Journal of Information Systems Engineering and Management
2025, 10(41s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

A Frequency-Based Approach to Stop word Detection for Enhanced Clustering

1Mrs.S. Sujatha, 2Dr. Grasha Jacob
1Research Scholar Department of Computer Science Rani Anna Government College for Women Tirunelveli-627008, Tamil Nadu, India
Affiliated to Manonmaniam Sundaranar University, Tirunelveli
E-mail:sanksuj@gmail.com
2Associate Professor Department of Computer Science Government Arts & Science College Nagalapuram-628904, Tamil Nadu, India
Affiliated to Manonmaniam Sundaranar University, Tirunelveli
E-mail: grasharanjit@gmail.com

ARTICLE INFO

Received: 29 Dec 2024

Revised: 12 Feb 2025

Accepted: 27 Feb 2025

ABSTRACT

Languages are the most beautiful way we get to communicate with one another. In India, we have a plethora of languages that are used in our urban areas and remote rural regions, each having their dialect and tone. The Tamil language is one of the oldest languages to ever exist in the world. Tamil, a Dravidian language, boasts a rich history and unique features, including being one of the oldest living languages with a literature spanning over two millennia, and it is the first Indian language to be printed and published.

Clustering unstructured text data is a significant challenge in natural language processing, especially for low-resource languages like Tamil [1]. Agglomerative clustering is a hierarchical clustering algorithm that follows a bottom-up approach, progressively merging individual data points into clusters based on similarity. Unlike partition-based methods, it does not require a predefined number of clusters, making it advantageous for exploratory data analysis [10]. This paper explores a proposed methodology that includes dynamic stopword identification, language-specific preprocessing, and sentence embedding using BERT. The embeddings are then normalized using L2 normalization, followed by dimensionality reduction with UMAP. This approach leads to improved clustering performance as indicated by favourable metric values. For identifying the dynamic stop words the method that have been proposed is frequency-based approach and in which the final dynamic stop words are obtained by combining the common static words.

Keywords: Clustering, Agglomerative, Stop Words, Dynamic stop words, Tokenization, Dimensionality reduction.

1. INTRODUCTION

Unstructured data, particularly **Tamil text documents**, pose significant challenges in natural language processing due to their rich morphology and complex syntactic structures [14]. Clustering such data requires efficient preprocessing techniques to handle noise, stop words, and semantic variations[7].

One perilous facet of preprocessing is **stop word identification**, as stop words can significantly impact the clustering quality. Static stop word lists are traditional and may not capture dataset-specific frequent but non-informative words. **Dynamic stop word identification** addresses this limitation by automatically detecting and filtering out dataset-dependent stop words, leading to improved clustering performance.

Among various clustering approaches, **Agglomerative Hierarchical Clustering (AHC)** is widely used due to its ability to reveal hierarchical relationships between data points. Unlike partitioning methods like K-Means, AHC does not require a predefined number of clusters, making it suitable for exploratory data analysis.

This paper focusses on AHC's effectiveness in **Tamil document clustering** and evaluates it using standard clustering metrics, incorporating **dynamic stop word identification** to enhance cluster quality. The tokenization process is done using Indic_ NLP tokenization in the proposed method. The documents are pre-processed only after dynamic stop word identification. In the proposed methodology the few predefined stop words are defined and the frequency-based stop words are identified dynamically, the predefined and the obtained frequency-based stop words

are combined to get the final stop words list. The texts are preprocessed with the BERT embedding. After embedding the texts are normalized using L2 normalization. The embeddings are reduced to 50D using the UMAP dimensionality reduction before applying the AHC algorithm. It is evaluated using three metrics such as Silhouette score, Davis Bouldin Index, Calinski Harbassz Score for various cluster sizes. For experimentation two datasets were used one is Tamil News dataset and second dataset contains the general Tamil documents.

This paper is organized into four sections such as Section II is about related work, Section III is about Proposed algorithm, Section IV is about Experimental Analysis and Section V is Conclusion.

2. RELATED WORK

Clustering is a widely used unsupervised learning technique for text classification and topic modeling. K-Means, K-Medoids, and Hierarchical Clustering are commonly employed for textual data analysis. K-Means requires predefining the number of clusters, making it less flexible for exploratory analysis. In contrast, Agglomerative Clustering generates a hierarchical structure, which is more suitable for discovering complex relationships in unstructured text data [7].

Hierarchical techniques produce a nested sequence of partitions, with a single, all its inclusive cluster at the top and singleton clusters of individual points at the bottom. Each intermediate level can be viewed as combining two clusters from the next lower level (or splitting a cluster from the next higher level). The result of a hierarchical clustering algorithm can be graphically displayed as tree, called a dendrogram. This tree graphically displays the merging process and the intermediate clusters[15].

Multilingual large language models (MLLMs) have shown impressive capabilities across a variety of languages. However, efficacy can differ greatly between different language families, especially for those with limited linguistic resources. This report presents MERaLiON-TextLLM, a series of open-source language models specifically tailored to improve understanding and generation in Chinese, Indonesian, Malay, and Singlish. The initial released model is built on Llama-3-8B-Base and refined through a meticulously crafted process of continued pre-training and weight merging[9].

Fox[4] created a stopword list for English using Brown Corpus of 1,014,000 words. Here the author manually added the words that appeared more than 300 times in the corpus in a list and finalized it by manually analyzing. The stopword list contained 421 words. This approach is domain-independent and is widely used in retrieval systems.

Hao et al. [6] generated a stopword list using the weighted Chi-squared statistic technique for the Chinese language. Researchers observed that the suggested methodology effectively improved the F1 classification score by nearly 7%.

Raulji et al. proposed a dictionary-based approach to remove the stopwords for the Sanskrit Language. They used a predefined word list, compared it with the targeted text, and removed the stopwords. The researchers stated that from 87,000 words corpus, 11,200 words were removed as stopwords. This reduced the corpus size by 13% and reduced the feature space and CPU cycle [8].

Document summarization plays a vital role in the use and management of information dissemination across different languages [13]. This paper investigates a method for the production of summaries from Tamil newspaper text source. The primary goal is to create an effective and efficient tool that is able to summarize the given text documents in a form of meaningful extract of the original text document using centroid-based algorithm. The paper focuses on generating summaries using a centroid-based algorithm, which represents group of words that are statistically important for a document. Each sentence in a document is considered as a vector in a multi-dimensional space. The sentences that are nearest to the centroid value are considered as the most important sentences. The importance of a sentence is determined by three parameters the centroid value, the positional value, and the first sentence overlap. The score for each sentence is calculated and the redundancy between the sentences is eliminated using CSIS. Finally, the sentences are ranked and the sentences with highest score values are selected as summary [12].

