IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, November 2026
Impact Factor: 8.2

AI-Based Document Clustering System

Rohan Sopan Pawar, Yogesh Bharat Satalkar, Sakshi Sunil Gaikwad,
Alka Ashok Khandagle, Dr. P. P Rokade
Department of Information Technology
SND College of Engineering & Research Center, Yeola, Maharashtra, India

Abstract: *The rapid growth of digital information has increased the volume of unstructured documents that users must store, organize, search, and analyse. Manual classification of large document collections is time-consuming and may lead to inconsistent grouping, slow retrieval, and difficulty identifying related information. This paper presents an AI-Based Document Clustering System designed to automatically organize textual documents according to their content similarity. The system accepts PDF, DOCX, and TXT files and extracts their textual content for processing. The extracted text is cleaned through normalization and tokenization, after which Term Frequency–Inverse Document Frequency (TF-IDF) converts each document into a numerical feature vector. The resulting document-feature matrix is processed using the K-Means clustering algorithm to group documents with similar terms and contextual patterns. For supervised category prediction when labelled examples are available, the project also considers K-Nearest Neighbours (KNN), Support Vector Machine (SVM), and Random Forest classifiers. Principal Component Analysis (PCA) is used to reduce feature dimensions for two-dimensional visualization of document clusters. The proposed workflow includes file validation, text extraction, preprocessing, feature generation, clustering, optional classification, evaluation, and result visualization. System performance can be examined using Accuracy, Precision, Recall, and F1-score for labelled classification tasks, along with the Silhouette Score for cluster separation. The system aims to reduce manual document organization effort, improve the accessibility of related documents, and provide a structured foundation for document management. Since completed experimental measurements and dataset details are not specified in the project material, this paper describes the proposed design and evaluation approach without claiming unverified performance results.*

Key Words: Document clustering, text mining, TF-IDF, K-Means, KNN, SVM, Random Forest, PCA, machine learning.

1. INTRODUCTION

Digital documents are generated continuously in educational institutions, businesses, research environments, and personal information systems. These documents commonly exist in different file formats and contain diverse topics, writing styles, and levels of detail. As a collection grows, manually assigning documents to folders or categories becomes inefficient. Keyword search can retrieve documents containing particular terms, but it does not necessarily reveal groups of documents that discuss related subjects using different vocabulary.

Document clustering addresses this problem by organizing documents into groups based on their content similarity. Unlike classification, which generally relies on predefined class labels, clustering is an unsupervised learning task: the algorithm identifies structure in the data without requiring a known category for every document. Text clustering commonly begins by transforming documents into numerical feature vectors. TF-IDF is a widely used representation that weights terms according to their importance in an individual document and their distribution across the collection [1], [2].

The proposed project, titled “AI-Based Document Clustering System,” focuses on accepting PDF, DOCX, and TXT files, extracting their text, cleaning and tokenizing the content, and generating TF-IDF vectors. K-Means is used to

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, November 2026 Impact Factor: 8.2

form clusters, while KNN, SVM, and Random Forest are included to support class prediction and comparative evaluation. PCA is included in the workflow to reduce dimensionality and visualize document distributions. The intended output is an automatically grouped document collection with evaluation information and a two-dimensional representation of the feature space.

This paper follows the system scope described in the supplied Phase-I presentation. It explains the motivation, related work, problem formulation, objectives, architecture, algorithms, evaluation measures, expected outcomes, limitations, and future development. The supplied material does not provide implementation screenshots, dataset details, or measured experimental results; therefore, the paper distinguishes proposed functionality from results that would require actual execution.

II. MOTIVATION AND PROBLEM STATEMENT

A. Motivation

- Large collections of unstructured documents are difficult to organize manually.
- Automatic grouping can reduce repetitive document-management work and improve retrieval.
- Text representation methods such as TF-IDF provide a practical way to convert documents into machine-readable features.
- Machine-learning algorithms can identify groups or predict document categories based on learned patterns.

A document management workflow becomes more useful when it can process more than one file type through a common pipeline. Supporting PDF, DOCX, and TXT files allows the same clustering process to be applied to documents originating from different applications.

B. Problem Statement

The problem addressed is the development of an AI-based system that accepts documents in PDF, DOCX, and TXT formats, extracts their textual content, preprocesses the text, converts it into TF-IDF feature vectors, and automatically groups similar documents. The system should also provide a basis for evaluating clustering and classification behaviour through appropriate metrics and visualization.

III. SCOPE

The scope is limited to text-based document analysis. The presentation specifies file ingestion, text extraction, cleaning, tokenization, TF-IDF feature generation, K-Means clustering, KNN/SVM/Random Forest prediction, evaluation, PCA visualization, and clustered output. It does not specify OCR for scanned PDFs, multilingual processing, semantic transformer embeddings, a web deployment stack, or a particular dataset.

IV. LITERATURE REVIEW

The Phase-I presentation identifies research directions involving keyword-based clustering, similarity measurement, feature weighting, concept-space representations, ontology-based methods, and geometric similarity. The following summary retains the studies and method descriptions given in the supplied presentation. Bibliographic details for entries marked as unverified in that source should be checked against the original publications before final academic submission.

