International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

Gen Text Summarization AI: Generative Text Summarization Using BERT and Transformer Models

[1]Bheemanathi Harini,[2]Dr.M.Ajay Kumar,

[1]M.Tech Scholar,Dept. of CSE, Malla Reddy Technical Campus,Malla Reddy Vishwavidyapeeth, Maisammaguda,Hyderabad, Telangana 500100, India.

Mail id :bheemanathi2204@gmail.com

[2]Associate Professor, Dept. of ECE, Malla Reddy Technical Campus,Malla Reddy Vishwavidyapeeth, Maisammaguda,Hyderabad, Telangana 500100, India.

Mail id: ajaykumar.miryala@mrvv.edu.in

ABSTRACT

Users often encounter challenges while trying to read, comprehend, and extract essential information from massive amounts of text due to the exponential increase of digital material across reports, papers, and multimedia platforms. Summarizing large papers or audio information by hand is a tedious and wasteful process. Although they use phrases taken straight from the source material, traditional extractive summarizing methods often miss the mark when it comes to capturing the content's actual context and semantic significance. In light of these difficulties, the current research suggests a Smart Text Summarization System that can generate relevant and succinct summaries from a variety of input formats by using the GenAI PEGASUS Model. Summarizing material from many sources is made possible via a single platform by the suggested system, which supports text, PDF, DOCX, and voice inputs. Prior to summarization, voice inputs undergo speech-to-text processing, which transforms audio into text. By comprehending the whole context instead than just extracting phrases, the PEGASUS transformer-based abstractive summarization model produces summaries that seem human-like. An approachable user interface is provided by the system's implementation, which makes use of Python, deep learning models, and Natural Language Processing (NLP) methods. Users are able to quickly produce summaries, read the extracted content, and download the PDF result from the program. The accuracy, scalability, and simplicity of maintenance are guaranteed by the modular architecture. The approach considerably reduces the time necessary for information comprehension while producing coherent, relevant, and accessible summaries, as shown by the experimental findings. All things considered, this project provides a smart and effective way to automatically summarize text and speech, which is great for academics, professionals, and anybody generating content.

INTRODUCTION

Users are facing difficulties in digesting information due to the exponential expansion of digital material across multimedia platforms, academic repositories, and the internet. Research papers, reports, and manuals are lengthy materials that need a lot of reading and comprehension. For this reason, smart systems that can automatically condense large amounts of material into its most important points are in high demand. Summarization of texts is a significant use of Natural Language Processing (NLP) that attempts to condense lengthy texts while maintaining their essential concepts. The first methods for creating concise summaries used extractive approaches, which use statistical traits to choose key lines from the source material. These approaches can provide incomplete summaries since they lose the context and semantic understanding, even if they are successful to a certain degree. Text summarization has been greatly enhanced by the latest developments in deep learning and transformer-based architectures. Models for abstract summarization, like PEGASUS, may learn the meaning and context of the input text and use that knowledge to generate new sentences. By better capturing the original content's purpose and

providing summaries that seem natural, these models outperform more conventional approaches. Text, PDF, DOCX, and audio recordings are all acceptable input formats for this project's proposed Smart Text Summarization System. The system utilizes Whisper for audio inputs and employs an abstractive summarizing model based on transformers to provide accurate summaries. Built as a safe web app using Flask and Python, the system makes information more accessible, reading is faster, and it's useful for work, school, and daily life.

MOTIVATION

Users are facing a major problem with information overload due to the exponential growth of digital material across websites, academic platforms, and multimedia sources. Reading and understanding long pieces of writing, articles, and recorded material may be a real challenge for many people. There is an increasing need for automated systems that can effectively summarize massive amounts of data into digestible chunks due to the complexity of the problem. Students, academics, and professionals that often deal with large amounts of textual material find traditional reading techniques and manual summarizing to be particularly inefficient and time-consuming. Furthermore, current summarizing technologies often only work with a small selection of input formats, which isn't enough to handle the variety of contemporary material types including audio recordings, presentations, and papers. Because of this restriction, a unified summarizing platform that is capable of handling various data sources without any hitches is essential. Recent developments in deep learning models and Natural Language Processing have shown incredible promise in comprehending and producing language that is similar to human speech. By identifying textual contextual and semantic links, systems based on transformers make abstractive summarization possible. Intelligent systems that can increase readability and coherence in addition to properly summarizing text beyond classic extraction methods may be developed by using these technologies. The goal of this project is to create a sophisticated summary system that combines speech-to-text functionality with powerful natural language processing models in an approachable and intuitive way. The system's goal is to increase productivity, decrease cognitive load, and improve information accessibility by allowing text, document, and voice inputs inside a unified web-based platform. In the end, this initiative helps with the development of smart digital tools that people may use to better manage their information in a future where data is king.

OBJECTIVES

Smart Text Summarization aims to accomplish the following primary goals: • Construct an automated system that can sift through mountains of content and provide useful, succinct summaries. Among these sources of unstructured text are text files, PDFs, and Word documents; the goal is to clean and preprocess the data. • Summarizing abstractive text using deep learning models based on transformers. 1. To facilitate audio input by means of text-to-speech conversion and the generation of precise summaries. • To cut down on data overload by gleaning important points while keeping the whole context intact. • To cut down on data overload by gleaning important points while keeping the whole context intact.

SCOPE

The goal of this project is to create an automated system for Smart Text Summarization that can sift through mountains of text and extract relevant and short summaries. This system is very flexible and easy to use since it accepts a wide variety of input forms, including plain text, PDFs, Word documents, and voice input. It uses deep learning models based on transformers and state-of-the-art Natural Language Processing methods to guarantee accurate summarization. By effectively displaying crucial information, the initiative hopes to alleviate information overload and save users' time. For the sake of convenience and ease of use, a web-based interface is set up. Both dependability and performance are enhanced by efficient processing techniques and safe file management. Use cases in academia, research, and the professional realm are well-suited to the system. You can use it hands-free thanks to voice-based summarization. The produced summaries are available for users to download for offline usage. You may extend the scope to include multilingual summarizing, which would allow users to summarize text in multiple

languages and make the system available to a bigger worldwide audience. Other possible extensions include cloud deployment, mobile application integration, and URL summarization.

LITERATURE SURVEY

Abstractive Text Summarization

Abstractive text summarization generates concise summaries by understanding the semantic meaning of the input content and creating new sentences. Unlike extractive methods, it produces more natural and human-like summaries, making it effective for long documents and speech-based content.

