International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

HYBRID STOPWORD DETECTION FOR CLUSTERING TAMIL TEXT DATA

Mrs.S.Sujatha[1], Dr. Grasha Jacob[2]

Submitted: 02/09/2024 **Revised:** 10/10/2024 **Accepted:** 24/10/2024

[1]Research Scholar, Department of Computer Science, Rani Anna Government College for Women, Tirunelveli-627008, Tamil Nadu, India, Affiliated to Manonmaniam Sundaranar University, Tirunelveli,-mail: sanksuj@gmail.com

[2]Principal, Department of Computer Science,Government Arts & Science College, Ottapidaram-628401, Thoothukudi, Tamil Nadu, India, Affiliated to Manonmaniam Sundaranar University, Tirunelveli,E-mail: grasharanjit@gmail.com

ABSTRACT: The Traditional stopword removal techniques rely on static lists or individual statistical filters, which often fail to capture the contextual irrelevance of words across domains and languages. This research paper proposes a hybrid dynamic stopword detection framework that integrates three methods such as frequency analysis, TF-IDF scoring, and frequency-normalized centrality from word co-occurrence graphs. The frequency- normalized method adaptively filters contextually uninformative terms while preserving semantically rich content, significantly improving downstream clustering and topic modelling performance. The filtered text is embedded using feature fusion of Sentence-BER and TF-IDF vectors and, reduced via UMAP, and clustered using HDBSCAN, a density-based algorithm capable of identifying clusters of varying shapes and densities. The number of dynamic stopwords increased in the proposed hybrid method. Evaluations on Tamil text data demonstrate enhanced clustering quality measured by Silhouette score, DBI, and topic coherence, proving the method's effectiveness for morphological rich Tamil language.

Keywords: Text clustering, TF-IDF, Co-occurrence graph, Hybrid, dynamic stopword, HDSCAN algorithm.

1. INTRODUCTION

The mass growth in the volume of digital content in low-resource languages like Tamil the process of automatic text clustering has become vital for unifying and mining meaningful information. Generally, the clustering algorithms profoundly counts on the quality of text preprocessing, particularly in removing irrelevant or redundant words known as stopwords. The stopword removal process is well-studied in English but Tamil language faces unique challenges due to its rich morphology, lack of universal stopword lists, and the mixing of Tamil-English content in real-world datasets.

The morphological rich Tamil language faces very critical challenges in Tamil text clustering due to its affluence characters and the nonappearance of wide-ranging stopword lists. The traditional static or frequency-based stopword removal methods habitually miss the mark to capture contextually unproductive words which lead to noise in text representations.

The existing stopword filtering techniques such as static stopword, frequency-based filtering and TF-IDF filtering are effective to some level but these methods often filter out frequent but semantically important words and fails to remove contextually uncommunicative tokens. The Graph-based approaches which use word co-occurrence shows potential, but they tend to overstate high-frequency words that are structurally central in the co-occurrence graph which results in prejudiced selections.

The paper focusses these challenges by proposing a **frequency-normalized centrality-based stopword detection method** which constructs a word co-occurrence graph and ranks words based on their structural importance adjusted by token frequency. This method confirms that a word's importance in the graph is not merely due to repetition but due to meaningful contextual usage.

This paper is organized into six sections such as Section II is about related work, Section III is about the Frequency-Normalized Degree Centrality Stopword Detection Method, Section IV is about the proposed hybrid method of dynamic stopword detection with feature fusion HDBSCAN algorithm. Section V is about Experimental Analysis and Section VI is Conclusion.

2. RELATED WORK

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a popular density-based clustering algorithm that identifies clusters by locating core objects—data points that reside in dense neighborhoods—and expanding clusters from them [3]. A core object is defined as one that has at least a minimum number of neighboring points (MinPts) within a specified radius (ε). When such an object is found, the algorithm

International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

explores its neighborhood and recursively includes other core or directly reachable points, thereby forming dense regions that constitute clusters. Objects that do not meet the density criteria—i.e., they are not reachable from any core object and do not have sufficient neighbors—are labeled as noise or outliers. Unlike partitioning algorithms like K-Means, DBSCAN does not require the number of clusters to be specified in advance, and it can discover clusters of arbitrary shapes and varying sizes, making it highly effective for real-world, high-dimensional, or spatial datasets [11]. Liu et al. 2007 have modified the DBSCAN to deal with the datasets that are varied in densities. Their algorithm is called VDBSCAN. VDBSCAN is able to calculate the density threshold parameters automatically based on the Kdistance plotting [8].

According to Tiwari, Raguvanshi, and Jain ,2016, DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is not only capable of detecting clusters of arbitrary shape but also effectively identifies noise and outliers within a dataset. The authors emphasize that the algorithm performs efficiently and scales almost linearly with the size of the database, making it suitable for large datasets. DBSCAN utilizes the density distribution of data points to classify them into meaningful clusters. Points located in dense regions are grouped together as part of a cluster, while those in sparse regions are labeled as noise. This density-based approach allows the algorithm to classify nodes into distinct groups, each potentially representing a different class or concept within the data [12].

Hussain Ahmed Chowdhury et al., 2021 proposed UIFDBC (User-Input-Free Density-Based Clustering), a novel density-based algorithm designed to extract clusters of arbitrary shapes without requiring user-defined parameters (Chowdhury, Bhattacharyya, & Kalita, 2021). The method begins by computing the local density of each data point based on the size of its nearest-neighborhood, enabling dynamic adaptation to varying data distributions. Clusters are formed in a two-phase process: first, sub-clusters are generated to group dense regions, minimizing error propagation; second, these sub-clusters are merged or refined to effectively handle noisy or low-density instances. This two-step strategy enhances robustness and reduces sensitivity to parameter tuning when compared to traditional density-based methods. The DBSCAN algorithm can identify clusters of large spatial data sets watching the local density of blocks of data using a single input parameter. In addition, the user gets a suggestion that the parameter value which would be appropriate. Therefore, a minimum area of knowledge is required [5].

HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise), introduced by Campello, Moulavi, and Sander (2013) and later optimized by McInnes and Healy (2017), is an advanced density-based clustering algorithm that extends DBSCAN by allowing for the detection of clusters with varying densities. Unlike DBSCAN, which requires a global ε (epsilon) parameter to define neighborhood size, HDBSCAN eliminates the need for a fixed ε by building a hierarchical cluster tree based on varying density levels. This hierarchy is then condensed and analyzed to extract the most stable clusters, thereby improving robustness to noise and parameter sensitivity. HDBSCAN is particularly well-suited for complex datasets where clusters may have non-uniform densities, and it is known for its ability to provide a soft clustering output, assigning a probability of cluster membership for each point. These features make it highly effective for real-world applications in text, image, and anomaly detection domains [3],[9].

McInnes and Healy (2017) introduced Accelerated HDBSCAN*, a performance-optimized version of the original HDBSCAN algorithm that achieves computational speeds comparable to DBSCAN while retaining the ability to automatically detect clusters of varying density. This enhancement makes Accelerated HDBSCAN* not only scalable to large datasets but also highly effective in discovering complex cluster structures without requiring a global distance threshold, establishing it as a standard choice in modern density-based clustering applications [9].

Asyaky and Mandala (2021) addressed the challenge of short-text sparsity, such as that found in tweets, by integrating BERT and FastText embeddings with UMAP for dimensionality reduction and HDBSCAN for clustering. Their experimental results demonstrated superior performance in terms of purity and Normalized Mutual Information (NMI) when compared to traditional baseline clustering approaches, highlighting the effectiveness of semantic embeddings and density-based clustering for sparse, high-dimensional text data [1].

NVIDIA's RAPIDS implementation of HDBSCAN,2022, when integrated with Sentence-BERT embeddings and UMAP for dimensionality reduction, demonstrated the capability to efficiently process millions of news headlines. The use of GPU acceleration significantly enhanced scalability, while the algorithm's soft clustering outputs enabled a more nuanced analysis of overlapping or ambiguous topics, which is especially valuable in dynamic, high-volume text streams like news media [2].

Bot et al. ,2023 introduced FLASC, an enhancement of HDBSCAN* that detects flare sub-structures within clusters. By applying a post-processing step, FLASC identifies internal branches—revealing richer hierarchical subpopulations that standard HDBSCAN might overlook. Despite this added granularity, FLASC maintains similar computational complexity to HDBSCAN*, and shows robust performance on both synthetic and real-world datasets [3].

Sao, Prokopenko & Lebrun-Grandié ,2024 developed PANDORA, a novel parallel algorithm for constructing dendrograms in single-linkage clustering—including HDBSCAN—on GPUs. Utilizing recursive tree contraction, the approach is fully parallel and asymptotically optimal, dramatically reducing performance bottlenecks. It achieves a 2.2× speed-up terrestrially and 10–37× GPU acceleration, resulting in up to 6× overall HDBSCAN speed-up on large datasets [10].

International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Ghosh, Naldi, & Sander ,2024 proposed enhancements within HDBSCAN* by leveraging GLOSH (Global-Local Outlier Scores). They introduced an unsupervised strategy to automatically select the optimal minPts parameter and outlier threshold—enhancing HDBSCAN's ability to flag anomalies without manual tuning [7].

3. FREQUENCY-NORMALIZED DEGREE CENTRALITY STOPWORD DETECTION METHOD

The stopword removal process is an initial step in text preprocessing and it has to be effective particularly for low-resource and morphologically rich languages such as Tamil. To address the limitation in the existing methods a new graph-theoretic approach for dynamic stopword detection using Frequency-Normalized Degree Centrality has been proposed.

The proposed method of stopword detection combines word co-occurrence structure with term frequency statistics to identify context-insensitive words. A word is considered a potential stopword if it occurs frequently but is not structurally central within the document corpus**.**

The following are the steps which is used to identify the dynamic stopwords through frequency-normalized degree centrality is as follows,

Step 1: Co-occurrence Graph Construction

Each document from the Tamil text column is tokenized using a custom Tamil tokenizer Indic_NLP. In co-occurrence graph construction the sliding window size plays a vital role. A sliding window size refers to number of consecutive words that is considered at a time to detect co-occurrence relationships. The **window size** is the number of words in a segment of text where all possible **word pairs (combinations)** are considered **linked** (connected by edges in a graph). In a co-occurrence graph the nodes are the words and the edges are the words that appear within the window size.

In this method a **sliding window** of fixed size (typically 3) is applied to each document to capture **local word co-occurrence**.

- An **undirected weighted graph** is built:
  - **Nodes**: Unique Tamil words (excluding static stopwords)
  - **Edges**: Pairwise word co-occurrences within the window
  - **Edge Weight**: Frequency of co-occurrence.

For example,

பட்ஜெட் விலையில் மோட்டோ ஸ்மார்ட்போன் இந்தியாவில் அறிமுகம்

After tokenization

[‘பட்ஜெட்’ ‘விலையில்’ ‘மோட்டோ’ ‘ஸ்மார்ட்போன்’ ‘இந்தியாவில்’ ‘அறிமுகம்’];

If the window size is 3 then every group of 3 consecutive words need to be considered and all pairs of words within that group has to be connected.

