Section A-Research paper
ISSN 2063-5346

IMPROVED DRIVEN TEXT SUMMARIZATION USING PAGERANKING ALGORITHM AND COSINE SIMILARITY

Dr.S. Gunasundari [1], M. Jenifer Shylaja[2], Dr.S. Rajalaksmi[3], Mrs.KC. Aarthi[4]

Abstract

This study describes a system that uses NLP-based summarizing to streamline information retrieval. In order to extract the most pertinent and instructive sentences from the input text, our suggested approach makes use of features such as cosine similarities, PageRanking duplicate word, concentrating mapping, data clearing, stop phrase removal, word count. The methodology entails preparing the input text, determining each sentence's cosine similarity to the raw text, and sorting the sentences according to their word counts. Our approach achieves great accuracy in summarizing the major ideas of the documents while keeping coherence and readability when tested on a sample of various texts and review based on customers option. Unsupervised learning techniques like Text rank are employed for extracting summarization of texts. Typically, text summaries are produced solely using the text rank algorithm. However, in this study, summaries are extracted using the text rank algorithm in conjunction with cosine similarity. Our method is able to recognize significant and succinct statements because a word count and cosine similarities scores for the top-ranked phrases are 0.46. Overall, by quickly and accurately summarizing vast amounts of text using NLP approaches, our suggested system can greatly increase the effectiveness of information retrieval tasks.

Keywords— Automatic text summarization, Natural language processing, Ranking, Summarization techniques.

[1] Asst.Professor, Department of Computer Science and Engineering, Velammal Engineering College, Chennai

[2] PG Scholar, Department of Computer Science and Engineering, Velammal Engineering College, Chennai

[3]Assistant Professor, Department of Computer Science and Engineering, Velammal Engineering College, Chennai

[4]Assistant Professor, Department of Computer Science and Engineering, Velammal Engineering College, Chennai

I. Introduction

Natural language processing (NLP) relies heavily on automatic text summarizing, which entails creating a condensed version of a given text while maintaining its key points [1]. The exponential expansion of digital content has made it more and more difficult for consumers to effectively extract pertinent information. This issue is particularly severe in professions that depend on the capacity to swiftly comprehend and absorb huge amounts of material, such as journalism, medicine, and finance. The two kinds of text summarizing approaches are extraction and abstractive summarization. Selecting the key phrases or paragraphs from the source text and condensing them into a concise summary is known as extractive summarization. Abstractive summary entails rewriting and reorganizing the source material to provide a greater brevity. The concentrate concentrates on extractive summarization using cosine similarity, which is a popular method in natural language processing, which has been denoted in [2].

By calculating the cosine value of a given angle between two vectors, a metric known as cosine correspondence can be employed to compare the two vectors. When an essay has been summarized, the cosine relationship is used to determine how similar words or paragraphs are to one another. To understand how cosine similarity works, consider both vectors A and B. By multiplying the dots in the products of the two matrices by the total of the individual magnitudes, A and B's cosine correlation is determined. It is necessary to express phrases or paragraphs as vectors to compute the cosine similarity between them. Although there are several ways for vectoring text, the TF-IDF and bag-of-words approaches are the most often used ones. A phrase or paragraph is represented as a vector of word counts via the bag-of-words approach. This approach treats each word as a distinct feature, with the value of each feature being the number of words in the phrase or paragraph. The TF-IDF approach, on the other hand, gives words weights depending on how frequently they occur in the text and the corpus [3]. The top sentences or paragraphs will then be chosen to be included in the summary after we rate them according that. The effectiveness of text summarization algorithms is frequently assessed using indicators of performance metrics including accuracy, recall, and F1 score to gauge how comparable the created and reference summaries are to one another. By comparing the generated summary to a set of reference summaries, Bleu gauges the quality of the summary.

In verdict, the integration of Pytext with Pegasus provides a potent text summarizing solution that can be utilized in a variety of fields and application situations. Text summarization may be made more effective, accurate, and available to a larger variety of users with the use of deep learning models and performance measures.

Text summarization retrieval means generating a shorter version of a text while retaining its important information. It helps users to quickly extract relevant information from large volumes of text data, saving time and increasing efficiency.

The system's objective is to speed up the data retrieval process by giving users accurate and succinct summaries of lengthy texts, which will save them time and effort. The uses of this technology can be beneficial in a variety of industries, including journalism, law, and medicine, where experts must quickly analyse huge volumes of text.

