Summarization of Document using Feature Selection Method: TF-IDF

Debisankar Jena
Dept. of Computer Science & Engineering
Odisha University of Technology and Research
Bhubaneswar, Odisha, India

Jyotirmayee Rautaray
Dept. of Computer Science & Engineering
Odisha University of Technology and Research
Bhubaneswar, Odisha, India

Pranati Mishra
Dept. of Computer Science & Engineering
Odisha University of Technology and Research
Bhubaneswar, Odisha, India

ABSTRACT
In NLP, text summarization is the technique of condensing information from huge texts to smaller one. The phases in the summarization process includes reading the texts, normalizing the data, removing stop words, stemming, morphological analysis, and producing the summary. It falls under the extractive, abstractive, and hybrid categories. For the suggested Indian Language text summarization, extractive text summarization is being used. One method for extractive text summarization is PageRank. Each sentence in the document functions as a vertex on a graph, which is the basis of how it functions. Each node's initial score is determined by the number of words in the sentence, and the edges between nodes are determined by the cosine similarity of the sentences preprocessing, feature extraction, and graph building are the three main processes in PageRank technique. For a better understanding of the context, one of the simple things to take is feature extraction. We employ a specific way to apply weights to specific terms in our document before modeling them once the initial text has been cleaned and normalized. TF-IDF (Term Frequency-Inverse Document Frequency) is used for CNN dataset which produces better summary as compared to bag of words. Precision, recall and f score is calculated for generated summary and tf idf delivers best result.

Keywords
ATS, Pagerank, bag of words, Fscore, tf idf, feature extraction and extractive

1. INTRODUCTION
The term "Natural Language Processing" (NLP) refers to a subset of artificial intelligence that describes a computer's capacity to process simple speech, which includes speech, text, and other forms of human communication. The ultimate goal is to build a system that can "understand" the content of papers, even the minute language differences that are employed in various situations [1-4]. Text and speech processing, morphological analysis, syntactic analysis, lexical semantics, relational semantics, discourse, etc. are all common tasks for which NLP is employed. Natural Language Understanding (NLU) and Natural Language Generation (NLG) are the two components of NLP. While NLG turns structured data produced by the system into human-readable text, NLU concentrates on capturing context and purpose [5].

The task of text summarization is to provide a succinct, fluid summary while maintaining the essential information's content and overall significance. Text summary is a method for reducing a text's length without sacrificing its logical organization. Automatic text summarization attempts to compress lengthy papers into manageable sizes because doing it manually could be time-consuming and expensive. Summaries are delivered with condensed text material. Automatic text summarization aims to distil lengthy works down to their core ideas. Manual text summarizing could be time and money consuming. Text summary can be categorized as Single Document or Multi Document depending on the input type [6-9]. Text summary can be categorized as extractive summarization depending on the output type, in which only the most crucial sentences or phrases are chosen from the original text and extracted. Here, the already-existing sentences are only utilized, and new phrases are formed from the original information. The sentences created by abstractive summarization might not have existed in the source text [10]. Text summarization can be categorized as Generic, Domain Specific, or Query based, depending on its intended use.

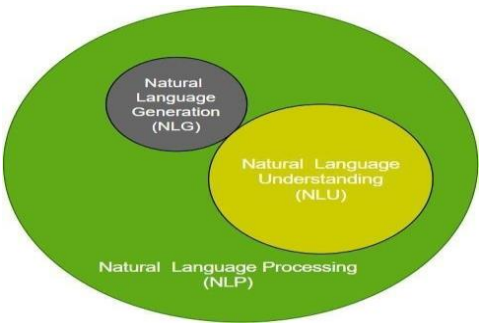

Fig. 1. Types of Natural Language Processing

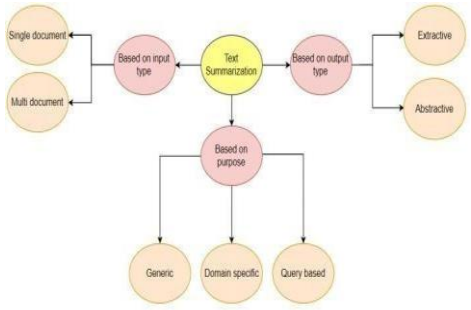

Fig. 2. Types of Text Summarization

Text summaries can also be placed in one of two categories: Informative Summaries and Descriptive Summaries [11].

Informative Summaries provide the information contained in a text or other sort of resource in a clear and concise manner. They provide objective reports on the text's content. Descriptive summaries depict the actual text (material) instead of presenting the information directly [12, 13].

