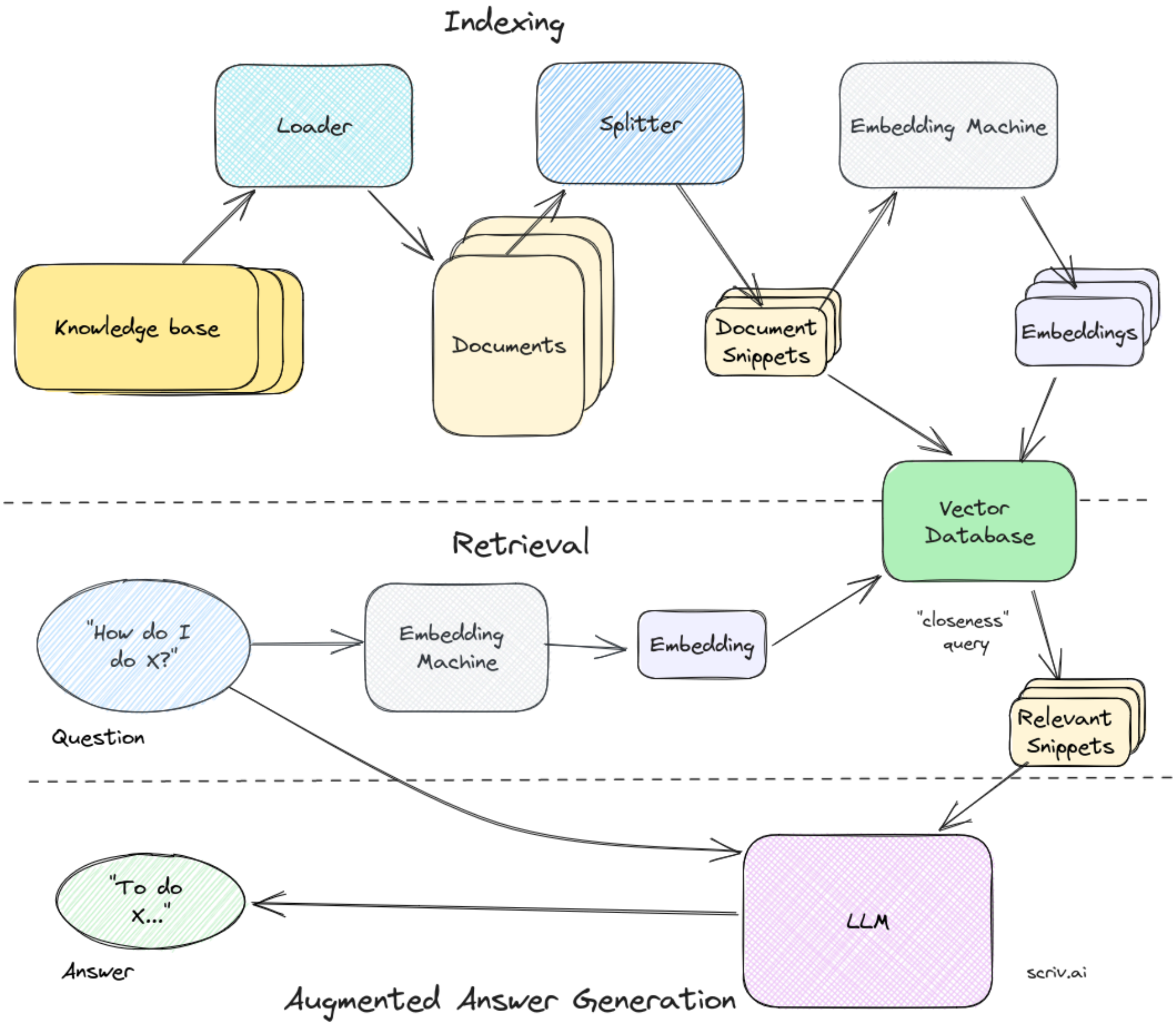
Building a Domain-Specific Chatbot Using Retrieval-Augmented Generation

Project Overview

The objective of this project was to develop a domain-specific chatbot which leverages the Retrieval-Augmented Generation (RAG) methodology to provide educational assistance on deep learning topics. I have titled my chatbot as “DeepTeacer”. This chatbot aims to serve students, educators, and enthusiasts by offering precise, contextually relevant information derived from authoritative texts within the domain.

