International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Development and Evaluation of Extended Text Pre-processing Techniques for Hindi Document Clustering.

[1]Mukta M. Deshpande , [2]Dr. Prafulla B. Bafna

Submitted: 04/09/2023 **Revised**: 22/10/2023 **Accepted**: 05/11/2023

Abstract: Data pre-processing, which involves cleaning and converting raw text data into an appropriate format for analysis, is a vital stage in text analytics. Clustering is a widely used technique in text analytics for grouping similar data points. However, the pre-processing techniques applied to the data can greatly influence the quality and effectiveness of clustering results. The goal of this study is to examine how the pre-processing methods that has been suggested affects clustering algorithm performance. Several distinct combinations of pre-processing methods have been applied to produce document clustering. The goal was to identify the optimal pre-processing combination that produces the most accurate and meaningful clusters. The effects of the clustering technique are assessed after applying the Normalized Mutual Information (NMI), silhouette score, and Adjusted Rand Index (ARI). Principal Component Analysis (PCA) and dendrograms are two visualization techniques explored in this study to gain insights into the clustering results. The findings from this study can help enhance our understanding of the pre-processing techniques required in the clustering process and help researchers and practitioners implement clustering algorithms to achieve greater accuracy.

Keywords*: Pre-processing, feature extraction, Tokenization, Stopwords, Lemmatization, Hindi Document Clustering*

1. Introduction:

In many fields, including text mining and information retrieval, text analytics is essential. Text data often contains noise in the form of punctuation, special characters, HTML tags, URLs, and other irrelevant information [1]. Pre-processing methods such as stripping HTML tags and deleting special characters and stop words can help minimize noise and focus on important content. It improves the accuracy and effectiveness of text analytics algorithms and enables meaningful extraction of features and insights from the text corpus [2,3]. The most common methods for text pre-processing comprise tokenizing text, removing Stop words, stemming and lemmatization, special characters removal, and case conversion [4,5]. Eliminating stop-words helps to get rid of frequently used words that have no meaning in a text. These words are mostly articles, prepositions, and conjunctions. Lemmatization is another pre-processing step to break down the word to its basic form [4-8]. The quality of the pre-processing methods used in the data has a significant impact on how clustering algorithms perform. Eliminating words that are often used but are less relevant minimizes the dimensionality of the data. The effectiveness and computational performance of clustering algorithms can be enhanced by this pre-processing, making them more scalable and manageable, especially for large datasets [10-15]. There are many opportunities for text analytics in Indian languages due to the availability of various forms of domain-specific data. In the Hindi language, some words like “रह” (Rah) meaning “stay”, “तु” (Tu) meaning “you”, “मे” (Mai) meaning “I”, “वो” (Voh) meaning “they”, “थे” (The) meaning “were”, “था” (Tha) meaning “was”, and “वह” (Vah) meaning “he” are examples of stop words. These phrases are commonly used in the corpus and have no particular significance. Removing stop words reduces corpus size, thereby saving CPU cycles and memory, which ultimately reduces model training time [7]. After the stop words are removed, the next step is lemmatization, which is carried out to bring the word to its base form [9]. For instance, the word “लेने” (lene) meaning “pick up” is converted to its base form “ले” (le) meaning “take” after performing lemmatization on the corpus. In the process of pre-processing Hindi data, Python libraries often ignore Hindi stop words that are not included in their vocabulary. However, identifying custom stop words and lemmatized words can further reduce the size of the data. In this research, we performed document clustering on the Hindi BBC news article dataset[1] with three clustering methods: Gaussian mixture models, Hierarchical Agglomerative Clustering (HAC) and K-means. To ensure that our pre-processing aligns with the goals and requirements of our analysis, we manually select custom stop words. This allows us to consider the context and significance of certain words that may not be captured by basic available stop word list. We use python’s the Indicnlp library for tokenization and the StanfordNLP library for lemmatization. Additionally, we utilize a basic stopword list that includes stop words,

[1]*Research Scholar Symbiosis Institute of Computer Studies and Research Pune, India*
Mukta_deshpande12@yahoo.com
[2]*Assistant Professor, Symbiosis Institute of Computer Studies and Research Pune, India*
prafulla.bafna@sicsr.ac.in

special characters, digits, and punctuation in Hindi. This list is downloaded from the internet[2].

The summary of the literature review performed for this investigation is shown in Table 1. These works were done to solve text summarization, document clustering, and text classification problems in natural language processing. Data pre-processing was done for these experiments.

[1] https://github.com/goru001/nlp-for-hindi

[2] https://github.com/stopwords-iso/stopwords-hi/blob/master/stopwords-hi.txt

Table 1. Summary of Literature Review

Sr.no	**Citation**	**Techniques**	**Pre-processing**	**Algorithms**	**Data Set**	**Results**
1	Kumar, S., & Singh, T. D. (2022)	Hindi fake news Classification	Tokenization Stemming, Stop Words Removal	Logistic regression, Naive Bayes, LSTM	Hindi Fake and True Dataset	LSTM gives best result for hindi news classification
2	Chang, I., et.al (2021)	Topic modelling, LDA, Clustering	Tokenization, Stop word removal lemmatization, Lower case	KMeans, LDA	SSCI journals abstract	The K-means method was better than the LDA
3	Rani, R., & Lobiyal, D. K. (2018)	Automatic generic stopword list generation	Tokenization Stop word removal lemmatization	Algorithm for generic Stopword list	articles published in NaiDunia	Generic Stop word lists are identified
4	Verma, P., & Verma, A. (2020)	Text Summarization	Tokenization Stop word removal lemmatization, Lower case	NA	NA	Lack of Indian context text summarization due to the unavailability of tools
5	Bafna, P. B., & Saini, J. R. (2020).	Document Term matrix for Marathi (DTMM)	Tokenization Stop word removal lemmatization	Cosine similarity	Sample verses and Poem of Marathi language	Similarity between documents is calculated using DTMM
6	Bafna, P. B., & Saini, J. R. (2020).	Application of Zips law	Tokenization Stop word removal lemmatization, Stemming	Zip's Law, BaSa	Hindi and Marathi poems and stories	BaSa, a context-based term extraction technique provides better results
7	Raulji, J. K., & Saini, J. R. (2016)	Stop word removal algorithm for Sanskrit language	Tokenization Stop word removal	Design of Sanskrit stop word removal algorithm	Sample Sanskrit text	Sanskrit stop words removed from sample text
8	Ladani, D. J., & Desai, N. P. (2020)	Survey on Stop word	N/A	NA	N/A	Significantly less research is conducted on non-English languages