M.S. Faathima Fayaza and Surangika Ranathunga in their proposed study on Tamil document clustering using Word embedding suggested that they have observed that fastText is able to identify words written in different styles by different publishers. Also, fastText can handle inflected words, in contrast to TF-IDF. One pass clustering algorithm outperforms the affinity propagation algorithm with both document representation approaches. It may be due to having a high number of single article clusters in the dataset[11].

Fayaza, Faathima & Farook, Farhath in their paper presented an approach to list out the stopwords in Tamil, which is a low-resource language. They have stated that so far, there is no predefined published stopword list for Tamil. The

Journal of Information Systems Engineering and Management
2025, 10(41s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

widely used technique for stopwords identification is based on term frequency. In their study, TF*IDF with threshold value is used to identify the stopwords for Tamil. The research resulted in the generation of stopword lists for general domain and domain-specific ones for local, international, sport, and entertainment domains. To evaluate its impact on Tamil NLP, it was used in document clustering using TF-IDF with one pass algorithm and FastText with the one-pass algorithm. The results revealed that the removal of stopwords at the preprocessing stage improved F-score, mean, median, and standard deviation in both the approaches [3].

3. PROPOSED ALGORITHM

The proposed algorithm dynamically identifies noisy, frequent words based on the specific dataset and Agglomerative Clustering with optimized linkage is used. Our method achieves more accurate and meaningful grouping of documents compared to traditional static methods.

3.1 DataSet Collection

To analyse the clustering of Tamil documents, the dataset named as “EnglishTamilText” and Tamil NewsTest have been taken which are in CSV format and the first dataset contains 119K records under two columns and the second dataset contains 3632 records with Tamil News and Tamil category columns.

3.2 Proposed Methodology

Unlike many studies that focus on general document clustering, this process specifically addresses Tamil text data, a language with unique linguistic characteristics and challenges. By incorporating dynamic stop words and adapting preprocessing techniques to handle the complexities of Tamil, this approach ensures more accurate clustering results for Tamil documents.

Traditional agglomerative clustering research uses basic text vectorization methods, this approach integrates state-of-the-art BERT-based word embeddings (such as paraphrase-multilingual-MiniLM-L12-v2) followed by post-processing steps like L2 normalization and UMAP for dimensionality reduction. This integration allows for more meaningful clustering by representing semantic relationships between words and reducing the dimensionality of high-dimensional embeddings.

Most studies use generic stop word lists and basic tokenization. In contrast, this algorithm dynamically identifies additional stop words from the data itself, ensuring that the stop word list is more tailored and relevant to the dataset that better captures the nuances of Tamil syntax and semantics.

While many studies rely on basic clustering metrics, this incorporates multiple evaluation metric such as Silhouette Score, Davies-Bouldin Index, and Calinski-Harabasz Score which provides a more comprehensive analysis of clustering quality.

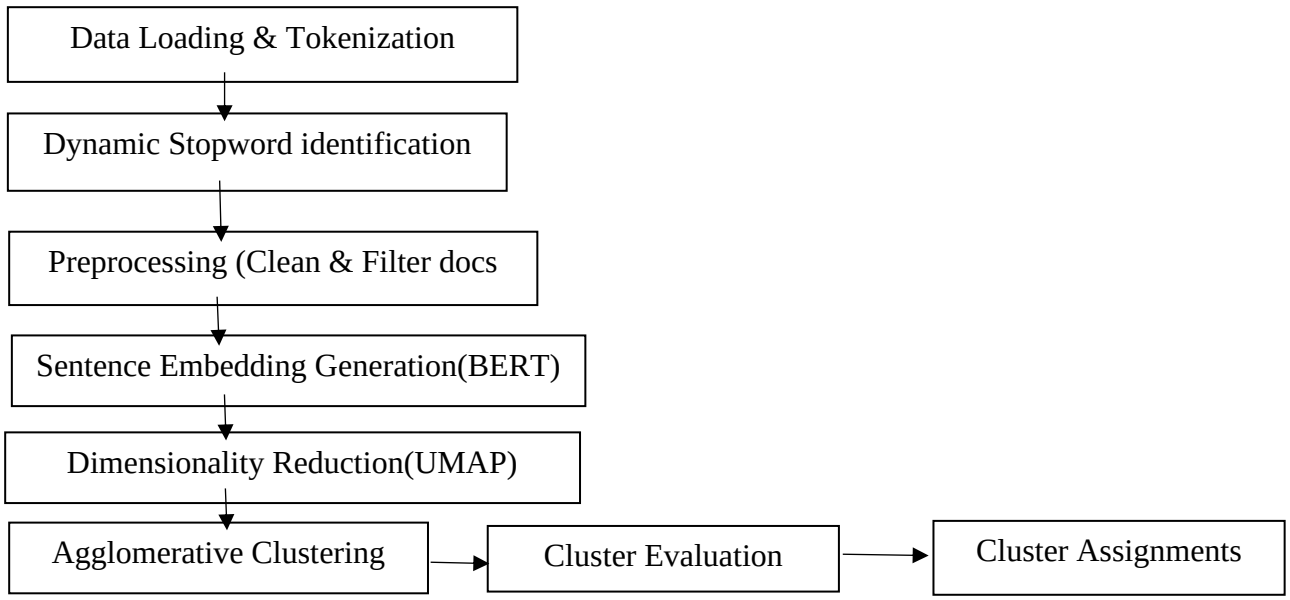
3.2.1 Proposed Algorithm1: Frequency-Based Method

- Load the Tamil text data.
- Preprocess the text.
- Identify dynamic stopwords using the following method:
  - Frequency Method: Find words that occur very frequently across documents.
- Remove the dynamic stopwords from the documents.
- Generate BERT embeddings for the cleaned Tamil documents.
- Apply UMAP to reduce the dimensionality of BERT embeddings (make it easier for clustering).
- Cluster the reduced embeddings using Agglomerative Clustering.
- Evaluate the clusters using Silhouette Score, Davies-Bouldin Index, and Calinski-Harabasz Score.

