"A Comparative Approach for Abstract Clustering Through Text Preprocessing and Sophisticated Feature Engineering"

Rakibul Hasan Shakil
Computer Science and Engineering
American International University Bangladesh
Dhaka, Bangladesh
22-49462-3@student.aiub.edu

Sumaiya Akter Shimu
Computer Science and Engineering
American International University Bangladesh
Dhaka, Bangladesh
22-48668-3@student.aiub.edu

Md. Siam-Ul-Islam
Computer Science and Engineering
American International University Bangladesh
Dhaka, Bangladesh
22-49466-3@student.aiub.edu

Ayesha Mehjaben Tonima
Computer Science and Engineering
American International University Bangladesh
Dhaka, Bangladesh
21-45841-3@student.aiub.edu

***Abstract*—The rapid increase in scientific publications, especially in fields like large language models (LLMs), has made automated methods essential for sorting and understanding research abstracts. Manual evaluation falls short in detecting new themes, underscoring the significance of text mining, dimensionality reduction, and clustering methods. This research presents a structured pipeline for abstract clustering that incorporates text preprocessing, TF–IDF vectorization, and dimensionality reduction through Principal Component Analysis (PCA). Three clustering algorithms—K-Means, Hierarchical Clustering, and DBSCAN—were assessed through heuristic metrics (silhouette scores, elbow method) and visualization techniques (PCA scatter plots, word clouds, and bar charts). Findings indicate that all three algorithms reliably point out two major thematic categories: (i) specific technical jargon associated with software tools and AI models, and (ii) conceptual and evaluative language centered on critique, accuracy, and performance evaluation. Although K-Means offers general categorization, Hierarchical clustering reveals finer details, and DBSCAN efficiently distinguishes dense clusters from outliers. The results indicate that integrating preprocessing, feature engineering, and comparative clustering creates a dependable framework for examining extensive research corpora. This study presents a replicable approach that enhances precision, interpretability, and scalability in abstract clustering, enabling more efficient monitoring of developing research trends in rapidly changing scientific domains.**

***Index Terms*—large language models (LLMs), text mining, clustering, unsupervised learning, .**

I. Introduction

The rapid expansion of scientific publications, particularly in fast-paced fields like large language models (LLMs), has created a significant need for automated methods to organize and assess research findings. Because manual reviews are not enough to capture new trends, unsupervised text mining and clustering are crucial for identifying topic structures in large abstract corpora. Although abstract clustering has been the subject of numerous studies, dimensionality reduction, systematic preprocessing, and comparing various classical clustering algorithms remain unexplored.

Kostikova et al. (2025) [1] conducted a large-scale automated literature evaluation that concentrated on the limitations of large language models (LLMs) using a corpus of over 250,000 academic publications from arXiv and ACL sources between 2022 and 2025. Their method, which combines conventional text preparation, dimensionality reduction, and clustering algorithms to arrange abstracts into logical topics, is particularly relevant to the current study.

As the first stage of their preprocessing pipeline, the authors cleaned and filtered abstracts and titles using keyword-based selection. To enhance the corpus, they classified abstracts as either relevant or irrelevant to LLM constraints using massive language models. Textual data was then transformed into numerical representations using embedding models, and the features were improved by extracting keyphrases. Prior to clustering, high-dimensional embeddings were reduced using Principal Component Analysis (PCA) in order to preserve semantic structure and increase computing efficiency.

This work proposes a comprehensive pipeline for finding topics from research abstracts. The technique combines dimensionality reduction techniques, such as PCA, TF-IDF vectorization, and robust text preprocessing, to generate compact feature spaces. In addition to heuristic methods such as the elbow technique, K-Means, Hierarchical, and DBSCAN clustering algorithms are evaluated and compared. Finally, the clustered results are presented and explained using representative abstracts and top terms to guarantee clarity and valuable insights. By combining preprocessing, feature engineering, and comparative clustering experiments, this work

provides an organized framework for improving the accuracy, interpretability, and effectiveness of abstract clustering. In the end, this will enable more accurate tracking of research trends in fields that are changing quickly.

