International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 10 | Oct 2026 www.irjet.net p-ISSN: 2395-0072

"Summarization of Research Papers"

Prof. V. A Gahire[1], Pradnya Shirsath[2], Nandini Sure[3], Vaishali Thorat[4]

[1]Assistant Professor, Dept.of Computer Engineering, ICEEM, Maharashtra, India
[234]BE Student, Dept.of Computer Engineering, ICEEM, Maharashtra, India

---

Abstract – *The fast increase in information has made it harder for people to read and understand long documents quickly. This paper introduces SummarizeAI a web-based application that is suggested to make it easier to understand documents using intelligence and natural language processing. The system is meant to create summaries from long text by using transformer-based models like BART and PEGASUS. Along with summarization the application has keyword extraction to find terms, numeric data detection to help show values visually and a question-answering tool that helps find needed information in the original text. The system brings all these features together in one place to make reviewing documents and finding key points easier. The application is meant for use in study, research report analysis and general information processing. Its success should be tested by how good the summarizer how relevant the information is how fast it works and what users think. SummarizeAI gives a starting point for a tool to understand documents. Future plans include adding retrieval-augmented generation, support, for languages and better numerical analysis.*

Keywords: Text Summarization, Natural Language Processing, BART, PEGASUS, Keyword Extraction, Data Visualization, Question Answering, Artificial Intelligence.

INTRODUCTION

The rapid growth of information has produced a large volume of text-based content, including research papers, academic reports, news articles and technical documents. Reading and analyzing these documents manually can be time-consuming especially when users need to spot information quickly. This creates a need for systems that can cut long text into concise meaningful summaries while keeping the key ideas of the original content.

Text summarization is an application of Natural Language Processing (NLP) that aims to produce a shorter version of a document without losing its essential meaning. Traditional extractive methods simply select sentences from the source text whereas abstractive methods try to generate new sentences that express the main ideas. Transformer-based models, such as BART and PEGASUS provide ways to produce informative summaries.

It is often required to understand what the content of a document says. This can include summarising, identifying key terms, extracting numerical data and answering questions. This can become inconvenient if you need to use tools to do these. Having one application with all these functions can make it easier for the user to explore the information.

SummarizeAI is a web application that can help understand text within documents. It does this using AI to summaries text, key term extraction, numerical data extraction and visualization, and question answering. The system will be developed with Python on the backend. It will support two summarization models: BART and PEGASUS. If numerical data has been identified within the text it can be shown in charts. The question answering function helps the user find the information they want.

The primary goal of SummarizeAI is to create a space where text can be summarized and explored. This can be assessed by looking at how well the summaries are, how relevant the information is, how long it takes to process and user feedback. Overall, SummarizeAI explores how we can integrate different functions to help understand documents. There is scope for improvement in the future such as retrieval augmented generation, multilingual and more accurate numerical analysis.

1. PROPOSED SYSTEM

1.1 System Overview

The proposed SummarizeAI has four components (as shown in fig. 1). (a) Input unit: This will take text from research papers, articles, reports etc. (b) Text processing and summarization unit: Built in python, it uses transformer models like BART and PEGASUS. (c) Information analysis unit: Using the text it will extract keywords, extract numbers, create visuals if appropriate and answer questions. (d) The user interface will give a summary, summary stats, keywords, number visuals and answers to questions asked by the user. It provides an integrated environment for summarizing and exploring text using a web application.

