© April 2024 | IJIRT | Volume 10 Issue 11 | ISSN: 2349-6002

Text Summarization in The Age of Deep Learning: A Comprehensive Analysis

SACHIN BALVIR[1], SAMEER TEMBHURNEY2, MAITREYA MOHARIL[3], SAMIKSHA ANJARKAR[4], BHUSHAN DHAWALE[5], SHARVARI KAMBLE[6], KARAN DOGRA[7]
[1, 2, 3, 4, 5, 6, 7] *B. Tech CSE SBJITMR, Nagpur, Maharashtra India*

Abstract— Massive volumes of data are accessible online thanks to the digital revolution, but finding accurate and pertinent data is not always simple. Search engines are still able to retrieve considerably more information than the average user can handle or control. To make it easy for someone to understand the meaning without reading the entire paper, it is necessary to offer the information in an abstract manner. This work proposes a generator of a literature review summary for a single document. A term-sentence matrix is constructed from the document. To provide a streamlined version of the original text, our work mostly focuses on analytical details. Sentence position, sentence position in relation to the paragraph, the number of named entities, and the feature matrix of each sentence are examples of characteristics that are utilized to increase accuracy without sacrificing the text's core meaning. The paper primarily examines two primary kinds of text summarizing techniques in depth. They are text summaries that are abstractive and extractive. This data set's static analysis validates the outcome of our experiment.

Indexed Terms- Natural language Processing, Machine learning, Neural Network, Abstractive and Extractive Method.

I. INTRODUCTION

A literature review summary generator is a valuable tool in the realm of academic research and scholarly exploration, streamlining the intricate process of distilling, summarizing, and synthesizing information from an extensive body of existing literature. As researchers delve into the vast sea of scholarly articles, books, and publications, the need for efficient and effective methods to distill essential insights becomes paramount. The literature review summary generator emerges as a solution to this challenge, offering a systematic and automated approach to extract key findings, themes, and contributions from diverse sources.

The primary goal of this tool is to save researchers valuable time and enhance overall efficiency in the literature review process. Traditionally, scholars would manually review and synthesize each piece of literature, a time-consuming endeavor that could potentially lead to oversight or information overload. The literature review summary generator, however, automates this process, allowing researchers to focus on the interpretation and analysis of distilled insights rather than spending excessive time on the extraction of raw data.

Furthermore, the generator serves to distill the essence of diverse studies, facilitating the identification of patterns, trends, and gaps in the existing body of literature. By condensing information into digestible summaries, researchers gain a clearer understanding of the landscape within their chosen field. This aids in the identification of common themes, disparities in findings, and areas where further investigation is warranted. The functionality of a literature review summary generator is rooted in advanced algorithms and natural language processing capabilities. These algorithms analyze the content of selected literature, identifying key concepts, methodologies, and outcomes. The generator then crafts concise summaries that encapsulate thecore elements of each source. This not only accelerates the review process but also ensures that critical details are not overlooked.

II. FUNDAMENTALS

1. Text: Text is a versatile medium that serves as a cornerstone of human communication and a valuable source of information for computational analysis. It can be written, digital, or spoken. Text elements include characters, words, sentences, paragraphs, and syntax. Whereas natural language processing concentrates on the communication

between computers and human language, text mining draws importantinsights and patterns from massive amounts of textual data. Text categorization uses content to divide text into pre-established classes. It is essential to comprehend semantics and context in order to understand data accurately. Text is used for information retrieval, communication, and knowledge representation. However, challenges in text processing include ambiguity, variability, and subjectivity. Despite these challenges, text remains a valuable source of information for computational analysis and human communication.