Related study	Method and finding	Research gap
1. Chang and Hsu — Using Topic Keyword Clusters for Automatic Document Clustering	Keyword extraction and KNN graph; topic-based clustering.	Complexity for large collections.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, November 2026
Impact Factor: 8.2

Related study	Method and finding	Research gap
2. Shyu et al. — Affinity-Based Similarity Measure for Web Document Clustering	Affinity / probabilistic similarity; web access-pattern similarity.	Requires web-specific information.
3. Chen et al. — Weighted k-Means Algorithm Based Text Clustering	Weighted K-Means; reduces irrelevant attributes.	Requires feature weighting.
4. Muflikhah and Baharudin — Document Clustering Using Concept Space and Cosine Similarity Measurement	LSI / PCA with Fuzzy C-Means; improved F-measure reported in source.	More complex processing pipeline.
5. Novel Similarity Measure for Document Clustering Based on Topic Phrases	Topic-phrase similarity; intended to improve semantic similarity.	Phrase extraction is domain-dependent; author details unverified in source.
6. Yang et al. — Research on Ontology-Based Text Clustering	Ontology and word-relativity approach; uses semantic relations.	Ontology construction cost.
7. Document Clustering for Event Identification and Trend Analysis in Market News	News-document clustering for event and trend analysis.	Domain-dependent; author details unverified in source.
8. Heidarian and Dinneen — A Hybrid Geometric Approach for Measuring Similarity Level Among Documents and Document Clustering	TS-SS, cosine similarity, and Euclidean distance; improved clustering purity reported in source.	Similarity model is complex.

The literature summary indicates that document clustering quality depends on both the representation of text and the similarity or grouping method. Keyword and term-frequency approaches are relatively direct, whereas concept-space, ontology, and hybrid geometric approaches introduce additional processing or domain knowledge. The proposed project adopts TF-IDF as its core representation and compares a compact set of commonly used machine-learning algorithms within one processing pipeline.

V. RESEARCH GAP

The research gaps listed in the supplied presentation point to practical trade-offs: keyword graphs may become complex for large collections; web-specific similarity requires contextual information; weighted clustering requires feature-weight selection; concept-space and ontology approaches add pipeline complexity; and phrase-based similarity may depend on the domain. The project addresses a practical system-level gap by bringing multi-format text extraction, common preprocessing, TF-IDF representation, clustering, supervised prediction, and visualization into a single workflow.

This is an implementation-oriented gap rather than a claim that TF-IDF or K-Means is a new algorithm. The contribution is the integration of the specified processing stages into an accessible document organization pipeline and the comparison of multiple algorithmic approaches under a consistent feature representation.

VI. OBJECTIVES

- Automate clustering of PDF, DOCX, and TXT documents.
- Extract textual content from the supported document formats.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, November 2026 Impact Factor: 8.2

- Apply text cleaning and tokenization before feature generation.
- Represent documents using TF-IDF feature vectors.
- Implement K-Means for unsupervised grouping of similar documents.
- Use KNN, SVM, and Random Forest for document class prediction and comparison.
- Evaluate system behaviour using accuracy, precision, recall, F1-score, and silhouette score, where applicable.
- Use PCA to visualize document representations in a two-dimensional feature space.

VII. PROPOSED SYSTEM ARCHITECTURE

The architecture shown in the Phase-I presentation (page 6) defines a sequential pipeline consisting of document input, text extraction, preprocessing, TF-IDF feature generation, machine-learning analysis, evaluation with PCA, and clustered output. Each stage prepares data for the next stage and contributes to the final organization of the uploaded document collection.

1. Input	**PDF / DOCX / TXT**
2. Extraction	Read and extract document text
3. Preprocessing	Clean text and tokenize
4. Representation	Generate TF-IDF feature vectors
5. ML layer	K-Means / KNN / SVM / Random Forest
6. Analysis	Evaluation metrics and PCA
7. Output	Display clustered documents

Fig. 1. Functional stages of the proposed AI-based document clustering pipeline.

A. Document Input and Text Extraction

The system accepts PDF, DOCX, and TXT files. Text extraction converts the content of each supported file into a common text representation. The extraction stage should preserve document boundaries so that each extracted text remains associated with its original file name. The presentation does not specify the extraction libraries or handling of scanned/image-only PDF files.

B. Text Preprocessing

Preprocessing prepares raw extracted text for feature generation. The presentation explicitly identifies cleaning and tokenization. Cleaning may be implemented to normalize whitespace and remove unwanted symbols or formatting artefacts. Tokenization divides text into individual terms or tokens. Any additional operations, such as stop-word removal, stemming, or lemmatization, should be documented only if they are implemented in the final system.

