Analytics Vidhya

Environment Setup Guide

This is a useful reference guide you can leverage when working on your Agentic AI Capstone Project.

Installing Dependencies

Python Dependencies

Make sure you are using the following python libraries with these exact versions to ensure you are using a stable version as well as to ensure consistency when you build your solution

```
autogen==0.10.1
crewai==0.203.0
crewai-tools==0.76.0
langchain==0.3.27
langchain-chroma==0.2.5
langchain-community==0.3.27
langchain-core==0.3.80
langchain-openai==0.3.30
langgraph==0.6.5
pydantic==2.11.10
pandas==2.3.3
```

There are two ways you can install these, you can choose either of the following methods and install it in your environment.

1. **Terminal (bash / cmd / PowerShell):**

```
pip install autogen==0.10.1 crewai==0.203.0 crewai-tools==0.76.0 langchain==0.3.27
langchain-chroma==0.2.5 langchain-community==0.3.27 langchain-core==0.3.80 langchain-
openai==0.3.30 langgraph==0.6.5 pydantic==2.11.10 pandas==2.3.3
```

OR

2. **Jupyter notebook cell:**

```
!pip install autogen==0.10.1 crewai==0.203.0 crewai-tools==0.76.0 langchain==0.3.27
langchain-chroma==0.2.5 langchain-community==0.3.27 langchain-core==0.3.80 langchain-
openai==0.3.30 langgraph==0.6.5 pydantic==2.11.10 pandas==2.3.3 --quiet
```

SQL Dependencies

If you are doing the first capstone project (Text2SQL Agent), then please ensure you have SQLite installed in your environment. There are two ways you can install these, you can choose either of the following methods and install it in your environment. If you already have SQLite installed then you do not need to run the following command.
Do note, if you are using a different operating system (OS) you might need to follow and use different commands for installing SQLite based on that OS.

1) Terminal (Ubuntu / Debian shell):

```
sudo apt-get install -y sqlite3
```

OR

2) Jupyter notebook cell:

```
!apt-get install -y sqlite3
```

Using Microsoft Azure OpenAI API

For this project you will use **GPT-4.1-mini** as the primary LLM. If you need to create a vector database (for Capstone 2), you will use the **text-embedding-3-small** model for embeddings.
In the course you would have learned how to use the regular OpenAI APIs. Here we show how you can easily replace standard OpenAI APIs with **Azure OpenAI** in your code.

Required Azure configuration
First, make sure you have the following configuration values from your Azure OpenAI resource. Remember these are just representative values, you will need to get the actual credential values from Fractal which will be provided to you.

```
AZURE_OPENAI_API_KEY = "<API KEY>"
AZURE_OPENAI_ENDPOINT = "https://your_endpoint.azure.com/"
EMBEDDINGS_DEPLOYMENT_NAME = "text-embedding-3-small"
MODEL_DEPLOYMENT_NAME = "gpt-4.1-mini"
OPENAI_API_VERSION = "2025-04-01-preview"
PROJECT_ID = "TEST"
```

What each of these means:

- **AZURE_OPENAI_API_KEY**
  Your secret API key for the Azure OpenAI resource. This authenticates your requests. Treat it like a password.

- **AZURE_OPENAI_ENDPOINT**
  The base URL of your Azure OpenAI resource. It usually looks like:
  `https://<your-resource-name>.openai.azure.com/`

- **EMBEDDINGS_DEPLOYMENT_NAME**
  The deployment name you configured in Azure for the **text-embedding-3-small** model.
  You will use this when creating embeddings. **Check the name to make sure you are using the right model.**

- **MODEL_DEPLOYMENT_NAME**
  The deployment name you configured in Azure for the **gpt-4.1-mini** model.
  You will use this for chat and completion style interactions. **Check the name to make sure you are using the right model.**

- **OPENAI_API_VERSION**
  The API version for Azure OpenAI. This controls which features and models are available. **Make sure you are using the right version number based on the deployment else you will get an error.**
  Example: `"2025-04-01-preview"`.

- **PROJECT_ID**
  A logical identifier for your project or environment. You can put any value like `"TEST"` or `“CAPSTONE”`.

Using Azure OpenAI with LangChain
If you are connecting to LLMs using **LangChain**, you need to switch from the standard OpenAI classes to the Azure-specific ones. You can also load the credentials mentioned above using an environment file (**.env**) if you wish to using python libraries like **python-dotenv**

Imports
Instead of:
```
from langchain_openai import ChatOpenAI, OpenAIEmbeddings
```

use:
```
from langchain_openai import AzureChatOpenAI, AzureOpenAIEmbeddings
```

Initializing the LLM client
Previously (standard OpenAI):
```
llm_client = ChatOpenAI(model="gpt-4.1-mini", temperature=0)
```

With Azure OpenAI:
```
llm_client = AzureChatOpenAI(
    # your Azure OpenAI endpoint URL
    azure_endpoint=AZURE_OPENAI_ENDPOINT,
    # your Azure OpenAI LLM API version
    api_version=OPENAI_API_VERSION,
    # your GPT-4.1-mini model deployment name
    azure_deployment=MODEL_DEPLOYMENT_NAME,
    temperature=0,
    # comment out the api_key if using AD tokens
    api_key=AZURE_OPENAI_API_KEY,
    # uncomment the azure_ad_token line below if using AD token auth mechanism
    # azure_ad_token=token,
    default_headers={
        "projectId": PROJECT_ID
    }
)
```

Initializing the Embeddings client
Previously (standard OpenAI):

```
embedding_client = OpenAIEmbeddings(model="text-embedding-3-small")
```

With Azure OpenAI:

```
embeddings_client = AzureOpenAIEmbeddings(
    # your Azure OpenAI endpoint URL
    azure_endpoint=AZURE_OPENAI_ENDPOINT,
    # your Azure OpenAI LLM API version
    api_version=OPENAI_API_VERSION,
    # your Azure text-embedding-3-small model deployment name
    azure_deployment=EMBEDDINGS_DEPLOYMENT_NAME,
    # comment out the api_key if using AD tokens
    api_key=AZURE_OPENAI_API_KEY,
    # uncomment the azure_ad_token line below if using AD token auth mechanism
    # azure_ad_token=token,
    default_headers={
        "projectId": PROJECT_ID
    }
)
```

You can now use `llm_client` and `embeddings_client` in the rest of your LangChain pipelines just like you would with the regular OpenAI clients, but all calls will go through **Azure OpenAI**.
You are free to use Azure OpenAI APIs in any other way also based on your own research.