ISSN 2278-3091

Volume 13, No.3, May - June 2024

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCS1E/static/pdf/file/ijatcse021332024.pdf

https://doi.org/10.30534/ijatcse/2024/021332024

Optimizing Text Clustering: A Methodological Approach for Determining the Optimal Number of Clusters

Oussama Chabih[1], **Sara Sbai**[2], **Mohammed Reda Chbihi Louhdi**[3], **Hicham Behja**[4]

[1]LRI - Laboratory ENSEM, Hassan II University Casablanca, Morocco, ossama.chabih@ensem.ac.ma

[2]LRI - Laboratory ENSEM, Hassan II University Casablanca, Morocco, sara.sbai4@gmail.com

[3]LIS - Laboratory Faculty of Sciences Ain Chock, Hassan II University Casablanca, Morocco, chbihi@gmail.com

[4]LRI - Laboratory ENSEM, Hassan II University Casablanca, Morocco, h.behja@ensem.ac.ma

Received Date : April 12, 2024 Accepted Date: May 13, 2024 Published Date: June 06, 2024

ABSTRACT

Developing a method to determine the optimal number of clusters is a crucial endeavor, particularly in the domain of text clustering where the sheer volume of variations poses significant challenges. Recognizing this, our study is specifically tailored to address this challenge within the realm of unsupervised text analysis. We put forth an innovative approach that marries the K-means algorithm with Bregman distance, meticulously crafted to accommodate the idiosyncrasies inherent in textual data. Our iterative methodology is designed with a dual purpose: to mitigate the adverse effects of noise and to ensure the stability of the clusters formed, all underpinned by the sophisticated metric of Kullback-Leibler divergence. Through rigorous experimentation, we validated the efficacy of our method in effectively segmenting texts into coherent clusters. Notably, our approach outperformed an initial categorization, providing a more nuanced and representative depiction of the diverse array of topics present within the corpus. In essence, our study offers a promising avenue to enhance unsupervised text analysis, heralding potential advancements and avenues for further exploration in this dynamic field.

Key words : Kmeans, Number of clusters, Text document clustering, Unsupervised classification.

1. INTRODUCTION

In fields like natural language processing, a specific form of classification known as 'textual clustering' is used. This method organizes sets of texts into homogeneous groups without prior supervision-that is, without predefined labels. Each group, or cluster, brings together texts that are similar according to certain criteria, such as vocabulary used, style, subject, or even syntactic structure.

Text clustering is particularly useful for managing large volumes of textual data, such as those from social networks, digital archives, or research databases. For example, it enables businesses to analyze customer feedback by grouping comments into themes, making it easier to quickly identify common issues or suggestions. In the academic domain, it helps researchers discover trends and patterns in scientific literature, simplifying the literature review process and extracting relevant information.

Thus, whether used to explore unlabeled data, simplify information management, or even improve decision-making processes, textual clustering proves to be a valuable tool in the arsenal of modern data processing.

Determining the number of clusters remains a major challenge in unsupervised classification, especially in the context of text classification. This challenge arises from the absence of a universal method for identifying the optimal number of groupings in unlabeled data. Unlike supervised clustering, where classes are predefined, unsupervised clustering involves an unknown number of clusters that must be determined empirically. This uncertainty can result in insignificant groupings or over-segmentation of data, thereby impacting the quality of classification results and the ability to extract relevant information from texts. Consequently, the issue of determining the number of clusters remains a significant challenge in the field of unsupervised classification, particularly when applied to textual data.

To address this challenge, we propose integrating a robust method for determining the optimal number of clusters upstream of the K-means algorithm. This approach combines the effectiveness of the K-means algorithm with an adaptive K selection method, specifically designed to optimize the clustering process for textual data. Through the utilization of this method, we aim to gain deeper insights into the inherent complexity of text data and achieve more meaningful and coherent groupings. By enhancing the ability of K-means to dynamically adapt to the optimal number of clusters, our solution offers a more robust and efficient approach for text analysis and classification.