The remaining sections of this paper are organized as follows. Section 2 provides a brief overview of the related work. Section 3 presents a review of the methodology used in the system, detailing the algorithm employed. Section 4 provides a comprehensive description of the proposed system, including the pseudo code of the algorithm. Section 5 discusses the implementation of the proposed system. Section 6 presents the experimental design and the results obtained from the proposed system. Finally, Section 7 concludes the work by providing a summary and suggestions for future improvements.

II. Literature Review

Pal et al. (2022) proposed a method based on combining cosine similarity and natural language processing to summarize scholarly literature on the coronavirus. The COVID-19 Open Research Dataset (CORD-19), which has 236 336 academic full-text articles as of July 19, 2021, is used in the following study. To manage many articles, the system used tokenization, N-gram extraction, and part-of-speech tagging. The topic's keywords were automatically identified, and the corresponding information has been condensed and presented to the user using cosine similarity. Researchers, medical experts, and other participants may gain from the system. Additionally, it may be trained on datasets from many areas to produce relevant summaries for diverse user groups, like students [4].

Adwani et al. (2021) created a BERT- and PageRank-based system for extractive text summarization. The system recognizes the keywords in the query automatically and then applies cosine similarity to condense the pertinent data before displaying it to the user. Researchers, medical experts, and other individuals might find this study useful. The same service may also be used to train on datasets from various domains (like education) to give a pertinent summary for other user groups, like students. The extracted summary is reliant on compression ratio, taking into account minimizing redundancy based on sentence overlapping, and the PageRank approach uses a customizable number of rounds to obtain the number that produces the best summary results [5].

Pokharkar et al. (2022) implemented extractive and abstractive text summarization approaches, using the linguistic and statistical properties of the sentences, the extractive summary technique chooses relevant phrases, paragraphs, etc. from the original text and concatenates them into a shorter form. The abstractive summary approach, on the other hand, uses linguistic approaches to study and analyse the text before identifying new concepts and phrases to produce fresh, succinct copy that successfully communicates. The authors suggested utilizing NLP to create a Text Summarizer [6].

Senthamizh and Arutchelvan (2022) talks about the newly emerging area of text summarizing (TS), which permits the production of summaries from one or more sources. TS does not need to be semantic to be organized as an exposition to gather accents from its specific location, and the usage of non-ASCII characters, pronunciation, tokenizing, and lemmatization result in the development of a summary. The authors suggest an automated text summarizing method that makes use of named entity recognition and document clustering. [7].

Reddy et al. (2022) proposed a method of document summarization that uses a modified PageRank algorithm to efficiently condense long documents without leaving out crucial information. Rouge-N and Rouge-L were utilized to analyse the experimental data, while more complex natural language processing techniques were used in the study to evaluate and summarize the material. The results showed that, with an F1 measure of 76.85%, the suggested model performed better than industry-recognized methods. [8].

Adhika Pramita Widyassari Et al. (2022) has proposed a Review of automatic text summarization techniques & methods. On the Journal of King Saud University - Computer and Information SciencesVolume 34, Issue 4, April 2022, Pages 1029-1046.Text summarization automatically produces a summary containing important sentences and includes all relevant important information from the original document. One of the main approaches, when viewed from the summary results, are extractive and abstractive. An extractive summary is heading towards maturity and now research has shifted towards abstractive summation and real-time summarization. Although there have been so many achievements in the acquisition of datasets, methods, and techniques published, there are not many papers that can provide a broad picture of the current state of research in this field. The results of the analysis provide an in-depth explanation of the topics/trends that are the focus of their research in the field of text summarization [9].

Minakshi Tomer & Manoj Kumar (2022) has implemented Multi-document extractive text summarization based on firefly algorithm. Extracting relevant information from a large amount of data is a challenging task. Automatic text summarization is a potential solution for obtaining this information. In this paper, a nature inspired swarm intelligence-based algorithm viz. firefly algorithm for multi-document text summarization is proposed. The performance of the algorithm has been evaluated using ROUGE score. The performance of the proposed algorithm is compared with some other nature inspired ones such as particle swarm optimization (PSO) and genetic algorithm (GA)[10].