VIII. TF-IDF Feature Extraction

TF-IDF represents each document as a weighted vector over the vocabulary of the collection. Term frequency reflects how often a term appears in a document, while inverse document frequency reduces the influence of terms that occur in many documents. A common formulation is:

$$TF\text{-}IDF(t,d) = TF(t,d) \times \log(N / DF(t))$$

where t is a term, d is a document, N is the number of documents, and DF(t) is the number of documents containing t. The resulting sparse vectors provide numerical input for clustering and classification. Vocabulary construction and vectorizer settings should be fitted on training data only when a supervised train/test evaluation is performed.

IX. ALGORITHMS USED

A. K-Means Clustering

K-Means is the unsupervised clustering method specified in the proposed system. It partitions feature vectors into K clusters by iteratively assigning each document to a cluster centroid and updating the centroids from assigned samples. The objective is to reduce within-cluster squared distance. For TF-IDF vectors, cosine distance or normalized-vector

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, November 2026
Impact Factor: 8.2

configurations may also be considered, but the precise distance metric must be stated in the implementation report. K-Means requires the number of clusters to be selected or estimated.

B. K-Nearest Neighbours

KNN is a supervised, instance-based classifier. It predicts the class of a document by examining the labels of its nearest training samples in the feature space. Its behaviour depends on the selected number of neighbours, distance metric, and weighting scheme. KNN is therefore suitable for comparison when labelled examples are available, but it is not itself an unsupervised clustering algorithm.

X. SUPPORT VECTOR MACHINE

SVM constructs a decision boundary that separates labelled classes with a maximum-margin principle. In text classification, linear SVM is commonly considered for high-dimensional sparse TF-IDF vectors. The final implementation should record the kernel, regularization parameter, and class-weight settings used for reproducibility.

XI. RANDOM FOREST

Random Forest combines predictions from multiple decision trees trained using randomized samples and feature subsets. It can model nonlinear decision patterns and provides an alternative supervised classifier. For sparse, high-dimensional text features, its performance and computational cost should be measured rather than assumed to be superior.

PCA Visualization

Principal Component Analysis transforms correlated input features into a smaller set of orthogonal components that retain as much variance as possible. The project presentation specifies a two-dimensional PCA visualization to display document vectors and their cluster distribution. PCA is a visualization and dimensionality-reduction technique; it does not, by itself, assign semantic categories to documents.

XII. EVALUATION FRAMEWORK

The presentation lists Accuracy, Precision, Recall, F1-score, and Silhouette Score as evaluation measures. Their interpretation depends on whether the task is supervised classification or unsupervised clustering. Accuracy, precision, recall, and F1-score require ground-truth labels. Silhouette score evaluates cluster cohesion and separation without requiring class labels.

A. Classification Metrics

For a labelled classification task, let TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives. The standard measures are:

- Accuracy = (TP + TN) / (TP + TN + FP + FN)
- Precision = TP / (TP + FP)
- Recall = TP / (TP + FN)
- F1-score = 2 × Precision × Recall / (Precision + Recall)

For multi-class document classification, the report should state whether macro, micro, or weighted averaging is used. A held-out test set should be kept separate from model fitting to avoid data leakage.

B. Silhouette Score

For each document i, a(i) is the mean distance from i to other documents in its own cluster, and b(i) is the lowest mean distance from i to documents in another cluster. The silhouette coefficient is:

$$s(i) = (b(i) - a(i)) / max(a(i), b(i))$$

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, November 2026
Impact Factor: 8.2

The score ranges from −1 to +1. A higher value generally indicates that documents are closer to their assigned cluster than to neighbouring clusters. The distance function used for TF-IDF vectors should be reported.

XIII. EXPERIMENTAL REPORTING REQUIREMENTS

The supplied presentation does not include dataset size, class labels, cluster count, train/test split, vectorizer parameters, hyperparameters, runtime, or metric values. These must be recorded after implementation. The paper therefore does not fabricate a comparison table or claim that one algorithm achieved a particular accuracy.

XIV. EXPECTED RESULTS AND OUTPUT

According to the Phase-I presentation (page 7), the expected system behaviour includes automatic grouping of similar documents, support for multiple-file upload and extraction for PDF/DOCX/TXT, prediction output from KNN/SVM/Random Forest, metric reporting, and a two-dimensional PCA visualization.

- Documents are grouped into clusters based on their TF-IDF feature similarity.
- Users can inspect the cluster assignment associated with each uploaded file.
- Supervised prediction results are displayed when labelled training data is available.
- Accuracy, precision, recall, and F1-score are reported for labelled classification experiments.
- Silhouette score is reported for clustering quality.
- PCA plots provide a visual overview of document distribution in two dimensions.

These are expected outputs, not measured outcomes. Actual results should be added after the software has been implemented and tested on a documented dataset.

XV. DISCUSSION

The proposed workflow is modular and can be evaluated stage by stage. Text extraction determines whether useful content reaches the analysis pipeline. Preprocessing affects the vocabulary and the resulting feature space. TF-IDF provides a transparent baseline representation, but it mainly captures term occurrence and weighting rather than deeper semantic relationships. Documents that discuss the same topic using different vocabulary may therefore be placed farther apart than expected.