Transformer-Based Language Models (PEGASUS)

PEGASUS is a transformer-based deep learning model specifically designed for text summarization tasks. It uses self-attention mechanisms to capture long-range dependencies in text, enabling accurate and context-aware summary generation.

Natural Language Processing (NLP)

Natural Language Processing enables machines to analyze, interpret, and generate human language. It involves text preprocessing, tokenization, semantic analysis, and language generation, which collectively support efficient summarization of textual data.

Text Processing for Multiple Input Types

Text processing enables the system to handle and normalize content from various input sources such as typed text, PDF documents, Word files, and voice recordings. Voice inputs are first converted into text using speech recognition techniques, after which all inputs undergo cleaning, tokenization, and preprocessing to ensure consistent and accurate summarization using NLP-based models.

Chunking

Long documents are divided into smaller chunks to handle memory limitations and improve summarization efficiency.

Flask Web Framework

Flask is a lightweight Python web framework used to build the backend of the summarization system. It handles user requests, file uploads, model execution, session management, and response rendering, enabling seamless interaction between users and the AI models through a web interface.

RELATED WORK

1. Ani Nenkova and Kathleen McKeown's "A Survey on Automatic Text Summarization" (2012) provides a thorough overview of methods for automatic text summarizing, including both abstractive and extractive approaches. A solid theoretical basis for current NLP-based summarizers is laid forth in the discussion of assessment criteria, difficulties, and summarizing system implementations.Two, "A Study of Abstract and Extracted Texts" Findings from the study "Summarization" by K. Y. Chen and S. H. Liu (2019) contrast abstractive and extractive summarizing strategies, drawing attention to the drawbacks of extractive methods when it comes to maintaining coherence. This project's methodology is supported by the study's findings that abstractive summarization provides summaries that are more readable and relevant. Ryangsoo Kim and Sung-Hyon Myaeng's (2017) "Text

Summarization Using Deep Neural Networks" delves into the use of deep neural network models to automate the process of text summarization. By successfully extracting contextual and semantic information from massive manuscripts, the authors show that neural techniques beat conventional statistical methods. Using Extracted Gap Sentences for Abstractive Pre-training (PEGASUS) A review article published in 2020 by Jingqing Zhang et al.: The PEGASUS transformer paradigm, developed for the express purpose of abstractive text summarization, is presented in this study. The model's exceptional performance on several benchmark datasets makes it an excellent choice for document, PDF, and article summarizing applications. 5. In their 2019 study, "Abstractive Text Summarization Using Transformer Models," Liu and Lapata examine structures based on transformers for abstractive summarization. They find that these designs significantly increase fluency and contextual comprehension. Intelligent summarization frameworks should include transformer models, according to the results. 6. In their paper titled "Abstractive Summarization of Spoken Content Using Neural Models," Yang Liu, Daniel Hakkani-Tür, and Larry Heck (2021) used speech-to-text approaches and neural abstractive summarizing models to transform spoken audio into text.

EXISTING SYSTEM

For the most part, current text summarizing systems depend on human editors or simple extraction approaches to cherry-pick relevant phrases from the original text. In order to find important phrases, many online summarizers rely on basic frequency-based or rule-based algorithms that fail to grasp the underlying semantic context of the material. The summarizing capabilities of these tools are generally restricted to plain text, and they do not support or only partially support document formats like as PDF and DOCX. Also, voice summarization is very uncommon and usually requires a separate manual speech-to-text conversion before it can be used. In order to extract text for summarization, users often need to manually convert files or utilize many separate applications. Current systems are not entirely automated, have limited format support, and are often fragmented.

Limitations of Existing System:

1. The majority of current summarizing systems use extractive approaches, which often lead to incoherent summaries that don't convey the intended meaning.
2. The majority of current summarizing systems use extractive approaches, which often result in summaries that are disjointed and miss the mark when it comes to conveying the essential ideas.
3. When it comes to input support, the majority of tools fall short. They usually only work with plain text and don't have a smooth way to integrate with PDFs, DOCX files, or voice-based inputs.

PROPOSED SYSTEM

Using a variety of input types such as text, PDFs, DOCX files, and voice inputs, the suggested system effectively creates smart text summaries that are both succinct and relevant. The system uses the PEGASUS model, developed for high-quality text summarizing, and other deep learning-based abstractive summarization approaches. Instead of just pulling lines from the source material, the system uses PEGASUS to understand the context and create summaries that seem human. The approach incorporates speech-to-text processing to transform audio into text before applying summarization to voice inputs.

Advantages of Proposed System:

The system's versatility and ease of use are enhanced by its support for many input types, including text, PDF, DOCX, and voice. It can also analyze and summarize material from these sources. • Superior Abstractive Summarization: The system outperforms conventional extractive approaches in terms of accuracy and significance by producing summaries that mimic human writing style and take context into account (PEGASUS deep learning

model). • Saves Time and Reduces Effort: Users can swiftly comprehend massive amounts of information thanks to the automatic summary process, which drastically cuts down on manual reading.

IMPLEMENTATION

SOFTWARE REQUIREMENTS

· **Operating System:** Windows 10 / Windows 11 / Linux

· *Programming Language:* ***Python 3.8 or above*** · *Web Framework:* ***Flask*** · *Frontend Technologies:* ***HTML, CSS, JavaScript*** · *Deep Learning Framework:* ***PyTorch***

NLP & AI Libraries: Transformers (PEGASUS model), Whisper

Document Processing Libraries: PyPDF2 (PDF text extraction), pythondocx (DOCX handling), ReportLab (PDF generation)

HARDWARE REQUIREMENTS

Processor: Intel Core i5 or higher

RAM: Minimum 8 GB (16 GB recommended for faster processing)