2. Summarization: Text summary is the process of taking the most important concepts, details, and information out of a text while keeping its context and meaning intact. It is used with a variety of writings, such as research papers, articles, and documents. The process of summarizing might be carried out either computationally or manually. There are two different kinds of summaries: abstractive summarizing, which creates summaries by paraphrasing and rephrasing the original text, and extractive summary, which takes certain lines or phrases straight out of the original text. Key content identification, information compression, language development, scoring, and ranking are all steps in the summarizing process. Challenges in summarization include content selection, preserving meaning, handling ambiguity, and domain specificity. Applications of summarization include news summaries, document summaries, email summaries, legal and compliance documents, social media summaries, and speech summaries. Text summarization enhances information retrieval and understanding by providing concise and meaningful representations of textual content.

3. Text Preprocessing: Text preprocessing, which converts unprocessed text input into a format appropriate for analysis, is an essential stage in natural language processing (NLP) and machine learning tasks. To improve the performance of the next NLP tasks and make the text easier to handle, it attempts to eliminate noise, unnecessary information, and inconsistencies. Lowercasing, tokenization, deleting punctuation, eliminating numerals, eliminating stop words, stemming, lemmatization, eliminating HTML elements and special characters, managing contractions, eliminating URL links, cutting extra whitespace, and spell checking are examples of common approaches. These procedures aid in the creation of a clear, consistent representation of text that may be applied to tasks such as topic modeling, text classification, and sentiment analysis. Although the NLPtask's particular criteria and the text data's qualities differ, they all strive to provide a clear, consistent representation ofthe text.

III. LITERATURE SURVEY

- *Yaser Keneshloo, Tian Shi, Naren Ramakrishnan and Chandan K. Reddy. "Deep Reinforcement learning for Seq2SSeq."*

Enhancements such as self-attention models, pointer- generation models, and attention-based models have been suggested by researchers. These issues have recently been addressed in seq2seq models through the application of reinforcement learning (RL) techniques. The paper presents a survey that combines RL methods with seq2seq models to overcome the limitations of current approaches. [1] The authors provide insights intothe problems and propose better RL models for abstractive text summarization. The paper also includes the source code for implementing RL models and presents targeted experiments on performance and training time.

- *Shuai Zhao, Fucheng You and Zeng Yauan Liu "Abstractive Text Summarization Using Pretrained Language Models. (December 31, 2020).*

It was shown that pretrained language models are improving the quality of abstractive text summarization by making summaries more fluent and coherent.However, abstractive summaries may not be factually accurate, and pretrained models may not have the domain-specific knowledge needed to generate accuratesummaries for all types of text. [2] The paper suggests that pretrained language models can be fine-tuned with less labeled data than traditional methods, and that they can generate summaries that are consistently of high quality.

- *Eduard Hovy and Chin-Yew Lin "Automated text summarization and the summarist system."(Southern California 4676 Admiralty Way Marina del Rey, CA 90292-6695).*

The purpose of this paper is to present the summarist automated multilingual text summarization system and discuss its modules and performance, provide a preliminary typology of summaries and explore methods to evaluate summaries. [3] In addition to discussing topicinterpretation, cue phrase usage, and acquiringbackground knowledge in the context of the summarist system, the authors hope to shed light on the nature of text summarization and the difficulties associated with evaluating summaries and summarization. The authors stress the significance of text summarization as a difficult task in natural language processing and thepossibility of beneficial outcomes from system enhancements.

- *Anish Jadhav, Rajat Jain, Steve Fernandes and Mrs. Sana Shaikh "Text Summarization using Neural Networks."(2019).*

To produce brief summaries from original text sources, this work proposes a text summary method that includes both extractive and abstractive approaches. [4] Based on statistical and semantic characteristics, the algorithmfinds the key phrases and joins them to provide a comprehensive report. After that, an encoder-decoder model is used to the combined report to provide a brief synopsis that encompasses the full piece. article.