9	Nandathilaka, M., Ahangama, S., & Weerasuriya, G. T. (2018)	Development of lemmatizer for Sinhala language	N/A	Development of Lemmatization algorithm	Sample Sinhala social media posts	Rule based Sinhala lemmatizer implemented
10	Jacksi, K et.al (2020)	Document clustering	Tokenization Stop word removal	KMeans, HAC	100 Movies IMDB & WIKI	KMeans algorithm has high score.
11	K. R. Shahapure and C. Nicholas 2020	Cluster Analysis	N/A	k-means	S-1 data set	The silhouette score was calculated for many values of k .
12	Seki, K., Ortiz, M. S., & Mostafa, J. (2019)	Evaluation of Effectiveness and efficiency of clustering algorithms	N/A	Mini Batch k-means, SVD NMF Spectral	OHSUMED data Set	Mini batch KMeans found best for cluster quality
13	Sen, M. Pandey and K. Chakravarty, (2020)	Proposed algorithm for cluster quality improvement	Tokenization Stop word removal, Stemming Lemmatization	K means Genetic algorithm Topic modelling	Speech data of Narendra modiji	Proposed method improves cluster quality
14	Younas, F et.al (2021)	word semantic measure for Hindi	NA	word semantic synonym	HindMonoCorp	A word semantic measure Developed for Hindi
15	Mehta, V. et.al (2021)	Text clustering for large dataset	NA	Word Embedding	k-means, Agglomerative, clustering	Word embedding based proposed clustering gives improvement.

Hindi news articles are collected from various sources and analyzed using pre-processing and feature extraction methods for classification problem [1]. Data pre-processing steps are performed based on research requirements. Using ten-year samples of abstracts, titles, and keywords [2] employs the text mining approach to look at various research topics from the Web of Science (WoS) environmental education research articles database. Stop word removal helps to enhance efficient corpus indexing and improves the functionality of information retrieval systems. [3] Constructs Hindi language's automatic generic stop word list. A review of text summarizing techniques for Indian languages was conducted. Languages including Hindi, Panjabi, Bengali, Marathi, Tamil, and Kannada were used to conduct this survey [4]. The DTMM corpus was introduced to convert unstructured data into a tabular format for the Marathi language. However, it does not consider the meanings of phrases, which creates a problem with synsets. To address this issue, the DSMM corpus was proposed, which compiles synonyms of words for Marathi language [5]. These synsets aid in reducing dimension. Dimension reduction and the polysemy issue have been addressed using the proposed DSMM approach. BaSa is a method for extracting terms from the context created by [6]. Compared to Zip's law, it produces superior results. A simple technique to design stop-word removal for the Sanskrit language is proposed by [7]. The dictionary-based method is applied and implemented to remove Sanskrit stop words. Lemmatizing is a difficult process where the base or root of terms is derived. Sinhala is the official language of Sri Lanka. [8] performs surveys on various stop word list generation methods in different languages.[9] developed rule-based Sinhala lemmatizer. A semantic clustering algorithm has been suggested and applied by [10] for the movie datasets obtained from Wikipedia and IMDB. One of the essential steps in data mining is clustering. Choosing the ideal number of clusters is challenging. [11] uses the silhouette score to do a cluster analysis.[12] The effectiveness and efficiency of grouping biomedical documents using several clustering algorithms and metrics, such as AMI (Adjusted mutual index). [13] presented a new technique called the Genetic algorithm to enhance the quality of the k-means cluster which will be helpful for topic modeling and document clustering. A semantic word measure for the Hindi language based on artificial intelligence has been developed by [14]. The word-embedding-based text clustering approach is proposed by [15], and their proposed technique shows improvement on the traditional approach.

2. Methodology

The internet has transformed the way we access information. Nowadays, we have access to a wealth of textual data from social media, blogs, and articles. However, to retrieve information quickly and accurately, it must be sorted into categories. Natural Language Processing, which is aided by machine learning algorithms, is a valuable tool for extracting essential and actionable data from text [4-9]. Since computers and algorithms cannot understand text or characters, it’s crucial to transform the text into a machine-readable representation, such integers or binary code, before analyzing it. Available stop-word lists may not always be adequate for capturing all the stop words and noise from dataset. In such cases, selection of custom stop words may be necessary. For this research work Custom Stop word list is created to include commonly used words that appear in this dataset. Using custom stop words helps to filter out noise, which ultimately improves the quality of analysis and clustering results.

2 Steps in Research methodology

Following Figure 1. describes the steps in the research methodology.

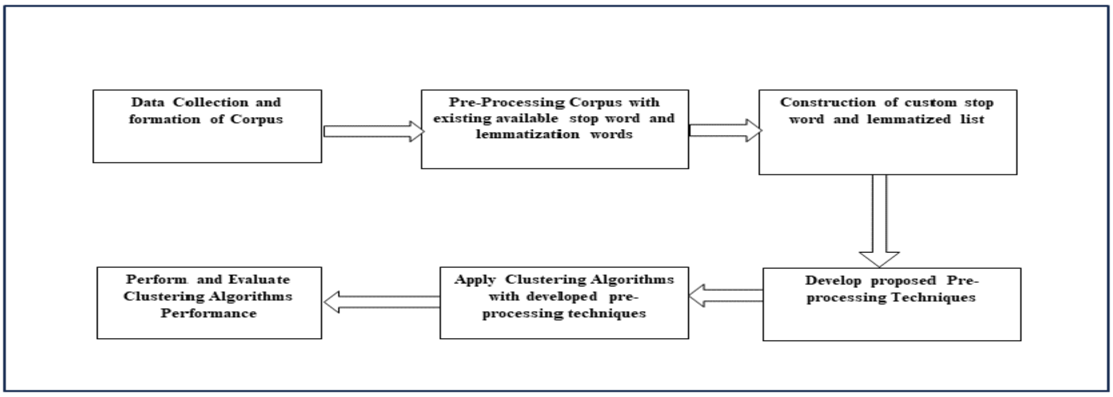

Data Collection and formation of Corpus
Pre-Processing Corpus with existing available stop word and lemmatization words
Construction of custom stop word and lemmatized list
Develop proposed Pre-processing Techniques
Apply Clustering Algorithms with developed pre-processing techniques
Perform and Evaluate Clustering Algorithms Performance

Fig 1. Steps in research methodology

This section covers the collection, pre-processing, and feature representation of Hindi BBC news data for document clustering.

2.1 Corpus Collection Process

The Hindi BBC news dataset was collected from the internet[1] to perform this study. News pertaining to many categories can be found in this dataset. There are 14 different news categories, including India, international, national, sports, and entertainment. There are 3467 records in the entire dataset.

2.2 Data Pre-processing

It takes cleaning and transformation to make unstructured text data usable for analysis. Stop-word removal, lemmatization, stemming, and tokenization are all covered. The following steps were taken to ensure high-quality input data:

- **Tokenization**:

In natural language processing, tokenization is the initial step in every pipeline. It involves dividing unstructured data into smaller, more manageable pieces, or tokens, that can be analyzed separately. These tokens can then be used to represent the document as a vector.

• **Stop word Removal**:

Stop words are eliminated with available stop word list from github[2].

