BRIEFIFY - A Text Summarization tool

Dr Rakesh Kumar M
Computer Science and Engineering
Rajalakshmi Engineering College,
Chennai, India
rakeshkumar.m@rajalakshmi.edu.in

Sivanantham D
Computer Science and Engineering
Rajalakshmi Engineering College,
Chennai, India
210701250@rajalakshmi.edu.in

Gopal K
Computer Science and Engineering
Rajalakshmi Engineering College,
Chennai, India
210701517@rajalakshmi.edu.in

Abstract—**In order to effectively condense long documents into brief summaries, this paper presents a novel text summarization system. The system makes use of natural language processing methods, such as cosine similarity and PageRank algorithms, to determine the semantic relevance and importance of key sentences.**

When used in conjunction with a Streamlit web application, users can upload text files and obtain output that has been summarized, facilitating quick understanding and decision-making. Thorough testing on a variety of datasets shows how well the system generates accurate and coherent summaries. This strategy has potential for a number of information condensation applications, such as content curation and document summarization. With its robust algorithmic framework and easy-to-use interface, the suggested system responds to the increasing need for effective text summarization tools in academic and practical contexts.

Keywords— Text summarization, natural language processing, cosine similarity, PageRank algorithm, Streamlit web application, document summarization, semantic relevance, key sentences, decision-making.

I. Introduction

Text summarization reduces long documents into brief summaries, which is important for knowledge management and information retrieval. Recent advances have enabled the development of sophisticated summarization systems that can produce precise and coherent summaries by utilizing natural language processing (NLP) techniques.

Text summarizing plays a crucial role in knowledge management and information retrieval by producing concise summaries from extensive materials. Recent advances have made it possible to create sophisticated summarizing systems that use natural language processing (NLP) approaches to generate accurate and coherent summaries. This work introduces a novel text summarizing system that uses PageRank and cosine similarity algorithms to identify relevant sentences inside documents.

When combined with a Streamlit online application, users can quickly upload text files and receive output that has been summarized, which speeds up understanding and decision-making. Comprehensive testing on a range of datasets assesses the system's efficacy and shows that it can produce high-caliber summaries in a number of different genres and domains.

In recent years, the exponential growth of digital information has resulted in an overwhelming volume of textual data, making it increasingly challenging for individuals to extract relevant insights efficiently. Text summarization techniques offer a solution to this problem by automatically generating condensed representations of textual content, enabling users to quickly grasp the main ideas and key points within documents. By leveraging NLP algorithms, text summarization systems analyze the semantic structure of documents, identify important sentences, and construct summaries that preserve the essential information while minimizing redundancy. This paper presents a comprehensive overview of the text summarization process, highlighting the key components and methodologies involved in generating effective summaries.

The suggested text summarization technique can discover important sentences that encapsulate the content of the document by measuring the semantic similarity between sentences using cosine similarity. The system also uses the PageRank method to rank sentences according to how important they are in the document's semantic network. The system creates summaries that faithfully capture the information and organization of the source document by fusing various algorithms into a unified framework. In addition, the system is integrated into a Streamlit online application, giving customers an easy-to-use platform to input documents and view real-time output summaries.

II. Literature Survey

[1] Text summarization has emerged as a vital tool for managing the abundance of online textual data. This paper underscores the challenges of manual summarization and the necessity for automated systems to condense text while maintaining its essence. It explores the intertwined nature of text mining and summarization, categorizing approaches and evaluating their effectiveness based on key parameters.

[2] Text summary becomes an indispensable tool in the age of expanding data, helping to gain succinct information, cut down on reading time, and streamline research work. However, because of their complexity, summarization is difficult in languages like Bangla. In order to close this gap, this research suggests an extraction-based summarizing strategy designed specifically for Bangla language. Its efficacy is demonstrated by experimental results, which point to potential future developments in smart machine technology for industry 4.0.

[3] This study proposes a three-stage automatic rule reduction technique-based text categorization and summarizing approach: token creation, feature identification, and categorization/summarization. When evaluated with sample input texts, the built text analyzer yields impressive results. The effectiveness of the method and parameter choices for text classification have been confirmed by extensive testing. Many real-world uses, such as word sense disambiguation, information extraction, web resource categorization, and document retrieval, could benefit from this study.

