AI Engineer Evaluation Case

CEYDA BAŞOĞLU

ceyda_basoglu2000@hotmail.com

05532728028

Project Documentation

Overview

Welcome to the ai engineer evalution case documentation! This document provides detailed information about the ai engineer evalution case, including its functionality, usage instructions, project structure, and more.

The ai engineer evalution project is a RESTful API built using Flask framework, designed to serve various tools including vector search, basic math operations, and rapid question-answering on PDF documents.

Table of Contents

Requirements

• Python 3.12.0 or newer.

• Docker and Docker Compose

• Flask framework.

• Chroma

• Langchain

• OpenAI embeddings

• Chat models (gpt-3.5-turbo)

Project Structure

The ai engineer evalution case project consists of the following main components:

• **app.py:** This file contains the main Flask application and defines the API endpoints for various functionalities including uploading text, performing vector search, executing math operations, and rapid question-answering on PDF documents.

• **math_tool.py:** Implementation of the MathTool class with methods for performing basic math operations such as addition, subtraction, multiplication, division, exponentiation, and square root.

• **third_tool.py:** Functionality for rapid question-answering on PDF documents. Includes functions to load PDF documents, split text into chunks, create a similarity search database, and generate responses to user queries.

• **Dockerfile:** Specifies the environment and dependencies required to run the Flask application inside a Docker container.

• **docker-compose.yml:** Defines the services and configuration for building and running the Docker container.

• **tests/:** Directory containing unit tests for each component of the project.

Installation

To install and set up the ai engineer evalution project locally, follow these steps:

1. Clone the repository to your local machine:
    a. git clone <repository-url>
2. Navigate to the project directory:
    a. cd <project-directory>
3. Install the required dependencies:
    a. pip install -r requirements.txt
4. Run the Flask application:
    a. python app.py

Add and configure requirements such as Chroma Vector Stores and OpenAI embeddings.

Usage

The ai engineer evalution project provides several endpoints for performing different tasks. Here's how you can use them:

1. **Upload Text:** POST request to /upload-text endpoint to upload a text file and convert it to a vector using OpenAI embeddings.

2. **Vector Search:** POST request to /vector-search endpoint to perform vector similarity search based on user input.

3. **Math Operations:** POST request to /math endpoint to perform basic math operations (addition, subtraction, multiplication, division, exponentiation, square root) based on user input.

4. **Rapid QA on PDFs:** POST request to /rapid-qa-on-pdfs endpoint to generate rapid question-answering responses on PDF documents based on user queries

API Endpoints

The following API endpoints are available:

- POST /upload-text
- POST /vector-search
- POST /math
- POST /rapid-qa-on-pdfs

Once the API has started successfully, you can verify that it is running by visiting http://127.0.0.1:5000/.

For detailed information on request and response formats, refer to the source code documentation or API documentation.

Testing

Unit tests have been provided for each component of the project to ensure code reliability and functionality. To run the unit tests, execute the following command:

- python -m unittest discover -s tests

Docker Deployment

To deploy the API using Docker, follow these steps:

1. Make sure Docker is installed on your machine.

2. Build the Docker image using the provided Dockerfile:

- docker build -t <image-name>

3. Run the Docker container using Docker Compose:

- docker-compose up

Once the API has started successfully, you can verify that it is running by visiting http://127.0.0.1:5000/

Streamlit Integration

If you want to utilize the Streamlit functionality provided in the project, follow these steps:

1. Install Streamlit by running the following command:

   - pip install streamlit

2. Make sure you have Python 3.7 or newer installed on your system.

3. Navigate to the project directory and run the following command to start the Streamlit app:

   - streamlit run app.py

4. Once the Streamlit app is running, you can access it by visiting http://localhost:8501 in your web browser.

OpenAI Embeddings Integration

To integrate OpenAI embeddings into your project, you'll need to follow these steps:

1. Sign up for an OpenAI account and obtain an API key.

2. Install the langchain_openai library by running the following command:

   - pip install langchain_openai

3. In your project code, import the OpenAIEmbeddings class from langchain_openai and initialize it with your API key:

```
from langchain_openai import OpenAIEmbeddings

openai_api_key = "your_openai_api_key"

openai_embeddings = OpenAIEmbeddings(openai_api_key=openai_api_key)
```

You can now use the openai_embeddings object to perform embedding operations in your code.

Conclusion

The ai engineer evaluate project provides a comprehensive solution for various tool-based functionalities including vector search, math operations, and rapid question-answering on PDF documents. The documentation provided here aims to assist users in understanding, deploying, and utilizing the API effectively.