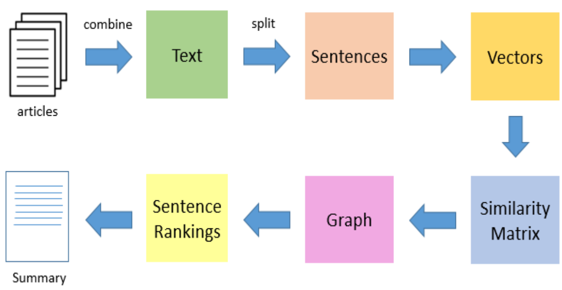
Sharaff, et al (2022) has proposed a study on Feature based cluster ranking approach for single document summarization. Text Summarization is a process of creating gist of large set of documents. It creates a summary which depicts the overall information contained in large text documents in a short and accurate way. A model for generating single document text summarization is presented in this paper. This model is based on extractive summarization. The proposed work extracts the informative features and generates the scoring of sentences by using similarity measure technique. Once the score of sentences is generated then clusters of sentences are formed. Clusters and sentences in each cluster are ranked and highly ranked sentences from each cluster of relative importance are included in the final summary [11].

Divakar Yadav Et al. (2022) has implemented Qualitative Analysis of Text Summarization Techniques and Its Applications in Health Domain.For the better utilization of the enormous amount of data available to us on the Internet and in different archives, summarization is a valuable method. We implemented five different algorithms, namely, term frequency-inverse document frequency (TF-IDF), LexRank, TextRank, BertSum, and PEGASUS, for a summary generation. After performing a qualitative analysis of the above algorithms, we observe that for both the datasets, i.e., Reddit-TIFU and MultiNews, PEGASUS had the best average F-score for abstractive text summarization and TextRank algorithms for extractive text summarization, with a better average F-score[12].

Overall, the literature survey on the proposed topic reveals that the problem of various summarization techniques is not completely addressed, necessitating the development of an effective algorithm to improve the performance of text summarization in an effective manner. In this paper, a system is introduced that utilizes NLP-based summarization to streamline information retrieval.

III. Methodology

In this research, a model was put out by fusing two approaches to automatic text summarization. Text rank algorithm and cosine similarity are the methods used to summarize the text. A cosine similarity approach evaluates the cosine angle across papers that consist of groups of sentences or vectors of one phrase (non-zero vector) to determine how similar they are. In order to use cosine similarity to assess sentence similarity, we structured every phrase in this paper as a vector. The phrases contained in the matrices are comparable if the angle determined using the cosine inverse technique equals zero. As shown in figure 1 describes the general workflow of the text summarization method.



1

. Text Summarization Flow

ISSN 2063-5346

cosine_similarity (A, B) = A * B / ||A|| * ||B||

-------------------(1)

Where ||A|| and ||B|| represent the Euclidean norms of the two vectors, A and B stand for the two vectors that represent the documents, * denotes the dot product of the two vectors, and A and B. The vectors are often constructed in the context of automatic text summarization by modelling each phrase or document as a bag-of-words model, where the frequency of each word in the document is utilized as a feature in the vector.

Unsupervised learning techniques like Text rank are employed for extracting summarization of texts. Typically, text summaries are produced solely using the text rank algorithm. However, in this study, summaries are extracted using the text rank algorithm in conjunction with cosine similarity.

3.1 PageRank Algorithm

The PageRank Algorithm are used by Google Search to rank websites in their search engine results. After Larry Page, one of the Google founders, created PageRank, A way to gauge the significance of internet pages is through PageRank. The PageRank algorithm generates a probability distribution that is used to illustrate the possibility that any given page would be reached by a random link clicker. Any number of documents in a collection can be used to calculate PageRank. Several research publications make the supposition that, at the start of the computational process, the distribution is distributed equally among all documents in the collection. A number of "iterations"—or passes—through the collection are needed for the PageRank computations in order to modify the approximative PageRank values so that they more closely resemble the theoretical true value.

3.2 TextRank Algorithm

Natural Language Processing uses TextRank, a text summarization algorithm, to produce document summaries. TextRank is an unsupervised graph-based text summarizing method that employs an extractive approach. Search engines like Google employ the PageRank algorithm to determine the ranking of online sites. The PageRank algorithm is the foundation of TextRank.

The PageRank method is used in particular by the Python package PyTextRank to rate textual content. It is useful for operations like extracting keywords and summarizing. PyTextRank is used in information retrieval using NLP-based summary to rank phrases according to their significance in the source material, which is essential for summary. A more precise and succinct summary can be produced by the system by using PyTextRank to recognize and extract the important lines that express the document's core idea. There are many different sectors and use cases that can benefit from the Improved Driven Text Summarization utilizing Pytext and Pegasus. Using Pytext entails optimizing pre-trained models like Pegasus on certain datasets.