[4] In the field of Natural Language Processing (NLP), this study suggests a text summary technique that emphasizes phrase selection from the source material. It uses structured diagrams to represent unstructured texts and preprocesses them so that different feature extraction techniques can be used. Without requiring in-depth language expertise, the method is flexible enough to work with a variety of languages and evaluates sentence relevance by linear weighting.

[5] Selecting important information is made more difficult by the exponential rise of data. This is addressed by text summarizing using Natural Language Processing (NLP), which reduces data without losing its essential meaning. This paper suggests a model that uses NLTK for sentiment analysis on news content with the goal of effectively extracting and presenting pertinent information to users.

[6] The rapidly developing field of automated text summarization (ATS) aims to automatically reduce massive text volumes in order to save users time and effort. This overview covers extractive, abstractive, and hybrid techniques established since the 1950s, and it explores the problems and advancements in ATS. Even with advancements, computer summaries frequently deviate from those produced by humans. For researchers in the subject, the study provides an extensive investigation of ATS, covering approaches, obstacles, applications, and assessment metrics.

[7] Automatic Text Summarization is the process of extracting important information from a text using either Extractive or Abstractive techniques. In this study, a new statistical technique for extractive summarization of individual documents is presented. A high-quality summary that can be recorded as audio is created by extracting highly ranked sentences from a list of sentences ranked according to predetermined weights.

[8] Condensing significant information from a text into a brief summary is known as text summarizing, and it aims to meet the growing need for simplicity in the news, business, and research domains. The objective of this work is to assess the implementation time, accuracy, and human-like quality of generated summaries by comparing extractive and abstractive summarization algorithms. The study assesses each method's advantages and disadvantages using manual examination and summary rating.

[9] Effective text summarizing techniques are imperative in the Big Data era due to the exponential expansion of textual data across many languages. Large texts are automatically summarized into shorter forms by either reformulating (Abstractive) or extracting entire sentences (Extractive). There are many methods for English and other European languages, but there aren't as many for Indian languages. This work addresses ongoing research issues and proposes a machine learning-based approach for summarizing texts in Hindi. It also analyzes text summary techniques in foreign and Indian languages.

[10] Because text summarizing reduces communication network overhead by condensing textual information, it is essential given the ongoing advancements in internet technology. In order to demonstrate lower transmission costs, this research suggests an architecture in which users' edge devices get summarized text from a central base station. By using LSTM-RNN, summaries are guaranteed to be effective and expenditures are reduced by an average of 85%.

III. EXISTING SYSTEM

There are several different ways to text summarization; they include abstraction-based techniques like KL-Sum and Genetic Algorithms, as well as extraction-based techniques like Latent Semantic Analysis (LSA), TextRank, and Luhn's Algorithm. Automated summarizing solutions are provided by market products such as SummarizeBot, SMMRY, and QuillBot. These products use sophisticated natural language processing (NLP) algorithms to condense text from many sources into brief and useful summaries that meet the needs and tastes of varying users.

QuillBot: Text summary is one of the many tools offered by the flexible online writing and editing tool QuillBot. Users may input text into the platform with ease thanks to its user-friendly interface, and QuillBot's sophisticated algorithms quickly provide succinct summaries. In order to extract pertinent information and select important sentences from the input text while maintaining readability and coherence, the summarizing process entails examining the text.

Modern natural language processing (NLP) techniques are employed by QuillBot to generate precise summaries that encapsulate the key ideas of the original content. Users are able to modify the summary length based on their needs and preferences. Whether it is a research paper, document, pdf ,story or article , Quillbot can easily summarise all the above most effectively.

GRAMMARLY : Grammarly, while primarily known for grammar checking, also offers text summarization capabilities. It analyzes text for key points, concisely condensing content into shorter summaries. Utilizing advanced NLP algorithms, Grammarly ensures clarity and coherence in the generated summaries, aiding in efficient information consumption and communication.

SMMRY : Using cutting-edge NLP approaches, SMMRY is an online text summation tool that helps users reduce long papers and articles into succinct summaries. SMMRY use abstraction techniques to paraphrase words while maintaining their original meaning, as opposed to conventional extraction-based methods, guaranteeing coherence and readability. By giving priority to the most important content and removing unnecessary details, it produces precise summaries that successfully convey the main ideas of the original text. SMMRY is a widely used tool that professionals and individuals in a variety of industries use to efficiently consume information and make decisions. SMMRY's exact summarization skills and user-friendly interface make it an invaluable tool for rapidly extracting pertinent insights from large amounts of textual content.

