Date of publication 0026 10, 2024, date of current version 0026 10, 2024.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Enhancing Document Segregation Through Adaptive Clustering and Semantic Analysis

DHANALAKSHMI RANGANAYAKULU[1], SAHAYA BENI PRATHIBA[1] (Member, IEEE), VIJAY ARUNACHALAM[2], UODIT VISHVA[2], VEERA KARTHICK[2], SURIYA PRIYA R ASAITHAMBI[3]

[1]Centre for Cyber Physical Systems, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai 600127, India; (e-mail: dhanalakshmi.r@vit.ac.in, prathiba.sbb@vit.ac.in)

[2]School of Computer Science and Engineering, Vellore Institute of Technology, Chennai 600127, India; (e-mail: apvijay.arunachalam2022@vistudent.ac.in, uodit.vishvaa2022@vitstudent.ac.in, veerakarthick.v2022@vitstudent.ac.in)

[3]Software Systems, Institute of Systems Science, National University of Singapore, Singapore 119615; (e-mail: suria@nus.edu.sg)

Corresponding author: Sahaya Beni Prathiba (e-mail: prathiba.sbb@vit.ac.in).

This work is supported by the Vellore Institute of Technology, Chennai, India.

ABSTRACT The rapid increase in image-based documents across industries like healthcare, law, and government emphasizes the need for efficient techniques to organize and extract meaningful insights from unstructured datasets. Traditional methods, including manual sorting and rule-based clustering, fail to effectively handle large-scale, noisy, and heterogeneous datasets, highlighting a significant research gap. To address this, we propose the Enhancing Document Segregation (EDS) model, a framework designed to cluster image-based datasets using a combination of Optical Character Recognition (OCR), semantic analysis, and advanced clustering algorithms. The EDS pipeline extracts text from images via OCR, preprocesses the data to eliminate noise, and generates embeddings using transformer-based models to capture semantic relationships. These embeddings are clustered using K-means, DBSCAN, Gaussian Mixture Models, and agglomerative clustering techniques to verify variable data changes. Empirical analysis demonstrates the EDS model's robustness in improving clustering accuracy and efficiency, particularly in noisy and complex datasets. Integrating theoretical foundations with practical clustering methodologies ensures the EDS model delivers a scalable solution for real-world challenges, enhancing document organization and retrieval in critical domains.

INDEX TERMS Document Clustering, Clustering Models, Image-Based Datasets, Comparative Analysis, Pattern Recognition.

I. INTRODUCTION

As the volume of image-based documents continues to grow across sectors such as healthcare, legal systems, and government services, the demand for efficient document organization and retrieval methods has become more pressing than ever. Timely access to well-segregated documents enhances decision-making, streamlines workflows, and supports informed action across critical domains. For example, clustering patient records in healthcare can enable personalized treatments and drive insights into public health trends [1]. Legal professionals benefit from rapid access to case-relevant files [2], while in government agencies, organized documentation ensures more efficient policy referencing and administrative processing [5].

Despite the importance of such systems, traditional classification methods like manual sorting or rule-based keyword matching fall short in scalability and adaptability. These conventional approaches are ill-equipped to handle noisy, unstructured, or diverse datasets and cannot capture the deeper semantic relationships present in real-world text [3].

Researchers have proposed various machine learning and deep learning models for document segregation to overcome these challenges. These include Long Short-Term Memory (LSTM) networks for classification [12], paradigmatic clustering approaches in NLP [2], and deep learning techniques for multi-page document processing [10]. Hierarchical clustering algorithms [9] and transformer-based embedding models [4] have also shown promising results in capturing textual context and enhancing clustering accuracy.

Figure. 1 illustrates a comprehensive taxonomy of existing

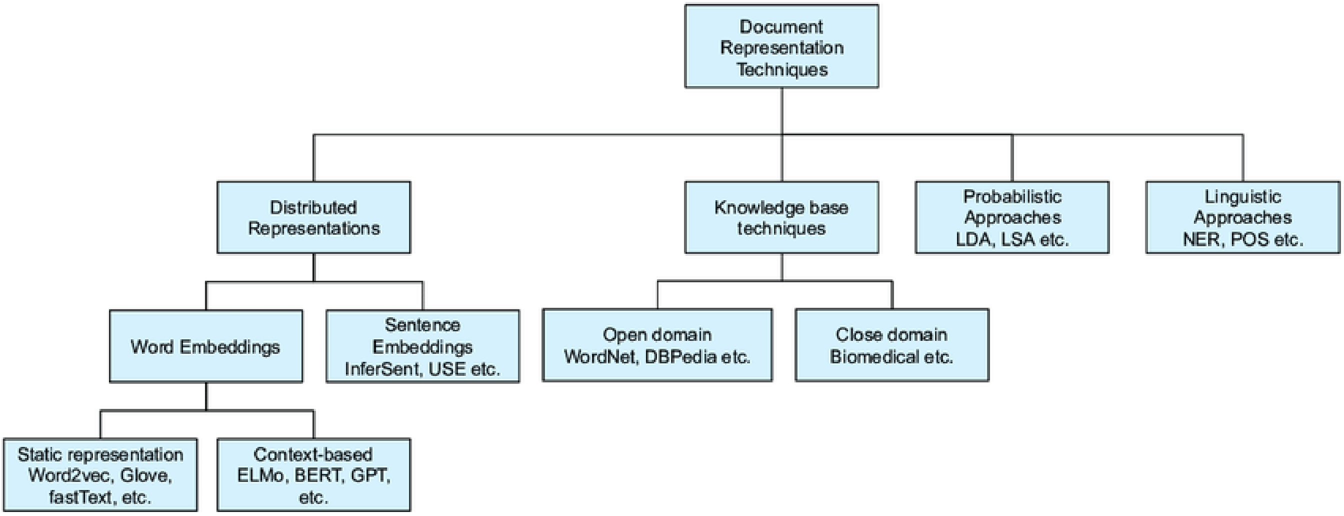

Document Representation Techniques
Distributed Representations
Knowledge base techniques
Probabilistic Approaches LDA, LSA etc.
Linguistic Approaches NER, POS etc.
Word Embeddings
Sentence Embeddings InferSent, USE etc.
Open domain WordNet, DBPedia etc.
Close domain Biomedical etc.
Static representation Word2vec, Glove, fastText, etc.
Context-based ELMo, BERT, GPT, etc.

FIGURE 1. Taxonomy of existing text-based document segregation methods

document segregation approaches. It reflects the evolution from rule-based systems to more adaptive, semantically-aware models, while highlighting current trends in unsupervised and semi-supervised learning techniques. However, these methods are still constrained by static clustering logic, limited scalability, and poor adaptability to dynamic datasets or non-standard formats.

To address these persistent gaps, this paper introduces the Enhancing Document Segregation through Adaptive Clustering and Semantic Analysis (EDS) model. EDS is a robust framework that leverages adaptive learning mechanisms, semantic embedding models, and hybrid clustering algorithms to offer a scalable and accurate solution for document classification in complex, real-world settings.

The main contributions of this work are:

- An adaptive learning pipeline for dynamically processing OCR-extracted text from scanned documents
- Integration of transformer-based embeddings to model rich semantic relationships, improving clustering accuracy
- A hybrid clustering strategy incorporates K-means, DBSCAN, and Agglomerative Clustering to accommodate varying data distributions and noise levels
- A domain-agnostic architecture proven effective across healthcare, legal, and governmental datasets, where document accuracy is paramount

The rest of the paper is organized as follows: Section II details the architecture and methodology of the proposed EDS model. Section III presents the implementation, while Section IV covers performance metrics. Section V discusses in depth the results obtained. Finally, Section VI concludes the paper and outlines future directions for research.

II. PROPOSED EDS: ENHANCED DOCUMENT SEGREGATION

The EDS model's approach to document clustering is underpinned by a sophisticated mathematical framework that accounts for the individual and collective characteristics of the processed documents. The process can be mathematically represented as follows:

$$\mathcal{I} = \mathcal{C}\left(\sum_{i=1}^{n} \alpha_i \cdot \mathcal{R}\left(\mathcal{E}(\mathbf{I}_i)\right) + \beta \cdot \mathcal{T}\left(\mathbf{I}\right)\right) \tag{1}$$

where $\mathbf{I}_i$ represents the individual text-containing images that need to be processed, where n denotes the total number of images within the dataset. The function $\mathcal{E}(\mathbf{I}_i)$ performs Optical Character Recognition (OCR) on such images and converts them from visual to textual representations. This is the critical first step, as the later stages will adopt more sophisticated processing through NLP, denoted by $\mathbf{R}\left(\mathcal{E}(\mathbf{I}_i)\right)$. It will analyze the extracted textual data, including semantic relatedness and how different textual parts of the dataset relate.

The symbol $\mathbf{I}$ represents the collective insights or clusters acquired through processing the entire dataset. It embodies the meaningful relations and patterns encompassed within the joint engagement of semantic and global transformations, thereby rendering document clustering efficient and accurate.

The semantics of the images and texts collected from different images will, therefore, be represented by a weighted value sum of $\sum_{i=1}^{n} \alpha_i \cdot \mathbf{R}$, where α_i are the weights which reflect the importance or relevance of each text segment in the overall analysis. Besides, $\beta \cdot \mathbf{T}\left(\mathbf{I}\right)$ is any other transformation application to the whole image dataset, where β is a scaling factor that is used to adapt the role of these transformations in the overall clustering.