Storage: Minimum 50 GB free disk space

GPU (Optional): NVIDIA GPU with CUDA support for faster model inference

IMPLEMENTATION OF MODULES

A modular design was used to construct the Smart Text Summarization system. This architecture guarantees scalability, flexibility, and simplicity of maintenance. With the use of Natural Language Processing (NLP) and Generative AI approaches, each module carries out a distinct function to transform raw input data from various formats into brief yet relevant summaries. Module 1: User Authentication Secure access to the system is maintained by this module, which handles user registration and login capabilities. Secure access to the system is maintained by this module, which handles user registration and login capabilities. 2. The Input Handling Module Users are able to submit material in various forms with the help of the input handling module. • Entering text into a designated space • The ability to upload PDF files • Document XML file uploading • Audio file uploading This section determines the kind of input and sends it on to the appropriate part of the computer. Module 3: Data Extraction • The uploaded files may have their text content extracted by this module. • PyPDF2 is used to process PDF files in order to extract text that can be read. • The content extraction operation for DOCX files is handled by python-docx. • The Whisper speech-to-text paradigm is used to transform audio recordings into text. Preprocessing relies heavily on the retrieved text. 4. The Module for Preprocessing Text To get the retrieved text ready for summarization, the preprocessing module cleans and prepares it. • Eliminating noise, excess spaces, and symbols that aren't needed. • Enhancing text uniformity via normalization. • Reducing the size of lengthy papers for faster processing. The summarizing model will work at its best with this setup. Module No. 5: Abstractive Summarization The system's central module is this. • Abstractive summaries are generated using the PEGASUS transformer-based approach. • Rather than just paraphrasing, the model takes into account context and semantics. To create a final, cohesive summary, chunk-wise summaries are integrated. 6. The Output Module for Summary The created summary is shown to the user via the web interface via the Summary Output Module. Instantly upon completion of the summarizing procedure, the system presents the summary on the screen in a succinct and comprehensible format. Users may easily grasp the main ideas of the source material without any further processing

International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

since the output is appropriately structured to enhance readability. Module 7: PDF Generation • Users may download the summary in a portable document format (PDF) using the PDF Generation Module. • The ReportLab library is used by this module to transform the summary text into a properly organized PDF file. • The system's usability and practical utility are enhanced since users may keep, distribute, or view the summary material offline using the created PDF.

RESULTS

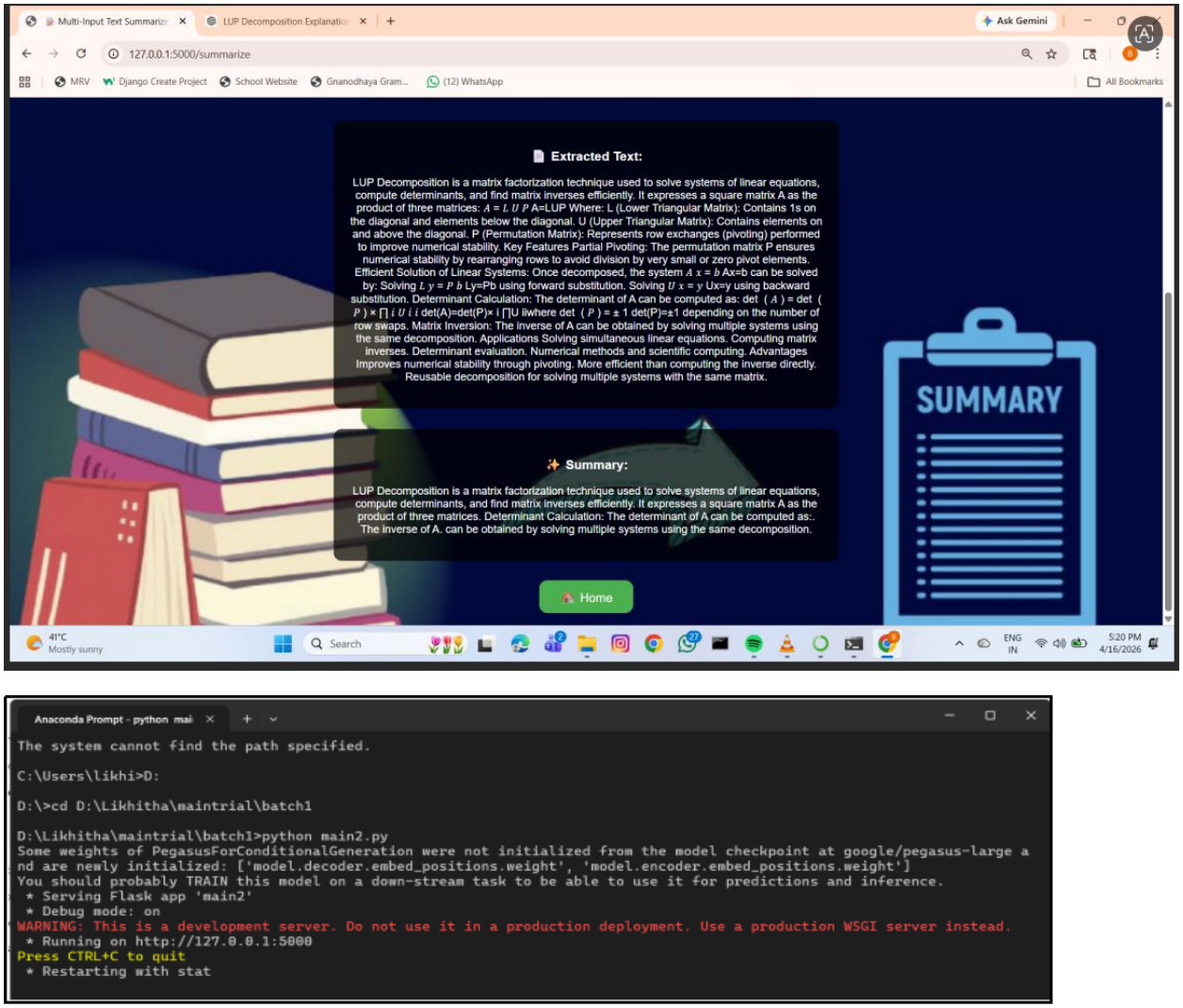


Fig 1: Application Running in Anaconda Prompt

This image depicts the Smart Text Summarization application, which is built using Flask and runs on Python, successfully executing. At http://127.0.0.1:5000, the Flask development server launches locally, and the PEGASUS model is loaded. Common warnings about using a development server and partly populated model weights are also shown on the console.