IV. PROPOSED SYSTEM

A. Objectives

- Develop an automated text summarization tool using NLP techniques.
- Extract key sentences and generate concise summaries from large text documents.
- Utilize algorithms such as cosine similarity and PageRank for semantic analysis.
- Offer a user-friendly interface for easy text upload and summarized output retrieval.
- Enhance accessibility and usability for efficient comprehension and information retrieval.
-

B. Approach

- Data Collection :

First, text data is gathered by the system from a variety of sources, such as papers, publications, and websites. The text summarizing technique uses this data, which spans a wide range of subjects and areas, as its input.

- PreProcessing :

The text goes through preprocessing procedures like tokenization, stopword removal, and sentence splitting after data collection. The text data is ready for additional analysis and summary after these procedures.

- Similarity Calculation :

Using cosine distance or other similarity metrics, the algorithm determines how similar the sentences are to one another in the text. In this step, each sentence's significance and relevance within the whole document's context are assessed.

- Graph Construction:

considering these factors, farmers can align their crop choices with the prevailing weather patterns and select crops that are resilient and adaptable to the local climate.

- Ranking and Selection:

Based on their value within the content, the sentences are ranked by the system using graph-based techniques such as PageRank. In order to ensure that the most pertinent information is provided, the summary is formed from the sentences that rank highest.

- Summary Generation :

Lastly, the summary is created by combining the chosen sentences. The technology provides users with important insights without requiring them to read the entire input material by producing a succinct and educational summary.

By using similarity calculations and graph-based algorithms to find and extract key information from the input text, this method simplifies the text summarizing process.

V . WORKING

This text summarization web app, built with Streamlit, uses cosine similarity and PageRank algorithms to condense uploaded text files into concise summaries. Users simply upload a file, and the app intelligently extracts the most relevant sentences, providing an efficient way to digest large amounts of text.

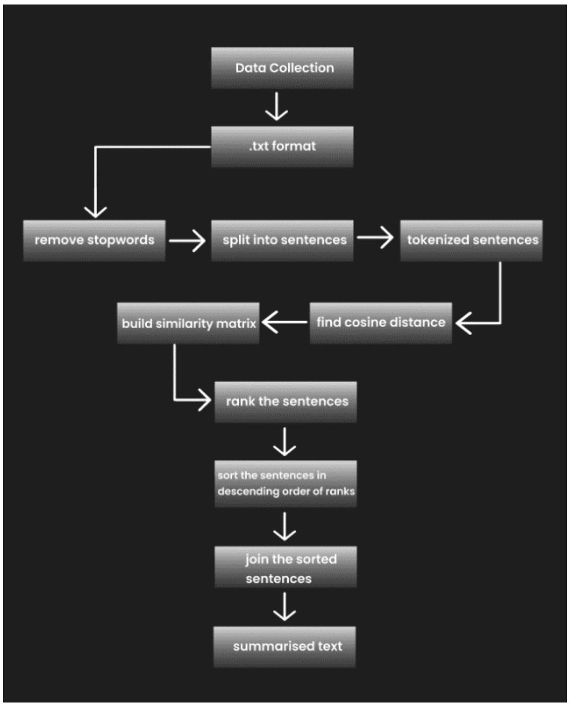
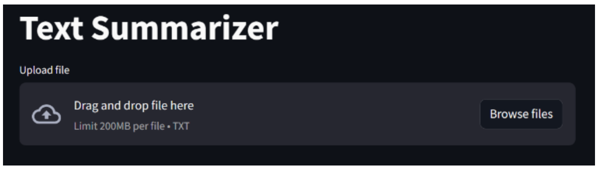


Fig. Architecture diagram of BRIEFIFY

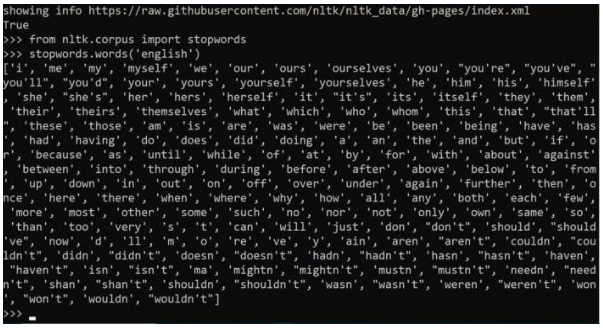
A. Data Collection

- This phase makes it easier to obtain text data for summarization that comes from files ending in.txt. Users use the online interface to upload files, which are limited to.txt files.
- The system checks the file format after submission to make sure it complies.
- Assuming that sentences conclude with a period and a space, it divides the text into separate sentences. After that, each sentence is tokenized into words, with non-alphabetic characters removed. Ultimately, the function yields a list of tokenized sentences that are prepared for additional processing, including summarization or similarity calculations.



