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

Revolutionizing Single Document Extractive Text Summarization with Improved PageRank

Jyotirmayee Rautaray [*1], Sangram Panigrahi [2], Ajit Kumar Nayak [2]

Submitted: 25/01/2024 **Revised**: 03/03/2024 **Accepted**: 11/03/2024

Abstract: In recent days due to the exponential growth of data on the internet, it is now quite challenging to extract information within the time frame specified. A crucial approach to address this issue is an effective and efficient automatic text summarization. This paper focuses on extractive text summarization of single document, taking into account the type of document and summary. This study introduces the improved-PageRank algorithm, a graph-based text summarization technique that captures the aboutness of text content, which is an enhanced version of the modified PageRank algorithm. The proposed technique is evaluated against two other approaches, TextRank and modified PageRank, using the dataset from the Document Understanding Conference, i.e. DUC 2002, DUC 2003 and DUC 2005. ROUGE value, range, and coefficient of variation are used to compare the effectiveness of each algorithm. This experimental study clearly indicates that the improved-PageRank technique provides the best result when compared to other techniques.

Keywords: TextRank, PageRank, Modified PageRank, Cosine similarity, Dimensionality Reduction

1. Introduction:

The exponential growth of text-based data on the internet necessitates efficient summarization methods. Manual summarization is time-consuming and costly due to the overwhelming volume and complexity of the content [1, 2]. Automatic Text Summarization (ATS) emerges as a solution, aiming to extract key information while eliminating redundancies. ATS finds application in various domains such as news, queries, emails, graphs, microblogs, stories, and legal documents [3, 4]. By condensing large bodies of text, ATS enhances accessibility and comprehension of information, addressing the challenges posed by the data overflow in today's digital landscape.

Automatic Text Summarization (ATS) methods are divided into two main categories: extractive and abstractive [5]. Extractive approaches select relevant sentences based on parameters like position and similarity to the title, while abstractive methods generate summaries using human-like language [6]. ATS also varies based on input: single-document and multi-document summarization [7-9]. Multi-document summarization presents additional complexities. Summaries can be generic, providing an overview of key ideas, or query-based, tailored to user queries [10,11]. Query-based summaries incorporate user keywords, unlike generic summaries [10,11]. These distinctions showcase the diverse applications and challenges within the field of text summarization.

Extractive summarization involves preprocessing, intermediate processing, and postprocessing steps [12,13]. Preprocessing includes tasks like sentence segmentation and stop word removal. Intermediate processing, known as keyword extraction, employs various techniques such as graph-based [14], statistical-based [15], clustering-based [16], and linguistic-based [17] methods. Sentence scoring occurs in this phase to assign importance. Postprocessing generates and evaluates the final summary, utilizing high-scoring sentences [18]. Automatic text summarization addresses time constraints in document comprehension, aiding in content organization, language learning, and multilingual communication. It condenses lengthy documents, reducing reading time significantly. This study advocates a novel single-document graph-based technique for extractive summarization, aiming to provide efficient and high-quality summaries.

The paper's structure unfolds as follows: Part 2 outlines the literature review, section 3 demonstrates proposed framework, section 4 presents the experiment outcomes, and in section 5, the paper concludes, offering potential avenues for further study.

2. Related Work:

Graph-based algorithms are pivotal in text summarization due to the abundance of online content [16]. These algorithms leverage directed weighted graphs, with nodes representing sentences and edges indicating connections [4]. Keyword extraction heavily influences candidate

[1]*Department of Computer Science & Engineering, Siksha 'O' Anusandhan (deemed to be University), Bhubaneswar-751030, Odisha, India*
[2]*Department of Computer Science & Information Technology, Siksha 'O' Anusandhan (deemed to be University), Bhubaneswar-751030, Odisha, India*
[*] *Corresponding Author Email: jyotirmayee.1990@gmail.com*

sentence selection. Notably, PageRank, utilized for ranking websites, evaluates link quantity and quality [23]. However, PageRank is restricted to web page summarization and cannot summarize text [25]. This approach finds application across various fields including social science, homeland security, economics, healthcare, web analysis, and linguistics.

TextRank[20], akin to PageRank for webpages, summarizes text by treating sentences as nodes in a weighted graph. Extracted sentences form nodes, with edge weights reflecting similarity measured by common tokens. Higher similarity yields heavier edges. This algorithm efficiently summarizes text by prioritizing significant sentence connections, as outlined.

TextRank generates a dense graph from a text's similarity matrix, refined through iterative PageRank application. Summaries entail selecting the initial k sentences. Limitations include disregarding contextually meaningful yet infrequent words [22], and sentence scores often closely resembling each other due to feature extraction yielding many related tokens [19]. A novel method for text summarization called LexRank [21] was proposed by Gunes Erkan and Dragomir R. Radev in 2004 which determines a sentence's significance by considering the eigenvector centrality of each sentence in a graph representation of sentences. The DUC 2003 and 2004 datasets were used for experiment purpose.

For the automatic summarization of documents, Federico Barrios et al. [14] provided additional alternatives to the TextRank algorithm's similarity function in 2016 like BM25, BM25+, Cosine Similarity, and Longest Common Substring (LCS). The BM25 and BM25+ strategies delivered the best results on the DUC 2002 dataset.

In 2019, Mallick et al. [10] proposed a modified TextRank algorithm enhancing text summarization by capturing document aboutness. Unlike traditional cosine similarity, their method adjusts for varying sentence lengths and term relevance degrees through isf-modified-cosine similarity. Evaluation demonstrates its effectiveness in summarization, offering a promising advancement.

K Usha Manjari in 2020 [18] established TextRank method for Telugu document summarization where for feature extraction bag of words is used, by which words having less frequency won't be taken into consideration. Limitation of TextRank is that similar token in the sentences is not given importance and it suffers this kind of problem in Telugu language.

Jun LI et al. [26] introduced a naïve approach for keyword extraction. To overcome the challenge faced when dealing with short texts using Latent Dirichlet Allocation (LDA) topic model, here authors have exhibited commendable performance with longer texts by using Word2Vec and Doc2Vec technique incorporating with Textrank algorithm. Moreover, this model exhibited consistent and reliable performance in longer texts as well as short text.

Yadav et al. [23] puts forward a new approach called TGETS (Textual graph extractive Text Summarization) using lemmatization process algorithm that involves techniques like graph representation, sentence weighting and graph analysis for BBC news dataset.

Shanshan Yu et al. [14] introduced "iTextRank," an unsupervised machine learning approach, surpassing TextRank's scalability for keyword extraction and text summarization. They tested it on 3600 articles from BBC, CNN, NBC, and Gawker datasets. iTextRank computes sentence similarity, adjusts node weights based on statistical and linguistic features, and evaluates summaries using ROUGE metric.

In this section, various graph-based techniques have been reviewed along with their benefits and drawbacks. The subsequent section will cover a revolutionary graph-based approach.

3. Proposed Model

The summarization process relies on input features to score sentences, with quality hinging on feature selection. The proposed improved-PageRank technique employs cosine similarity to compare features and reduces redundancy through threshold constraints. The model comprises pre-processing, feature extraction, feature selection, modified PageRank application, and summary extraction, streamlining summarization by efficiently selecting relevant features for generating concise summaries. The flow chart of proposed approach is depicted in Figure 1.

