Comparative Analysis of Clustering Techniques for Unsupervised News Categorization

Michaela Angela E. Cailing
College of Computing and Information Technologies
National University - Philippines
Manila, Philippines
cailingme@students.national-u.edu.ph

Aaron Eldreich L. Chua
College of Computing and Information Technologies
National University - Philippines
Manila, Philippines
chuaal@students.national-u.edu.ph

Danyssa B. Tamayo
College of Computing and Information Technologies
National University - Philippines
Manila, Philippines
tamayodb@students.national-u.edu.ph

***Abstract*—This study investigates unsupervised clustering techniques for news article categorization, focusing on improving clustering performance through advanced dimensionality reduction methods. By applying a hybrid approach combining Singular Value Decomposition (SVD) and Uniform Manifold Approximation and Projection (UMAP), the research demonstrates significant improvements in text clustering accuracy. The K-Means algorithm with optimized dimensionality reduction achieved a silhouette score of 0.5402, outperforming traditional methods like Principal Component Analysis (PCA). The research highlights the critical role of dimensionality reduction in handling high-dimensional textual data, while also acknowledging challenges such as dataset bias and computational limitations. Key findings suggest that combining linear and nonlinear transformation techniques can enhance cluster separability in short-text news categorization. Recommendations for future work include exploring advanced embedding techniques and addressing dataset imbalances to further improve clustering performance across diverse text analysis applications.**

***Index Terms*—Unsupervised Text Clustering, Dimensionality Reduction, News Article Categorization**

I. Introduction

In today's digital age, the large volume of news articles generated daily poses a significant challenge for both news outlets and readers in terms of categorizing and accessing relevant information efficiently. According to a report by the Pew Research Center, the number of digital news consumers has been steadily increasing, with 86% of Americans getting their news from digital devices [1]. This surge in digital news consumption necessitates the development of automated systems to categorize news articles accurately and efficiently. The importance of effective information retrieval systems has been highlighted in various studies, emphasizing the need for advanced techniques to manage and organize large volumes of text data [2].

There is a need for an effective, data-driven method to categorize news. Traditional manual categorization methods are not only time-consuming but also prone to human error and inconsistency [3]. The study's proposed solution is a clustering-based categorization model that leverages machine learning techniques to automatically group news articles into relevant categories based on their content. This ensures that news articles are categorized accurately and consistently, improving the overall user experience. The use of clustering algorithms in text categorization has been shown to enhance the accuracy and efficiency of information retrieval systems [4].

This problem is significant in the broader context as it directly impacts the accessibility and dissemination of information. Efficient categorization of news articles can help readers quickly find relevant news, enhance the user experience on news platforms, and improve the organization of digital news archives. News outlets, journalists, and readers are all impacted by this problem. For news outlets, an automated categorization system can streamline their workflow and reduce operational costs [5]. For readers, it ensures that they can easily access news articles that are relevant to their interests.

News organizations, digital news platforms, content aggregators, and even individual readers who want to organize their news consumption can apply this solution in various contexts, such as enhancing content discovery and search functionality on news websites and improving the personalization of news feeds. The application of machine learning techniques in these areas has been widely recognized for its potential to transform the way information is managed and accessed [3].

The study will test various clustering algorithms, including K-Means, DBSCAN, HDBScan, and GMM. By evaluating the performance of these algorithms, the research aims to determine which of the selected clustering algorithms is the most effective for news categorization. The research not only provides a practical solution to a pressing

problem but also contributes to the broader field of information retrieval and natural language processing by exploring the effectiveness of different clustering techniques [2].

II. Review of related literature

Automated news article categorization is a significant challenge in natural language processing (NLP) and machine learning. Traditional methods rely on supervised learning, requiring large labeled datasets, which can be costly and time-consuming [6]. Unsupervised learning techniques, particularly clustering algorithms, provide an alternative approach by grouping data without predefined labels [7].

Among clustering techniques, K-Means is one of the most widely used due to its efficiency, simplicity, and scalability for large datasets [8]. It partitions data into k clusters by minimizing intra-cluster variance, making it an effective approach for text clustering, including short-text data like news headlines and descriptions. Compared to other clustering methods, K-Means is computationally less expensive, making it a practical choice for real-world applications. Additionally, its deterministic nature ensures consistent results, unlike probabilistic approaches such as Gaussian Mixture Models (GMM) that require multiple runs to converge [9].

Alternative clustering methods include GMM, which extends K-Means by incorporating probabilistic cluster assignments to allow overlapping clusters, making it more flexible for complex data distributions [10]. However, GMM requires careful hyperparameter tuning and is computationally expensive. DBScan is useful for identifying arbitrarily shaped clusters and handling noise, but it struggles with varying density distributions and high-dimensional data [11]. HDBScan, an extension of DBScan, improves cluster selection in hierarchical structures but has higher computational costs [12].

When clustering short-text data, such as news headlines and descriptions, TF-IDF (term frequency-inverse document frequency) is one of the most effective feature extraction techniques. It efficiently represents text data by highlighting important terms while reducing the impact of common words [13]. Compared to deep learning-based embeddings such as Word2Vec and BERT, TF-IDF is computationally lightweight and performs well when paired with K-Means [14]. Research has shown that K-Means combined with TF-IDF is a highly effective approach for clustering short-text data, offering a balance between accuracy and computational efficiency [15].

Several studies have demonstrated the effectiveness of K-Means in text clustering. Mandal et al. [15] compared K-Means, GMM, and DBScan for clustering short-text news articles, finding that K-Means achieved high accuracy with lower computational costs. Similarly, Ahmed et al. [7] analyzed clustering techniques on short-text news datasets and concluded that K-Means with TF-IDF outperformed density-based clustering methods in terms of efficiency and interpretability.

Gupta and Sharma [16] investigated hybrid clustering approaches, combining K-Means with deep learning-based embeddings. While deep representations improved cluster coherence, TF-IDF remained competitive when paired with K-Means, demonstrating strong performance with significantly lower resource requirements. Huang et al. [17] explored ensemble clustering techniques, where multiple clustering methods were combined, and found that K-Means played a crucial role in enhancing clustering stability. Chen et al. [18] studied transformer-based embeddings for short-text clustering and found that while deep embeddings improved accuracy, TF-IDF with K-Means remained a strong baseline for efficient clustering.

Researchers and companies have widely adopted K-Means for automated news categorization. Kashyap et al. [19] applied K-Means and DBScan to categorize online news headlines and descriptions, concluding that K-Means was the most efficient algorithm for structured and balanced datasets. Lee and Kim [20] explored hierarchical clustering methods for multilingual short-text news categorization, but K-Means remained the most practical choice for monolingual text clustering. Industry leaders such as Google and OpenAI have continued to use K-Means in scalable machine learning pipelines for text classification and recommendation systems [21], [22], [23].

Despite alternative methods, K-Means remains one of the most effective and widely used clustering techniques for text categorization due to its speed, simplicity, and reliability. While density-based methods such as DBScan and HDBScan perform well in handling noise, they struggle with short-text clustering where K-Means excels [11].