II. Literaure Review

There is a awful requirement for automated techniques to arrange and evaluate enormous sets of research abstracts due to the exponential rise of scientific publications. The development of text mining, dimensionality reduction, and clustering approaches for knowledge discovery is driven by the realization that traditional manual reviews are no longer adequate to catch developing themes. Recent research has shown how well preprocessing processes and unsupervised learning techniques can identify significant structures from textual data.

Movva et al. (2024) [2] applied a pipeline of text preprocessing, embeddings, dimensionality reduction, and clustering to evaluate 16,979 arXiv publications related to LLM. After being cleaned and relevance-checked, the abstracts were integrated using INSTRUCTOR-XL. PCA was used to compress the embeddings to 200 dimensions, and UMAP was used to visualize the results. To create cohesive topic groups, the authors used hierarchical agglomerative clustering (Ward's technique) on the LLM subset and K-means on the entire corpus. However, the study relied on manual adjustment for cluster selection rather than using DBSCAN or systematic heuristics such as the elbow technique.

Ding et al. (2023) [3] used a curated corpus of approximately 198 articles to perform a systematic review of LLM research. To extract thematic clusters, they use transformer-based embeddings in conjunction with topic modeling (BERTopic) and traditional text preprocessing techniques (tokenization, stopword removal, lowercasing, etc.). The elbow approach and density-based algorithms like DBSCAN are not mentioned, nor is there any proof that PCA or other explicit dimensionality-reduction techniques like it are employed in their pipeline. The primary method for clustering is the topic modeling/embedding process, which most likely compares outcomes qualitatively.

III. Methodology

This part outlines the structured approach employed for clustering and examining research abstracts on large language models (LLMs). The method is presented in a sequential format, with each phase explained thoroughly.

A. Overview of Steps

The overall workflow is illustrated in Figure 1. The steps are as follows:

1) **Data Collection:** Gather research abstracts from various websites for conference and journal publications
2) **Data Preprocessing:** In this step, the text data was cleaned and normalized.
3) **Feature Extraction:** A TF-IDF weighted Document-Term Matrix (DTM) was computed here.
4) **Clustering:** Apply multiple clustering algorithms (K-Means, Hierarchical, and DBSCAN).
5) **Dimensionality Reduction:** Reduce the high-dimensional TF-IDF space using PCA for visualization.
6) **Visualization and Interpretation:** Visualize clusters and extract top terms to interpret cluster topics.

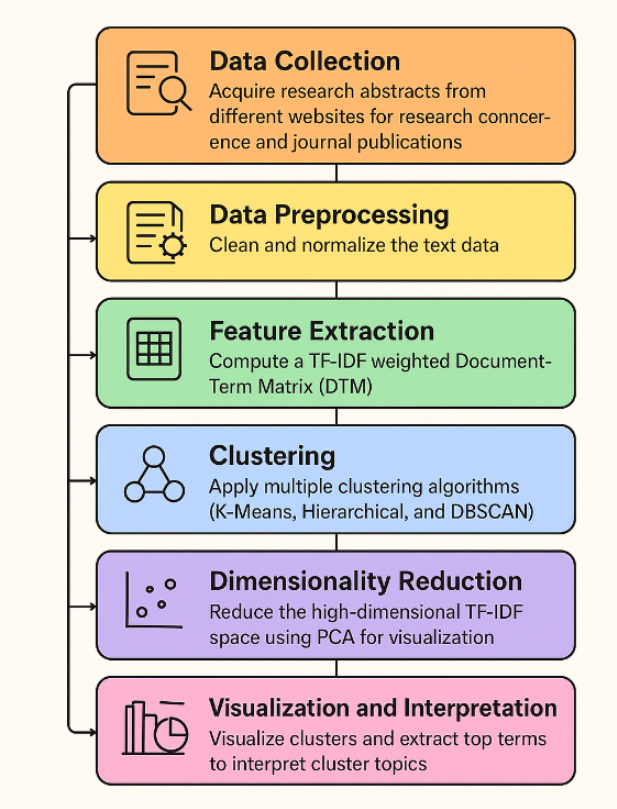

Fig. 1. Stepwise workflow for clustering and analyzing LLM research abstracts.

