International Journal of Basic and Applied Sciences, 14 (SI-1) (2025) 529-537

SPC

International Journal of Basic and Applied Sciences

Website: www.sciencepubco.com/index.php/IJBAS
https://doi.org/10.14419/7cvtfq26
Research paper

Clustering Algorithms for Queries: A Comparative Analysis of Farmer Call Center Data

C. Kiruthiga [1] *, Dr. K. Dharmarajan [2]

[1] Research Scholar, Vels Institute of Science Technology & Advanced Studies (VISTAS), Assistant Professor, PG Department of Information Technology and BCA, Dwaraka Doss Goverdhan Doss Vaishnav College, Affiliated to the University of Madras, Chennai. Chennai, India
[2] Professor, Department of Information Technology, Vels Institute of Science Technology & amp; Advanced Studies (VISTAS), Chennai, India
**Corresponding author E-mail: csnkirthi@gmail.com*

Received: July 15, 2025, Accepted: July 24, 2025, Published: November 1, 2025

Abstract

Extracting insights from queries and feedback helps identify trends, enhance products and services, personalize customer interactions, and craft effective marketing strategies. Data clustering, a powerful method, organizes unstructured data and refines queries by offering suggestions based on similar or related inputs, ultimately enhancing the search experience. This study compares the performance of several clustering algorithms, including Agglomerative Clustering, K-Means (KM), Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN), and Density-Based Spatial Clustering of Applications with Noise (DBSCAN), as well as various embeddings, such as Term Frequency-Inverse Document Frequency (TF-IDF)—Sentence-Bidirectional Encoder Representations from Transformers (SBERT), Word2Vec, and GloVe. The Calinski-Harabasz Index, Davies-Bouldin Index, and Silhouette Score are used to measure the effectiveness of these algorithms. Results indicated that HDBSCAN outperformed other clustering algorithms within the farmer helpline dataset. The conclusions were derived from the medium-level performance of clustering algorithms. The findings showed that HDBSCAN, combined with different embeddings, achieved a Silhouette Score of 0.85, a Davies-Bouldin Index of 0.66, and a Calinski-Harabasz Index of 4,239.9.

Keywords: *Agglomerative Clustering; Calinski-Harabasz Index; DBSCAN; HDBSCAN; GloVe; K-meansSBERT; Silhouette Score; TF-IDF; Word2Vec.*

1. Introduction

Customer query processing is a crucial component of customer service and support systems. It involves efficiently handling, analyzing, and resolving customer inquiries received through various channels, including email, social media, and phone calls.

Growing awareness of organic vegetables has led individuals to develop their own in small spaces, such as terraces, balconies, and outdoor areas around their homes. After the epidemic, many people have gotten into the agricultural industry. New Generation farmers often need assistance and guidance in their farming practices. The Kisan Call Center for Farmers is a government initiative designed to support farmers in agricultural, horticultural, and floricultural activities. Varieties, sowing, weeding, weather, fertilizers, soil testing, nutrient management, plant protection, rotational crop schemes, credits, loans, and other related areas are the topics on which farmers frequently raise questions from various parts of India.

New technologies in agriculture, such as smart farming, generate a significant amount of data. Analyzing data can yield valuable insights into the common issues farmers face and help improve the support provided to them. The proliferation of data has necessitated the development of robust data analysis techniques, including statistical methods, ML algorithms, and DL models used to analyze vast volumes of data.

The research has extensively addressed structured queries; however, processing unstructured queries remains challenging, particularly in areas such as customer care. In the Farmer Helpline data, there are 11 attributes, two of which are unstructured: Query Text and KeyAns. Farmer queries are recorded under the Query Text attribute, while answers are stored under KeyAns. Although the queries are categorized with labels, additional subcategories are included within these labels for further classification. Grouping similar queries based on crop and their query type can be done using clustering algorithms. Clustering algorithms are potent tools for grouping similar data points, and they can be particularly effective in categorizing farmer queries in helpline data. By clustering these queries, we can identify prevalent problems, seasonal trends, and region-specific issues, which can inform better decision-making and resource allocation.

2. Literature Review

This section presents related research on text similarity clustering, drawing on document clustering, queries, and research articles from 2020 to 2023.

The "Comparative Analysis of Clustering Methods" explores hierarchical and partitional clustering approaches. Notable advancements have been achieved in the process of initial seeding and centroid selection. Additionally, hierarchical clustering algorithms have been optimized to reduce their complexity to log-linear, enhancing their suitability for analyzing high-dimensional data. This is supported by the fact that K-Means (0.1271) performs faster than others [1]—a key finding in the context of clustering and cluster validation techniques within the domain of emotional intelligence datasets. The evaluation of the Reclust algorithm performed highlights its superiority over other methods, including K-means, Expectation-Maximization, Hierarchical Clustering, and Self-Organizing Map, which it produces with precision (0.986) [2].

Query datasets of various sizes can be clustered using the Apache Spark and Apache Hadoop frameworks, along with an access plan recommendation approach based on MapReduce. The Spark platform's enhanced capabilities enable large-scale query sets to be processed more efficiently.[3].

Clustering algorithms for literature and university datasets are utilized in literature query services. The system's retrieval efficiency, achieved using R-tree structures and the K-Means algorithm, yielded an efficiency score of 875, along with precision, recall, and F1 scores of 86.3%, 88.6%, and 87.4%, respectively.[4].

Embedding Words in a Clustering Method for Large Datasets, such as K-means and Agglomerative, with fine-tuning of the BERT model to specific datasets, could yield more accurate contextual word embeddings.[5].

Ontology-based Clustering optimizes memory usage, execution time, and processing speed, achieving an accuracy rate of 96%. [6].