Where 1- Preprocessing
i. Tokenization
ii. Normalization

2- Feature Extraction
3- Feature Matrix (M_{cos_sim})
4- Dimensionality Reduction
5- Candidate summary
6- Reduced feature matrix
7- Graph generation
8- Apply modified pagerank
9- Ordering of sentences for summary generation
10- Summary evaluation

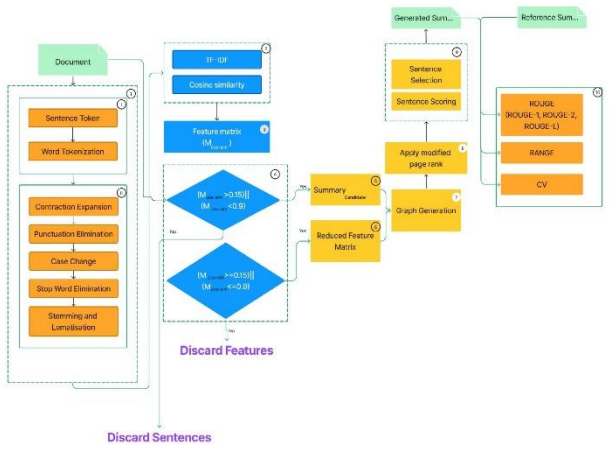

Fig 1: *Proposed Text summarization model*

3.1 Pre-processing

Pre-processing refers to cleaning of data and removing data that is unnecessary for summarization [27]. The following pre-processing steps are performed on text data before processing.

3.1.1 Tokenization

Tokenization itself emphasizes the term "token". The tokenization method separates a raw text into sentences or words. A token can be a word or a complete sentence that has been obtained from a document.

3.1.1.1 Sentence Tokenization:

The technique of breaking up a raw text into sentences is called sentence tokenization. The beginning and end of sentences are usually denoted by characters like "periods," "exclamation marks," and "newlines". Sentence tokenization technique searches for all periods, exclamation marks, and newlines in a text data.

3.1.1.2 Word Tokenization:

A technique for dividing a raw text into individual words is called word tokenization. In most cases, a "space" is used to separate words. In order to divide the text data into words, the word tokenization technique finds all of the spaces in a piece of text. The word tokenization technique has been employed in this case to extract words from sentences in order to determine their frequency.

3.1.2 Normalization:

Normalization helps to reduce number of unique tokens in a text by eliminating redundant information and cleaning the text by removing irrelevant elements. A few techniques employed in the normalization process include contraction expansion, punctuation elimination, case change, stop words elimination, stemming, and lemmatization. The various text normalization techniques are thoroughly discussed here.

3.1.2.1 Contractions Expansion

Contraction is a special kind of word which combines two or more words into a single word in an abbreviated form, typically replacing letters with an "apostrophe. In this instance, a word contraction would be to say "don't" instead of "do not". Therefore, contraction in the text data needs to be expanded for better analysis.

3.1.2.2 Punctuations Elimination

Punctuation marks have no semantic meaning in text data, which can make it more challenging to distinguish between words and analyse the content. Punctuation elimination is the process of removing special characters and punctuation from a text document.

3.1.2.3 Case Change

Typically, words in text documents are written in capitalization case letters, lower case letters, and upper-case letters. For instance, the computer treats words like "Flower," "flower," and "FLOWER" are different even though they all have the same meaning. All of the words must be converted to lower case in order to reduce word duplication and text feature extraction approaches like accurate frequency counts and computation of tf-idf values.

3.1.2.4 Stop Words Elimination

The most often used terms in text documents are pronouns, articles, prepositions, and a few other words, even though these words don't always convey the meaning of the texts eg: “these”, “in”, “a”, “an”, “with”, “this”, “that”, and so on. Stop words are eliminated from documents in text summarization systems because the presence of stop words bulk up the text, are less important to analyst than other words, and are therefore not regarded as keywords.

3.1.2.5 Stemming and Lemmatization

Stemming is the method of reducing a word to its root stem. For instance, the words "work", "works", "working" and "worked" are all derived from the word "work". The basic concept of stemming is to remove the prefix or suffix from words like "ing", "s", "es", etc. Porter and Snowball stemmers are common stemming algorithms, but their effectiveness is limited. Lemmatization, preferred over stemming methods, systematically reduces words to their simplest form, offering a more accurate solution for word root extraction.

3.2 Feature Extraction

One usually works with a substantial amount of raw text data when attempting to address text processing-related challenges. Most important characteristic of these massive data sets is that they contain a large number of different variables. A lot of processing power is required to process these variables. The word tokens are produced after cleaning and normalizing of textual data. These tokens cannot be inputted directly into the machine for further processing, because the computer cannot process textual data. Therefore, since numbers are easy for the computer to process, decide to represent individual words numerically. The process of numerical representation of each token is known as feature extraction. It is the

technique of finding numerical features from the original dataset. The amount of redundant data in the data set can be reduced by feature extraction [27,28]. The various feature extraction techniques employed in this model are described in the following:

3.2.1 Bag-of-words

The Bag-of-words is a technique for extracting features from text that is useful for modelling and is a vocabulary matrix comprising all distinctive terms, where each row represents a sentence or document and each column represents a word. Moreover, it maintains a record of how often each word appears in a text document. Additionally, each sentence or document is represented as a fixed-length vector of term frequencies. The resulting matrix, which contains a lot of zero scores, is referred to as a sparse matrix or sparse representation. Sparse matrices require more memory and processing resources for modelling. This could affect the computational efficiency and make it challenging to interpret the outcomes.

3.2.2 TF-IDF

The word frequency count is a key component of the bag-of-words method. Therefore, while employing the bag-of-words approach, the words with high frequency are given more importance than the words with low frequency which is not necessarily true. To resolve this bag-of-words approach issue, the TF-IDF technique was put forward. It establishes the significance of a word within a gathering of documents or corpus that emphasizes rare words while ignoring common words. The frequency with which a word occurs in a document raises its TF-IDF value, while the number of documents in the corpus that contain the word lowers its TF-IDF value [29,30].

$$\text{TF-IDF} = \text{TF} * \text{IDF} \qquad \text{.....eq}^{n}\ 1$$

3.2.2.1 Term Frequency (TF)

The TF of a term reflects how important it is. If a term appears frequently in the text of a document, it is important.

$$tf(t,d) = N(t,d) \qquad \text{.....eq}^{n}\ 2$$

Where, *tf(t, d)* = term frequency for a term t in document d and *N(t, d)* = count of a term t in document d.

Longer texts will be assigned greater weight because of term frequencies. To avoid this issue, the term frequency might be standardized.

$$tf(t,d) = \frac{N(t,\ d)}{||D||} \qquad \text{.....eq}^{n}\ 3$$

Where, ||D|| = Total number of terms in the document.