B. Data Collection

Research abstracts were collected from Google Scholar and ResearchGate, after which a consolidated dataset was prepared in CSV format. This dataset was subsequently imported into the `R` programming environment, where each abstract was assigned a unique document identifier to ensure traceability.

C. Data Preprocessing

A number of preprocessing procedures were methodically applied to produce a polished textual representation suitable for analysis:

- For consistency, all text was changed to lowercase.
- Regular expressions were utilized to eliminate non-alphabetic symbols
- The `tidytext` package was employed to break down the text into separate words.
- Frequent stopwords were eliminated to decrease noise and improve term significance.
- Tokens with less than three characters were eliminated.

- Lexical variation was reduced by implementing lemmatization (`textstem`) and stemming (`SnowballC`) for word normalization

D. Feature Extraction

A Document–Term Matrix (DTM) was constructed using Term Frequency–Inverse Document Frequency (TF–IDF) weighting. This representation attenuates the influence of highly frequent yet semantically uninformative terms, while emphasizing those that contribute more effectively to distinguishing among documents. The resulting DTM was subsequently transformed into a numerical matrix, rendering it suitable for application in clustering algorithms.

E. Clustering Approaches

Three clustering algorithms were employed:

1) **K-Means:** Evaluated for $k = 2$ to 10. The best k was established by maximizing the average Silhouette score and verified using the elbow method. Several random initializations ($nstart = 25$) guaranteed consistency
2) **Hierarchical Clustering:** Ward's linkage (`ward.D2`) was utilized for agglomerative clustering. The dendrogram was sliced to the same k as K-Means for evaluation.
3) **DBSCAN:** Employing Truncated SVD (`irlba`, 50 components), dimensionality reduction was executed. DBSCAN was utilized with minPts $= 15$ and ε set based on the k-nearest neighbor distance graph to detect dense clusters and outlier points

F. Dimensionality Reduction and Visualization

To simplify visualization, the high-dimensional TF–IDF feature space was reduced to two principal components through Principal Component Analysis (PCA). Scatter plots were employed to demonstrate the resulting clusters, with distinct colors representing the clusters generated by each algorithm

G. Cluster Interpretation

Cluster interpretation was facilitated through two complementary visualization techniques.

- **Word Clouds:** Word clouds were generated to display the top 100 terms in each cluster, ranked by their average TF–IDF scores.
- **Bar Charts:** Bar charts were employed to present the top 10 terms per cluster in a ranked format, enabling closer examination.

These visualizations supported semantic labeling of the clusters; for example, clusters oriented toward pretraining prominently featured terms such as "transformer" and "fine-tune", whereas clusters emphasizing evaluation highlighted terms like "benchmark" and "metrics".

IV. Implementation

The system was implemented in `R` (version RStudio 2025.05.1+513) using various libraries to facilitate text processing, clustering, and visualization. Key implementation details are as follows:

A. Programming Environment

- **Software:** RStudio 2025.05.1+513, RStudio IDE.
- **Operating System:** Windows 11.
- **Hardware:** Intel i5 CPU, 8 GB RAM.

B. Libraries and Packages

- `tidyverse` and `tidytext` for data manipulation and text tokenization.
- `tm`, `SnowballC`, and `textstem` for text preprocessing.
- `cluster` and `factoextra` for clustering and silhouette analysis.
- `dbscan` for density-based clustering.
- `irlba` for truncated SVD dimensionality reduction.
- `ggplot2`, `wordcloud`, and `RColorBrewer` for visualization.