3.3 Pegasus

A cutting-edge pre-trained transformer model created by Google with a focus on abstractive text summarization is called PEGASUS. It has been demonstrated to be very successful in producing excellent summaries of lengthy materials.

The PEGASUS model can be used to create succinct, accurate summaries of lengthy documents within the framework of the Streamlining Information Retrieval using NLP-Based Summarization project while maintaining the content and context of the original text. Users can rapidly scan the summary in order to understand the main ideas of the article before selecting whether to read the complete document or not, which can increase the efficiency and efficacy of information retrieval operations. Combining Pytext with Pegasus offers a potent text summarizing solution that may be used in a variety of fields and use situations. Users may use Pytext to fine-tune.

As shown in figure 2 illustrates the workflow of the proposed architecture diagram, which is explained briefly in the following session.

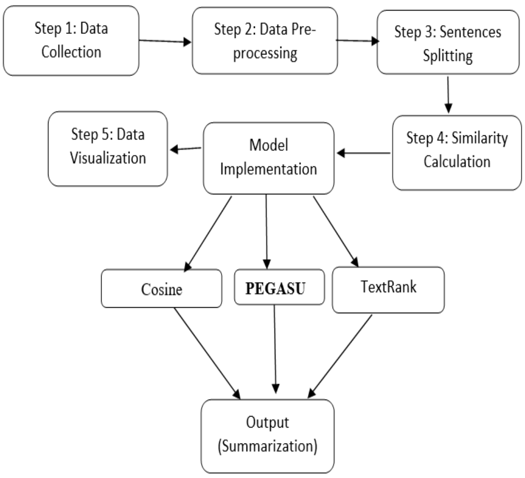


Figure 2. Proposed System Architecture

IV. Proposed System

4.1 Components of Proposed System:

In the text summarization method, the process starts with data collection, where relevant text data is gathered. Next, data preprocessing is performed to clean and prepare the collected text. The text is then split into sentences to enable independent analysis. Similarity calculation techniques are applied to measure the similarity between sentences and identify important information. Finally, in the summary generation step, important sentences are selected and concatenated to create a shorter summary. This workflow includes data collection, preprocessing, sentence splitting, similarity calculation, and summary generation.

4.1.1 Data Collection

The act of obtaining information or data from numerous sources is referred to as data collection. Data collection in the context of a word file may entail extracting information from the file or using the file as a source of information for analysis. Numerous applications of better-driven text exist, including news aggregation, academic research, legal documents, business reports, and social media analysis summarizing utilizing Pytext and Pegasus. These applications may be found in a variety of fields and for a wide range of use cases. This technology can help users save time and improve the efficiency of their processes by accelerating information retrieval and summarization. As shown in ,table 1 that the data mainly originates from various shopping websites such as Amazon, Snapdeal, Flipkart, and others. The dataset consists of customer reviews collected from online platforms. The dataset attributes include ID, Product ID, User ID, Profile Name, Score, and Time. In total, the dataset comprises 355,000 entries, while the summarization text consists of 90,000 entries.

Table 1: Dataset from Website

S.NO	Platform	No. of Dataset
1	Amazon	**1,00,000**
2	Flipkart	**50,000**
3	Snapdeal	**50,000**
4	Meesho	**1,00,000**
5	Ajio	**55,000**
Total		**3,55,000**

4.1.2 Data Preprocessing

Pre-processing is the process of removing any unnecessary information from the text, such as stop words, redundant punctuation, and special characters. The text can also be steamrolled or lemmatized to cut down on word count and improve the efficacy of the summary method.

Duplicate Dropping: It is one of the preprocessing modules which are used to drop the duplicate in the give text or sentences.

Contraction Mapping: In concerting mapping, is generally used to eliminate the word which are used in the shortcut or in short format. For example, instance of this it has been mentioned as tis.

Data Clearing: In data cleaning, special characters are typically removed or eliminated from the text.

Stop Word: The text does not get much information from these terms, which are among the most prevalent in any language (along with articles, prepositions, pronouns, conjunctions, etc.).

4.1.3 Sentence splitting

The text is divided into distinct sentences using sentence segmentation tools like NLTK or Spacy. Then, each phrase is represented as a vector using the bag-of-words model.