Analyzing different methods of Clustering based on documents that leverage semantic similarity measures, including Cosine, Jaccard, Euclidean, Dice, and TF-IDF. The clustering techniques examined various clusters using Hierarchical Clustering, Fuzzy C-means, K-means, and Bisecting K-means. According to the survey, the clusters are accurate and dependable. [7]. Another approach for document clustering, which uses a fuzzy-cluster-based semantic model, achieves an accuracy of approximately 89% and a recall of around 88%.[8].

TREC Web track, such as Interp-f Query Concat, Statistical approaches, Initial, ClustMRF, GeoClust, and Interp-f were used to determine that documents are essential to the user's inquiry by using the correct information from various documents, particularly in a record-based cluster retrieval system, and attaining a precision rate of 89% [9].

Analyzing algorithms based on various types of calculations, such as cosine, Jaccard, Euclidean, Dice, TD-IDF, and clustering algorithms. K-means and SOM (Self-organizing Maps) were the most suitable among the methods analyzed for simulated and real data sets.[10]. The Geostatistical Fuzzy C-Means algorithm is more effective and optimized for execution time in distributed databases.[11]

Table 1: Comparative Study of Performance of the Clustering Algorithms with Different Datasets

Sno		Title	Dataset	Algorithm Compared	Result
1.	1	Comparative analysis of clustering methods	Artificial dataset	K-means algorithm Hierarchical algorithm	Execution time: Hierarchical algorithm – 0.8864 K-Means- 0.1271
2.	2	Replastering mixed-based and validation of the cluster value	Emotional intelligence	Hierarchical Cluster, expectation-maximization.	Precision-0.986
3.	3	Evaluate the performance and recommendations with Apache Hadoop and Apache Spark.	Different sizes of queries	Apache Spark and Apache Hadoop frameworks	Execution time: for nine nodes, Hadoop -235 sec Spark- 39 sec
4.	4	clustering algorithm for massive scientific value in LQS	University Science Technology Literature datasets	K-means clustering Algorithm, R-tree clustering model (proposed model)	Precision 86.3
5.	5	Text clustering technique for a large dataset	Research article 20NG-long-8131 documents	K-means, agglomerative Clustering, and DBSCAN	Execution time: 2.274 sec
6.	6	QAOC: novel query analysis clustering for data management in Hadoop	Bottleneck issues in Hadoop	Weighted ontology-based clustering method	Accuracy -96%
7.	7	State-of-the-art document clustering algorithms based on semantic similarity	Research article	hierarchical Clustering, bisecting k-means, fuzzy c-means, k-means	Accuracy -96%
8.	8	A cluster semantic information retrieval system	Obtaining documents that align with the user's request	IR model -proposed model	Precision - 89%
9.	9	Cluster-Based Document Retrieval with Multiple Queries	TREC- Web rack (text retriever conference	Statistic approach, Initial, ClustMRF, GeoClust, Interp-f	Execution time:377 sec
10.	10	Clustering algorithms: a comparative approach	Stimulated and real data sets	Cosine, Jaccard, Euclidean hierarchical Clustering, affinity propagation, and spectral Clustering. k-means	K-Means and SOM are the best methods.
11.	11	A time-efficient clustering algorithm for query optimization in a distributed database [11]	Data is fragmented and replicated to several sites in the form of load and chunks, 50 different Query Execution plans.	Geostatistical fuzzy c-mean	Execution time- 28 sec

Although several clustering algorithms, such as K-Means and DBSCAN, have been widely researched and used elsewhere, these algorithms also have their limitations. Despite its popularity due to its simplicity and efficiency, K-Means is quite sensitive to initialization, which can

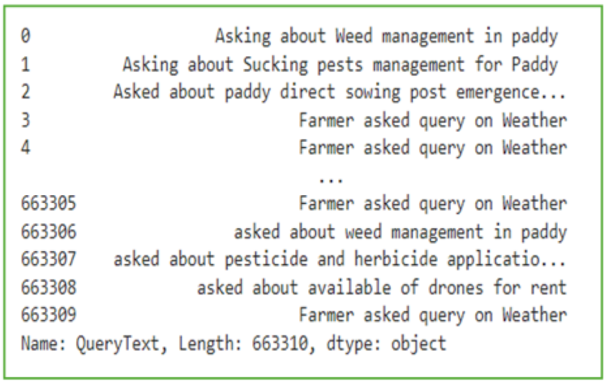
lead to an inefficient clustering process, especially in datasets with different densities and Noise. These problems drive the quest to develop stronger clustering algorithms, as noted in earlier research (e.g., the sensitivity of K-Means in [1]) and other studies.
To address these shortcomings, the present study combines HDBSCAN, which does not require the specification of a specific number of clusters and is highly resistant to initialization issues. HDBSCAN is ideal for working with noisy data and identifying clusters of any shape, making it the most appropriate choice for the Farmer's query data in this research. Also, DBSCAN, introduced in, addresses such obstacles using the density-based clustering methodology, which is particularly helpful in clustering complex and noisy data.
Word2Vec, as proposed in the word embedding case, provides high-quality word representations in the form of vectors, which can capture semantic relationships and improve clustering quality. The embeddings are especially applicable to the analysis of agricultural queries, where context and meaning are crucial. Together with Word2Vec, the proposed study should serve as a more effective way of processing unstructured query information, allowing for the overcoming of the weaknesses of conventional clustering algorithms and making the findings more accurate and relevant.

3. Materials and Methods

The Tamil Nadu call records used in the study were obtained from the Call Center between January 2020 and December 2023.