C. Parameter Settings

- K-Means: k selected by silhouette score; $nstart = 25$.
- Hierarchical Clustering: Ward's method (`ward.D2`); dendrogram cut at k.
- DBSCAN: ε determined from kNN distance plot; minPts $= 15$.
- PCA: 50 components used for dimensionality reduction prior to DBSCAN and visualization.

D. Visualization

- PCA scatter plots to visualize clusters.
- Word clouds for top 100 terms per cluster.
- Bar charts for top 10 terms per cluster to interpret cluster semantics.

V. Result Analysis

A. Clustering Analysis Using PCA Visualization

The dataset was projected onto a 2D space using Principal Component Analysis (PC1 vs. PC2) to visualize cluster structures. Three clustering algorithms—K-Means, Hierarchical Clustering, and DBSCAN—were applied.

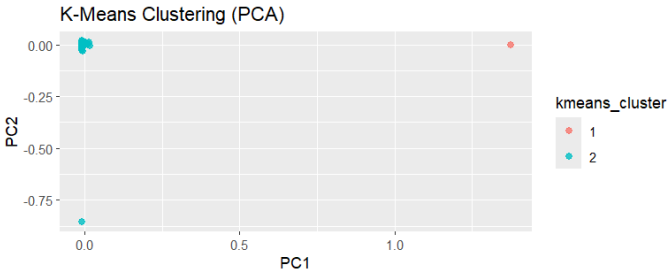

Fig. 2. K-Means clustering results visualized in PCA space.

1) K-Means Clustering: K-Means produced two clusters: 1 (pink) and 2 (cyan). Nearly all points are assigned to the dominant cluster, with a few forming a secondary cluster. This indicates that K-Means captures the main cluster structure but may misclassify outliers or irregularly shaped clusters.

Observation: K-Means presumes spherical shapes for clusters and is not as efficient at identifying sparse points

2) Hierarchical Clustering:

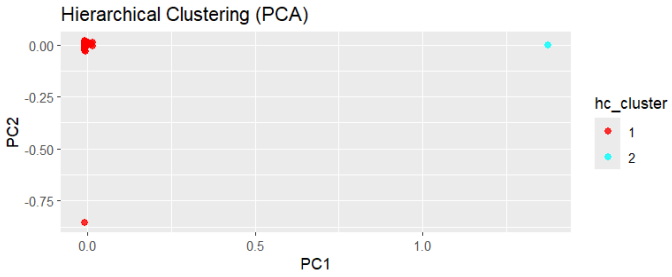

Fig. 3. Hierarchical clustering results visualized in PCA space.

Hierarchical clustering produced two groups: 1 (red) and 2 (cyan). The distribution reveals a primary cluster along with a smaller distinct cluster, akin to K-Means.

Observation: Hierarchical clustering identifies the general structure and can distinguish isolated points, but it is affected by the selection of linkage method

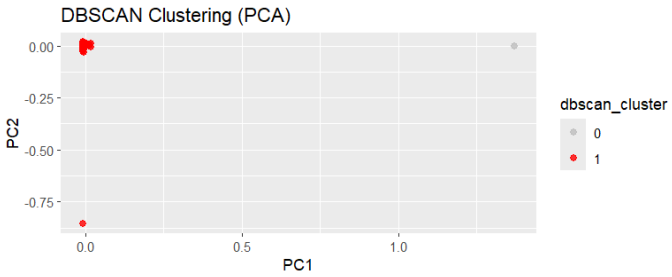

Fig. 4. DBSCAN clustering results visualized in PCA space.

3) DBSCAN Clustering: The DBSCAN visualization presents two cluster designations: 0 (gray) and 1 (red). The majority of points are closely packed in cluster 1, whereas some are categorized as noise or outliers in cluster 0. This shows that DBSCAN effectively identifies the primary dense area while categorizing sparse points as noise.

