Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Enhanced Birch Clustering With Optimal Threshold Value

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

[1]Research 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
[2]Associate 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

Document clustering refers to the nonoverlapping partitioning of a set of documents into classes based on their content. This method is used in computer science to organize and summarize large collections of documents for various purposes such as collection browsing, corpus summarization, and document classification. Clustering, in the general sense, is the nonoverlapping partitioning of a set of objects into classes. Text can be clustered at various levels of granularity by considering cluster objects as documents, paragraphs, sentences, or phrases. This paper performs clustering on Tamil text data using a combination of three feature extraction methods—TF-IDF, LDA, and BERT—and clusters the data with the BIRCH algorithm. The proposed approach enhances the clustering quality by leveraging lexical, topical, and semantic information. This paper also out focusses the cluster performance improvement by incorporating the optimal threshold value to obtain the proper metrics values. The efficiency of the enhanced BIRCH algorithm is demonstrated through experimental analysis. Comparative experiments were conducted using the standard BIRCH algorithm, the proposed method without the optimal threshold, and the proposed method with the optimal threshold. The results clearly show that incorporating the optimal threshold significantly improves performance, thereby validating the effectiveness of the proposed approach.

Keywords – Clustering, BIRCH, Term Frequency, Stop Words, Dynamic stopwords, Tokenization, Stemming, Optimal Threshold.

1. INTRODUCTION

Clustering plays a pivotal role in the field of text mining and natural language processing (NLP), especially when dealing with large-scale and high-dimensional datasets [1]. In the context of Tamil text data, effective clustering techniques are essential for uncovering hidden structures, grouping similar documents, and facilitating various downstream applications such as information retrieval, topic modeling, and sentiment analysis [2]. However, traditional clustering algorithms often struggle with scalability, efficiency, and the complexities of processing morphologically rich languages like Tamil [6].

Document clustering, which extracts and visualizes complex relations inherent to scientific literature, has been a powerful approach to find and retrieve documents from rapidly growing datasets. Document clustering is the important techniques for organizing files in an unsupervised manner. When documents are represented as term vectors, the clustering methods can be applied. The document space is continually of large dimensionality, ranging from various hundreds to thousands.

Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) is a widely recognized hierarchical clustering method known for its ability to handle large datasets efficiently [13]. BIRCH incrementally and dynamically clusters incoming data points, making it suitable for massive datasets that do not fit entirely in memory. Despite its efficiency, the standard BIRCH algorithm may not fully capture the intricacies of Tamil text data, especially when dealing with varying document lengths, rich morphology, and the presence of context-dependent stopwords [11].

This research paper proposes an enhanced BIRCH algorithm tailored for Tamil text data clustering. The key innovations in the proposed approach include the integration of dynamic stopword identification, an optimal threshold selection mechanism, and the combination of diverse feature representations using TF-IDF, Latent Dirichlet Allocation (LDA) [3], and Bidirectional Encoder Representations from Transformers (BERT) [4]. Dynamic stopword identification ensures that context-specific and domain-relevant stopwords are effectively removed, thereby improving the quality of feature extraction [6]. The optimal threshold selection mechanism aims to refine the clustering process by adaptively determining the threshold that yields the most coherent and distinct clusters The fusion of TF-IDF, LDA, and BERT features captures lexical, topic-level, and contextual information, providing a comprehensive representation of the Tamil text data. Moreover, the metrics values are assessed for cluster sizes of 5,10,15 and 20 and a comparison is done by analysing the metrics values with and without optimal threshold value.

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

2. RELATED WORK

Clustering is the process of classifying objects into different groups, or partitioning a data set into subsets or clusters. Clustering classifies data objects without consulting a known class label. Cluster analysis is a data analysis tool which aims at sorting different objects into groups in a way that the degree of association between two objects is maximal if they belong to the same group and minimal otherwise.