3.1. Description of the dataset

The dataset comprises 83 distinct varieties of crops, each associated with 63 unique types of queries.



```
0                        Asking about Weed management in paddy
1                   Asking about Sucking pests management for Paddy
2                  Asked about paddy direct sowing post emergence...
3                                    Farmer asked query on Weather
4                                    Farmer asked query on Weather
                                               ...
663305                               Farmer asked query on Weather
663306                          asked about weed management in paddy
663307       asked about pesticide and herbicide applicatio...
663308                      asked about available of drones for rent
663309                               Farmer asked query on Weather
Name: QueryText, Length: 663310, dtype: object
```

Fig. 1: Sample Dataset.

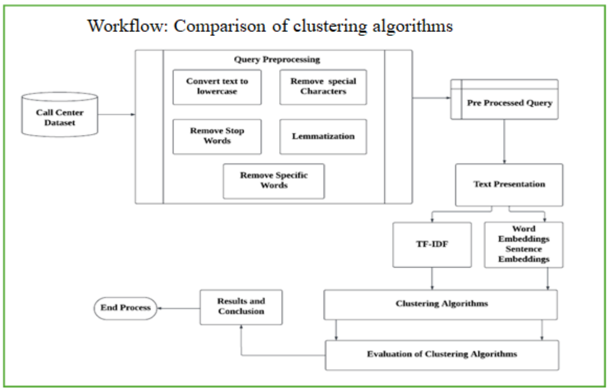


Fig. 2: Workflow of the Study.

3.2. Preprocessing methods

3.2.1. Feature transformation

3.2.1.1. Normalization

- Lowercasing: All characters should be changed to lowercase to maintain consistency.
- Lemmatization: This technique reduces words like "needed" to "need" and "pests" to "pest," which helps text-based models operate more efficiently by lowering the dimensionality of text data.

3.2.1.2. Removing noise

- Stop erasing general elements that do not provide much meaning to the parameters. (e.g., "in", "on").
- Punctuation removal involves removing punctuation marks.
- Special Characters and Specific Words Removal: Removing non-alphanumeric characters and specific words that misguide the cluster process.

3.2.1.3. Feature selection

Paddy Dhan and Nutrient Management have been selected for further examination due to the recurring inquiries from farmers regarding crop-related issues.

3.3. Text presentation

3.3.1. TF-IDF (term frequency-inverse document frequency)

In text processing using NLP, TF-IDF is a standard method for determining the importance of a document in a corpus of documents. It is frequently employed in text mining, classification, and information retrieval.
TF evaluates term frequency (t) appearing in a document (d).

$$\mathrm{TF(t,d)} = \frac{\text{Number of times t appears in d}}{\text{Total number of terms in d}} \tag{1}$$

IDF gives less weight to standard terms and more weight to rare ones.

$$\mathrm{IDF(t)} = \log\left(\frac{\text{(Total number of documents}}{\text{) Number of documents contain t}}\right) \tag{2}$$

TF-IDF Calculation,

$$\mathrm{TF - IDF(t,d) = TF(t,d) \times IDF(t)} \tag{3}$$

3.3.2. Word embeddings and sentence embeddings

Embeddings effectively represent words, phrases, or other discrete things in continuous vector spaces in ML and NLP. This method's ability to capture the relationships and semantic meanings between objects is helpful for text classification, sentiment analysis, and other applications.
Word2Vec: Word2Vec is a predictive model that utilizes neural networks to learn word vectors by predicting words based on their context. Continuous Bag of Words (CBOW) and Skip-gram are two variations of Word2Vec. Skip-gram forecasts context words from a desired word, whereas CBOW concentrates on predicting a desired word from its context.

vector("organic")−vector("pesticide")+vector("health")≈vector("nutrition")

GloVe: GloVe is a word embedding technique that utilizes statistical information from the entire corpus. GloVe detects the record data based on matrix factorization of the document co-occurrence matrix.GloVe is more memory-efficient and can be trained on large corpora with relatively less computational power. It defines a cost function that models the ratio of word co-occurrence probabilities, comparing the actual co-occurrence possibility to the estimated likelihood from the word vectors at a low level.

$$\mathrm{J} = \sum_{j,k=1}^{V} f(Z_{jk})\left((w_j^T . w_k) + b_j + b_k - \log(Z_{jk})\right)^2 \tag{4}$$

Z_{ij} Is the co-occurrence of the counts of words j and k. w_j and w_k are the word vectors of words j and k, Respectively. BJ and BK are biases associated with each word. $f(Z_{jk})$ It is a weighting function that reduces the influence of persistent word pairs.
SBERT: Sentence-BERT is an adapted version of the Bidirectional Encoder Representations from Transformers (BERT), designed for tasks involving the generation of sentence embeddings. These tasks encompass semantic similarity, Clustering, and information retrieval. SBERT is employed to categorize similar texts unsupervised and effectively align questions with pertinent answers.

3.4. Cluster algorithms

K-means is an unsupervised learning algorithm for Clustering. K-means categorizes objects into groups that share similarities but differ from those in other groups. The distance of each point to both centroids is calculated to predict the long-range relationship between the data values and the randomly selected centroids. The centroid with the shortest distance is given to each point. After assigning all the data points, the centroid of every Cluster is recalculated as the mean of all the values within it. This new one replaces the old centroid. When the centroids no longer shift significantly, data points are allocated to a small range around the center point of the object, and the centroids are updated continuously. Alternatively, the data points assigned to clusters remain the same. The algorithm splits the values of the data into K clusters and assigns each end to the Cluster with the closest centroid.
Calculating the new Cluster's centroid

$$\mu_i = \frac{1}{n_i}\sum_{j=1}^{n_i} x_j \tag{5}$$

Hierarchical Clustering is a clustering technique used to construct a hierarchy of clusters. It is classified into two types: Agglomerative and Divisive. In Agglomerative Clustering, the two closest clusters are identified based on the broadest range and linked to the center point of the performance. More clusters are merged into a single cluster of the processor. The distances between the new Cluster and all other existing clusters are recalculated. Integrating the closest clusters is repeated until all the data points are combined into a single cluster or until the desired number of clusters is reached. In Divisive Clustering, all data points start in a single cluster. The large Cluster is split into smaller clusters, and the distances are recalculated. Clusters are split until each data point is assigned to its own Cluster or until the desired number of clusters is reached.

