Climate Change Chatbot Report

Introduction

This report outlines the development process of the Climate Change Chatbot, detailing the approach taken, challenges encountered, and solutions implemented to create an effective tool for understanding climate change.

Approach

1. Data Preparation

The chatbot utilizes the **IPCC AR6 WGII Technical Summary** as its primary data source. The process involved several key steps:

- **Data Extraction**: Used the `pypdf` library to extract text from the PDF document, ensuring that all relevant information was captured.
- **Text Cleaning**: Filtered out unnecessary sections, such as headers and footers, and removed redundant strings to prepare the text for analysis.
- **Text Splitting**: Employed text splitters (e.g., `RecursiveCharacterTextSplitter`) to divide the cleaned text into manageable chunks suitable for processing.

2. Information Retrieval

To enhance the chatbot's responsiveness, a vector database (ChromaDB) was utilized:

- **Document Storage**: The cleaned and split text chunks were stored in a ChromaDB collection, allowing for efficient retrieval based on user queries.
- **RAG Implementation**: Implemented a Retrieval-Augmented Generation (RAG) approach to retrieve relevant documents for each user query before generating a response.

3. Response Generation

- **OpenAI Integration**: Integrated the OpenAI GPT-3.5-turbo model to generate context-aware responses. The model was fed both the user’s query and the relevant retrieved documents to produce accurate answers.

4. User Interface

Developed a user-friendly interface using **Streamlit** to facilitate interaction with the chatbot, allowing users to input questions easily and receive responses in real-time.

Challenges Faced

1. Data Quality and Format

Challenge: The initial extraction from the PDF led to inconsistent formatting and unwanted characters.

Solution: Implemented robust text cleaning procedures using regular expressions to standardize the text and remove extraneous information.

2. Efficient Document Retrieval

Challenge: Ensuring relevant documents were retrieved for a wide range of potential user queries.

Solution: Employed a vector database (ChromaDB) that allows for flexible querying and fine-tuned the chunking strategy to optimize the relevance of retrieved documents.

3. API Key Management

Challenge: Ensuring the OpenAI API key was securely managed and accessible during development.

Solution: Used environment variables to store the API key, preventing it from being hard-coded into the application.

4. Model Limitations

Challenge: The OpenAI model may occasionally produce responses that lack context or contain inaccuracies.

Solution: Structured the prompt to include both user queries and relevant documents, ensuring that the model has the necessary context to generate informed responses.

Overcoming Challenges

Through iterative testing and refinement, the development team addressed the challenges faced during the project. Key strategies included:

- Conducting thorough testing on various user queries to identify weaknesses in response generation and retrieval accuracy.
- Continuously refining text preprocessing and chunking strategies based on feedback from initial runs.
- Engaging in regular code reviews and collaborative discussions to share insights and improvements.

Conclusion

The Climate Change Chatbot successfully leverages the IPCC AR6 WGII Technical Summary to provide users with reliable information on climate change. Despite facing several challenges, the development team implemented effective solutions, resulting in a robust and user-friendly tool for public engagement on this critical topic.