The hierarchical method creates a hierarchical decomposition of the given set of data objects. A hierarchical method can be classified such as agglomerative or divisive, based on the way hierarchical decomposition is formed. The agglomerative approach has two approaches, namely bottom-up and top-down. The bottom-up approach takes place as follows; each object forms a separate group initially. Consistently combine objects or groups that are close to each other, until all groups are combined into one (the top level of the hierarchy), or until the condition of the termination of the relationship occurs. While the top-down approach, begins with all objects in the same cluster divided into smaller groups, until each object in a cluster finally or until a condition of termination occurs. The hierarchical method contains the fact that after the merger or split step is carried out, the divisive process cannot be cancelled. There are two approaches to improve the quality of hierarchical grouping: (1) carry out careful analysis of the "linkage" object on each partition hierarchy, as in Chameleon, or (2) integrates hierarchical agglomeration and other approaches by first using the agglomerative hierarchy of group object algorithms into micro clusters, and then macro clustering micro clusters using other grouping methods such as repeated relocation. One algorithm that belongs to the hierarchical method is BIRCH (Balanced Iterative Reducing and Clustering Using Hierarchies). BIRCH is an integrated hierarchy grouping algorithm. BIRCH introduces two concepts, clustering features and clustering feature trees (CF trees), which are used to describe cluster summaries [12].

BIRCH uses two main data structures to represent the clusters: Clustering Feature (CF) and Sub-Cluster Feature (SCF). CF is used to summarize the statistical properties of a set of data points, while SCF is used to represent the structure of sub clusters. BIRCH attempts to minimize the memory requirements of large datasets by summarizing the information contained in dense regions or clusters as Clustering Feature (CF) entries. A CF tree is a height-balanced tree with two parameters, branching factor and threshold. The CF-tree is a very compact representation of the dataset because each entry in a leaf node is not a single data point but a subcluster. Every entry in a CF tree contains a pointer to a child node and a CF entry made up of the sum of CF entries in the child nodes. The tree size is a function of the threshold. The larger the threshold is, smaller the tree is.

There are three parameters in this algorithm, which needs to be tuned.

1. **threshold** : threshold is the maximum number of data points a sub-cluster in the leaf node of the CF tree can hold.
2. **branching factor** : This parameter specifies the maximum number of CF sub-clusters in each node (internal node).
3. **number of clusters** : The number of clusters to be returned after the entire BIRCH algorithm is complete i.e., number of clusters after the final clustering step. If set to None, the final clustering step is not performed and intermediate clusters are returned.

BIRCH provides a clustering method for very large datasets. It finds a good clustering with a single scan and improves the quality with a few additional scans.

A parameter variant of BIRCH, A-BIRCH states that choosing the correct parameters for clustering algorithm is often difficult as it requires information about the dataset, which is often not available. This is also true for BIRCH, which requires the cluster count k as well as a threshold T in order to compute the clusters correctly. For this reason, they removed the global clustering phase, thus rendering the cluster count parameter k unnecessary, and proposed a method that automatically estimates the threshold T, which is achieved using Gap Statistic to determine cluster properties. The evaluation proved the applicability of their approach in a very robust manner for two-dimensional isotropic Gaussian distributions with roughly the same variance, regardless of the number of clusters or it’s elements [7].

The authors Fanny Ramadhani, Muhammad Zarlis and Saib Suwilor in their paper proposed a solution to this deficiency in general BIRCH algorithm by modifying the Threshold value to dynamic so that it can produce good cluster quality and be validated using silhouette coefficient (SC). There is a very clear difference between the standard BIRCH algorithm and the BIRCH algorithm on the modified T parameter (BIRCH (CF-Leaf (modif)). The CF-Node result, the total CF-Entries and Total CF-Leaf Entries produced 60% less than CF-Node, the total CF-Entries and Total CF-Leaf Entries in the standard BIRCH algorithm. Modified BIRCH can make clusters more accurate and better can be validated with the SC method. So that it can be concluded that the BIRCH algorithm on the modified T parameters results in a much better cluster quality compared to the standard BIRCH algorithm [8].

3. ALGORITHM

The eventual aspect of the enhanced BIRCH algorithm is to yield the optimal metric values for the fine clusters. The enhanced BIRCH algorithm is added with two new features such as the Dynamic Stopword identification and automatically determining the optimal threshold value for BIRCH Clustering.

