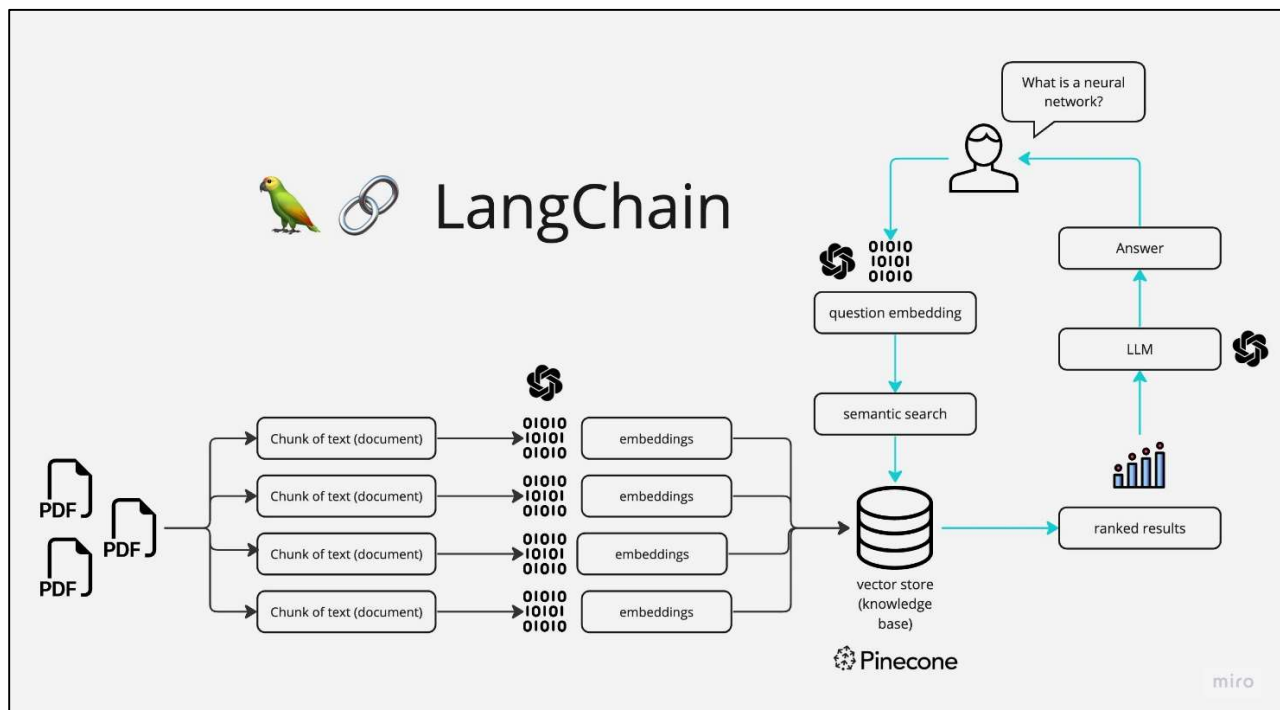


# Documentation

## Chatbot

### # Approach Explanation with Results:

- **Approach:** The approach involves processing PDF documents to extract text data, splitting the text into chunks, generating embeddings for the text chunks using a pre-trained model, and creating a conversational retrieval chain for answering user queries based on the embeddings.
- **Results:** The code implementation allows users to upload PDF documents, input questions related to the document content, and receive answers based on the document context.



## # Model Architecture and Reasoning:

- **Model Architecture:** The model architecture involves using a pre-trained language model (LLM) (***Mistral Model***) for natural language understanding and generation, combined with a conversational retrieval chain for context-based responses. For embeddings I have used (***Instructor model***) as it was light weight for quick usage.
- **Reason for Selection:** The pre-trained LLM, such as Hugging Face's Mistral model, is chosen for its ability to understand and generate human-like responses based on textual input. The conversational retrieval chain complements the LLM by providing context-aware responses using vector embeddings of text chunks.

## # Hardware Requirements:

- For processing PDF documents, sufficient RAM may be needed to handle the text extraction and embedding generation tasks efficiently.
- It's recommended to have a machine with at least 8GB of RAM and a GPU with CUDA support for faster processing, especially if dealing with large documents or complex models.

## # Specific Installations Required:

- The code requires installation of various libraries and packages, including PyPDF2, langchain, Hugging Face's Transformers library, and Streamlit for the user interface, which is included in the requirements file, also in the .ipynb file.
  - Additionally, you are required to have Hugging Face APIs for using their models.
  - In env file, set (***HUGGINGFACEHUB\_API\_TOKEN = your token***). I have already included a token for your testing.
-