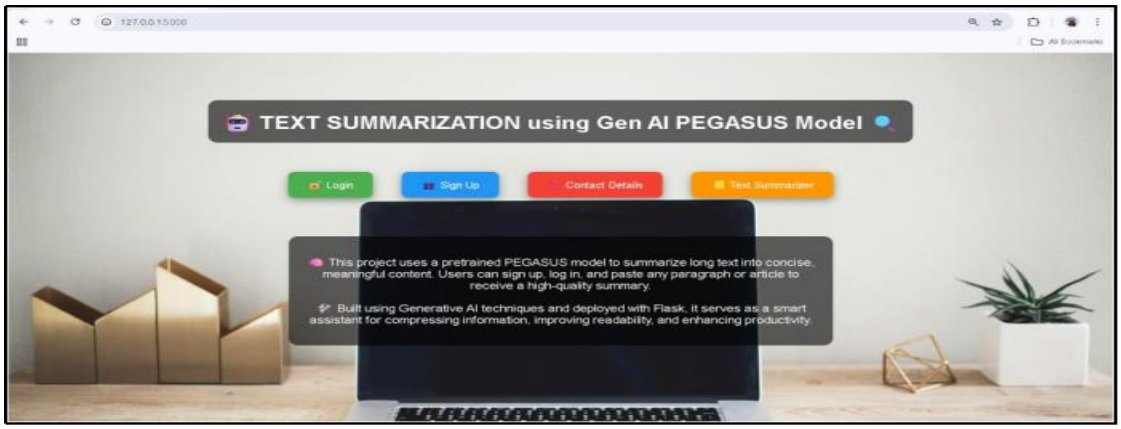


Fig 2: Home Page

The home page is where users may access the Gen AI PEGASUS model-based Smart Text Summarization system. Users may swiftly access system functionalities using the straightforward navigation choices provided, which include Login, Sign Up, Contact Details, and Text Summarizer. The project's goal is to use sophisticated Generative AI methods to create relevant summaries from extensive texts. It will be done using a Flask-based web application.

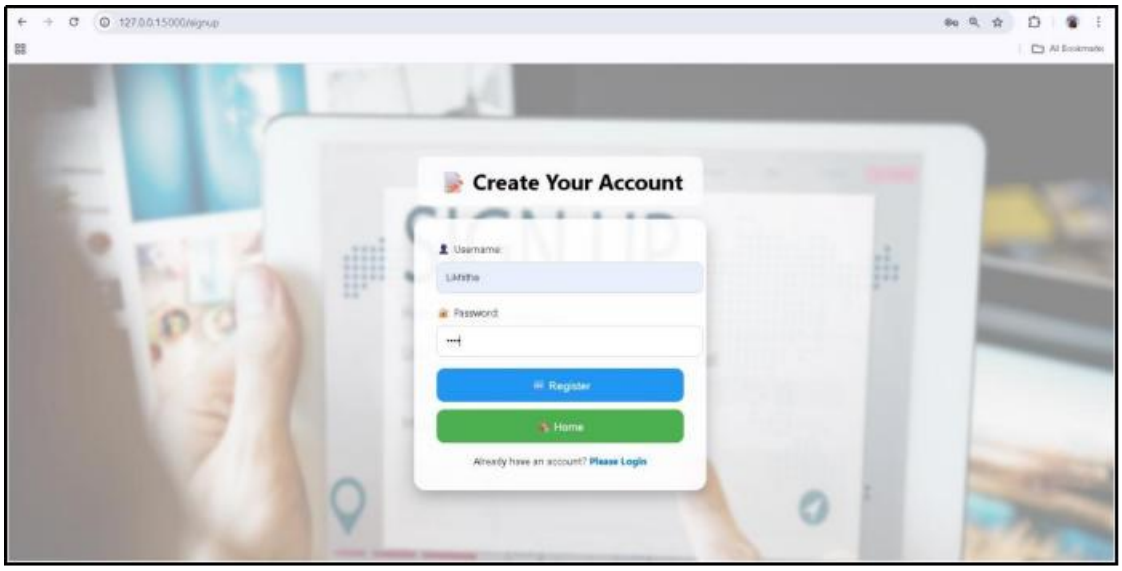


Fig 3: Signup Page

To register for an account, visitors may enter their details on the SignUp page. Secure input fields and choices to register, return to home, or browse to login are part of its sleek and user-friendly layout. This site allows users to register for the Smart Text Summarization system in an authorized way, which assures restricted access.

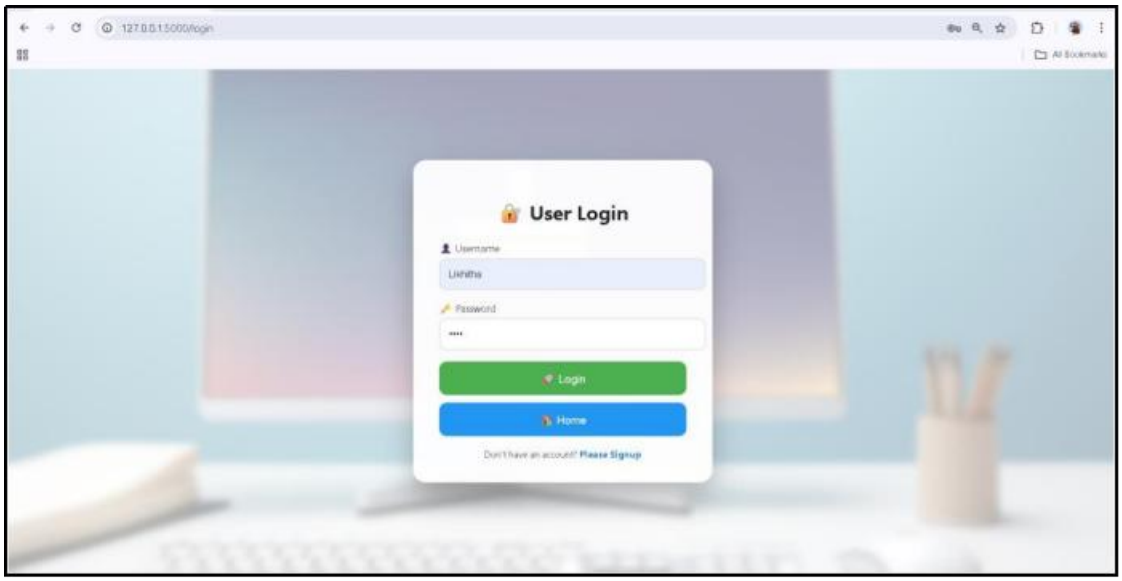


Fig 4: Login Page

By entering their username and password on the Login page, registered users may safely use the text summarizing system. With its user-friendly layout, you can easily log in, go back to the homepage, or access the sign-up page. This page verifies that only authorized users may access the summary features that are based on PEGASUS.