3.2.2.2 Inverse Document Frequency (IDF)

The IDF of a term indicates percentage of corpus documents that contain the term. The IDF approach reduced the score for frequently occurring terms and increased the score for rare terms that were found in the corpus.

$$idf(t) = log\ (\frac{N}{df(t)}) \qquad \text{.....eq}^{n}\ 4$$

3.2.2.3 Cosine Similarity

The TF-IDF model determines the significance of a word to a document within a collection of documents. The TF-IDF method only takes into account words that are relevant and scores each word individually. Additionally, the TF-IDF ignores factors like word location, word order, semantic connections, and the context-specific meaning of words. More importantly, the TF-IDF approach can produce a very high dimensional feature space with sparse data, which could lead to issues with over-fitting as well as processing efficiency. For applications that use sparse data, cosine similarity is useful because it ignores 0-0 matches. The similarity scores would be inflated if 0-0 matches were taken into account in sparse data. The cosine similarity determines the degree of similarity between two vectors by computing the cosine of the angle between them.

$$\text{similarity} = \cos(\theta) = \frac{A\ .\ B}{||A||\ ||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} {A_i}^2}\sqrt{\sum_{i=1}^{n} {B_i}^2}}$$

....eq^{n} 5

where, A and B are the two vectors.

A.B = dot product between the two vectors.

A cosine similarity is recommended for determining the degree of similarity between any two sentences or text whose value ranges between 0 and 1. The angle between the two sentences is calculated by considering the two sentences as vectors. A cosine similarity of 0 would indicate that there are no similarities between the two sentences, while a cosine similarity of 1 suggests that the two sentences are precisely the same.

3.3 Dimensionality Reduction

As the number of features in a dataset increases, modeling becomes more challenging due to increased complexity, reducing the efficiency of learning algorithms and making data visualization, interpretation, and comprehension more difficult.

Sparsity factor may occur in large datasets where "sparsity" is used to refer to features, where many values are zero, exacerbate processing challenges and demand more storage space. To address this dimensionality problem, dimensionality reduction techniques are employed, preserving the essential features of the original dataset.

Cosine similarity is utilized to assess the similarity between sentences, aiding in text summarization. A higher cosine similarity suggests greater significance, indicating a sentence's potential inclusion in a summary. A threshold value of 0.15, determined through iterative experimentation, is employed. Sentences surpassing this threshold are considered for the summary, while those

exceeding a similarity score of 0.9 are deemed redundant, with the latter sentence being excluded to streamline the summary. These conditions act as a form of dimensionality reduction, crafting a concise candidate summary from the original document. This approach enables the extraction of salient sentences while discarding redundant or less significant ones, thus facilitating the creation of a comprehensive document summary.

3.4 Applying Modified PageRank Algorithm

A Modified PageRank [31] is primarily based on characteristics of the PageRank algorithm. PageRank is one of the most prominent and well-known algorithms used to determine the relevancy of a webpage. The fundamental idea behind PageRank is that a web page's significance is determined by the quantity and quality of web pages that point to it. To determine the significance of a webpage, PageRank employs a graph model in which Web pages are nodes and hyperlinks are edges. The PageRank score is calculated by using the equation (8)

$$PR(p_i) = \frac{1-d}{N} + d * \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)} \quad \ldots\ldots eq^n\ 6$$

Here, $PR(p_i)$ is the PageRank score of p_i, p_j is the adjacent node to p_i, d is the damping factor= 0.85, $M(p_i)$ is the set of pages that link to p_i, $L(p_j)$ is the number of outbound links on page p_j, N is the total number of pages ($p_1, p_2, p_3, \ldots, p_N$ are the pages) under consideration.

The Modified PageRank Algorithm is a modified version of the PageRank algorithm, which employs sentences of the documents in place of the web pages and also incorporates the weighted edges between nodes, such as sentences. The weights of the edges between nodes are computed using cosine similarity.

$$MPR(p_i) = \frac{1-d}{N} + d * \sum_{p_j \in M(p_i)} \frac{PR(p_j) * COSIM(p_i, p_j)}{N-1} \quad \ldots\ldots eq^n\ 7$$

Where, $MPR(p_i)$ calculates the new rank of node p_i, $PR(p_j)$ is the current rank of sentence p_j. The weight of the edge connecting two sentences is represented by $COSIM(p_i, p_j)$. The algorithm used to determine the Modified PageRank is shown below.

Input: Graph G.

Output: Scored Graph, G'

Procedure Modified PageRank(G)

Set d=0.85; *// damping factor*;

Set δ = 0.000001; *//threshold value*

Set N = Number of Nodes in G;

Set iter = 0, max_iter = 100; *// Maximum number of iterations in power method*

for all p_i in the graph **do**

oldPR[p_i] ← Count number of nouns in sentence p_i.

end for

// Compute the new rank of each sentence.

while iter < max_iter **do**

for all p_i in the graph **do**

$$newPR(p_i) = \frac{1-d}{N} + d * \sum_{p_j \in M(p_i)} \frac{PR(p_j) * COSIM(p_i, p_j)}{N-1}$$

if (newPR[p_i] - oldPR[p_i]) < δ

break;

end if

end for

oldPR ← newPR *//update PageRank*

iter ← iter + 1

end while

return G' *// scored graph*

end Procedure

3.5 Summary Extraction

Sentence scores are obtained after applying the modified PageRank algorithm on candidate summary document. The sentences are sorted based on the significance of the sentence score. Sentences with highest scores are given precedence. To obtain the summary of a specific text content, sentences are retrieved one at a time based on the rank or relevance of sentence score and added to the summary until the condition is satisfied. The compression ratio is used as a condition. A condition of 40% is employed in the suggested method for summarizing. This indicates that 40% of the original document's sentences will be involved in the summary which gives a condensed summary.

3.6 Proposed Algorithm

In this section the proposed approach is discussed in order to extract a meaningful summary from the single document. The input document D = {S_1, S_2, … , S_n}. Depending on the order in which the sentences appear in the input document, each sentence is assigned an index, such as S_i, where i range from 1 to n. In first step, the input document is splitted up into sentences, then tokenize each sentence into a collection of terms. The token was then put through a series of