Tokenization: Tokenization is the process of breaking up the input data into a series of meaningful components, such as a word, an image patch, or a textual phrase. Tokenization is the process of breaking down input data into components (symbols) that may be mapped (embedded) into a vector space.

4.1.4 Similarity calculation

The cosine similarity formula is used to compare the vectors corresponding to each sentence and the vector corresponding to the entire document. This will show how much the original text is retained in each sentence.

Cosine Similarity: Cosine similarity is one of the metrics used in natural language processing to compare the text in two documents, regardless of their size. A word is visualized as a vector. The text documents are visualized as vector objects in an n-dimensional space. We must first count the number of times each word appears in each document in order to calculate the cosine similarity. We can use the Scikit-Learn library's CountVectorizer or TfidfVectorizer methods to count the number of times a word appears in each document. Where cosine similarity calculated using Eq(1).

Pegasus: A cutting-edge pre-trained transformer model created by Google with a focus on abstractive text summarization is called PEGASUS. It has been demonstrated to be very successful in producing excellent summaries of lengthy materials. The PEGASUS model can be used to create succinct, accurate summaries of lengthy documents within the framework of the Streamlining Information Retrieval using NLP-Based Summarization project while maintaining the content and context of the original text.

4.1.5 Summary Generation

Pytext is a platform for natural language processing that enables users to create and enhance deep learning models for text summarization. A prominent method for summarizing that compares the similarity of phrases based on their word frequency is known as cosine similarity. In text summarization tasks, the pre-trained transformer-based model Pegasus has shown outstanding results. Users can get better results by combining Pytext with Pegasus for tasks like summarizing academic material or breaking news. The phrases with the highest similarity scores are those that will be included in the summary. A threshold similarity score can be used to predetermine or base the number of sentences that will be in the summary.

Pseudo Code of Improved Driven Text Summarization Using Pageranking Algorithm And Cosine Similarity:

1: Collect data from various website
2: Preprocess the data
3: Drop the duplicate
4: Do contraction mapping.
5: Clear the special character like [\([^)]*\]

6: Split the Sentence.
7: Pegasus are used for length reduction for the text
8: Apply cosine similarity formula and do compare
9: Apply PageRank Algorithm.

IV. Implementation

Proposed system is implemented on the PC with Colab. Which has the following flow in the format. Some of the User defined Preprocess are preprocess data collection, removes duplicate, convert mapping, clear data and remove stop word.

1: Data Collection

 # Collect data from various website platforms
collected_data=collect_data_from_websites (["Amazon","Flipkart",” Snapdeal”,” Meesho”,” Ajio”])

2: Data Preprocessing

 # Preprocess the collected data

preprocessed_data=preprocess_data(collected_data)

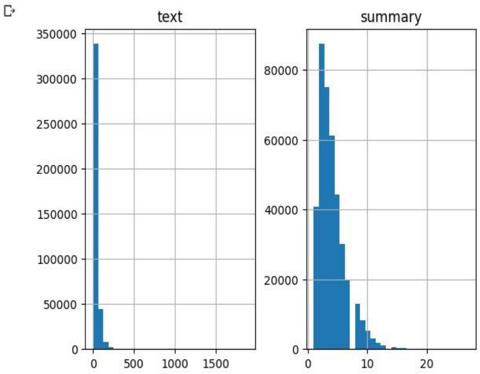
 # Remove duplicates

preprocessed_data=remove_duplicates(preprocessed_data)
 # Convert mapping
preprocessed_data=convert_mapping(preprocessed_data)
 # Clear data
preprocessed_data=clear_data(preprocessed_data)
 # Remove stop words
preprocessed_data=remove_stop_words(preprocessed_data)

3: Sentence Splitting

 # Tokenize the preprocessed data

tokenized_data=tokenize_data(preprocessed_data)
4: Cosine Similarity



 # Compare sentences with the entire document using cosine similarity

similarity_scores=compute_cosine_similarity (tokenized_data)

5: Text Length Reduction (Using Pegasus)

 # Reduce the length of the text in the document using Pegasus model

reduced_text=reduce_text_length(tokenized_data)

6: Summarization Generation (TextRank Algorithm)
 # Apply TextRank algorithm for summarization
summary=generate_summary (tokenized_data, similarity_scores)

V. Results

To provide precise summaries from longer texts, the suggested method for expediting the retrieval of data with NLP-based summarization employs many techniques, including word counts, word vectors, cosine similarities, and stop words. The approach entails preprocessing the text, which includes deleting stop phrases, tokenizing, and lemmatizing the text. Following this, summary sentences are generated depending on the significance of each phrase, which is