K-Means provides a direct unsupervised baseline, while KNN, SVM, and Random Forest address supervised prediction when labelled examples exist. These methods serve different purposes and should not be compared as if they all perform the same task. A fair evaluation should use a consistent labelled dataset for the supervised models and separate cluster-quality analysis for K-Means. PCA plots can support interpretation, but a visually separated two-dimensional projection does not prove high classification accuracy.

The project's practical value lies in combining common document formats with a shared analysis pipeline. A user can submit a collection of files and obtain a structured view of related documents, reducing the need for manual sorting. The usefulness of the final system will depend on extraction reliability, preprocessing choices, cluster configuration, and the clarity of the output interface.

XVI. LIMITATIONS AND FUTURE SCOPE

A. Current Scope Limitations

- The presentation does not identify a benchmark dataset or the source and size of the document collection.
- Scanned PDF processing and Optical Character Recognition are not specified.
- TF-IDF is based on lexical statistics and may not capture contextual semantic similarity.
- K-Means requires a selected cluster count and can be sensitive to initialization.
- Supervised methods require labelled examples, which are not described in the Phase-I material.
- No measured experimental results are provided in the supplied presentation.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, November 2026
Impact Factor: 8.2

B. Future Enhancements

Future development may evaluate OCR for scanned documents, multilingual text normalization, word or sentence embeddings for semantic representation, and hierarchical clustering for collections where the number of groups is not known in advance. Search and filtering interfaces could be added to support retrieval within clusters. For reproducible evaluation, the project should define a public or institutionally approved dataset, document train/test separation, publish model parameters, and report runtime alongside quality metrics. These are possible extensions and are not claimed as completed features.

XVII. CONCLUSION

This paper presents the design of an AI-based document clustering system based on the supplied Phase-I project presentation. The system accepts PDF, DOCX, and TXT documents, extracts their text, performs cleaning and tokenization, and converts the text into TF-IDF feature vectors. K-Means is specified for unsupervised document grouping, while KNN, SVM, and Random Forest provide supervised prediction approaches. PCA supports two-dimensional visualization, and the evaluation framework includes classification metrics and silhouette score. The proposed architecture offers a practical foundation for organizing unstructured document collections. Final claims about accuracy or clustering quality must be supported by implementation experiments using a documented dataset.

XVIII. DETAILED IMPLEMENTATION WORKFLOW

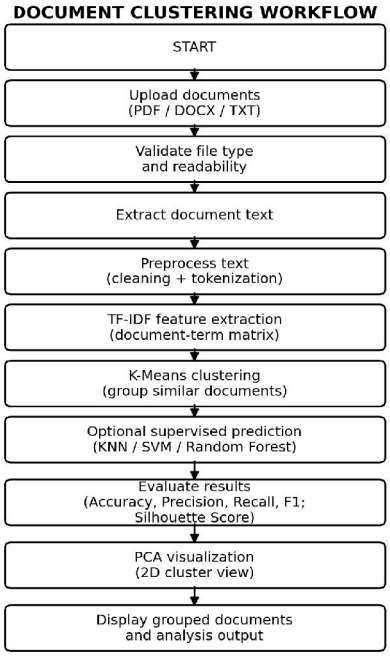


Fig. 1. Workflow of the proposed AI-based document clustering system.

A. File Validation and Collection Management

A robust implementation should validate each uploaded file before extraction. The validation stage should check the file extension, confirm that the file can be opened, and reject unsupported or empty files with a clear message. Each document should be assigned an internal identifier and retain its original filename so that cluster assignments can be traced back to the source document. When several files are submitted together, the system should process them as a

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, November 2026
Impact Factor: 8.2

collection while preserving document-level boundaries. A failure in one file should ideally be reported without silently discarding the remaining valid files.

B. Extraction Quality and Text Normalization

Text extraction is a critical dependency because downstream algorithms operate on the extracted text rather than on the original page layout. PDF files may contain selectable text, scanned images, tables, headers, footers, or mixed content. DOCX files may contain paragraphs, headings, and tables, while TXT files generally provide plain text. The project presentation confirms PDF, DOCX, and TXT support but does not state whether OCR or table-aware extraction is implemented. The final system documentation should specify the supported PDF type and explain how extraction errors, empty pages, and encoding issues are handled.

After extraction, normalization should produce a consistent textual representation. Cleaning and tokenization are explicitly part of the proposed pipeline. Additional normalization decisions should be applied consistently across all documents and recorded as configuration choices. For example, the implementation may decide whether to lowercase tokens, remove punctuation, remove stop words, or apply stemming. Such operations can affect vocabulary size and similarity, so they should not be introduced without documenting their effect.

XIX. FEATURE MATRIX CONSTRUCTION

Once the documents have been preprocessed, the TF-IDF vectorizer builds a vocabulary and transforms each document into a sparse numerical vector. Each row of the document-term matrix corresponds to one document, and each column corresponds to a vocabulary term. The matrix can be high-dimensional because a collection may contain many distinct terms. Sparse storage is therefore appropriate because most documents contain only a small fraction of the complete vocabulary. The same fitted vectorizer must be reused when transforming new documents so that feature dimensions and term indices remain consistent.