Proposed Algorithm for Agglomerative Clustering of Tamil Documents

Journal of Information Systems Engineering and Management
2025, 10(41s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article



A. Data Loading and Tokenization:

i) Load the dataset containing Tamil text documents (Engtamtext.csv) and extract the relevant text column (e.g., tamil).
ii) Tokenize each document using Indic NLP's indic_tokenize library, which supports Tamil language tokenization.
For Example sample documents from Engtamtext.csv file,

ID	Tamil Text
1	மத்திய அரசு புதிய வரி சட்டங்களை அறிவித்தது.
2	தமிழக அரசு கல்வி திருத்தங்களை அறிமுகப்படுத்தியது.
3	இந்திய கிரிக்கெட் அணி உலகக் கோப்பையை வென்றது.
4	புதிய தொலைக்காட்சி சேனல் துவக்கம்.
5	அரசாங்கம் மருத்துவ திட்டங்களை மேம்படுத்துகிறது.

After tokenization

ID	Tokens
1	['மத்திய', 'அரசு', 'புதிய', 'வரி', 'சட்டங்களை', 'அறிவித்தது']
2	['தமிழக', 'அரசு', 'கல்வி', 'திருத்தங்களை', 'அறிமுகப்படுத்தியது']
3	['இந்திய', 'கிரிக்கெட்', 'அணி', 'உலகக்', 'கோப்பையை', 'வென்றது']
4	['புதிய', 'தொலைக்காட்சி', 'சேனல்', 'துவக்கம்']
5	['அரசாங்கம்', 'மருத்துவ', 'திட்டங்களை', 'மேம்படுத்துகிறது']

B. Dynamic Stop Word Identification:

i) Define a set of general Tamil stop words
(e.g., "அது", "இது", etc.).
ii) Identify additional stop words dynamically based on the frequency of terms in the entire corpus. The most frequent tokens are considered as additional stop words if they occur more than a threshold (e.g., 10 times).
Sample dynamic stopwords
['இது', 'அது', 'மற்றும்', 'ஆகிய', 'அரசு', 'அரசாங்கம்', 'புதிய']

C. Preprocessing:

Journal of Information Systems Engineering and Management
2025, 10(41s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

i) Clean the documents by removing numerals and special characters, and filter out tokens that appear in the combined list of general, and dynamically identified stop words.
ii) Apply tokenization again on the cleaned documents, converting each token to lowercase and reconstructing the preprocessed documents.
Sample documents after cleaning the documents.

ID	After Removing Stopwords
1	['வரி', 'சட்டங்களை', 'அறிவித்தது']
2	['தமிழக', 'கல்வி', 'திருத்தங்களை', 'அறிமுகப்படுத்தியது']
3	['இந்திய', 'கிரிக்கெட்', 'அணி', 'உலகக்', 'கோப்பையை', 'வென்றது']
4	['தொலைக்காட்சி', 'சேனல்', 'துவக்கம்']
5	['மருத்துவ', 'திட்டங்களை', 'மேம்படுத்துகிறது']

D. Sentence Embedding Generation:
i) Uses a pre-trained multilingual BERT model (paraphrase-multilingual-MiniLM-L12-v2) from the Sentence Transformers library to encode the pre-processed Tamil documents into dense vector embeddings.
ii) Applied L2 normalization to the generated embeddings to ensure they are suitable for clustering algorithms.
E. Dimensionality Reduction:
i) Reduce the dimensionality of the embeddings using **UMAP (Uniform Manifold Approximation and Projection)**. This step helps in improving the efficiency of clustering by reducing the number of features to a manageable size (e.g., 50 dimensions), while preserving the semantic structure of the embeddings. UMAP is found to be the best method.
F. Agglomerative Clustering:
i)Apply **Agglomerative Clustering** to the reduced embeddings with a chosen number of clusters (n_clusters = 5 in this case) and a specified linkage method. This algorithm hierarchically clusters the documents based on their similarity in the embedding space.
G. Cluster Evaluation:
Evaluate the quality of the clustering using three clustering metrics:
i. **Silhouette Score**: Measures how similar each document is to its own cluster compared to other clusters.
ii. **Davies-Bouldin Index:** Measures the average similarity ratio of each cluster with the cluster that is most similar to it.
iii. **Calinski-Harabasz Score:** Measures the ratio of between-cluster dispersion to within-cluster dispersion.
These metrics helps to assess the validity and robustness of the clustering.
H. Cluster Assignments:
i) Assign documents to their corresponding clusters based on the labels generated by the Agglomerative Clustering model. The documents in each cluster are printed, along with the first few documents in each cluster for review.

4. EXPERIMENTAL ANALYSIS

To analyse the proposed agglomerative clustering algorithm two **Tamil text datasets** were used. In the proposed methodology the dynamic stop words are identified based on frequency-based method. Few common words are defined as static words. The final dynamic stop word lists are obtained after combining the static and dynamic stop words. To improve the number of dynamic stop words and clean the documents effectively the dynamics stop words are identified by frequency-based method,
The cleaned documents are passed to the BERT embedding. BERT (Bidirectional Encoder Representations from Transformers) is a deep learning model developed by Google to understand the meaning of words in context.
Example: In Tamil, the word "கோடு" could mean "line" or "boundary" depending on the sentence.

BERT understands these meanings properly by reading both before and after the word. In particular a pretrained Sentence Transformer model “all-MiniLM-L6-v2” BERT embedding is used. It is used because it supports multiple languages like English, Tamil, Hindi, etc.

In the proposed algorithm the cleaned Tamil text is passed into the BERT model. It produces a fixed-size 384-dimensional vector for each Tamil document. This vector captures the semantic meaning of the entire text even

Journal of Information Systems Engineering and Management
2025, 10(41s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

for short Tamil texts. It understands deep relationships between Tamil words. It is better than simple word count or TF-IDF, which don't understand meaning.

UMAP (Uniform Manifold Approximation and Projection) dimensionality reduction technique which is like PCA, but better for complex data. UMAP is needed here because the BERT embeddings contain 384 dimensions. The High dimensions generally slow down clustering, introduces noise and it makes visualization hard. UMAP reduces 384-D to 5-D vectors. It preserves the structure (similar documents stay close). It removes redundant information.

In the proposed algorithm it takes all BERT vectors (for Tamil documents) and it builds a low-dimensional (5-D) version that keeps nearby points close and far points far. Now, the data is ready for fast clustering. UMAP is good for Tamil text because Tamil documents are very semantic (deep meaning even in short texts). UMAP preserves local meaning much better than PCA.