Observation: DBSCAN performs well with datasets that have varying density, as shown by the limited count of points identified as outliers.

All three algorithms detect a significant dense cluster and a lesser group of sparse points. K-Means identifies the main cluster structure but struggles more with detecting outliers. Hierarchical clustering establishes a distinct hierarchical relationship but relies on linkage criteria. DBSCAN is highly proficient in detecting outliers

B. Clustering Analysis Using Word Clouds Visualization

1) K-Means Clustering Using Word Clouds: K-Means produced two distinct clusters. Cluster 1 is a broad technical and miscellaneous category containing terms like `sql`, `html`, and `trialgpt`. Cluster 2 is a thematically coherent group focused on evaluation and human judgment, with key words being `critique`, `correct`, and `human`. The algorithm successfully isolated one primary topic.

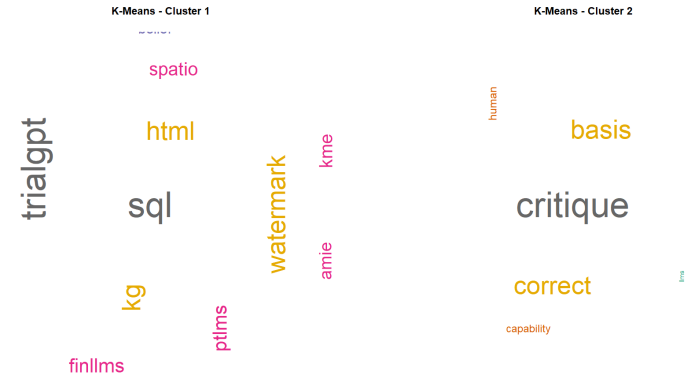

Fig. 5. K-Means Clustering Results.

2) Hierarchical Clustering Using Word Clouds: This method also identified two main themes. Cluster 1 contains technical terms like `sql` and `ptlms`.Cluster 2 is a highly concentrated evaluation cluster, focusing tightly on `critique` and `correct`. It provides a more granular result than K-Means.

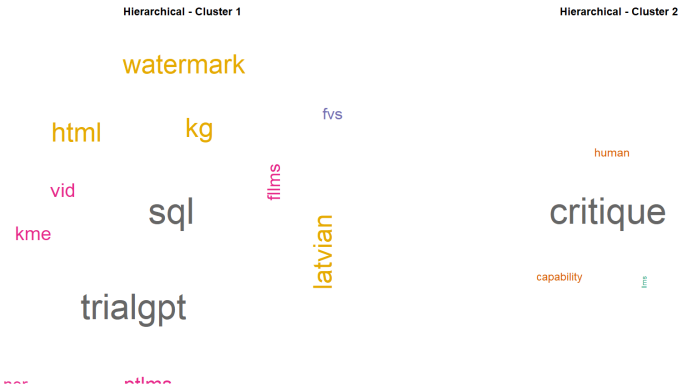

Fig. 6. Hierarchical Clustering Results.

3) DBSCAN Clustering Using Word Clouds: As a density-based method, DBSCAN created very specific, dense clusters. Cluster 0 is almost singularly focused on the word `critique`. Cluster 1 groups short technical terms like `sql` and `kg`. This approach excels at isolating core, high-density concepts.

All three algorithms successfully identified a core theme of evaluation.

- **K-Means** provides a broad categorization.
- **Hierarchical** provides a more concise view.
- **DBSCAN** isolates the purest and most dominant topics.