- **Lemmatization:**

Lemmatization is a practice used in text analytics to normalize words to their base form. For this study, basic lemmatization was performed using the StanfordNLP library for Python.

2.3 Construction of custom domain stop word list and custom domain lemmatization list

Many common words vary across domains and have no significance within a particular domain. To improve text mining operations such common phrases should be removed from corpora. Therefore, for the news domain, we identified new common prepositions, conjunctions, and pronouns in the Hindi BBC dataset and manually selected them to create a custom stop word list. We manually chose these terms with the help of a Hindi language expert. In the Hindi BBC news dataset, we discovered an additional 336 custom stop words. After lemmatizing the dataset using the Python StanfordNLP library, we found that only a few words were left for which manual retrieval of the lemma was necessary. We retrieved the lemma manually from the newly identified words. Our custom domain lemmatization list now includes 251 additional words.

2.4 Clustering Algorithms implementation and performance evaluation

In order to cluster the Hindi BBC data set, K-means, Gaussian Mixture Models, and Hierarchical Clustering algorithm were utilized. The Python Sci-kit-learn library was used for both tf-idf feature extraction and algorithm execution. Experiments were carried out using four defined techniques to form clusters.

2.4.1 Clustering Algorithms implementation

The focus of this research paper is on three commonly used clustering techniques: Hierarchical Agglomerative Clustering (HAC), K-Means and Gaussian Mixture Model (GMM). K-Means is a powerful yet simple algorithm that partitions data into K clusters by iteratively optimizing cluster centroids. On the other hand, HAC constructs a hierarchical representation of data by forming clusters through a series of agglomerations or divisive steps. In contrast, GMM approaches clustering probabilistically by modeling data as a mixture of Gaussian distributions, providing more flexible and nuanced clustering solutions.

2.4.2 Feature extraction using tf-idf

In text analytics, the tf-idf (term frequency-inverse document frequency) technique is frequently used to extract features from text data and numerically express them. It establishes a term's relative importance within a corpus of documents. In order to achieve this, tf-idf uses two metrics: inverse document frequencies and term frequencies. By counting the number of times, a word appears in a document and dividing that total by the number of words in that document, term frequency (tf) is determined. It can be modeled mathematically as follows:

tf (t, d) = t in d count / word count in d --------------------------------- (1)

idf can be modeled mathematically as follows:

idf (t, d) = log (number of documents in total / number of documents containing the phrase t) - (2)

The product of these two metrics (tf * idf) gives the tf-idf count of a term in a document. Higher count signifies term importance.

2.4.3 Clustering Algorithm Implementation with developed data pre-processing techniques

In this research, three clustering algorithms are implemented along with the use of four developed Data pre-processing techniques. For this research work, we have developed four techniques for data pre-processing. These techniques are listed below:

- **Existing Technique**: Clustering with basic stop words and basic lemmatization.
- **CBSCL Technique**: Clustering with basic stop words and custom lemmatization.
- **CCSBL Technique**: Clustering with custom stop words and basic lemmatization.
- **CCSCL Technique**: Clustering with custom stop words and custom lemmatization.

Cluster visualization techniques are used to represent the results of various clustering techniques. The purpose behind cluster visualization is to help users understand the structure and patterns within the data. This is achieved by revealing insights into how data points are grouped or separated in a multi-dimensional environment. In this research work, cluster visualization for hierarchical agglomerative clustering is shown with dendrogram plots. Cluster visualization for K-means and GMM clustering algorithms is depicted using PCA (Principal Component Analysis). To provide a novel method for cluster visualization, the top 3 words from each cluster were extracted. These words are appeared most frequently in that particular cluster.

2.5 Clustering algorithms Performance evaluation

Following are the three statistical metrics that are commonly utilized for assessing clustering techniques.

2.5.1 Silhouette score

The primary standard for evaluating cluster quality is the silhouette score. In comparison to other clusters, it measures how well a data point fits into a specific cluster. The value for a silhouette score could vary between -1 and +1. Python Scikit-learn's silhouette score function determines the mean silhouette score for each sample. One can calculate the silhouette coefficient for a sample as:

Silhouette score = (Y-X)/max (X, Y) --- (3)

Where X and Y are mean and nearest cluster distance.

A sample is considered to be in the right cluster if its silhouette score is about +1. A score that is close to 0 indicates that the data point is from another cluster. A score near to -1, on the other hand, indicates that the data point is in the incorrect cluster.

2.5.2 Adjusted Rand Index

The ARI score is a more reliable measure of similarity between two clustering’s.

ARI = (Rand Index – expected Rand Index) / (Max Rand Index – expected Rand Index) ------ (4)

ARI, or Adjusted Rand Index, is used to measure the quality of clustering or to compare two different clusters. By accounting for chance agreement, it evaluates the agreement between the true labels and the expected cluster assignments. ARI considers every pair of data points, and its range is from -1 to 1.

2.5.3 Normalized Mutual Information

It is a frequently used metric to assess the quality of clusters or compare two clusters. Unlike other metrics, NMI is normalized which allows for a more comprehensible comparison between different datasets or clustering results. The NMI value can vary from 0 to 1.

2.5.4 Normalized Mutual Information

NMI is a metric frequently used to evaluate cluster quality or compare two clusters. Unlike other metrics, NMI is normalized, which provides a more comprehensible comparison between different datasets or clustering results. The NMI value ranges from 0 to 1.

NMI (A, B) = 2 * (I; B) / [H(A)+H(B)] ---(5)

where, A is class labels and B is cluster labels

H () is Entropy and I (A; B) is Mutual Information between A and B.

3. Results & Discussion

3.1 Results obtained with pre-processing and Clustering Algorithm Implementation

In Table 2, we demonstrate the pre-processing of a sample sentence using basic and custom stop words, as well as lemmatized words. From the sample sentence, it is observed that tokens "के" (ke) and "है" (hai) are very common terms in Hindi language and are already included in basic stop words, so they are removed during basic stop word removal. The token "भी" (bhi) is an actual stop word that is frequently available in this text and does not carry any significant meaning. Therefore, it was added to our custom stop word list and will be removed with a custom stop word. During lemmatization using the Python StandfordNlP library, the token "पड़ोसी" (padosi), meaning "neighborhood," and "चलती" (chalati), meaning "moving," are changed to their root form "पड़ोस" (pados) and "चल" (chal), respectively. This is basic lemmatization. The token "ड्राइवरलेस" (driverless) is a combined word with "ड्राइवर" (driver) and "लेस" (without), meaning "without the driver." We identified this combined word and converted it into its lemma, "ड्राइवर" (driver), during custom lemmatization. Although "ड्राइवरलेस" is an English word, it is commonly used in Hindi conversations as a Hindi word. In Hindi BBC news dataset, it was available in its Hindi transliterated form therefore, we considered such English words as Hindi words for this research work.