- *Chandra Prakash and Dr. Anupam Shukla2"Text Summarizer "SAAR" using Reinforcement Learning." (Madhya Pradesh, India – 474010-2014).*

This paper mainly focused on human-aided text summarizer called "SAAR" for single documents, which generates a summary by creating a term-sentence matrix and using reinforcement learning to assign weights to theentries in the matrix. The goal of the study is to discuss the difficulty of selecting accurate and pertinent facts from the abundance of information on the internet. It recommends providing information in an abstract manner to make it easier to grasp without having to readthe full text. [5]

- *Urvashi Khandelwal, Kevin Clark, Dan Jurafsky, Lukasz Kaiser"Text Summarization Using a Single Pre-Trained Transformer."(May 2019).*

This work investigates the use of pre-trained language models (LMs) to abstractive summarizing, a type of sample efficient text summarization. The authors suggesta decoder-only network dubbed Transformer LM, in which the source and summary are encoded using the same LM. This method makes the model more straightforward, lowers the number of parameters, and guarantees that all model weights including attention over source states are pre-trained. [6]

- *Thevatheepan Priyadharshan and Sagara Sumathipala "Text Summarization for using NLP"(2018).*

The paper aims to automatically create an extractive summary. The research work focuses on improving the accuracy of the summary without losing the main idea ofthe text. [7]

- *Jingwei Cheng, Fu Zhang and Xuyang Guo "A Syntax-Augmented and Headline-Aware Neural Text Summarization Method" (2020)*

In the 1950s, Luhn introduced the first artificial text summarizing method, which extracted phrases from the source text using statistical data like word frequency.[8] Since then, scholars have been more interested in automatic text summarizing.

- *Laifu Chen and Minh Le Nguyen "Sentence Selective Neural Extractive Summarization with Reinforcement Learning" (2019)*

Their models' output is presented in the publication alongwith a comparison of them using the CNN and Daily Mail datasets against other cutting-edge extractive models. Their robust baseline model, called "BiLSTM," performs better than both the REFRESH model and the unsupervised Lead-3 technique. [9]

- *Mahmoud R. Alfarra, Abdalfattah M. Alfarra and Jamal M. Alattar "Graph-based Fuzzy Logic for Extractive Text Summarization (GFLES)"*

The DUC 2004 dataset, a benchmark for multi-document summarization written in English, is cited in the study. [10] It has several articles covering a wide range of topics, all of which are grouped together basedon the issues they cover.

- *Chuleepohn Yongkiatpanich and Duangdao Wichadakul "Extractive Text Summarization Using Ontology and Graph-Based Method" (2019)*

A review article, also known as a literature review, offers readers a fresh viewpoint by summarizing research works in connected subjects. It is advised to read review papers to get background information before beginning research. [11]

- *Cengiz HARK, Taner UÇKAN, Ebubekir SEYYARER and Ali KARCI "Graph-Based Suggestion for Text Summarization"*

By choosing phrases from the original text and applyinga linear weighting system to each sentence's relevance, the study suggests a technique for summarizing texts. The approach does not require in-depth language understanding; instead, it uses theoretically obtained pictures to represent the text. [12] It is adaptable to several languages.

- *Mrs. Chetana Badgujar, Mrs. Vimla Jethani and Mr. Tushar Ghorpade" Abstractive Summarization using Graph Based Methods" (ICICCT 2018)*

In the age of big data, text summarization is an essentialfield of study that aims to extract meaningful information and display it succinctly for ease of comprehension. Text summarization can be done in two ways: extractive and abstractive. Whereas abstractive summarizing creates new sentences utilizing significant information from the source content, extractive summarization focuses on removing essential elements from the source document. [13]

- *Rahul, Surabhi Adhikari and Monika "NLP based Machine Learning Approaches for Text Summarization" (ICCMC 2020)*

It is important to note that the paper focuses on text summarization models and their comparison scores, rather than specific datasets used for training or evaluation purposes The report also notes that data is important to us and that we use wireless and electronic devices to store and retrieve it. [14]

- *Samridhi Murarka and Akshat Singhal "Query-based Single Document Summarization using Hybrid Semantic and Graph-based Approach" (2020)*

Beginning with Luhn's model from 1950, which suggested automatically selecting material based on keyword recurrence, the study explores several summarizing frameworks used in related research efforts.. MEAD experimented with single and multi-document sources in 2001,[15] using features like sentence location, keywords, LCS, and TF-IDF methodology for generating summaries.

