16. Kulkarni, A., Pedersen, T.: Name discrimination and e-mail clustering using unsupervised clustering of similar contexts. J. Intell. Syst. **17**(1–3), 37–50 (2008)
17. Reed, J.W., Jiao, Y., Potok, T.E., Klump, B.A., Elmore, M.T., Hurson, A. R.: TF-ICF: a new term weighting scheme for clustering dynamic data streams. In: 2006 5th International Conference on Machine Learning and Applications (ICMLA'06), pp. 258–263. IEEE (2006)
18. Hermans, F., Murphy-Hill, E.: Enron's spreadsheets and related emails: a dataset and analysis. In: 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering vol. 2, pp. 7–16. IEEE (2015)
19. Al-Anzi, F.S., AbuZeina, D.: Toward an enhanced Arabic text classification using cosine similarity and latent semantic indexing. J King Saud Univ-Comput. Inf. Sci **29**(2), 189–195 (2017)
20. Bernard, J.: Python data analysis with pandas. In: Python Recipes Handbook, pp 37–48. Apress, Berkeley, CA (2016)
21. Gupta, R., Jivani, A.G.: Analyzing the stemming paradigm. In: International Conference on Information and Communication Technology for Intelligent Systems, pp 333–342. Springer, Cham (2017)
22. Capó, M., Pérez, A., Lozano, J.A.: An efficient approximation to the K-means clustering for massive data. Knowl. Based Syst. **117**, 56–69 (2017)
23. Syakur, M.A., Khotimah, B.K., Rochman, E.M.S., Satoto, B.D.: Integration k-means clustering method and elbow method for identification of the best customer profile cluster. In: IOP Conference Series: Materials Science and Engineering, vol. 336, no. 1, p. 012017. IOP Publishing (2018)
24. Zhou, S., Xu, Z., Liu, F.: Method for determining the optimal number of clusters based on agglomerative hierarchical clustering. IEEE Trans. Neural Netw. Learn. Syst. **28**(12), 3007–3017 (2016)
25. Day, W.H., Edelsbrunner, H.: Efficient algorithms for agglomerative hierarchical clustering methods. J. Classif. **1**(1), 7–24 (1984)
26. Ferreira, L., Hitchcock, D.B.: A comparison of hierarchical methods for clustering functional data. Commun Stat. Simul. Comput **38**(9), 1925–1949 (2009)
27. Kent, D., Toris, R.: Adaptive autonomous grasp selection via pairwise ranking. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2971–2976. IEEE (2018)