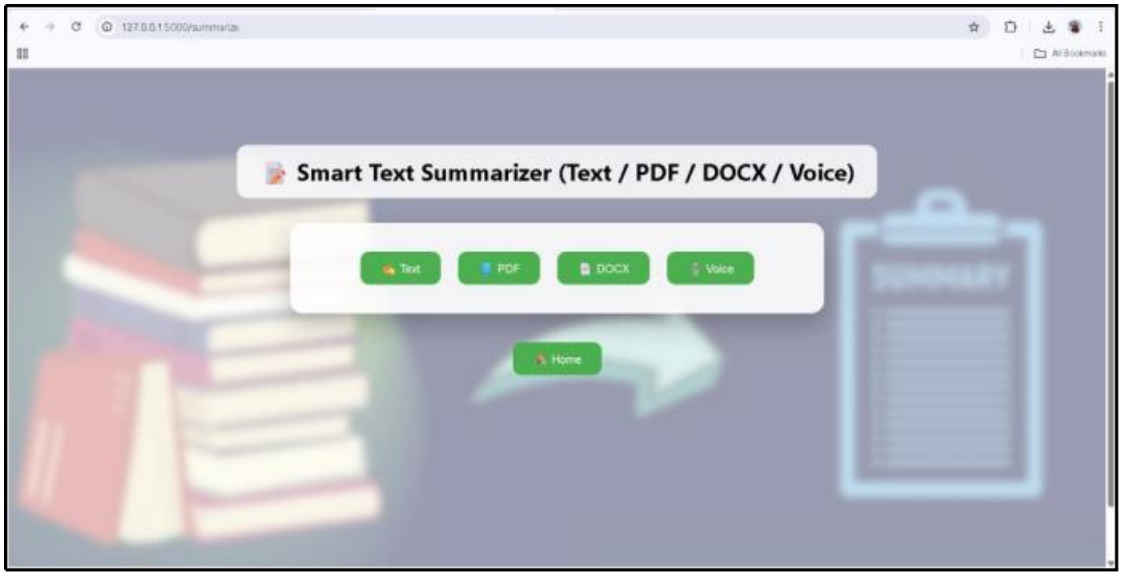


Fig 5: Summarization Page

Users may enter plain text, PDF, DOCX, or even voice commands into the Text Summarization site to get brief summaries. To get accurate and useful summaries, users may choose the input type they choose, upload or contribute material, and process it using the PEGASUS-based generative AI model. Then, it's simple to go back to the main site.

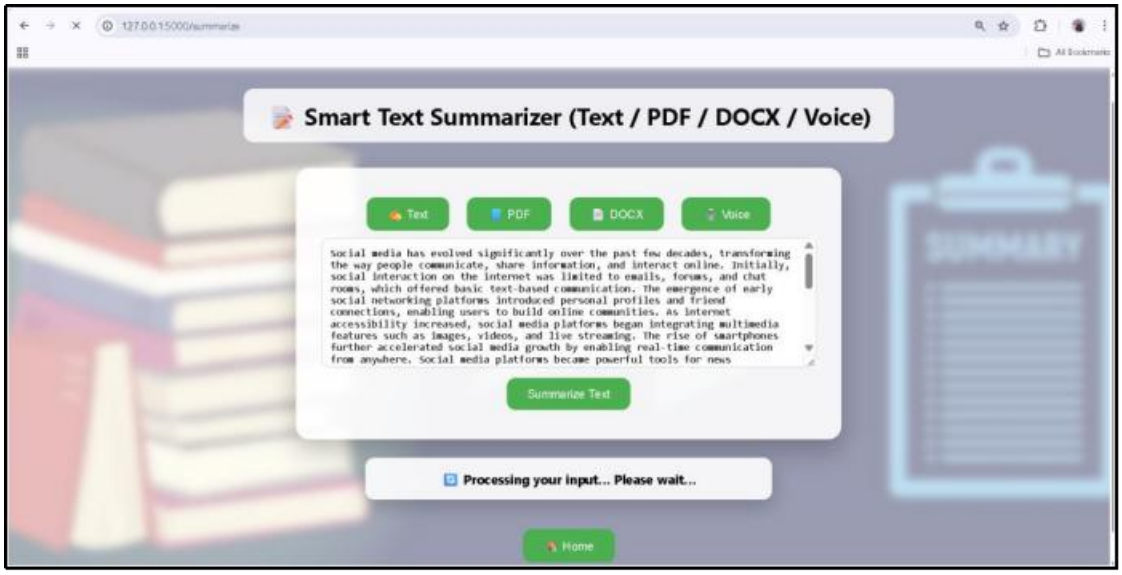


Fig 6: Text (Input)

The Smart Text Summarizer application's user interface is seen on this screen. When users choose the Text input option for summary, they may paste or enter lengthy textual information straight into the text box. By clicking the Summarize Text button after entering the information, the user may see the "Processing your input" status message appear while the system processes the input using the PEGASUS model.