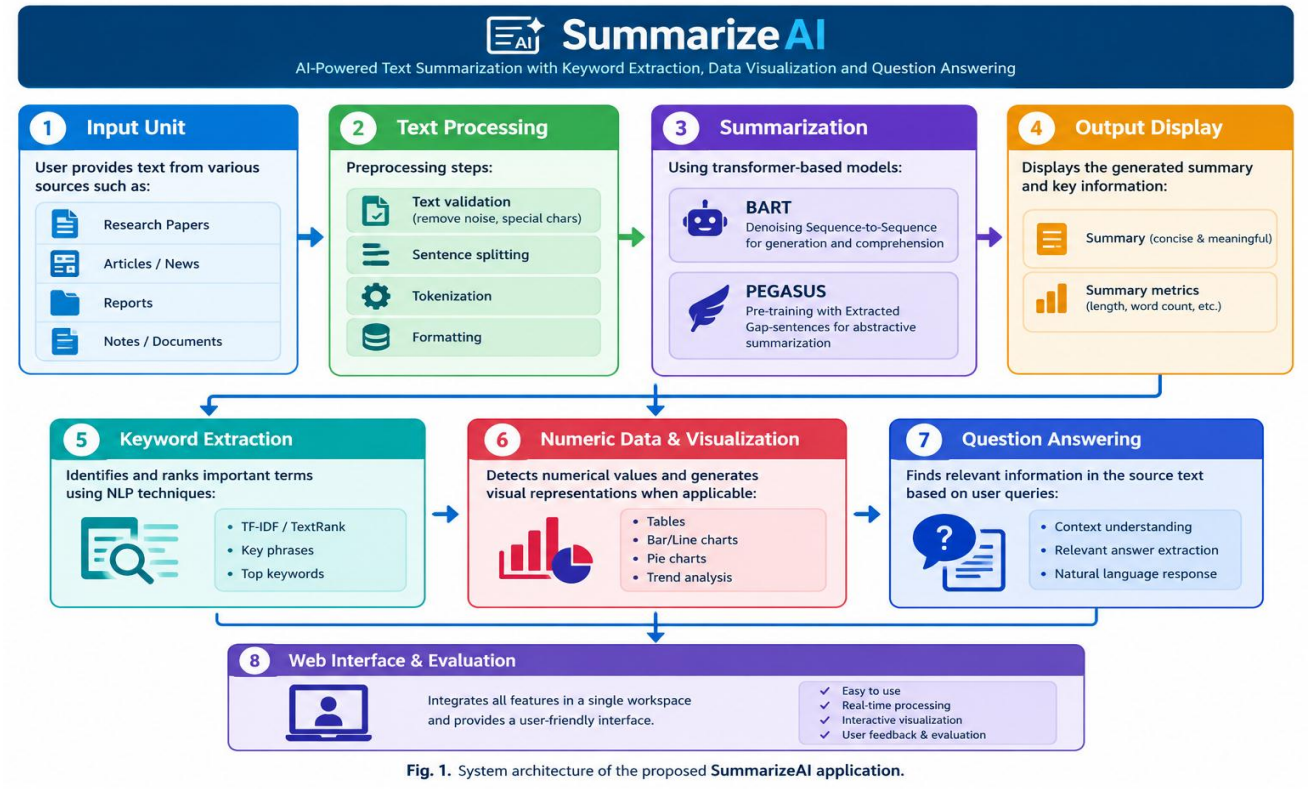


Fig. 1. System architecture of the proposed **SummarizeAI application.**

The proposed SummarizeAI system operates through the following steps:

Step 1: Text Input:

The user enters / pastes their text into the web application. This text may be taken research papers, articles, reports or any other document

Step 2: Text Preprocessing:

This checks what text has been entered ,cleans the text by removing unnecessary characters and splits the text into sentences

Step 3: Text Summarization:

This then takes the processed text and passes it to a summarization model, (such as BART / PEGASUS),to create a summary of the text

Step 4: Display Summary:

The summary is then displayed within the web application. Information on this page includes: original word count, summary word count and the percentage reduction

Step 5: Extract Keyword:

The system will extract key words and phrases from the source text.

Step 6: Identify Numeric Data:

The system will identify any numerical data. If this can be presented meaningfully, it will display it in a chart.

Step 7: Question Answering:

The user will enter a question related to the document. The question answering module will search through the text and provide an answer.

Step 8: Output:

The summary, keywords, numerical data and answer to the question are displayed within this web application. The summary can be copied /downloaded.

1.2 Software Components And Specifications

The main components of the SummarizeAI application are listed below in Table I. The application leverages NLP, Transformer models for summarization, keyword extraction, numerical data visualizations and question answering capabilities via a web based interface.

