

# AI Text Summarizer

---

## Abstract

In today's digital world, people are overwhelmed with large amounts of text data, ranging from news articles to research papers. Reading long passages consumes time and effort. This project, AI Text Summarizer, is designed to solve this problem by automatically generating concise summaries using advanced Natural Language Processing (NLP) techniques. The system is built using a Flask backend for text processing and a Streamlit frontend for user interaction. The summarizer leverages the BART transformer model to produce accurate and readable summaries. The application also provides word count, readability score, and maintains a history of previous summaries.

## Introduction

Text summarization is a crucial task in NLP that aims to shorten long documents while retaining essential information. Unlike manual summarization, which is time-consuming, AI-driven approaches enable automatic summarization within seconds. The primary goal of this project is to design a web-based system that:

- Takes user input text.
- Generates a meaningful summary.
- Evaluates readability and word count.
- Provides an intuitive interface for easy use.

## Objectives

- To implement an AI-based summarization system using the facebook/bart-large-cnn model.
- To design a user-friendly web interface using Streamlit.
- To create a Flask API backend for handling summarization requests.
- To calculate word count and readability score for better text evaluation.
- To maintain a history of past summaries for reference.

## System Architecture

Frontend (Streamlit): Collects user input, displays summary, word count, readability score, and history, with interactive UI.

Backend (Flask + Hugging Face Transformers): Handles POST requests from the frontend, processes text using the BART model, and returns the summary in JSON format along with analysis metrics.

## Modules

1. User Input Module – Takes raw text from the user.
2. Summarization Module – Uses the BART model to generate summaries.
3. Analysis Module – Calculates word count and readability score.
4. History Module – Stores past summaries in session state.
5. Frontend UI Module – Displays results with custom design and styles.

## Technologies Used

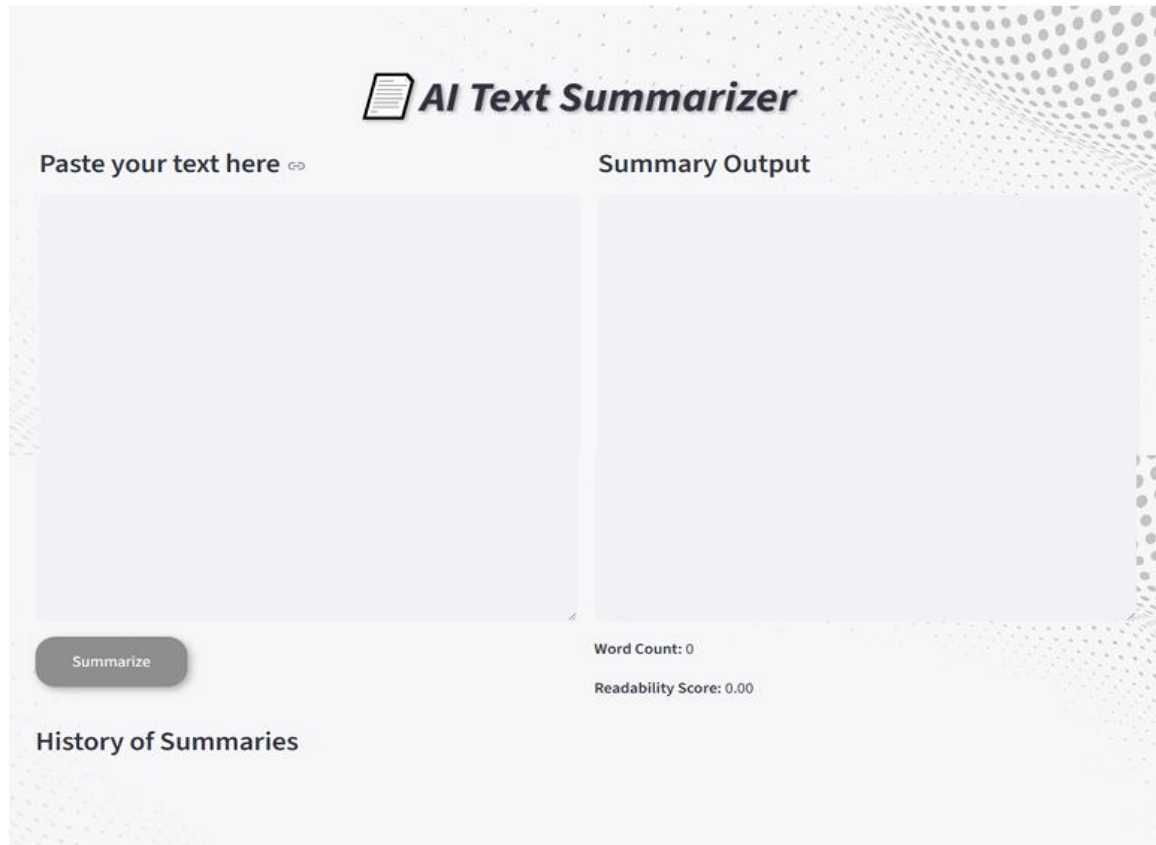
- Programming Language: Python
- Frontend Framework: Streamlit
- Backend Framework: Flask
- Libraries: Torch, Transformers, Textstat, Requests
- Model: facebook/bart-large-cnn (Hugging Face)

## Results

The system successfully:

- Generates concise summaries from long text inputs.
- Provides readability scores based on the Flesch Reading Ease metric.
- Tracks summary history with expandable details.
- Offers a responsive and visually appealing interface.

## Application Output Screenshot



## Conclusion

The AI Text Summarizer is an efficient web application that simplifies the process of understanding large texts by generating accurate and readable summaries. With its combination of Flask backend and Streamlit frontend, it demonstrates the practical integration of NLP models in real-world applications.

## Future Enhancements

- Support for multiple summarization models (e.g., T5, Pegasus).
- Option to upload and summarize documents (PDF, DOCX).
- Multilingual summarization support.
- Integration of keyword extraction for topic highlighting.
- Cloud deployment for broader accessibility.

## References

- Hugging Face Transformers: <https://huggingface.co/transformers/>
- Streamlit Documentation: <https://docs.streamlit.io/>
- Flask Documentation: <https://flask.palletsprojects.com/>
- Textstat Library: <https://pypi.org/project/textstat/>