The final clustering function will be $\mathbf{C}$, along with a good number of practical clustering algorithms. It chooses CAGR-oriented clustering algorithms to interpret different data types and cluster images in a semantic, meaningful clustering scheme. Hence, the jobs of the clusters will be

The explanation of the model's work is organized into five sections.

A. TEXT EXTRACTION

When it comes to the OCR process, at the outset of the text recognition process, the text extraction module relies on the OCR technology, which is Tesseract. In addition, such will be the process that systematically transcribes the text within images into machine-readable text and thus makes it open for further analysis and processing.

An implementation of the Tesseract would then ensure that there is high accuracy in recognition and extraction from various images. The solid capability of Tesseract concerning OCR is essential in guaranteeing precision for text extraction, which thus makes it one of the system's critical components.

Using Tesseract implies greater accuracy, which is important for applications that digitize printed or handwritten documents. Subsequently, it makes the data visible to many computational tasks, such as text analysis or information retrieval and data mining, among others, and such data access dramatically improves the efficiency or effectiveness of these processes.

B. LOSS FUNCTION

A combination of loss functions is employed to train the model: Sparse Categorical Cross-Entropy (SCCE) to improve accuracy and perceptual loss to maintain the structural coherence and perceptual quality of document clustering.

The SCCE is defined as:

$$\text{SCCE} = -\frac{1}{m}\sum_{j=1}^{m} \log q_{\tilde{y}_j} \tag{2}$$

Where:

Let m denote the total number of samples, where t_j represents the actual class label for the j-th sample, and $\tilde{y}_j$ is the predicted class for the same sample. The predicted probability of the correct class t_j is given by $q_{\tilde{y}_j}$. The summation operator $\sum$ aggregates values across all m samples.

C. PRE-PROCESSING

The pre-processing phase is crucial in standardizing the input images and ensuring they are in a suitable format for effective processing by the model. This phase involves several key steps. First, stop-word elimination filters out functional words that do not have significant meaning, thereby reducing noise in the natural language data. Next, links and numbers are removed using regular expressions, which helps eliminate irrelevant elements such as URLs and numeric values. Finally, tokenization is applied to break a text stream into individual words. These preprocessing steps are critical in guaranteeing that the text data is formatted correctly and efficiently for subsequent analysis.

D. FEATURE EXTRACTION

After extracting the text from images using OCR, the word embeddings module employs open-source BERT embeddings [11] (paraphrase-MiniLM-L6-v2) to identify relationships between images and pair them effectively. SBERT (Sentence-BERT) captures intricate patterns and semantic relationships within textual data. The module compares and matches images based on their content by encoding textual descriptions into high-dimensional vectors. This capability enhances applications in image retrieval, classification, and content recommendation by understanding semantic connections between images, significantly improving performance and accuracy.

Algorithm 1 Text Encoding and Feature Engineering using SentenceTransformer

```
Input: Preprocessed text dataset $X_{\text{text}}$ (DataFrame)
Output: Transformed feature matrix $X_{\text{features}}$ and corresponding labels $y$

Initialize the SentenceTransformer model:
  $model \leftarrow \text{SentenceTransformer('paraphrase-MiniLM-L6-v2')}$
Generate sentence embeddings from the input text:
  $X_{\text{encoded}} \leftarrow model.encode(X_{\text{text}})$
Apply normalization to each embedding vector for consistency:
for each $\mathbf{x}_i \in X_{\text{encoded}}$ do
    $\mathbf{x}_i \leftarrow \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|_{2+\epsilon}}$ {$\epsilon = 10^{-9}$ for numerical stability}
end for
Perform dimensionality reduction using Principal Component Analysis (PCA):
  $X_{\text{features}} \leftarrow \text{PCA}(X_{\text{encoded}}, k = 50)$ {Retain 50 principal components}
Extract ground-truth labels from the dataset:
  $y \leftarrow \text{extract\_labels}(X_{\text{text}})$
Output the transformed feature matrix and labels:
  Return: $X_{\text{features}}, y$
  =0
```

E. WORD EMBEDDINGS

The process of encoding text data begins with the preprocessing of the text, where the input data is cleaned and formatted into a suitable structure. The text, typically stored in a DataFrame, is then processed using a pre-trained SentenceTransformer model. In this case, the model used is paraphrase-MiniLM-L6-v2, known for its efficiency in generating sentence embeddings. The model takes the cleaned text as input and encodes it into high-dimensional vectors, which represent the semantic content of the text.

After encoding, the resulting vectors are flattened to ensure uniformity and consistency in their dimensions. These vectors are then stacked vertically to form a feature matrix, which serves as the input for further machine-learning tasks such as clustering or classification. Alongside this, the accurate labels for each data point are extracted from the DataFrame and converted into a list format to correspond with the encoded features. This feature matrix and the list of accurate labels are then returned as the final output, ready for analysis or model training.

Enhancing clustering algorithms can be achieved by incorporating dimensionality reduction techniques such as autoencoders, t-distributed Stochastic Neighbor Embedding (t-SNE), and Principal Component Analysis (PCA). These methods are essential for improving the performance and precision of clustering tasks by reducing the complexity of high-dimensional data, allowing for better analysis and visualization.

Autoencoders, a sophisticated neural network model, are widely used for unsupervised learning of efficient data representations. Autoencoders efficiently reduce input data to a lower-dimensional latent space, preserving the most significant features while eliminating extraneous noise and redundancy. The input data is reconstructed from this compressed representation, allowing for effective feature extraction. Mathematically, the autoencoder model optimizes an objective function, which may be written as follows, to reduce reconstruction error:

$$L(\mathbf{X}, \hat{\mathbf{X}}) = \|\mathbf{X} - \hat{\mathbf{X}}\|^2, \tag{3}$$

where the reconstructed output is indicated by $\hat{\mathbf{X}}$ and the original input data is represented by X .

An advanced state-of-the-art nonlinear dimensionality reduction approach is t-SNE, which is primarily used for mapping high-dimensional data to a lower-dimensional space for visualization. This technique is well-suited for finding clusters and revealing patterns in data since it focuses more on preserving local data structures. The formula for calculating pairwise similarities in the high-dimensional space, the first one of several steps that are critical in the t-SNE process, is as follows:

$$s_{ab} = \frac{\exp\left(-\|z_a - z_b\|^2 / 2\beta_a^2\right)}{\sum_{c \neq a} \exp\left(-\|z_a - z_c\|^2 / 2\beta_a^2\right)}, \tag{4}$$

where β_a is a scaling factor and z_a and z_b stand for data points in the high-dimensional space. Pairwise similarities are calculated similarly in the lower-dimensional space, but the distances are squared and scaled inversely:

$$t_{ab} = \frac{\left(1 + \|w_a - w_b\|^2\right)^{-1}}{\sum_{c \neq d} \left(1 + \|w_c - w_d\|^2\right)^{-1}}, \tag{5}$$

where w_a and w_b are the corresponding points in the low-dimensional space. The objective of t-SNE is to minimize the Kullback-Leibler divergence between the two distributions S and T:

$$\mathrm{KL}(S\|T) = \sum_a \sum_b s_{ab} \log \frac{s_{ab}}{t_{ab}}, \tag{6}$$

where S is the high-dimensional distribution and T is the low-dimensional distribution.

PCA is a linear dimensionality reduction technique that reorganizes the data into a new coordinate system, and this axis aligns along the maximum variation. The first step in this is just centering the data by subtracting the mean for each feature:

$$\mathbf{X}_{\text{centered}} = \mathbf{X} - \bar{\mathbf{X}}, \tag{7}$$

where $\bar{\mathbf{X}}$ is the mean of the data. The covariance matrix is then computed as:

$$\mathbf{C} = \frac{1}{N-1} \mathbf{X}_{\text{centered}}^T \mathbf{X}_{\text{centered}}, \tag{8}$$

where N denotes the number of samples. The principal components are obtained by initially computing the eigenvectors and eigenvalues of the covariance matrix, then multiplying the mean-centered data matrix by the eigenvector matrix.

$$\mathbf{X}_{\text{transformed}} = \mathbf{X}_{\text{centered}} \mathbf{V}, \tag{9}$$

where $\mathbf{V}$ represents the matrix of eigenvectors associated with the largest eigenvalues.

Applying these dimensionality reduction techniques transforms the data into a more manageable form, revealing significant patterns and structures. This approach lowers computational costs while improving the precision of clustering outcomes. Applying autoencoders, t-SNE, and PCA ensures that clustering algorithms work with a refined dataset, leading to better performance and more meaningful insights.