ENHANCED BIRCH ALGORITHM:

```
1. Load and Preprocess Data
      •  Read the CSV file.
      •  Extract 'News' and 'TCategory' columns.
      •  Clean and tokenize the text (remove digits, special characters).

2. Identify Dynamic Stopwords
    Frequency-based stopwords:
    Build a token frequency dictionary.
    FOR each token:
    IF frequency < min_freq OR frequency > max_freq:
    Add token to frequency-based stopwords list.

▪  Endif
▪ END FOR
    TF-IDF-based stopwords:
    Compute TF-IDF scores for all tokens.
    FOR each token:
    IF average TF-IDF < tfidf_threshold:
    Add token to TF-IDF-based stopwords list.

▪  Endif
▪  END FOR
•       stopwords list = Union of frequency-based + TF-IDF-based stopwords.

3. Remove Stopwords
Remove tokens found in the final stopwords list.

4. Feature Extraction
•       TF-IDF features: Compute TF-IDF vectors.
•       LDA topic vectors: Train LDA model, extract topic distributions.
•       BERT embeddings: Generate BERT vector representations.
```

5. Combine Features
    - Concatenate TF-IDF vector + LDA vector + BERT vector.

6. Dimensionality Reduction (UMAP)
    - Apply UMAP to reduce feature vectors to lower dimensions (e.g., 50D).

7. Find Optimal BIRCH Threshold
- Initialize best_score = -1.
- FOR threshold T from min_thresh to max_thresh:
    - Cluster using BIRCH with threshold T.
    - Compute Silhouette Score.
    - IF score > best_score:

Update best_score and best_thresh.

END FOR

8. Final Clustering
- Run BIRCH with best_thresh.
- Assign cluster labels.

The enhanced BIRCH algorithm is considered both hierarchical and partition-based because it incorporates the characteristics of both clustering approaches during its construction and clustering phases.

The Hierarchical Nature of the enhanced BIRCH algorithm is done through CF Tree Construction. The Incremental Hierarchical Structure is given below,

- BIRCH builds a CF (Clustering Feature) tree, where each node represents a cluster at various levels of granularity.
- If a new data point fits an existing cluster (based on the threshold), it is absorbed into that cluster.
- Else a new cluster is created or existing nodes are split, leading to a hierarchy of clusters.

The Hierarchical structure has a Top-Down Approach. Here the CF tree starts with a root node and grows by adding child nodes (clusters). The non-leaf nodes summarize subclusters, while leaf nodes hold fine-grained clusters. This structure naturally creates a hierarchy where the Higher levels represent broader clusters and the Lower levels show finer divisions. The CF tree is built using dynamic stopword-filtered TF-IDF, LDA, and BERT features. Clusters are formed at different hierarchical levels based on the threshold and branching factors.

The Partition-Based Nature is done through the Global Clustering on Leaf Nodes. Once after the CF tree is built, partition-based clustering (e.g., K-Means or directly BIRCH's centroid-based approach) is applied to the leaf nodes. This step divides data into distinct partitions based on the final cluster centroids. Each data point is assigned to one and only one cluster, ensuring clear partitions. Clusters are formed by minimizing intra-cluster distance (similar to K-Means) and the leaf nodes act as representatives (centroids) for partitioning the dataset. Final clustering is performed by partitioning the leaf nodes into a fixed number of clusters. Each document is assigned to a specific partition based on combined features.

The enhanced BIRCH algorithm for clustering Tamil text data uses combined TF-IDF, LDA, and BERT features, the CF (Clustering Feature) Tree plays a vital role in handling large datasets efficiently.

A CF Tree is a height-balanced tree designed to summarize the dataset for incremental and hierarchical clustering. It consists of Non-leaf nodes which contains summaries (CF entries) of their child nodes. The Leaf nodes: Contain CF entries representing clusters and are linked for quick traversal.

Each cluster is represented by a Clustering Feature (CF) triple:

CF= (N, LS, SS)

Where:

N: Number of data points in the cluster

$LS=\sum_{i=1}^{N} x_I$ Linear sum of data points

$SS=\sum_{i}^{N}{}_{=1} x_i^2$ Squared sum of data points

From these, the enhanced BIRCH algorithm computes the following,

- **Centroid:** $\mu = \frac{LS}{N}$
- **Radius:** $R = \sqrt{\frac{SS}{N} - (\frac{LS}{N})^2}$
- **Diameter:** Based on pairwise distances within the cluster

The working of the CF Tree in the enhanced BIRCH algorithm is given below,

1. **Insertion:**

Each feature vector (combined from TF-IDF, LDA, BERT) is inserted into the CF tree. The algorithm finds the closest leaf node using a distance metric which is the cosine distance in the enhanced BIRCH algorithm.