Window 1: பட்ஜெட் விலையில் மோட்டோ

All the two-word combinations are

பட்ஜெட்- விலையில்
பட்ஜெட்- மோட்டோ
விலையில்- மோட்டோ

Window 2 : விலையில் மோட்டோ ஸ்மார்ட்போன்

All the two-word combinations are

விலையில்- மோட்டோ (Already counted in window 1)
விலையில-ஸ்மார்ட்போன்
மோட்டோ- ஸ்மார்ட்போன்

Window 3 : மோட்டோ ஸ்மார்ட்போன் இந்தியாவில்

All the two-word combinations are

மோட்டோ- ஸ்மார்ட்போன்(Already counted in window size 2)
மோட்டோ- இந்தியாவில்
ஸ்மார்ட்போன்-இந்தியாவில்

Window 4 : ஸ்மார்ட்போன் இந்தியாவில் அறிமுகம்

All the two-word combinations are

ஸ்மார்ட்போன்-இந்தியாவில்(Already counted in window size 3)
ஸ்மார்ட்போன-அறிமுகம்
இந்தியாவில்-அறிமுகம்

The final unique co-occurring pairs are,

Table 3.1 Co-Occurring pairs of words for sample statement.

Word 1	Word 2
பட்ஜெட்	விலையில்
பட்ஜெட்	மோட்டோ
விலையில்	மோட்டோ
விலையில்	ஸ்மார்ட்போன்
மோட்டோ	ஸ்மார்ட்போன்
மோட்டோ	இந்தியாவில

International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

ஸ்மார்ட்போன்	இந்தியாவில
ஸ்மார்ட்போன்	அறிமுகம்
இந்தியாவில்	அறிமுகம்

For the co-occurrence graph the total number of edges formed are 9.

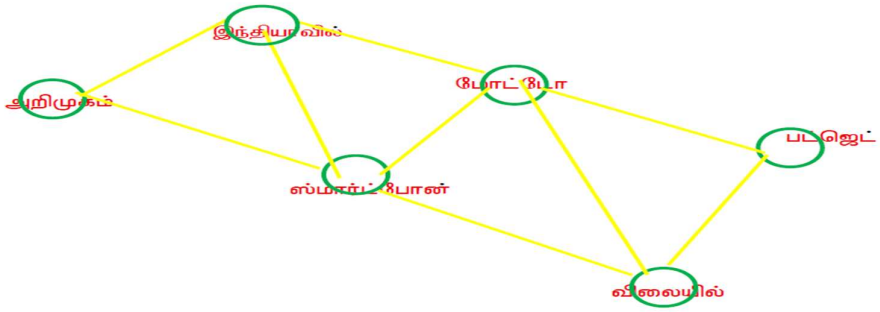

Fig 3.1 Co-Occurrence Graph with window size = 3

The words like மோட்டோ and ஸ்மார்ட்போன் are strongly connected and which means they are contextually relevant. This structure helps to identify which words are central and which are loosely linked.

Step 2: Compute Degree Centrality

For each word i.e node the Degree Centrality is the number of connections it has with other nodes. Degree of a word is the number of other words that co-occurs directly with it.

Degree Centrality measures how structurally important a word is, based on how many other words it frequently co-occurs with.

For example, the word பட்ஜெட் (node 1) is connected to the edge 2(விலையில்) and edge 3(மோட்டோ) so the degree is 2. Similarly, the degree for all the nodes are tabulated below,

Table 3.2 Degree of words for Sample Statement

Node	Word	Connected to Edges	Degree
1	பட்ஜெட்	விலையில் மோட்டோ	2
2	விலையில்	பட்ஜெட் மோட்டோ ஸ்மார்ட்போன்	3
3	மோட்டோ	பட்ஜெட் விலையில் மோட்டோ ஸ்மார்ட்போன்	4
4	ஸ்மார்ட்போன்	விலையில் மோட்டோ ஸ்மார்ட்போன் அறிமுகம்	4
5	இந்தியாவில்	மோட்டோ ஸ்மார்ட்போன் அறிமுகம்	3
6	அறிமுகம்	ஸ்மார்ட்போன் இந்தியாவில்	2

- For each node w in the graph, compute **degree centrality**:

$$DegreeCentrality(w) = \frac{Number\ of\ direct\ connections\ of\ w}{N-1} \quad 1$$

where N is the total number of nodes.

The Degree Centrality for each node is tabulated below,

Table 3.3 Degree Centrality for Sample Statements

Node	Word	Connected to Edges	Degree	DegreeCentrality = Degree/N-1 N=6 N-1 =5
1	பட்ஜெட்	விலையில் மோட்டோ	2	0.4
2	விலையில்	பட்ஜெட் மோட்டோ ஸ்மார்ட்போன்	3	0.6
3	மோட்டோ	பட்ஜெட் விலையில் மோட்டோ ஸ்மார்ட்போன்	4	0.8
4	ஸ்மார்ட்போன்	விலையில் மோட்டோ ஸ்மார்ட்போன் அறிமுகம்	4	0.8
5	இந்தியாவில்	மோட்டோ ஸ்மார்ட்போன் அறிமுகம்	3	0.6
6	அறிமுகம்	ஸ்மார்ட்போன் இந்தியாவில்	2	0.4

International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

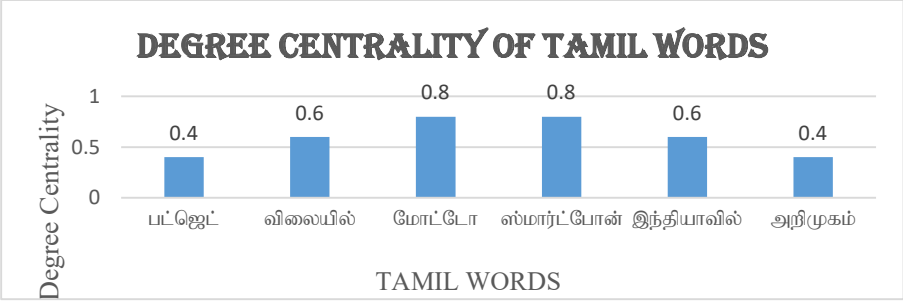

Fig 3.2 Degree Centrality of Tamil Words

Step 3: Compute Token Frequency

The raw frequency of each token w in the entire corpus is calculated using a Counter. The token frequency is the number of times the word occurs in the entire dataset.