In this article, we commence with an exploration of existing solutions addressing the challenge of selecting the number of clusters in Section 2. We delve into the diverse methodologies present in the literature aimed at determining the optimal number of clusters across various data analysis contexts. Section 3 outlines our rationale for selecting K-means as the foundation for evaluating our method. We elucidate the factors underpinning the choice of K-means as the primary algorithm for our clustering approach.

Moving forward, Section 4 elaborates on our proposed approach to resolving the challenge of determining the optimal number of clusters. We introduce an adaptive K selection method integrated upstream of the K-means algorithm and provide a comprehensive explanation of its procedural steps.

In Section 5, we present the outcomes of our experiments. Here, we scrutinize the performance of our method through experimentation on diverse textual datasets and juxtapose its results with those yielded by existing approaches.

Finally, Section 6 encapsulates the conclusion of our article. We summarize the contributions of our study, deliberate on its implications, and delineate avenues for future research in this dynamic realm of textual data analysis.

2. RELATED WORKS

Use either SI (MKS) or CGS as primary units. (SI units are strongly encouraged.) English units may be used as secondary

2.1 Elbow Method

The Elbow method [1], commonly utilized in data analysis and machine learning, is employed to determine the optimal number of clusters in a set of unlabeled data. It derives its name from the shape of its graph, which illustrates the within-cluster variance versus the number of clusters, typically displaying a characteristic "elbow" or flex point.

To implement the Elbow method, we typically utilize a clustering algorithm such as K-means. Subsequently, we calculate the sum of the squares of the distances between each data point and its centroid, denoted as W(k), for each number of clusters k. This value is computed using the formula:

$$W(k) = \sum_{i=1}^{k} \sum\nolimits_{x \in C_i} ||x - \mu_i||^2$$

Where C_i is the i-th cluster, μ_i is the centroid of the i-th cluster and $||x - \mu_i||^2$ represents the Euclidean distance between a data point x and its centroid μ_i.

Next, we plot a graph of $W(k)$ versus k. As k increases, $W(k)$ generally decreases, as each data point is more likely to be closer to its centroid. However, there comes a point where adding more clusters does not significantly improve the within-cluster variance. This point corresponds to the "elbow" of the graph, indicating the optimal number of clusters.

2.2 The Silhouette Method

The silhouette [2] method is another technique used to assess the quality of clusters in a dataset. Unlike the elbow method, which focuses on within-cluster variance, the silhouette measures how similar each object is to its own cluster compared to other clusters, providing an indication of the compactness and separation of clusters.

To calculate the silhouette of a given point, two measurements are used: "a," representing the average similarity between this point and the other points in its cluster (intra-cluster coherence), and "b," representing the average similarity between this point and the points of the closest neighboring clusters (inter-cluster coherence). The silhouette of a point is then calculated as follows:

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

A silhouette value close to +1 indicates that the object is well-fitted to its own cluster and poorly fitted to neighboring clusters, while a value close to 0 indicates cluster overlap, and a value close to -1 indicates that the object could be better assigned to a neighboring cluster.

The average silhouette of all points in a dataset is then calculated to assess the overall quality of the clusters. A high mean silhouette indicates good separation between clusters, while a low mean silhouette suggests poor partitioning of the data into distinct clusters.

2.3 Other Methods

In this study [3], a novel spectral clustering algorithm was developed to solve the crucial challenge of automatically determining the number of clusters. By analyzing the angles between data points in the feature space, this algorithm identifies the optimal number of clusters, thus providing an efficient and accurate solution for clustering. The promising results obtained during experiments on several simulated datasets demonstrate the effectiveness of this approach. Furthermore, validation on a set of real industrial data concerning alumina evaporation confirms the relevance and practical applicability of the developed algorithm.

In [4], Yu and C. Zhou study the selection of the number of clusters in the K-means clustering algorithm. Based on bootstrap sampling, a new method is proposed to determine the best number of clusters based on the estimated interval between the actual value of the intra-cluster total sum of squares and its estimated interval. Experimental results, based on the UCI Machine Learning benchmark and randomly generated artificial simulated test datasets, demonstrate a significant improvement achieved by this method, overcoming the problems of converging to a local maximum due to an unreasonably large number of clusters selected.