It is observed that students how lengthy documents slowed down learning in the real world. These difficulties affect not only students but also people from different fields of study. By extracting all the crucial information, automatic text summarization will assist in compressing articles, scaling back the reading time from minutes to only seconds. This makes it far simpler to quickly process important information. An automatic summarizer is developed that will result in a summary that contains the key points of the input text while taking up the least amount of space possible and reducing repetition based on rouge scores in order to identify the best approach for text summarization on a single document. The rest of the section of the paper is organized as follows. Section 2 defines the literature review, section 3 is given as proposed method, section 4 is defined by the implementation and result analysis. Finally, section 5 defines the conclusion and future work.

2. LITERATURE SURVEY

Aakash Sinha et. al. in 2018 proposed a totally data-driven technique based on neural networks for automatic text summarization by testing the model on standard DUC 2002 dataset, and it produced results that were equivalent to models at the cutting edge without any verbal input. This study demonstrated that a straightforward approach may provide outcomes that are on par with those of complex deep networks or sequence-based models.

Mehdi Allahyari et. al. in 2017 explored various extractive approaches for single and multi-document summarization and concluded that approaches for representation of topic, frequency-based methods, techniques based on graph and machine learning algorithms are some of the most often utilized methodologies.

Reda Elbarougy et. al. in 2020 proposed using the Modified PageRank Algorithm with multiple iterations to solve the problem of locating nouns in the Arabic Language. The implementation was carried out on the standard EASC corpus, yielding an F-measure of 67.98.

Shashi Narayan et. al. in 2018 proposed a training algorithm to optimize a reward function that is relevant to the job at hand while exploring the universe of candidate summaries on the CNN and Daily Mail datasets. The experimental results reveal that the REINFORCE algorithm is an excellent way to push our model toward producing informative, fluent, and concise summaries, surpassing state-of-the-art extractive and abstractive systems.

Jyotirmayee Rautaray et al. in 2022 introduced a comparison study of different graph-based techniques on DUC datasets.

Soumi Dutta et. al.in 2015 proposed a graph-based method to summarize tweets. First, the graph is created based on how similar the tweets are, and then community detection algorithms are applied to the graph to group the similar tweets. The proposed approach achieves better performance than Sumbasic, one of the algorithms. This algorithm works on WordNet and Graph Clustering.

Muhamad Fahmi Fakhrezi et. al. in 2021 proposed that as the Quran's vocabulary is not that easy to understand hence a Quranic vocabulary encyclopedia will be helpful in explaining the words in it. Automatic Text summarization using TextRank algorithm is the approach to build such an encyclopedia.

Shrabanti Mandal et. al. in 2018 proposed a PSO algorithm using similarity measurement as a fitness function and three well-defined constraints to provide a better outcome in extractive summarization. Australian legal cases from the Federal Court of Australia (FCA) are represented in ten separate papers which are taken as the dataset. To check the accuracy, we used ROUGE-1 to optimize recall, precision, and f-factor, and ROUGE-2 to improve precision.

Kaichun Yao et. al. in 2018 proposed an approach to deep reinforcement learning based on a Deep Q-Network (DQN) similar to current deep learning models. The RNN-RNN structure performs somewhat better than the CNN-RNN structure because of long-term interdependence. The deep reinforcement learning- based extractive summarization technique used CNN/Daily corpus, the DUC 2002 dataset and the DUC 2004 dataset that can be modified directly using Rouge metrics, by elimination of the need for extractive sentence-level labeling.

Chiranatana Mallick et. al. in 2018 proposed a graph-based method for extractive text summarization using lexical chains. The linguistic features which were ignored in the statistical summarization approaches are being processed by this method. Besides using lexical chains, they have also used coherence metrics to select the summary sentences.

ShivaKumar KM and Soumya R in 2015 proposed a method that focuses on extractive summarization, and uses accuracy and precision measures to compare the results with those of standard systems. This method increases phrase simplification and decreases duplication, according to the results. This summarizer either uses clustering or SVM technique.

R.C. Balabantaray et.al. in 2012 proposed a paper in which he talks about taking the fonts of the documents in consideration for calculating the weight and rank of the sentences while summarizing any text, with reference to MS Word.

Ani Nenkova et. al. in 2012 proposed how the success of KL divergence as a method for rating sentences directly integrates an intuition about what makes a good summary. This paper demonstrates how summarizing algorithms must be tailored to diverse genres, such as web pages and journal papers, while taking contextual information into account.