Algorithm 2 Autoencoder Pseudocode

```
Input: Data $\mathbf{x} \in \mathbb{R}^{n \times \text{input\_dim}}$
Output: Reconstructed data $\mathbf{x}' \in \mathbb{R}^{n \times \text{input\_dim}}$
1. Define input_dim, latent_dim
2. Create encoder with layers:
2.1. Linear layer: input_dim → 512
2.2. Apply ReLU activation: $f(x) = \max(0, x)$
2.3. Linear layer: 512 → latent_dim
3. Create decoder with layers:
3.1. Linear layer: latent_dim → 512
3.2. Apply ReLU activation: $f(x) = \max(0, x)$
3.3. Linear layer: 512 → input_dim
3.4. Apply Sigmoid activation: $f(x) = \frac{1}{1+e^{-x}}$
4. Pass input data $\mathbf{x}$ through encoder to obtain encoded representation $\mathbf{z}$
5. Pass encoded representation $\mathbf{z}$ through decoder to reconstruct input data $\mathbf{x}'$
6. Output: Return reconstructed data $\mathbf{x}'$
```

F. LOSS FUNCTION:

A blend of loss functions is utilized to enhance the model's performance. Mean Squared Error (MSE) ensures pixel-level accuracy, while perceptual loss is applied to preserve both structural integrity and perceptual fidelity.

The Mean Squared Error (MSE) can be formulated as:

$$\mathrm{MSE} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{Y}_i)^2 \tag{10}$$

where y_i is the actual value for the i-th data point, $\hat{Y}_i$ is the predicted value, and N is the total number of data points. The summing is then performed over all N data points.

```
Algorithm 3 Agglomerative Clustering Pseudocode
Input:
1. $X$: Feature matrix from encoded data, where $X \in \mathbb{R}^{n\times m}$ is the matrix with $n$ samples and $m$ features
2. $k$: Desired number of clusters (integer)
Start:
3. Initialize $model = \text{AgglomerativeClustering}(n_{\text{clusters}} = k)$
4. Fit the model: $model.\text{fit}(X)$
Process:
5. Obtain cluster labels: $labels = model.\text{labels\_}$, where $labels \in \mathbb{R}^n$ is a vector of cluster assignments for each sample
Output:
6. Display $labels$: The array of labels indicating the cluster assignment for each data point
```

G. CLUSTERING

Clustering is an unsupervised machine-learning method. From a statistical standpoint, clustering methods can be classified into non-parametric techniques and probabilistic model-based approaches. Examples of widely used non-parametric approaches include k-means, fuzzy c-means (FCM), and hierarchical clustering. The Gaussian Mixture Model (GMM), on the other hand, is a probabilistic model-based approach that utilizes the Expectation-Maximization (EM) algorithm to estimate the likelihood of the mixture distribution [17].

It is a key technique for grouping images based on their similarities. K-means clustering, DBSCAN (Density-Based Spatial Clustering of Applications with Noise) method, and combinations of cosine similarity with DBSCAN, Agglomerative Clustering, and Gaussian Mixture Models (GMM) are among the clustering algorithms used in EDS. These methods help in grouping the vectors into distinct, meaningful clusters.

Agglomerative Clustering is a hierarchical technique that progressively merges clusters based on their proximity, forming a tree-like structure that organizes data into clusters of varying sizes and shapes. The minimum Euclidean distance between each pair of points, one from each cluster, is the distance between two clusters, represented as $D(P, Q)$:

$$D(P,Q) = \min_{u\in P, v\in Q} \|u - v\| \tag{11}$$

where $\|\cdot\|$ is the Euclidean distance and $D(P, Q)$ is the distance between clusters P and Q.

Let μ_l, Σ_l, and π_l represent the mean vector, covariance matrix, and weight of the l-th Gaussian distribution, respectively. The collection of parameters for all L Gaussian components is given by $\Theta = \{\mu_l, \Sigma_l, \pi_l \mid l \in \{1, \ldots, L\}\}$.

Firstly, the random Gaussian parameters Θ are taken as the initial starting point. An iterative algorithm is then used, as described below, until convergence.

Expectation step: For a given Θ, compute the responsibilities of the m-th sample (the posterior probability of the l-th Gaussian distribution given a data point m) as

$$p(z_m = l \mid x_m, \mu_l, \Sigma_l) = \frac{\pi_l p(x_m \mid z_m = l, \mu_l, \Sigma_l)}{\sum_{l=1}^{L} \pi_l p(x_m \mid z_m = l, \mu_l, \Sigma_l)}, \tag{12}$$

where the m-th sample is part of the l-th Gaussian distribution when $z_m = l$, and

$$p(x_m \mid z_m = l, \mu_l, \Sigma_l) = \mathcal{N}(x_m \mid \mu_l, \Sigma_l), \tag{13}$$

where $\mathcal{N}(\cdot)$ denotes the Gaussian distribution.

Maximization step: Θ is updated by maximizing the expected complete log-likelihood given by

$$\max_{\Theta} \mathbb{E}[\ln(p(x, z \mid \Theta))] = \max_{\Theta} \sum_{m=1}^{M} \sum_{l=1}^{L} p(z_m = l \mid x_m, \mu_l, \Sigma_l) \left(\ln \pi_l + \ln \mathcal{N}(x_m \mid \mu_l, \Sigma_l)\right). \tag{14}$$

Parameter update: The estimated parameters in each iteration are given by

$$\hat{\pi}_l = \frac{\sum_{m=1}^{M} p(z_m = l \mid x_m, \mu_l, \Sigma_l)}{M}, \tag{15}$$

$$\hat{\mu}_l = \frac{\sum_{m=1}^{M} p(z_m = l \mid x_m, \mu_l, \Sigma_l) x_m}{\sum_{m=1}^{M} p(z_m = l \mid x_m, \mu_l, \Sigma_l)}, \tag{16}$$

$$\hat{\Sigma}_l = \frac{\sum_{m=1}^{M} p(z_m = l \mid x_m, \mu_l, \Sigma_l)(x_m - \mu_l)(x_m - \mu_l)^T}{\sum_{m=1}^{M} p(z_m = l \mid x_m, \mu_l, \Sigma_l)}. \tag{17}$$

If the parameters do not converge, then the parameters in each step are updated as $\{\pi_l, \mu_l, \Sigma_l\} \leftarrow \{\hat{\pi}_l, \hat{\mu}_l, \hat{\Sigma}_l\}$.

The K-means clustering algorithm can be seen as a specific instance of GMM where the covariance matrix is set as a scaled identity matrix. The GMM algorithm has proven more effective than the K-means algorithm in identifying clusters with complex ellipsoidal shapes, regardless of the cluster sizes or the distribution of data points within them [18]. Additionally, the weights are defined so that just one element weighs 1, while all other elements have weights of 0. This leads to a challenging assignment for K-means.

Algorithm 4 Gaussian Mixture Model (GMM) Clustering

```
Input:
1. $X$: Feature matrix from encoded data, where $X \in \mathbb{R}^{n \times m}$ is the matrix with $n$ samples and $m$ features
2. $k$: Desired number of clusters (integer), representing the number of Gaussian components in the mixture.
Start:
3. Initialize the GMM model:
    $model = \text{GaussianMixture}(n_{\text{components}} = k, \text{random\_state} = 0)$
    Here, $n_{\text{components}}$ specifies the number of Gaussian distributions, and random_state ensures reproducibility
4. Fit the model to the data:
    $model.\text{fit}(X)$
    This step estimates the parameters of the GMM (mean, covariance, and weight for each component)
Process:
5. Obtain cluster labels:
    $labels = model.\text{predict}(X)$
    $labels \in \mathbb{R}^n$ is a vector where each element represents the cluster assignment of the corresponding data point
Output:
6. Display $labels$: The array of cluster assignments for each sample in $X$
```

K-means Clustering

This algorithm divides the vectors into a specified number of clusters by minimizing the variance within each cluster.

Algorithm 5 K-means Clustering with Optimal K

```
Input: $X$ (feature matrix), random_state, $n_{\text{init}}$
Output: $labels$ (cluster labels), Unique labels in $labels$
1. Determine Optimal K:
    Compute WCSS for each k in range, identify elbow point
    Compute silhouette scores for each k, select k with highest score
2. Fit:
    $model = \text{KMeans}(n_{\text{clusters}} = k, \text{random\_state}, n_{\text{init}})$
    $model.\text{fit}(X)$
3. Process:
    $labels = model.\text{predict}(X)$
4. Output:
    Display $labels$, Display unique labels in $labels$
```

$$J = \sum_{i=1}^{k} \sum_{x_j \in C_i} \|x_j - \mu_i\|^2 \tag{18}$$

where:

In the clustering process, k represents the total number of clusters. Each cluster i consists of a set of points denoted as C_i. A data point within the dataset is represented as x_j, while μ_i denotes the mean of all points within cluster i. The Euclidean distance, represented by $\|\cdot\|$, is the distance metric for measuring similarity between data points.

DBSCAN

By classifying vectors according to density, this technique efficiently manages noise and finds clusters of various sizes and forms.

$$\text{Density} = \frac{|\text{Number of points within radius } \epsilon|}{\text{Total number of points}} \tag{19}$$

where

ϵ is the radius for neighborhood search,

The algorithm classifies points as core, border, or noise based on density.