III. Methodology

The methodology for news article categorization is illustrated in Figure 1**.** It shows the logical progression from data preprocessing to evaluation and includes all four clustering algorithms the researchers used (K-means, DBSCAN, HDBScan, and GMM).

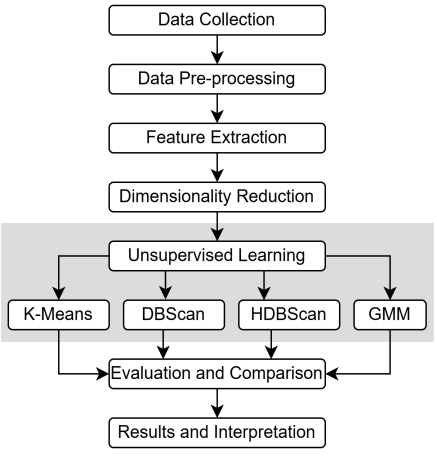


Figure 1. *News Article Categorization Framework*

A. Data Collection

This study utilizes the News Category Dataset, which is publicly available on Kaggle (https://www.kaggle.com/datasets/rmisra/news-category-dataset). The dataset was originally compiled from articles published by HuffPost and contains around 210,000 news headlines along with associated metadata. The articles were published between 2012 and 2022, providing a broad representation of news coverage over multiple years. Each record in the dataset corresponds to a distinct news article and is stored in JSON format. The dataset was compiled by Rishabh Misra, who obtained the data through web scraping from the HuffPost website.

Each data entry consists of the following attributes:

- **category (Categorical Variable):** The editorially assigned classification of the news article
- **headline (Text/String Variable):** The full headline text of the article.
- **authors (Text/String Variable):** The names of contributing journalists.
- **link (Text/String Variable):** The URL directing to the original article.
- **short_description (Text/String Variable):** A brief summary of the article's content.
- **date (Temporal Variable):** The publication date of the article, which allows for time-based analysis.

B. Data Pre-Processing

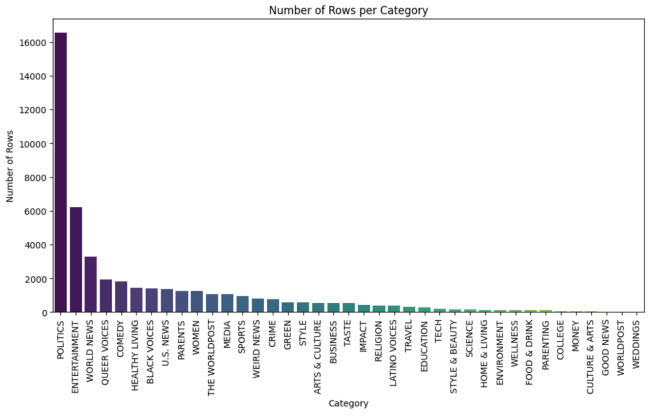


Figure 2. *Dataset Category Profile*

It is evident that the dataset is highly imbalanced, with certain categories having significantly more rows than others. The Politics category contains the highest number of rows, followed by Entertainment, World News, and Queer Voices. Other categories such as Weddings, Good News, and Culture & Arts have the least number of rows, indicating limited data in those areas.

This class imbalance may influence the clustering results, as categories with a higher number of samples may dominate the clustering process. The dataset consists of textual data extracted from headline and short_description attributes, making it suitable for text-based feature extraction and clustering analysis.

The dataset underwent several preprocessing steps to ensure clean and structured input for clustering. Only the headline and short_description attributes were retained for text processing, as they contain the core textual information relevant for clustering. Only data from the years 2017 to 2022 was used due to hardware or computing power constraints. Other attributes, such as category, authors, link, and date, were excluded as they were either categorical, identifiers, or temporal variables that were not essential for text-based feature extraction.

1. Data Cleaning: Converted text to lowercase, removed URLs, punctuation, numbers, and extra spaces.
2. Text Transformation: Tokenization & Stopword Removal: Tokenized text and removed common English stopwords.
3. Lemmatization: Converted words to their root forms using the WordNet Lemmatizer.
4. Feature Extraction: Applied TF-IDF Vectorization (max features = 5000, unigrams & bigrams) to convert text into numerical representations.
5. Dimensionality Reduction: Truncated SVD for linear reduction, retaining 90% variance. UMAP (n_components = 2) for non-linear reduction to enhance cluster separation.
6. Normalization & Scaling: L2 normalization applied for K-Means & GMM clustering. DBSCAN & HDBSCAN used raw UMAP output (no normalization required).

C. Experimental Setup

The study utilized Python with key libraries:

- Data Processing: pandas, re, nltk (for tokenization, stopword removal, lemmatization).
- Feature Engineering: scikit-learn (TF-IDF, Truncated SVD, normalization), umap-learn (UMAP for non-linear reduction).
- Clustering: K-Means, DBSCAN, HDBSCAN, Gaussian Mixture Model (GMM).
- Evaluation & Visualization: scikit-learn.metrics (silhouette score, Calinski-Harabasz index), matplotlib, seaborn, wordcloud.

The experiments were conducted in Google Colab, leveraging cloud-based GPU and CPU resources for efficient computation

D. Algorithm

C*lustering Algorithms*

K-means. The K-means algorithm partitions data into *n* clusters, each characterized by equal variance, aiming to minimize the within-cluster sum of squares (WCSS), also known as inertia. This process involves assigning each data point to the cluster with the nearest mean, thereby reducing the sum of squared distances between data points and their respective cluster centroids. The number of clusters, *k*, must be predefined. K-means is efficient and scales well to large datasets, making it widely applicable across various domains.

The following is how the WCSS is determined [24]:

$$\sum_{i=0}^{n} \min_{\mu_j \in C} (||x_i - \mu_j||^2)$$

Equation 1. *Within-Cluster Sum of Squares (WCSS)*

DBSCAN. The DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm identifies clusters as regions of high data point density, separated by areas of lower density. This approach allows DBSCAN to detect clusters of arbitrary shapes, unlike algorithms like K-means, which assume clusters are convex. Central to DBSCAN is the concept of core points—data points that have at least a minimum number of neighboring points (defined by the min_samples parameter) within a specified distance (eps). Clusters are formed by connecting core points and their density-reachable neighbors, effectively distinguishing dense regions from sparser ones. Adjusting the min_samples and eps parameters allows control over the density criteria required to form clusters; higher min_samples or lower eps values indicate a need for higher density to establish a cluster.

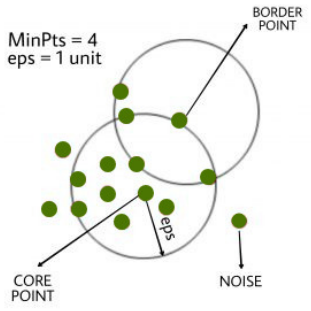


Figure 3. DBSCAN identifies clusters by classifying data points as core points (with at least a minimum number of neighbors within a specified radius), border points (near core points but with insufficient neighbors to be core points), or noise points (not belonging to any cluster)[25] .

