AI Writer with Text Summarization

Introduction

With the rapid increase in digital content, reading through lengthy articles, reports, or documents can be time-consuming and inefficient. This project focuses on developing an AI-powered web application that can generate concise and meaningful summaries of large articles. By leveraging advanced Natural Language Processing (NLP) models, the AI Writer helps users quickly grasp the core information of any text, enhancing productivity and saving time.

Abstract

This project uses state-of-the-art transformer models from Hugging Face to build a summarization tool that condenses long-form text into digestible summaries. The BART summarization model is utilized due to its accuracy and effectiveness. The application provides a clean, user-friendly web interface where users can input lengthy text and receive a summarized output along with statistics such as word count and readability score. The system also supports saving the history of generated summaries for future reference. Built with Python and Flask, the tool is accessible via any modern web browser.

Tools Used

- Python: Core programming language for backend logic and model integration.
- Hugging Face Transformers: Pre-trained BART summarization model and tokenizers.
- Flask: Web framework used for developing the web application.
- textstat: Python library to compute readability scores.
- Bootstrap, HTML, CSS: Technologies for frontend design and responsive user interface.
- PyTorch or TensorFlow: Backend deep learning frameworks required to run the transformer models.

Steps Involved in Building the Project

1. Model Selection and Environment Setup
   - Selected the `facebook/bart-large-cnn` model from Hugging Face for text summarization.
   - Installed necessary libraries and ensured either PyTorch or TensorFlow was available.
2. Frontend Development
   - Designed a responsive web interface with Bootstrap and Flask templates.
   - Created pages for text input, displaying summaries, and viewing history.

3. Summarization Logic
   - Implemented text chunking to handle long inputs.
   - Integrated Hugging Face pipeline to generate summaries.
4. Additional Functionalities
   - Added word count display for original and summarized texts.
   - - Enabled saving of summaries as individual text files in a dedicated folder.
5. History Management
   - Developed a history page that lists all saved summaries for easy access.
6. Testing and Deployment
   - Tested the application locally and refined the user experience.
   - Structured the project for easy deployment on servers.

Conclusion

The AI Writer project effectively combines cutting-edge NLP models with web development technologies to produce a practical and user-friendly summarization tool. It streamlines information consumption by automatically condensing lengthy texts while providing useful statistics and maintaining a history of summaries. This project serves as a solid foundation for further enhancements such as multi-language support, document uploads, and personalized user accounts.