- *Mana Ihori, Naoki Makishima, Tomohiro Tanaka, Akihiko Takashima, Shota Orihashi and Ryo Masumura "Masked Pointer Generator Network forSequence-To-Sequence Pre-Training" (2021)*

A self-supervised learning technique for pointer-generator networks called the proposed Masked Pointer-Generator Network (MAPGN) [16] effectively trains the copy mechanism by teaching it to decide whether to copy or produce tokens versus masking span. In two spoken-text normalization tasks, MAPGN performedbetter than standard approaches, particularly in out-of- domain (OOD) tasks and when there is a minimal quantity of paired training data.

IV. SUMMARIZATION METHODS

A. Abstractive summarization

Abstractive summarization is a natural language processing (NLP) approach in which the meaning of the original text is interpreted and understood, and then a condensed, coherent representation is produced using natural language. Abstractive summarization methods may require rephrasing,paraphrasing, or even creating new terms to understand the context, extract important information, and provide succinct and coherent summaries. It calls for a thorough comprehension of the language, context, and the capacity to produce material that isn't stated clearly in the input. Abstractive summarizing approaches are continuously improved as NLP research advances to increase the quality ofgenerated summaries and their ability to be mistake-free from human-provided summaries.

B. Extractive summarization

In natural language processing (NLP), an approach known as "extractive summarization" creates a summary by picking and merging preexisting phrases or paragraphs from the original text. When extractive summarization is used, sentences are usually given relevance scores based on severalcharacteristics like word frequency, sentence length, or keyword occurrence. Even if extractive summarizing is effective and may provide coherent summaries fast, abstractive approaches may be able to capture context and subtleties more fully.

V. NATURAL LANGUAGE PROCESSING

Within the topic of artificial intelligence, natural language processing primarily focuses on human-computer interaction. Text summarization relies heavily on Natural Language Processing (NLP), which uses sophisticated algorithms and linguistic analysis to extract the most important information from a given document. By using methods like named entity recognition, tokenization, and part-of-speech tagging, naturallanguage processing (NLP) separates text into meaningful chunks, finds important phrases, and recognizes entities like names and locations. This semantic understanding is further enhanced by sentiment analysis, enabling the system to capture the emotional tone of the content.

By evaluating sentences according to a variety of linguistic characteristics, natural language processing (NLP) makes extractive summarizing easier and guarantees that the summary produced captures the most important ideas from the source material. NLP models use language creation and deep learning methods in abstractive summarization to generate fresh, logical phrases that capture the main idea. Furthermore, syntactic and contextual analysis contributes topreserving the summary's coherence and flow.

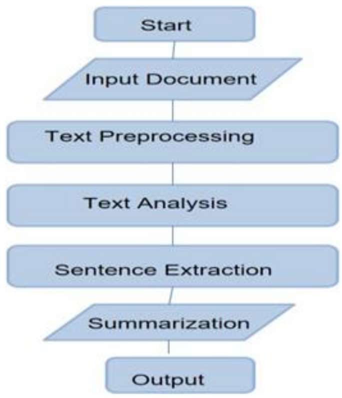
Key components of Text Summarization include: -

1) *Tokenization*: Tokenization is the process of dividing a text into smaller chunks, such words or tokens that are subwords. It is an essential phase in the processing of text.
2) *Text Representation:* Transform the text into a numerical representation so that machine learning algorithms may utilize it. Word embeddings (e.g., Word2Vec, GloVe), Bag-of-Words (BoW), and Term Frequency-Inverse Document Frequency (TF-IDF) are common techniques.
3) *Extractive Summarization:* In extractive summarization, important sentences or phrases are identified and selected from the original document to createextractive summary of it.
4) *Abstractive Summarization:* A summary is produced by abstractive summarization, which reinterprets and rewords the text. It entails coming up with original sentences and comprehending the situation.
5) *Attention Mechanism:* Abstractive summarization depends on attention processes. They allow the model to selectively collect essential information by allowing it to focus on different sections of the input text when creating the summary.
6) *Encoder-Decoder Architecture:* In sequence-to-sequence models, the summary is produced by the decoderusing the context representation that the encoder created after processing the input text.