- DBSCAN is suitable for applications with data containing Noise and clusters of varying shapes and sizes. The density of values in the data serves as the basis for identifying clusters using arbitrary shapes, and the number of clusters can be calculated automatically.

DBSCAN uses arbitrary shapes to find clusters. DBSCAN is well-suited for large datasets and can handle situations where clusters vary in density.

Calculating distance between all points using the Euclidean distance formula:

$$\mathrm{Distance}(s,t) = \sqrt{(a_2 - a_1)^2 + (b_2 - b_1)^2} \tag{6}$$

HDBSCAN cluster aims to create a hierarchy of groups. There are two methods of Hierarchical Clustering: divisive and agglomerative. Agglomerative is a bottom-up approach that enhances the DBSCAN algorithm's version for handling complex datasets with varying densities. HDBSCAN treats points in low-density areas as Noise and those with high density as clusters, creating a hierarchical dendrogram that can be used to extract clusters at different levels of granularity. HDBSCAN initially computes the core distance to its k-th nearest neighbor, where $k = \mathrm{MinPts}$.Then, it computes Mutual Reachability,

$$\text{Mutual reachability distance}(s,t) = \max\big(\text{core distance}(s), \text{core distance}(q)\big) \tag{7}$$

3.5. Cluster evaluation metrics

Calculated to the quality range of clusters is an essential step in any clustering process. Cluster evaluation techniques help determine how well the clustering algorithm has grouped the data values, whether the resulting clusters are meaningful, and how effectively the clusters separate from one another. Evaluating the efficiency and performance of a clustering algorithm can involve several metrics and approaches.

Intrinsic Quality Indicators

The quality of the clusters is evaluated based solely on the data, without the use of external information (ground truth labels), in these metrics.

a) Silhouette Score

- Measures the value of similarity between a point and its Cluster compared to other clusters.

Calculation of point:

$$\text{Silhouette score (a)} = \frac{y(a) - x(a)}{\max(x(a), y(a))} \tag{8}$$

Where $x(a)$ is the average proximity of a to every other point( )Yes, the smallest mean long range of the values and all kinds of values in the neighboring Cluster.

A higher value indicates better Clustering; values range from -1 to 1. The point is well-clumped if the value is near 1. The value is on or near the decision margin between two clusters if its value is close to zero. The value was assigned to the incorrect Cluster if the value is negative.

b) Davis-Bouldin Index

- calculates the same mean score between every Cluster and its more types

The formula for calculating the Davis-Bouldin Index is

$$\text{Davis-Bouldin Index} = \frac{1}{k}\sum_{i=1}^{k} \max_{j \neq i}\left(\frac{\sigma_i + \sigma_j}{d(c_i, i_j)}\right) \tag{9}$$

Where σ_iSpacing among all values in the Cluster i to the center of the point of i. $d(c_i, i_j)$Spacing between the centroids of clusters i and j.

c) Calinski-Harabasz Index

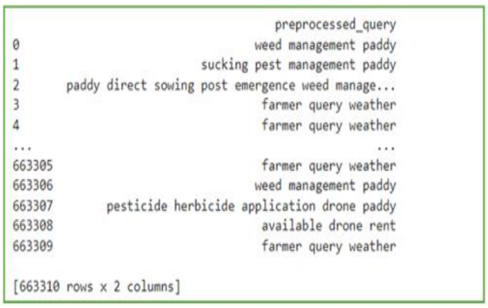
- The Calinski-Harabasz Index, also known as the " different percentage benchmark ", is a metric used to calculate clustering quality internally by calculating the proportion of inter-cluster variability to intra-cluster variability. A higher Calinski-Harabasz score indicates that the clusters are well-distinguished, and the points within each Cluster are close to their respective centroids.
- The formula for calculating the Calinski-Harabasz Index is:

$$\text{Calinski} - \text{Harabasz Index} = \frac{t(B_k)}{t(W_k)} \times \frac{n-k}{k-1} \tag{10}$$

Where B_k Measures the dispersion of the cluster centers (centroids) relative to the overall centroid of the dataset. W_{eek}Calculations involve separating data values within each Cluster around their respective cluster centroids. n, number of data points. k, number of clusters.

4. Results and Discussion

The following is the output of the Querytext preprocessing.



	preprocessed_query
0	weed management paddy
1	sucking pest management paddy
2	paddy direct sowing post emergence weed manage...
3	farmer query weather
4	farmer query weather
...	...
663305	farmer query weather
663306	weed management paddy
663307	pesticide herbicide application drone paddy
663308	available drone rent
663309	farmer query weather

[663310 rows x 2 columns]

Fig. 3: Preprocessed Query.

