# [***MultiPDF Chat App***](https://github.com/alejandro-ao/ask-multiple-pdfs/blob/main/readme.md#multipdf-chat-app)

## [**Introduction**](https://github.com/alejandro-ao/ask-multiple-pdfs/blob/main/readme.md#introduction)

Large language models (LLM) are very large [deep learning](https://aws.amazon.com/what-is/deep-learning/) models that are pre-trained on vast amounts of data. The underlying transformer is a set of [neural networks](https://aws.amazon.com/what-is/neural-network/) that consist of an encoder and a decoder with self-attention capabilities. The encoder and decoder extract meanings from a sequence of text and understand the relationships between words and phrases in it.

Transformer LLMs are capable of unsupervised training, although a more precise explanation is that transformers perform self-learning. It is through this process that transformers learn to understand basic grammar, languages, and knowledge.

Unlike earlier recurrent neural networks (RNN) that sequentially process inputs, transformers process entire sequences in parallel. This allows the data scientists to use GPUs for training transformer-based LLMs, significantly reducing the training time.

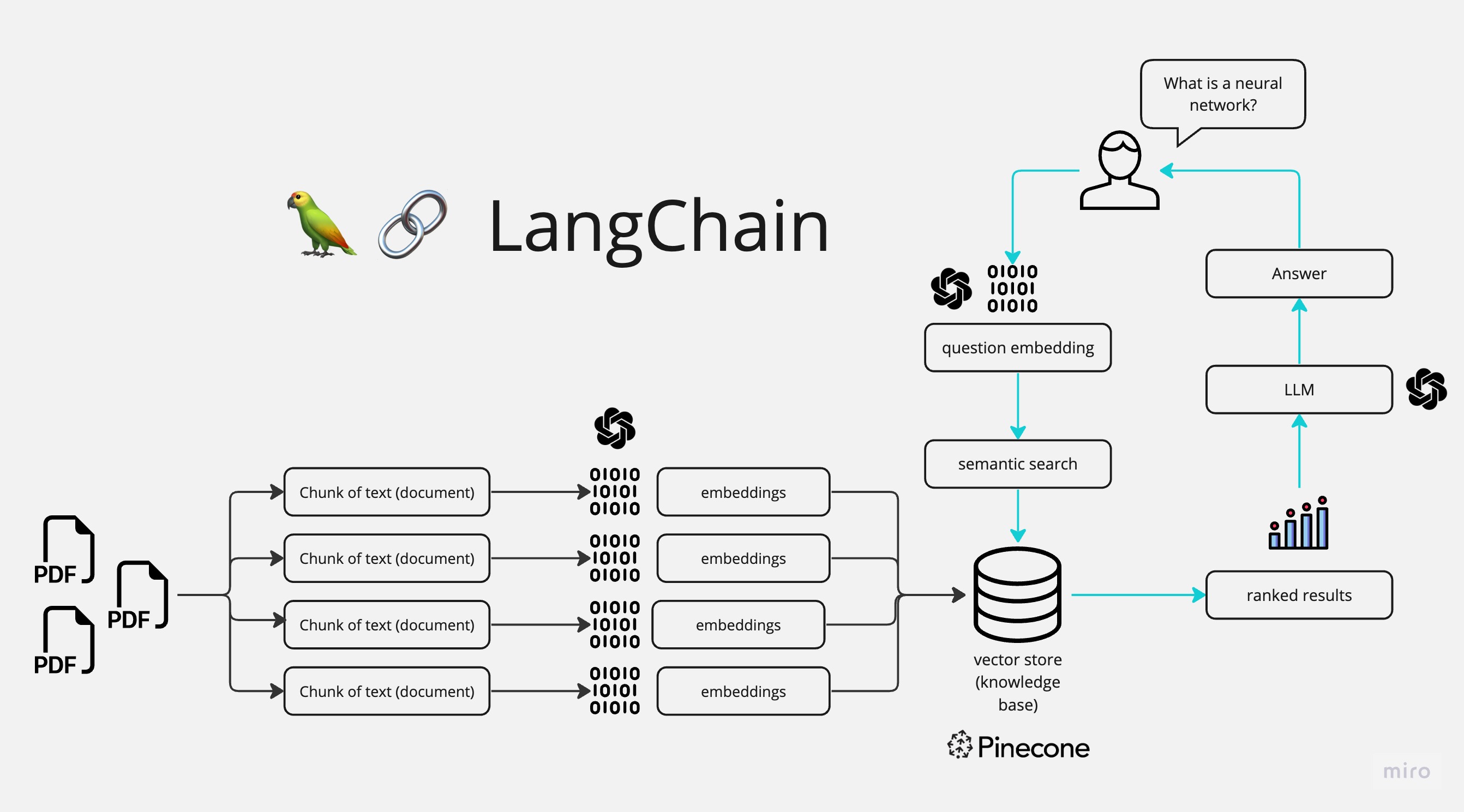
Transformer neural network architecture allows the use of very large models, often with hundreds of billions of parameters. Such large-scale models can ingest massive amounts of data, often from the internet, but also from sources such as the [Common Crawl](https://registry.opendata.aws/commoncrawl/), which comprises more than 50 billion web pages, and Wikipedia, which has approximately 57 million pages.