Algorithm 6 DBSCAN Clustering Pseudocode with Variables and Formula

```
Input:
1. $X_{\text{transformers}}$: Feature matrix from encoded text data, where $X_{\text{transformers}} \in \mathbb{R}^{n \times m}$, with $n$ data points and $m$ features
Parameters:
2. $\epsilon$: The maximum distance between points to be regarded as neighbors (real number)
3. minPts: The minimum number of points needed to constitute a cluster (integer)
Output:
4. Cluster labels for each data point in $X_{\text{transformers}}$
5. Unique cluster labels
Start:
6. Initialize:
    Initialize the DBSCAN model with parameters: $eps$, $minPts$
    Fit the DBSCAN model on $X_{\text{transformers}}$
7. Process:
    For each pair of points $(x_i, x_j)$ in $X_{\text{transformers}}$:
        Compute the distance between the points:
            $d(x_i, x_j) = \|x_i - x_j\|$
        If $d(x_i, x_j) < \epsilon$, consider $x_j$ as a neighbor of $x_i$
    For each point $x_i$, if the number of neighbors is greater than or equal to minPts, form a cluster
8. Extract:
    Extract the cluster labels from the fitted DBSCAN model
9. Output:
    Display the unique cluster labels
    Display the cluster labels for each data point
```

Image Categorization: These clustering methods enable efficient categorization of images based on their semantic content. This advanced categorization facilitates improved image organization and retrieval, enhancing the overall utility of the image dataset.

H. COSINE SIMILARITY WITH DBSCAN

Cosine similarity is a metric that calculates the cosine of the angle between two vectors to assess how similar they are. This metric is especially useful in clustering methods such as

DBSCAN (Density-Based Spatial Clustering of Applications with Noise), particularly for high-dimensional datasets like text or images, where the vector direction is more important than its magnitude.

By computing the dot product of the vectors, normalized by the product of their magnitudes, cosine similarity calculates how similar two vectors are to one another. This metric ranges from -1 to 1, where 1 denotes identical vectors, 0 indicates orthogonality (no similarity), and -1 signifies opposed vectors.

In the context of DBSCAN, cosine similarity can be used as a distance metric to identify core points, border points, and noise in a dataset.

Algorithm 7 DBSCAN with Cosine Similarity for Document Clustering

```
Input: 1. Set of documents $D = \{d_1, d_2, \ldots, d_n\}$, where each $d_i$ represents a document
Output: 2. Document clusters, where each cluster contains documents that are closely related based on cosine similarity
Start:
3. Convert each document into a vector representation (e.g., using word embeddings or a bag-of-words model)
4. Calculate the cosine similarity between each pair of document vectors:
        $S_{ij} = \frac{d_i \cdot d_j}{\|d_i\|\|d_j\|}$
   where $S_{ij}$ is the cosine similarity between documents $d_i$ and $d_j$.
5. Construct the similarity matrix $S$ where $S_{ij}$ is the cosine similarity between documents $i$ and $j$.
6. Transform the similarity matrix $S$ into a distance matrix $D$ where:
        $D_{ij} = 1 - S_{ij}$
7. Set DBSCAN parameters:
7.1. Epsilon ($\epsilon$): The maximum distance between points to be considered neighbors
7.2. MinPts: represents the minimum number of data points necessary to define a cluster.
8. Run DBSCAN using the distance matrix $D$ with parameters $\epsilon$ and MinPts
9. Examine the clusters formed by DBSCAN:    Documents in the same cluster are closely related based on cosine similarity.
```

Mathematically, the cosine similarity between two vectors U and V is computed as follows:

$$\text{cosine\_similarity}(U, V) = \frac{U \cdot V}{\|U\|\|V\|} \tag{20}$$

Here, U and V are the two vectors, $U \cdot V$ represents their dot product, $\|U\|$ denotes the magnitude of vector U, and $\|V\|$ denotes the magnitude of vector V. The dot product is calculated as:

$$U \cdot V = \sum_{j=1}^{m} U_j V_j \tag{21}$$

The magnitudes of the vectors are calculated as:

$$\|A\| = \sqrt{\sum_{i=1}^{n} A_i^2} \quad \text{and} \quad \|B\| = \sqrt{\sum_{i=1}^{n} B_i^2} \tag{22}$$

In some clustering algorithms like DBSCAN, we need a distance rather than a similarity measure. For this purpose, we can transform cosine similarity into a distance metric:

$$\text{distance}(P, Q) = 1 - \text{cosine\_similarity}(P, Q) \tag{23}$$

This distance metric ranges from 0 (identical vectors) to 2 (opposite vectors).

A clustering algorithm called DBSCAN (Density-Based Spatial Clustering of Applications with Noise) identifies data points in dense regions and marks points in less dense regions as outliers. Two crucial criteria are needed:

The parameter δ defines the maximum allowable distance between two points to be considered neighbors. Additionally, MinPts specifies the minimum number of data points required to form a dense cluster.

The algorithm can be summarized as:

$$\text{DBSCAN}(X, \delta, \text{MinPts}) \rightarrow \text{Clusters} \tag{24}$$

Where X is the dataset. Using a cosine similarity-based distance measure, the δ-neighborhood of a point r is defined as:

$$N_\delta(r) = \{s \in X \mid \text{distance}(r, s) \leq \delta\} \tag{25}$$

This neighborhood includes all points s in the dataset X such that the distance between r and s is less than or equal to δ.

In Case 1, we employed clustering and word embeddings to identify relationships between the documents. In Case 2, we will utilize cosine similarity with DBSCAN on the vectors obtained from word embeddings to determine the relationships between the documents. Figure. 2 shows the architecture of the proposed document segregation through adaptive clustering and semantic analysis.

I. TECHNIQUES FOR IDENTIFYING THE OPTIMAL NUMBER OF CLUSTERS

1) Elbow Method

The Elbow Method [5] is a widely used approach for identifying the optimal number of clusters by analyzing the Within-Cluster Sum of Squares (WCSS). WCSS quantifies the total variance within clusters to minimize this variance.

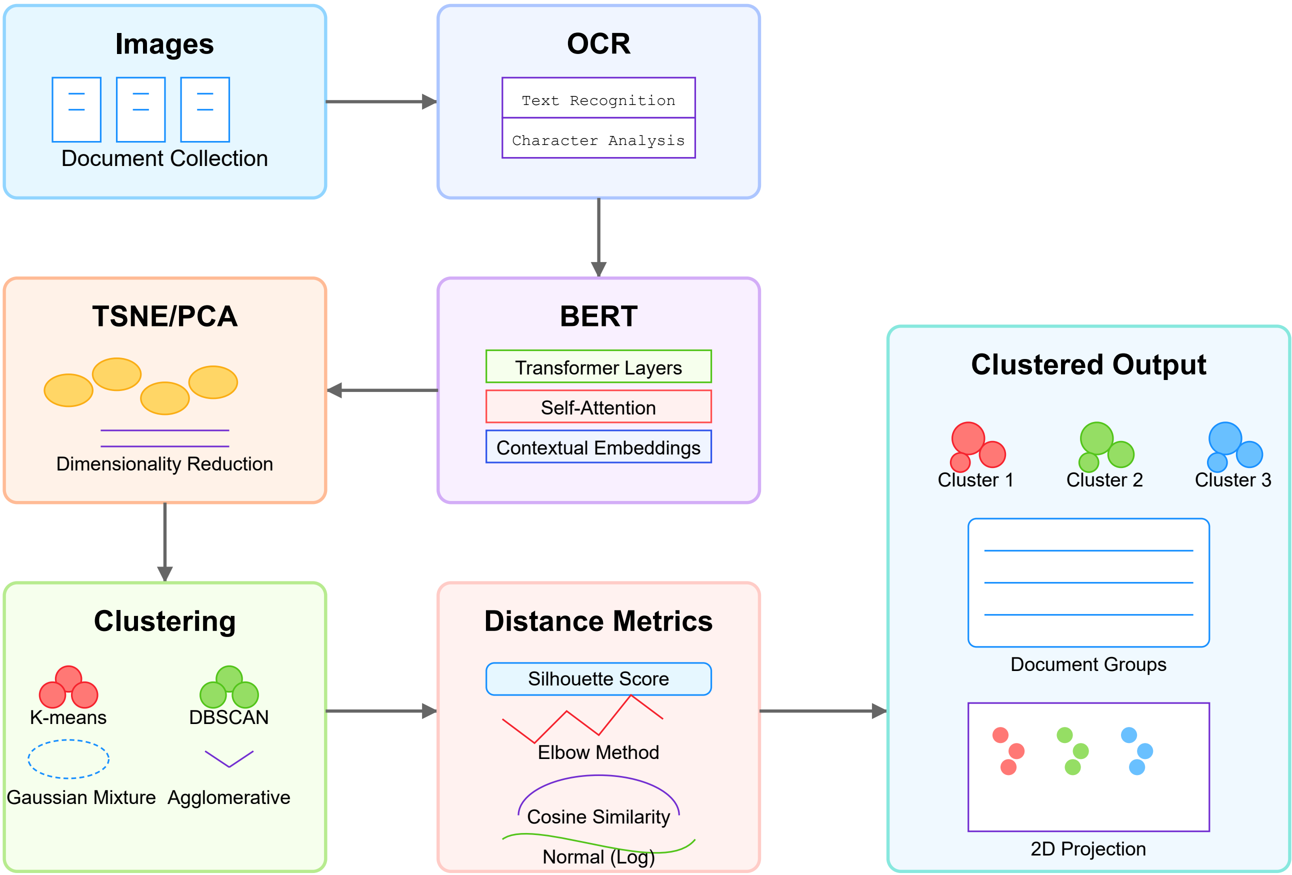


FIGURE 2. Proposed Model of Document Segregation Through Adaptive Clustering and Semantic Analysis

Procedure: The K-means clustering algorithm is applied for a range of cluster numbers, n. For each n, the Within-Cluster Sum of Squares (WCSS) is computed using the formula:

$$\mathrm{WCSS}(n) = \sum_{j=1}^{n} \sum_{y \in G_j} \|y - \nu_j\|^2 \tag{26}$$

where G_j represents the set of points in cluster j, and ν_j denotes the corresponding centroid. The WCSS values are then plotted against the respective cluster numbers. The optimal number of clusters is identified by locating the "elbow" point in the plot, corresponding to a notable reduction in the rate of WCSS decrease.