Implementation

Proposed Agglomerative clustering algorithm with dynamic stopword identification is implemented using **scikit-learn** in Python, with the three linkage methods.

The following libraries have been used,

```
import pandas as pd
from sklearn.cluster import AgglomerativeClustering
from sklearn.metrics import silhouette_score,davies_bouldin_score, calinski_harabasz_score
from sklearn.metrics import classification_report
from sentence_transformers import SentenceTransformer
import umap
from collections import Counter
from indicnlp.tokenize import indic_tokenize
import re
import matplotlib.pyplot as plt
import numpy as np
from sklearn.preprocessing import normalize
```

The following stop words have been used initially,

```
[
  "அது", "இது", "என்று", "ஒரு", "ஆகும்", "நான்", "நீ", "இந்த", "அந்த", "இல்லை",
  "ஆனால்", "முதல்", "மத்தும்", "என்", "என்ன"
]
```

Dynamic Stop Words Identification

EngTamText.CSV Dataset:

Dynamically Identified Stop Words: Frequency -based Method

Total no of dynamic stop words obtained using frequency-based method is 61

The sample dynamic stop words that have been identified using frequency-based method is as follows,

['.', ',', '\n', 'மற்றும்', 'மத்திய', 'பிரதமர்', ')', '-', '(', 'இந்த', 'அமைச்சரவை', 'திரு', 'ஒப்புதல்', 'ஒப்பந்தம்', 'மோடி', 'நரேந்திர', 'மூலம்', 'என்று', 'இது', 'புரிந்துணர்வு', 'இரு', 'இந்திய', 'வேண்டும்', 'விமான', 'உள்ள', 'இந்தியா', ';', ':', 'அவர்', 'தேசிய', 'பாதுகாப்பு', 'மேலும்', 'ஒரு', 'தலைமையில்', 'முதலீட்டு', 'ஒப்பந்தத்திற்கு', 'தனது', 'நாம்', 'ஏ', 'நிதி', 'நீண்டகால', 'என்ற', 'அலுவலகம்', 'இந்தியாவின்', 'பல்வேறு', 'நடைபெற்ற', 'இருக்கும்', 'ல்', 'அடிப்படையில்', 'குறித்து']

The final stop words samples obtained after removing all the duplicates using frequency-based method 64 and it is as follows,

'உள்ள', 'முதலீட்டு', 'பல்வேறு', 'ஆனால்', '-', 'முதல்', 'நான்', 'பிரதமர்', 'அடிப்படையில்', ',', ')', 'மேலும்', 'அமைச்சரவை', 'நிதி', 'நடைபெற்ற', 'ஆகும்', 'தலைமையில்', 'அந்த', 'இல்லை', '(', 'விமான', 'இரு', 'நாம்', 'மத்திய', 'மத்தும்', 'நரேந்திர', 'இந்தியா', 'மற்றும்', 'அலுவலகம்', 'இந்த', 'ல்', 'ஒப்புதல்', 'குறித்து', 'திரு', 'மூலம்', 'மோடி', 'அது', 'இந்திய', ';', 'இருக்கும்', 'ஒப்பந்தத்திற்கு', 'வேண்டும்', 'ஒரு', 'தேசிய', 'நீண்டகால', 'இந்தியாவின்', 'ஏ', '.', 'இது', 'என்', 'நீ', 'ஒப்பந்தம்', 'என்ன', '\n', 'புரிந்துணர்வு', 'தனது', 'அவர்', 'என்ற', 'என்று', 'பாதுகாப்பு', ':']

Journal of Information Systems Engineering and Management
2025, 10(41s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

The following metrics have been used to assess the clustering performance,

i) Silhouette Score
ii) Davies-Bouldin index
iii) Calinski-Harabasz Score

4.1 SILHOUETTE SCORE:

The **silhouette score** evaluates the quality of clustering by measuring how similar a point is to its own cluster (cohesion) compared to other clusters (separation). It ranges from -1 (poor clustering) to 1 (optimal clustering), with values near 0 indicating overlapping clusters.

Steps to Calculate Silhouette Score

For each data point i:

1. **Calculate intra-cluster distance (a(i))**
   - Compute the mean distance between i and all other points in the same cluster.
2. **Calculate nearest-cluster distance (b(i))**
   - Compute the mean distance between i and all points in the nearest different
   - cluster (i.e., the cluster with the smallest average distance to i).
3. **Compute silhouette score for point i**

$$S = \frac{b - a}{\max(a, b)}$$

where:

a = average intra-cluster distance (within the same cluster).

B = average nearest-cluster distance (to the closest different cluster).

- If s(i) is close to 1 → Well-clustered.
- If s(i) is close to 0 → Overlapping clusters.
- If s(i) is negative → Incorrect clustering.

4. **Compute overall silhouette score**
   - Average s(i) over all points to obtain the final silhouette score.

This process is typically computed using **Euclidean distance** but can be adapted to other distance metrics.[4].

4.2 DAVIES-BOULDIN INDEX

The Davies-Bouldin Index (DBI) evaluates clustering quality by measuring the average similarity between clusters. A lower DBI indicates better clustering.

$$DBI = 1/N\sum_{I=1}^{N} max_{j \neq i}(s_i + s_j)/d_{ij}$$

where:

- s_i, s_j = average distance between points in clusters i and j.
- d_{ij} = distance between cluster centers i and j.
- if DBI is lower then it produces Better-separated and more compact clusters.
- if DBI is higher then it leads to Poor clustering with overlapping or scattered clusters.[14]

4.3 CALINSKI-HARABASZ SCORE

The **Calinski-Harabasz Index (CH Index)**, also called the **Variance Ratio Criterion (VRC)**, evaluates clustering quality based on between-cluster and within-cluster dispersion. A **higher CH Index** indicates better clustering.

$$\mathrm{CH} = (B_K/(\mathrm{k}-1))/(W_k/(\mathrm{n}-\mathrm{k}))$$

where:

- B_K= between-cluster dispersion.
- W_K= within-cluster dispersion.
- k = number of clusters.
- n = total points.

If CH Index is higher it will produce better clustering (more compact and well-separated clusters).

If CH Index is lower it will lead to Poor clustering (overlapping or scattered clusters).[2]

Journal of Information Systems Engineering and Management
2025, 10(41s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

While most agglomerative clustering approaches for Tamil text rely on traditional text preprocessing and basic vectorization, this study distinguishes itself by incorporating advanced BERT-based embeddings, dynamic stop word detection, and flexible clustering evaluation metrics. The use of post-processing techniques like L2 normalization and UMAP for dimensionality reduction ensures more meaningful clustering of Tamil documents, making it particularly suited for language-specific clustering tasks.