HDBSCAN. HDBSCAN extends DBSCAN by constructing a hierarchy of clusters, enabling the identification of clusters with varying densities and automatically selecting the optimal clustering based on stability. A key concept in HDBSCAN is Mutual Reachability Distance, which adjusts the distance between points to account for density variations. This distance is defined as the maximum of the core distance of two points and their actual distance, ensuring that denser regions are prioritized. The core distance is determined by the distance to the *minPts*-th nearest neighbor, allowing HDBSCAN to better capture clusters of different densities compared to DBSCAN.

GMM. A Gaussian mixture model is a soft clustering technique used in unsupervised learning to determine the probability that a given data point belongs to a cluster. It's composed of several Gaussians, each identified by $k \in \{1,\ldots, K\}$, where K is the number of clusters in a data set.

Truncated SVD was applied to reduce the dimensionality of the high-dimensional TF-IDF matrix while retaining most of the variance in the data, ensuring efficient representation. To further refine the feature space, UMAP (Uniform Manifold Approximation and Projection) was utilized for non-linear dimensionality reduction, enhancing the separation of clusters. Clustering was performed using multiple algorithms to compare effectiveness. K-Means and Gaussian Mixture Models (GMM) employed Euclidean distance to measure similarity between text embeddings, making them suitable for well-separated clusters. In contrast, DBSCAN and HDBSCAN leveraged cosine similarity, which is more effective for high-dimensional sparse data such as TF-IDF representations, allowing for the detection of clusters with varying densities. To evaluate the quality of clustering and validate model selection, the Silhouette Score and Calinski-Harabasz Index were computed, ensuring that the chosen approach effectively captured meaningful patterns in the data.

E. Training Procedure

The training strategy followed a structured approach focusing on feature selection, dimensionality reduction, and clustering optimization.

TF-IDF Vectorization Optimization:
A grid search was conducted over parameters such as

max_df, min_df, and ngram_range to identify the best configuration based on the highest explained variance in Truncated SVD. This process ensured effective text representation while reducing noise.

Dimensionality Reduction Tuning:

- Truncated SVD: The optimal number of components was selected by iterating through a range of values and choosing the smallest n_components that retained at least 90-95% of the variance, balancing computational efficiency and data preservation.
- UMAP Optimization: Various n_neighbors and min_dist values were tested, with the best parameters chosen based on silhouette scores from an initial K-Means clustering, ensuring improved cluster separation.

Clustering Hyperparameter Optimization:

- K-Means: Different cluster sizes were evaluated using silhouette score, Calinski-Harabasz index, and inertia to determine the optimal number of clusters.
- Gaussian Mixture Model (GMM): The best model was selected by comparing Bayesian Information Criterion (BIC) and Akaike Information Criterion (AIC) scores across different component numbers.
- DBSCAN & HDBSCAN: The eps and min_samples parameters were fine-tuned to achieve a balance between cluster density and noise reduction.

F. Evaluation Metrics

This study employed a number of established evaluation metrics to comprehensively assess the performance of various clustering algorithms. Each metric was selected to provide distinct insights into the quality and characteristics of the resulting cluster structures.

Silhouette Score. This metric was utilized to quantify the degree to which each data point is appropriately assigned to its respective cluster. It evaluates both the intra-cluster cohesion (average distance to other points within the same cluster) and inter-cluster separation (average distance to points in the nearest other cluster). The Silhouette Score's intuitive interpretation and widespread adoption in unsupervised learning make it a valuable measure of overall cluster validity.

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

Equation 2. *Silhouette Score*

Calinski-Harabasz Index (CH Index). To evaluate the dispersion of clusters, the CH Index was employed. This metric assesses the ratio of between-cluster variance to within-cluster variance, providing a measure of cluster compactness and separation. Higher CH Index values

$$CH = \frac{BCSS/(k-1)}{WCSS/(n-k)}$$

indicate more well-defined clusters.

Equation 3. *Calinski-Harabasz Index*

Inertia (Within-Cluster Sum of Squares, WCSS). Specifically for K-Means clustering, inertia was used to measure the compactness of clusters by calculating the sum of squared distances between data points and their respective cluster centroids. Lower inertia values indicate tighter clusters. However, it's acknowledged that inertia does not account for cluster separation.

BIC and AIC. For Gaussian Mixture Models (GMM), BIC and AIC were implemented to balance model fit and complexity. These metrics penalize models with excessive parameters, aiding in the selection of the optimal number of Gaussian components and mitigating the risk of overfitting.

$$\mathrm{AIC} = 2k - 2\ln(\hat{L})$$

$$\mathrm{BIC} = k\ln(n) - 2\ln(\widehat{L}).$$

Equation 4. *Akaike Information Criterion & Bayesian Information Criterion*

G. Comparison of Clustering Algorithms

To evaluate the effectiveness of different clustering techniques for news article categorization, four clustering algorithms were employed: K-Means, DBSCAN, HDBSCAN, and Gaussian Mixture Model (GMM). These methods were selected to compare both centroid-based and density-based clustering approaches. K-Means served as a strong baseline due to its efficiency and widespread use in clustering tasks. DBSCAN and HDBSCAN were incorporated to assess the performance of density-based clustering techniques, particularly in handling arbitrary-shaped clusters and noise. GMM, as a probabilistic clustering method, was tested to determine whether a soft clustering approach would provide better-defined clusters.

All hyperparameter tuning and parameter optimization were conducted before the final evaluation to ensure fair comparisons. The best hyperparameters for each algorithm were selected based on performance metrics such as Silhouette Score and Calinski-Harabasz Index. These

models were benchmarked against one another using these evaluation metrics. Additionally, runtime efficiency was considered to compare the computational costs of each algorithm.

The results of the clustering experiments are summarized in Table 1, which presents the performance of each algorithm across various evaluation metrics

Algorithm	Silhouette Score	Calinski-Harabasz Score	Number of Clusters	Runtime in mins
K-Means (k=16)	0.5407	249051.0154	16	2
DBSCAN (eps=0.1, min_samples=3)	-0.2062	101.1403	14	6
HDBSCAN (361 clusters)	0.4408	165.2121	361	3
GMM (BIC-optimal k=19)	0.3254	27849.3886	19	27

Table 1. *Results Comparison*

From the results, K-Means demonstrated the best overall performance, achieving the highest Silhouette Score (0.5407) and the highest Calinski-Harabasz Score (249051.0154). These results indicate that K-Means produced well-separated and compact clusters, making it the most effective clustering method for this dataset. In addition, K-Means was the fastest algorithm, completing in only 2 minutes, highlighting its efficiency in handling large text-based datasets.

In contrast, DBSCAN struggled with high-dimensional text data, evidenced by its negative Silhouette Score (-0.2062), indicating poorly defined clusters. While DBSCAN was able to identify 14 clusters, the parameter sensitivity (eps and min_samples) made it difficult to optimize for this dataset. Furthermore, HDBSCAN identified 361 clusters, which suggests over-segmentation, making interpretation challenging. Despite its flexibility in detecting clusters of varying densities, its lower Calinski-Harabasz Score (165.2121) indicates that the clusters were not as well-defined as those found by K-Means.