Pre-trained Language Models: Pre-trained models with substantial performance in a variety of NLP tasks, including summarization, include BERT, GPT, and T5. Effective results may be obtained by fine-tuning these models on summarization datasets.

1) *Evaluation Metrics:* By contrasting generated summaries with reference summaries, metrics like ROUGE(Recall-Oriented Understudy for Gisting Evaluation) are frequently used to assess the quality of generated summaries.
2) *Dataset:* To train and test summarization algorithms, arepresentative and varied dataset is essential. For news stories, common databases include Gigaword and CNN/Daily Mail.
3) *Post-processing:* Refinement of the generated summary may involve removing redundant information, ensuring coherence, and improving readability.

These elements are combined in different ways based on theparticular summarizing task, the kind of text input, and the intended result. Scholars and professionals are still investigating and creating novel approaches to improve text summarizing systems' functionality.



VI. TEXT PREPROCESSING

When working with raw text data and cleaning it up to makeit readable for analysis, text preprocessing is an essential stepin natural language processing (NLP) and machine learning activities. To make the text easier to handle and perform better on subsequent NLP tasks, text preprocessing aims to eliminate noise, unnecessary information, and inconsistencies. In text preprocessing, the following methodsand procedures are often used:

1. Lowercasing: - Change every word to lowercase. Thisguarantees consistency and aids in preventing word repetitionin various instances.
2. Tokenization: - Divide the text into discrete words or phrases. Tokenization is an essential stage in many NLP activities since it facilitates word-level text analysis and processing.
3. Removing Punctuation: - Remove punctuation, such asexclamation points, commas, and periods, as these might addnoise and may not add much to some studies.
4. Removing Numerical data: - If numerical quantities are notpertinent to the analysis, leave them out of the text. When working on projects like topic modeling or sentiment analysis, this is typical.
5. Removing Stopwords: - Eliminate cliches (stop words) like "the," "and" and "is," which are frequently used in language but frequently have no bearing on particular analyses.
6. Stemming: - Simplify words to their most basic or fundamental form. Stemming, for instance, might change thewords "running," "ran," and "runner" to the root "run." This aids in expressing words' essential meanings.
7. Lemmatization: - Lemmatization distills words to theirmost basic form, just like stemming, but it takes word meaning and context into account. To determine a word's base or root form, morphological analysis and vocabulary areusually used.
8. Removing Whitespace: - To maintain consistency, removeextra space at the start and finish of phrases and paragraphs.

A crucial stage that changes according to the particulars of the NLP job and the properties of the text data is text preparation. It facilitates the creation of a clear, consistent representation of text that may be applied to a variety of tasks,including topic modeling, text categorization, sentiment analysis, and more.

VII. EXTRACTIVE SUMMARIZATION METHODS & FEATURES

In a research paper, extractive summarization is a method thatautomatically creates a summary of a text by picking out themost crucial lines or phrases. The summary is subsequently created by concatenating the chosen sentences or phrases. Extractive summarization techniques operate on the premisethat a text's most significant information is probably expressed in its most noticeable words or phrases.

Here are some methodologies of extractive summarization.

1) Frequency-based methods: These techniques rate each sentence in a text according to how frequently it appears. Thesentences chosen for the summary are then the ones with thegreatest ratings.
2) TF-IDF methods: These techniques use the term frequency-inverse document frequency (TF-IDF) of each phrase in a textto calculate a score. The value of a phrase in one document inrelation to its importance in other documents in a collection is measured by TF-IDF. The sentences chosen for the summary are therefore those with the highest TF-IDF scores.

Position-based methods: These techniques rate each sentence in a text according to where it falls within the text.

Sentences that start or close a document receive a higher score than phrases that fall somewhere in the center.