2) Silhouette Method

The Silhouette Method assesses the effectiveness of clustering by comparing how well each point aligns with its assigned cluster versus other clusters. It calculates a score based on cohesion and separation.

Procedure:

For each data point z, the mean distance to all other points within the same cluster (cohesion) is computed as $u(z)$, given by:

$$u(z) = \frac{1}{|H_p| - 1} \sum_{z' \in H_p, z' \neq z} \|z - z'\| \tag{27}$$

where H_p represents the cluster containing z. Next, the average distance to all points in the closest neighboring cluster (separation) is determined as $v(z)$, expressed as:

$$v(z) = \min_{H_q \neq H_p} \frac{1}{|H_q|} \sum_{z' \in H_q} \|z - z'\| \tag{28}$$

where H_q denotes the closest cluster to H_p. The silhouette score for each point z is then calculated as:

$$t(z) = \frac{v(z) - u(z)}{\max(u(z), v(z))} \tag{29}$$

where $t(z)$ ranges from -1 to 1, with higher values indicating better clustering. The overall clustering quality is assessed by computing the average silhouette score across all data points. Finally, the optimal number of clusters n is selected based on the highest average silhouette score.

III. IMPLEMENTATION

The model was trained on an NVIDIA RTX 3050 Ti GPU utilizing the multi-document dataset comprising 200 images.

The model exhibited remarkable performance metrics.

This paper's crucial and concluding part involves clustering, identifying similarities between images, and grouping them accordingly. After converting text from images using OCR, we apply word embeddings to discern relationships among the texts, creating high-dimensional vectors. These vectors are subsequently grouped using clustering algorithms.

In this study, we utilize various clustering methods to classify images. By reducing the variance within each cluster, the K-means method separates a dataset into a predefined number of clusters.

A. DBSCAN

DBSCAN, on the other hand, detects clusters based on the density of data points, which makes it ideal for handling outliers and spotting clusters with unusual shapes.

An alternative approach combines DBSCAN with cosine similarity as the distance metric, leveraging the cosine of the angle between vectors to assess their similarity. By adopting this integration, DBSCAN clusters vectors relative to their orientations, thus allowing surrounding semantics relationships among data points and creating even more meaningful groups.

B. GAUSSIAN MIXTURE MODELS (GMM)

Gaussian mixture models, or GMM, as it is colloquially known, refers to a statistical framework wherein the data are stated to originate from a mixture of several Gaussian distributions. These Gaussian distributions are unknown parameterized ones. Each cluster is represented as a Gaussian distribution, and its parameters are estimated using the Expectation-Maximization (EM) algorithm. The GMM successfully captures clusters of different shapes and sizes through its mixture of ellipsoidal distributions, making it a valuable tool for revealing complexly structured data.

They are clusters represented as Gaussians, the parameters of which will be estimated via the Expectation-Maximization (EM) algorithm. Gaussian Mixture Models (GMM) declare that the dataset is from a mix of several Gaussian distributions, each with unknown parameters. Therefore, GMM can fit clusters of different shapes and sizes using a mixture of ellipsoidal distributions, making it a valuable discovery tool for complexly structured data.

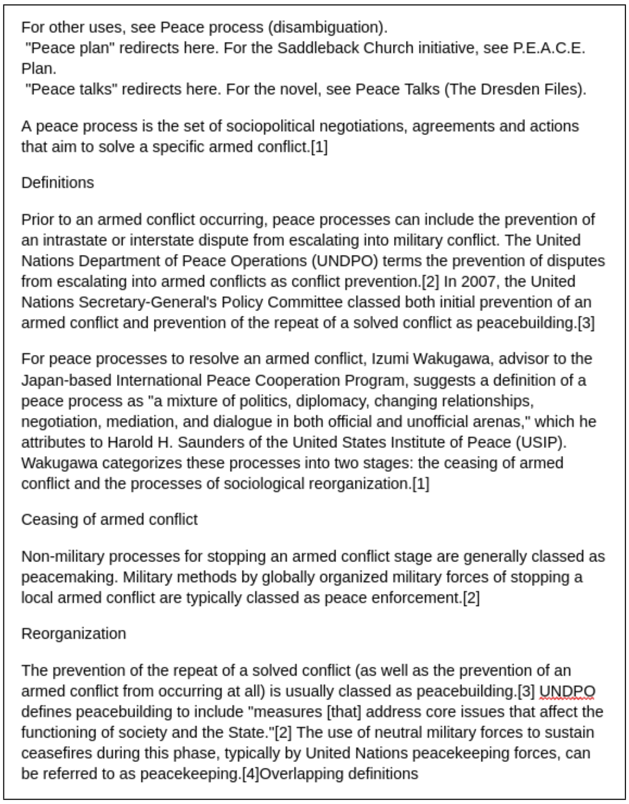
C. AGGLOMERATIVE CLUSTERING

This is a very hierarchical clustering technique called agglomerative clustering. The algorithm first treats each data point as a separate cluster. Then, it merges these clusters iteratively until a single overall cluster is achieved or until the desired number of clusters is attained. Each iteration merges the closest clusters according to a selected distance measure. This way of clustering produces a dendrogram that visually shows the links and structures of clusters at various levels, giving more insight into how the data is organized.

This approach enables us to effectively categorize images based on their semantic content, facilitating advanced image organization and retrieval. Applying these clustering methods ensures accurate capture and utilization of relationships between images for subsequent analysis. This method enhances the overall utility of the image dataset, providing a robust foundation for advanced image categorization and retrieval systems.

IV. PERFORMANCE METRICS

The evaluation of the performance of the EDS model has been completed in various states, such as text extraction, semantic embedding, dimensionality reduction, and clustering. The results show that an efficient process of analyzing the content of two correlated inter-text images is attained.



For other uses, see Peace process (disambiguation).
"Peace plan" redirects here. For the Saddleback Church initiative, see P.E.A.C.E. Plan.
"Peace talks" redirects here. For the novel, see Peace Talks (The Dresden Files).

A peace process is the set of sociopolitical negotiations, agreements and actions that aim to solve a specific armed conflict.[1]

Definitions

Prior to an armed conflict occurring, peace processes can include the prevention of an intrastate or interstate dispute from escalating into military conflict. The United Nations Department of Peace Operations (UNDPO) terms the prevention of disputes from escalating into armed conflicts as conflict prevention.[2] In 2007, the United Nations Secretary-General's Policy Committee classed both initial prevention of an armed conflict and prevention of the repeat of a solved conflict as peacebuilding.[3]

For peace processes to resolve an armed conflict, Izumi Wakugawa, advisor to the Japan-based International Peace Cooperation Program, suggests a definition of a peace process as "a mixture of politics, diplomacy, changing relationships, negotiation, mediation, and dialogue in both official and unofficial arenas," which he attributes to Harold H. Saunders of the United States Institute of Peace (USIP). Wakugawa categorizes these processes into two stages: the ceasing of armed conflict and the processes of sociological reorganization.[1]

Ceasing of armed conflict

Non-military processes for stopping an armed conflict stage are generally classed as peacemaking. Military methods by globally organized military forces of stopping a local armed conflict are typically classed as peace enforcement.[2]

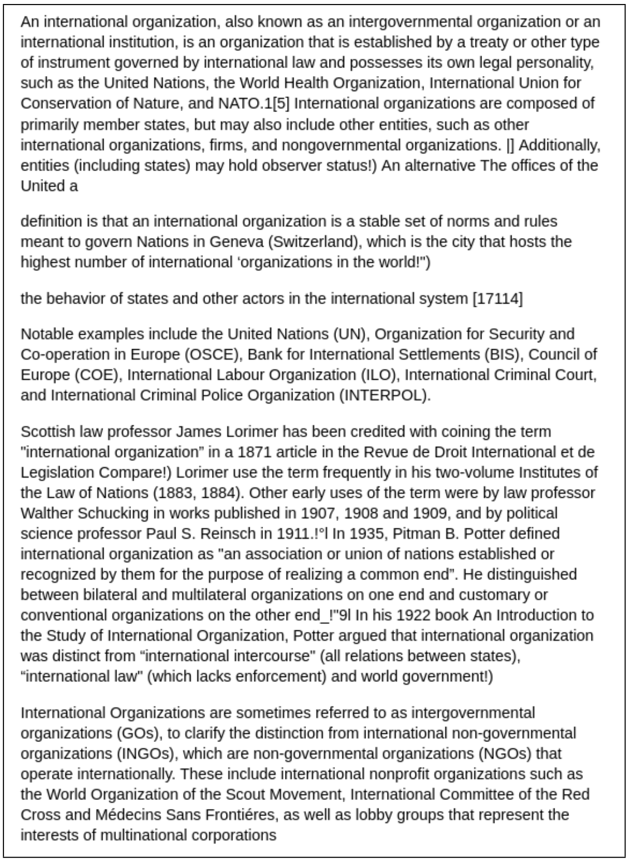
Reorganization

The prevention of the repeat of a solved conflict (as well as the prevention of an armed conflict from occurring at all) is usually classed as peacebuilding.[3] UNDPO defines peacebuilding to include "measures [that] address core issues that affect the functioning of society and the State."[2] The use of neutral military forces to sustain ceasefires during this phase, typically by United Nations peacekeeping forces, can be referred to as peacekeeping.[4]Overlapping definitions

FIGURE 3. Extracted text from Image 1: Details on peace processes.