In [5], a technique for determining the number of clusters in a corpus of short documents is proposed. Using a spectral algorithm adapted to short texts, the authors generate a dataset and study a Markov chain induced by the co-association matrix to observe a quasi-decoupling phenomenon over the iterations. A large spectral gap and a number of eigenvectors close to 1 are used to indicate the number of clusters. These results are demonstrated through experiments on several datasets.

The paper [6] presents an alternative method for selecting the number of clusters, based on "distortion," a measure of intra-cluster dispersion. This method, referred to as the "jump method," involves straightforward steps, such as running the k-means algorithm for various numbers of clusters and computing the corresponding distortions. By appropriately transforming the distortion curve, it becomes feasible to pinpoint the "true" number of clusters, thereby showcasing its effectiveness across a range of problems.

Paper [7] addresses the challenge of automatically determining the number of clusters in datasets by focusing on the fuzzy C-means (FCM) algorithm. The proposed algorithm introduces a new approach that reduces randomness in cluster initialization and combines splitting strategies with the basic FCM algorithm to automatically determine the number of clusters. Additionally, a new validity index that they named V_((WSJ)), is introduced to evaluate the quality of clustering results. The experimental results demonstrate the effectiveness of the new algorithm and the validity index.

The authors in [8] present a new spectral clustering algorithm which makes it possible to automatically determine the number of clusters in a dataset. Unlike established methods, their algorithm is based on a theoretical analysis of the spectral properties of block-diagonal affinity matrices, without normalizing the rows of the eigenvector matrix. Using a modification of the K-means algorithm, they exploit the simple geometric properties of the eigenvectors to detect whether the selected number of clusters is less than the actual number, thereby iteratively obtaining the number of clusters.

In [9], a new multiscale spectral algorithm is proposed to estimate the number of clusters in a dataset. Their algorithm iteratively calculates the Laplacian eigenvalues of the graph for a wide range of scale parameter values and estimates the number of clusters from the maximum deviation of the eigenvalues. Thus, the variation of the scale parameter is used to robustly and efficiently infer the number of clusters. The algorithm is validated on test datasets, both simulated and real-world, to confirm its performance.

Chabih et al [10] explore different unsupervised classification methods, focusing on improving the Hybrid Feature Selection Method) method HFSM method of Benghabrit et al [11]. To solve the problem of the optimal choice of the number of clusters, they propose an iterative approach based on repeating the algorithm with different values of k. They identify benchmark clusters that remain stable across multiple runs and use this information to determine the optimal number of clusters.

3. K-MEANS AND BERGMAN DISTANCE

The K-means algorithm [12] is one of the most widely used clustering techniques in data analysis and machine learning. Its objective is to partition a dataset into a predefined number of clusters, minimizing the intra-cluster variance. K-means works iteratively by assigning each data point to a cluster represented by its closest centroid, then recalculating the centroids based on the points assigned to each cluster, until convergence.
The updating centroids step in the K-means algorithm is done using the following formula:

$$\mu_i = \frac{1}{|C_i|}\sum\nolimits_{x\in C_i} x$$

Where μ_i is the centroid of the C_i cluster, and $|C_i|$ represents the number of points in the $|C_i|$ cluster. This formula calculates the new centroid by taking the average of all the coordinates of the points in the cluster.
To calculate the Bergmane distance (also called Jaccard distance on sets of binary features) between two sets of words, we use the following formula:

$$d(x,y) = 1 - \frac{|x \cap y|}{|x \cup y|}$$

Where $|x \cap y|$ represents the number of words common to sets x and y, and $|x \cup y|$ represents the total number of words present in both sets.