2. **Threshold (T) Influence:**
   - If adding a new point to a leaf node keeps the cluster diameter ≤ threshold (T), it updates the CF triple.
   - Otherwise, it splits the leaf node, creating new CF entries.

3. **Growth Control:**
   - The CF tree grows adaptively, and splits occur when the number of CF entries exceeds a branching factor.

4. **Final Clustering:**
   - Once the tree is built, global clustering is performed on the leaf nodes' centroids using BIRCH’s final pass.

The Combined features significantly improve clustering quality, reflected by higher Silhouette & CHI scores, lower DBI, and clearer cluster plots.

4. EXPERIMENTAL ANALYSIS

To experiment the enhanced BIRCH algorithm, the Tamil News dataset has been used. The dataset contains four columns and two columns namely the News (which contains Tamil News) and TCategory(which denotes the Tamil news category) have been taken. The dataset for experimentation has been downloaded from Kaggle. The enhanced BIRCH algorithm is implemented in Python. The following libraries have been used,

- **Data Handling:** pandas, numpy
- **Clustering:** Birch from sklearn.cluster
- **Text Processing:** TfidfVectorizer, indic_tokenize, re (regex for cleaning)
- **Embedding:** SentenceTransformer for BERT-based embeddings
- **Dimensionality Reduction:** umap, PCA
- **Topic Modeling:** LdaModel, Dictionary from gensim
- **Evaluation Metrics:** Silhouette, Davies-Bouldin, and Calinski-Harabasz scores
- **Visualization:** seaborn, matplotlib

First it loads a CSV file containing Tamil news data into a DataFrame and extracts the News and TCategory columns into lists. Finally, it Combines both columns into a single list for further processing.

The Text Cleaning removes digits, punctuation, and special characters using regex and the Tokenization process uses indic_tokenize for language-specific tokenization (Tamil).

In the Dynamic Stopword Identification first the dynamic stop words are identified by using the frequency based method and the TF-IDF based then Converts tokenized documents into TF-IDF vectors. Next it Calculates the average TF-IDF score for each word across all documents.The Words with low average TF-IDF (< 0.001) are considered dynamic stopwords since they are too common or not informative. Finally the frequency based dynamic stop words and TF- IDF based dynamic stop words are concatenated to get the final dynamic stop words. The total dynamic stop words obtained finally is 871.The dynamically identified stopwords from each document are filtered out and it Produces cleaner documents for topic modeling and clustering.

After the stopword removal process the LDA Topic Modeling process starts and the work is given below,

- Dictionary: Maps unique words to IDs.
- Corpus: Converts documents into a Bag-of-Words (BoW) format.
- LDA Model: Extracts 7 latent topics using the BoW corpus.
- Topic Distribution: For each document, extracts a probability distribution over the topics.
- Output: lda_features - A matrix where each row represents a document’s topic distribution.

In the Sentence Transformer Embeddings first it Loads a pre-trained multilingual model that supports Tamil and Converts documents into dense vector embeddings using the transformer. Finally, it normalizes the embeddings to have unit length. The normalized BERT embeddings are concatenated with LDA topic distributions and this fusion provides both semantic (BERT) and topical (LDA) information for clustering.

In UMAP Dimensionality Reduction (UMAP) it Reduces high-dimensional features to 50 dimensions using UMAP for better clustering performance and the cosine metric is used since it's suitable for textual embeddings.

For obtaining the optimal threshold value the Adaptive Threshold tuning is done by using the Silhouette Score. It Searches for the optimal enhanced BIRCH threshold between 0.1 and 1.0. Then it evaluates clustering quality using the Silhouette Score. Finally, it returns the threshold that yields the best clustering structure.

The enhanced BIRCH algorithm is applied with the optimized threshold to cluster the reduced embeddings and the labels contains the cluster assignment for each document.

The metrics values are printed after evaluating it. The three metrics that have been used are Silhouette Score, Davies-Bouldin Index (DBI) and Calinski-Harabasz Index (CHI**).** The metric values are analyzed with four cluster sizes.

