

Chatbot Project Report

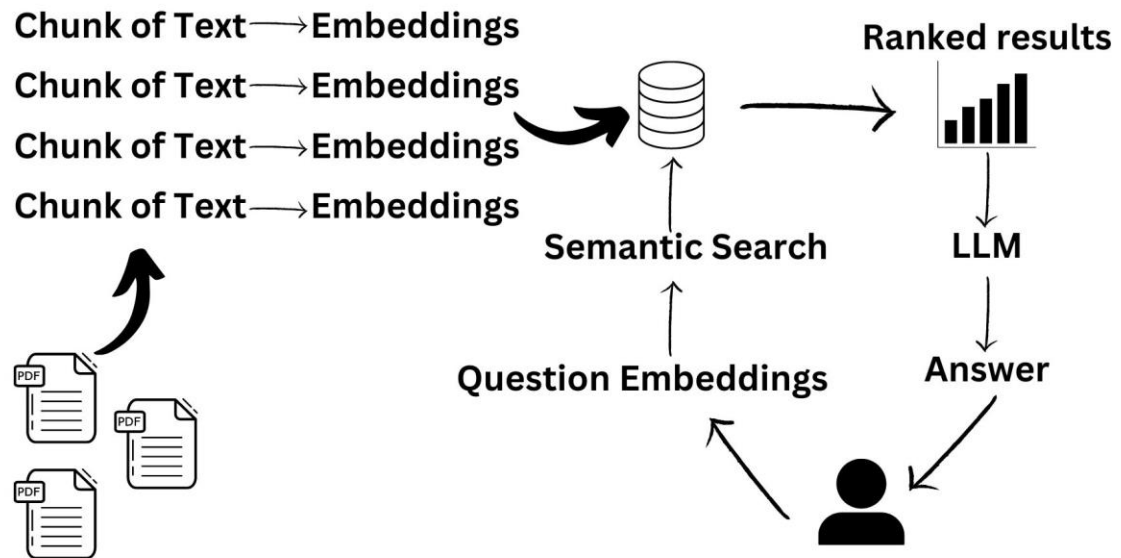
Abstract

In this project, we developed a chatbot that can understand and answer questions based on a specified document. The chatbot interacts with users through a web interface and provides responses in real-time. We used natural language processing techniques to build the chatbot, making it capable of understanding user queries and extracting relevant information from the document.

Process Overview

- **Getting Started:** We started by gathering requirements and planning the project. We defined what the chatbot should do, who would use it, and how it would work.
- **Research and Setup:** We researched different tools and libraries for natural language processing and chose the ones that best fit our needs. Then, we set up the development environment and organized our project structure.
- **Coding:** We wrote the code for the chatbot using Python. The backend handled tasks like processing documents and answering questions, while the frontend provided the user interface for interacting with the chatbot.
- **Testing:** We tested the chatbot to make sure it worked correctly and gave accurate responses. We fixed any bugs or issues we found during testing.
- **Deployment:** Once the chatbot passed testing, we deployed it to a server so users could access it. We made sure everything was set up properly and that the chatbot was secure.
- **Documentation:** We created documentation to help users understand how to use the chatbot and how it was built. This included README files, code comments, and user guides.
- **Support and Maintenance:** We continue to provide support for the chatbot, addressing any user feedback or issues that arise. We also make updates and improvements to keep the chatbot running smoothly.

Working



This diagram illustrates the process of semantic search, a method used to find answers to questions. It begins by extracting segments of text from PDF documents and converting them into embeddings, which are compact representations of the text.

When a question is posed, it's also transformed into an embedding known as "Question Embeddings." These embeddings are then used to search through the database of text embeddings for the most relevant matches.

The results are ranked based on their similarity to the question and passed on to a Language Model (LLM). The LLM analyzes the relevant text segments and generates an answer to the original question.

The key steps depicted include segmenting text into embeddings, storing them in a database, converting the question into an embedding, conducting semantic search, ranking the results, and using a language model to produce an answer.

Demo