C. Quantitative Analysis of Clustering

This analysis examines the clusters produced by K-Means, Hierarchical, and DBSCAN algorithms. The results are presented as bar charts displaying the top 10 terms for each

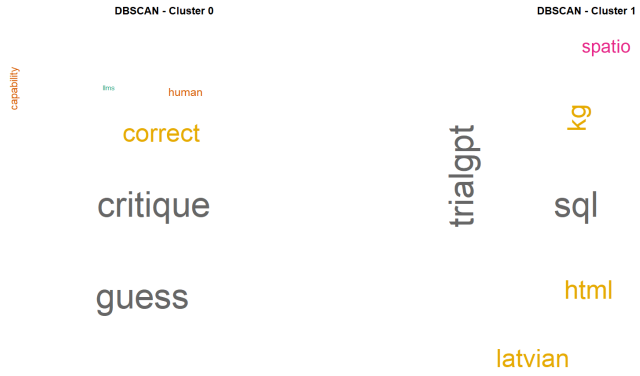

Fig. 7. DBSCAN Clustering Results.

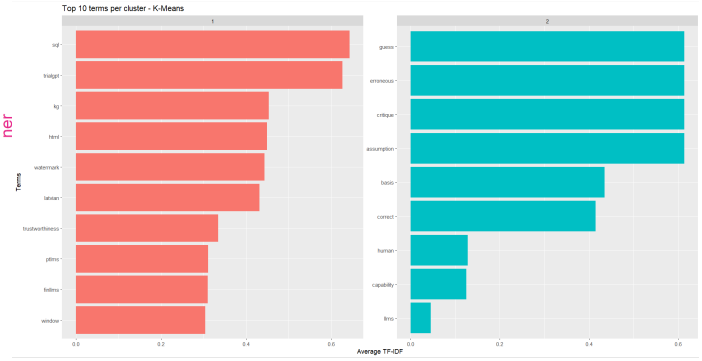

Fig. 8. K-Means clustering top terms for each cluster.

cluster, ranked by their average TF-IDF score, which measures term importance.

D. Clustering Results Analysis by Bar Chart

The following data shows the leading 10 terms for each cluster determined by three distinct clustering methods: Hierarchical, K-Means, and DBSCAN. The significance of each term in its cluster is measured by its average TF-IDF score. A review of these findings shows an impressive uniformity across all three approaches, indicating a robust and clearly defined thematic framework within the foundational text data.

1) K-Means Clustering Outcomes: When using $k = 2$, the K-Means algorithm yielded clusters closely aligned with those discovered by the Hierarchical approach. This alignment highlights the consistency and strength of the dataset's thematic framework.

- **Cluster 1: Specialized Language and Technical Terminology**
  Similar to Hierarchical clustering, this group includes specialized terminology like `sql`, `trialgpt`, and `html`, highlighting its emphasis on particular software platforms, AI technologies, and technical conversations.
- **Cluster 2: Language of Concepts & Evaluation**
  This cluster reflects Hierarchical Cluster 2, comprising abstract, evaluative words such as `guess`, `incorrect`, and `critique`. The uniformity among methods suggests a dependable categorization of documents that evaluate system performance, beliefs, and essential reasoning.

2) Hierarchical Clustering Results: The Hierarchical clustering method successfully split the dataset into two separate, thematically consistent clusters.

- **Cluster 1: Specialized Language & Technical Terminology**
  This group is characterized by specific technical terms and recognized entities. Essential terms like `sql`, `trialgpt`, `kg`, `html`, and `watermark` suggest that it mainly clusters documents related to software technologies, AI models, and specialized concepts.
- **Cluster 2: Conceptual and Assessment Language**
  The second cluster highlights terms associated with reasoning, evaluation, and performance measurement. Key terms encompass `guess`, `incorrect`, `critique`, `hypothesis`, and `foundation`, indicating material centered on assessing precision, constraints, and evaluative quality, frequently relating to AI or human assessment.