ISSN 2063-5346

Text	Text Cleaned Text	Cleaned Summary
I have bought several of the Vitality canned dog food products and have found them all to be of good quality. The product looks more like a stew than a processed meat and it smells better.	bought several vitality canned dog food products found good quality product looks like stew processed meat smells better labrador finicky appreciates product better	good quality dog food
Product arrived labeled as Jumbo Salted Peanuts...the peanuts were actually small sized unsalted. Not sure if this was an error or if the vendor intended to represent the product as "Jumbo".	product arrived labeled jumbo salted peanuts peanuts actually small sized unsalted sure error vendor intended represent product jumbo	not as advertised
This taffy is so good. It is very soft and chewy. The flavors are amazing. I would definitely recommend you buying it. Very satisfying!!	taffy good soft chewy flavors amazing would definitely recommend buying satisfying.	wonderful tasty taffy
Great taffy at a great price. There was a wide assortment of yummy taffy. Delivery was very quick. If you're a taffy lover, this is a deal.	great taffy great price wide assortment yummy taffy delivery quick taffy lover deal	great taffy
I'm disappointed with the flavor. The chocolate notes are especially weak. Milk thickens it but the flavor still disappoints. This was worth a try but I'll never buy again.	disappointed flavor chocolate notes especially weak milk thickens flavor still disappoints worth try never buy use left gone time thanks small cans	disappointed
This is a very healthy dog food. Good for their digestion. Also good for small puppies. My dog eats her required amount at every feeding.	healthy dog food good digestion also good small puppies dog eats required amount every feeding	healthy dog food

Table2: Sample Input and Output

determined using a variety of metrics, including word frequency and cosine similarity. To increase the precision of the generated summaries, the system employs an algorithm for machine learning that has been trained on a sizable corpus of texts.

According to a stated similarity of 43%, the suggested technique is effective at producing precise summaries from longer texts.

The dataset consists of customer reviews given on online platforms. Some of the dataset attributes include ID, Product ID, User ID, Profile Name, Score, and Time. The total count of the dataset is 355,000 entries, while the summarization text comprises 90,000 entries

Fig 3 represents the difference between the before and after summarization. The total review here is of 3,55,000 before summarization and filtering is done. Once filtrating and summarization done it reduce to 90,000 texts, which are fully in proper manner.

Here is Table 2, which represents the sample input and output of the PageRank summarization. In this process, the text is summarized, then the text is cleaned, and finally, the output is generated.

Fig 3. Text summarization Graph

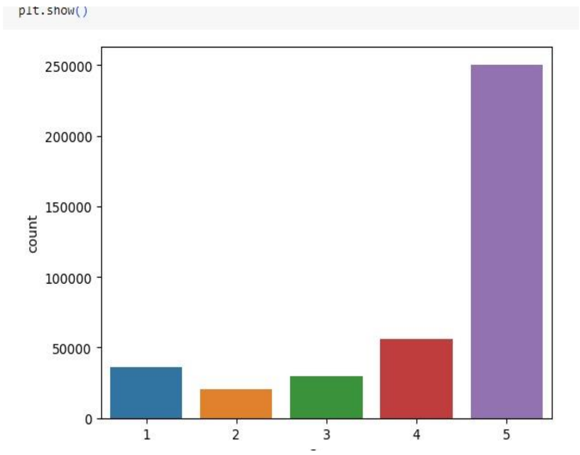
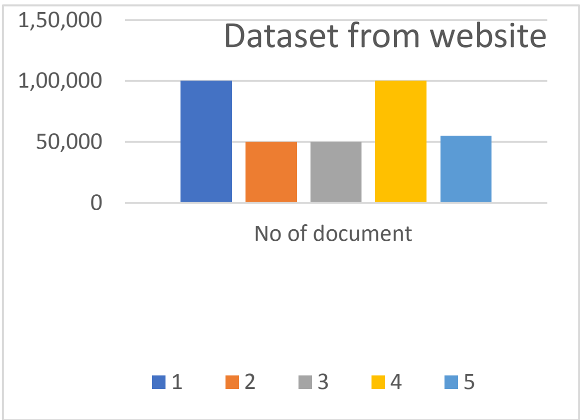