XX. MODEL EXECUTION AND RESULT ASSOCIATION

The unsupervised K-Means stage assigns a cluster identifier to each document. The result should be stored together with the document identifier and filename. The supervised models require labelled examples: the training process learns a mapping between TF-IDF vectors and known classes, and the trained model can then predict labels for unseen documents. The system should clearly distinguish a K-Means cluster identifier from a predicted semantic class because a cluster number has no inherent category meaning. If a cluster is later assigned a human-readable label, that label should be based on inspection or an explicitly defined labelling procedure.

Result Presentation

The output layer should provide a concise view of the document collection, including the original file name, assigned cluster, and any available predicted class. The PCA plot can provide an overview of the distribution of document vectors. The visualization should include a legend and a clear explanation that the displayed coordinates are a two-dimensional projection of a higher-dimensional feature space. A cluster table and a plot serve complementary purposes: the table supports precise file-level inspection, while the plot supports an overview of group structure.

XXI. EXPERIMENTAL DESIGN AND REPRODUCIBILITY

A. Dataset Description

An experimental evaluation requires a documented corpus. The dataset description should state the number of documents, file formats, source, language, subject domains, and whether multiple documents are near-duplicates. If supervised evaluation is planned, the number of labelled classes and the distribution of documents per class should also be reported. The supplied Phase-I presentation does not identify a dataset or provide class counts; consequently, the current paper treats dataset selection as an implementation requirement rather than reporting a completed experiment.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, November 2026
Impact Factor: 8.2

B. Train/Test Separation

For KNN, SVM, and Random Forest, labelled documents should be divided into training and test subsets. The split should be performed before fitting the TF-IDF vocabulary or any learned preprocessing component. Fitting the vectorizer on the complete corpus before splitting can expose the training process to information from the test set. A stratified split may be appropriate when class labels are imbalanced. The selected split ratio and random seed should be reported so that the evaluation can be reproduced.

XXII. PARAMETER SELECTION

The K-Means experiment should report the selected number of clusters, initialization strategy, number of restarts, stopping tolerance, and distance configuration. The supervised experiments should record the K value and distance metric for KNN, the kernel and regularization parameter for SVM, and the number of trees and other relevant settings for Random Forest. Hyperparameters should be selected using training data or cross-validation rather than by repeatedly checking the final test set.

XXIII. FAIR COMPARISON

A meaningful comparison requires the algorithms to be evaluated on the same underlying document split and feature representation wherever their task definitions permit. K-Means is an unsupervised method, whereas KNN, SVM, and Random Forest are supervised classifiers. Their metrics are therefore not directly interchangeable. For classification, all supervised models should use the same training and test documents and the same label definitions. For clustering, silhouette score can be reported independently, and external cluster validation measures may be added if reliable ground-truth labels are available.

Reporting Template

The final implementation report should include a dataset summary, preprocessing configuration, TF-IDF vocabulary settings, model hyperparameters, hardware/software environment, execution time, and evaluation values. Classification results should include per-class metrics or a confusion matrix when possible. Clustering results should include the selected cluster count and a silhouette score calculated using a stated distance metric. Reporting both strengths and failure cases is important: a high aggregate score can hide poor performance on minority topics or documents with very short text.

XXIV. SECURITY, PRIVACY, AND OPERATIONAL CONSIDERATIONS

Documents may contain personal, academic, business, or confidential information. A deployed version should define how uploaded files are stored, how long they are retained, and who can access the extracted text and model outputs. Temporary files should be deleted according to a clear retention policy. If documents are processed on a server, access control and secure transport should be considered. The Phase-I presentation does not describe a storage architecture or privacy mechanism, so these items are identified as operational requirements for a future implementation.

The system should also provide transparent error messages for unsupported file formats, password-protected files, empty documents, extraction failures, and documents that contain too little text for meaningful clustering. Logging should capture technical errors without unnecessarily recording sensitive document content. For larger collections, batch processing and progress indicators can improve usability, while resource limits can prevent a single large upload from exhausting memory.

XXV. CONSIDERATIONS FOR INTERPRETING CLUSTERS

A cluster is a group formed according to the selected feature representation and similarity objective. It should not automatically be interpreted as a definitive topic category. The most frequent or highest-weighted terms in each cluster can be inspected to help describe its dominant vocabulary, but the final label should be treated as an interpretation

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, November 2026
Impact Factor: 8.2

rather than a direct output of K-Means. Clusters may contain mixed topics when documents are short, vocabulary overlaps, or the collection covers several related domains.

Cluster quality depends on the choice of K, the TF-IDF configuration, preprocessing, and the distance measure. A practical implementation can evaluate several candidate values of K and compare silhouette scores, while also reviewing cluster sizes and representative documents. A very small cluster may represent a niche topic, an outlier, or an extraction problem. A very large cluster may indicate that the selected number of clusters is too small or that the corpus contains broad overlapping subjects. Quantitative measures should therefore be interpreted alongside sample-document inspection.