K-means with Bergman distance [13] is a good choice for text clustering for several reasons. First, Bergman distance is suitable for textual data because it takes into account similarity based on the presence or absence of words, which is crucial in the context of text where word frequency can vary significantly. Second, K-means is an efficient and scalable algorithm for clustering, making it suitable even for large datasets. Finally, K-means with Bergman distance is relatively simple to implement and interpret, making it an attractive choice for a wide range of text clustering applications, such as document categorization, text recommendation, content analysis, and sentiment analysis. Combining the power of the K-means algorithm with the relevance of the Bergman distance for text data provides a robust and efficient approach for text clustering.

4. PROPOSED METHOD

Determining the optimal number of clusters remains a delicate step in the clustering process, particularly complex when dealing with textual data. Indeed, textual data are often characterized by significant noise and variable density due to the diversity of the words used. Traditional methods such as the elbow method and silhouette coefficient are not always adequate in this context, as they can be influenced by the precision of the textual data, making the estimation of the optimal number of clusters less reliable.

In our approach, we propose an innovative method to determine the optimal number of clusters in texts. We opted for the K-means algorithm using Bregman distance, an approach renowned for its ability to generate relevant clusters in textual datasets. To refine this approach, we rely on reference clusters, exploiting the Kullback-Leibler (KL) [14] divergence with term frequency-inverse document frequency [15] (TF-IDF) to evaluate clusters and select the optimal k. This combined approach provides an accurate method for segmenting text data, contributing to more reliable and meaningful analysis.

The KL divergence is a measure that evaluates the difference between two probability distributions. It is defined mathematically as follows:

$$D_{KL}(P||L) = \sum_i P(i) log \frac{P(i)}{Q(i)}$$

In this formula, $P(i)$represents the probability of the event i according to the distribution P, and $Q(i)$ represents the probability of the event i according to the distribution $Q(i)$.

In the context of text analysis with TF-IDF (Term Frequency-Inverse Document Frequency) representation, KL divergence is used to quantify the divergence between the distribution of words in a cluster C and that in the entire corpus D.

Thus, for a cluster C with its word distribution $P_C(i)$ and the corpus D with its word distribution $P_D(i)$, the divergence of KL can be calculated as follows:

$$D_{KL}(C||D) = \sum_i P(i) log \frac{P(i)}{Q(i)}$$

In the calculation of $P_C(i)$, $P_D(i)$ is weighted by the term TF-IDF, which is defined as follows:

$$TF - IDF(t,d) = TF(t,d) \times IDF(t)$$

Where $TF(t,d)$ represents the frequency of the term t in the document $d(TF)$, and $IDF(t)$ represents the inverse of the frequency of term t in the corpus, calculated as $log \frac{N}{n_t}$, where N is the total number of documents in the corpus and n_t is the number of documents containing the term t in the corpus.

In the process of converging texts with TF-IDF, the KL divergence is used to determine which iteration of clustering we should stop at. By calculating the KL divergence for each stable cluster formed in each iteration, we can establish a stopping criterion by comparing this divergence with that of previous iterations. If the divergence converges to a stable value, this indicates that the clustering no longer evolves significantly and that the clusters have become stable. Thus, the KL divergence plays a crucial role in the iterative process of segmentation of texts into distinct clusters, by ensuring the convergence of the clustering while minimizing the impact of noise.

The approach involves running several iterations of the K-means algorithm, gradually increasing the number of clusters k. The algorithm starts with k = 2 and gradually increases this number with each iteration.

In each iteration, we identify stable clusters by checking whether they remain unchanged for at least two consecutive iterations. Once identified, texts belonging to these clusters are removed from the main corpus and kept as separate clusters.

Simultaneously, we calculate the KL divergence for all texts in each stable cluster. This divergence helps determine which iteration of k to stop at.

The calculated divergence is then used to create a divergence interval for each stable cluster. By evaluating the divergence of the remaining texts, we decide whether to stop the process based on the proximity of this divergence to the divergence intervals of the stable clusters.