Table 2. Demonstration of pre-processing on sample sentences

Sentence	sentence after basic stop word removal	Sentence after custom stop word removal	Sentence After basic lemmatization	Sentence after custom lemmatization
भारत के पड़ोसी देश चीन में भी ड्राइवरलेस मेट्रो चलती है. Bharat ke padosi desh cheen mai bhi driverless metro chalati hai Driverless metro also runs in India’s neighboring country China	‘भारत’,‘पड़ोसी’,’देश’,’चीन’,’भी’,’ड्राइवरलेस’,’मेट्रो ’, ‘चलती’ Bharat, padosi, desh cheen, bhi, driverless, metro, chalati India, neighboring, country, China, also, Driverless, metro, runs	‘भारत’,’पड़ोसी’,’देश’,’चीन’,’ड्राइ वर लेस’,’मेट्रो’,’ चलती’ Bharat, padosi, desh, cheen, bhi, driverless, metro, chalati India, neighboring, country, China, also, Driverless, metro, runs	‘भारत’,’पड़ोस’,’देश’,’चीन’ , ‘ड्राइवरलेस’ ‘मेट्रो’, ‘चल’ Bharat, pados, desh, cheen driverless, metro, chal India, neighbor, country, China, driverless, metro, run	‘भारत’, ‘पड़ोस’, ’देश’,’चीन’, ‘ड्राइवर’,’मेट्रो’, ‘चल’ Bharat, pados, desh, cheen driver, metro, chal India, neighbor, country, China, driver, metro, run

Table 3 The ARI, NMI, and silhouette score for HAC, GMM, and K-means to compare cluster quality with developed techniques.

	HAC			GMM			K-means		
Technique	**ARI**	**NMI**	**Silhouette score**	**ARI**	**NMI**	**Silhouette score**	**ARI**	**NMI**	**Silhouette score**
Existing Technique	0.095	0.233	0.030	0.086	0.237	0.036	0.086	0.237	0.036
CBSCL Technique	0.079	0.234	0.029	0.092	0.250	0.037	0.092	0.250	0.037
CCSBL Technique	0.084	0.246	0.031	0.112	0.268	0.043	0.111	0.268	0.043
CCSCL Technique	0.095	0.235	0.031	0.124	0.282	0.042	0.124	0.282	0.042

In Table 3, we present the results obtained for the clustering algorithm using the given matrices. With the existing technique, we observed a silhouette score of 0.036, 0.036 and 0.036 for HAC, GMM and K-means respectively. However, using the CBSCL technique, we observed a slightly lower silhouette score of 0.029 for HAC, but slightly higher at 0.037 for both GMM and K-means. The CCSBL technique produced silhouette scores of 0.031, 0.043, and 0.043 for HAC, GMM, and K-means clustering methods, respectively. On the other hand, the CCSCL technique produced silhouette scores of 0.031, 0.042, and 0.042 for HAC, GMM, and K-means clustering methods, respectively. The Adjusted Rand Index (ARI) obtained with the current technique was 0.095, 0.086, and 0.086 for HAC, GMM, and K-means clustering methods, respectively. The Adjusted Rand Index (ARI) has been observed to be 0.079, 0.092, and 0.092 for HAC, GMM, and K-means respectively, with the CBSCL technique. Meanwhile, the ARI with the CCSBL technique is observed to be 0.084, 0.112, and 0.112 for HAC, GMM, and K-means respectively. Also, the Adjusted Rand Index (ARI) with the CCSCL technique is observed to be 0.095, 0.124, and 0.124 for HAC, GMM, and K-means respectively. Additionally, the Normalized Mutual

Information (NMI) for the existing technique is observed to be 0.233, 0.237, and 0.237 for HAC, GMM, and k-means respectively. The NMI for the CBSCL is observed to be 0.234, 0.250, and 0.250 for HAC, GMM, and k-means respectively. Furthermore, the NMI for the CCSBL is observed to be 0.246, 0.268, and 0.268 for HAC, GMM, and k-means respectively. Based on the experimentation and statistical analysis conducted, it has been observed that the proposed CCSCL technique is effective in improving cluster quality for the three clustering algorithms, as evidenced by the normalized mutual information (NMI) scores of 0.235, 0.282, and 0.282 for HAC, GMM, and k-means, respectively. The results show that the CCSCL technique outperforms the remaining technique in terms of cluster quality improvement across all measures. Moreover, pre-processing the data with custom stop words and lemmatization has been found to significantly improve the quality of the clusters. Table 4 shows the transliteration and translation of the top 3 words from each cluster obtained using HAC, K-means, and GMM algorithms.

Table 4: Transliteration and translation of top 3 words from visualization plots

Top words from clusters	Transliteration	translation
रह	Rah	stay
कह	Kah	say
नहीं	Nahi	No
पार्टी	party	party
फ़िल्म	film	film
चुनाव	chunav	election
काम	kaam	work
बच्चा	baccha	kid
दे	de	give
मोदी	Modi	Modi
रन	run	run
खेल	Khel	game
विकट	wicket	wicket
चीनी	Cheeni	China
अमरीका	America	America
पाकिस्तान	Pakistan	Pakistan
सरकार	sarkar	government
पुलिस	pulice	police
हमला	Hamala	attack
महिला	Mahila	women
गुजरात	Gujarat	Gujarat
कांग्रेस	congress	congress
राष्ट्रपति	Rashtrapati	president
मामला	Mamala	matter
इजराइल	ijarail	israel

3.2 Visualization results obtained with Clustering Algorithm

The results of the implementation of three clustering methods' visualization are described in detail in the section below.