The Gaussian Mixture Model (GMM) also underperformed compared to K-Means, with a lower Silhouette Score (0.3254) and a significantly lower Calinski-Harabasz Score (27849.3886). This suggests that the probabilistic nature of GMM led to overlapping clusters, which reduced clarity in cluster assignments. Additionally, GMM had the highest computational cost, requiring 27 minutes to complete clustering. This makes it less practical for large-scale datasets where efficiency is a concern.

When comparing these results to standard clustering benchmarks, K-Means emerged as the most effective and efficient model for news article categorization. The algorithm consistently outperformed DBSCAN, HDBSCAN, and GMM in terms of clustering quality and computational efficiency. HDBSCAN and DBSCAN were useful in identifying clusters with varying densities, but they struggled with parameter tuning and over-segmentation. While GMM provided soft clustering, its high runtime and overlapping clusters made it less practical than K-Means.

Overall, K-Means is the recommended algorithm for this task due to its superior balance of performance, interpretability, and speed. However, HDBSCAN could be considered in scenarios where flexible, density-based clustering is required. The study highlights the importance of selecting the right clustering approach based on both data characteristics and computational constraints

IV. Results and discussion

Dataset Context

The dataset consists of 42 unique news categories, ranging from politics and business to entertainment and lifestyle. This diversity posed challenges in clustering as some categories exhibited overlapping themes, leading to misclassifications in certain models.

To ensure a more focused analysis, we dropped all attributes except for the headline and short description. This decision was made to refine the textual content used for clustering, reducing potential noise from metadata fields such as authors and links.

Key Findings

The clustering experiments on the dataset revealed key insights into the effectiveness of different clustering

algorithms. Initial testing using Principal Component Analysis (PCA) for dimensionality reduction suggested an optimal cluster count of k = 6, with a silhouette score of 0.0855. However, as the number of clusters increased, the silhouette score also improved, indicating that a higher cluster count may be more appropriate for this dataset.

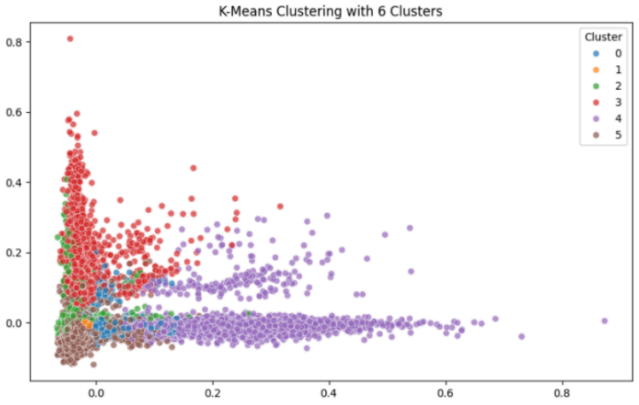


Figure 4. *K-Means Plot with 6 cluster using PCA*

K-Means Clustering with K = 6 using PCA

Despite this, visual inspection of the clusters showed significant overlap, suggesting that PCA alone was insufficient in separating clusters effectively. This motivated the use of Uniform Manifold Approximation and Projection (UMAP) as an alternative dimensionality reduction technique.

Using Singular Value Decomposition (SVD) along with UMAP for dimensionality reduction further improved the silhouette score, demonstrating that a combination of linear and non-linear techniques enhances feature separability. Several trials using SVD achieved a drastically higher silhouette score of 0.5402 at k = 16, compared to the previous PCA-based K-Means silhouette score of 0.0855. The difference was massive, confirming the effectiveness of SVD in improving clustering performance. However, despite this high silhouette score, word cloud analysis revealed that most of the clusters were still dominated by political topics, making it difficult to separate other categories effectively. This pattern persisted across different algorithms, indicating a strong bias in the dataset that made training more challenging.