A. TEXT EXTRACTION

The system successfully extracted textual data from two randomly selected images from the dataset clustered together by EDS. Figure. 3 detailed the concept of peace processes, including sociopolitical negotiations, conflict prevention, peacemaking, and peacebuilding. Figure. 4 focused on international organizations, their legal structures, and their roles in global governance. The coherence and completeness of the retrieved content confirmed the accuracy of the text extraction.



An international organization, also known as an intergovernmental organization or an international institution, is an organization that is established by a treaty or other type of instrument governed by international law and possesses its own legal personality, such as the United Nations, the World Health Organization, International Union for Conservation of Nature, and NATO.1[5] International organizations are composed of primarily member states, but may also include other entities, such as other international organizations, firms, and nongovernmental organizations. |] Additionally, entities (including states) may hold observer status!) An alternative The offices of the United a

definition is that an international organization is a stable set of norms and rules meant to govern Nations in Geneva (Switzerland), which is the city that hosts the highest number of international ‘organizations in the world!")

the behavior of states and other actors in the international system [17114]

Notable examples include the United Nations (UN), Organization for Security and Co-operation in Europe (OSCE), Bank for International Settlements (BIS), Council of Europe (COE), International Labour Organization (ILO), International Criminal Court, and International Criminal Police Organization (INTERPOL).

Scottish law professor James Lorimer has been credited with coining the term "international organization" in a 1871 article in the Revue de Droit International et de Legislation Compare!) Lorimer use the term frequently in his two-volume Institutes of the Law of Nations (1883, 1884). Other early uses of the term were by law professor Walther Schucking in works published in 1907, 1908 and 1909, and by political science professor Paul S. Reinsch in 1911.!°l In 1935, Pitman B. Potter defined international organization as "an association or union of nations established or recognized by them for the purpose of realizing a common end". He distinguished between bilateral and multilateral organizations on one end and customary or conventional organizations on the other end_!"9l In his 1922 book An Introduction to the Study of International Organization, Potter argued that international organization was distinct from “international intercourse" (all relations between states), “international law" (which lacks enforcement) and world government!)

International Organizations are sometimes referred to as intergovernmental organizations (GOs), to clarify the distinction from international non-governmental organizations (INGOs), which are non-governmental organizations (NGOs) that operate internationally. These include international nonprofit organizations such as the World Organization of the Scout Movement, International Committee of the Red Cross and Médecins Sans Frontiéres, as well as lobby groups that represent the interests of multinational corporations

FIGURE 4. Extracted text from Image 2: Information on international organizations.

B. SEMANTIC EMBEDDING USING SBERT

Post-extraction, SBERT embeddings were applied to the textual data. This step enabled the system to capture semantic relationships within and between the two texts. The embeddings revealed thematic congruence, with both images addressing global governance and conflict resolution, albeit from different perspectives: one through peace processes and the other through the organizational frameworks that facilitate such processes.

C. DIMENSIONALITY REDUCTION VIA AUTOENCODER

The high-dimensional embeddings were subsequently reduced using an autoencoder. This dimensionality reduction preserved the core semantic features while optimizing computational efficiency. The reduced vectors retained sufficient information to facilitate effective clustering.

D. SILHOUETTE SCORE

The clustering performance was evaluated using the Silhouette Score, which measures the cohesion within clusters and separation. The silhouette score S_p for a data point p is defined as:

$$S_p = \frac{D_Q(p) - D_R(p)}{\max(D_R(p), D_Q(p))} \tag{30}$$

where:

$$D_R(p) = \frac{1}{|G_p| - 1} \sum_{q \in G_p, q \neq p} \rho(p, q) \tag{31}$$

is the **average intra-cluster distance**, which denotes the mean distance between a point p and all other points q within the same cluster G_p, and

$$D_Q(p) = \min_{G_k \neq G_p} \frac{1}{|G_k|} \sum_{q \in G_k} \rho(p, q) \tag{32}$$

is the **mean distance to the nearest cluster**, which represents the minimum average distance between a point p and the points in the closest neighboring cluster G_k.

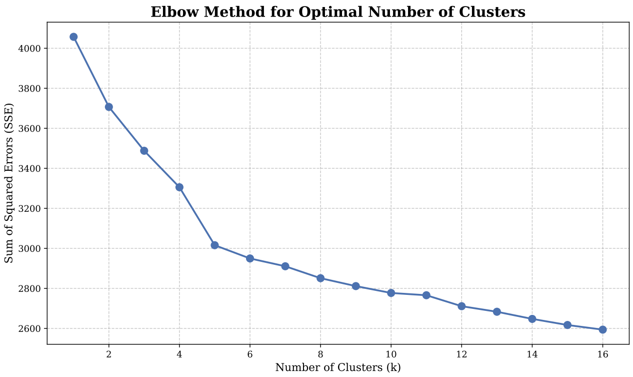


FIGURE 5. Visual representation of the Elbow method on the dataset.

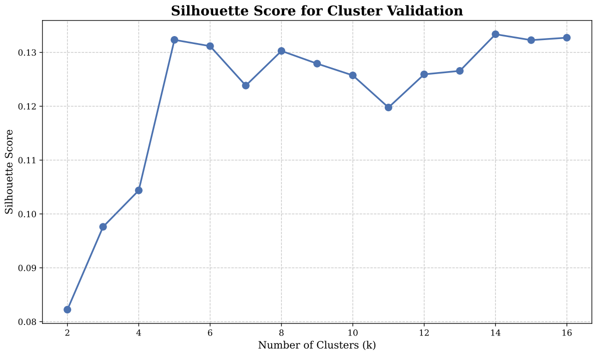


FIGURE 6. Visual representation of the silhouette method on the dataset

Here, $\rho(p, q)$ represents the distance between points p and q, while $|G_p|$ denotes the number of points in cluster G_p. The overall Silhouette Score $\overline{T}$ for the dataset is computed as:

$$\overline{T} = \frac{1}{M} \sum_{p=1}^{M} S_p \tag{33}$$

where M represents the total count of data points.

For the clustering process applied in this system, the computed Silhouette Score was:

$$\overline{T} = 0.08408122 \tag{34}$$

In the EDS (Enhanced Document Segregation) framework, a Silhouette Score close to zero indicates that the points are near the boundary between clusters, suggesting that clusters may overlap. However, this score reflects successful clustering in this context since the NLP-based approach has captured overlapping themes in textual data, such as global governance and conflict resolution. Thus, the near-zero score highlights the system's ability to handle semantically similar content effectively.

V. RESULTS AND DISCUSSION

This subsection provides the results of our experimental evaluation, which displays the output of the experimental evaluation performed on high-quality textual data. The analysis starts with determining the number of clusters through typical validation methods, i.e., the Elbow Method and the Silhouette Score. This is followed by investigating the visual results of clustering produced by different algorithms, i.e., K-Means, DBSCAN, Agglomerative Clustering, and Gaussian Mixture Models, to see how effective they are in putting together semantically sound groups. Also, the research explores semantic relations between peace processes and international organizations' concepts based on their text representations. This multidimensional assessment underlines the strengths and weaknesses of each clustering methodology while demonstrating how significant relations can be obtained from text, thus validating the applicability and soundness of the suggested approach.

A. CLUSTER VALIDATION USING ELBOW AND SILHOUETTE METHODS

Two popularly known cluster validation methods were employed to find the number of clusters with optimum performance: the Elbow Method and the Silhouette Score.

The Elbow Method is indicated in Figure 5, which is the plot of the sum of squared errors (SSE) against changing values of clusters. The SSE falls rapidly initially but then begins to level off, forming an "elbow" at a certain value of k. This inflection point, generally the optimal number of clusters, is where additional increases in clusters no longer significantly improve clustering performance. In our dataset, this elbow is notable at around $k = 6$, meaning that six clusters can model the underlying structure without overfitting.

Figure 6 illustrates the Silhouette Score method employed to determine cluster cohesion and separation. The silhouette score is higher as the clusters are more defined. It can be observed that the silhouette score is the highest at about $k = 6$, thereby confirming the result obtained using the Elbow Method. The cross-validation is performed to confirm that the number of clusters selected has tightness in the clusters and clear-cut separation between the clusters.

These methodology strategies provide empirical support for selecting the optimal number of clusters, compromising between model complexity and clustering quality. The combined analysis strengthens the validity of the clustering structure used in our proposed method.