The proposed algorithms have been evaluated in the three linkage methods such as complete, average and ward and the metrics values are obtained. To prove the efficiency of the proposed algorithm the agglomerative clustering algorithm is implemented with static stop words and compared with the proposed algorithm. The algorithms have been evaluated for five cluster sizes such as n=3,5,7,10,15. The agglomerative algorithm and the proposed algorithm have been implemented with three linkage methods such as 'ward', 'average' and 'complete'.

METRIC VALUES FOR EngTamText DataSet :

Agglomerative Algorithm with Static Stop Words:

Linkage Method = Ward

Clusters	Silhouette	DBI	CH SCORE
3	0.2596	1.9437	81.89
5	0.2614	1.684	70.55
7	0.2816	1.55	67.85
10	0.3487	1.3014	65.65
15	**0.3891**	**1.2302**	**62.64**

Table 4.1 Metric Values for Agglomerative Algorithm with Ward linkage-EngamText Dataset

Best cluster size: k = 15

Silhouette: 0.3891, DBI: 1.2302, CH Score: 62.64

Agglomerative Algorithm with Static Stop Words:

Linkage Method = Complete

Clusters	Silhouette	DBI	CH SCORE
3	**0.3624**	**1.2608**	**85.46**
5	0.3031	1.6218	77.85
7	0.31	1.5559	67.9
10	0.3461	1.3896	65.98
15	0.3562	1.2804	55.7

Table 4.2 Metric Values for Agglomerative Algorithm with Complete Linkage-EngamText Dataset

Best cluster size: k = 3

Silhouette: 0.3624, DBI: 1.2608, CH Score: 85.46

Agglomerative Algorithm with Static Stop Words:

Linkage Method = Average

Clusters	Silhouette	DBI	CH SCORE
3	0.2951	1.5813	85.08
5	0.2944	1.5778	70.73
7	0.3156	1.472	66.39
10	0.3584	1.3031	62.17
15	**0.41**	**1.1267**	**63.32**

Table 4.3 Metric Values for Agglomerative Algorithm with Average Linkage -EngamText Dataset

Journal of Information Systems Engineering and Management
2025, 10(41s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Best cluster size: k = 15
Silhouette: 0.4100, DBI: 1.1267, CH Score: 63.32

When applying agglomerative clustering to the *Engtamtext* dataset with **static stopwords**, the overall clustering performance was moderate, with none of the linkage methods achieving high-quality cluster separation. Among the three linkage strategies, **Average linkage at k=15** performed the best, yielding the **highest silhouette score of 0.4100** and the **lowest Davies-Bouldin Index (1.1267)**, indicating relatively more cohesive and better-separated clusters. **Ward linkage** followed closely with a silhouette score of 0.3891 and a DBI of 1.2302 at the same cluster size, but it lagged in Calinski-Harabasz score. The **Complete linkage** method showed the **highest CH score (85.46)** at **k=3**, but it had a lower silhouette score (0.3624) and a higher DBI (1.2608), suggesting less consistent clustering quality. Despite these relative differences, all three linkage methods under static stopword conditions produced **limited cluster separation**, as reflected in the **generally low silhouette scores and CH scores**, and **higher DBI values**. This indicates that static stopword removal may not be sufficient to achieve optimal clustering in complex, mixed-language text data like *Engtamtext*. Among the **static stopword results**, the **best clustering configuration** is achieved using the **Average linkage method with 15 clusters (k=15)**.

Proposed Algorithm -Frequency Based Method With Dynamic Stop Words
Linkage Method = Ward

Clusters	Silhouette	DBI	CH SCORE
3	**0.8282**	**0.4281**	**1464.04**
5	0.722	0.6347	1818.03
7	0.6824	0.6803	1942.54
10	0.6296	0.6863	2050.65
15	0.6199	0.786	2412.41

Table 4.4 Metric Values for Proposed Algorithm with Ward Linkage -EngamText Dataset

Best cluster size: k = 3
Silhouette: 0.8282, DBI: 0.4281, CH Score: 1464.04

Proposed Algorithm -Frequency Based Method With Dynamic Stop Words
Linkage Method = Complete

Clusters	Silhouette	DBI	CH SCORE
3	**0.8282**	**0.4281**	**1464.04**
5	0.6796	0.5471	1533.12
7	0.5878	0.6875	1708.52
10	0.6047	0.789	1965.24
15	0.5737	0.8131	2247.8

Table 4.5 Metric Values for Proposed Algorithm with Complete Linkage -EngamText Dataset

Best cluster size: k = 3
Silhouette: 0.8282, DBI: 0.4281, CH Score: 1464.04

Proposed Algorithm -Frequency Based Method With Dynamic Stop Words
Linkage Method = Average

Clusters	Silhouette	DBI	CH SCORE
3	**0.8282**	**0.4281**	**1464.04**
5	0.7705	0.5549	1629.02
7	0.6834	0.6792	1946.81
10	0.6169	0.6633	1877.92
15	0.6086	0.6976	2286.71

Table 4.6 Metric Values for Proposed Algorithm with Average Linkage -EngamText Dataset

Best cluster size: k = 3

Journal of Information Systems Engineering and Management
2025, 10(41s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Silhouette: 0.8282, DBI: 0.4281, CH Score: 1464.04

Among the evaluated cluster sizes for dynamic stop words using Agglomerative Clustering, the **best-performing cluster size is 3**, consistently across all three linkage methods — **ward**, **complete**, and **average**. This configuration achieved the **highest Silhouette Score of 0.8282**, indicating the most cohesive and well-separated clusters. It also recorded the **lowest Davies-Bouldin Index (DBI) of 0.4281**, reflecting compact and clearly distinct clusters. Although the Calinski-Harabasz (CH) score increases with the number of clusters (reaching a peak at size 15), this metric alone is not sufficient to justify larger cluster sizes since the Silhouette and DBI metrics favour 3. Therefore, **cluster size 3 is the most optimal choice** for this dataset with dynamic stop words under all linkage strategies.