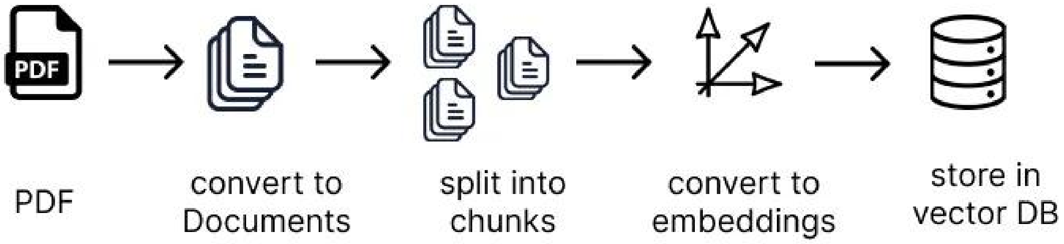


System Architecture

The chatbot integrates several advanced components:

- ➔ Language Model: Utilizes Microsoft's Phi3, a small language model with 3.8 billion parameters, providing a robust foundation for generating human-like text responses.
- ➔ Embedding Model: Employs HuggingFace's all-MiniLM-L6-v2 for converting text into dense vector embeddings, which facilitate the retrieval of relevant document segments.
- ➔ Document Sources: Leverages content from two primary texts:
  "Generative Deep Learning" by Ben Foster (2023)
  "Deep Learning with Python"

Methodology



Document Preparation:

1. Text Extraction: Text is extracted from the provided PDF documents using PyMuPDFLoader, ensuring that all content is accessible for processing.
2. Text Splitting: A RecursiveCharacterTextSplitter segments the extracted texts into manageable chunks (700 characters with an overlap of 40), optimizing them for embedding and retrieval.

Embedding Generation:

- The extracted text chunks are transformed into numerical embeddings using the all-MiniLM-L6-v2 model. These embeddings are crucial for the subsequent similarity search in the vector store.

Vector Store Creation:

- A FAISS vector store is employed to maintain and manage these embeddings efficiently. This setup allows for rapid retrieval of text chunks based on similarity to the user queries.

Retrieval-Augmented Generation:

1. History-Aware Retriever: Integrates past user interactions to enhance the contextuality of responses.
2. Retrieval Chain: Combines the history-aware retriever with a language model to generate responses that are not only accurate but also contextually enriched.

User Interface:

- Developed using Streamlit, the interface supports interactive engagement through text inputs for both direct questions and similarity search queries. Users can receive responses, explore related topics, or follow predefined prompts.
- Users can get answers to the built-in prompts.
- Users can also perform similarity searches against the pdf documents.

Results

The "Deep Learning Teacher" chatbot effectively demonstrates the capability of using RAG for educational purposes. The chatbot provides accurate, detailed, and contextually appropriate answers to queries related to deep learning. The utilization of specialized texts ensures that the responses are not only relevant but also of high educational value.

Conclusion

This project showcases the practical application of advanced NLP techniques to create a specialized educational tool. The integration of a state-of-the-art language model with retrieval-augmented capabilities offers a significant advancement in how educational content can be personalized and delivered. Future enhancements could include the expansion of document sources, the refinement of the user interface, and the incorporation of additional interactive elements to further enrich the user experience.

Future Work

- Expansion of Knowledge Base: Including more texts to cover a wider range of topics within deep learning.
- Enhanced Interaction: Introducing more interactive elements such as quizzes or topic summaries.
- Performance Optimization: Continuous optimization of retrieval processes and response generation speed.

This technical report outlines the creation and implementation of the "Deep Learning Teacher," illustrating the successful application of RAG to an educational chatbot. The methodology and results highlight the system's effectiveness and potential for broader application in educational technology.