Fig 4. Total Review given by customer

Fig 4 refers here the review rating of customer



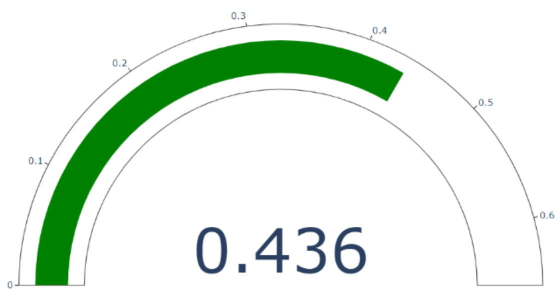


Fig 6. Gauge Meter Accuracy 1

given in the platform, which are scaled from 1 to 5. Were 1 is denoted the least rating and 5 is the higher rating.

Fig 5 represents the Dataset taken from various website which are used for summarization.

VI. Discussion

The approach that is proposed in the article "Improved Driven Text Summarization Using Pageranking Algorithm and Cosine Similarity"**,** seeks to offer a more effective and efficient method of summarizing large text volumes.

The numerous NLP methods included stopping word removal, stemming, and a similarity calculation model in the suggested system. They also describe the system's use of the Pegasus and Pytext ranking models to produce the summary.

According to the experimental findings, the proposed method obtains an accuracy of 43%, which is highly encouraging. The suggested system's numerous applications, including those for summarizing news articles, academic papers, and business reports, are also covered by the authors.

The authors agree that the suggested system can still be improved upon and propose that future research could concentrate on enhancing the system's functionality by including more sophisticated methods of natural language processing and models. In order to ensure the system's efficacy and generalizability, they also stress how crucial it is to evaluate it using a variety of texts and datasets.

Fig 6 represents the gauge meter accuracy for the cosine similarity, which is at 43%. The authors have suggested numerous applications for the system, including summarizing news articles, academic papers, and business reports.

These applications are also covered by the authors.

Overall, the study offers a thorough analysis of the suggested method, its uses, and its drawbacks, which can be helpful for academics and professionals involved in NLP-based summarization.

VII. Conclusion

In order to improve the effectiveness and efficiency of information retrieval, " Improved Driven Text Summarization Using Pageranking Algorithm and Cosine Similarity" summarizes documents. The proposed system preprocesses the text data using methods including word counts, word vectors, cosine similarity, stopping words, and stemming. Then, for similarity computation and summary creation, it uses the Pytext ranking and Pegasus models, respectively.

The proposed system can be applied to many other contexts, such as news items, academic papers, and legal documents, to assist users in quickly understanding a text's primary point without having to read the full material.

Dataset which are as a review given by the customer in the online platform. Total dataset taken here is 3,55,000 as a overall count and the summarization text are of 90,000.

As a result, the proposed system can greatly speed up information retrieval by offering precise and succinct summaries of documents. Particularly in fields where it is necessary to routinely process a large amount of text input, it can speed up work and enhance productivity.

VIII. References

[1] Manojkumar, V. K., Mathi, S., & Gao, X. Z. (2023). An Experimental Investigation on Unsupervised Text Summarization for Customer Reviews. Procedia Computer Science, 218, 1692-1701.

[2] Jain, R. (2022). Automatic Text Summarization of Hindi Text Using Extractive Approach. ECS Transactions, 107(1), 4469.

[3] Onah, D. F., Pang, E. L., & El-Haj, M. (2022). A Data-driven Latent Semantic Analysis for Automatic Text Summarization using LDA Topic Modelling.

[4] Pal, S., Chang, M., & Iriarte, M. F. (2022). Summary generation using natural language processing techniques and cosine similarity. In Intelligent Systems Design and Applications: 21st International Conference on Intelligent Systems Design and Applications (ISDA 2021) Held During December 13–15, 2021 (508-517). Cham: Springer International Publishing.

[5] Adwani, M. K. S., & Shelke, P. P. A Review Of Extractive Text Summarization Using Bert And Page Rank.

[6] Pokharkar, A., Dhumal, P., Singh, A., & Hadawale, H. (2022). Text Summarizer Using NLP. Available at SSRN 4097878.

[7] Senthamizh, S. R., & Arutchelvan, K. (2022). Automatic text summarization using document clustering named entity recognition. International Journal of Advanced Computer Science and Applications, 13(9).