The enhanced BIRCH algorithm is experimented with optimal threshold value and without optimal threshold value. Moreover, to prove the effectiveness of the cluster accuracy the enhanced BIRCH algorithm is compared with the standard BIRCH algorithm with static stop words. The metric values obtained through the standard BIRCH algorithm produces very low values which is not in the acceptable range. The Enhanced BIRCH algorithm with and without optimal threshold values both produces better results compared to the existing standard BIRCH algorithm. Comparing the metric values with and without optimal threshold values the enhanced BIRCH algorithm with optimal threshold values yields better results.
The following table lists the difference in metric values with and without the optimal threshold values.

Cluster Size	**Metric**	**BIRCH algorithm with static stop words and without optimal threshold**	**Proposed BIRCH Algorithm Without Optimal Threshold Value**	**Enhanced BIRCH Algorithm with Optimal Threshold Value**
5	Silhouette Score	0.1524	0.537	0.825
	Davies-Bouldin Index	2.8229	0.939	0.472
	Calinski-Harabasz Score	589.3391	6607.816	9635.012

Cluster Size	Metric	BIRCH algorithm with static stop words and without optimal threshold	Proposed BIRCH Algorithm Without Optimal Threshold Value	Enhanced BIRCH Algorithm with Optimal Threshold Value
10	Silhouette Score	0.1826	0.854	0.866
	Davies-Bouldin Index	3.6438	0.397	0.327
	Calinski-Harabasz Score	429.4187	58318.761	127656.611
15	Silhouette Score	0.1733	0.791	0.824
	Davies-Bouldin Index	3.6756	0.359	0.480
	Calinski-Harabasz Score	292.8037	158216.103	202024.900
20	Silhouette Score	0.1703	0.758	0.812
	Davies-Bouldin Index	3.7404	0.462	0.555
	Calinski-Harabasz Score	223.9065	83589.297	222929.567

Table 4.1 Metrics Values of Standard BIRCH algorithm, Enhanced BIRCH Algorithm with and Without Optimal Threshold Value.

Cluster size 10 gives the best performance with the highest Silhouette Score (0.866), lowest Davies-Bouldin Index (0.327), and a significantly high Calinski-Harabasz Score (127656.611). Cluster size 15 and 20 show decreasing silhouette scores and increasing Davies-Bouldin Index, indicating reduced cluster compactness and separation. Optimal threshold 0.9 suggests that this enhanced BIRCH algorithm with optimal threshold value provides the best clustering quality.

The figures represent the three metric values for the Standard BIRCH with static stop words and the enhanced BIRCH algorithm with and without optimal threshold value.

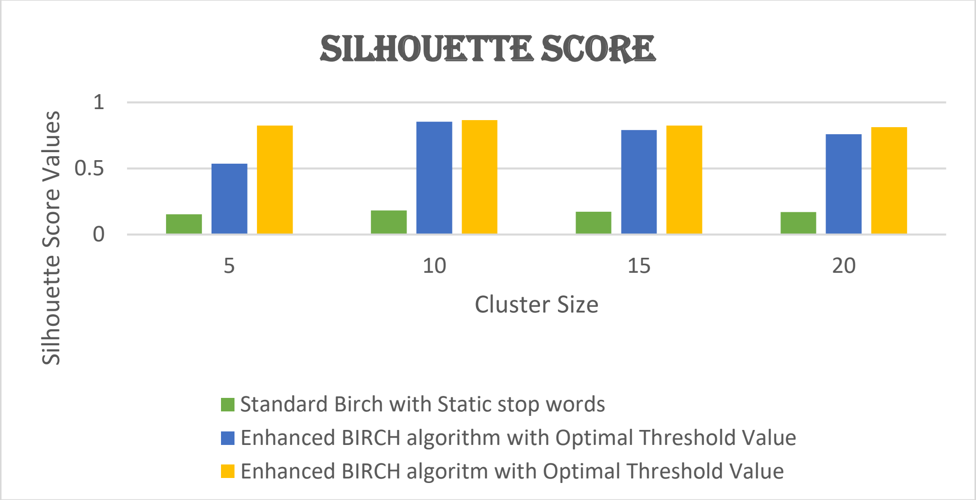

Figure 4.1 Silhouette Scores of Standard BIRCH with static stop words, Enhanced BIRCH algorithm with and without Optimal Threshold Value