XXVI. SUMMARY OF PROJECT MODULES

The project can be organized into six logical modules consistent with the proposed architecture. The input module manages PDF, DOCX, and TXT files. The extraction module converts their contents into plain text. The preprocessing module cleans and tokenizes the extracted text. The feature extraction module creates TF-IDF vectors. The machine-learning module performs K-Means clustering and supervised prediction through KNN, SVM, and Random Forest. Finally, the evaluation and reporting module calculates appropriate metrics and presents the PCA visualization and clustered output. This modular separation supports testing and maintenance because each stage has a defined input and output.

The expected data flow is document files → extracted text → normalized tokens → TF-IDF matrix → clustering/classification → evaluation and visualization → clustered document output. This sequence reflects the architecture in the supplied presentation and does not assume an additional model or service that is not described there.

XXVII. ALGORITHM DESIGN AND HIGH-LEVEL PROCEDURE

The following procedure summarizes the processing logic implied by the project architecture. It is a conceptual workflow and should be adapted to the actual software implementation. The procedure begins by accepting a set of supported files and validating each file. The system then extracts the text from each file and retains a mapping between the text and its source filename. Preprocessing is applied consistently to all extracted text. The TF-IDF vectorizer is fitted to the resulting document collection and produces a sparse document-term matrix. K-Means is executed on the feature matrix to obtain cluster identifiers. If labelled examples are available, the data can also be separated into training and test partitions and used to train KNN, SVM, and Random Forest models. Finally, the system calculates suitable evaluation measures and generates a PCA projection for visualization.

Conceptual procedure:

1. Receive a batch of PDF, DOCX, and TXT files.
2. Validate file type and readability; record the original filename.
3. Extract text from each valid file and retain one text record per document.
4. Clean the text and tokenize it using a consistent preprocessing configuration.
5. Fit a TF-IDF vectorizer and transform the documents into feature vectors.
6. Apply K-Means to group documents and store the cluster assignment.
7. When labelled data is available, train and evaluate KNN, SVM, and Random Forest.
8. Calculate classification metrics for labelled predictions and silhouette score for clustering.
9. Apply PCA to create a two-dimensional view of the document feature space.
10. Display the clustered files, prediction outputs, metrics, and visualization.

The distinction between steps 6 and 7 is important. K-Means discovers groups without predefined labels, whereas the supervised models learn from known class labels. A system interface should identify which output is a cluster assignment and which output is a predicted class. This avoids presenting an unsupervised cluster number as though it were a verified semantic category.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, November 2026
Impact Factor: 8.2

XXVIII. DESIGN CHOICES AND TECHNICAL TRADE-OFFS

A. Lexical Features Versus Semantic Features

TF-IDF is suitable as a baseline because it is relatively interpretable and can be applied to a collection without manually defining a vocabulary. Its main limitation is that it represents terms as separate dimensions and does not inherently understand synonymy or contextual meaning. For example, documents that use different words for the same concept may have lower lexical similarity. Conversely, documents that repeat common technical terms may appear similar even when their overall subject differs. These characteristics should be considered when interpreting clusters.

B. Choosing the Number of Clusters

K-Means requires the number of clusters to be specified. If the value is too small, unrelated subjects may be merged. If it is too large, one coherent topic may be divided into several small groups. The silhouette score can be calculated for a range of candidate cluster counts to support selection, but it should not be treated as the only decision criterion. Cluster sizes, representative documents, and the practical meaning of the resulting groups should also be examined. The selected value and selection rationale should be included in the final project report.

XXIX. SPARSE HIGH-DIMENSIONAL DATA

TF-IDF matrices are usually sparse and may contain thousands of vocabulary dimensions. Algorithms should be selected with this representation in mind. Linear SVM is often considered for sparse text features, while KNN prediction can become expensive when the reference collection is large. Random Forest can model nonlinear relationships, but a large number of sparse features may increase training cost. The project appropriately includes multiple methods for comparison; the final choice should be based on measured performance, resource use, and the intended collection size.

XXX. PCA AS AN INTERPRETATION AID

PCA creates a lower-dimensional projection that can make a high-dimensional collection easier to inspect. A two-dimensional plot may show broad separation or overlap among groups, but some information is necessarily lost during projection. Distances visible in the plot may not exactly represent the original high-dimensional distances. Consequently, PCA should be used together with quantitative evaluation and document-level inspection rather than as a stand-alone proof of clustering quality.

XXXI. TESTING STRATEGY AND QUALITY ASSURANCE

The implementation should be tested with representative files from all three supported formats. A basic functional test should confirm that a valid PDF, DOCX, and TXT file can each be uploaded, extracted, preprocessed, vectorized, and included in the output. A multi-file test should verify that document boundaries are preserved and that each output cluster assignment maps to the correct original filename. Negative tests should include an unsupported extension, an empty file, a corrupted document, a password-protected PDF, and a file with no extractable text. The expected behaviour for each case should be a clear error or warning rather than a silent failure.

The preprocessing stage should be checked with documents containing extra whitespace, punctuation, repeated terms, mixed letter case, and non-text artefacts. The TF-IDF stage should be checked to ensure that the number of matrix rows equals the number of successfully processed documents and that the feature matrix contains valid numeric values. Model testing should confirm that predictions are generated only after a model has been trained and that train/test separation is respected. PCA output should contain one plotted point per document included in the visualization.