If no other stable clusters are found and the remaining documents represent less than 20% of the corpus, we calculate the divergence of each non-stable cluster and compare it with the divergence intervals of the stable clusters. In the following figures 1,2 and 3, we will describe our algorithm.

```
Inputs:
    • corpus: Corpus of documents to be segmented into clusters.
    • divergence_clusters_stables: Array of cluster divergence scores (initialized to null on the first execution).
Outputs:
    • clusters_stables: List of stable clusters.
    • last_stable_cluster: Last stable cluster found.
    1. Initialization:
        o Set k =2.
        o Initialize table_clusters as an empty list.
        o Initialize continue to true.
    2. Main Loop:
        o While continue is true, do:
            1. Initialize clusters_stables as an empty list.
            2. Append clusters generated by executing the K-means algorithm with k clusters to table_clusters.
            3. Identify stable clusters from table_clusters using the identifier_clusters_stables function.
            4. If clusters_stables is not empty or a new stable cluster is found, do:
                1. Calculate the KL divergence for each stable cluster using the calculer_divergence_clusters
                   function and update divergence_clusters_stables.
                2. Remove documents belonging to stable clusters from the corpus.
                3. Calculate the divergence of the remaining corpus using the calculer_divergence function.
                4. If the divergence of the corpus exceeds 90% of the smallest divergence of stable clusters and
                   110% of the largest divergence of stable clusters, do:
                    ▪ Identify additional stable clusters in the corpus using the identifier_clusters_stables
                      function.
                    ▪ Set continue to false to exit the loop.
            5. Otherwise, execute the k_means_stable_clusters algorithm on the corpus with updated divergence
               scores.
            6. If no stable clusters are found, increment k by 1.
    3. Output:
Return clusters_stables.
```

Figure 1: Determination of stable clusters and optimal k algorithm

```
Function identifier_clusters_stables(clusters):
    • Initialize clusters_stables as an empty list.
    • For each cluster in clusters, do:
        o If the cluster is stable for at least two iterations, add it to clusters_stables.
    • Return clusters_stables.
```

Figure 2: Identifier clusters stables function

```
Function calcule_divergence_clusters(clusters_stables):
    • Initialize divergence_clusters_stables as an empty list.
    • For each cluster in clusters_stables, do:
        o Calculate the KL divergence for the texts of the cluster using tf-idf.
        o Add the divergence score to divergence_clusters_stables.
    • Return divergence_clusters_stables.
```

Figure 3: Calculate divergence clusters function

This iterative approach allows for the efficient segmentation of the corpus into distinct clusters while minimizing the impact of noise and ensuring the stability of the formed clusters. In the next section, we will examine the results obtained using this approach.

5. PROPOSED METHOD

To evaluate our method, we chose to classify a corpus whose best categorization is known. To this end, we used a dataset from the old Al Jazeera Sport website, already exploited by our team in the Benghabrit [11] work in 2013. This corpus includes 763 articles of different sizes, covering various sporting events current at the time. These articles are divided into ten distinct sports categories.

To start our experiments, we began with a sample of 39 articles in order to better visualize and work more quickly on our approach before using the full corpus. The following table 1 presents the distribution of articles in the two experiments:

Table 1: Sports Article Distribution of the Corpus 763 Articles and Its Sample of 39 Articles.

Type of sport	***Number of articles (sample of 39)***	***Number of articles (sample of 763)***
American sports	4	83
Athletics	4	27
Boxing	4	5
Cricket	4	175
Cycling	4	38
Golf	4	245
Formula1	4	20
Football	4	36
Rugby union	4	49
Tennis	3	85

To refine our corpus and obtain better results, we first applied a word processor. For example, in the process of removing punctuation, we removed all punctuation characters from articles, such as commas, periods, and quotation marks.

Next, we proceeded to remove stop words, also known as stop words, which are common, uninformative words such as "the", "of", and "and", etc.

After that, we performed tokenization, dividing each article into a sequence of words or "tokens."

Finally, we lowercase all the texts to standardize the case of words and avoid any variation due to case.

All these steps were carried out using Python's Spacy [16] library.

Once preprocessing was complete, we used the Term Frequency-Inverse Document Frequency (TF-IDF) method to represent the articles as numerical vectors, taking into account the relative importance of terms in each article and in the entire corpus.