The MultiPDF Chat App is a Python application that allows you to chat with multiple PDF documents. You can ask questions about the PDFs using natural language, and the application will provide relevant responses based on the content of the documents. This app utilizes a language model to generate accurate answers to your queries. Please note that the app will only respond to questions related to the loaded PDFs.

## [**How It Works**](https://github.com/alejandro-ao/ask-multiple-pdfs/blob/main/readme.md#how-it-works)

A key factor in how LLMs work is the way they represent words. Earlier forms of machine learning used a numerical table to represent each word. But, this form of representation could not recognize relationships between words such as words with similar meanings. This limitation was overcome by using multi-dimensional vectors, commonly referred to as word embeddings, to represent words so that words with similar contextual meanings or other relationships are close to each other in the vector space.

Using word embeddings, transformers can pre-process text as numerical representations through the encoder and understand the context of words and phrases with similar meanings as well as other relationships between words such as parts of speech. It is then possible for LLMs to apply this knowledge of the language through the decoder to produce a unique output



The application follows these steps to provide responses to your questions:

1. PDF Loading: The app reads multiple PDF documents and extracts their text content.
2. Text Chunking: The extracted text is divided into smaller chunks that can be processed effectively.
3. Language Model: The application utilizes a language model to generate vector representations (embeddings) of the text chunks.
4. Similarity Matching: When you ask a question, the app compares it with the text chunks and identifies the most semantically similar ones.
5. Response Generation: The selected chunks are passed to the language model, which generates a response based on the relevant content of the PDFs.

## **applications of large language models?**

There are many practical applications for LLMs.

### **Copywriting**

Apart from GPT-3 and ChatGPT, Claude, Llama 2, Cohere Command, and Jurassiccan write original copy. AI21 Wordspice suggests changes to original sentences to improve style and voice.

### **Knowledge base answering**

Often referred to as knowledge-intensive natural language processing (KI-NLP), the technique refers to LLMs that can answer specific questions from information help in digital archives. An example is the ability of AI21 Studio playground to answer general knowledge questions.

### **Text classification**

Using clustering, LLMs can classify text with similar meanings or sentiments. Uses include measuring customer sentiment, determining the relationship between texts, and document search.

### **Code generation**

LLM are proficient in code generation from natural language prompts. Examples include [Amazon CodeWhisperer](https://aws.amazon.com/codewhisperer/) and Open AI's codex used in GitHub Copilot, which can code in Python, JavaScript, Ruby and several other programming languages. Other coding applications include creating SQL queries, writing shell commands and website design.

### **Text generation**

Similar to code generation, text generation can complete incomplete sentences, write product documentation or, like Alexa Create, write a short children's story.

## [**Dependencies and Installation**](https://github.com/alejandro-ao/ask-multiple-pdfs/blob/main/readme.md#dependencies-and-installation)

To install the MultiPDF Chat App, please follow these steps:

1. Clone the repository to your local machine.
2. Install the required dependencies by running the following command:
3. pip install -r requirements.txt
4. Obtain an API key from OpenAI and add it to the .env file in the project directory.

OPENAI\_API\_KEY=your\_secrit\_api\_key

## [**Usage**](https://github.com/alejandro-ao/ask-multiple-pdfs/blob/main/readme.md#usage)

To use the MultiPDF Chat App, follow these steps:

1. Ensure that you have installed the required dependencies and added the OpenAI API key to the .env file.
2. Run the main.py file using the Streamlit CLI. Execute the following command:
3. streamlit run app.py
4. The application will launch in your default web browser, displaying the user interface.
5. Load multiple PDF documents into the app by following the provided instructions.
6. Ask questions in natural language about the loaded PDFs using the chat interface.

## [**Contributing**](https://github.com/alejandro-ao/ask-multiple-pdfs/blob/main/readme.md#contributing)

This repository is intended for educational purposes and does not accept further contributions. It serves as supporting material for a YouTube tutorial that demonstrates how to build this project. Feel free to utilize and enhance the app based on your own requirements.

## [**License**](https://github.com/alejandro-ao/ask-multiple-pdfs/blob/main/readme.md#license)

The MultiPDF Chat App is released under the [MIT License](https://opensource.org/licenses/MIT).