Busrat Jahan et. al. in 2021 proposed a summarization approach in which preprocessing, word tagging, replacing pronoun, sentence ranking and summary generation phases are used. This approach gives a new way of summarization of Bangla text Summarization. He also said that there is a lot of research work in English text summarization but not in Bangla due to its complexity in terms of sentence format, rules of the grammar, word reflection etc. In spite of all these hurdles his approach was a groundbreaking way of summarizing Bengali news.

3. PROPOSED MODEL

Here is the general representation of how the algorithm works.

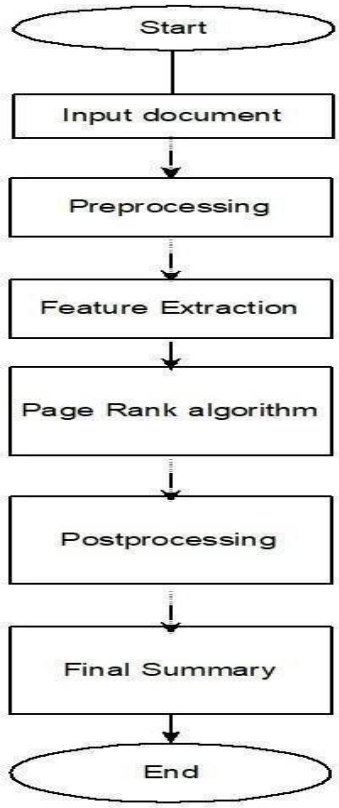

Fig. 3. General Representation of PageRank Algorithm

3.1 Preprocessing:

Pre-processing refers to cleaning of data and making it ready for further use. It aids in the removal of excess data and noise. NumPy, NLTK, pandas and many other python libraries are used for pre-processing the data. It is further divided into a number of processes: -Process of taking the text document as an input. The text document can be a huge dataset containing thousands of lines. Normalization is a process of eliminating variations in a text and lowering the number of unique tokens present in it. Tokenization emphasizes the term token. Splitting text into tokens, which might be words from sentences or sentences from documents, for example, is part of this process. Removing these stop words allows you to concentrate on more relevant information. We eliminate these stop words like "is," "the," from our data by importing a module called "stop words" from NLTK [16-18].

3.2 Feature Extraction:

When trying to resolve issues relating to language processing, we must deal with a lot of raw data. However, including such data in our models directly will not contribute to meaningful results. The process of selecting and/or merging variables into features, known as "feature extraction," drastically decreases the amount of data that needs to be processed while accurately and completely resembling the initial data set. These tokens are known as features. The frequency of words in a document's sentences is one type of feature, and cosine similarity is the other. The document's high-frequency terms define the relevance of a sentence. This stage primarily determines the relevance of a sentence based on the words in the sentence and their frequencies, as well as the frequency of words and phrases throughout the document. Some of the methods of feature extraction that we have used are:

Bag of words: Every piece of writing is considered to be a unit, which could be a phrase or a paragraph. A bag is this object. Create a vocabulary matrix of all distinguishable terms in the first stage, with each row representing a sentence and each column representing a document. The cells of the matrix are then filled, depending on whether the word is present or not. If the word is present, the cell is given a value of 1, and if the word is absent, the cell is given a value of 0.

TF-IDF: Even though some words don't appear often, they can nonetheless be quite important to the narrative. As a result, in addition to being directly proportional to a word's frequency in the corpus, a word's TF-IDF score is also inversely related to the number of documents in which it appears. The significance of a sentence is determined by the high- frequency terms in the document. One form of feature is the frequency of words in a document's sentences; the other is cosine similarity. Based on the words in the sentence and their frequencies, as well as the frequency of words and phrases used throughout the document, this stage primarily determines the relevance of a sentence.

Finding each word count: - The frequency of each word in a document sentence is determined in this step. This frequency is used to determine the cosine similarity.

Frequency using Bag of Words: Count Vectorizer can only find a small portion of the vocabulary's frequently used terms in each page. As a result, a vector with lots of zero scores, often known as a sparse vector is created. The sparse vector is then used to build the sparse matrix. It is converted into a data frame in order to be visualized.

Frequency using TF-IDF: Frequency lists the number of times a specific term (t) appears in document (d). Inverse Document Frequency (idf) typically assesses the significance of a phrase. Because tf considers all phrases to be equally important, we cannot simply use word frequencies to calculate a term's weight in the document [14,15]. The common terms must be scaled back while the uncommon terms are scaled up. We can resolve this difficulty with the help of logarithms. The idf of a term is calculated by dividing the total number of documents in the corpus by the term's document frequency.