Step 4: Frequency-Normalized Centrality

For each token w, compute its **normalized centrality** as:

$$NormalizedCentrality(w) = \frac{DegreeCentrality(w)}{Frequency(w)+1} \quad 2$$

The addition of 1 in the denominator avoids division by zero and smooths the score for very low-frequency words. For the example statement the Frequency-Normalized Centrality is tabulated below,

Table 2.4 Frequency Normalized Centrality for sample statement.

Node	Word	Frequency	Degree Centrality	Frequency Normalized Centrality (DC/F+1)
1	பட்ஜெட்	1	0.4	0.2
2	விலையில்	1	0.6	0.3
3	மோட்டோ	1	0.8	0.4
4	ஸ்மார்ட்போன்	1	0.8	0.4
5	இந்தியாவில்	1	0.6	0.3
6	அறிமுகம்	1	0.4	0.2

Step 5: Dynamic Stopword Selection

For dynamic stopword selection all tokens are sorted by their normalized centrality scores in descending order. The top 5% of tokens with the highest normalized centrality are selected as graph-based dynamic stopwords.

The above steps will be continued for complete dataset and identifies the total number of stopwords dynamically. The frequency-normalized degree centrality methods outperforms well than the existing graph based stopword detection.

4. PROPOSED HYBRID STOPWORD DETECTION AND FEATURE FUSION HDBSCAN ALGORITHM(HSDHDBSCAN)

The traditional graph-based centrality methods namely the degree centrality has certain limitations when applied to text data, especially for identifying stopwords in morphologically rich or low-resource languages like Tamil. The major issue is their bias toward high-frequency words. These words tend to co-occur with many others and thus achieve high centrality scores, even if they carry little semantic value. As a result, common function words like conjunctions and pronouns may dominate the centrality ranking, despite contributing minimally to document meaning. This can be rectified by the implementation of the Frequency-normalized degree centrality method.

4.1 HYBRID STOPWORD DETECTION AND FEATURE FUSION HDBSCAN ALGORITHM(HSDHDBSCAN)

Input: Tamil Dataset with Tamil Columns
Output: Dynamic Stopwords, Metric Values
Step 1. LOAD Dataset
    Read 'tamil' column → documents_raw
    Remove NaNs and convert to string
Step 2. CLEAN Text
  For each document in documents_raw:
    Remove non-Tamil characters using regex
  Store cleaned documents in documents_tamil
Step 3. TOKENIZE Tamil Text
  For each document in documents_tamil:
    Extract Tamil words (min 2 letters) using Unicode pattern
Step 4. FREQUENCY-Based Stopwords

International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

```
  Count frequency of all tokens
  Select top 5–10% most frequent tokens → freq_stopwords
Step 5. TF-IDF-Based Stopwords
  Compute TF-IDF matrix for documents
  Compute average TF-IDF score for each word
  Select words with TF-IDF below 40th percentile → tfidf_stopwords
Step 6. GRAPH-Based Frequency-Normalized Centrality
  Initialize empty co-occurrence graph G
  For each document in documents_tamil:
    Remove static stopwords from tokens
    For each sliding window of size 3:
      Connect all word pairs with edge in G (or increment weight)
  Compute degree centrality for each node in G
  Count token frequency for nodes in G
  For each token:
    normalized_score = degree_centrality / (frequency + 1)
  Select top 5% tokens by normalized_score → graph_stopwords
Step 7. FINAL STOPWORDS
  final_stopwords = freq_stopwords ∪ tfidf_stopwords ∪ graph_stopwords
Step 8. PREPROCESS Documents
  For each document:
    Tokenize and remove all final_stopwords
    Join back to clean sentences → preprocessed_docs
Step 9. EMBEDDING Generation
  Use SBERT to encode preprocessed_docs → sbert_embeddings
  Normalize sbert_embeddings
  Compute TF-IDF vectors → tfidf_vectors
  Concatenate SBERT + TF-IDF → combined_vectors
  Normalize combined_vectors with StandardScaler
Step 10. DIMENSIONALITY Reduction (UMAP)
  Apply UMAP on combined_vectors → umap_reduced (10D)
Step 11. CLUSTERING with HDBSCAN
  Initialize best silhouette score = -1
  FOR min_cluster_size = 3 to 10 DO:
    - Apply HDBSCAN with current min_cluster_size
    - Filter out noise points (label = -1)
    - IF more than 1 cluster found:
      - Compute silhouette score, DBI, CH score
      - IF silhouette > best_score:
        - Save current clusterer, labels, and reduced embeddings
    ELSE:
      - Skip evaluation (too few valid clusters)
  END FOR
Step 12.
        Print the metric values
```

The HSDHDBSCAN Tamil text clustering method is designed to dynamically detect stopwords and the documents are clustered using a hybrid feature approach. First the Tamil dataset is loaded and it reads the tamil text column which is needed for processing. As a next step the missing values are removed and it is ensured that the text is in string format. After this the documents undergo a cleaning process where non-Tamil characters are removed using Unicode-based regular expressions. The cleaned text is then tokenized to extract valid Tamil words that are at least two characters long.