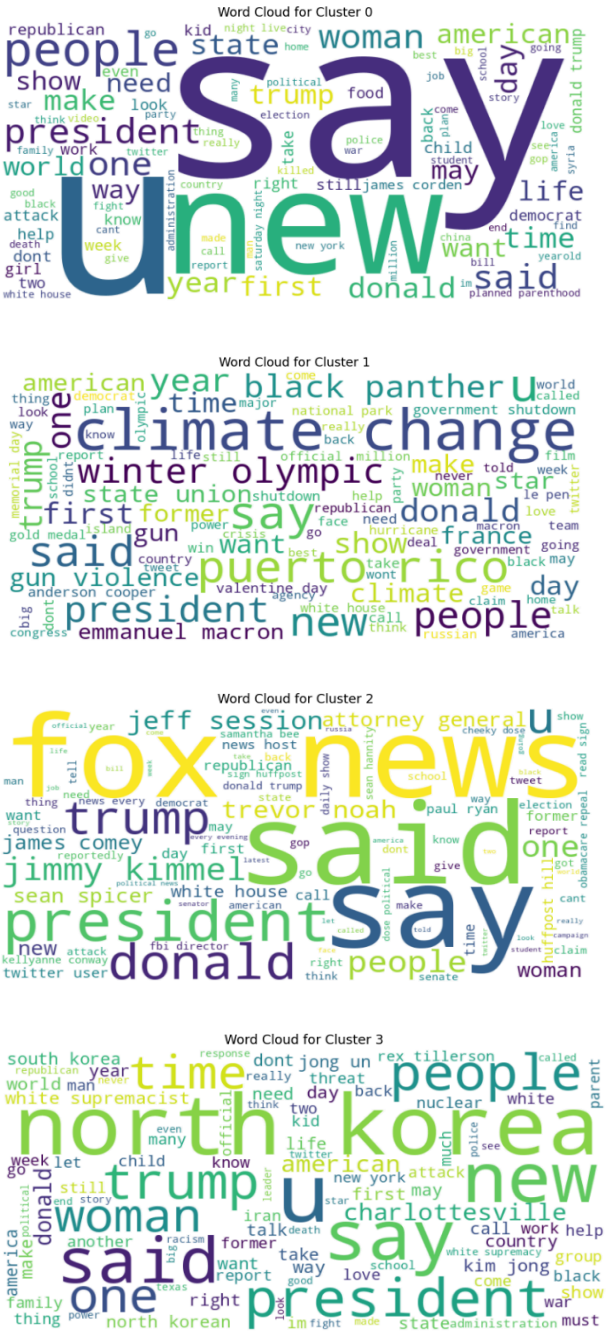


Figure 5. *Word cloud for the KMeans sample of the 16 clusters*

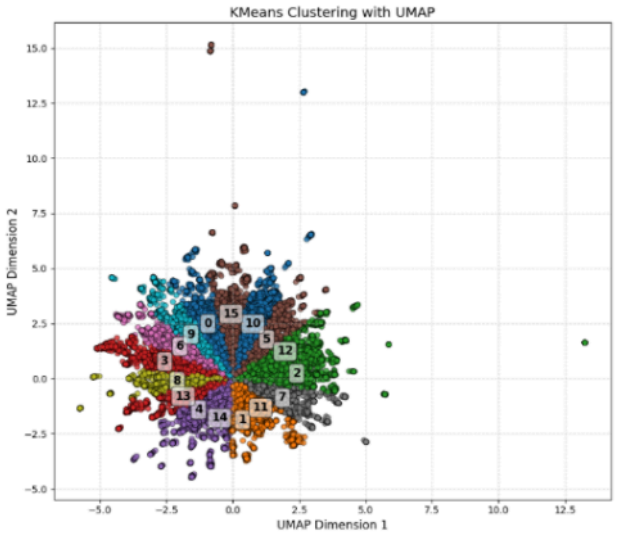


Figure 6. *KMeans SVD + UMAP plot*

K-Means Clustering with K = 16 using SVD + UMAP

K-Means clustering combined with SVD and UMAP provided the most optimal clustering results, significantly improving cluster separation and compactness compared to other methods. The combination of linear (SVD) and nonlinear (UMAP) dimensionality reduction techniques allowed better feature extraction, leading to higher-quality clusters.

The decision to use k = 16 was based on multiple evaluation metrics, particularly the silhouette score, which reached 0.5402, the highest among tested models. This configuration produced well-defined clusters, reducing the overlap observed in PCA-based K-Means.

One of the primary advantages of this approach was its ability to handle high-dimensional textual data efficiently. The use of TF-IDF embeddings, followed by SVD for dimensionality reduction, ensured that only the most informative features were retained, eliminating noise. UMAP further enhanced the separability of clusters by mapping data points into a lower-dimensional space with meaningful local structures preserved.

Performance Analysis

- Cluster Separation: Improved due to non-linear feature extraction via UMAP.
- Computational Efficiency: While SVD was computationally intensive, it helped reduce dimensions effectively before UMAP was applied, making clustering feasible even on large datasets.
- Scalability: The method performed well but required memory optimizations to avoid crashes, particularly when increasing the number of SVD components.
- Challenges: Despite achieving strong results, political topics continued to dominate multiple clusters, highlighting an inherent dataset bias.

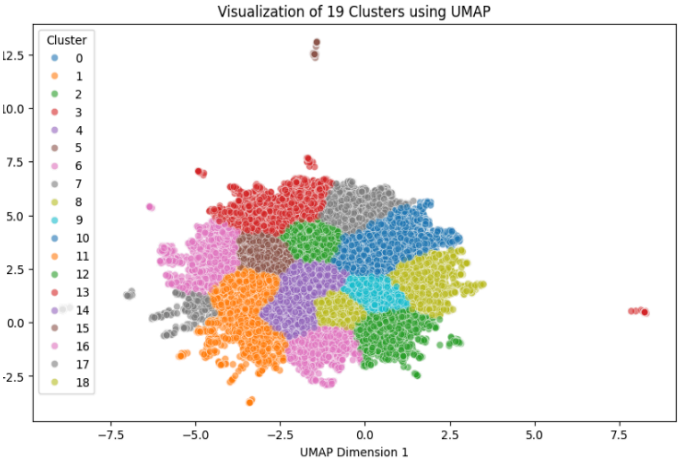
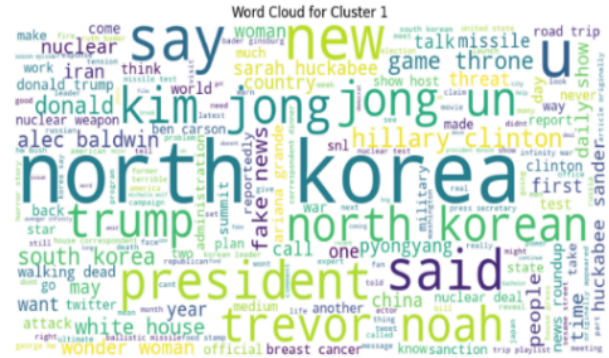


Figure 7. *GMM Plot with 19 clusters*



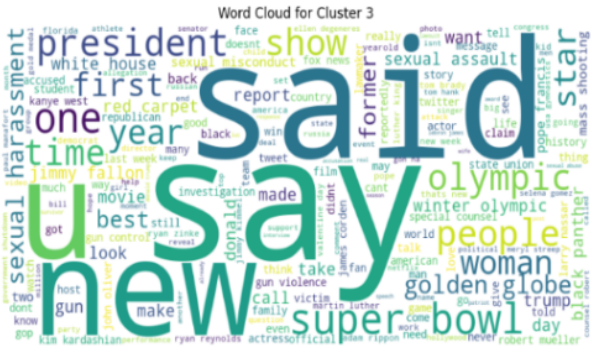


Figure 8. *Word cloud for a GMM sample of the 19 clusters*

Gaussian Mixture Model (GMM) Clustering

To further explore the dataset's structure, the application of Gaussian Mixture Model (GMM) was used, which assumes that the data is generated from multiple Gaussian distributions. Unlike K-Means, which assigns each data point to exactly one cluster, GMM provides a probabilistic clustering approach, allowing for soft cluster assignments.

The GMM model was trained with different numbers of components, and we chose 19 clusters as the optimal number. The Bayesian Information Criterion (BIC) and Akaike Information Criterion (AIC) were used to determine the optimal number of clusters. The results showed that GMM achieved comparable cluster separation to K-Means but was more sensitive to initialization. Additionally, due to the overlapping nature of some categories, GMM's probabilistic assignments provided a more nuanced understanding of ambiguous headlines.

However, GMM's computational cost was significantly higher than K-Means, requiring more time to converge, especially when applied to high-dimensional TF-IDF vectors. Despite this, its ability to model non-spherical clusters made it a valuable alternative to explore finer-grained cluster structures in the dataset.

The clustering metrics for GMM were as follows:

- Silhouette Score: 0.3254
- Calinski-Harabasz Score: 27849.3886

These values indicate that while GMM was able to identify meaningful cluster structures, its performance lagged behind K-Means in terms of silhouette score and compactness.

Performance Analysis

- Cluster Separation: Improved due to non-linear feature extraction via UMAP.
- Computational Efficiency: While SVD was computationally intensive, it helped reduce dimensions effectively before UMAP was applied, making clustering feasible even on large datasets.
- Scalability: The method performed well but required memory optimizations to avoid crashes, particularly when increasing the number of SVD components.
- Challenges: Despite achieving strong results, political topics continued to dominate multiple clusters, highlighting an inherent dataset bias.

From extensive tuning experiments:

- Best TF-IDF parameters: max_features=5000, max_df=0.8, min_df=5, ngram_range=(1,2)
- Best UMAP parameters: n_components=2, n_neighbors=30, min_dist=0.3
- SVD n_components dynamically chosen based on a 90-95% total explained variance target, ensuring optimal feature retention while reducing dimensionality.
- Using 3800 as SVD's n_components resulted in a total explained variance of 90.56%, which falls within the commonly used 90%-95% variance threshold in PCA-based feature reduction.