Table-I: Components and specifications

Component	Technology Specification
Programming Language	Python for backend processing and NLP

	integration
Web Framework	Flask for handling web requests and connecting the user interface to backend functions
Summarization Model	BART for generating concise, context-aware summaries
Alternative Summarization Model	PEGASUS for abstractive text summarization
Text Preprocessing	Text validation, sentence segmentation, and basic text cleaning
Keyword Extraction	NLP-basedkeyword identification using methods such as TextRank or TF-IDF
Numerical Data Detection	Regular expressions and numeric parsing to identify values in the input text
Data Visualization	Matplotlib or Plotly for displaying suitable numerical data through charts
Question Answering	Source-text sentence matching in the basic prototype; a retrieval-augmented generation model may be added in future versions
Frontend Interface	HTML, CSS, and JavaScript for text input, summary display, charts, and question answering

1.3 Flowchart of the AI-Based Text Summarization

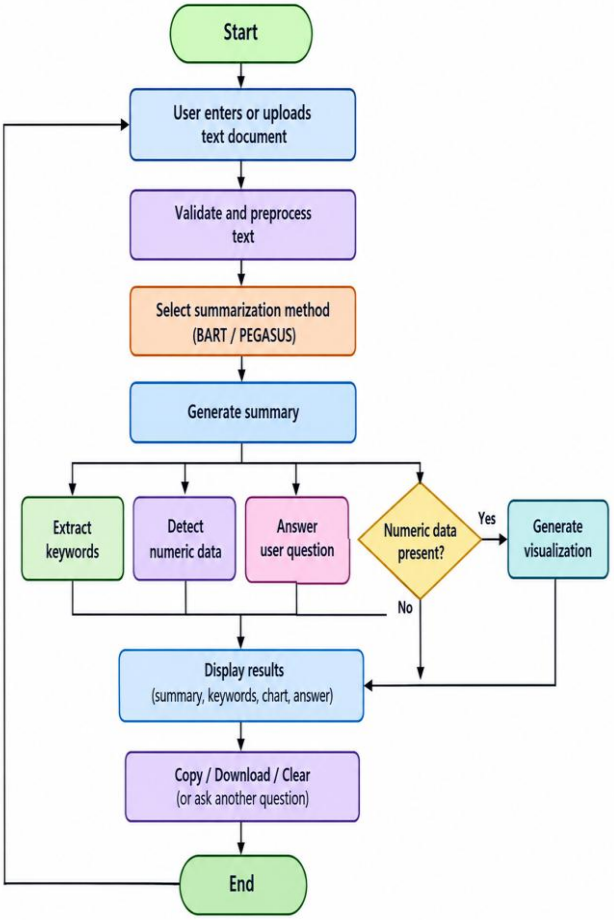


Fig. 4: Flowchart of the AI-Based Text Summarization and Document Understanding Process

Workflow covered:

1. Enter / upload text
2. The text is verified and pre-processed
3. Choose between BART and PEGASUS for summarising the text.
4. Summary is generated.
5. Keywords are extracted, numerical information identified and question-answering functionality is provided.
6. If enough numerical information is present then a visual will be created.
7. Results are shown. You can copy/download your summary, clear the results or ask another question.

Table-II: Control logic of the prototype

Condition	Action
Empty input	Request text input.
Insufficient text	Ask the user to provide more text.
Valid text submitted	Generate a demo summary.
BART/PEGASUS selected	Select the model option; backend integration is required.
Numeric values detected	Display a bar chart if enabled.
Insufficient numeric data	Display a warning message.
User asks a question	Find relevant sentences using keyword matching.
Copy selected	Copy the summary if supported.
Download selected	Download the summary as a text file.
Clear selected	Reset the input and results.

2. RESULTS AND DISCUSSION:

The SummarizeAI application provides an easy to use interface that allows the user to easily summarize large texts and explore the key information. The prototype shows the overall process of entering text, creating a basic summary, detecting numbers and visualizing them. A question and answer feature is also provided.

The application will show the original number of words, the number of words in the summary and the percentage decrease so the user knows how much the text has been reduced. Numbers are detected within the original text and visualized using a basic bar chart. Questions can be asked relating to the document and answers will be given based on matching sentences. The summary can also be copied or downloaded.

However, due to this being a prototype version, we have used demo models for the summarization functions and the question answering function currently uses keywords rather than a full AI agent. This means that whilst this project demonstrates the application and how it will work, the results do not reflect the performance of the advanced transformer models.