3.2.1 Visualization Results with Hierarchical Agglomerative Clustering:

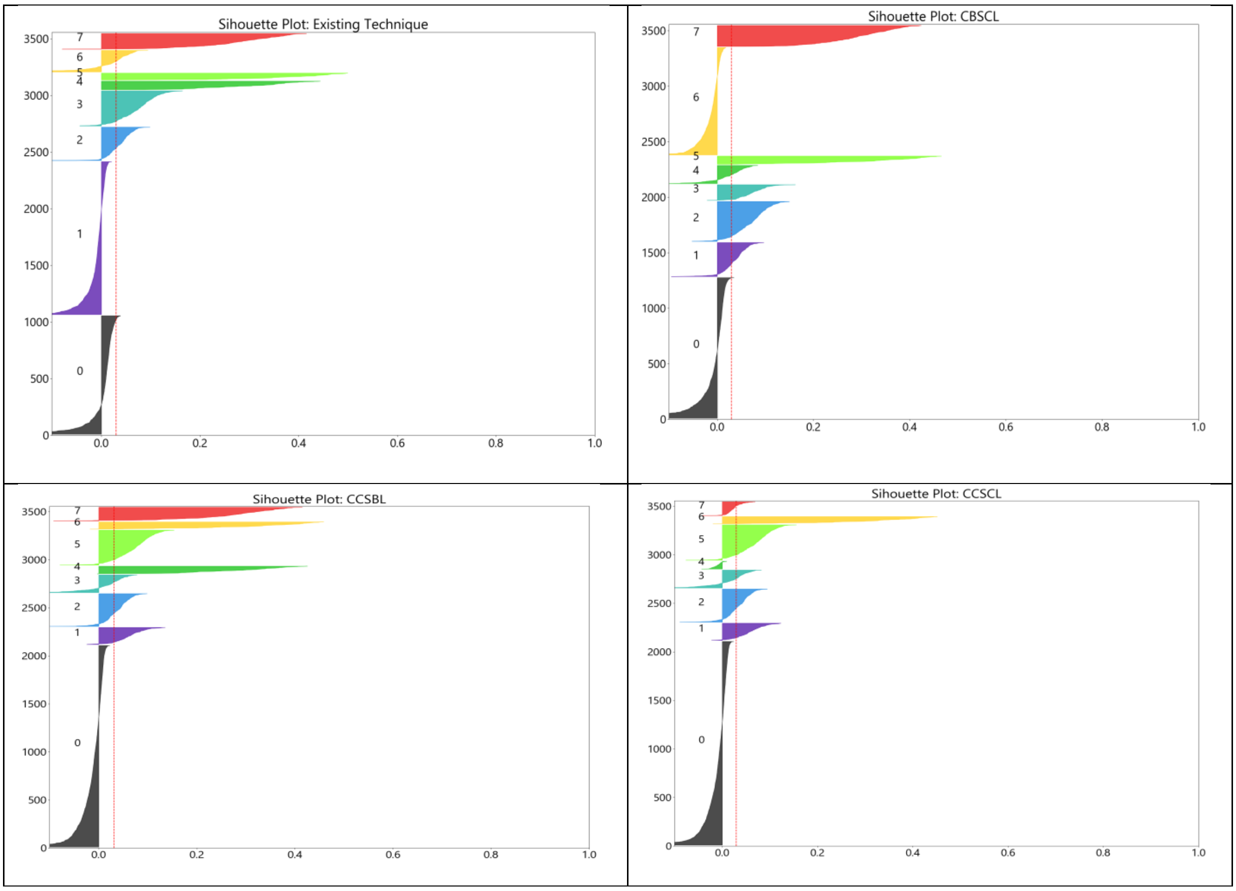
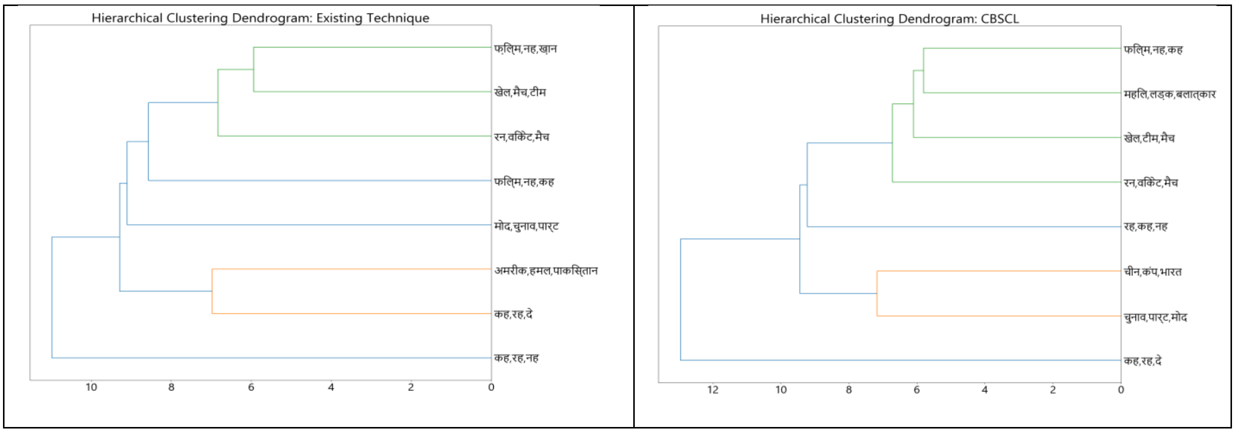


Fig 2: Silhouette score plots obtained with developed techniques for HAC algorithm.

Four techniques were developed and evaluated in this study. In Figure 2, the silhouette score plots reveal that For HAC algorithm the CCSCL technique outperformed the other techniques in terms of cluster quality, with only one cluster having a significant negative silhouette value. This indicates quantitative evidence of improved cluster quality with the CCSCL methodology. In Figure 3, the dendrograms for the four developed techniques are shown with the clusters labeled by the top 3 words from 25 sample documents. The results clearly demonstrate that the CCSCL technique has reduced the number of overlaps in the top 3 words between different clusters, providing qualitative evidence of improved cluster quality.



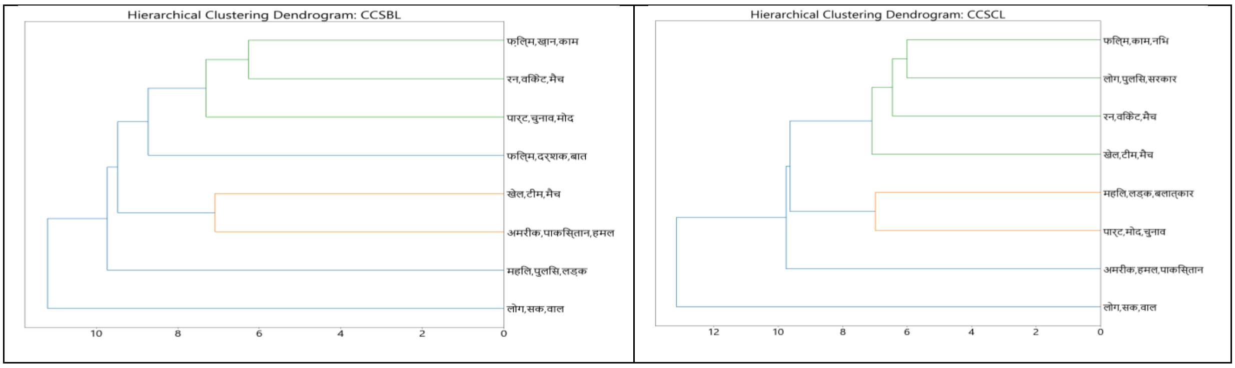


Fig 3: Dendrogram plots for HAC algorithm with developed techniques.

Figure 3 displays dendrograms of HAC with four techniques. Each leaf shows top 3 words. CCSCL technique outperforms others.

