Chatbot on Extractive Summarization Tool

Intern Name: Kaushal Sahu

Internship Duration: 2/7/2025 – 5/5/2025

Organization: NullClass

1. Introduction

The goal of this task was to develop a system that could generate concise summaries of long documents using extractive summarization techniques. Unlike abstractive methods, this approach selects the most important sentences from the original content without modifying the actual wording, ensuring a fast and reliable summarization method.

2. Background

In many real-world scenarios—news aggregation, research literature reviews, or business reports—concise summaries help users quickly understand long documents. Extractive summarization is a well-established technique that uses statistical or machine learning approaches to score and select key sentences, offering an efficient and interpretable summarization method.

3. Learning Objectives

- Understand the core concept of extractive summarization using natural language processing.
- Implement sentence tokenization, text preprocessing, and scoring using TF-IDF.
- Rank sentences based on importance and generate a summary from the top-ranked ones.

4. Activities and Tasks

- Used NLTK for tokenization and stopword removal (punkt, stopwords).
- Preprocessed text by filtering out stopwords and non-alphanumeric tokens.
- Applied TF-IDF Vectorization (sklearn) to calculate sentence importance.
- Scored each sentence by summing TF-IDF values.
- Extracted the top N sentences (user-defined summary length) to form the summary.

5. Skills and Competencies

- Technical:
  - NLP pipeline: tokenization, vectorization, and filtering
  - TF-IDF implementation using TfidfVectorizer
  - Python programming and NumPy for scoring and ranking
- Soft Skills:
  - Identifying relevant sentence structures
  - Measuring sentence importance using unsupervised techniques
  - Text summarization design thinking

6. Feedback and Evidence

- The summarizer produced accurate, concise summaries for sample paragraphs.
- Code was tested using a sample paragraph and showed clear extraction of the most informative sentences.
- Reviewers highlighted the clarity and readability of code and correct TF-IDF logic for sentence scoring.
- The sample screenshots of the model performance is uploaded in the folder name “sample”. The screenshot is of different scripts and there summarization.

7. Challenges and Solutions

1. NLTK stopwords not downloaded on first run

SOL- The stopwords have not downloaded at the first run so I included the code “nltk.download()” command at the top middle of the code.

2. To follow the training video as well as making your own code

SOL- The training video was 40% same, but the rest 60% I had to find it myself and implement in the code. Thanks to the teacher who made the foundation of each code snippet and there functionality.

8. Outcomes and Impact

- Built a lightweight, fast, and efficient summarization tool.
- The tool can be easily integrated into educational platforms, news digest systems, or productivity apps.
- Provided a solid foundation for more advanced models like abstractive summarization using transformers.

9. Conclusion

This task successfully demonstrated the ability to implement a simple yet effective extractive summarization system using standard NLP techniques. It emphasized the role of preprocessing and TF-IDF in identifying key information in a text. The project added an essential skillset in text processing, preparing the groundwork for more advanced summarization applications in the future.