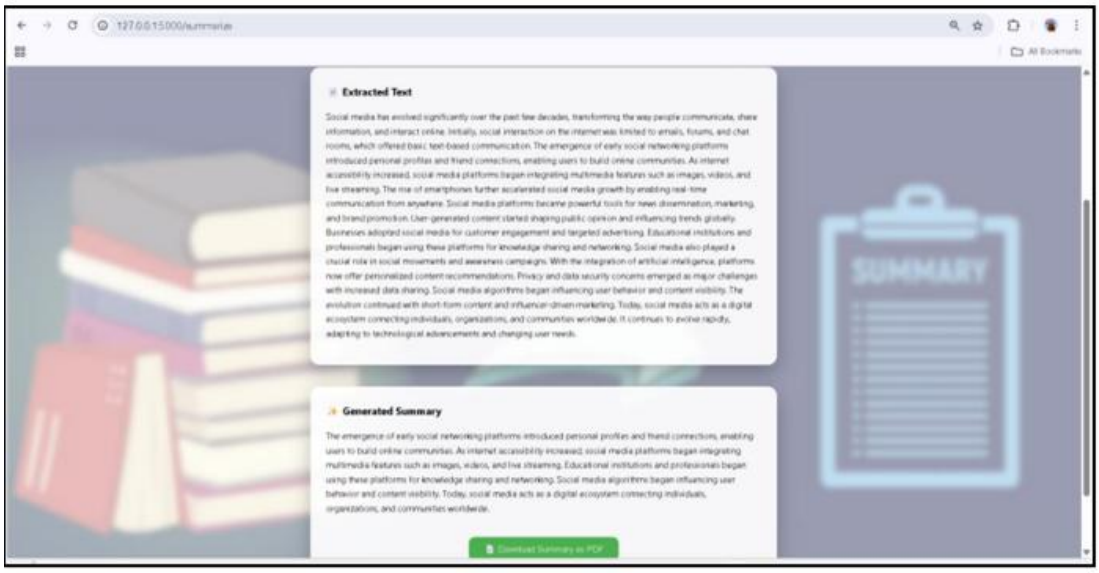


Fig 7 Text Summary Result

After text extraction and preprocessing, the user's input is shown in the top area as the Extracted Text. This includes the complete original content. Following it, in the Generated Summary section, you will find a meaningful and succinct summary that was generated using the PEGASUS abstractive summarizing model. This summary captures the main concepts from the original text. A "Download Summary as PDF" option is also available on the site, which makes the system more practical and user-friendly by letting users download the summary output for offline usage.

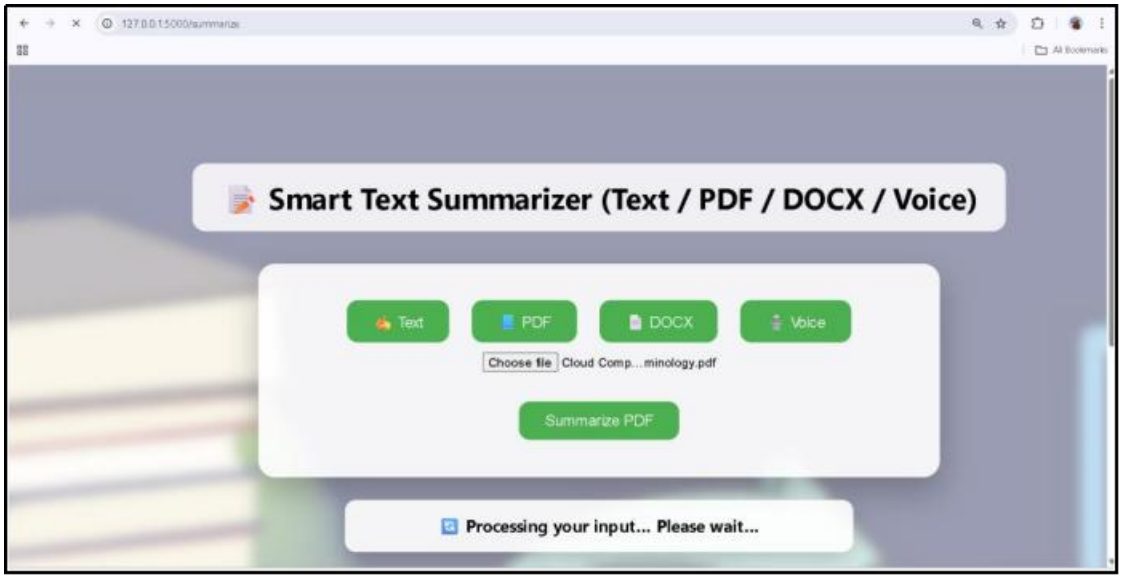
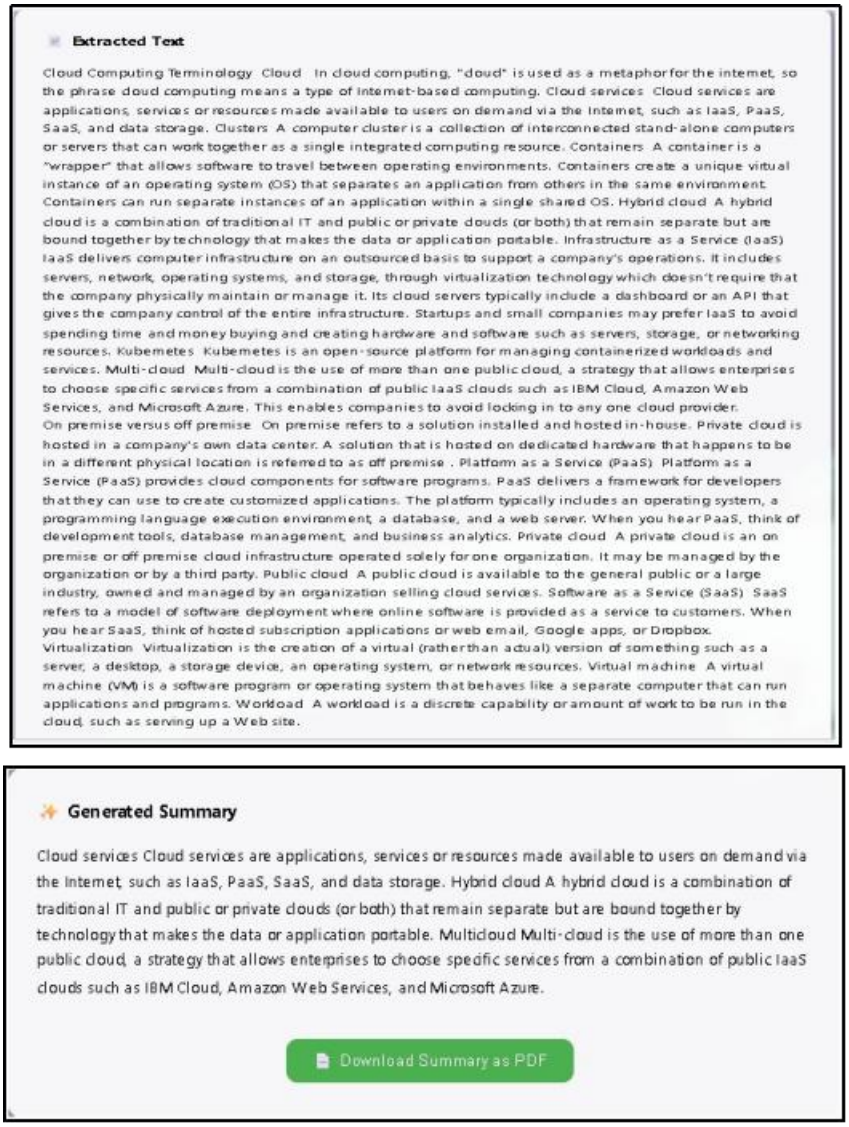


Fig 8: PDF (Input)

Here we can see the Smart Text Summarizer's PDF input interface. When utilizing the file selection option, users may upload a PDF document and choose it as the PDF input type for summarizing. After the user chooses a file, the system extracts text, does some preliminary processing, and then uses the PEGASUS model to provide a brief



summary.