The dynamic stopwords are identified by three strategies. First the dynamic stopwords are identified by using a frequency-based approach in which the occurrence of each token are accounted across all documents and selects the top 5–10% most frequent ones as potential stopwords.

The second method applies a TF-IDF-based method by computing the TF-IDF matrix of the corpus, determining the average score for each word, and selecting those below the 40th percentile as low-information terms.

The third method is the novel method in which the dynamic stopwords are identified by a co-occurrence graph from the documents. In the co-occurrence graph edges are created between words appearing within a sliding window of three tokens. First it calculates degree centrality for each word in the graph and normalizes this by token frequency.After obtaining the normalized frequency selects the selecting the top 5% of words with the highest normalized centrality as graph-based stopwords.

International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

The dynamic stopwords obtained from the three methods such as frequency-based, TF-IDF-based, and graph-based are then combined to form the final dynamic stopword list.

The ultimate intention of the proposed method is the implementation of the hybrid method of dynamic stopword detection. This hybrid method is essential because it accurately identifies and disregards non-informative words in Tamil text, which is a morphologically rich and low-resource language. Each method brings a inimitable perspective in identifying unimportant tokens, and their integration creates a more reliable and language-adaptive stopword list. The frequency-based method is used to identify the most commonly appearing words across the corpus. In this method the high-frequency terms often represent function words like particles, conjunctions, or auxiliary verbs in Tamil that carry little semantic value but appear repeatedly. Relying solely on frequency may erroneously remove frequent content-bearing words that are contextually important in some documents. The TF-IDF-based method focusses on evaluating the importance of a word to a document relative to the entire corpus. The words that appear frequently in many documents incline to have low TF-IDF scores and these are likely to be generic or irrelevant. Hence, low TF-IDF terms are good candidates for stopwords. TF-IDF alone might overlook structurally irrelevant terms that are contextually dominant.The graph-based centrality method, particularly frequency-normalized degree centrality, captures the structural importance of words in a co-occurrence network. Words that appear frequently with many others but do not form strong semantic hubs often have high centrality but low uniqueness. Normalizing centrality by frequency ensures that preference is given to structurally dominant yet semantically redundant tokens. This approach is particularly powerful for Tamil, where many suffix-based variants can form misleading co-occurrence patterns.

Each document is tokenized again and filtered to remove all identified stopwords in the preprocessing stage.The preprocessed document set is obtained by rejoining the remaining words into clean sentences. The Sentence-BERT (SBERT) embedding technique is used to generate semantic embeddings for the preprocessed documents. For generating SBERT technique TF-IDF vectors are also computed. These two types of feature vectors are concatenated to form a hybrid representation and it is normalized using StandardScaler.

UMAP dimensionality reduction technique is applied to reduce dimensionality by preserving the structure and the hybrid vectors are passed through UMAP to obtain a 10-dimensional representation. The reduced data is clustered using HDBSCAN for various values of min_cluster_size ranging from 3 to 10 repeatedly. The noise points are removed for each result and clustering quality is evaluated using three metrics such as Silhouette Score, Davies-Bouldin Index (DBI), and Calinski-Harabasz (CH) Score. The result yielding the highest Silhouette Score among the valid runs is chosen as the best. Finally, the metric values corresponding to the best clustering configuration are printed, providing an objective measure of clustering quality after dynamic stopword removal and hybrid feature embedding.

The HSDHDSCAN hybrid method ensures wide-ranging stopword detection by improving the quality of feature extraction, reducing noise, and enhancing clustering accuracy. This method influences statistical, semantic, and structural insights to address the limitations of any single method, making it highly suitable for Tamil text mining.

5. EXPERIMENTAL RESULTS AND ANALYSIS

To experiment the proposed method two datasets namely Tamil_News_Test and Tamil_News_Headlines were used. The eventual aspire of the proposed method is to show the increase in the number of dynamic stopwords by the use of hybrid method.

The identification of dynamic stopwords in plays a vital role in Tamil text clustering. It enhances clustering quality by eliminating contextually insignificant terms. On analyzing the number of dynamic stopwords identified through three distinct approache such as TF-IDF-based, frequency-based, and the proposed hybrid method,a notable progressive increase is observed in the count of dynamic stopwords.

The following table shows the increase in the number of dynamic words from the first to hybrid method.

Table 5.1 Dynamic Stopwords count

Data Set	No. of Dynamic stopwords obtained from TF-IDF Method	No. of Dynamic stop words obtained from Frequency Based Method	No. of Dynamic stop words obtained from Hybrid Method (Frequency +TF-IDF+ Frequency-Normalized Degree Centrality)
EngTamText	242	305	502
TamilNewsTest	419	886	1209
TamilNewsHeadlines	498	1693	1893

The hybrid method exhibits a clear increase in the number of dynamic stopwords compared to traditional TF-IDF and novel method frequency-based approaches. The TF-IDF method identifies globally inconsequential terms and frequency-based methods ensigns the repetitive words. The hybrid approach with frequency-normalized degree centrality synergizes both along with contextual graph-based insights. This hybrid method removes the more

International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

comprehensive and contextually irrelevant terms which enhances the clustering algorithm's ability to focus on semantically rich features.