3.2.2 Visualization Results with Gaussian Mixture Models:

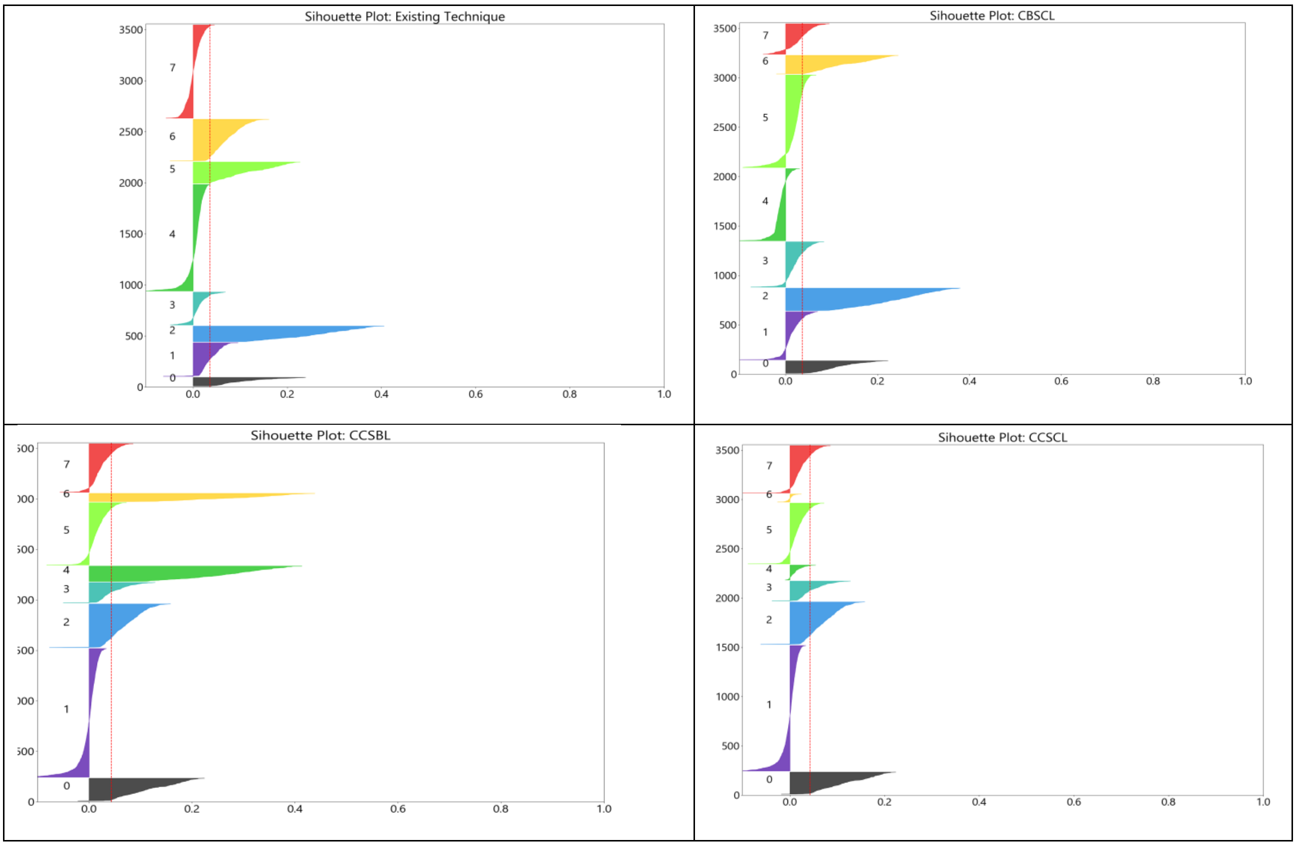


Fig 4: Silhouette Score plots for the developed techniques for GMM

CCSCL technique has shown clear improvement in cluster quality compared to other techniques, similar to HAC.

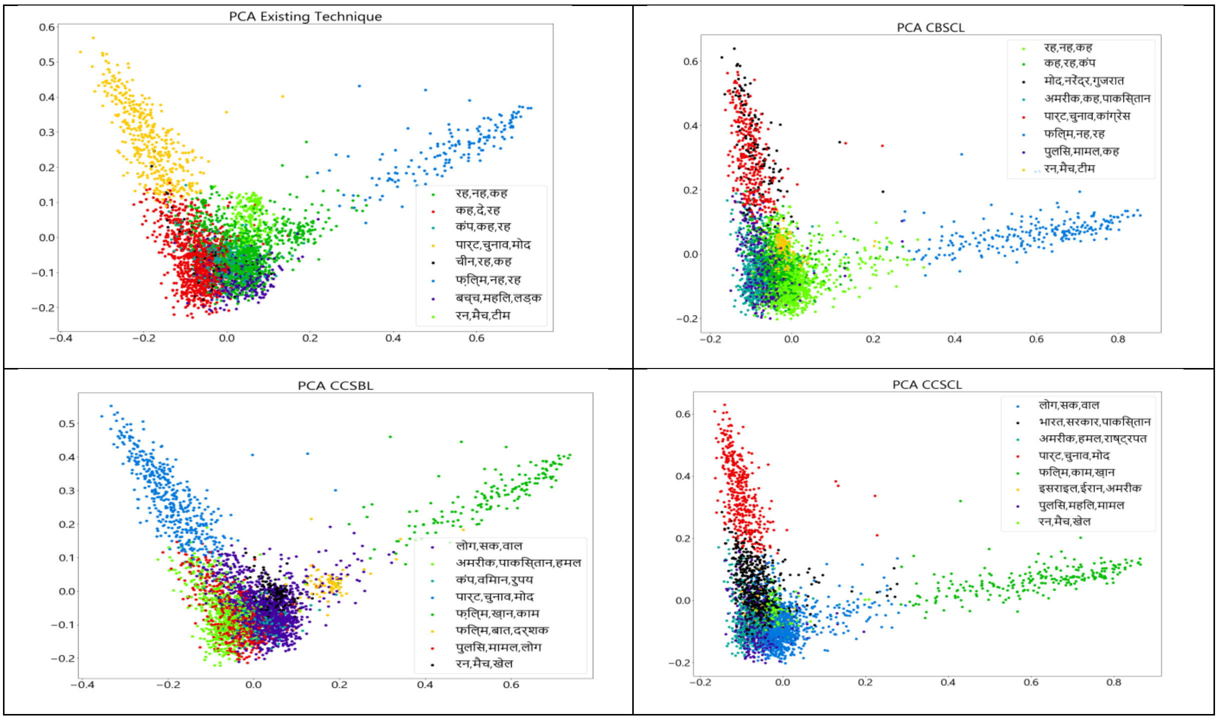
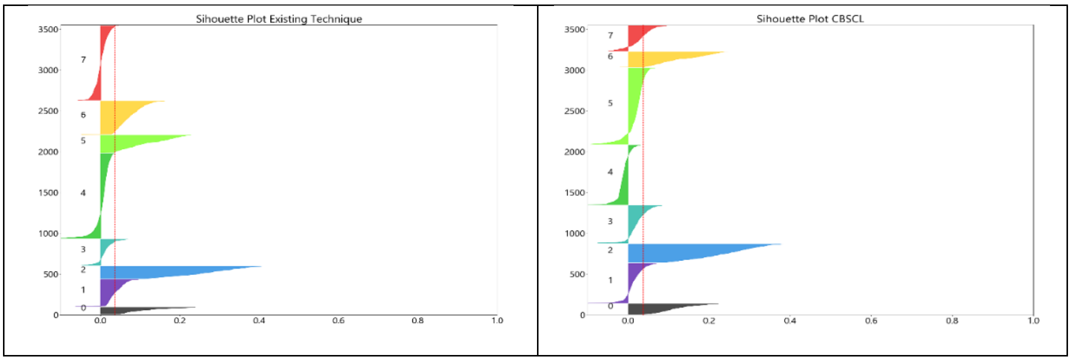