Finally, we developed a clustering program using the K-means algorithm with Bregman distance, notably using the NumPy[17] and scikit-learn[18] Python libraries. This program allowed us to cluster sports articles based on their textual content, with the aim of discovering underlying structures and grouping similar articles into distinct clusters.

To ensure the reliability and consistency of our findings, we rigorously maintained certain protocols throughout our experiments. Specifically, we conducted 100 iterations while keeping the random state fixed, thereby guaranteeing the reproducibility of our results across multiple runs. Additionally, to imbue the KL divergence with genuine significance, we systematically eliminated stable clusters from the entire corpus after each iteration. This meticulous approach not only enhanced the robustness of our analysis but also facilitated a clearer understanding of the dynamics at play within the clustering process.

By testing our sample with our method, we found 12 clusters, distributed as follows (table 2):

Table 2: Result obtained on our sample.

Stable Clusters	**Type of sport**	***Number of articles (sample of 39)***
Cluster 1	American sports	2
Cluster 2	Athletics	2
Cluster 3	Boxing	4
Cluster 4	Cricket	4
Cluster 5	Cycling	4
Cluster 6	Golf	4
Cluster 7	Formula1	4
Cluster 8	Football	4
Cluster 9	Rugby union	4
Cluster 10	Tennis	3
Cluster 11	American sports	2
Cluster 12	Athletics	2

When analyzing the results, we found that all sports were well categorized, but we observed an excessive number of categories compared to Al Jazeera Sport's initial categorization. In particular, two categories appeared to be over-represented: American sports and athletics.

When examining the content of these two categories in the sample, we noticed that two articles in the American sports category were about baseball and the other two were about basketball. This diversity in disciplines explains the excessive number of articles in this category. Similarly, for athletics, we found two articles on Usain Bolt, a sprint runner, and the other two on Kenyan marathon runners. This variety in disciplines also contributes to the over-representation of this category.

These observations led us to conclude that our method found an optimal number of clusters, which better corresponds to the diversity of topics present in our corpus compared to the initial categorization.

In order to compare our method to other existing methods, we chose the most classic and best-known methods for determining the optimal number of clusters: the elbow method and the silhouette coefficient. To do this, we developed a code that traces the two curves.

We started with runs of K-means ranging from k=1 to k=20, in order to cover a range of potential cluster values and observe how the elbow and silhouette coefficient curves evolve over time as a function of k.

When developing the elbow and silhouette coefficient methods, we maintained the same environment as when implementing our approach. Figures 4 and 5 represent the results of the two experiments on our sample.

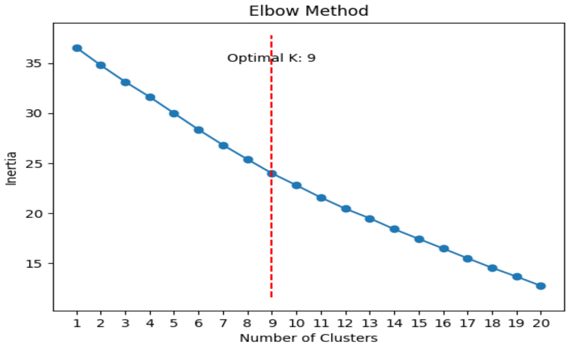

Figure 4: Result of the elbow method on sample of 39

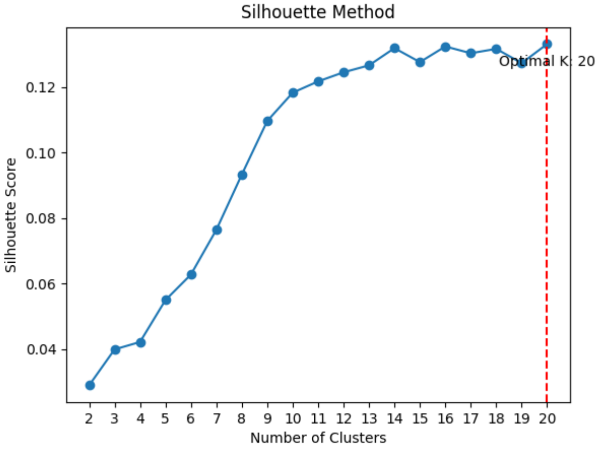

Figure 5: Result of the silhouette method on sample of 39.