3) Graph-based methods: These techniques visualize a text asa graph, with nodes standing in for phrases and edges for connections between them. The location of a sentence in thegraph then indicates how important it is. Compared to abstractive summary techniques, extractive techniques offer several benefits.

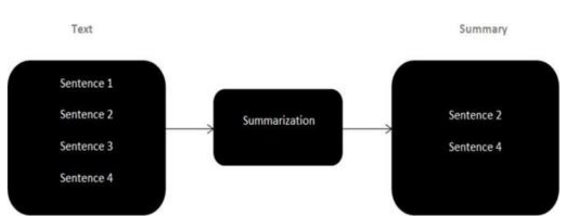
Some key features of extractive summarization methods.

1. Accuracy: The most significant sentences or phrases in a document may usually be found using extractive summarization techniques, which are often quite accurate.

2. Efficiency: Extractive summarization methods are typically very efficient, and they can generate summaries of long textsin a short amount of time.

3. Interpretability: Extractive summarization methods are typically very interpretable, and it is easy to understand why a particular sentence or phrase was selected for the summary.

4. Versatility: News stories, academic papers, and legal documents are just a few of the text kinds that may be summarized using extractive summarization techniques.



VIII. ABSTRACTIVE SUMMARIZATION METHODS & FEATURES

The process of automatically creating a condensed version ofa text document while maintaining the most crucial details and main ideas is known as abstractive summarization. Compared to extractive summarizing, which just chooses phrases from the source text to create the summary, it is a more difficult process.

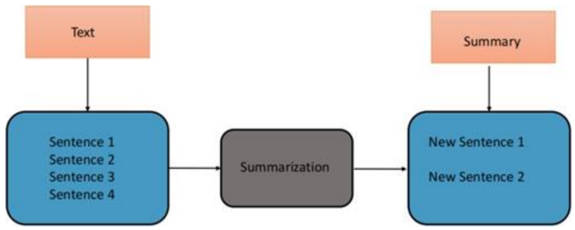
Here are some methods of abstractive summarization.

1. It was Bahdanau et al. (2014) that first proposed attention- based models. When creating the summary, they employ an attention mechanism to concentrate on the most pertinent passages from the input text. The various words in the input text are given weights by the attention mechanism, a learnable function. Each word's weight in the summary is determined by its level of relevance to the word that is currently being formed.

Cho et al. introduced encoder-decoder models for the first time (2014). The input text is encoded into a vector representation using an encoder, and the vector representation is then used by a decoder to produce the summary. Recurrent neural networks (RNNs), like long short-term memory (LSTM) networks, are commonly used asthe encoder. Similar to the encoder, the decoder is likewise usually an RNN, albeit it could be of a different kind.

2. The first introduction of graph-based models came from Yang et al. (2016). The words in the text are the nodes, and the relationships between the words are the edges, in their representation of the input text as a graph. The text's most crucial passages are then found using the graph. Usually, network algorithms like Text Rank and PageRank are used for this.

Pegasus Model: - Pegasus is a natural language processing model for abstract text summarization created by Google Research. Unlike extractive summarization, it concentrates on producing succinct and cohesive summaries. Using asizable dataset, the model is pre-trained and fine-tuned, acquiring contextual representations and linkages. By shuffling the input texts at random, it presents a new training target termed "gap sentence generation". Pegasus has variations such as PEGASUS-large and PEGASUS-xsum, which let customers select a model according to their own needs and available processing power.



IX. WEB INTERFACE: STREAMLIT

Streamlit is an open-source Python framework that facilitates the development and dissemination of stunning, unique online applications for data science and machine learning. It'sbasically a framework that lets you quickly create interactive web apps out of Python data scripts with very little coding knowledge.

Some of the key features of streamlit include:

1. Simple and intuitive: Streamlit boasts a clean and straightforward API that makes it easy for developers of all skill levels to get started, even those with limited web development experience.