Overall though this has provided a starting point for building an application that can summarize text, explore numerical data and answer questions based on documents. To improve this further we could look at comparing the performance of BART and PEGASUS using metrics such as ROUGE. We could look at measuring how long it takes to process the data as well as assessing how accurate and useful the summaries are. We could also conduct user testing to see if the application makes documents easier to read.

Table-III: Traditional irrigation vs. proposed system

Aspect	Traditional method	Proposed system
Summarization	Manual	AI-Based (BART/PEGASUS)
Time	More Efforts	Aims to Save time
Keywords	Manual	Planned Automatic extraction
Data Analysis	Manual	Basic Charts
Question Answering	Manual Search	Keyword Matching

2.1 LIMITATIONS:

1. Summary accuracy depends on the quality of the input text.
2. BART and PEGASUS require backend integration.
3. Numeric charts may not represent meaningful relationships between values.
4. Question answering currently relies on keyword matching.
5. Large documents may require more processing time and memory.

2.2 FUTURE SCOPE:

The system can be extended in several ways:

- **Advanced AI Models:** Integrate BART and PEGASUS for improved summarization.
- **Multilingual Support:** Summarize text in multiple languages.
- **Smart Question Answering:** Implement AI agents and RAG for accurate answers.
- **Advanced Visualization:** Generate meaningful charts from numerical data.
- **Cloud Integration:** Support large documents and faster processing.
- **Personalization:** Allow users to customize summary length and format.

CONCLUSION

SummarizeAI is a text summarization application designed to simplify long documents using NLP techniques. It combines summary generation, numerical data visualization, and question answering in a single interface. The prototype demonstrates the basic workflow, while integration of BART and PEGASUS can further improve summarization capabilities. Future work will focus on improving accuracy, supporting larger documents, and providing more reliable, context-aware answers.

REFERENCES

[1] M. Lewis, Y. Liu, N. Goyal, et al., "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension," Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7871–7880, 2020. DOI: 10.18653/v1/2020.acl-main.703.

[2] J. Zhang, Y. Zhao, M. Saleh, and P. Liu, "PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization," Proceedings of the 37th International Conference on Machine Learning, vol. 119, pp. 11328–11339, 2020.

[3] R. Mihalcea and P. Tarau, "TextRank: Bringing Order into Text," Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, pp. 404–411, 2004.

[4] C.-Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," Text Summarization Branches Out, pp. 74–81, 2004.

[5] A. See, P. J. Liu, and C. D. Manning, "Get To The Point: Summarization with Pointer-Generator Networks," Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pp. 1073–1083, 2017.

[6] A. R. Fabbri, W. Kryściński, B. McCann, C. Xiong, R. Socher, and D. Radev, "SummEval: Re-evaluating Summarization Evaluation," Transactions of the Association for Computational Linguistics, vol. 9, pp. 391–409, 2021. DOI: 10.1162/tacl_a_00373.

[7] A. Vaswani et al., "Attention Is All You Need," Advances in Neural Information Processing Systems, vol. 30, 2017.

[8] T. Zhang et al., "BERTScore: Evaluating Text Generation with BERT," International Conference on Learning Representations, 2020.

[9] J. Maynez, S. Narayan, B. Bohnet, and R. McDonald, "On Faithfulness and Factuality in Abstractive Summarization," Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 1906–1919, 2020.

BIOGRAPHIES

PROF. S. CHECHARE
Pursuing (CSE) ICEEM INTERNATIONAL CENTER OF EXCELLENCE IN ENGGINERING & MANAGEMENT(ENGINEERING)

PROF. V. A. GAHIRE
Pursuing (CSE) ICEEM INTERNATIONAL CENTER OF EXCELLENCE IN ENGGINERING & MANAGEMENT(ENGINEERING)

MS. PRADNYA SHIRSATH
Pursuing (CSE) ICEEM INTERNATIONAL CENTER OF EXCELLENCE IN ENGGINERING & MANAGEMENT(ENGINEERING)

MS. NANDINI SURE
Pursuing (CSE) ICEEM INTERNATIONAL CENTER OF EXCELLENCE IN ENGGINERING & MANAGEMENT(ENGINEERING)

MS. VAISHALI THORAT
Pursuing (CSE) ICEEM INTERNATIONAL CENTER OF EXCELLENCE IN ENGGINERING & MANAGEMENT(ENGINEERING)