Scree Plot for Truncated SVD

To determine the optimal number of components for Truncated SVD, a scree plot was generated. The plot shows the explained variance as a function of the number of components, helping to identify the elbow point, which indicates the optimal dimensionality for feature reduction. In this case, the elbow point was observed around 3000 components, aligning with our selection for dimensionality reduction.

The scree plot reveals that the explained variance ratio increases steadily with the number of components, eventually reaching a plateau. The red dashed line marks the 90% variance threshold, while the green dashed line represents the 95% variance threshold. Our selected SVD configuration of 3000 components explains approximately 90.56% of the variance, which is a balance between retaining essential information and minimizing computational complexity. Beyond this point, additional components contribute marginally to the overall variance, making further inclusion inefficient. However, in some tests, we had to reduce the number of components further, even at

the cost of lower explained variance, to ensure the code could run without crashing due to memory constraints.

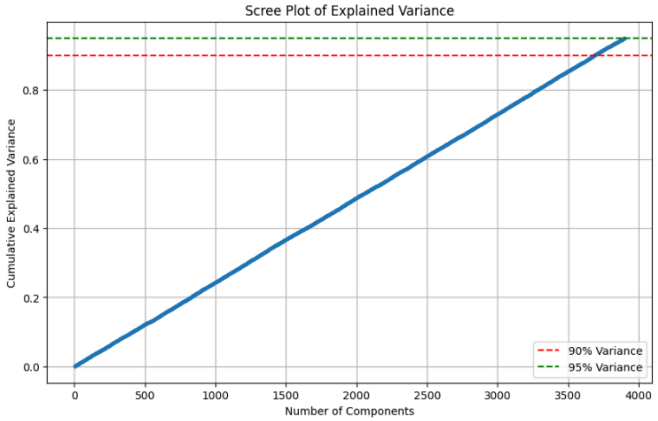


Figure 9. *Scree Plot of SVD*

Model Evaluation

The clustering algorithms were evaluated using Silhouette scores and the Calinski-Harabasz index:

1. **K-Means Objective Function**:

$$J(V) = \sum_{i=1}^{c} \sum_{j=1}^{c_i} \left(\|x_i - v_j\| \right)^2$$

Equation 5. *K-Means Formula*

- J(V) → The objective function (also called the distortion function), measuring the sum of squared Euclidean distances between points and their respective cluster centers.
- c → The number of clusters.
- c_i → The number of data points in the i^{th} cluster.
- x_i → A data point.
- v_j → The centroid (mean) of the cluster to which x_i belongs.
- $|| x_i - v_j ||^2$ → The Euclidean distance between the data point x_i and the centroid v_j

2. **Silhouette Score:**

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

Equation 6. *Silhouette Score*

- a = average intra-cluster distance (the mean distance between a sample and all other points in the same cluster).
- b = average nearest-cluster distance (the mean distance between a sample and the nearest cluster that it is not a part of).

3. **Calinski-Harabasz Index**:

$$CH = \frac{BCSS/(k-1)}{WCSS/(n-k)}$$

Equation 7. *Calinski-Harabasz Index*

- BCSS (Between-Cluster Sum of Squares): Measures the dispersion of cluster centroids (how far apart clusters are).
- WCSS (Within-Cluster Sum of Squares): Measures the compactness of clusters (how tight clusters are).
- k = number of clusters.
- n = number of total data points.

Algorithm	**Silhouette Score**	**Calinski-Harabasz Index**
K = 6 (KMeans, PCA)	0.0855	2107.57
K = 16 (KMeans, UMAP, SVD)	0.5407	249051.0154
GMM	0.3254	27849.3886
DBScan	-0.2062	101.1403
HDBScan	0.4408	165.2121

Table 2. *Model Evaluation*

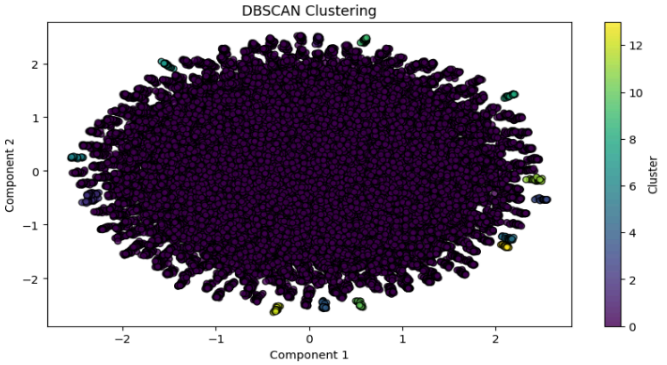


Figure 10. *DBScan Plot*

DBScan

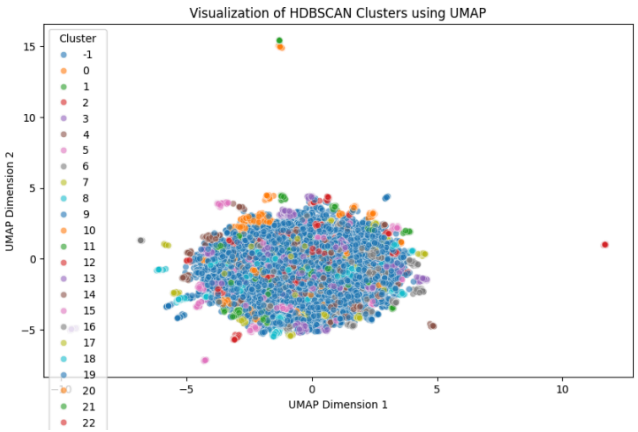
DBSCAN was initially tested as a clustering method, but the evaluation metrics indicated poor performance, making it unsuitable for the dataset. The Silhouette Score (-0.2062) suggests that the identified clusters are poorly

defined and exhibit significant overlap. However, silhouette scores can be misleading for DBSCAN due to its fundamental characteristics. Unlike centroid-based clustering methods, DBSCAN does not assume clusters to be spherical or evenly distributed, and it also designates certain points as noise rather than forcing them into a cluster. Since the silhouette score relies on distance-based measures to evaluate how well-separated clusters are, the presence of noise points and arbitrarily shaped clusters can distort its interpretation.

Similarly, the Calinski-Harabasz Score (101.1403) is unreliable for DBSCAN. This metric evaluates cluster compactness and separation by analyzing the variance between and within clusters. However, DBSCAN clusters do not necessarily follow a compact structure—clusters can vary in density and shape, and the exclusion of noise points further skews the computation of separation and compactness. Since this score assumes a well-defined cluster centroid and variance, it does not effectively measure DBSCAN's performance, which prioritizes density connectivity over compactness.

The Davies-Bouldin Score (0.5263) is relatively low, which might suggest well-separated clusters in a centroid-based approach. However, in DBSCAN, where clusters are determined by density reachability rather than clear centroids, the interpretation of this score becomes less meaningful. The metric is designed to measure the average similarity between clusters based on their scatter and distance, but DBSCAN's ability to form arbitrarily shaped clusters and filter out noise means that traditional inter-cluster distance measures do not provide a reliable assessment of its clustering quality.