normalization processes, such as the removal of all stop words, the conversion of uppercase to lowercase and then use the lemmatization process to determine each token's root words. The subsequent stage involves extracting the desired features by calculating each term's TF-IDF ($t_{tf\text{-}idf}$) and creating a TF-IDF matrix ($M_{tf\text{-}idf}$). Afterwards, the Cosine Similarity Matrix ($M_{cos\text{-}sim}$) is produced using the TF-IDF matrix ($M_{tf\text{-}idf}$). The similarity score between two sentences, S_i and S_j, is represented by the $M_{cos\text{-}sim}$[i][j] in the Cosine Similarity Matrix ($M_{cos\text{-}sim}$). To reduce the dimensionality of the original document D, read sentence S_i from it and add it to the candidate summary (i.e., $Summary_{candidate}$) if the similarity scores of two sentences, Si and Sj, are higher than 0.15 but less than 0.9. This $Summary_{candidate}$ serves as the basis for the document's summary. Furthermore, in order to reduce the dimensionality of $M_{cos\text{-}sim}$ matrix, remove the sentence S_j if the similarity score between two sentences, Si and Sj, is less than or equal to 0.15 (i.e., $M_{cos\text{-}sim}$[i][j] <= 0.15) or greater than or equal to 0.9 (i.e., $M_{cos\text{-}sim}$[i][j] >= 0.9). When two sentences have similarity scores of less than or equal to 0.15, one of the sentences is eliminated because it may contain terms that are uncommon or exceptional and hence not eligible for generation of summaries. When two sentences have similarity scores greater than or equal to 0.9, one of the sentences is removed since it is considered redundant. After removing a sentence, update the $M_{cos\text{-}sim}$ matrix. Subsequently a graph G is used to represent the $M_{cos\text{-}sim}$. Consider the graph G = (V, E, W), where V is the set of vertices that represents sentence E is the set of edges connecting V_i and V_j; W represents set of weights. A connection between vertices V_i and V_j is represented by an edge whose weight is $M_{cos\text{-}sim}$[i][j]. Initially, G is empty. In order to draw the graph, consider a vertex, V_i, and determine whether or not it belongs in V, If $V_i \notin V$, then add V_i into V. Similarly, If $V_j \notin V$, then add V_j into V . If the condition ($V_j \notin V$ && $V_i \neq V_j$) is true, then add the edge between V_i and V_j into E, and the $M_{cos\text{-}sim}$[i][j]. For every vertex V_i, the procedure is repeated until the final graph G is drawn. After that, apply the Modified Page Rank algorithm on graph G; the outcome is a scored graph G'. Each vertex V_i' is a component of the graph G'. The sentence S_i corresponds to the vertex V_i' and Each sentence S_i has a score, such as SC_i. To obtain the final summary, first reorder the sentences in the $Summary_{candidate}$ in decreasing order by using the obtained SC_i score. After that, select the first M sentences from $Summary_{candidate}$. The pseudo code of the proposed approach is provided below.

Input: D is the single document; M : size of the summary.

Output: D' is an M-size summary of document D.

// Preprocessing of input document

sent_tokenize() // Split input document into sentences

for each sentence S_i **do**

word_tokenize() // Normalization of each token

for each token Ti **do**

remove_stopwords(); // Remove all stop word

expand_contractions(); // Expand contraction

remove_punctuation(); // Remove punctuations from token

lower_token(); // Convert the token into lowercase

wordnet_lemmatizer(); // Lemmatize all tokens using WordNetLemmatizer

end for

end for

// Calculate the TF-IDF of each term ($t_{tf\text{-}idf}$) and generate a matrix ($M_{tf\text{-}idf}$)

for each processed tokenize sentence $S_i \in D$ **do**

// Compute the term frequency (*tf*) of each term

$tf(t, S_i) = \frac{N(t,S_i)}{||D||}$

// Compute idf of each term

$idf(t, D) = log\ (\frac{N}{df(t)})$

// Compute the *tf-idf* of each term

$tf\text{-}idf = tf(t, S_i) * idf(t, D)$

end for

// Calculate Cosine Similarity Matrix ($M_{cos\text{-}sim}$) from $M_{tf\text{-}idf}$ matrix

for (int i = 0; i < N; ++i) **do**

for (int j = 0; j < N; ++j) **do**

if (i == j) {

$M_{cos\text{-}sim}$[i][j] = 1.0;

}

// If two vectors, P and Q are not the same, such as P ≠ Q, then

else {

// Calculate the dot product of two vectors, P and Q

double dot_Product = 0.0;

for (int k = 0; k < N; ++k) {

dot_Product += $M_{tf\text{-}idf}$[i][k] * $M_{tf\text{-}idf}$[j][k];

}

// The magnitudes of vectors P and Q are magnitude_P and magnitude_Q, respectively.

double magnitude_P = 0.0, magnitude_Q = 0.0;

for (int k = 0; k < N; ++k) **do**

magnitude_P += ($M_{tf\text{-}idf}$[i][k])^2;

magnitude_Q += ($M_{tf\text{-}idf}$ [j][k])^2;

end for

magnitude_P = sqrt(magnitude_P);

magnitude_Q = sqrt(magnitude_Q);

// Determine the sentence-to-sentence cosine similarity.

$M_{cos\text{-}sim}$[i][j] = dot_Product / (magnitude_P * magnitude_Q);

}

end for

end for

// Reduce the dimensionality of $M_{cos\text{-}sim}$[i][j] and generate the $\text{Summary}_{\text{candidate}}$

for (int i = 0; i < N; ++i) **do**

for (int j = 0; j < N; ++j) **do**

if ($M_{cos\text{-}sim}$[i][j] > 0.15 || $M_{cos\text{-}sim}$[i][j] < 0.9) **do**

Read the sentence S_i from original document D, and then add it to $\text{Summary}_{\text{candidate}}$;

end if

if ($M_{cos\text{-}sim}$[i][j] <= 0.15 || $M_{cos\text{-}sim}$[i][j] >= 0.9) **do**

Remove the sentence S_j from $M_{cos\text{-}sim}$[i][j] in order to update the $M_{cos\text{-}sim}$[i][j];

end if

end for

end for

// Draw the graph G from $M_{cos\text{-}sim}$.

for (int i = 0; i < N; ++i) **do**

if ($V_i \notin G$) **do**

add V_i into V.

else

for (int j = 0; j < N; ++j) **do**

If ($V_j \notin G$ && $V_i \neq V_j$) **do**

Add the V_j into V;

Add the edge between V_i and V_j into E;

Add the $M_{cos\text{-}sim}$[i][j] as weight of the edge between V_i and V_j into W;

end if

end for

end if

end for

//The Modified PageRank algorithm is applied on graph G and the outcome is scored

// graph G'.

Modified PageRank(G)

// generate the final summary

for each sentence $S_i \in \text{Summary}_{\text{candidate}}$ **do**

Using the obtained SC_i score, rearrange the sentence S_i in decreasing order.

end for

for (int i = 0; i < M; ++i) **do**

Summary ← select S_i from $\text{Summary}_{\text{candidate}}$

end for

4. Results discussion and analysis

To evaluate the proposed summarizing approach, the experiment is carried out using the DUC-2002, DUC 2003, and DUC 2005 datasets because these datasets are frequently used by researchers for the study of information retrieval and text Summarization. The dataset is briefly described in subsection 4.1. The performance of the proposed summarizer has been compared to baselines and relevant research using the ROUGE metrics [29] such as ROUGE-1, ROUGE-2, ROUGE-L, in addition to Range and Coefficient of Variation (CV), which are most commonly employed automated evaluation tool in text summarizing [30]. The subsection 4.2 provides a quick overview of the evaluation metrics. The computational efficiency of the proposed algorithm has been compared with other benchmark algorithms, such as the TextRank and modified PageRank algorithms. The computational efficiency of the proposed method has been addressed in subsections 4.3 and 4.4, respectively, using graph analysis and result analysis.

4.1 Dataset Description

The suggested technique is evaluated using the standard datasets like DUC 2002, DUC 2003, and DUC 2005 from Text Retrieval Conference (TREC) website [32]. Table 1 provides an outline of the experimental datasets.

