Promtior
Technical Test

Full name: Lisandro Ariel Canteros
LinkedIn

Description

Develop and deploy a chatbot assistant that uses the RAG architecture to answer questions about the content of the Promtior website, based on the LangChain library.

Technologies used

- LangChain main component used to integrate and use LLM in Python. Provides an interface to operate with different LLM with the RAG architecture.
- LangServe an easy way to deploy LangChan applications as a REST API.
- Docker to containerize the app and make it easier to deploy anywhere.
- Azure for deploying the app as a container on a VM in the cloud.

Solution

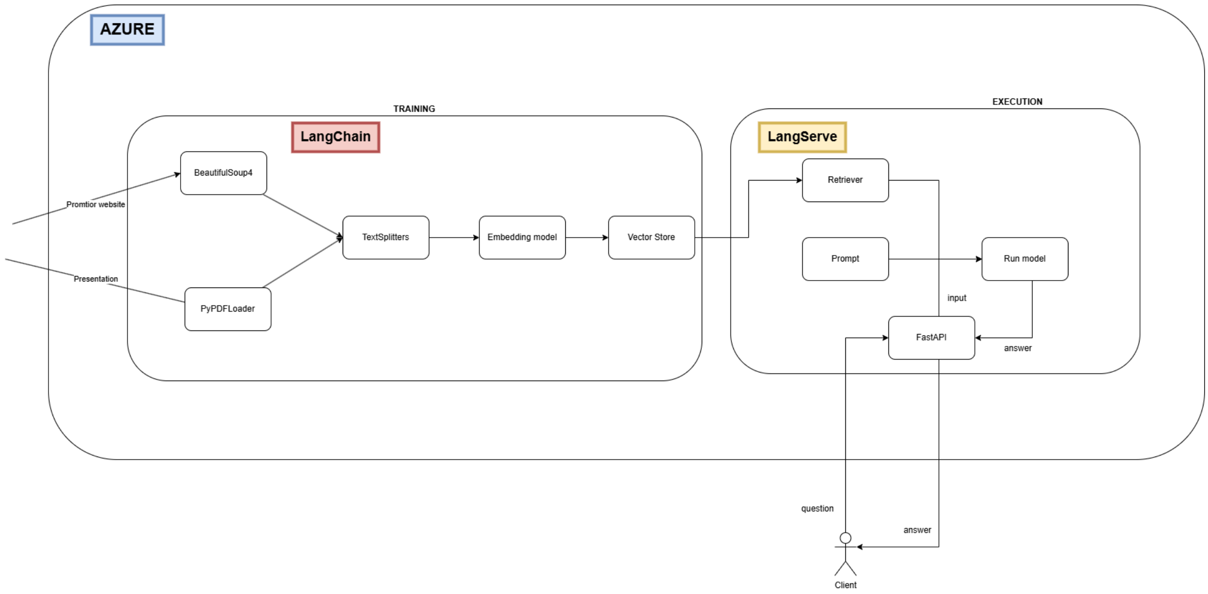
The application was developed using LangChain in Python and functions as a ChatBot powered by Ollama's LLaMA 3.2 model. The primary reason for choosing Ollama is its ability to run locally without any payment requirements; however, the downside is that the host machine needs substantial resources to run the application efficiently.

The initial stage of development involved fine-tuning the LLaMA 3.2 model to provide the required answers. This process consists of three steps: loading information, indexing it, and retrieving it. First, the Promtior website is parsed using BeautifulSoup4 to extract its contents. Additionally, the presentation's content is loaded using PyPDFLoader. In the second step, all the information is divided into chunks of a specific size, making it easier for the model to process the data without analyzing it all at once. These chunks are then utilized as a retriever to fetch relevant information based on a query. Finally, with the help of chains, the application processes user input, applies a pre-defined prompt, and sends it to the LLM, which returns an answer. All this logic is contained within a single file named chain.py.

To facilitate interaction, another file named server.py is used. With the help of the FastAPI library, the application is presented as a REST API. This file contains only one additional route, which is necessary to utilize the previously created chain.

Regarding the deployment, the application is hosted on Azure and operates as a REST API. Thanks to LangServe and FastAPI, it also provides a simple yet powerful browser interface to interact with the ChatBot. The application is deployed as a container in Azure, which is why a Dockerfile is included in the GitHub repository. It's currently running on a 4 core CPU and 16gb memory.

Components diagram


AZURE
TRAINING
LangChain
Promtior website
Presentation
BeautifulSoup4
PyPDFLoader
TextSplitters
Embedding model
Vector Store
EXECUTION
LangServe
Retriever
Prompt
Run model
input
FastAPI
answer
question
answer
Client