Analyzing the results of the elbow and silhouette coefficient methods, we see that the results are not very clear. For example, the elbow of the elbow method is almost invisible in Figure 4, making it difficult to determine an optimal k. Regarding the silhouette coefficient method in Figure 5, we found four almost identical results with slight differences for k = 14, 16, 18, and 20, which are considered optimal according to this method. However, all of these results have silhouette indices between 0.123 and 0.126, which is considered relatively low because the best silhouette results should converge towards 1.

Looking at these results, we see a discrepancy with our prior knowledge of the actual categorization, which has 10 categories. Our method resulted in the discovery of 12 clusters, which better corresponds to the actual complexity of the corpus. Thus, the elbow and silhouette coefficient results, which suggest 9 and 20 optimal clusters, respectively, seem to deviate from true optimality.

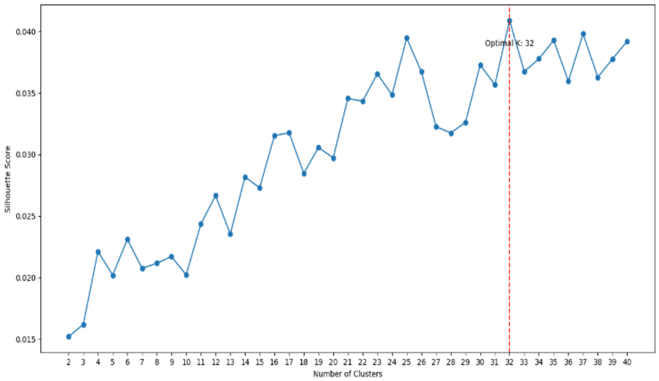

Figure 6: Result of the silhouette method on sample of 763

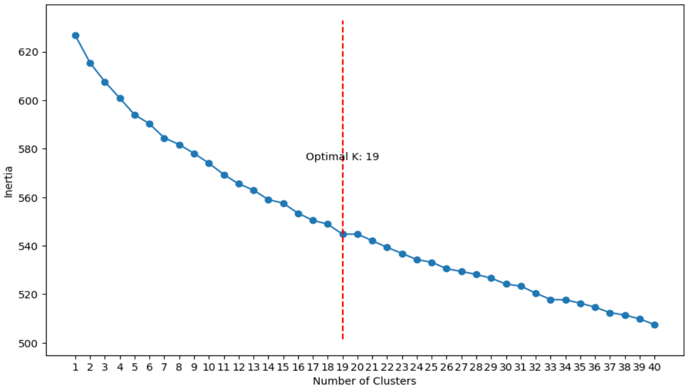

Figure 7: Result of the elbow method on sample of 763

For the test on the entire corpus, we chose to follow the same environment criteria as in the sample, maintaining a fixed random state and performing 100 iterations. However, the silhouette results were less satisfactory than in the sample, with the best result reaching only 0.039 (see Figure 6), which is very low. Similarly, for the elbow method, it was difficult to visually determine the optimal number of clusters.

Regarding our method, the results were 90% correct. We obtained 16 clusters, and in 90% of these clusters, we found texts of the same category as those identified by Al Jazeera. However, due to the large number of documents, we could not conduct an in-depth analysis of the results, which will be considered in our future studies.

6. CONCLUSION

In conclusion, our study addressed the challenges of unsupervised classification, with a particular focus on determining the optimal number of clusters in the context of text analysis. We reviewed various existing approaches, such as the elbow method and the silhouette method, as well as several innovative methods developed recently.

From these analyses, we proposed an innovative method that combines an iterative approach based on KL divergence and the TF-IDF method to determine the optimal number of clusters. This approach was implemented using the K-means algorithm with Bregman distance, specially adapted to the particularities of textual data.