2. Interactive visualizations: Streamlit allows you to easily build interactive charts, graphs, and dashboards to showcase your data in a visually appealing and engaging manner.

3. Rapid prototyping: With Streamlit, you can quickly prototype your data app ideas and get them up and running in minutes, significantly speeding up your development process.

4. Sharing and collaboration: Streamlit apps are easily deployed and shared online, allowing you to collaborate withothers, present your findings, or even share them publicly.

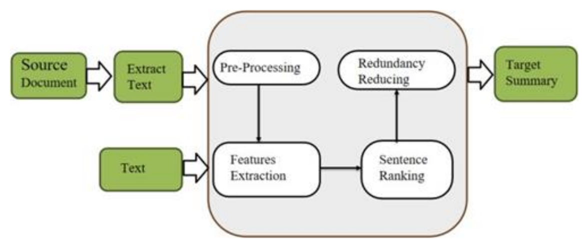
Streamlit is primarily used by data scientists, machine learning engineers, and analysts who want to:

1. Showcase their data insights: Streamlit offers a great way to create interactive dashboards and reports that effectively communicate complex data analysis to stakeholders.
2. Prototype and experiment: Streamlit's rapid prototyping capabilities make it ideal for quickly testing out ideas and experimenting with different data visualization techniques.
3. Build internal tools: Streamlit can be used to create custom web applications for internal use within organizations, such as data exploration tools or machine learning prediction models.

Streamlit is actively maintained and constantly being improved with new features and functionalities. There is a growing community of Streamlit users who share their apps and ideas, providing a valuable resource for learning and inspiration.

A large user base may utilize Streamlit because it is open source and free.

X. SYSTEM ARCHITECTURE DIAGRAM



1. File upload: It allows the user to upload the document file to the user interface manually by selecting files from their local disk.
2. Extracting document text: This module extracts the textual information out of the uploaded document so this information can be preprocessed further.
3. Text preprocessing: This covers text preparation techniques including stop word removal, tokenization, and lemmatization, among others. This is an important step in cleaning the data and training the summarization model withorganized, clean data.
4. Sentence ranking: This module refers to the task of determining the relevance or importance of different sentences within a given document or set of documents.
   Several sentence ranking techniques include TF-IDF, Word embeddings, Graph based methods and so on.
5. Abstractive summarization: This module generates concise summaries by understanding the meaning of the input text and creatively expressing key information. It can provideshorter, more readable summaries of the original information by

rewording sentences and using new terms, in contrast to extractive summarization.

6. Extractive summarization: To provide a summary, this module picks and extracts important sentences or phrases from the original content. It relies on identifying the most relevant information without generating new content, providing a condensed version that retains the wording of the original document.

ACKNOWLEDGMENTS

We would like to thank Prof. Sachin Balvir and Prof. SameerTembhurney of the Department of Computer Science & Engineering for being our Project Guide and Co. Guide, respectively, for being the backbone of our project. Their unwavering encouragement and direction throughout our moments of uncertainty allowed us to continue with our initiative.

This research would not have been feasible without the essential direction, encouragement, support, and inspiration from Dr. Mrudula Nimbarte, Head of Department, ComputerScience & Engineering.

We especially appreciate the support and well wishes we received from Dr. S. L. Badjate, Principal of the S.B. Jain Institute of Technology, Management & Research.

We are deeply grateful to the management of S.B. Jain Institute of Technology, Management & Research for providing the lab facilities and required infrastructure.

We would also like to thank the Computer Science & Engineering Department teachers and non-teaching personnel for their assistance in seeing our project through tocompletion.