Fig 5 PCA plots implementation with developed techniques for GMM algorithm

A popular unsupervised machine learning technique for reducing the size of information is principal component analysis (PCA). Its main purpose is to aid in data visualization. When combined with CCSCL techniques, the data points can be easily observed and compared, since they generally tend to be well separated and closely clustered. In order to introduce novelty in visualization, the top three words from each cluster for each developed technique are implemented. While using existing techniques, we observed that some stop words, such as "रह", "कह", and "नह", which do not have any meaning, are represented as the top three words from that cluster. When clustering is performed with the CCSCL technique, such noise is eliminated and the top three words are correct and meaningful words from Hindi text. It has been observed that adding custom stopword removal, custom lemmatization, or both to the existing technique leads to an increase in both ARI and NMI. Results obtained with the CBSCL and CCSBL techniques show that custom stopword removal has a more significant impact on ARI and NMI than custom lemmatization, even though both have a positive effect. Therefore, custom stopword removal is more important than custom lemmatization. This can also be seen in Figure 8, where the silhouette plot of the CCSCL technique shows fewer clusters with negative silhouette scores compared to the effects of the other techniques. However, both cases show improvement over the existing technique, which has neither custom stopword removal nor custom lemmatization.

3.2.3 Visualization Results with K-means Clustering



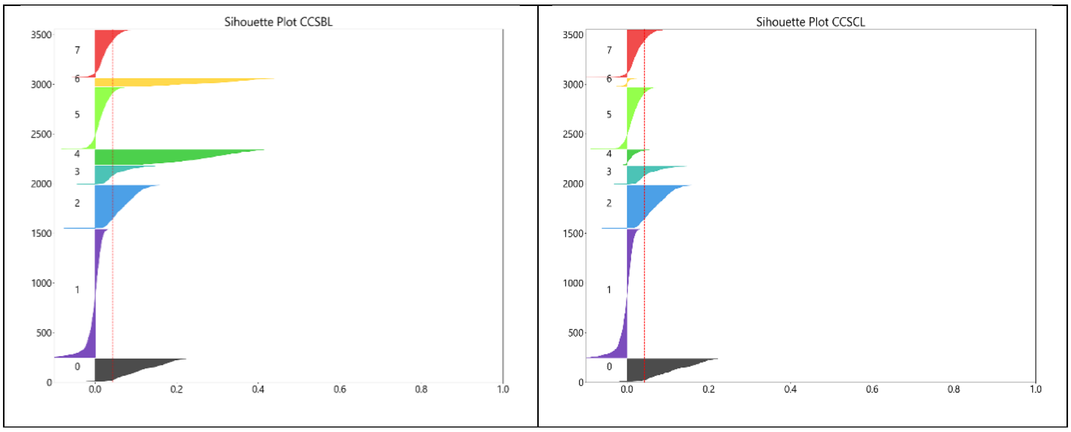


Fig 6: Silhouette score plots for K means Clustering with developed techniques.

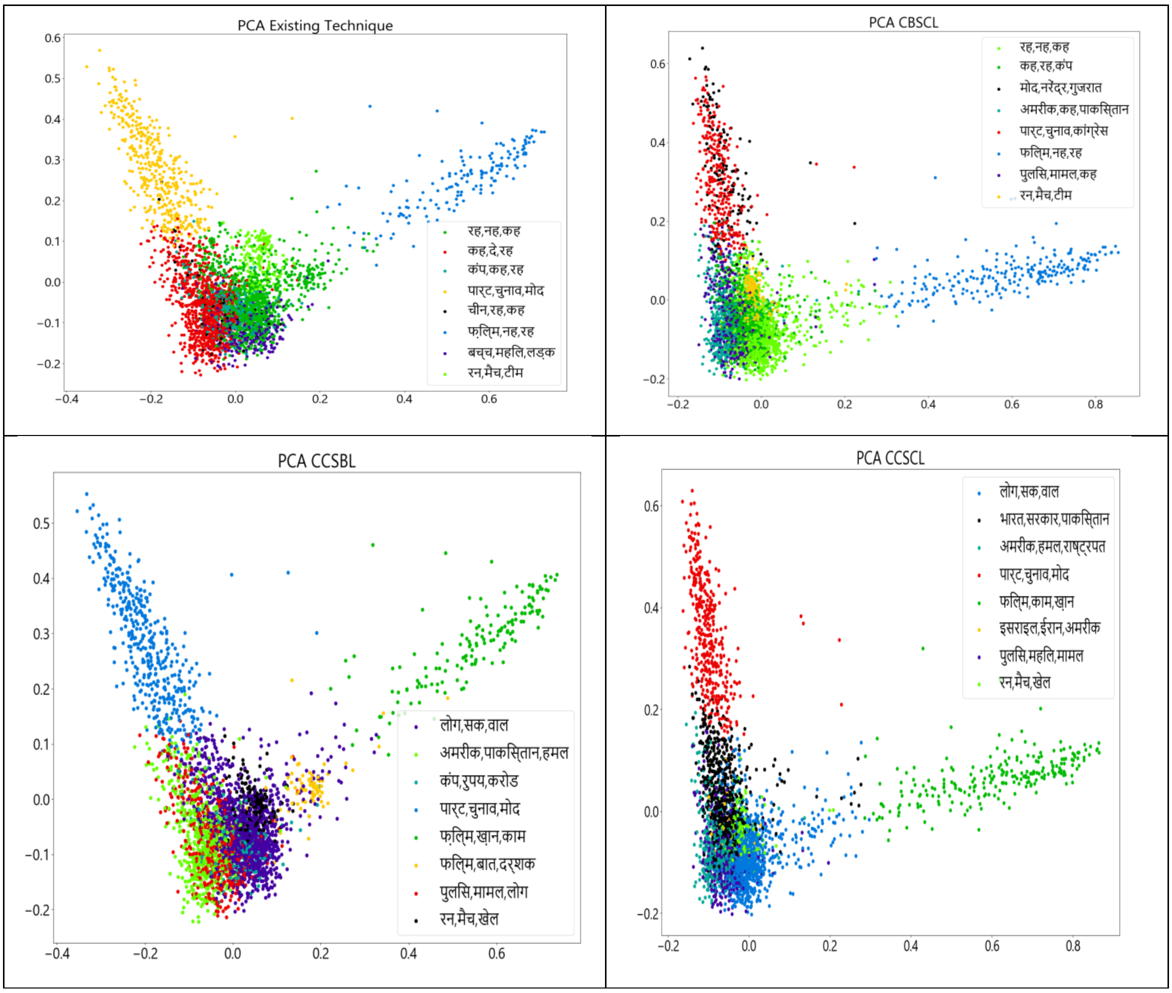


Fig 7. PCA plots for K-means algorithm implementation with developed techniques

The Silhouette score and PCA plots for the K-means algorithm are displayed in figures 6 and 7. The results demonstrate that the proposed CCSCL technique yields improved outcomes, as evidenced by the well-formed clusters and the top 3 words that are both meaningful and unique.