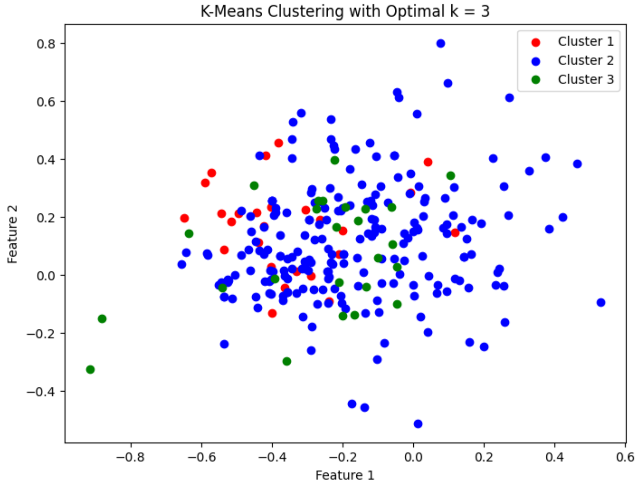


FIGURE 7. Visualization of K-Means Clustering with the Elbow Method.

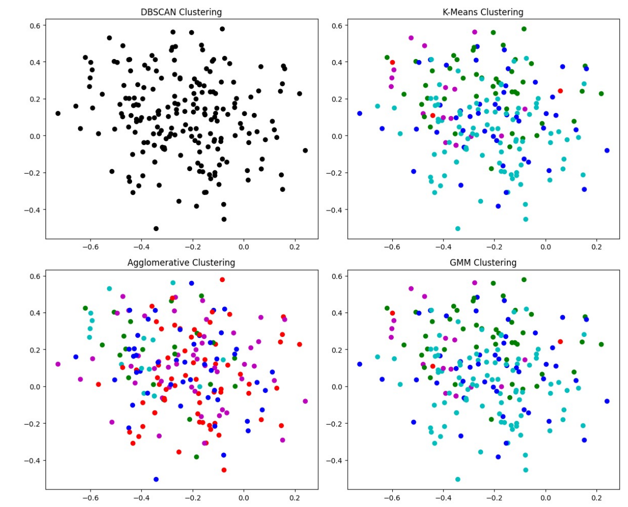


FIGURE 8. Visualization of multiple clustering models on the dataset

B. CLUSTERING OUTPUT VISUALIZATION USING ELBOW AND SILHOUETTE CRITERIA

The performance of K-Means clustering under varying optimal k-values obtained from the Elbow and Silhouette techniques is shown in Figures 7 and **??**, respectively.

Figure 7 illustrates the clustering outcome with $k = 3$, as identified using the Elbow Method. The graph indicates data point distribution among three well-differentiated groups. While groups are separable, the small number of groups can result in overlapping or generalization between semantically different texts. The above visualization implies that while $k = 3$ provides computational convenience, it could under-segment intricate thematic data.

On the other hand, Figure **??** shows the output from the Silhouette Method that suggested an optimum $k = 7$. The image depicts higher granularity with points dispersed over seven separated groups. It captures more precision and isolates finer thematic changes over the corpus, thus providing more semantic coherence over each group. Silhouette-optimal clusters depict lower overlap and better compactness, which promotes higher efficacy of this approach toward document-level semantic discrimination.

These graphical comparisons of Figures 7 and **??** empirically support a clustering approach consistent with silhouette-based optimization in situations involving fine-grained semantic separation.

C. COMPARATIVE VISUALIZATION OF CLUSTERING ALGORITHMS

A visual analysis was conducted utilizing DBSCAN, K-Means, Agglomerative Clustering, and Gaussian Mixture Models (GMM) to assess the efficacy of different clustering models on semantically rich textual data. The resulting clusters are illustrated in Figure 8, providing insight into how each algorithm distributes data points within the reduced feature space.

DBSCAN does not detect clear clusters in the top-left quadrant, marking most points as noise. This reflects DBSCAN's sensitivity to parameter adjustment and its weakness when dealing with overlapping or dense clusters with ambiguous boundaries, like those found in semantic textual data.

The top-right figure illustrates K-Means clustering, which clearly segments the data into distinct clusters. While some overlap is evident, the clustering demonstrates a good central tendency, which suits situations with relatively spherical cluster shapes.

The bottom left illustrates Agglomerative Clustering, which exhibits tighter and more compact cluster formations. This technique preserves hierarchical structure and better suits datasets with embedded or hierarchical semantics.

Lastly, the bottom-right quadrant represents the output of GMM, which describes data as a mixture of Gaussians. It shows smooth and overlapping boundaries of clusters and hence is appropriate for capturing probabilistic relationships and soft cluster assignments among complicated textual features.

This comparative visualization highlights the potential and limitations of each model. While K-Means and Agglomerative Clustering appear promising in detecting different semantic patterns, GMM is flexible enough for overlapping themes. DBSCAN, while good at some spatial applications, performs poorly with high-dimensional, semantically rich data.

D. COMPARATIVE ANALYSIS OF CLUSTERING ALGORITHMS

Table 1 compares the clustering algorithms employed in this research, their performance, strengths, and weaknesses. K-means, Agglomerative Clustering, and Gaussian Mixture Models (GMMs) performed exceptionally well when applied to semantically consistent document clusters. K-means worked best with large datasets due to its simplicity and low computational overhead, while Agglomerative Clustering could detect hierarchical patterns and fine-grained relationships in the data.

TABLE 1. Performance Comparison of Clustering Algorithms

Clustering Algorithm	**Performance**	**Strengths**	**Weaknesses**
K-means	Excellent	Efficient for large datasets, accurate grouping of related documents	Requires predefined number of clusters (K)
Agglomerative Clustering	Excellent	Hierarchical structure captures nuanced relationships	Can be computationally expensive for large datasets
Gaussian Mixture Models	Excellent	Handles complex cluster shapes, probabilistic assignment	Sensitive to initialization, requires predefined number of clusters
DBSCAN	Good	Finds arbitrarily shaped clusters, handles noise	Struggles with clusters of varying density, sensitive to parameter settings
Cosine Similarity + DBSCAN	Fair	Works well with document embeddings, flexible with cluster shapes	Less effective for subtle semantic distinctions

GMMs worked well in handling overlapping clusters by assigning label probabilities to data points, but were initialization parameter sensitive. DBSCAN performed optimally in identifying clusters with non-linear boundaries and identifying outliers, but performed poorly with datasets containing clusters of varying densities and required parameter tuning. Combining Cosine Similarity with DBSCAN strengthened it, but it could not detect subtle semantic variations.

Based on these results, K-means, Agglomerative Clustering, and GMMs were the most appropriate algorithms for clustering operations with semantically rich text-based data.

E. SEMANTIC RELATIONSHIP BETWEEN PEACE PROCESSES AND INTERNATIONAL ORGANIZATIONS

Figures 3 and 4 illustrate the textual information of two alternative sources, based on the concepts of peace processes and international organizations, respectively. Table 2 outlines the correspondence among key terms of both pictures to analyze the semantic connection between sociopolitical mechanisms seeking conflict resolution and institutions in and through which they operate.

The initial source is a comprehensive perspective of peace processes, focusing on multi-dimensional activities like peacebuilding, peacemaking, conflict prevention, peacekeeping, and peace enforcement. The processes express proactive and reactive efforts to armed conflict, like ceasefire agreements, negotiations, and reengineering of society in the long term.

The second source provides a descriptive overview of international institutions, categorizing them by their legal nature, international administrative instruments, and their primary activity of facilitating collaborative action between governments. Notable examples, such as the United Nations (UN), World Health Organization (WHO), and NATO, are distinguished by their functional role in coordinating global security, health, and peacekeeping activities.

Table 2 shows the intersections between specific aspects of peace processes and the mandates and activities of major international agencies. The UN, for example, is most prominent in peacemaking and conflict prevention, while the WHO is associated with peacebuilding in the context of health-oriented interventions in states of conflict resolution. NATO is associated with peace enforcement and theater-level stabilization operations.

TABLE 2. Relation Between Words in Figure 3 and Figure 4

Word from Figure 3	**Word from Figure 4**	**Relationship**
Peace process	*International organization*	Both involve global governance mechanisms
Conflict prevention	*UN (United Nations)*	The UN engages in conflict prevention
Peacebuilding	*World Health Organization*	WHO's activities can contribute to peacebuilding
Peacemaking	*NATO*	NATO's role includes peacemaking in conflicts

This research highlights the importance of the interdependence of theoretical peace paradigms and international institutions' applied functions. Identifying these semantic relations enhances text modeling in textual data for applications of clustering and classification in natural language processing pipelines.

VI. CONCLUSION

This paper explores applying techniques such as K-means, DBSCAN, DBSCAN with cosine similarity, Agglomerative Clustering, and GMM in the processing of images after Optical Character Recognition (OCR) and word embeddings. Our experimental results showed improved quality and accuracy in clustered image data by combining spatial and semantic information from OCR and embeddings. K-means and DBSCAN, in particular, facilitated nuanced clustering, enhancing data organization and extraction, which is crucial for applications like document analysis and content-based image retrieval. The results showed that both methods could effectively group similar images and documents, improving the retrieval and analysis processes.

However, DBSCAN's performance [21] [20] was suboptimal due to its sensitivity to parameter settings, such as epsilon and minPts, leading to poor differentiation between clusters and noise, particularly in high-dimensional spaces. The inability of DBSCAN to handle clusters of varying densities in such spaces also impacted its effectiveness. However, K-means showed more consistent results, so the challenge of determining the optimal number of clusters persists. Future work will focus on optimizing DBSCAN for high-dimensional data, exploring hybrid models that combine clustering with advanced feature extraction techniques, and addressing scalability challenges for large-scale image datasets. Developing models that can adjust to diverse data distributions and efficiently handle large datasets will be essential to improve real-world applications.