METRIC VALUES FOR Tamil NewsTest DataSet :
Agglomerative Algorithm with Static Stop Words:
Linkage Method = Ward

Clusters	Silhouette	DBI	CH Score
3	0.1563	2.5556	421.7
5	0.1766	2.1333	400.59
7	0.1805	1.8849	365.88
10	0.1967	1.9431	338.46
15	**0.2258**	**1.8392**	**303.08**

Table 4.7 Metric Values for Agglomerative Algorithm with Ward linkage -TamilNewsTest Dataset

Best cluster size: k = 15
Silhouette: 0.2258, DBI: 1.8392, CH Score: 303.08

Agglomerative Algorithm with Static Stop Words:
Linkage Method = complete

Clusters	Silhouette	DBI	CH Score
3	0.1434	2.8038	435.35
5	0.1396	2.2204	377.53
7	0.1397	2.3926	320.36
10	0.1518	2.265	284.18
15	**0.211**	**1.9302**	**279.49**

Table 4.8 Metric Values for Agglomerative Algorithm with Complete linkage -TamilNewsTest Dataset

Best cluster size: k = 15
Silhouette: 0.2110, DBI: 1.9302, CH Score: 279.49

Agglomerative Algorithm with Static Stop Words:
Linkage Method = Average

Clusters	Silhouette	DBI	CH Score
3	0.1859	1.0897	169.52
5	0.1629	1.9992	321.36
7	0.1422	1.9227	285.31
10	0.1811	1.8838	287.88
15	**0.2146**	**1.6989**	**262.4**

Journal of Information Systems Engineering and Management
2025, 10(41s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Table 4.9 Metric Values for Agglomerative Algorithm with Average linkage -TamilNewsTest Dataset

Best cluster size: k = 15

Silhouette: 0.2146, DBI: 1.6989, CH Score: 262.40

Three linkage methods (Ward, Complete, and Average) were evaluated using Agglomerative Clustering on various cluster sizes (k = 3, 5, 7, 10, 15). Performance was measured using Silhouette Score, Davies-Bouldin Index (DBI), and Calinski-Harabasz (CH) Score. The **Ward Linkage** method produced the **best overall performance** at **k = 15**, achieving the **highest Silhouette score (0.2258)** and **CH score (303.08)**, indicating well-separated and cohesive clusters. It is very clear that **Ward linkage with k = 15** is the most effective configuration for clustering the TamilNewsTest dataset using static stopwords.

METRIC VALUES FOR TamilNewsTest DataSet :
Proposed Algorithm -Frequency Based Method With Dynamic Stop Words
Linkage Method = Ward

Clusters	Silhouette	DBI	CH SCORE
3	**0.541**	**0.927**	**4722.003**
5	0.467	0.989	3713.541
7	0.402	1.142	3600.251
10	0.389	1.12	3263.177
15	0.321	1.205	2718.2

Table 4.10 Metric Values for Proposed Algorithm with Ward Linkage -TamilNewsTest Dataset

Best cluster size: k = 3

Silhouette: 0.541, DBI: 0.927, CH Score: 4722.003

Proposed Algorithm -Frequency Based Method With Dynamic Stop Words
Linkage Method = Complete

Clusters	Silhouette	DBI	CH SCORE
3	**0.507**	**0.794**	**3798.188**
5	0.407	1.164	3859.732
7	0.344	1.185	3356.683
10	0.318	1.301	3015.102
15	0.324	1.334	2739.074

Table 4.11 Metric Values for Proposed Algorithm with Complete Linkage -TamilNewsTest Dataset

Best cluster size: k = 3

Silhouette: 0.507, DBI: 0.794, CH Score: 3798.188

Proposed Algorithm -Frequency Based Method With Dynamic Stop Words
Linkage Method = Average

Clusters	Silhouette	DBI	CH SCORE
3	**0.526**	**0.985**	**4674.715**
5	0.456	0.947	3883.974
7	0.383	1.156	3500.527
10	0.38	1.195	3244.222
15	0.314	1.233	2614.896

Table 4.12 Metric Values for Proposed Algorithm with Average Linkage -TamilNewsTest Dataset

Best cluster size: k = 3

Journal of Information Systems Engineering and Management
2025, 10(41s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Silhouette: 0.526, DBI: 0.985, CH Score: 4674.715

The **Proposed Frequency-Based Algorithm with Dynamic Stopwords** was evaluated using three linkage methods — **Ward**, **Complete**, and **Average** — across cluster sizes (k = 3, 5, 7, 10, 15). Performance was measured using **Silhouette Score**, **Davies-Bouldin Index (DBI)**, and **Calinski-Harabasz (CH) Score)**. The **Ward Linkage method with k = 3** delivers the best clustering quality with **Highest Silhouette Score, Strong DBI and Best CH Score.** Dynamic stop word filtering significantly improves clustering performance, especially with Ward linkage.

In both the agglomerative algorithm with static stop words and proposed algorithm with dynamic stop words the best cluster size with the metric values is automatically obtained.

In the following tables the best metric values obtained from each dataset is listed to show the improvement in clustering performance.

S.No	Metrics	Agglomerative clustering algorithm with statics stops words	Proposed Algorithm Frequency-based method with dynamic stop words
1.	Silhouette Score	0.4100	**0.8282**
2.	Davies BouldIn Index	1.1267	**0.4281**
3.	Calinski-Harabasz Index	63.32	**1464.04**

Table 4.13 Best Metric Values for EngTamtext Dataset

The proposed algorithm, which uses a frequency-based method with dynamic stopword removal, significantly outperforms the agglomerative clustering algorithm that relies on static stopwords. The silhouette score, which measures the quality of clustering in terms of cohesion and separation, improved markedly from 0.3891 to 0.620, indicating much better-defined clusters. The Davies-Bouldin Index decreased from 1.2302 to 0.786, suggesting enhanced cluster compactness and separation. Additionally, the Calinski-Harabasz Index showed a dramatic increase from 62.64 to 2412.407, reflecting far superior inter-cluster dispersion relative to intra-cluster cohesion. These improvements across all three metrics clearly demonstrate that the proposed dynamic stopword approach yields more meaningful, compact, and well-separated clusters compared to the traditional static method.