4. Conclusions

After conducting research on data pre-processing techniques, including stop word and lemmatization, for clustering algorithms, we have concluded that stop word removal is more significant in data pre-processing. Our study shows that using custom stop words and custom lemmatization improves cluster quality across multiple metrics. We have presented clustering results across multiple algorithms, and it is evident that the CCSCL technique enhances cluster quality. Silhouette plot analysis has helped us identify good and bad clusters and gauge cluster quality qualitatively. K-means and GMM algorithms have shown similar performance in terms of cluster quality and performance metrics. However, HAC does not benefit as much as K-means and GMM from the proposed methodology. For accurate data pre-processing for Indian languages, we suggest working on new feature extraction technique for document clustering as future work.

References

[1] S. Kumar and T. D. Singh, “Fake news detection on Hindi news dataset,” Glob. Transit. Proc., vol. 3, no. 1, pp. 289–297, Jun. 2022, doi: 10.1016/j.gltp.2022.03.014.

[2] I.-C. Chang, T.-K. Yu, Y.-J. Chang, and T.-Y. Yu, “Applying Text Mining, Clustering Analysis, and Latent Dirichlet Allocation Techniques for Topic Classification of Environmental Education Journals,” Sustainability, vol. 13, no. 19, p. 10856, Sep. 2021, doi: 10.3390/su131910856.

[3] R. Rani and D. K. Lobiyal, “Automatic Construction of Generic Stop Words List for Hindi Text,” Procedia Comput. Sci., vol. 132, pp. 362–370, 2018, doi: 10.1016/j.procs.2018.05.196.

[4] P. Verma and A. Verma, “Accountability of NLP Tools in Text Summarization for Indian Languages,” J. Sci. Res., vol. 64, no. 01, pp. 258–263, 2020, doi: 10.37398/JSR.2020.640149.

[5] P. B. Bafna and J. R., “Marathi Document: Similarity Measurement using Semantics-based Dimension Reduction Technique,” Int. J. Adv. Comput. Sci. Appl., vol. 11, no. 4, 2020, doi: 10.14569/IJACSA.2020.0110419.

[6] P. B. Bafna and J. R., “An Application of Zipf’s Law for Prose and Verse Corpora Neutrality for Hindi and Marathi Languages,” Int. J. Adv. Comput. Sci. Appl., vol. 11, no. 3, 2020, doi: 10.14569/IJACSA.2020.0110331.

[7] J. K. and J. R., “Stop-Word Removal Algorithm and its Implementation for Sanskrit Language,” Int. J. Comput. Appl., vol. 150, no. 2, pp. 15–17, Sep. 2016, doi: 10.5120/ijca2016911462.

[8] D. J. Ladani and N. P. Desai, “Stopword Identification and Removal Techniques on TC and IR applications: A Survey,” in 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India: IEEE, Mar. 2020, pp. 466–472. doi: 10.1109/ICACCS48705.2020.9074166.

[9] M. Nandathilaka, S. Ahangama, and G. T. Weerasuriya, “A Rule-based Lemmatizing Approach for Sinhala Language,” in 2018 3rd International Conference on Information Technology Research (ICITR), Moratuwa, Sri Lanka: IEEE, Dec. 2018, pp. 1–5. doi: 10.1109/ICITR.2018.8736134.

[10] K. Jacksi, R. Kh. Ibrahim, S. R. M. Zeebaree, R. R. Zebari, and M. A. M. Sadeeq, “Clustering Documents based on Semantic Similarity using HAC and K-Mean Algorithms,” in 2020 International Conference on Advanced Science and Engineering (ICOASE), Duhok, Iraq: IEEE, Dec. 2020, pp. 205–210. doi: 10.1109/ICOASE51841.2020.9436570.

[11] K. R. Shahapure and C. Nicholas, “Cluster Quality Analysis Using Silhouette Score,” in 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA), sydney, Australia: IEEE, Oct. 2020, pp. 747–748. doi: 10.1109/DSAA49011.2020.00096.

[12] K. Seki, M. S. Ortiz, and J. Mostafa, “Effectiveness and Efficiency for Document Clustering in Biomedicine,” in 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), San Diego, CA, USA: IEEE, Nov. 2019, pp. 1620–1623. doi: 10.1109/BIBM47256.2019.8983328.

[13] A. Sen, M. Pandey, and K. Chakravarty, “Random Centroid Selection for K-means Clustering: A Proposed Algorithm for Improving Clustering Results,” in 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA), Gunupur, India: IEEE, Mar. 2020, pp. 1–4. doi: 10.1109/ICCSEA49143.2020.9132921.

[14] V. Mehta, S. Bawa, and J. Singh, “WEClustering: word embeddings-based text clustering technique for large datasets,” Complex Intell. Syst., vol. 7, no. 6, pp. 3211–3224, Dec. 2021, doi: 10.1007/s40747-021-00512-9.

[15] “An Artificial Intelligence Approach for Word Semantic Similarity Measure of Hindi Language,” KSII Trans. Internet Inf. Syst., vol. 15, no. 6, Jun. 2021, doi: 10.3837/tiis.2021.06.006.

[16] Kulkarni, A. P. ., & T. N., M. . (2023). Hybrid Cloud-Based Privacy Preserving Clustering as Service for Enterprise Big Data. International Journal on Recent and Innovation Trends in

Computing and Communication, 11(2s), 146–156. https://doi.org/10.17762/ijritcc.v11i2s.6037

[17] Mr. Kankan Sarkar. (2016). Design and analysis of Low Power High Speed Pulse Triggered Flip Flop. International Journal of New Practices in Management and Engineering, 5(03), 01 - 06. Retrieved from http://ijnpme.org/index.php/IJNPME/article/view/45

[18] Yadav, N., Saini, D.K.J.B., Uniyal, A., Yadav, N., Bembde, M.S., Dhabliya, D. Prediction of Omicron cases in India using LSTM: An advanced approach of artificial intelligence (2023) Journal of Interdisciplinary Mathematics, 26 (3), pp. 361-370.