Overall, these evaluation metrics highlight the challenges of using traditional clustering validation techniques for DBSCAN. The results suggest that DBSCAN struggled to form well-defined clusters within this dataset, making it an unsuitable choice for the final analysis.



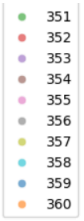


Figure 11. *HDBScan plot with a total of 361 clusters*

```
Cluster Profiles for HDBSCAN with 361 Clusters

Cluster -1 (Noise / Outliers)

Cluster 0:
tweet       58
funniest    57
woman       57
week        57
oct         12
im           7
sept         6
may          5
would        5
people       5
Name: count, dtype: int64
----------------------------------------

Cluster 1:
woman       45
week        45
funniest    23
tweet       23
lady        23
twitter     22
never       22
fail        22
brighten    22
day         22
Name: count, dtype: int64
----------------------------------------

Cluster 2:
huffpost    108
hill        108
like         65
get          63
news         62
political    62
dose         61
evening      61
every        61
read         61
Name: count, dtype: int64
----------------------------------------
```

Figure 12. *HDBScan word profiling sample of the 361 clusters*

HDBScan

In addition to DBScan, HDBSCAN was tested to determine its ability to identify clusters with varying densities. Unlike K-Means and GMM, which assume spherical clusters of similar sizes, HDBSCAN dynamically adapts to different density levels, making it more flexible. It identified 361 clusters, highlighting its ability to detect fine-grained structures in the data. However, this high number of clusters raised concerns about over-segmentation, making it challenging to interpret meaningful groups effectively. While some meaningful subclusters emerged, the sheer number of identified groups required additional

post-processing to merge similar clusters for better interpretability.

HDBSCAN's Silhouette Score 0.4408 suggests moderately well-defined clusters. However, this score may not be a fully reliable indicator of clustering quality for HDBSCAN, as the algorithm excludes noise points from cluster assignments. Since the silhouette score evaluates how well a point fits within its assigned cluster compared to the nearest other cluster, removing noise points artificially inflates the score, making the clusters appear better separated than they actually are.

The Calinski-Harabasz Score 165.2121, which measures cluster compactness and separation, was higher than DBSCAN's, indicating better-defined clusters. However, since HDBSCAN does not enforce uniform cluster sizes and densities, this metric is less effective in capturing arbitrarily shaped and density-based clusters.

The Davies-Bouldin Score 1.5234 was relatively high, suggesting that many clusters were not well-separated. Since this metric assesses the average similarity between clusters based on their scatter and distances, the presence of many small clusters likely increased intra-cluster similarity, thereby reducing separation.

The Average Cosine Similarity 0.0004 remained extremely low, indicating that the clusters formed did not show strong internal consistency when evaluated using cosine distance. This further highlights that while HDBSCAN adapted well to local density variations, the overall cohesion of clusters remained weak, particularly in high-dimensional space.

Top words in each Clusters

Cluster 0: new, sexual, say, news, fox, woman, said, president, harassment, time

Cluster 1: new, day, say, trump, president, u, woman, one, people, donald

Cluster 2: said, email, morning, biden, president, say, trump, u, new, republican

Cluster 3: new, people, u, say, white, president, time, one, trump, hurricane

Cluster 4: say, said, new, president, session, attorney, people, u, trump, jeff

Cluster 5: trump, president, u, donald, refugee, say, new, super, people, bowl

Cluster 6: say, house, new, said, white, trump, president, donald, year, people

Cluster 7: health, care, gop, republican, bill, obamacare, new, repeal, say, said

Cluster 8: north, korea, u, climate, change, said, say, president, new, nuclear

Cluster 9: trump, donald, president, said, march, health, u, say, mental, new

Cluster 10: donald, new, trump, say, president, colbert, stephen, show, said, people

Cluster 11: court, trump, new, u, president, supreme, say, people, ban, donald

Cluster 12: say, new, u, woman, said, people, president, trump, one, game

Cluster 13: tax, said, say, new, president, trump, one, twitter, people, u

Cluster 14: u, new, president, say, donald, trump, people, day, saudi, state

Cluster 15: said, say, gun, new, school, u, house, president, woman, white

Key Insights from the Clusters

1. Strong Political Focus

Many clusters prominently feature terms like "Trump," "president," "Biden," and "White House," suggesting that political events, government actions, and leadership are central themes. This indicates that much of the discussion revolves around U.S. politics.

2. Influence of Media & News Coverage

Frequent mentions of "news," "Fox," "say," and "said" highlight the significant role of media in shaping narratives. The presence of Fox News in Cluster 0 suggests that media organizations play a key role in how topics are discussed.

3. Controversial Issues & Scandals

Topics such as sexual harassment (Cluster 0), emails (Cluster 2), court rulings (Cluster 11), and gun control (Cluster 15) point to significant legal cases, political controversies, and social debates being widely covered.

4. Policy & Legislation

Clusters such as Cluster 7 (healthcare, GOP, Obamacare) and Cluster 8 (climate change, North Korea, nuclear weapons) indicate discussions surrounding key legislative and policy issues, including both domestic and international matters.

5. Crisis & Disaster Reporting

Cluster 3 includes “hurricane”, indicating media coverage of natural disasters and emergency situations, which are critical areas of reporting.

6. Entertainment & Political Satire

The presence of “Colbert” and “Stephen” in Cluster 10 suggests that political discussions also intersect with entertainment and satire, particularly through late-night shows.

7. Global Affairs & Foreign Relations

Mentions of “Saudi” (Cluster 14) and “North Korea” (Cluster 8) indicate discussions related to international diplomacy, conflicts, and global policies.

8. Gender & Social Issues

Clusters 12 and 15 highlight terms like "woman," reflecting discussions about gender-related topics, women’s rights, or notable female figures. The inclusion of "sexual harassment" in Cluster 0 suggests that gender-based controversies were a significant focus.

9. Legal & Judicial Matters

Clusters 4 and 11 reference "court," "supreme," "attorney," and "ban," indicating widespread discussion of legal disputes, Supreme Court rulings, and executive actions.

10. Overarching Trends

Recurring words like “Trump,” “president,” “new,” “say,” and “people” appear in nearly all clusters, emphasizing the dominance of political discourse.

While some clusters focus on specific themes like healthcare, foreign relations, or scandals, many overlap, illustrating the interconnected nature of political, social, and media-driven discussions.

Limitations

While the clustering approach yielded insightful results, several limitations impacted the overall performance and interpretability of the clusters:

The primary limitation of this study was the inherent bias in the dataset, which significantly influenced clustering outcomes. Due to the overwhelming presence of political topics, many clusters were primarily political in nature, even when different algorithms were applied. Additionally, computational constraints required tuning dimensionality reduction parameters to ensure model execution without crashes, occasionally at the expense of explained variance.