Consider the insects are in different locations while moving randomly one of them has found some food at green location. This green location is so far the based and this insect will inform others about his location. However, this is still not optimal or biggest target. So how they will reach that target. Imaginary living particle where they can smell hidden source food the one who is closest to the food makes loudest sound then the other particles move around them. Any moving particles come closer to the target then the first one making out the sound others moved to it. This mechanism continues until First one has it.

3.3 Pagerank Algorithm

Generally, PageRank algorithm is used by Google to rank its web pages. It counts the number and quality of links to the page and the estimates on how useful or important the page is. PageRank is a link analysis algorithm which means association/relation between different objects of different types. It assigns numerical weighting to a hyperlinked set of documents i.e., measuring the importance of the link and comparing it with the other ones in the set. The numerical weight assigned to any element A is referred to as “PageRank of A” or PR(A). The extractive text summarization method PageRank uses graphs as its foundation. The range of the cosine similarity between two documents is 0 to 1[16]. For each node in the graph, a PageRank score is

determined. The PageRank score is calculated by the given formula: -

$$PR(pi) = \frac{1-d}{N} + d \sum pj \in M(pi) \frac{PR(pj)}{L(pj)};$$

Here,

PR (pi) = PageRank score of pi node,
d = Damping factor,
N=No of nodes,
pj=adjacent node to pi
L(pj) = the in degree to the vertex

By this method the PageRank score for each vertex is calculated. The ones with higher PageRank value are taken into account to summarize the text and the one with least PageRank value is neglected. This clearly means that the one with higher PageRank score is more important than the one with least PageRank score [17,18].

4. IMPLEMENTATION AND RESULT ANALYSIS

Input: CNN Dataset
Output: Summary of paragraph

1. Input paragraph

2. for each sentence in paragraph:
   Normalization()

ROUGE	PRECISION	RECALL	F1-SCORE
ROUGE-1	0.020408163265306 12	0.5	0.03921568552095 348
ROUGE-2	0.006369426751592 357	0.3	0.012499999632031 262
ROUGE-L	0.020408163265306 12	0.5	0.03921568552095 348

   Tokenization()
   Stop Words Removal()
   Lemmatization()

3. Dataframe1 <- Frequency using TF-IDF Dataframe2 <- Frequency using Bag of words

4. Generate graphs based on the cosine similarities using TF-IDF and bag of words G1- TF-IDF and G2- bag of words

5. For each node in G1: Calculate PageRank score Sort PageRank scores in decreasing order

6. For each node in G2: Calculate PageRank score Sort PageRank scores in decreasing order.

7. Input n<- No of lines required in summary

8. For both the methods, perform: For i=1 to n: Generate a summary string adding sentences according to PageRank score.

9. Calculate the rouge scores by comparing the summary generated and the reference summary (expected summary)

10. Compare the F1-scores of the rouge scores generated through both the summaries.

The goal of our algorithm is to summarize larger text into smaller ones. Here, we have applied our algorithm for summarization using two different feature extraction methods and compared their results.

INPUT DOCUMENT: CNN Dataset

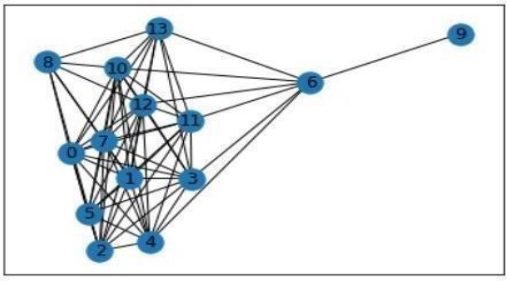

Fig. 4. Graph made using TF-IDF

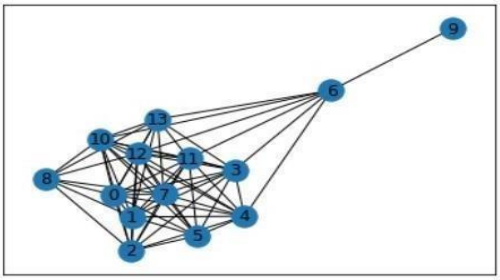

Fig. 5. Graph made using bag of words

CASE-1: WHEN LENGTH OF SUMMARY IS GREATER THAN REFERENCE SUMMARY

Table 1. For TF-IDF the ROUGE scores are as follows

Table 2. For bag of words, the ROUGE scores are as follows

ROUGE	PRECISION	RECALL	F1-SCORE
ROUGE-1	0.018348623853211 01	0.5	0.03539822940559 168
ROUGE-2	0.005780346820809 248	0.3	0.011363636028538 232
ROUGE-L	0.018348623853211 01	0.5	0.03539822940559 168