Quality assurance should also include repeatability checks. When a fixed random seed and identical configuration are used, the workflow should produce consistent outputs to the extent supported by the algorithms and software environment. The project report should identify any stochastic component and record the relevant seed or initialization

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, November 2026
Impact Factor: 8.2

settings. Performance testing should measure processing time as the document collection grows, because a system that works for a small demonstration set may require optimization for larger collections.

XXXII. APPLICATION AREAS

An AI-based document clustering workflow can support academic document repositories, where lecture notes, assignments, reports, and research articles may be grouped by subject. In an office environment, policy documents, forms, project records, and internal reports may be organized into related collections. Research teams may use clustering to explore a large set of papers before performing a more detailed literature review. News or market-analysis collections may also be grouped to support topic exploration, although domain-specific preprocessing and evaluation would be needed. These are potential application areas of the proposed approach and are not reported deployments from the Phase-I presentation.

Across these settings, the value of clustering is exploratory organization: it helps users discover related documents and inspect collections more efficiently. It does not replace expert review, metadata management, or a validated search system. Users should be able to open the original document and verify its content rather than relying only on an automatically assigned cluster.

XXXIII. DATA FLOW AND MODULE INTERFACES

A modular design makes the proposed system easier to implement and validate. The input component should provide a list of accepted file objects and metadata such as file name, extension, and processing status. The extraction component should return the extracted text and an extraction status for each file. The preprocessing component should accept one text record and return normalized tokens or a normalized text string. The feature extraction component should accept the complete set of preprocessed documents and return the document-term matrix together with the fitted TF-IDF vectorizer. The machine-learning component should consume this matrix and return cluster identifiers or class predictions. The reporting component should combine those outputs with file metadata and evaluation results.

The interfaces between modules should be explicit. For example, if extraction fails for one file, the system should mark that file as failed and avoid passing an empty string into clustering without notification. If a document is successfully extracted but contains very few terms, the system may still process it, but the interface should allow the user to recognize that the result may be unreliable. Similarly, if there are fewer documents than the requested number of clusters, the K-Means stage should validate the configuration and return an actionable message.

A consistent document identifier is useful throughout the pipeline. The identifier should be generated at ingestion and retained in extraction records, vector rows, model outputs, and visualization metadata. This avoids relying on row order alone to match a plotted point or cluster assignment to an uploaded file. The output should also preserve the original file name for user-facing display.

XXXIV. ETHICAL AND RESPONSIBLE USE

Automated clustering is a decision-support mechanism rather than a definitive authority on document meaning. A cluster assignment may be affected by document length, extraction quality, vocabulary, and preprocessing choices. Users should therefore be able to review the source document and should not treat an automatically generated group as a final judgement about the document's purpose, accuracy, or importance.

If the system is used with sensitive institutional or business documents, the implementation should follow the organization's data-handling requirements. Access to uploaded documents and extracted text should be restricted to authorized users. The system should avoid exposing document contents through diagnostic logs or public visualizations. Any future cloud-based deployment should clearly disclose where data is processed and how long it is retained.

The system should also be assessed for uneven performance across document types and subject areas. For example, a pipeline tuned on long English reports may perform differently on short forms, scanned pages, or documents written in

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, November 2026 Impact Factor: 8.2

another language. A representative test collection and clear reporting of supported languages and formats can help users understand these boundaries.

XXXV. OVERALL CONTRIBUTION

The project provides an integrated workflow for converting heterogeneous text documents into a common representation and organizing them through machine-learning methods. Its principal contribution is the combination of multi-format ingestion, text extraction, cleaning and tokenization, TF-IDF feature generation, K-Means clustering, supervised model comparison, PCA visualization, and evaluation in one proposed system. The approach is intentionally based on established methods so that the implementation can be developed and assessed using a transparent baseline.

The project also establishes a practical evaluation plan. It separates unsupervised grouping from supervised prediction, identifies suitable measures for each task, and highlights the dataset and parameter details required for a reproducible experiment. Once the system is implemented, the next step is to execute the workflow on a documented corpus and replace expected outcomes with measured results, including failure analysis and computational cost.

REFERENCES

[1] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," Information Processing & Management, vol. 24, no. 5, pp. 513–523, 1988. https://doi.org/10.1016/0306-4573(88)90021-0

[2] C. C. Aggarwal and C. X. Zhai, "A survey of text clustering algorithms," in Mining Text Data, Springer, pp. 77–128, 2012. https://doi.org/10.1007/978-1-4614-3223-4_4

[3] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," Journal of Computational and Applied Mathematics, vol. 20, pp. 53–65, 1987. https://doi.org/10.1016/0377-0427(87)90125-7

[4] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, pp. 273–297, 1995. https://doi.org/10.1007/BF00994018

[5] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," Journal of the American Society for Information Science, vol. 41, no. 6, pp. 391–407, 1990. https://doi.org/10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," Journal of Machine Learning Research, vol. 3, pp. 993–1022, 2003. https://www.jmlr.org/papers/v3/blei03a.html