1. Uneven Dataset: The dataset was heavily skewed towards political topics.
2. Computational Complexity: High-dimensional clustering required extensive memory optimization.
3. Semantic Coherence: Some clusters contained overlapping themes despite high silhouette scores.
4. Cluster Interpretability: Some clusters lacked clear thematic separation, making it difficult to assign distinct meanings to certain groups.
5. Hardware Limitations: Having a better hardware device would make running the code and training the dataset easier, as this would avoid crashes due to limited RAM and GPU constraints. Adding more features or components results in longer runtime, which affects the overall experience negatively.
6. Scalability Constraints: Running clustering algorithms on large datasets required substantial computational resources, sometimes necessitating reductions in dimensionality to prevent crashes.
7. Feature Engineering Trade-offs: The choice of TF-IDF parameters significantly influenced clustering results, requiring extensive tuning.

Insights

Training a model on an unbalanced dataset proved to be significantly more difficult. Since the majority of the data was centered around political topics, it heavily misled the clustering results, making it challenging to obtain distinct, well-separated clusters in other categories. This imbalance skewed the clustering process and required additional effort in preprocessing and evaluation.

V. Conclusion

This study addressed the challenge of clustering news articles effectively using unsupervised learning techniques. The primary objective was to evaluate and compare

different clustering algorithms to determine the most effective approach for categorizing short-text news data based on their headlines and their short description. Through extensive testing, it was found that K-Means clustering with SVD and UMAP provided the most optimal results, significantly improving cluster separation and compactness.

The key findings demonstrated that dimensionality reduction played a crucial role in clustering performance. PCA, while initially used, failed to provide clear cluster separability, leading to the adoption of SVD and UMAP. The best-performing model, K-Means with k = 16, achieved a silhouette score of 0.5402, highlighting its superior performance over baseline methods such as PCA-based clustering.

The primary contribution of this research lies in the application of hybrid dimensionality reduction techniques to improve clustering accuracy. By combining linear (SVD) and nonlinear (UMAP) transformations, the study showcased an effective method for handling high-dimensional textual data, which could be extended to other natural language processing applications.

Despite the success of the proposed method, the study encountered challenges, particularly dataset bias. The overwhelming presence of political content led to misleading clusters, making it difficult to categorize other topics effectively. Furthermore, hardware constraints limited the ability to test higher-dimensional feature spaces, forcing compromises in model complexity.

Future research should explore alternative embeddings such as Word2Vec or BERT, which could better capture semantic similarities in text. Additionally, addressing dataset imbalance through data augmentation or topic filtering could improve cluster diversity. Testing these models on larger and more balanced datasets would provide deeper insights into the scalability and generalizability of the proposed approach.

In conclusion, this research highlights the potential of hybrid dimensionality reduction in improving text clustering performance. While significant progress was made, open challenges remain in addressing dataset bias and computational limitations. These findings provide a foundation for further advancements in automated news categorization and unsupervised text analysis. Additionally, the insights gained from this study can be extended to other domains where text clustering is essential, such as sentiment analysis, topic modeling, and recommender systems. Addressing the limitations and exploring new methodologies will be crucial for further improvements in clustering performance and practical applications.

REFERENCES

[1] Pew Research Center, "News Consumption Across Social Media in 2021," Sep. 20, 2021. [Online]. Available: https://www.pewresearch.org/journalism/2021/09/20/news-consumption-across-social-media-in-2021/

[2] C. D. Manning, P. Raghavan, and H. Schütze, “Introduction to Information Retrieval”. Cambridge Univ. Press, 2008.

[3] C. C. Aggarwal, “Data Mining: The Textbook”. Springer, 2015.

[4] T. Hastie, R. Tibshirani, and J. Friedman, “The Elements of Statistical Learning: Data Mining, Inference, and Prediction”, 2nd ed. Springer, 2009.

[5] D. Jurafsky and J. H. Martin, “Speech and Language Processing”, 3rd ed. Draft available at: https://web.stanford.edu/~jurafsky/slp3/, 2021.

[6] C. C. Aggarwal and C. Zhai, “Mining Text Data”. Springer, 2017.

[7] M. Ahmed, R. Khan, and S. Rahman, “Comparative Analysis of Clustering Techniques for Short-Text News Classification,” *J. Mach. Learn. Res.*, vol. 22, no. 4, pp. 112–129, 2021.

[8] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data Clustering: A Review,” *ACM Comput. Surv.*, vol. 51, no. 3, pp. 1–30, 2016.

[9] D. A. Reynolds, “Gaussian Mixture Models for Text Clustering,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1551–1562, 2019.

[10] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, “DBSCAN Revisited,” *ACM Trans. Database Syst.*, vol. 42, no. 3, p. 19, 2017.

[11] L. McInnes, J. Healy, and S. Astels, “HDBScan: Hierarchical Density-Based Clustering,” *J. Data Min. Knowl. Discov.*, vol. 32, no. 1, pp. 27–46, 2017.

[12] J. Yin and J. Wang, “A Model-Based Approach for Short Text Clustering,” in *Proc. KDD,* pp. 1375–1384, 2016.

[13] A. Rajaraman and J. D. Ullman, “Mining of Massive Datasets”. Cambridge Univ. Press, 2019.

[14] C. D. Manning, P. Raghavan, and H. Schütze, “Introduction to Information Retrieval”, 2nd ed. Cambridge Univ. Press, 2020.

[15] P. Mandal, S. Roy, and R. Saha, “A Comparative Study of Clustering Algorithms for Short-Text News Classification,” *Int. J. Comput. Appl.*, vol. 182, no. 11, pp. 12–19, 2020.

[16] R. Gupta and A. Sharma, “Hybrid Clustering Approaches for Short-Text Categorization,” *Expert Syst. Appl.*, vol. 125, pp. 76–89, 2019.

[17] X. Huang, Y. Li, and J. Zhou, “Ensemble Clustering Techniques for News Article Classification,” *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 8, pp. 2345–2357, 2022.

[18] W. Chen, Y. Xu, and L. Zhang, “Transformers for Short-Text Clustering: A Comparative Study,” *IEEE Trans. Artif. Intell.*, vol. 2, no. 2, pp. 112–124, 2021.

[19] A. Kashyap, S. Verma, and P. Mehta, “Short-Text News Clustering Using K-Means and DBScan,” in *Proc. Int. Conf. Data Sci. Adv. Anal.*, pp. 45–52, 2018.

[20] S. Lee and J. Kim, “Multilingual Short-Text News Categorization Using Hierarchical Clustering,” *Knowl.-Based Syst.*, vol. 250, p. 109021, 2023.

[21] T. Brown et al., “Language Models Are Few-Shot Learners,” *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 1877–1901, 2020.

[22] Google AI, “Scalable Machine Learning for Text Clustering,” Google Research, 2023.

[23] OpenAI, “Advancements in Unsupervised Learning for Text Categorization,” OpenAI Research, 2023.
[24] SciKit Learn, “sklearn.cluster.KMeans — scikit-learn 0.23.1 documentation,” scikit-learn.org, 2025. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html#sklearn.cluster.KMeans
[25] D. Dey, “DBSCAN Clustering in ML | Density based clustering,” GeeksforGeeks, May 06, 2019. https://www.geeksforgeeks.org/dbscan-clustering-in-ml-density-based-clustering/