REFERENCES

[1] S. Bakkali, S. Biswas, Z. Ming, M. Coustaty, M. Rusiñol, O. R. Terrades, and J. Lladós, "TransferDoc: A Self-Supervised Transferable Document Representation Learning Model Unifying Vision and Language," arXiv preprint arXiv:2309.05756, 2023.

[2] A. Arora, X. Yang, S.-Y. Jheng, and M. Dell, "Linking Representations with Multimodal Contrastive Learning," arXiv preprint arXiv:2304.03464, 2023.

[3] M. Polewczyk and M. Spinaci, "ClusterTabNet: Supervised clustering method for table detection and table structure recognition," arXiv preprint arXiv:2402.07502, 2024.

[4] D. DK, "Topic Modelling with BERTopic," Medium blog, 2024. [Online]. Available: https://medium.com/@danushidk507/topic-modelling-with-bertopic-249095144555

[5] J. Doe, "K-Means Clustering using PySpark," Machine Learning Plus, 2023. [Online]. Available: https://www.machinelearningplus.com/pyspark/pyspark-mllib-k-means-clustering/

[6] P. K. Sharma and A. Gupta, "Document classification using transformers: Advances and applications," *Journal of Artificial Intelligence Research*, vol. 5, no. 1, pp. 15-28, 2023.

[7] A. Petukhova, J. P. Matos-Carvalho, and N. Fachada, "Text clustering with LLM embeddings," *arXiv preprint arXiv:2403.15112*, 2024. [Online]. Available: https://arxiv.org/abs/2403.15112

[8] R. K. Paladugu and G. R. Kancherla, "Harnessing Deep Learning Techniques for Text Clustering and Document Categorization," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, no. 8, pp. 125ˇ2013139, 2023. https://doi.org/10.17762/ijritcc.v11i8.7930

[9] A. Agarwal, S. Khanna, H. Li, and P. Patil, "Sublinear Algorithms for Hierarchical Clustering," *arXiv preprint arXiv:2206.07633*, 2022. [Online]. Available: https://arxiv.org/abs/2206.07633

[10] R. Singh, V. Sharma, R. Kashyap, and M. Manwal, "Automated Multi-Page Document Classification and Information Extraction for Insurance Applications using Deep Learning Techniques," in *Proceedings of the 11th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO)*, Noida, India, 2024, pp. 1-7, doi: 10.1109/ICRITO61523.2024.10522111.

[11] Z. Jiang, Q. Liu, and H. Sun, "PromptBERT: Improving BERT Sentence Embeddings with Prompts," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022, pp. 8826–8835. [Online]. Available: https://aclanthology.org/2022.emnlp-main.600

[12] N. Masuyama, Y. Nojima, C. K. Loo, and H. Ishibuchi, "Multi-label Classification via Adaptive Resonance Theory-based Clustering," *arXiv preprint arXiv:2103.01511*, 2021. [Online]. Available: https://arxiv.org/abs/2103.01511

[13] T. Balaji, V. Khanna, and T. Nalini, "A Hybrid Machine Learning Approach for Document Classification: A Comparative Study," in *Proceedings of the 2nd International Conference on Edge Computing and Applications (ICECAA)*, Namakkal, India, 2023, pp. 1198-1201, doi: 10.1109/ICECAA58104.2023.10212421.

[14] *Wikipedia*, https://www.wikipedia.org/.

[15] A. Akdoğan and M. Kurt, "ExTTNet: A Deep Learning Algorithm for Extracting Table Texts from Invoice Images," *arXiv preprint arXiv:2402.02246*, 2024. [Online]. Available: https://arxiv.org/abs/2402.02246

[16] L. Dhulipala, J. Lee, J. Łącki, and V. Mirrokni, "TeraHAC: Hierarchical Agglomerative Clustering of Trillion-Edge Graphs," *arXiv preprint arXiv:2308.03578*, 2023. [Online]. Available: https://arxiv.org/abs/2308.03578

[17] J. Wang and J. Jiang, "An unsupervised deep learning framework via integrated optimization of representation learning and GMM-based mod-

eling," *Neurocomputing*, vol. 433, pp. 199-211, 2021. [Online]. Available: https://doi.org/10.1016/j.neucom.2020.12.082

[18] A. Alipour Yengejeh, "Optimizing AI with Advanced Data Structuring: A Comparative Analysis of K-means and GMM Clustering Techniques," *Data Science and Data Mining*, vol. 21, 2024. [Online]. Available: https://stars.library.ucf.edu/data-science-mining/21

[19] B. Knisely and H. H. Pavliscsak, "Research proposal content extraction using natural language processing and semi-supervised clustering: A demonstration and comparative analysis," *Scientometrics*, vol. 128, no. 1, pp. 1-20, 2023. [Online]. Available: https://doi.org/10.1007/s11192-023-04689-3

[20] Y. Wu, X. Zhang, and L. Wang, "A fast parallelized DBSCAN algorithm based on OpenMP for large-scale scientific data," *Frontiers in Big Data*, vol. 6, p. 1292923, 2023. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fdata.2023.1292923/full

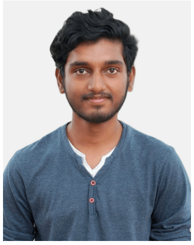
[21] G. Mo, S. Song, and H. Ding, "Towards Metric DBSCAN: Exact, Approximate, and Streaming Algorithms," *arXiv preprint arXiv:2405.06899*, 2024. [Online]. Available: https://arxiv.org/abs/2405.06899

[22] A. Alam, "Hybridization of K-means with improved firefly algorithm for automatic clustering in high dimension," *arXiv preprint arXiv:2302.10765*, 2023. [Online]. Available: https://arxiv.org/abs/2302.10765

DHANALAKSHMI RANGANAYAKULU is a Ph.D holder from Anna University Chennai in Information Security and Networking. She holds a B.E degree in Computer Science from Bharathidasan University and an M.Tech in Advanced Computing from SASTRA University. She has a rich academic experience of more than 25 Years. She has vital research experience as a Research Associate in the NTRO-sponsored project, and she collaborated to direct basic research on Smart and Secure Environment at Anna University under the consortium of IIT Madras. She is an associate professor at the School of Computer Science and Engineering. She is also associated with the Research Center for Cyber Physical Systems, Vellore Institute of Technology (VIT), Chennai Campus. She has numerous research papers in reputed Journals and conferences, including Elsevier, Springer, IEEE, IFIP, and IGI Global. She has organized and chaired the sessions of International Conferences and Workshops. She is a Certified Network Administrator. She has received various awards, including the IET CLN Women Engineer Award, Best Teaching Faculty, and Best Mentor. She is involved in various industrial Projects, is a Trainer, and is an Active collaborator with MNCs such as HITACHI, Mind Tree, SERVION, RANE, FLDSmidth, and GSLab|GAVS Technologies.

SAHAYA BENI PRATHIBA (Member, IEEE) received Bachelor's and Master's degrees in Computer Science and Engineering from Anna University, Chennai. She has secured 23rd rank among 2581 candidates in the Master of Engineering. She completed her Ph.D. in 2022 at Anna University, Chennai. She is an Assistant Professor at the Centre for Cyber Physical Systems, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai. She was also a recipient of the Anna Centenary Research Fellowship. Her research interests include 5G/6G, Vehicle-to-Everything, Software Defined Networking, Autonomous Vehicular Networks, Industry 5.0, and Metaverse. Reputed journals like IEEE, Elsevier, Springer, and MDPI recognized her research. She also serves as a reviewer for IEEE TRANSACTIONS and reputed Elsevier journals.

VIJAY ARUNACHALAM is currently pursuing a B.Tech in Computer Science and Engineering with a specialization in Cyber-Physical Systems at Vellore Institute of Technology (VIT), Chennai, India. His research interests include machine learning and natural language processing(NLP), focusing on LLM-based agents, AI agents, and chatbots.



UODIT VISHVA is a B.Tech student at Vellore Institute of Technology (VIT), Chennai, India, specializing in Artificial Intelligence and Machine Learning. His research focuses on NLP and Image Processing.

VEERA KARTHICK is a B.Tech student at Vellore Institute of Technology (VIT), Chennai, India, specializing in Artificial Intelligence and Machine Learning. His research interests include Natural Language Processing (NLP) and Federated Learning, focusing on LLM-based systems, AI agents, and chatbots.

SURIYA PRIYA R ASAITHAMBI is working as a Member at the Institute of Systems Science, National University of Singapore. She designed customised courses for industry professionals and MTech students in emerging topics. She consults in the areas of Data Engineering and Software Engineering. Her twenty-four-year-long teaching and consulting portfolio includes (1) Evaluating various reference architecture and choosing between hybrid design alternatives (2) Building customised technical stacks that satisfy given business needs using a combination of cloud native managed services, full-stack frameworks, libraries, programming languages, API endpoints, and related testing tools), and (3) Building analytics solutions to provide operational and strategic business insights using mathematical, data ETL pipelines, machine learning algorithms, and data visualisation libraries. She holds a PhD from the School of Computing, National University of Singapore. She is a certified professional in TOGAF, GCCC, and IBM products. Her current research emphasis is on cloud computing and data engineering.