Table 1: An outline of the experimental datasets

Dataset description	**DUC 2002**	**DUC 2003**	**DUC 2005**
Number Of Documents	59	182	50
Documents Category	Weather	Economic	Legal
Number of documents taken for experiment purpose	6	6	6
Average Sentence Per Document	30	25	25
Summary Length(words)	200.3	150.8	176.5

The DUC 2002 contains 59 sets of documents connected to weather-related data, 6 of which were selected for experimentation. Here, the length of the average sentence in a document is 30, while the length of the summary is 200.3 words. The DUC 2003 contains 182 sets of documents related to an economics report, among which 6 documents are taken into consideration for analysis. The average sentence length per document is 25, and the summary contains 150.8 words. Similarly, DUC 2005 comprises 50 sets of legal related documents, however only 6 documents were chosen for this study. The average length per document has 25 words, while the summary has a length of 176.5 words.

4.2 Evaluation metrics

In order to evaluate the efficacy of this study, the text summary produced by the system and the reference summary created by the expert will be compared. The evaluation used the Recall-Oriented Understudy for Gusting Evaluation (ROUGE) metric [29], which determine degree of similarity between system-generated summaries and human generated summaries [28]. The ROUGE approach compares N-grams of generated summary with the reference summary [27]. For different angularities, the ROUGE-metric is calculated using equation (10).

$$ROUGE - N = \frac{\sum_{S \in (RefSum)} \sum_{N-gram \in (S)} Count_{match}\,(N-gram)}{\sum_{S \in (RefSum)} \sum_{N-gram \in (S)} Count\,(N-gram)} \quad \text{……..eq}^{n}\ 8$$

Where,

The generated and reference summaries are denoted by the letters *S* and *Refsum*, respectively.

N is the length of an N-gram, & is considered to have values 1 & 2.

$Count_{match}\,(N-gram)$ is the total number of N-gram matches identified in the reference and candidate summaries.

There are numerous ROUGE metrics variants, such as ROUGE-1, ROUGE-2, ROUGE-L. The ROUGE-1 is the uni-gram comparison, ROUGE-2 is the bi-gram comparison and ROUGE-L is a technique used to identify the longest common word sequence (LCS) in both the system generated summary and a reference summary. Three measurements are produced by the ROUGE evaluation: *recall*, *precision*, and *f-measure,* which is shown in eq^{n} 11, eq^{n} 12 and eq^{n} 13 respectively.

$$Precision = \frac{Cand_S \cap Ref_S}{m} \quad \text{……..eq}^{n}\ 9$$

$$Recall = \frac{Cand_S \cap Ref_S}{n} \quad \text{……..eq}^{n}\ 10$$

Rouge-L determines the weighted harmonic mean also known as the f-measure, by using the precision score and the recall score.

$$f - score = \frac{(1+\beta^2)(\text{Precision} * \text{Recall})}{Recall + \beta^2 * Precision} \quad \text{……..eq}^{n}\ 11$$

Here β is the ratio between *Precision* and *Recall*.

In addition to ROUGE scores, two other metrics like range and coefficient of variation (CV) are used. Range is the difference between the best ROUGE score and the worst ROUGE score, and it is represented by equation (14).

$$\text{Range} = \text{Rouge}_{\text{best}} - \text{Rouge}_{\text{worst}} \quad \text{……..eq}^{n}\ 12$$

CV is defined as the ratio of the Range to the average of the Rouge scores, as shown in equation (15).

$$\text{CV} = \frac{\text{Range}}{\text{Avg Rouge}} * 100 \quad \text{……..eq}^{n}\ 13$$

4.3 Graph Analysis

To evaluate the proposed approach, the DUC 2002, DUC 2003, and DUC 2005 datasets are used. Consider the graph G = (V, E, W), where V: set of vertices that corresponds to a sentence; E: set of edges connecting vertices V_i and V_j; W: set of weights. A connection between vertices V_i and V_j is represented by an edge whose weight w_i. Figure 3 shows that our proposed method produces simple graph with lesser number of connected edges as compared to other existing techniques for DUC 2002 dataset.

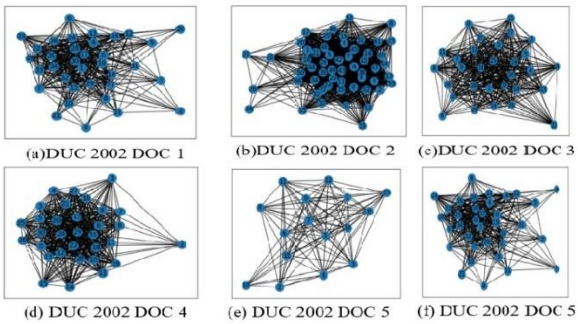

Fig 2: Applying PageRank Algorithm on DUC 2002 dataset.

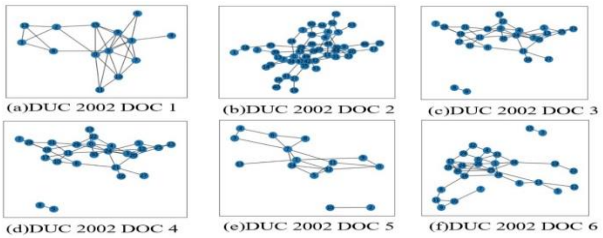

Fig 3: Applying Improved PageRank Algorithm on DUC 2002 dataset.

Table 2: Rouge scores of Text Rank, modified PageRank and Improved-PageRank Algorithm.

		TextRank			Modified PageRank			Improved-PageRank (Proposed Technique)		
		Rouge 1	Rouge 2	Rouge L	Rouge 1	Rouge 2	Rouge L	Rouge 1	Rouge 2	Rouge L
DUC 2002	DOC 1	0.670	0.520	0.581	0.682	0.553	0.682	0.833	0.704	0.833
	DOC 2	0.680	0.530	0.633	0.660	0.544	0.647	0.695	0.586	0.683
	DOC 3	0.700	0.585	0.685	0.720	0.625	0.720	0.892	0.726	0.829
	DOC 4	0.654	0.501	0.642	0.664	0.502	0.690	0.794	0.671	0.794
	DOC 5	0.710	0.612	0.647	0.783	0.631	0.703	0.849	0.748	0.847
	DOC 6	0.640	0.607	0.669	0.735	0.624	0.735	0.850	0.760	0.850
DUC 2003	DOC 1	0.674	0.521	0.598	0.706	0.580	0.699	0.787	0.687	0.781
	DOC 2	0.612	0.567	0.645	0.709	0.610	0.703	0.776	0.675	0.773
	DOC 3	0.587	0.489	0.601	0.694	0.567	0.694	0.800	0.676	0.800
	DOC 4	0.592	0.498	0.594	0.652	0.539	0.652	0.709	0.593	0.709
	DOC 5	0.623	0.567	0.604	0.719	0.622	0.717	0.813	0.717	0.813
	DOC 6	0.641	0.581	0.676	0.700	0.606	0.700	0.725	0.640	0.725
DUC 2005	DOC 1	0.642	0.573	0.621	0.699	0.624	0.699	0.884	0.801	0.884
	DOC 2	0.571	0.527	0.569	0.654	0.607	0.654	0.944	0.881	0.944
	DOC 3	0.605	0.601	0.672	0.743	0.645	0.743	0.892	0.806	0.892
	DOC 4	0.613	0.547	0.581	0.679	0.624	0.677	0.798	0.724	0.798
	DOC 5	0.658	0.609	0.643	0.714	0.637	0.710	0.849	0.787	0.849
	DOC 6	0.583	0.578	0.528	0.686	0.609	0.683	0.725	0.650	0.725

4.4 Result Analysis

This subsection delivers a thorough evaluation of the proposed approach. The performance of improved-PageRank algorithm has been evaluated in comparison with well-known, techniques for text summarization, like modified PageRank and TextRank for DUC 2002, DUC 2003, and DUC 2005 dataset and ROUGE-1, ROUGE-2 and ROUGE-L scores are measured. A more thorough discussion of the outcomes analysis is provided in Table 2.