Comparing the above ROUGE scores calculated through different methods the comparison can be demonstrated as:

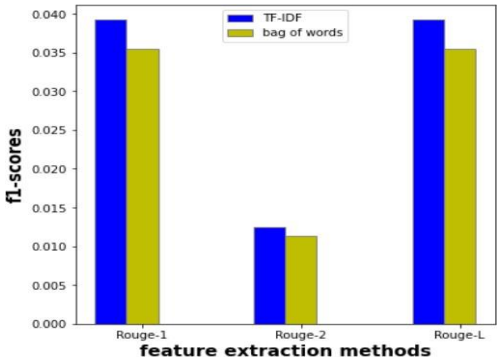

Fig. 6. Graphical representation of comparison of F1-scores using TF-IDF and bag of words in case 1

CASE-2: WHEN LENGTH OF SUMMARY IS EQUAL TO REFERENCE SUMMARY

Table 3. For TF-IDF the ROUGE scores are as follows

ROUGE	PRECISION	RECALL	F1-SCORE
ROUGE-1	0.35714285714285715	0.47619047619047616	0.4081632604081633
ROUGE-2	0.13888888888888889	0.20833333333333334	0.16666666186666684
ROUGE-L	0.30357142857142855	0.40476190476190477	0.346938770612245

Table 4. For bag of words, the ROUGE scores are as follows

ROUGE	PRECISION	RECALL	F1-SCORE
ROUGE-1	0.3488372093023256	0.35714285714285715	0.3529411714128035
ROUGE-2	0.05454545454545454	0.0625	0.05825242220756005
ROUGE-L	0.3023255813953488	0.30952380952380953	0.3058823479418686

Comparing the above ROUGE scores calculated through diff methods the comparison can be demonstrated as:

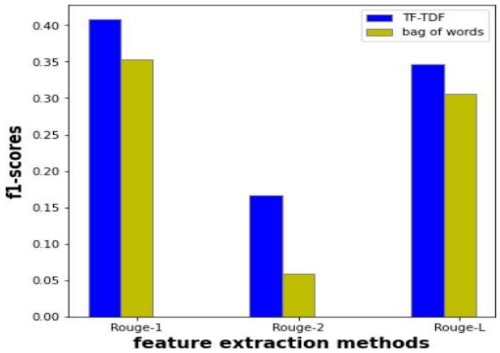

Fig. 7. Graph Graphical representation of comparison of F1-scores using TF- IDF and bag of words in case 2

CASE-3: WHEN LENGTH OF SUMMARY IS LESS THAN REFERENCE SUMMARY

Table 5. For TF-IDF the ROUGE scores are as follows

ROUGE	PRECISION	RECALL	F1-SCORE
ROUGE-1	0.4133333333333333	0.5081967213114754	0.4558823479941609 4
ROUGE-2	0.1545454545454545	0.2328767123287671	0.1857923449311715 3
ROUGE-L	0.3866666666666666	0.4754098360655737 4	0.4264705832882786

Table 6. For bag of words, the ROUGE scores are as follows

ROUGE	PRECISION	RECALL	F1-SCORE
ROUGE-1	0.3953488372093023	0.2786885245901639	0.3269230720728551
ROUGE-2	0.0545454545454545 4	0.0410958904109589	0.0468749950988 7747
ROUGE-L	0.3488372093023256	0.2459016393442623	0.2884615336113166

Comparing the above ROUGE scores calculated through diff methods the comparison can be demonstrated as:

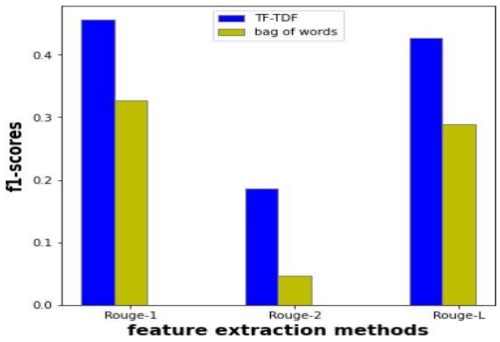

Fig. 8. Graphical representation of comparison of F1-scores using TF-IDF and bag of words in case 3

5. CONCLUSION AND FUTURE SCOPE

After implementing the PageRank algorithm employing the Bag of Words and TF-IDF feature extraction methods against a variety of text documents, we note that the f1-scores of ROUGE-1, ROUGE-2, and ROUGE-L are higher for TF-IDF than for the Bag of Words model. As a result, we draw the conclusion that using TF-IDF with cosine similarity during the preprocessing stage results in more accuracy as compared to the Bag of Words model with cosine similarity. Different feature extraction techniques and text summarization approaches need to be discovered which one performs the best to improve performance measures and prediction accuracy.