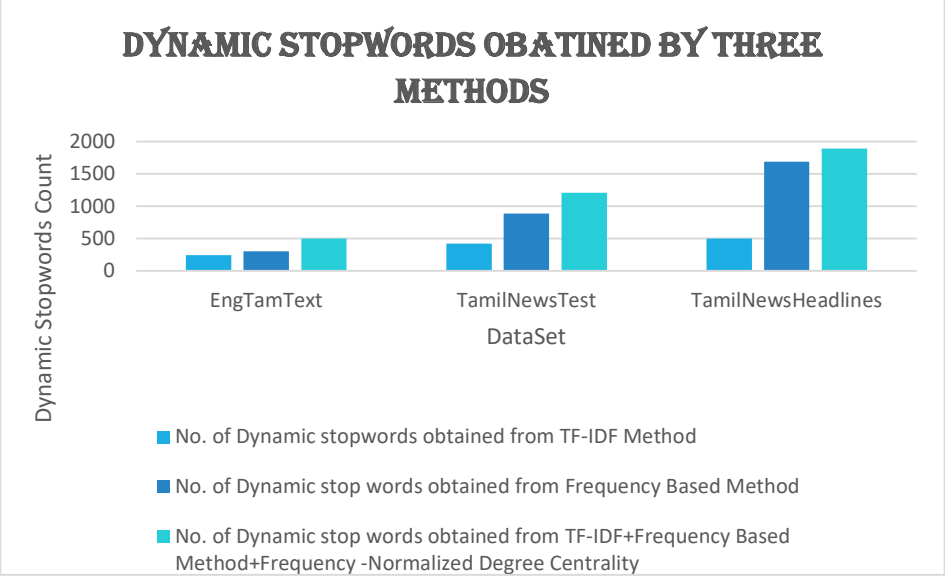

Fig 5.1 Dynamic Stopwords Count

The HDBSCAN clustering algorithm was employed in this study because of its ability to identify variable-density clusters without requiring a predefined number of clusters. The main key parameter in HDBSCAN is min_cluster_size. This parameter determines the minimum number of data points required to form a dense and valid cluster. In this study the min_cluster_size systematically varies from 3 to 10 which helps to assess the clustering granularity and helps to discover the natural grouping structure within the Tamil text dataset.

The impetus behind varying this parameter lies in its direct influence on the scale and resolution of discovered clusters. Generally, smaller values tend to detect finer subgroups, potentially capturing nuanced semantic themes, while larger values favour broader, more generalizable groupings. To avoid both under-segmentation (merging distinct topics) and over-segmentation (splitting cohesive topics) this variation in min_cluster_size is used.

The hybrid document representation which is formed by merging Sentence-BERT embeddings with TF-IDF vectors and this is reduced UMAP dimensionality reduction, exhibits a strong and consistent density structure. The following results of the datasets implied this.

The proposed HSDHDBSCAN algorithm has implemented in three datasets of varied min_cluster_size(3 to 10). In the TamilNewsHeadlines dataset the number of clusters for min_cluster_size is 3 and for all the other sizes the number of clusters is two. For TamilNewsTest and EnglishTamilText dataset the number of clusters formed is two for all min_cluster_sizes.

It is observed that in the existing the metric values are not up to the level because the method produces a greater number of clusters and unable to find the proper dense regions for a dataset.

Table 5.2 Performance Metrics for Tamil Hews HeadLines dataset

Cluster Size	Metric	Existing HDBSCAN with static stop words	Proposed Hybrid HDBSCAN with hybrid dynamic stopwords and feature fusion
3	Silhouette Score	0.5035	0.7939
	Davies Bouldin Index	0.6050	0.3353
	Calinski Harabasz Score	1507.6217	1436.4355
4	Silhouette Score	0.5214	0.8004
	Davies Bouldin Index	0.5773	0.1613
	Calinski Harabasz Score	1753.8457	2873.1842
5	Silhouette Score	*0.5331*	0.8004
	Davies Bouldin Index	*0.5744*	0.1613

International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

	Calinski Harabasz Score	*1869.8290*	2873.1842
6	Silhouette Score	0.5207	0.8004
	Davies Bouldin Index	0.5811	0.1613
	Calinski Harabasz Score	1942.4662	2873.1842
7	Silhouette Score	0.4218	0.8004
	Davies Bouldin Index	0.6396	0.1613
	Calinski Harabasz Score	1005.1356	2873.1842
8	Silhouette Score	0.3040	0.8004
	Davies Bouldin Index	0.6945	0.1613
	Calinski Harabasz Score	751.5328	2873.1842
9	Silhouette Score	0.3294	0.8004
	Davies Bouldin Index	0.6699	0.1613
	Calinski Harabasz Score	908.9087	2873.1842
10	Silhouette Score	0.3347	0.8004
	Davies Bouldin Index	0.6653	0.1613
	Calinski Harabasz Score	964.1953	2873.1842

Table 5.3 Performance Metrics for Tamil News Test Dataset

Min Cluster Size	Metric	Existing HDBSCAN with static stop words	Proposed Hybrid HDBSCAN with hybrid dynamic stopwords and feature fusion
3	Silhouette Score	*0.5339*	0.8099
	Davies Bouldin Index	*0.5678*	0.1482
	Calinski Harabasz Score	*548.7995*	1006.87
4	Silhouette Score	0.4698	0.8099
	Davies Bouldin Index	0.6246	0.1482
	Calinski Harabasz Score	328.3364	1006.87
5	Silhouette Score	0.4090	0.8099
	Davies Bouldin Index	0.5819	0.1482
	Calinski Harabasz Score	152.8560	1006.87
6	Silhouette Score	0.2126	0.8099
	Davies Bouldin Index	0.6960	0.1482
	Calinski Harabasz Score	128.0621	1006.87
7	Silhouette Score	0.1617	0.8099
	Davies Bouldin Index	0.8221	0.1482
	Calinski Harabasz Score	149.2523	1006.87
8	Silhouette Score	0.1874	0.8099
	Davies Bouldin Index	0.7850	0.1482
	Calinski Harabasz Score	167.7845	1006.87
9	Silhouette Score	0.1817	0.8099
	Davies Bouldin Index	0.7719	0.1482
	Calinski Harabasz Score	196.7827	1006.87
10	Silhouette Score	0.2950	0.8099
	Davies Bouldin Index	0.7185	0.1482
	Calinski Harabasz Score	234.2658	1006.87