[8] Reddy, S. S. S., Kumar, C. D., Pisati, S. R., & Kolli, R. D. (2022, December). Document Summarization Model Using Modified Pagerank Algorithm. In Artificial Intelligence and Data Science: First International Conference, ICAIDS 2021, Hyderabad, India, December 17–18, 2021, Revised Selected Papers (430-439). Cham: Springer Nature Switzerland.

[9] Gupta, S., Sharaff, A., & Nagwani, N. K. (2022). Biomedical text summarization: a graph-based ranking approach. In Applied Information Processing Systems: Proceedings of ICCET 2021 (147-156). Springer Singapore.

[10] Adhika Pramita Widyassari, Supriadi Rustad , Guruh Fajar Shidik , Edi Noersasongko , Abdul Syukur , Affandy Affandy , De Rosal Ignatius Moses Setiadi . Journal of King Saud University - Computer and Information Sciences, Volume 34, Issue 4, April 2022, 1029-1046

[11] Minakshi Tomer, Manoj Kumar.Journal of King Saud University - Computer and Information Sciences Volume 34, Issue 8, Part B, September 2022, 6057-6065

[12] Aakanksha Sharaff, Mohit Jain & Geethika Modugula. International Journal of Information Technology volume 14, 2057–2065 (2022)

[13] Divakar Yadav,Naman Lalit, Riya Kaushik, Yogendra Singh, Mohit, Dinesh, Arun Kr. Yadav, Kishor V. Bhadane, Adarsh Kumar and Baseem Khan. Computational Intelligence and Neuroscience Journal Volume 2022 | Article ID 3411881

[14] Gulati, V., Kumar, D., Popescu, D. E., & Hemanth, J. D. (2023). Extractive Article Summarization Using Integrated TextRank and BM25+ Algorithm. Electronics, 12(2), 372.

[15] Yadav, D., Katna, R., Yadav, A. K., & Morato, J. (2022). Feature Based Automatic Text Summarization Methods: A Comprehensive State-of-the-Art Survey. IEEE Access, 10, 133981-134003.

[16] Chandrika, C. P., & Kallimani, J. S. (2022). Extractive Text Summarization of Kannada Text Documents Using Page Ranking Technique. In Intelligent Data Communication Technologies and Internet of Things: Proceedings of ICICI 2021 (683-695). Singapore: Springer Nature Singapore.

[17] Chandrika, C. P., & Kallimani, J. S. (2022). Extractive Text Summarization of Kannada Text Documents Using Page Ranking Technique. In Intelligent Data Communication Technologies and Internet of Things: Proceedings of ICICI 2021 (683-695). Singapore: Springer Nature Singapore.

[18] Baruah, N., Sarma, S. K., Borkotokey, S., Borah, R., Phukan, R. D., & Gogoi, A. (2022). A Graph-Based Extractive Assamese Text Summarization. In Computational Methods and Data Engineering: Proceedings of ICCMDE 2021 (1-12). Singapore: Springer Nature Singapore.

[19] Pulugurtha, M. H., & Jagadamba, P. V. S. L. (2022). Programmed Speech to Text Summarization Using Graph Based Algorithm. open science index 16 2022, 2, 112.

[20] 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) (274-279). IEEE.

[21] Yadav, M., & Katarya, R. (2023). A Systematic Survey of Automatic Text Summarization Using Deep Learning Techniques. In Modern Electronics Devices and Communication Systems: Select Proceedings of MEDCOM 2021 (397-405). Singapore: Springer Nature Singapore.

[22] Ramachandran, R., Jayachandran, S., & Das, V. (2022, October). A Novel Method for Text Summarization and Clustering of Documents. In 2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT) (1-6). IEEE.

[23] Yadav, D., Lalit, N., Kaushik, R., Singh, Y., Yadav, A. K., Bhadane, K. V., ... & Khan, B. (2022). Qualitative analysis of text summarization techniques and their applications in the health domain. Computational Intelligence and Neuroscience, 2022.

[24] Balaji, N., Megha, N., Kumari, D., Kumar, S., & Bhavatarini, N. (2022, October). Text Summarization using NLP Technique. In 2022 International Conference on Distributed

Section A-Research paper
ISSN 2063-5346

Computing, VLSI, Electrical Circuits and Robotics (DISCOVER) (30-35). IEEE.

[25] Royan, R., Jayakumaran, C., & Stephan, T. (2022). Text summarization for automatic grading of descriptive assignments: A hybridapproach. In Enterprise Digital Transformation (251-273). Auerbach Publications.