Fig 9: PDF Summary Result

The Smart Text Summarizer application's PDF Summary Result is shown on this screen. The Extracted Text, which includes all of the text from the uploaded PDF after text extraction was complete, is shown in the top portion. The

PEGASUS abstractive summarization approach creates a clear and insightful summary by extracting the main ideas from the source content; this process is detailed in the Generated Summary section below. Users may save the summary material for offline use or sharing using the "Download Summary as PDF" button. This output view shows how well the algorithm was able to condense long PDFs into concise, understandable summaries.

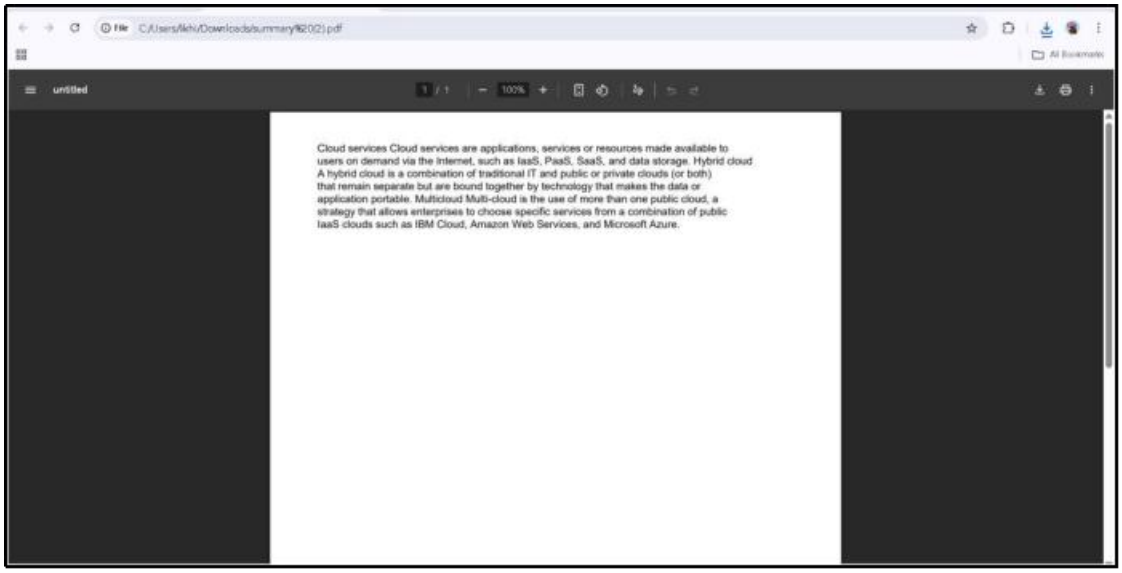


Fig10: Downloaded PDF Summary Result

This picture is of the PDF summary that the Smart Text Summarizer program created and saved for later use. This one-page PDF provides a brief overview of cloud services, including infrastructure as a service (IaaS), platform as a service (PaaS), software as a service (SaaS), hybrid cloud, and multi-cloud. Converting the system-generated summaries into a polished and user-friendly PDF document guarantees clear readability, simple sharing, and offline access.

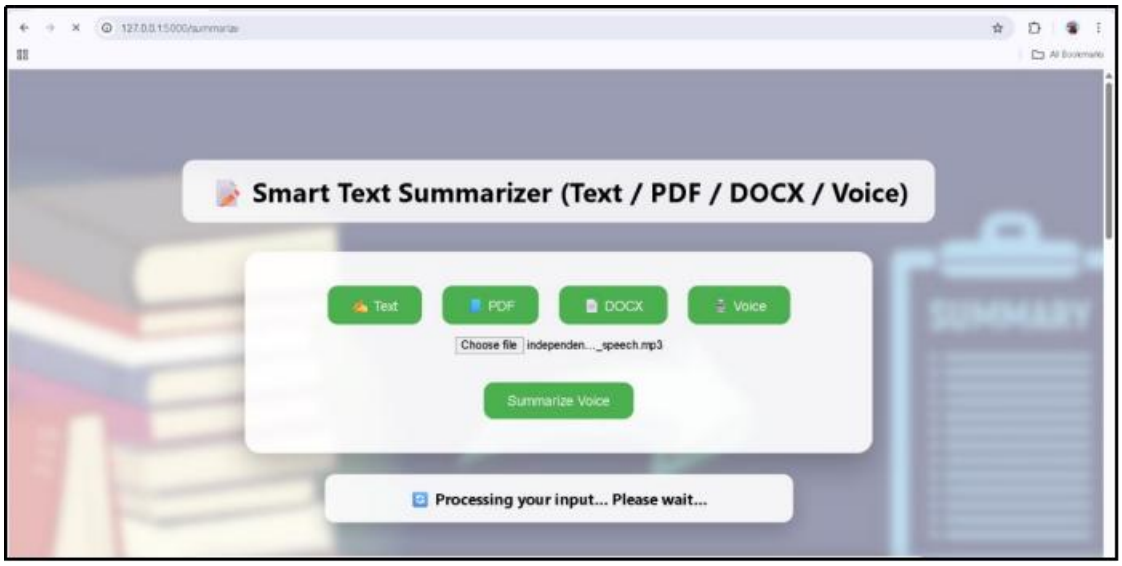


Fig 11: Audio (Input)

The Smart Text Summarizer system's Voice Input capability is seen on this screen. Instead of text or documents, users may enter voiced sounds. The system uses speech-to-text processing to transform the voice input into text. The extracted text is then fed into the PEGASUS summarizing model, which produces a clear and short summary.

Fig 12: Audio Summarization

The Smart Text Summarizer system's Voice Summarization output is shown on this screen. To ensure transparency, the speech-to-text module is used to turn the voice input into text. The extracted material is then shown under the Extracted Text section. The PEGASUS abstractive summarization model is then used to create a clear and succinct summary of the speech that is shown under the Generated Summary section. This summary effectively conveys the

main points of the speech. This function is great for audio-based material like speeches and lectures since users can easily compare the actual transcribed speech with the summarized output and download the final summary as a PDF.