Table 5.4 Performance Metrics for English Tamil Text Dataset

Min Cluster Size	Metric	Existing HDBSCAN with static stop words	Proposed Hybrid HDBSCAN with hybrid dynamic stopwords and feature fusion
3	Silhouette Score	*0.6822*	0.8155
	Davies Bouldin Index	*0.4087*	0.1468

International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

	Calinski Harabasz Score	*956.0528*	5216.2451
4	Silhouette Score	*0.6822*	0.8155
	Davies Bouldin Index	*0.4087*	0.1468
	Calinski Harabasz Score	*956.0528*	5216.2451
5	Silhouette Score	0.5886	0.8155
	Davies Bouldin Index	0.3851	0.1468
	Calinski Harabasz Score	531.6383	5216.2451
6	Silhouette Score	0.5900	0.8155
	Davies Bouldin Index	0.3903	0.1468
	Calinski Harabasz Score	531.5320	5216.2451
7	Silhouette Score	0.5900	0.8155
	Davies Bouldin Index	0.3903	0.1468
	Calinski Harabasz Score	531.5320	5216.2451
8	Silhouette Score	0.5929	0.8155
	Davies Bouldin Index	0.3751	0.1468
	Calinski Harabasz Score	523.6919	5216.2451
9	Silhouette Score	0.5900	0.8155
	Davies Bouldin Index	0.3903	0.1468
	Calinski Harabasz Score	531.5320	5216.2451
10	Silhouette Score	0.5900	0.8155
	Davies Bouldin Index	0.3903	0.1468
	Calinski Harabasz Score	531.5320	5216.2451

In this method the min_cluster_size parameter in the HDBSCAN algorithm was systematically executed for varied sizes from 3 to 10. This is used to assess the impact on clustering granularity and stability. In the both the TamilNewsTest and EnglishTamilText the algorithm consistently identified exactly **two valid clusters** across all configurations. The outcome of all the datasets strongly implies that the underlying data contains two semantically dense and well-separated groups, which are robust to parameter changes. The outcome confirms the effectiveness of the hybrid feature fusion and dynamic stopword detection and elimination strategy.

The consistency in clustering has been validated through the clustering quality metrics. The silhouette score of above 0.8 implies that the clusters are highly cohesive and well-separated. The stable Davis Bouldin Index and Calinski-Harabasz score confirms the strength and stability of the clusters.

It is very clear from the results that the varying min_cluster_size parameter which is intended to adjust the granularity of density-based clustering does not uncover additional stable clusters because it has no other dense regions existing in the UMAP-reduced space. Instead, the two discovered clusters consistently satisfy all clustering conditions and yield the best possible quality scores. Therefore, this result empirically supports that **the two-cluster solution is not only valid but also optimal** for this dataset.

The following tables list the best silhouette, Davies Bouldin Index and Calinski-Harabasz values of all the three datasets.

Best Silhouette Scores of all the dataset

Table 5.5 Best Silhouette Score

S.No	Dataset	Existing HDBSCAN	Proposed Hybrid Method
1	EnglishTamilText	0.6822	0.8155
2	TamilNewsTest	0.5339	0.8099
3	TamilNewsHeadlines	*0.5331*	0.8004

International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

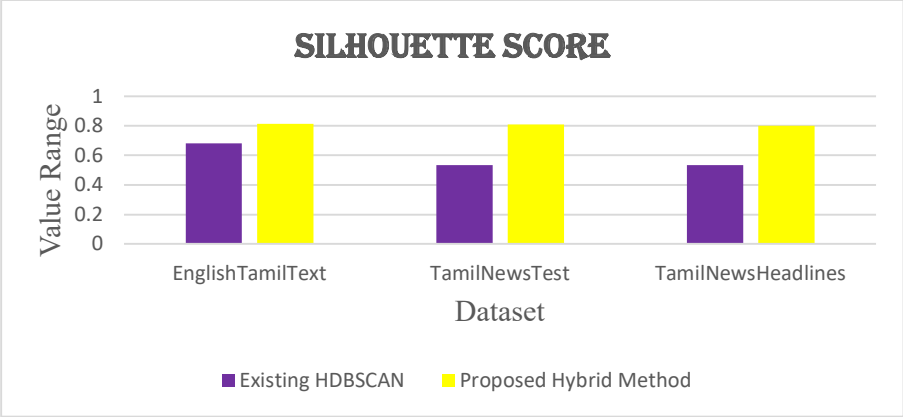

Fig 5.2 Best Silhouette Score of all the Dataset

Best Davies Bouldin Index

Table 5.6 Best Davies Bouldin Index

S.No	Dataset	Existing HDBSCAN	Proposed Hybrid Method
1	EnglishTamilText	0.4087	0.1468
2	TamilNewsTest	0.5678	0.1482
3	TamilNewsHeadlines	0.5744	0.1613