Figure 3: Special characters, stop words, and specific terms like "paddy"(since processing only paddy query) and "asked" were removed from all statements, and all remaining words were converted to lowercase.
Performance across different embeddings with various clustering algorithms

Table 2: Comparison of Embeddings with Clustering Based on Silhouette Score(high)

Clustering Technique	SBERT	TF-IDF	Word2Vec	GloVe
K-Means (n=450)	0.76	0.79	0.81	0.79
Agglomerative (n=450)	0.78	0.8	0.81	0.79
DBSCAN	0.19	0.86	0.73	0.75
DBSCAN (Excluding Outliers)	0.2	0.71	0.73	0.94
HDBSCAN	0.77	0.75	0.66	0.75
HDBSCAN (Excluding Outliers)	0.84	0.87	0.84	0.83

Figure 4, Table 2: A high silhouette score is desirable, and embeddings such as K-means and agglomerative Clustering show strong performance when the values for k and n match the no of clusters identified by DBSCAN. Among the embeddings, Word2Vec (0.80) and TF-IDF (0.79) yield the highest scores, followed by GloVe, with SBERT (0.76) ranking fourth. By gradually increasing the k and n values to 5, 25, 100, 300, and eventually 450, a point was reached where the metric values began to decline, indicating the optimized k value had been reached. K-means performance improves as the no of clusters increases. DBSCAN, including outliers, performs best with TF-IDF (0.86), surpassing the other embeddings. Excluding outliers further enhances DBSCAN's performance, with GloVe (0.94) achieving the highest score, while SBERT (0.25) proves ineffective in conjunction with DBSCAN. HDBSCAN (excluding outliers) consistently delivers the best results across all embeddings, particularly with TF-IDF (0.87), SBERT (0.84), Word2Vec (0.84), and GloVe (0.83).

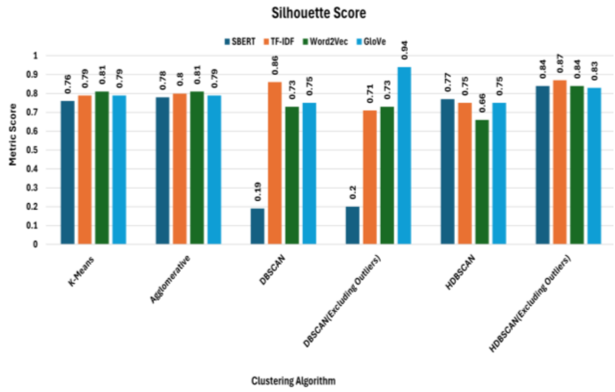


Fig. 4: Embeddings with Clustering Based on Silhouette Score.

Table 3: Comparison of Embeddings with Clustering Based on Davies-Bouldin Index(Low)

Clustering Technique	SBERT	TF-IDF	Word2Vec	GloVe
K-Means Clustering (k=450)	1.42	1.69	1.18	1.41
Agglomerative (n=450)	1.62	1.77	1.21	1.48
DBSCAN	1.25	0.54	0.99	1.09
DBSCAN (Excluding Outliers)	0.82	1.16	0.99	0.57
HDBSCAN	1.26	1.35	0.16	1.18
HDBSCAN (Excluding Outliers)	0.8	0.68	0.57	0.59

Figure 5, Table 3: In the comparison of clustering techniques using different embeddings based on the Davies-Bouldin Index, K-Means clustering and Agglomerative Clustering with k=450 shows that TF-IDF performs the best, followed by SBERT, GloVe, and Word2Vec. When applying DBSCAN, SBERT (1.25) performs best, followed by GloVe (1.09) and Word2Vec (0.99), while TF-IDF (0.54) performs poorly. Excluding outliers from DBSCAN leads to a reversal, with TF-IDF (1.16) performing best, while Word2Vec (0.99), SBERT (0.82), and GloVe (0.59) score lower. For HDBSCAN, TF-IDF (1.30) continues to outperform, followed by GloVe (1.18), SBERT (1.26), and Word2Vec (0.16). Excluding outliers from HDBSCAN alters the landscape, with SBERT (0.80) performing best, followed by TF-IDF (0.68), GloVe (0.59), and Word2Vec (0.57).

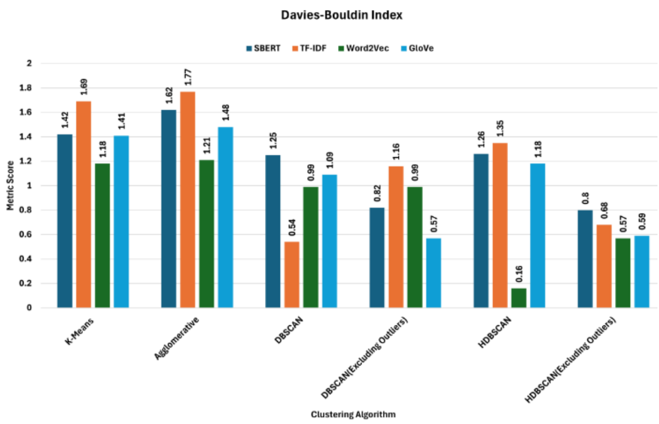


Fig. 5: Embeddings with Clustering Based on the Davies-Bouldin Index.