B. NLTK Stopwords

- This phase ensures that only relevant textual content is processed for summarization by filtering out common English stopwords using NLTK's built-in stopwords corpus. NLTK provides a comprehensive list of stopwords, including articles, prepositions, and conjunctions, which are often not indicative of the central theme or meaning of the text.
- By excluding these stopwords from the analysis, the system focuses on significant words and phrases, improving the accuracy and relevance of the generated summaries. This step optimizes the summarization process by prioritizing essential content while reducing noise and irrelevant information.



```
showing info https://raw.githubusercontent.com/nltk/nltk_data/gh-pages/index.xml
True
>>> from nltk.corpus import stopwords
>>> stopwords.words('english')
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "
you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself'
, 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself', 'they', 'them',
 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', "that'll
", 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has
', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'o
r', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against'
, 'between', 'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from
', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'o
nce', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few',
 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so',
 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "should
've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'aren', "aren't", 'couldn', "cou
ldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'haven',
 "haven't", 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'needn', "need
n't", 'shan', "shan't", 'shouldn', "shouldn't", 'wasn', "wasn't", 'weren', "weren't", 'won
', "won't", 'wouldn', "wouldn't"]
>>> _
```

C. Similarity Matrix

Quantifying the similarity between sentence pairs is a necessary step in creating the similarity matrix for text summarization. This is the general procedure:

1. Tokenization and Preprocessing: Every sentence is tokenized into individual words, and stopwords and a lowercase conversion are done as part of the preprocessing step.

2. Creating the Matrix: Using the tokenized representations of each pair of sentences, a similarity score is calculated. This score shows the degree of content similarity between the sentences.

3. Cosine Similarity: When determining how similar two vectors are to one another, cosine similarity is frequently employed. The similarity between the vector representations of two sentences is calculated here.

4. Matrix Representation: Each cell in the matrix reflects the degree of similarity between the matching pair of sentences, and the similarity scores are kept there. The square matrix's dimensions correspond to the total number of sentences in the manuscript.

5. PageRank Usage: The PageRank algorithm is based on the similarity matrix. By giving the connection information between sentences, it enables PageRank to evaluate each sentence's significance in relation to other phrases.

In general, the process of creating a similarity matrix serves as the basis for locating significant sentences within the text, which makes the process of summarizing it easier.

D. PageRank Algorithm

The PageRank algorithm is used in text summarization to order sentences according to their significance in the content. Sentences are viewed as nodes in a network, and the similarity between sentences is represented by the connections (edges) that connect them.

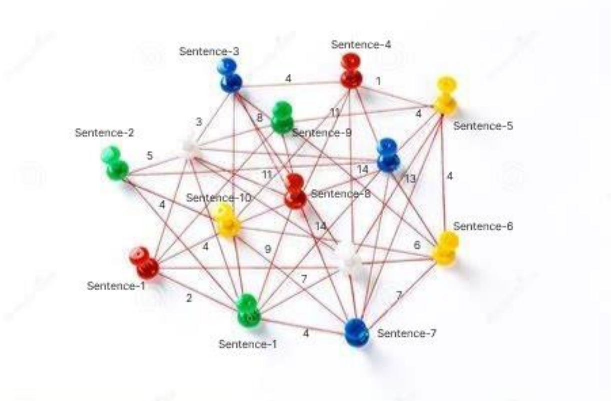
The creators of Google, Larry Page and Sergey Brin, created PageRank, which gives each text a score determined by how important the sentences it is connected to.

1. Building the Network: We start by creating a graph in which every sentence is a node and the connections (edges) between sentences are determined by how similar they are to each other.

2. Determining Sentence Importance: PageRank uses an iterative process to rate each sentence's importance by adding the scores of all the sentences that are connected to it. In this instance, a sentence's importance is increased in proportion to its similarity score.

3. Convergence: This process keeps on until the scores converge, which shows that sentences' relative importance has stabilized.