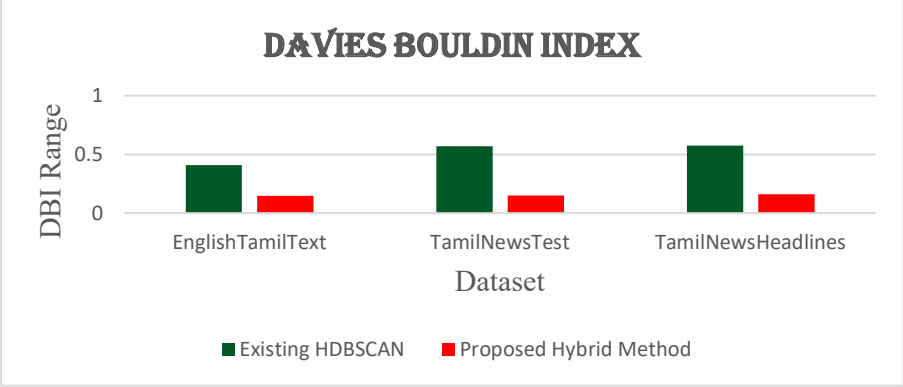

Fig 5.3 Best Davies Bouldin Index of all the dataset

Best Calinski- Harabasz Score

Table 5.7 Best Calinski-Harabasz Score

S.No	Dataset	Existing HDBSCAN	Proposed Hybrid Method
1	EnglishTamilText	956.0528	5216.2451
2	TamilNewsTest	548.7995	1006.87
3	TamilNewsHeadlines	1869.8290	2873.1842

International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

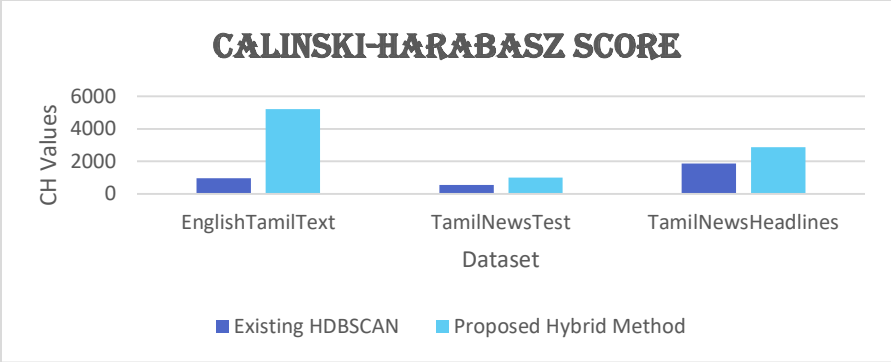

Fig 5.4 Best Calinski-Harabasz Score of all the Dataset

6. CONCLUSION

The proposed HSDHDBSCAN algorithm combines hybrid dynamic stopword detection with feature fusion to deliver a robust clustering solution for complex and low-resource languages such as Tamil. In this method, the graph-based normalized degree centrality for adaptive stopword identification is integrated with frequency-based stopword detection and TF_IDF methods to yield effective dynamic stopwords., The framework boosts the quality of input and cluster separability by merging syntactic and semantic features. Dimensionality reduction is done by using UMAP which further enhances structure preservation, resulting in more coherent clusters. Evaluation results show that this approach consistently outperforms traditional methods in terms of cluster quality metrics, while its adaptability makes it effective for multilingual and code-mixed datasets.

REFERENCES

1. Asyaky M S & Mandala R,2021, “Improving the performance of HDBSCAN on short text clustering by using word embedding and UMAP”, ICAICTA, IEEE, doi:10.1109/ICAICTA53211.2021.9640285.
2. Becker N & Nolet C,2022, “Faster HDBSCAN soft clustering with RAPIDS cuML”, NVIDIA Developer Blog.
3. Bot A. A, Semedo G. D, Zaidi N & Cameron J, 2023, “FLASC: A flare-sensitive HDBSCAN post-processing routine”, arXiv preprint arXiv:2311.15887,doi.org/10.48550/arXiv.2311.15887.
4. Campello R J G B, Moulavi D & Sander J, 2013, “Density-based clustering based on hierarchical density estimates”, P-1 KDD conferences, Vol(7819), Part 2, pp.160–172, doi.org/10.1007/978-3-642-37456-2_14.
5. Chowdhury, H. A., Bhattacharyya, D., & Kalita, J,2021, “UIFDBC: User-input-free density-based clustering”, Knowledge-Based Systems, 214, 106741. doi.org/10.1016/j.knosys.2020.106741.
6. Ester M, Kriegel H P, Sander J & X Xu, 1996, "A density-based algorithm for discovering clusters in large spatial databases with noise," in Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD), Portland, OR, USA, pp. 226–231.
7. Ghosh A, Naldi M C & Sander J, 2024, “GLOSH: Global-local outlier scores for HDBSCAN”, In Proceedings of the 27th International Conference on Extending Database Technology (EDBT 2024), Paestum, Italy, OpenProceedings.org. https://doi.org/10.48786/edbt.2024.17.
8. Liu P, Zhou D & Wu N J, 2007, “VDBSCAN: Varied Density Based Spatial Clustering of Applications with Noise,” in proceedings of IEEE International Conference on Service Systems and Service Management, Chengdu, China.
9. McInnes L & Healy J,2017, “Accelerated hierarchical density-based clustering”, IEEE ICDM Workshops, pp.33–42, doi.org/10.1109/ICDMW.2017.12.
10. Sao S, Prokopenko S & Lebrun-Grandie D, 2024, “PANDORA: Parallel dendrogram construction for HDBSCAN clustering”, arXiv preprint arXiv:2401.06089, doi.org/10.48550/arXiv.2401.06089.
11. Schubert E, Sander J, Ester M, Kriegel H P & Xu X, 2017, “DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN”, ACM Transactions on Database Systems (TODS), vol(42(3)), pp.1–21. doi.org/10.1145/3068335.
12. Tiwari K K, Raguvanshi V & Jain A,2016, “DBSCAN: An assessment of density-based clustering and its approaches”, International Journal of Scientific Research & Engineering Trends, vol (2(5)), ISSN (Online): 2395-566X.