[7] P. J. F. Groenen and P. J. Rousseeuw, "Clustering with the K-means algorithm," in Handbook of Market Research, Springer, 2019. https://doi.org/10.1007/978-3-319-05542-8_8

[8] R. Xu and D. Wunsch, "Survey of clustering algorithms," IEEE Transactions on Neural Networks, vol. 16, no. 3, pp. 645–678, 2005. https://doi.org/10.1109/TNN.2005.845141

[9] M. Steinbach, G. Karypis, and V. Kumar, "A comparison of document clustering techniques," in KDD Workshop on Text Mining, 2000. https://cs.fit.edu/~pkc/classes/ml-internet/papers/steinbach00kddTM.pdf

[10] S. Bisht and A. Paul, "Document clustering: A review," International Journal of Computer Applications, vol. 73, no. 11, pp. 26–33, 2013. https://doi.org/10.5120/12787-0024

[11] W. N. I. Al-Obaydy, H. A. Hashim, Y. A. Najm, and A. A. Jalal, "Document classification using term frequency-inverse document frequency and K-means clustering," International Journal of Electrical and Computer Engineering, vol. 27, no. 3, pp. 1517–1524, 2022. https://doi.org/10.11591/ijeecs.v27.i3.pp1517-1524

[12] M. H. Ahmed, S. Tiun, N. Omar, and N. S. Sani, "Short text clustering algorithms, application and challenges: A survey," Applied Sciences, vol. 13, no. 1, Art. no. 342, 2023. https://doi.org/10.3390/app13010342

[13] S. Zhou et al., "A comprehensive survey on deep clustering: Taxonomy, challenges, and future directions," ACM Computing Surveys, vol. 57, no. 3, Art. no. 69, 2024. https://doi.org/10.1145/3689036

[14] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in European Conference on Machine Learning, pp. 137–142, 1998. https://doi.org/10.1007/BFb0026683

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, November 2026 Impact Factor: 8.2

[15] Y. Yang and X. Liu, "A re-examination of text categorization methods," in SIGIR, pp. 42–49, 1999. https://doi.org/10.1145/312624.312647
[16] F. Sebastiani, "Machine learning in automated text categorization," ACM Computing Surveys, vol. 34, no. 1, pp. 1–47, 2002. https://doi.org/10.1145/505282.505283
[17] C. D. Manning, P. Raghavan, and H. Schütze, "Introduction to Information Retrieval," Cambridge University Press, 2008. https://nlp.stanford.edu/IR-book/
[18] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," Journal of Documentation, vol. 28, no. 1, pp. 11–21, 1972. https://doi.org/10.1108/eb026526
[19] G. Salton and M. J. McGill, "Introduction to Modern Information Retrieval," McGraw-Hill, 1983. https://ciir.cs.umass.edu/downloads/
[20] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in Proc. Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297, 1967. https://projecteuclid.org/euclid.bsmsp/1200512992
[21] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," ACM Computing Surveys, vol. 31, no. 3, pp. 264–323, 1999. https://doi.org/10.1145/331499.331504
[22] A. K. Jain, "Data clustering: 50 years beyond K-means," Pattern Recognition Letters, vol. 31, no. 8, pp. 651–666, 2010. https://doi.org/10.1016/j.patrec.2009.09.011
[23] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in KDD, pp. 226–231, 1996. https://www.aaai.org/Papers/KDD/1996/KDD96-037.pdf
[24] U. von Luxburg, "A tutorial on spectral clustering," Statistics and Computing, vol. 17, pp. 395–416, 2007. https://doi.org/10.1007/s11222-007-9033-z
[25] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in Advances in Neural Information Processing Systems 14, pp. 849–856, 2002. https://proceedings.neurips.cc/paper_files/paper/2001/file/801272ee79cfde7fa5960571fee36b9b-Paper.pdf
[26] T. Hofmann, "Probabilistic latent semantic indexing," in SIGIR, pp. 50–57, 1999. https://doi.org/10.1145/312624.312649
[27] M. A. Hearst and J. O. Pedersen, "Reexamining the cluster hypothesis: Scatter/gather on retrieval results," in SIGIR, pp. 76–84, 1996. https://doi.org/10.1145/243199.243216
[28] D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey, "Scatter/gather: A cluster-based approach to browsing large document collections," in SIGIR, pp. 318–329, 1992. https://doi.org/10.1145/133160.133214
[29] Y. Zhao and G. Karypis, "Criterion functions for document clustering: Experiments and analysis," Machine Learning, vol. 55, pp. 311–331, 2004. https://doi.org/10.1023/B:MACH.0000027785.44527.d6
[30] Y. Zhao and G. Karypis, "Evaluation of hierarchical clustering algorithms for document datasets," in CIKM, pp. 515–524, 2002. https://doi.org/10.1145/584792.584877
[31] C. C. Aggarwal, "Machine Learning for Text," Springer, 2018. https://doi.org/10.1007/978-3-319-73531-3