S.No	Metrics	Agglomerative clustering algorithm with statics stop words	Proposed Algorithm Frequency-based method with dynamic stop words
1.	Silhouette Score	0.2258	**0.541**
2.	Davies BouldIn Index	1.8392	**0.927**
3.	Calinski-Harabasz Index	303.08	**4722.003**

Table 4.14 Best Metric Values for TamilNewsTest Dataset

The comparison between agglomerative algorithm with static stop words and the proposed algorithm with dynamic stopword removal in Tamil text clustering clearly demonstrates the superiority of dynamic stopwords. Using Agglomerative Clustering with Ward linkage, the dynamic approach achieved a silhouette score of 0.541, which is more than double the best score obtained with static stopwords (0.2398), indicating better-defined and more cohesive clusters. The Davies-Bouldin Index, which measures cluster separation and compactness, was significantly lower with dynamic stopwords (0.927 vs. 1.6627), showing improved inter-cluster separation. Additionally, the Calinski-Harabasz Score, which evaluates the ratio of between-cluster dispersion to within-cluster dispersion, was markedly higher for dynamic stopwords (4722.00 compared to 266.84), further confirming the quality of clustering.

Journal of Information Systems Engineering and Management
2025, 10(41s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Notably, the optimal number of clusters for dynamic stopwords was just 3, compared to 15 for static, indicating more natural grouping in the data. Overall, these metrics conclusively prove that dynamic stopword removal results in more meaningful and effective clustering than static stopwords.

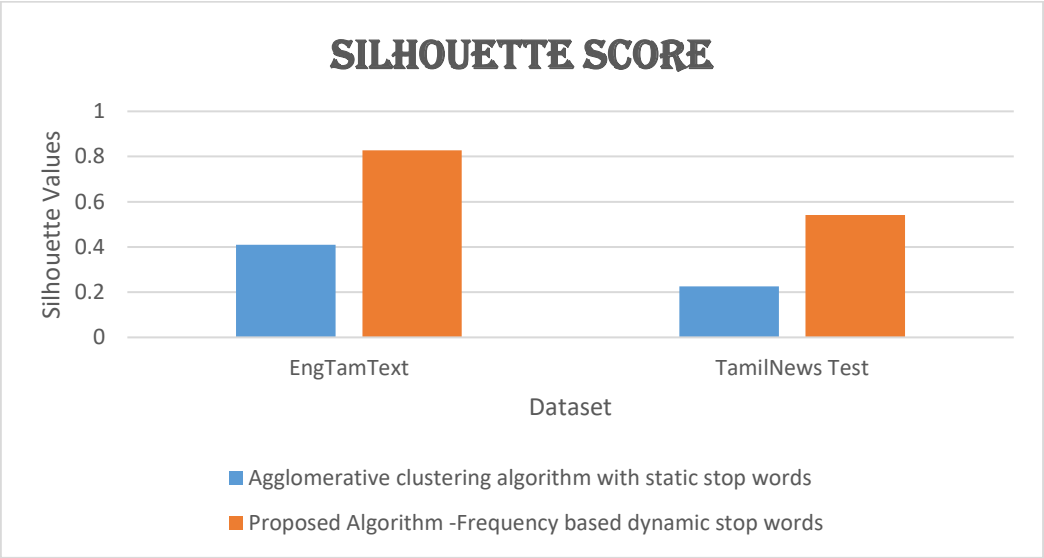


Fig 4.1 Silhouette Scores of Agglomerative algorithm with static stop words and Proposed Algorithm -frequency based method with dynamic stop words

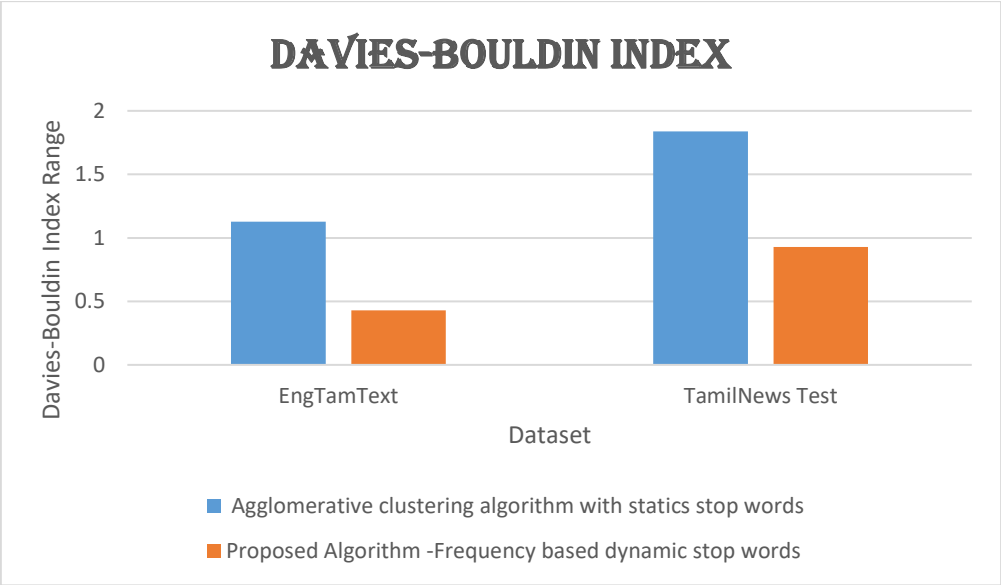


Fig 4.2 Davies Bouldin Index of Agglomerative algorithm with static stop words and Proposed Algorithm - frequency based method with dynamic stop words

Journal of Information Systems Engineering and Management
2025, 10(41s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

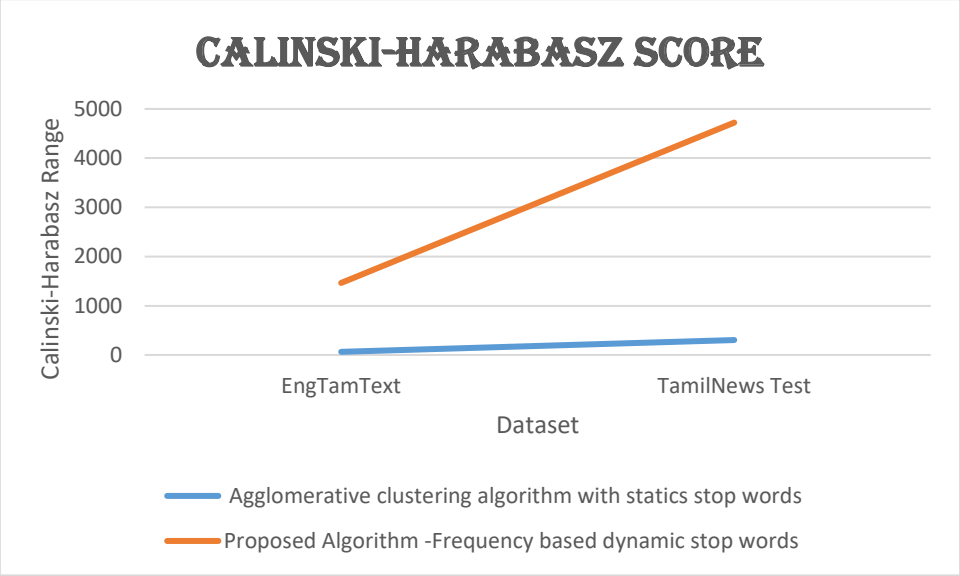


Fig 4.3 Calinski-Harabasz Index of Agglomerative algorithm with static stop words and Proposed Algorithm - frequency based method with dynamic stop words