CONCLUSION

These days, a text summary is a helpful tactic. It may condense a large amount of information into a more comprehensible manner. The rapid advancement of technology and the pervasive usage of internet connection often surpass the capacity for information transport. Text summarizing may simply fix the problem and provide the user with a summary document. The two most important features are that it can provide efficiency and that information can be generated faster. It may cut out redundant material, concentrate on key elements, and get rid of extraneous details in text summaries. We must distinguish between a variety oftext summary techniques in this review study to summarize the texts. improvement and growth in the future scope of thisproject. NLP model and algorithm developments might improve summarization and guarantee even higher coherence and accuracy when extracting difficult study information. Contextual embeddings and transformer architectures are two cutting-edge approaches that might be used to create summaries that are more contextually aware and detailed. To sum up, the potential applications of automated research paper summary include ongoing improvement, incorporating new technologies, and broadening the tool's range of linguistic and subject-specific settings.

FUTURE SCOPE

Natural Language Processing (NLP) techniques for automated generation of concise research paper summaries marks a significant stride towards addressing the challenges posed by the exponential growth of scientific literature. Looking ahead, the future scope of this endeavor holds promising avenues for refinement and expansion. Further advancements in NLP algorithms and models could enhance the summarization process, ensuring even greater accuracy and coherence in distilling complex research content. The integration of cutting-edge techniques such as contextual embeddings and transformer architectures may contribute to more nuanced and contextually aware summaries. In conclusion, the future scope of automated research paper summarization lies in continual refinement, technological integration, and expanding the tool's adaptability to diverse linguistic and subject-specific contexts. This trajectory holds the potential to revolutionize how researchers navigate and engage with the ever-expanding landscape of scientific literature.

REFERENCES

[1] Yaser Keneshloo, Tian Shi, Naren Ramakrishnan and Chandan K. Reddy. “Deep Reinforcement learning for Seq2SSeq.”

[2] Shuai Zhao, Fucheng You and Zeng Yauan Liu "Abstractive Text Summarization Using Pretrained Language Models. (December 31, 2020)

[3] Eduard Hovy and Chin-Yew Lin "Automated text summarization and the summarist system."(Southern California 4676 Admiralty Way Marina del Rey, CA 90292-6695)

[4] Anish Jadhav, Rajat Jain, Steve Fernandes and Mrs. Sana Shaikh "Text Summarization using Neural Networks."(2019)

[5] Chandra Prakash and Dr. Anupam Shukla2"Text Summarizer "SAAR" using Reinforcement Learning."(Madhya Pradesh, India – 474010-2014)

[6] Urvashi Khandelwal, Kevin Clark, Dan Jurafsky, Lukasz Kaiser"Text Summarization Using a Single Pre-Trained Transformer."(May 2019)

[7] Thevatheepan Priyadharshan and Sagara Sumathipala "Text Summarization for using NLP"(2018).

[8] Jingwei Cheng, Fu Zhang and Xuyang Guo "A Syntax- Augmented and Headline-Aware Neural Text SummarizationMethod" (2020)

[9] Laifu Chen and Minh Le Nguyen "Sentence Selective Neural Extractive Summarization with Reinforcement Learning" (2019)

[10] Mahmoud R. Alfarra, Abdalfattah M. Alfarra and Jamal M. Alattar "Graph-based Fuzzy Logic for Extractive Text Summarization (GFLES)"

[11] Chuleepohn Yongkiatpanich and Duangdao Wichadakul "Extractive Text Summarization Using Ontology and Graph- Based Method" (2019)

[12] Cengiz HARK, Taner UÇKAN, Ebubekir SEYYARER and Ali KARCI "Graph-Based Suggestion for Text Summarization".

[13] Mrs. Chetana Badgujar, Mrs. Vimla Jethani and Mr. Tushar Ghorpade" Abstractive Summarization using Graph Based Methods" (ICICCT 2018)

[14] Rahul, Surabhi Adhikari and Monika "NLP based Machine Learning Approaches for Text Summarization" (ICCMC 2020)

[15] Samridhi Murarka and Akshat Singhal "Query-based Single Document Summarization using Hybrid Semantic and Graph-based Approach" (2020)

[16] Mana Ihori, Naoki Makishima, Tomohiro Tanaka, Akihiko Takashima, Shota Orihashi and Ryo Masumura "Masked Pointer Generator Network for Sequence-To- Sequence Pre-Training" (2021)