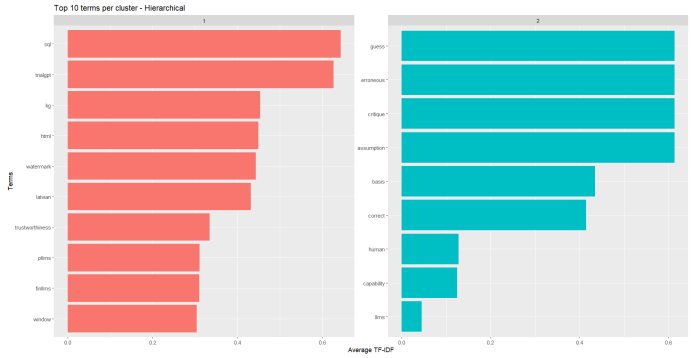

Fig. 9. Hierarchical clustering top terms for each cluster.

3) DBSCAN Clustering Outcomes: DBSCAN, a clustering method based on density, reinforces the existence of two main thematic categories. Even though the cluster labels vary, the content and structure closely match those of the earlier methods.

- **Cluster 0: Conceptual and Assessing Language**
  Designated as "0" by DBSCAN, this cluster aligns with Cluster 2 from Hierarchical and K-Means clustering. Terms like `conjecture`, `incorrect`, `assessment`, and `presumption` highlight its emphasis on evaluation, reasoning, and critical analysis.
- **Cluster 1: Specialized Language & Technical Domains**
  This cluster corresponds to Cluster 1 from the other approaches, featuring technical terms centered around `sql`, `trialgpt`, `kg`, and `html`. It reliably signifies documents addressing software, AI models, and other subject-specific themes.

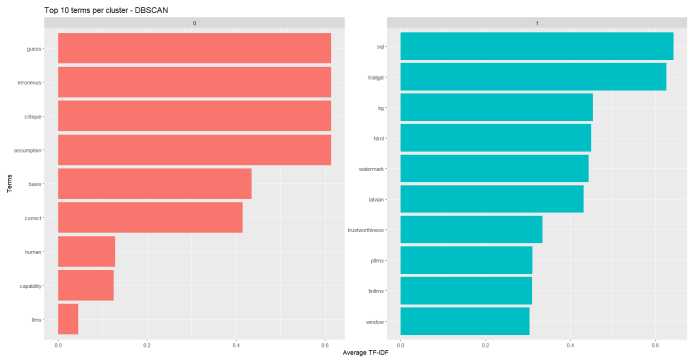

Fig. 10. DBSCAN clustering top terms for each cluster.

VI. Conclusion

This research offered a systematic method for grouping research abstracts in the area of large language models (LLMs) through thorough preprocessing, TF-IDF feature extraction, PCA-driven dimensionality reduction, and comparative analysis of K-Means, Hierarchical Clustering, and DBSCAN. The findings consistently showed two main thematic categories: one focusing on technical jargon associated with software tools and AI models, and the other concentrated on evaluative language relating to critique, precision, and effectiveness. K-Means delivered general classifications, Hierarchical Clustering provided more nuanced differences, and DBSCAN successfully detected compact clusters and separated outliers. Utilizing these approaches, the study exhibited a strong and adaptable structure for arranging extensive groups of abstracts, thus aiding in more precise tracking of new research trends. The results highlight the importance of integrating preprocessing, feature engineering, and clustering to enhance the interpretability and dependability of literature analysis in fast-changing scientific disciplines

References

[1] A. Kostikova, Z. Wang, D. Bajri, O. Pütz, B. Paaßen, and S. Eger, "Lllms: A data-driven survey of evolving research on limitations of large language models," *arXiv preprint arXiv:2505.19240*, 2025.

[2] R. Movva, S. Balachandar, K. Peng, G. Agostini, N. Garg, and E. Pierson, "Topics, authors, and institutions in large language model research: trends from 17k arxiv papers," *arXiv preprint arXiv:2307.10700*, 2023.

[3] Q. Ding, D. Ding, Y. Wang, C. Guan, and B. Ding, "Unraveling the landscape of large language models: a systematic review and future perspectives," *Journal of Electronic Business & Digital Economics*, vol. 3, no. 1, pp. 3–19, 2024.