Our experiments showed that our method successfully segmented texts into relevant clusters, reducing the impact of noise and ensuring the stability of the formed clusters. We validated our method on a corpus of real data, thus demonstrating its ability to produce significant results and to better reflect the diversity of subjects present in the corpus compared to an initial categorization.

In our future perspectives, we plan to compare our method with other approaches, especially spectral clustering, in order to test our approach in more complex challenges and use larger data corpora.

In summary, our approach offers a promising solution to address the complex challenge of determining the optimal number of clusters in unsupervised text analysis. It also paves the way for future research to further explore the applications and potential improvements of this method in various text data processing contexts. .

REFERENCES

1. R. Tibshirani, G. Walther and T. Hastie, "**Estimating the number of clusters in a data set via the gap statistic**,", Standford CA 94305, Standford University, 2000.
2. P. J. Rousseeuw, "**Silhouettes: A graphical aid to the interpretation and validation of cluster analysis**", J. Comput. Appl. Math, vol. 20, no. C, pp. 53-65, 1987.
3. G B. Chen, Y. Wang, Fan-Ying Gong, X. Wang and C. Yang, "**A spectral clustering algorithm for automatically determining clusters number**," Proceeding of the 11th World Congress on Intelligent Control and Automation, Shenyang, 2014, pp. 3723-3728.
4. L. Yu and C. Zhou, "**Determining the Best Clustering Number of K- Means Based on Bootstrap Sampling**," 2018 2nd International Conference on Data Science and Business Analytics (ICDSBA), Changsha, 2018, pp. 78-83.
5. A. Goyal, M. K. Jadon and A. K. Pujari, "**Spectral approach to find number of clusters of short-text documents**," 2013 Fourth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), Jodhpur, 2013, pp. 1-4.
6. C. A. Sugar and G. M. James. "**Finding the number of clusters in a dataset: an information-theoretic approach**". Journal of the American Statistical Association, 2003.

7. H. Suna, S. Wang and Q. Jiang, "**FCM-Based Model Selection Algorithms for Determining the Number of Clusters**," in Pattern Recognition 37, 2004, pp. 2027-2037.
8. G. Sanguinetti, J. Laidler, and N. D. Lawrence, "Automatic determination of the number of clusters using spectral algorithms," in Proc. IEEE Workshop Mach. Learn. Signal Process., Sep. 2005, pp. 55-60.
9. Anna Little, Alicia Byrd, "**A Multiscale Spectral Method for Learning Number of Clusters**", 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), pp.457-460, 2015.
10. Oussama Chabih et al., "**New approach to determine the optimal number of clusters K in unsupervised classification**", 6th IEEE Congress on Information Science and Technology (CiSt), 2020.
11. Asmaa Benghabrit, Brahim Ouhbi, Bouchra Frikh, El Moukhtar Zemmouri, Hicham Behja, "**Text Document Clustering with Hybrid Feature Selection**". In Proceedings of International Conference on Information Integration and Web-based Applications & Services (IIWAS '13), in Vienna, Autriche, 2013, pp. 600-605.
12. S. Lloyd, "**Least squares quantization in PCM**", IEEE Trans. Inf. Theory, vol. IT-28, no. 2, pp. 129-137, Mar. 1982.
13. L. M. Bregman, "**The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming**", USSR Comput. Math. Math. Phys., vol. 7, no. 3, pp. 200-217, 1967.
14. S. Kullback and R.A. Leibler, "**On Information and Sufficiency**," Annals of Mathematical Statistics, vol. 22, no. 1, pp. 79-86, Mar. 1951.
15. R. Baeza-Yates and B. Ribeiro-Neto, "**Modern Information Retrieval**," IEEE Transactions on Knowledge and Data Engineering, vol. 15, no. 3, pp. 833-835, May 2003.
16. **https://spacy.io** 'Industrial-strength Natural Language Processing in Python. Version 3.0.7' version 3.7.4, 2024.
17. **https://numpy.org** 'NumPy: The fundamental package for scientific computing with Python' version 1.26.4, 2024.
18. **https://scikit-learn.org/stable/** 'scikit-learn: Machine Learning in Python,' version 1.4.2, 2024.