The proposed technique produces better results, as demonstrated in Figure 4(a),4(b) and 4(c) when the ROUGE-1, ROUGE-2 and ROUGE-L value is compared to TextRank and modified PageRank using the DUC 2002 dataset. The improved-PageRank consistently performs better than the other approaches. The ROUGE-L values of Improved-PageRank, TextRank, and modified PageRank vary from 68.3% to 85%, 58.1% to 68.5%, and 64.7% to 73.5%, respectively. These data show that the improved-PageRank performs noticeably better than existing methods.

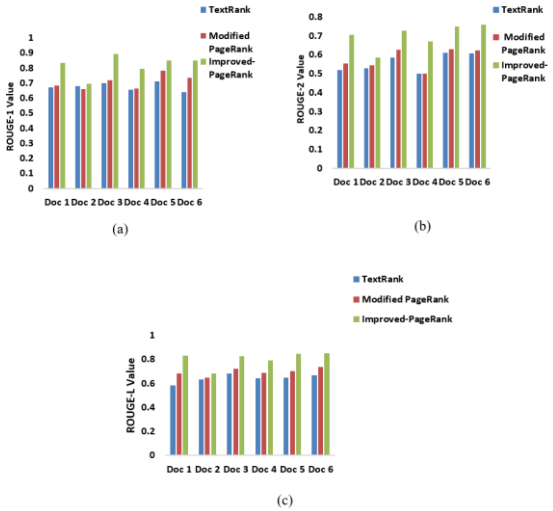

Fig 4: For the DUC 2002 dataset, (a) Rouge-1, (b) Rouge-2, and (c) Rouge-L scores for Text Rank, modified PageRank, and the Improved-PageRank algorithm.

In the context of the DUC 2003 dataset, it becomes evident that the ROUGE-1 score for the proposed method ranges from 70.9% to 81.3%, higher than that of TextRank and modified PageRank, which have ROUGE-1 scores that range from 58.7% to 67.4% and 65.2% to 71.9%, respectively. In Figure 5(a), this comparison is shown graphically. Table 2 of the DUC 2003 dataset shows that the ROUGE-2 and ROUGE-L scores of improved-

PageRank often outperform the results of the standard approaches, as shown in Figures 5(b) and 5(c).

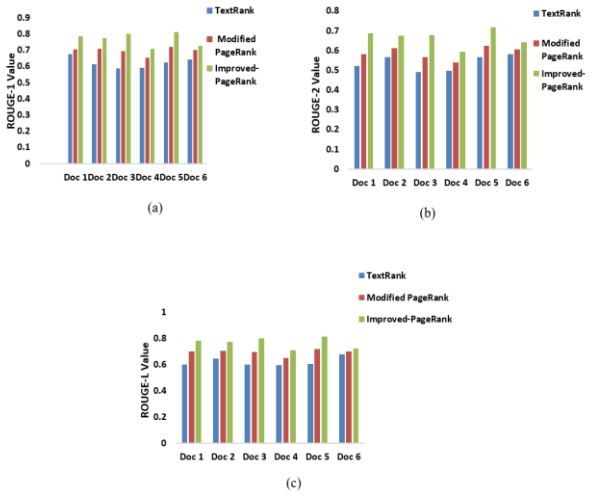

Fig 5: For the DUC 2003 dataset, (a) Rouge-1, (b) Rouge-2, and (c) Rouge-L scores for Text Rank, modified PageRank, and the Improved-PageRank algorithm

For the DUC 2005 dataset, the ROUGE-1 and ROUGE-L scores for improved-PageRank algorithm are identical, which range from 72.5% to 94.4%. With respect to the DUC 2005 dataset, the proposed approach performs better than the other well-established approaches as shown in Figure 6(a), 6(b), and 6(c), respectively.

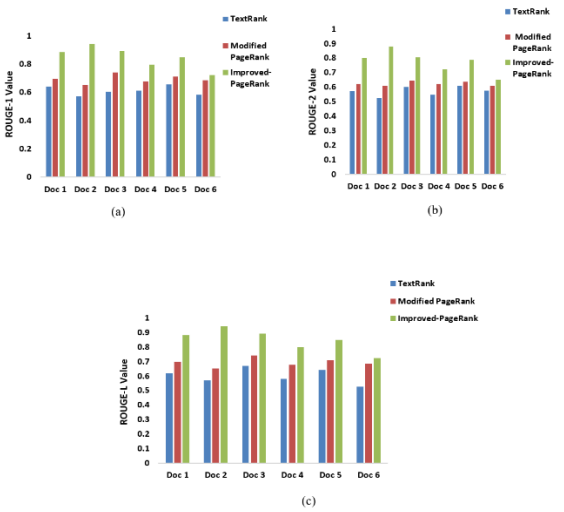

Fig 6: For the DUC 2005 dataset, (a) Rouge-1, (b) Rouge-2, and (c) Rouge-L scores for Text Rank, modofied PageRank, and the Improved-PageRank algorithm.

Table 3 presents an analysis of the TextRank, modified PageRank, and Improved-PageRank algorithms in terms of several metrics: best case, worst case, average case performance, range, and coefficient of variation (CV) for ROUGE-1, ROUGE-2, and ROUGE-L scores across the specified datasets. In the context of DUC 2002, the proposed approach performs strongly in the best-case, worst-case, and average-case scenarios when compared to TextRank and modified PageRank for ROUGE-1, ROUGE-2, and ROUGE-L, respectively, as shown in Figure 7(a).

Table 3: Analysis of Text Rank, modified PageRank and improved-PageRank Algorithm on DUC 2002, DUC 2003 and DUC 2005

		TextRank					**Modified PageRank**					**Improved-PageRank (Proposed Technique)**				
		Best case	Worst case	Average case	Range	CV	Best case	Worst case	Average case	Range	CV	Best case	Worst case	Average case	Range	CV
DUC 2002	Rouge 1	0.710	0.640	0.675	0.070	10.37	0.783	0.660	0.707	0.123	17.39	0.892	0.695	0.818	0.197	24.08
	Rouge 2	0.612	0.501	0.559	0.111	19.85	0.631	0.502	0.579	0.131	22.62	0.760	0.586	0.699	0.174	24.89
	Rouge L	0.685	0.581	0.642	0.104	16.19	0.735	0.647	0.696	0.088	12.64	0.850	0.683	0.798	0.167	20.92
DUC 2003	Rouge 1	0.674	0.587	0.621	0.087	13.24	0.719	0.652	0.696	0.067	9.62	0.813	0.709	0.768	0.104	13.54
	Rouge 2	0.581	0.489	0.537	0.092	17.13	0.622	0.539	0.587	0.083	14.13	0.717	0.593	0.664	0.124	18.67
	Rouge L	0.676	0.594	0.619	0.082	10.06	0.717	0.652	0.694	0.065	9.36	0.813	0.709	0.766	0.104	13.57
DUC	Rouge 1	0.658	0.571	0.612	0.087	14.21	0.743	0.654	0.695	0.089	12.80	0.944	0.725	0.848	0.219	25.82