Table 4: Comparison of Embeddings with Clustering Based on Calinski-Harabasz Index(High)

Clustering Technique	SBERT	TF-IDF	Word2Vec	GloVe
K-Means Clustering (k=450)	686.32	640.91	1919.61	896.78
Agglomerative (n=450)	699.02	635.32	1908.01	910.36
DBSCAN	157.17	2547.88	324.41	284.89
DBSCAN (Excluding Outliers)	191.45	438.99	325.26	7299.87
HDBSCAN	257.95	728.2	487.07	406.62
HDBSCAN (Excluding Outliers)	2888.65	3567.49	2772.656	7730.77

Figure 6, Table 4: When comparing the Calinski-Harabasz Index across various clustering techniques, Word2Vec (1919.61) achieves the highest score with K-Means clustering (k=450), followed by GloVe (896.79), SBERT (686.32), and TF-IDF (640.91). A similar trend is observed with agglomerative Clustering (n = 450), where Word2Vec (1908.01) leads, followed by GloVe (910.36), SBERT (699.02), and TF-IDF (635.32). DBSCAN shows a significant contrast, with TF-IDF (2547.87) achieving a much higher score than the other embeddings, while Word2Vec (324.41), GloVe (284.89), and SBERT (157.17) fall behind. Excluding outliers with DBSCAN, GloVe (7299.87) stands out with an exceptional score, while the other embeddings, such as TF-IDF (438.99), Word2Vec (325.26), and SBERT (191.45), perform significantly lower. HDBSCAN without outliers further emphasizes GloVe's (7730.77) dominance, with TF-IDF (3567.49), SBERT (2888.65), and Word2Vec (2772.66) following, including outliers in HDBSCAN results in TF-IDF (728.20) performing best, followed by Word2Vec (487.07), GloVe (406.62), and SBERT (257.95.

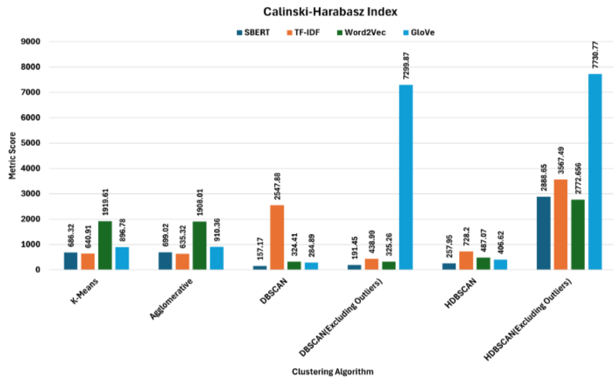


Fig. 6: Embeddings with Clustering Based on the Calinski-Harabasz Index.

Average performance of the clustering algorithms with embeddings

The performance of various clustering algorithms was assessed using different embeddings. In some instances, it was challenging to determine the effectiveness of the clustering algorithms due to discrepancies in the evaluation metrics. Consequently, the average scores for each algorithm, along with their corresponding embeddings, were calculated, and conclusions were drawn based on these average values. Overall, HDBSCAN demonstrated superior performance compared to the other clustering algorithms.

Table 5: Comparison of Embeddings with Clustering Based on Calinski-Harabasz Index

Clustering Technique	Silhouette score	Davies-Bouldin Index	Calinski- Harabasz Index
K-Means	0.79	1.43	1035.91
Agglomerative	0.8	1.52	1038.18
DBSCAN	0.63	0.97	828.59
DBSCAN (Excluding Outliers)	0.65	0.89	2063.89
HDBSCAN	0.73	0.99	469.96
HDBSCAN (Excluding Outliers)	0.85	0.66	4239.9

Figure 7, Based on the performance rankings across different clustering techniques, HDBSCAN demonstrates the best results with GloVe outperforming all other embeddings, followed by Word2Vec, TF-IDF, and SBERT. In K-Means clustering, Word2Vec takes the lead, followed by GloVe, SBERT, and finally TF-IDF. For DBSCAN, TF-IDF outperforms the rest, followed by Word2Vec, GloVe, and SBERT in descending order. Similarly, Agglomerative Clustering places Word2Vec at the top, followed by GloVe, SBERT, and TF-IDF.
Performance ranking on Clustering Algorithms with Embeddings.

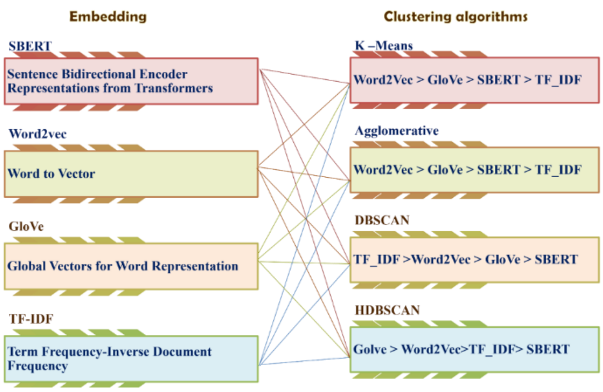


Fig. 7: Performance Ranking on Clustering Algorithms with Embeddings.

As shown in Fig. 7, GloVe is more effective than other embeddings when combined with HDBSCAN. This high performance is explained by the fact that GloVe can retrieve rich semantic relations between words. Unlike other embeddings, such as Word2Vec and TF-IDF, GloVe is explicitly designed to leverage global statistical data from a corpus, making it more effective at mapping words into a continuous space. This enables HDBSCAN to identify more informative clusters by grouping queries with similar semantic content, which is crucial in agricultural queries where context and nuance are key factors. The large Silhouette Score and small Davies-Bouldin Index in Fig. 7 illustrate how GloVe enhances the clustering process by increasing the distance between clusters and the closeness of data points within each Cluster.

There is a certain amount of redundancy in the presentation of similar performance metrics of the different clustering algorithms and embeddings in Tables 2-4. To simplify the results, we summarize these tables in a single table, emphasizing the most important findings, with a focus on key measures such as the Silhouette Score, Davies-Bouldin Index, and Calinski-Harabasz Index. This summary table enables the algorithm and embedding to be compared more concisely, allowing the reader to determine which combinations proved most effective.