The proposed algorithm produces better result because the Silhouette score is nearer to 1, the DBI is less than 1 and CH is higher than the agglomerative algorithm with static stop words.
The findings of the proposed algorithm are listed below

- **Effective Stopword Removal**

The Dynamic stop word by frequency-based method removes irrelevant high-frequency terms missed by static lists.

- **Better Topic Separation**

Clusters formed after preprocessing with dynamic stop words exhibited clearer separation and thematic consistency compared to baseline methods.

- **Better Clustering Performance**

The Clustering metrics (Silhouette, Davies-Bouldin, Calinski-Harabasz) shows significant improvement when compared to other stop word methods.

- **Adaptability Across Datasets**

The method adjusted automatically to different datasets and domains.

Novel Aspects of the Proposed Algorithms:

1. **Dynamic Stop Word Identification**:
   - The algorithm dynamically identifies additional stop words based on the frequency of terms in the corpus, ensuring that irrelevant or excessively common terms are filtered out.

2. **Use of Multilingual BERT for Sentence Embeddings**:
   - Many studies rely on the Word embeddings but the proposed algorithms leverages a pre-trained multilingual BERT model to generate embeddings for Tamil documents, which helps capture semantic information that traditional bag-of-words models might miss.
3. **Clustering with Agglomerative Hierarchical Approach**:
   - The use of **Agglomerative Clustering** allows the algorithm to form a hierarchy of clusters, potentially identifying more nuanced groupings than flat clustering algorithms like K-means.
4. **Dimensionality Reduction with UMAP**:
   - The application of **UMAP** for dimensionality reduction improves clustering performance by making it computationally efficient to handle high-dimensional data while preserving important relationships.
5. **Comprehensive Evaluation**:
   - The algorithm includes multiple metrics for cluster evaluation, allowing for a detailed assessment of clustering quality and robustness.

Journal of Information Systems Engineering and Management
2025, 10(41s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

6. **Visualization with Cluster Centroids**:
   - The algorithm enhances interpretability by visualizing not only the clustering results but also the centroids, helping users understand the core of each cluster.

5. CONCLUSION

The proposed algorithm with frequency-based method of dynamic stop words detection when incorporated with Agglomerative clustering proves to be an effective method for text clustering, especially for Tamil language data. Incorporating **dynamic stopword identification** enhances the clustering process by removing dataset-specific noise, leading to better-separated and more coherent clusters. The proposed algorithm with frequency-based method with agglomerative clustering outperforms well for Tamil text document.

FUTURE WORK

The Dynamic Stop Word Identification can be enhanced by developing adaptive algorithms that refine stop word lists based on topic coherence or entropy measures. Also, we can try to explore deep learning models (e.g., transformer-based embeddings) for context-aware stop word removal. Future research can explore its optimization using parallel computing for large-scale datasets and advanced linguistic processing for stopword identification.

REFERENCES:

[1] Aggarwal, C. C., and Zhai, C. “A Survey of Text Clustering Algorithms in Mining Text Data”, 77-128, Springer, 2012.
[2] Angiulli, F., and Fiumara, G. “Clustering algorithms for large-scale datasets”,A review and comparison. Computers, 2021.
[3]Fayaza, Faathima & Farook, Farhath., “Towards Stopwords Identification in Tamil Text Clustering”. International Journal of Advanced Computer Science and Applications,2021.
[4] Fox.C. "A Stop List for General Text," ACM SIGIR Forum, vol. 24, no. 1–2, 19–21, doi: 10.1145/378881.378888., 1989.
[5] Hennig, C. (2015). “Cluster analysis: A survey”, Wiley Interdisciplinary Reviews: Computational Statistics, *7*(6), 345-358,2015.
[6] L. Hao and L. Hao, "Automatic identification of stop words in chinese text classification," Proc. - Int. Conf. Comput. Sci. Softw. Eng. CSSE 2008, vol. 1, 718–722, 2008.
[7] Jain, A. K., Murty, M. N., and Flynn, P. J. “Data Clustering: A Review. ACM Computing Surveys”, 31(3), 264-323,1999.
[8] K.Jaideepsinh and Saini, Jatinderkumar “Stop-Word Removal Algorithm and its Implementation for Sanskrit Language”, International Journal of Computer Applications vol. 150, no. 2, 15–17, 2016.
[9] Malavarayar Vijayanandan Vithulan, “Cross-Lingual Document Clustering for Sinhala, Tamil, And English Using Pre-Trained Multilingual Language Models”, Department of Computer Science and Engineering University of Moratuwa Sri Lanka, July 2022.
[10] Manning, C. D., Raghavan, P., and Schütze, H. “Introduction to Information Retrieval”, Cambridge University Press,2008.
[11] Mohamed, S.S. and Hariharan, S. “Experiments on document clustering in Tamil language,” ARPN Journal of Engineering and Applied Sciences. 13. 3564-3569,2018.
[12] M. S. Faathima Fayaza and S. Ranathunga, "Tamil News Clustering Using Word Embeddings," 2020 Moratuwa Engineering Research Conference (MERCon), Moratuwa, Sri Lanka, 277-282, 2020.
[13] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., “Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research”, 12, 2825-2830,2011.
[14] R. Anita and C. N. Subalalitha, "An Approach to Cluster Tamil Literatures Using Discourse Connectives," 2019 IEEE 1st International Conference on Energy, Systems and Information Processing (ICESIP), Chennai, India, 2019, 1-4,2019.
[15] Steinbach, Michael and Karypis, George and Kumar, Vipin. “A Comparison of Document Clustering Techniques”, Proceedings of the International KDD Workshop on Text Mining,2009.
[16] Zhang, Y., and Wang, C. “A comparative study of clustering validation indices.” International Journal of Machine Learning and Cybernetics, 11(1), 93-106. 2020.