2005	Rouge 2	0.609	0.527	0.572	0.082	14.33	0.645	0.607	0.624	0.038	6.08	0.881	0.650	0.775	0.231	29.80
	Rouge L	0.672	0.528	0.602	0.144	23.92	0.743	0.654	0.694	0.089	12.82	0.944	0.725	0.848	0.219	25.82

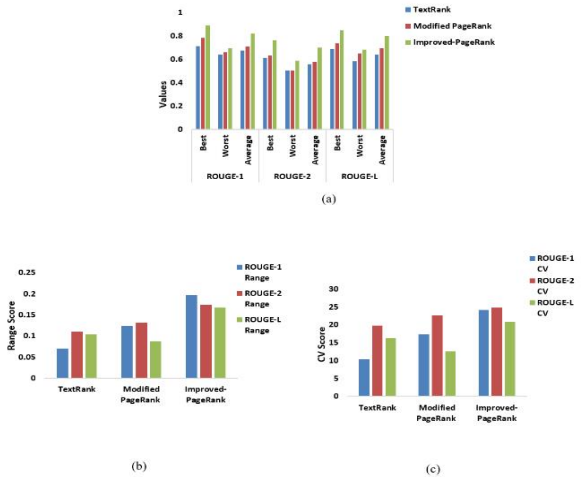

Fig 7: For the DUC 2002 dataset, (a) Analysis at best, worst and average case, (b) Range scores, (c) CV scores for Text Rank, modified PageRank, and the Improved-PageRank algorithm.

In the case of ROUGE-1, the range value of TextRank is a minimal (0.070) when compared to the range values of modified PageRank (0.123) and the approach that is proposed (0.197). In relation to ROUGE-2, the range value of the Improved-PageRank is larger (0.174) than to both modified PageRank (0.131) and TextRank (0.111). However, for ROUGE-L, the modified PageRank (0.088) method has a smaller range value than Improved-PageRank (0.167) and TextRank (0.104). The range value analysis for the DUC 2002 dataset is shown graphically in Figure 7(b). The coefficient variation (CV) value of the proposed technique for ROUGE-1, ROUGE-2 and ROUGE-L are 24.08, 24.89 and 20.92, respectively. More specifically, the CV of the proposed method for ROUGE-2 is substantially greater than the CV values for ROUGE-1 & ROUGE-L. A graphic representation of the CV analysis is presented in Figure 7(c). It is evident from Figure 7(c) that the CV value for the proposed technique performs better than TextRank & modified PageRank.

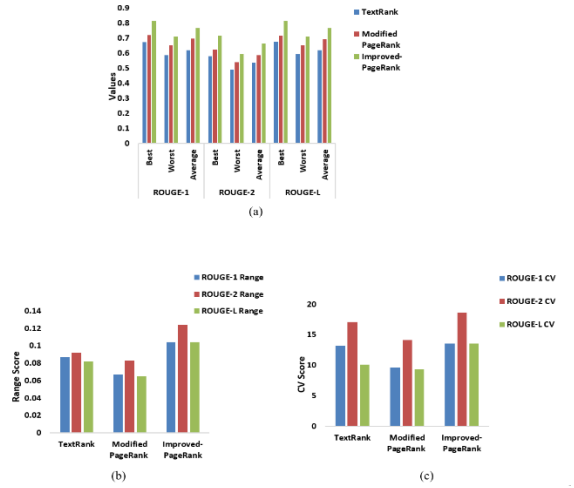

Fig 8: For the DUC 2003 dataset, (a) Analysis at best, worst and average case, (b) Range scores, (c) CV scores for Text Rank, modified PageRank, and the Improved-PageRank algorithm.

In the DUC 2003 datasets, both improved PageRank and modified PageRank produce significantly superior results than TextRank in all circumstances when compared to the best-case analysis. Additionally, it is noted that in the best-case analysis, the proposed method produces an identical value of 0.813 for ROUGE-1 and ROUGE-L, which is greater than the ROUGE-2 score of 0.717. Furthermore, the average- and worst-case performance of proposed approach consistently beats that of other existing methods in all scenarios which is displayed in figure 8(a). The table shows that for both ROUGE-1 and ROUGE-L, the range value of proposed model is identical, specifically 0.104. Figure 8(b), which displays the analysis of the range value, demonstrates that Improved-PageRank performs better than TextRank and modified PageRank. It is visible that the coefficient of variation (CV) for modified PageRank in the case of ROUGE-1 and ROUGE-L is notably lower, at 9.62 and 9.36, respectively, compared to the CV scores of TextRank and the improved-PageRank. In case of ROUGE-2, the proposed model achieves a higher CV score at 18.67, outperforming both TextRank (17.13) and modified PageRank (14.13). Figure 8(c) provides a graphic representation of these data.

In the DUC 2005 dataset, it is observed that the best-case, worst-case, and average-case analyses of the ROUGE-1 and ROUGE-L metrics produce the identical results for the improved-PageRank algorithm. Notably, the best-case analysis achieves 0.944. Furthermore, which is shown in figure 9(a). According to the table, the modified PageRank range value for ROUGE-2 is 0.038, which is lower than the

improved-PageRank (0.231) and TextRank (0.082). The range value analysis is shown in Figure 9(b). The table reveals that the coefficient of variation (CV) value for the proposed method, specifically for ROUGE-2 at 29.80, is greater than that of other established methods. Figure 9(c) presents a graphic representation of the analysis of CV scores, which shows that the proposed method is performing better than TextRank and modofied PageRank in all scenarios.

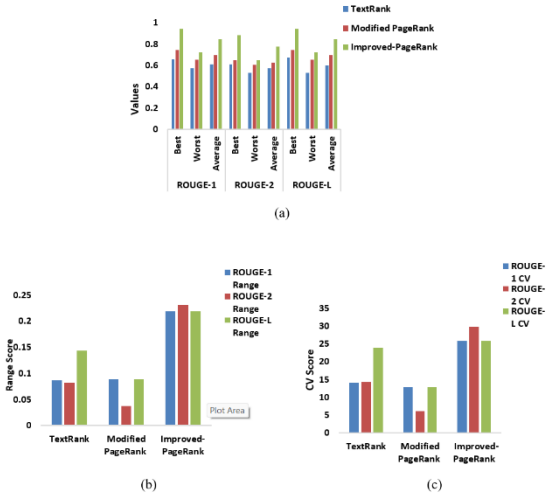

Fig 9: For the DUC 2005 dataset, (a) Analysis at best, worst and average case, (b) Range scores, (c) CV scores for Text Rank, modified PageRank, and the Improved-PageRank algorithm.