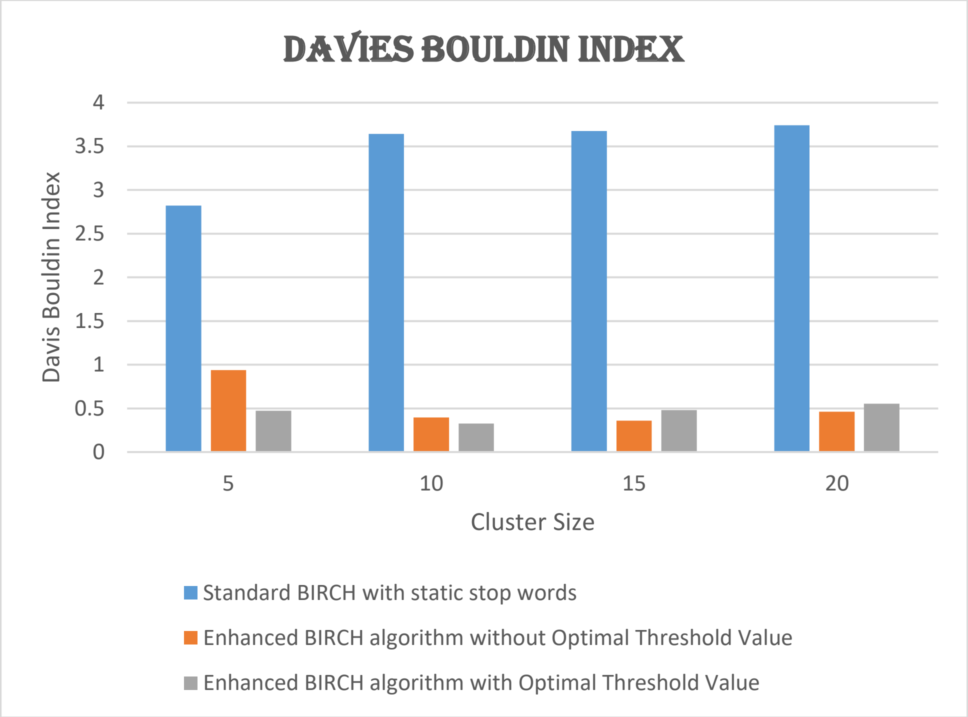

Figure 4.2 Davies Bouldin Index of Standard BIRCH with static stop words, Enhanced BIRCH algorithm with and without Optimal Threshold Value

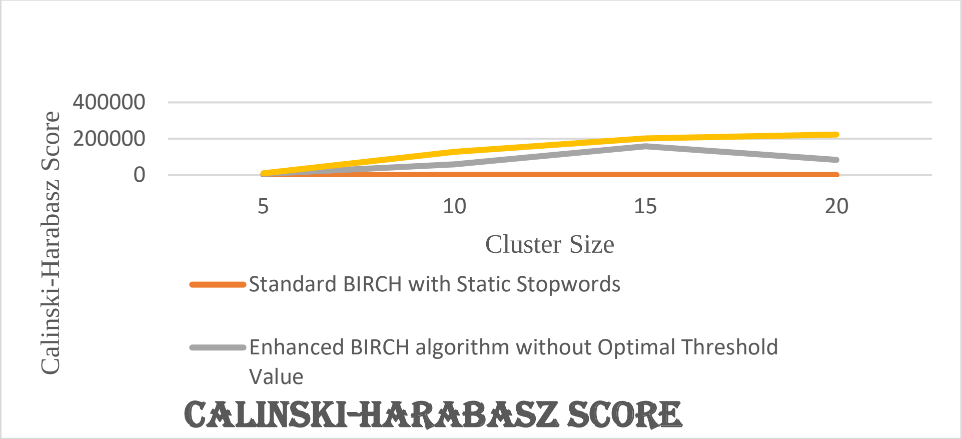

Figure 4.3 Calinski Harabasaz Score of Standard BIRCH with static stop words, Enhanced BIRCH algorithm with and without Optimal Threshold Value.

The Combined features (TF-IDF + LDA + BERT) enhances cluster cohesion by ensuring points within a cluster share similar term importance, topics, and semantic context. The enhanced BIRCH algorithms yield high cohesion which means the data points in the same cluster share similar Tamil text context and topics. The Silhouette score in the enhanced BIRCH algorithm indirectly reflects cohesion which means if score is high then it yields to better cohesion.