The outliers in the farmer dataset are region or query-specific and not typical of the rest of the data points. These outliers may be associated with specific farming problems that do not occur uniformly across the entire dataset, such as infrequent crop diseases, unusual agricultural practices, or geographical issues unique to the farm sector. These questions could be based on niche issues that are very relevant to specific regions and are not used in other areas. Those outliers can be used to gain a deeper understanding of the diverse problems faced by farmers in various geographical areas, and knowledge of these outliers can be leveraged to refine the accuracy of assistance offered to farmers via farmer helplines.

DBSCAN and HDBSCAN methods are density-sensitive algorithms that can detect outliers as Noise and provide helpful ways to analyze these special queries. The algorithms can focus on the more frequent and repetitive queries by removing these outliers, which may result in improved clustering performance for most of the data. Nevertheless, by investigating these outliers individually, we can uncover patterns or new issues that are region-specific and not generalized in mainstream queries. The extra layer of analysis may provide a more in-depth understanding of the agricultural problems faced by specific farming communities, and ultimately lead to a refinement of assistance to farmers.

5. Practical Implications

The practical implications of the study's results for agricultural support systems are substantial. Through the application of clustering algorithms such as HDBSCAN, query grouping becomes more efficient, and the response time to the query posed by the Farmer is shortened. For instance, questions about specific crops, pest control, or climatic conditions can be aggregated so that they can be explicitly answered, and farmers can be provided with the pertinent guidance as soon as possible. Additionally, the clustering outcomes can inform the guidance of agricultural policy by highlighting similar regional concerns. This data could be utilized to allocate resources effectively, develop region-specific training programs for farmers, and enhance agrarian infrastructure. This can also improve the functionality of the helpline services by streamlining the query categorization process, which will contribute to lower operational costs and improved overall efficiency of the agricultural support system, while also enabling a more responsive and resource-efficient system.

6. Conclusion

This paper focuses on a comparative evaluation of the effectiveness of the various clustering algorithms, specifically K-means, Agglomerative, DBSCAN, and HDBSCAN, integrated with embeddings and assessed through Farmer's query. The study finds that HDBSCAN (excluding outliers) performs well on all types of embeddings. Word2Vec and GloVe deliver consistent performance. At the same time, SBERT and TF-IDF exhibit more variability depending on the clustering method, as indicated by the clustering quality, as measured by comparison of evaluation metrics. In the future, efficiency can be improved by optimizing execution time and incorporating deep learning models for embeddings. These enhancements will significantly boost the overall effectiveness of the clustering process.

Future Directions

Future studies might examine the use of transformer-based models, such as BERT or GPT, to create more realistic embeddings for use in clustering agricultural queries. These models have demonstrated encouraging performance in representing complex semantic relationships and may be better suited for clustering semantics than conventional algorithms, such as Word2Vec. Moreover, other agricultural datasets, such as weather data, soil health data, or crop yield prediction data, may be clustered, and their analysis may yield additional information about existing farming issues in the region, thereby enhancing decision-making in agricultural support systems. Other studies that can be used in future research include real-time Clustering in operational call centers. Query data in a call center is constantly changing, and the system must evolve dynamically to new topics.

References

[1] M. Ma, M. Liang, and Y. Ji, "Comparison and Evaluation of Clustering Algorithms," 2023 International Conference on Computers, Information Processing and Advanced Education (CIPAE), Ottawa, ON, Canada, 2023, pp. 213-219, https://doi.org/10.1109/CIPAE60493.2023.00047.

[2] Arockiam, A. J. M. S., and Elizabeth Shanthi Irudhayaraj. "Reclust: An efficient clustering algorithm for mixed data based on reclustering and cluster validation." Indonesia. J. Electr. Eng. Comput. Sci 29.1 (2023): 545-552. https://doi.org/10.11591/ijeecs.v29.i1.pp545-552.

[3] Azhir, Elham, et al. "Performance Evaluation of Query Plan Recommendation with Apache Hadoop and Apache Spark." OSF Preprints, 17 Sept. 2022. Web. https://doi.org/10.31219/osf.io/mgpr7.

[4] Zhang, C. (2021). Research on Literature Clustering Algorithm for Massive Scientific and Technical Literature Query Service. Computational Intelligence and Neuroscience, 2022(1), 3392489. https://doi.org/10.1155/2022/3392489.

[5] Mehta, V., Bawa, S., & Singh, J. WEClustering: word embeddings-based text clustering technique for large datasets. Complex Intell. Syst. 7, 3211–3224 (2021). https://doi.org/10.1007/s40747-021-00512-9.

[6] D. Pradeep, C. Sundar, QAOC: Novel query analysis and ontology-based Clustering for data management in Hadoop,Future Generation Computer Systems,Volume 108, 2020,Pages 849-860,ISSN 0167-739X, https://doi.org/10.1016/j.future.2020.03.010.

[7] Salih, Niyaz Mohammed, and Karwan Jacksi. "State of the art document clustering algorithms based on semantic similarity." Jurnal Informatika 14.2 (2020): 58-75. https://doi.org/10.26555/jifo.v14i2.a17513.

[8] D. Mahapatra, C. Maharana, S. P. Panda, J. P. Mohanty, A. Talib and A. Mangaraj, "A Fuzzy-Cluster based Semantic Information Retrieval System," 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2020, pp. 675-678, https://doi.org/10.1109/ICCMC48092.2020.ICCMC-000125.