4. Ranking: The final step involves assigning importance scores to each sentence, with the highest-scoring sentences being chosen to create the summary.



To sum up, PageRank assists in producing a succinct and enlightening summary by evaluating the relationships between the most significant sentences in a document.

VI . Conclusion

Our project, BRIEFIFY, is a text summary tool that we developed with Streamlit and Python with the goal of giving people an effective way to create succinct summaries from text files that they upload. By utilizing graph algorithms and natural language processing techniques, the program automates the summary process.

The tool's primary job is to preprocess uploaded text files in order to separate sentences into individual words by tokenizing them. Based on the cosine similarity between sentence pairs, a similarity matrix is created. The relevance of each sentence inside the document is ranked using the PageRank algorithm. Ultimately, the summary is formed by choosing the sentences that rank highest.

The Streamlit user interface, which is implemented in Python, allows users to submit files and read summaries with ease. For text preprocessing tasks like stopword removal and tokenization, the NLTK package is useful, and NetworkX is used for graph-based operations, including implementing the PageRank algorithm.

There is room for improvement even though the existing design offers a useful tool for summarizing. To improve the tool's resilience, for example, error handling techniques might be improved to handle edge circumstances like erroneous file uploads or empty documents. Moreover, performance optimization of the code could improve user experience, particularly for huge documents.

All things considered, our project provides a useful way to automate text summary jobs by using graph algorithms and natural language processing methods to produce insightful summaries from text documents. It could be a useful tool for many applications that need text summarizing skills with further development and improvement.

References

[1] S. R. Rahimi, A. T. Mozhdehi and M. Abdolahi, "An overview on extractive text summarization," 2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI), Tehran, Iran, 2017, pp. 0054-0062, doi: 10.1109/KBEI.2017.8324874.

[2] T. Islam, M. Hossain and M. F. Arefin, "Comparative Analysis of Different Text Summarization Techniques Using Enhanced Tokenization," 2021 3rd International Conference on Sustainable Technologies for Industry 4.0 (STI), Dhaka, Bangladesh, 2021, pp. 1-6, doi: 10.1109/STI53101.2021.9732589.

[3] C. Lakshmi Devasena and M. Hemalatha, "Automatic Text categorization and summarization using rule reduction," IEEE-International Conference On Advances In Engineering, Science And Management (ICAESM -2012), Nagapattinam, India, 2012, pp. 594-598.

[4] C. HARK, T. UÇKAN, E. SEYYARER and A. KARCI, "Graph-Based Suggestion For Text Summarization," 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), Malatya, Turkey, 2018, pp. 1-6, doi: 10.1109/IDAP.2018.8620738.

[5] A. Mishra, A. Sahay, M. a. Pandey and S. S. Routaray, "News text Analysis using Text Summarization and Sentiment Analysis based on NLP," 2023 3rd International Conference on Smart Data Intelligence (ICSMDI), Trichy, India, 2023, pp. 28-31, doi: 10.1109/ICSMDI57622.2023.00014.

[6] B. Khan, Z. A. Shah, M. Usman, I. Khan and B. Niazi, "Exploring the Landscape of Automatic Text Summarization: A Comprehensive Survey," in IEEE Access, vol. 11, pp. 109819-109840, 2023, doi: 10.1109/ACCESS.2023.3322188.

[7] J. N. Madhuri and R. Ganesh Kumar, "Extractive Text Summarization Using Sentence Ranking," 2019 International Conference on Data Science and Communication (IconDSC), Bangalore, India, 2019, pp. 1-3, doi: 10.1109/IconDSC.2019.8817040.

[8] P. Raundale and H. Shekhar, "Analytical study of Text Summarization Techniques," 2021 Asian Conference on Innovation in Technology (ASIANCON), PUNE, India, 2021, pp. 1-4, doi: 10.1109/ASIANCON51346.2021.9544804.

[9] P. Shah and N. P. Desai, "A survey of automatic text summarization techniques for Indian and foreign languages," 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), Chennai, India, 2016, pp. 4598-4601, doi: 10.1109/ICEEOT.2016.7755587.

[10] R. Alfred, J. H. Obit, C. P. -Y. Chin, H. Haviluddin and Y. Lim, "Towards Paddy Rice Smart Farming: A Review on Big Data, Machine Learning, and Rice Production Tasks," in IEEE Access, vol. 9, pp. 50358-50380, 2021, doi: 10.1109/ACCESS.2021.3069449.