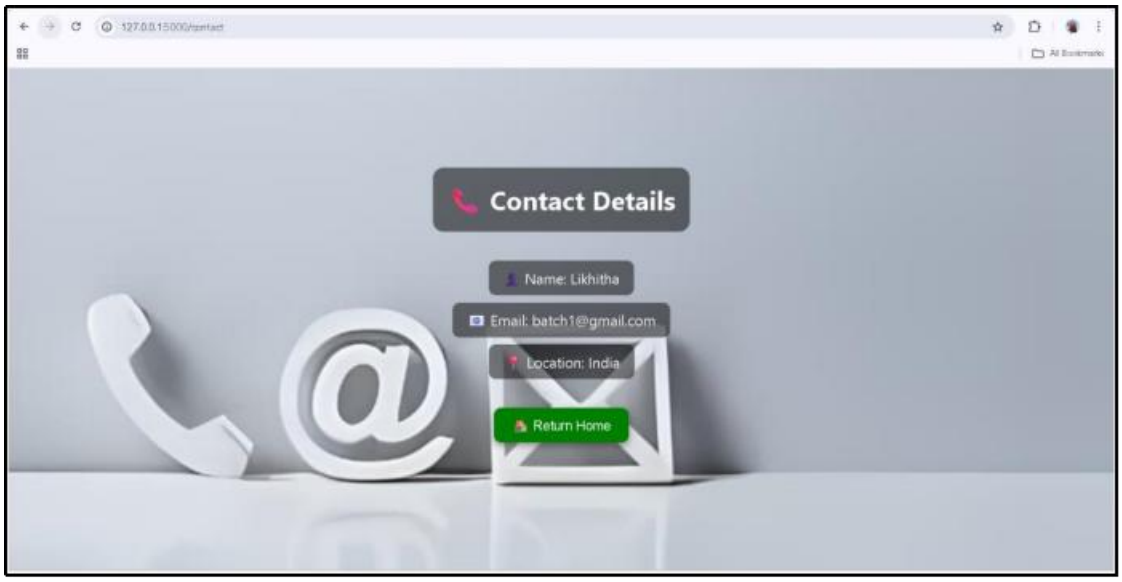


Fig 13: Contact Page

CONCLUSION

An effective tool for combating the increasing problem of information overload is the Smart Text Summarizer that employs the Gen AI PEGASUS Model. Users in the modern digital age are bombarded with audio and textual material in the form of speeches, articles, reports, and papers, which makes it challenging to swiftly extract crucial information. This project offers a smart and efficient solution by combining text, PDF, DOCX, and voice inputs to create summaries that are brief, relevant, and sound natural. Instead of only extracting phrases, the system is able to comprehend context thanks to the integration of modern Natural Language Processing (NLP) methods with the PEGASUS transformer-based abstractive summarization paradigm. The suggested methodology differs from conventional extraction methods in that it generates consistent summaries while maintaining the content's original meaning. The system's usability is enhanced by speech-to-text processing, which summarizes spoken inputs and makes it suited for audio notes, lectures, and speeches. All things considered, this study proves that generative AI works well for summarizing text and shows how current NLP models may boost efficiency, help with decision-making, and make users' lives easier. If you need a concise and accurate summary of a lot of text or audio, this approach is a great tool for students, professionals, researchers, and organizations.

FUTURE SCOPE

One way to make the Smart Text Summarization utilizing Gen AI PEGASUS Model project even better is to include more input options. Eventually, the system will be able to accept URLs as inputs, so users will be able to automatically summarize text from websites, articles, blogs, and research portals without having to manually copy and paste it. With this addition, the software would be more easier to use and better able to collect data from the internet in real time. The incorporation of multilingual summarizing support is another significant enhancement. The ability to analyze and summarize materials in more than one language may be achieved by using language detection algorithms with multilingual transformer models. Students, scholars, and professionals from all around the world

would be able to use the app more effectively with this update as it would be available in their native languages. Incorporating audio-based output elements into the project might further improve it. Using text-to-speech technology, the system can not only translate voice input into text, but it can also produce speech-based summaries. Users who prefer audio material or are visually challenged would find this feature particularly useful, since it would enable them to listen to summaries and save them as audio files. The last step in improving the system's scalability, performance, and availability is to put it on a cloud platform like AWS, Azure, or Google Cloud. The program could effectively manage massive use if it were deployed to the cloud. Making the system available as an app for iOS and Android would allow users to access it from anywhere, expanding the solution's reach even further.

REFERENCES

1. A. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," *arXiv/ACL*, 2021.
2. M. Zaheer *et al.*, "BigBird: Transformers for longer sequences," *NeurIPS*, 2021.
3. T. Scialom *et al.*, "Fine-tuning large language models for abstractive summarization," *EMNLP*, 2021.
4. J. Tay *et al.*, "Efficient transformers: A survey," *ACM Comput. Surveys*, 2022.
5. Z. Liu *et al.*, "Pre-trained models for natural language processing: A survey," *IEEE Trans. Knowl. Data Eng.*, 2022.
6. H. Zhang *et al.*, "Neural abstractive summarization with improved attention mechanisms," *IEEE Access*, vol. 10, 2022.
7. S. Gupta and A. Jain, "Deep learning approaches for text summarization: A review," *IEEE Access*, 2023.
8. Y. Chen *et al.*, "Hybrid transformer models for abstractive text summarization," *IEEE Access*, 2023.
9. R. Kumar *et al.*, "AI-based document summarization using deep learning," in *Proc. IEEE Int. Conf. AI & Data Sci.*, 2023.
10. L. Wang *et al.*, "Multi-modal summarization using transformer networks," *IEEE Trans. Multimedia*, 2024.
11. A. Das *et al.*, "Improving abstractive summarization using reinforcement learning," *IEEE Access*, 2024.
12. P. Singh and K. Verma, "Real-time text summarization using NLP and deep learning," in *Proc. IEEE Int. Conf. Smart Computing*, 2024.
13. N. Verma *et al.*, "Large language models for text summarization: Challenges and opportunities," *IEEE Access*, 2025.
14. V. Sharma and R. Gupta, "Explainable AI for abstractive summarization systems," *IEEE Access*, 2025.