[9] Kfir Bernstein, Fiana Raiber, Oren Kurland, and J. Shane Culpepper. 2020. Cluster-Based Document Retrieval with Multiple Queries. In Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval (ICTIR '20). Association for Computing Machinery, New York, NY, USA, 33–40. https://doi.org/10.1145/3409256.3409825

[10] Sikdar S, Mukherjee A, Marsili M. Unsupervised ranking of clustering algorithms by INFOMAX. PLoS One. 2020 Oct 26;15(10):e0239331. PMID: 33104709; PMCID: PMC7588117. https://doi.org/10.1371/journal.pone.0239331.

[11] Rodriguez MZ, Comin CH, Casanova D, Bruno OM, Amancio DR, Costa LDF, Rodrigues FA. Clustering algorithms: A comparative approach. PLoS One. 2019 Jan 15;14(1):e0210236. PMID: 30645617; PMCID: PMC6333366. https://doi.org/10.1371/journal.pone.0210236.

[12] Juhi Srivastava,Prof. Gayatri Pandi,Computer Engineering, L.J. Institute of Engineering and Technology, Gujarat, India. A Time-Efficient Clustering Algorithm for Query Optimization in Distributed Database© 2018 IJCRT | Volume 6, Issue 2 April 2018 | ISSN: 2320-2882

[13] Frédéric Ros, Rabia Riad, Serge Guillaume, "Deep Clustering Framework Review Using Multicriteria Evaluation," Knowledge-Based Systems, Volume 285, 2024, 111315, ISSN 0950-7051. https://doi.org/10.1016/j.knosys.2023.111315.

[14] K. R. Alla and G. Thangarasu, "Robust Text Clustering to Cluster the Text Documents in A Meta-Heuristic Optimization," 2023 IEEE 13th Symposium on Computer Applications & Industrial Electronics (ISCAIE), Penang, Malaysia, 2023, pp. 181-185. https://doi.org/10.1109/ISCAIE57739.2023.10165352.

[15] Faruque, O., Nji, F.N., Cham, M., Salvi, R.M., Zheng, X., Wang, J. (2023). Deep Spatiotemporal Clustering: A Temporal Clustering Approach for Multi-dimensional Climate Data. In: De Francisci Morales, G., Perlich, C., Ruchansky, N., Kourtellis, N., Baralis, E., Bonchi, F. (eds) Machine Learning and Knowledge Discovery in Databases: Applied Data Science and Demo Track. ECML PKDD 2023. Lecture Notes in Computer Science(), vol 14175. Springer, Cham. https://doi.org/10.1007/978-3-031-43430-3_6.

[16] M. Ma, M. Liang and Y. Ji, "Comparison and Evaluation of Clustering Algorithms," 2023 International Conference on Computers, Information Processing and Advanced Education (CIPAE), Ottawa, ON, Canada, 2023, pp. 213-219, https://doi.org/10.1109/CIPAE60493.2023.00047.

[17] Oyewole, Gbeminiyi John, and George Alex Thopil. "Data clustering: application and trends." Artificial Intelligence Review 56.7 (2023): 6439-6475. https://doi.org/10.1007/s10462-022-10325-y.

[18] Abiodun M. Ikotun, Absalom E. Ezugwu, Laith Abualigah, Belal Abuhaija, Jia Heming, K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data, Information Sciences, Volume 622,2023,Pages 178-210,ISSN 0020-0255, https://doi.org/10.1016/j.ins.2022.11.139.

[19] Kamal Taha,Semi-supervised and un-supervised Clustering: A review and experimental evaluation, Information Systems, Volume 114,2023,102178,ISSN 0306-4379. https://doi.org/10.1016/j.is.2023.102178.

[20] Oyewole, G.J., Thopil, G.A. Data clustering: application and trends. Artif Intell Rev 56, 6439–6475 (2023). https://doi.org/10.1007/s10462-022-10325-y.

[21] Kamal Taha,Semi-supervised and un-supervised Clustering: A review and experimental evaluation, Information Systems, Volume 114,2023. https://doi.org/10.1016/j.is.2023.102178.

[22] Shahid, N. Comparison of hierarchical Clustering and neural network clustering: an analysis on precision dominance. Sci Rep 13, 5661 (2023). https://doi.org/10.1038/s41598-023-32790-3.

[23] Unciano, N. (2025). AI-Augmented Metasurface-Aided THz Communication: A Comprehensive Survey and Future Research Directions. Electronics, Communications, and Computing Summit, 3(2), 1–9.

[24] Madhanraj. (2025). Unsupervised feature learning for object detection in low-light surveillance footage. National Journal of Signal and Image Processing, 1(1), 34–43.

[25] Surendar, A. (2025). Hybrid Renewable Energy Systems for Islanded Microgrids: A Multi-Criteria Optimization Approach. National Journal of Renewable Energy Systems and Innovation, 27-37.

[26] Rahim, R. (2025). Lightweight speaker identification framework using deep embeddings for real-time voice biometrics. National Journal of Speech and Audio Processing, 1(1), 15–21.

[27] Ramchurn, R. (2025). Advancing autonomous vehicle technology: Embedded systems prototyping and validation. SCCTS Journal of Embedded Systems Design and Applications, 2(2), 56–64.

[28] Vardhan, K. V., & Musala, S. (2024). Thermometer Coding-Based Application-Specific Efficient Mod Adder for Residue Number Systems. Journal of VLSI Circuits and Systems, 6(2), 122–129. https://doi.org/10.31838/jvcs/06.02.14.

[29] Rahim, R. (2024). Energy-Efficient Modulation Schemes for Low-Latency Wireless Sensor Networks in Industrial Environments. National Journal of RF Circuits and Wireless Systems, 1(1), 21–27.

[30] Uvarajan, K. P. (2025). Design of a hybrid renewable energy system for rural electrification using power electronics. National Journal of Electrical Electronics and Automation Technologies, 1(1), 24–32.