4.1 Novel Aspects of the Enhanced BIRCH Algorithm with Optimal Threshold Value:

1. **Automatic Threshold Optimization**
   - The Enhanced BIRCH algorithm uses a Silhouette Score–based threshold tuning mechanism that systematically selects the optimal BIRCH threshold value, eliminating the need for manual tuning or reliance on pre-clustering.
2. **Semantic Feature Fusion for Text Clustering**
   - The enhanced BIRCH algorithm combines TF-IDF, LDA topic distributions, and Sentence-BERT embeddings to create a rich hybrid feature space, enabling more accurate and meaningful clustering of unstructured multilingual text data.
3. **Dynamic Stopword Identification**
   - Our algorithm implements a data-driven stopword filtering approach based on frequency-based approach and average TF-IDF scores which allows removal of low-information tokens specific to the dataset, especially beneficial for resource-limited languages like Tamil.
4. **UMAP-Based Dimensionality Reduction**
   - In this algorithm UMAP is used with cosine distance to preserve semantic neighborhood structure in high-dimensional space before clustering, enhancing BIRCH’s ability to form compact and distinct clusters.
5. **Text-Specific Adaptability**

- The proposed algorithm is tailored for multilingual and low-resource text datasets (e.g., Tamil-English news), which traditional BIRCH and its variants are not inherently designed to handle.

6. End-to-End Workflow Integration

- Our proposed algorithm provides a unified framework from text preprocessing to feature extraction, adaptive clustering, and evaluation—facilitating reproducibility and application to other languages or domains.

5. CONCLUSION

The enhanced BIRCH algorithm, which is experimented with dynamic stopword identification and an optimal threshold value, significantly improves clustering quality for Tamil text data. The dynamic stop word approach ensures that only meaningful words contribute to clustering, reducing noise and enhancing topic coherence. Optimizing the threshold value balances cluster compactness and separation, leading to more accurate and interpretable clusters. The enhanced BIRCH algorithm with optimal threshold value outperforms traditional methods and it is a robust choice for large-scale Tamil text clustering. The optimal threshold (0.9) significantly improves clustering for smaller cluster sizes (5 and 10), leading to better cohesion and separation. However, for larger cluster sizes (15 and 20), it may reduce cluster compactness, which is observed through the higher Davies-Bouldin Index values.

REFERENCES:

1. Aggarwal, & Zhai, C., “Mining text data”. Springer Science & Business Media,2012.
2. Balakrishnan, Srinivasan, & Raman, S. . "Clustering methods for Tamil text documents." Journal of Computer Science, 175-184,2016.
3. David M. Blei, Andrew Y. Ng, Michael I. Jordan, "Latent Dirichlet Allocation." Journal of Machine Learning Research, 3, 993-1022,2003.
4. D.Jacob, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding." NAACL-HLT,2019.
5. Endres D.M and J. E. Schindelin, "A new metric for probability distributions." IEEE Transactions on Information Theory, 1858-1860,2003.
6. Kumar. S., & Devi. K, "Challenges in Tamil NLP: A comprehensive survey." International Journal of Computational Linguistics, 90-105,2019.
7. Lorbeer, Boris & Kosareva, Ana & Deva, Bersant & Softić, Dženan & Ruppel, Peter & Küpper, Axel. A-BIRCH: Automatic Threshold Estimation for the BIRCH Clustering Algorithm. 169-178, 2017.
8. Ramadhani, Fanny & Zarlis, Muhammad & Suwilo, Saib. “ Improve BIRCH algorithm for big data clustering”, IOP Conference Series: Materials Science and Engineering. 725. 2020.
9. Ramos, Juan, "Using TF-IDF to determine word relevance in document queries." Proceedings of the First Instructional Conference on Machine Learning, 242, 133-142,2003.
10. Röder, Michael & Both, Andreas & Hinneburg, Alexander., "Exploring the space of topic coherence measures." WSDM '15: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, 399-408, 2015.
11. Sundararaja.R & Raman.S, "Text clustering for Tamil corpus using improved hierarchical methods." Asian Journal of Information Technology, 19(1), 10-18,2020.
12. Suganya R, Pavithra M and Nandhini P , “ Algorithms and challenges in big data clustering”, International Journal of Engineering and Techniques, 40- 47,2018.

13. Tian Zhang, Raghu Ramakrishnan, &Miron Livny, "BIRCH: An efficient data clustering method for very large databases." SIGMOD '96 Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, 103-114,1996.