5. Conclusion and Future Work

Text summaries are being considered to be quite beneficial for readers to quick understand the main idea of lengthy texts while also save their time and effort. In the modern digital world, extracting high-quality keywords from the vast volume of web/document data has become very challenging. In this paper, we introduce a novel graph-based text summarization technique, referred to as the improved PageRank algorithm, which adeptly reduces the size of text while preserving its integrity. This work focuses on extractive text summarization from a single document. This study aims to effectively address the difficulties associated with text rank and modified PageRank approaches. Furthermore, the proposed method is evaluated against TextRank and Modified PageRank, using the Document Understanding Conference dataset (DUC 2002, DUC 2003, DUC 2005). To identify closely related salient features within pre-processed documents, multiple feature extraction techniques have been employed, including TF-IDF and cosine similarity. Additionally, dimensionality reduction has been simulated using a precise threshold condition which ensures non-redundancy in the generated summaries. The performance of the proposed summarizer algorithm has been compared to performance of TextRank and Modified PageRank using the ROUGE metrics, such as ROUGE-1, ROUGE-2, ROUGE-L, in addition to the Range and Coefficient of Variation (CV). The obtained results demonstrate that the proposed algorithm outperforms existing methods in terms of accuracy on several metrics, including Coefficient of Variance, Range, and ROUGE scores. In the future, we will focus on extractive text summarizing technique, which summarizes information from multiple documents.

Author contributions

Jyotirmayee Rautaray: Conceptualization, Methodology, Writing-Original draft preparation **Sangram Panigrahi:** Visualization, Investigation, Writing-Reviewing and Editing, **Ajit Kumar Nayak:** Data curation, Software Validation, software Field study.

Conflicts of interest

The authors declare no conflicts of interest.

Reference

[1] Nenkova, A., & McKeown, K. (2012). A survey of text summarization techniques. *Mining text data*, 43-76

[2] El-Kassas, W. S., Salama, C. R., Rafea, A. A., & Mohamed, H. K. (2021). Automatic text summarization: A comprehensive survey. *Expert systems with applications*, *165*, 113679

[3] Nenkova, A., & McKeown, K. (2012). A survey of text summarization techniques. *Mining text data*, 43-76

[4] El-Kassas, W. S., Salama, C. R., Rafea, A. A., & Mohamed, H. K. (2020). EdgeSumm: Graph-based framework for automatic text summarization. *Information Processing & Management*, *57*(6), 102264

[5] Patil, S. P., &Rautray, R. SMATS: Single and Multi Automatic Text Summarization. *Karbala International Journal of Modern Science*, *9*(1), 6

[6] Saini, N., Saha, S., Jangra, A., & Bhattacharyya, P. (2019). Extractive single document summarization using multi-objective optimization: Exploring self-organized differential evolution, grey wolf optimizer and water cycle algorithm. *Knowledge-Based Systems*, *164*, 45-67

[7] Erkan, G., & Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, *22*, 457-479.

[8] Fang, C., Mu, D., Deng, Z., & Wu, Z. (2017). Word-sentence co-ranking for automatic extractive text summarization. *Expert Systems with Applications*, *72*, 189-195.

[9] Goldstein, J., Mittal, V.O., Carbonell, J.G., Kantrowitz, M., 2000. Multi-document summarization by sentence extraction, in. In: NAACL-ANLP 2000 Workshop: Automatic Summarization

[10] Mallick, C., Das, A. K., Dutta, M., Das, A. K., & Sarkar, A. (2019). Graph-based text summarization using modified TextRank. In *Soft Computing in Data Analytics: Proceedings of International Conference on SCDA 2018* (pp. 137-146). Springer Singapore

[11] Fatima, Q., &Cenek, M. (2015, August). New graph-based text summarization method. In *2015 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)* (pp. 396-401). IEEE

[12] Patil, V., Krishnamoorthy, M., Oke, P., &Kiruthika, M. (2004). A statistical approach for document summarization. *Department of Computer Engineering Fr. C. Rodrigues Institute of Technology, Vashi, Navi Mumbai, Maharashtra, India*

[13] Rautray, R., Balabantaray, R. C., & Bhardwaj, A. (2015). Document summarization using sentence features. *International Journal of Information Retrieval Research (IJIRR)*, *5*(1), 36-47

[14] Yu, S., Su, J., Li, P., & Wang, H. (2016). Towards high performance text mining: a TextRank-based method for automatic text summarization. *International Journal of Grid and High-Performance Computing (IJGHPC)*, *8*(2), 58-75

[15] Liu, Z., Li, P., Zheng, Y., & Sun, M. (2009, August). Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of the 2009 conference on empirical methods in natural language processing* (pp. 257-266)

[16] Castillo, E., Cervantes, O., &Vilarino, D. (2017). Text analysis using different graph-based representations. *Computación y Sistemas*, *21*(4), 581-599

[17] 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

[18] Manjari, K. U. (2020, October). Extractive summarization of Telugu documents using TextRank algorithm. In *2020 Fourth international conference on I-SMAC (IoT in social, mobile, analytics and cloud)(I-SMAC)* (pp. 678-683). IEEE

[19] Barrios, F., López, F., Argerich, L., &Wachenchauzer, R. (2016). Variations of the similarity function of textrank for automated summarization. *arXiv preprint arXiv:1602.03606*

[20] Mihalcea, R., &Tarau, P. (2004, July). Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing* (pp. 404-411)

[21] Erkan, G., &Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, *22*, 457-479

[22] Li, J., Huang, G., Fan, C., Sun, Z., & Zhu, H. (2019). Key word extraction for short text via word2vec, doc2vec, and textrank. *Turkish Journal of Electrical Engineering and Computer Sciences*, *27*(3), 1794-1805

[23] Yadav, A. K., Ranvijay, Yadav, R. S., & Maurya, A. K. (2023). Graph-based extractive text summarization based on single document. Multimedia Tools and Applications, 1-27

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

[25] Hua, Z., Fei, L., & Jing, X. (2023). An improved risk prioritization method for propulsion system based on heterogeneous information and PageRank algorithm. *Expert Systems with Applications*, *212*, 118798.

[26] He, S., Guo, F., & Zou, Q. (2020). MRMD2. 0: a python tool for machine learning with feature ranking and reduction. *Current Bioinformatics*, *15*(10), 1213-1221.

[27] Kadriu, K., & Obradovic, M. (2021). Extractive approach for text summarisation using graphs. *arXiv preprint arXiv:2106.10955*.

[28] Chin-Yew Lin and E.H. Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In Proceedings of 2003 Language Technology Conference (HLT-NAACL 2003), Edmonton, Canada

[29] Lin, C. Y. (2004). Rouge: A package for automatic evaluation of summaries. Text summarization branches out. 74–81.

[30] Fabbri, A. R., Kryściński, W., McCann, B., Xiong, C., Socher, R., & Radev, D. (2021). Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, *9*, 391-409.

[31] Gräßler, I., Thiele, H., Oleff, C., Scholle, P., & Schulze, V. (2019, July). Method for analysing requirement change propagation based on a modified pagerank algorithm. In *Proceedings of the Design Society: International Conference on Engineering Design* (Vol. 1, No. 1, pp. 3681-3690). Cambridge University Press.

[32] Text Retrieval Conference (TREC) website : https://trec.nist.gov