6. ACKNOWLEDGMENTS

Our thanks to the experts who have contributed towards development of the template.

7. REFERENCES

[1] Elbarougy, R., Behery, G., & El Khatib, A. (2020). Extractive Arabic text summarization using modified PageRank algorithm. Egyptian informatics journal, 21(2), 73-81

[2] Sinha, A., Yadav, A., & Gahlot, A. (2018). Extractive text summarization using neural networks. arXiv preprint arXiv:1802.10137.

[3] Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., & Kochut, K. (2017). Text summarization techniques: a brief survey. arXiv preprint arXiv:1707.02268.

[4] Narayan, S., Cohen, S. B., & Lapata, M. (2018). Ranking sentences for extractive summarization with reinforcement learning. arXiv preprint arXiv:1802.08636.

[5] Rautray, R., & Balabantaray, R. C. (2017). Cat swarm optimization based evolutionary framework for multi document summarization. Physica a: statistical mechanics and its applications, 477, 174-186.

[6] Sanchez-Gomez, J. M., Vega-Rodríguez, M. A., & Pérez, C. J. (2018). Extractive multi-document text summarization using a multi-objective artificial bee colony optimization approach. Knowledge-Based Systems, 159, 1-8.

[7] Dutta, S., Ghatak, S., Roy, M., Ghosh, S., & Das, A. K. (2015, September). A graph-based clustering technique for tweet summarization. In 2015 4th international conference on reliability, infocom technologies and optimization (ICRITO) (trends and future directions) (pp. 1-6). IEEE.

[8] Fakhrezi, M. F., Bijaksana, M. A., & Huda, A. F. (2021). Implementation of automatic text summarization with TextRank method in the development of Al-qur'an vocabulary encyclopedia. Procedia Computer Science, 179, 391-398

[9] Mandal, S., Singh, G. K., & Pal, A. (2018). A Constraints Driven PSO Based Approach for Text Summarization. Journal of Informatics & Mathematical Sciences, 10(4).

[10] Yao, K., Zhang, L., Luo, T., & Wu, Y. (2018). Deep reinforcement learning for extractive summarization. Neurocomputing, 284, 52-62. document

[11] Mallick, C., Dutta, M., Das, A. K., Sarkar, A., & Das, A. K. (2019). Extractive summarization of a document using lexical chains. In Soft Computing in Data Analytics: Proceedings of International Conference on SCDA 2018 (pp. 825-836). Springer Singapore.

[12] Al-Saleh, A., & Menai, M. E. B. (2018, August). Ant colony system for multi-document summarization. In Proceedings of the 27th International Conference on Computational Linguistics (pp. 734-744).

[13] Rautray, R., & Balabantaray, R. C. (2018). An evolutionary framework for multi document summarization using Cuckoo search approach: MDSCSA. Applied computing and informatics, 14(2), 134-144.

[14] Verma, P., & Om, H. (2019). A variable dimension optimization approach for text summarization. In Harmony Search and Nature Inspired Optimization Algorithms: Theory and Applications, ICHSA 2018 (pp. 687-696). Springer Singapore.

[15] Tomer, M., & Kumar, M. (2022). Multi-document extractive text summarization based on firefly algorithm. Journal of King Saud University-Computer and Information Sciences, 34(8), 6057-6065.

[16] Shivakumar, K., & Soumya, R. (2015). Text summarization using clustering technique and SVM technique. International Journal of Applied Engineering Research, 10(12), 28873- 28881.

[17] Mutlu, B., Sezer, E. A., & Akcayol, M. A. (2020). Candidate sentence selection for extractive text summarization. Information Processing & Management, 57(6), 102359.

[18] Rautaray, J., Panigrahi, S., & Nayak, A. (2022, August). An Empirical and Comparative Study of Graph based Summarization Algorithms. In 2022 International Conference on Machine Learning, Computer Systems and Security (MLCSS) (pp. 274-279). IEEE.