Assignment 3: Using LangChain prebuilt chains for AI applications

Instructions

LangChain is an orchestrator library designed to streamline the process of building AI application. When it comes to AI applications, simply providing a prompt and receiving an answer has limited applications. Complicated applications require "chaining" of multiple components. These applications are routinely referred to as RAGs, retrieval augmented generation.

Orchestors like LangChain are meant to be modular. You can use components from different LLMs and pre-built libraries that streamline typical tasks in setting up custom applications. ALthough chains can be built from scratch, many prebuilt chains are available.

In this assignment, you will practice using LangChain API to setup some of the pre-built chains for different applications. The main challenge here is to read the documentation (always changing) and use the examples provided as well as add your own creative twist onto it.

Tasks

1. Review the LangChain documentation and review the different LangChain premade chains available to be used.

   LangChain Chains

2. Implement at least three different LCEL chains based on the documentation (do not implement legacy chains). For creating chains, you may need to instantiate vector databases and document loaders among other components. The documentation for using all the different loaders is available from LangChain and other sources. In class, we used the *create_stuff_documents_chain* to demonstrate reading data from a webpage (Durham's website) using a web document loader(crawler). You can use this chain again in this part of the assignment but should have a different application than one discussed in class (as an example perhaps read a PDF document rather than a webpage).

3. Use one of the pre-built LCEL chains in the previous step and try to rebuild it from scratch using basic components. An example of this type of modular chain was demonstrated in class implementing *create_stuff_documents_chain*. It is a python notebook under Week 9 lecture and is called *03_Custom_RAG_chain_From_components* which you can use for reference. However, you should do something other than *create_stuff_documents_chain* since this setup was already done in class (again, you can use it as a reference to see how to go about designing chains from components).

Submission

Submit the following items:

1. Any code that you used for creating your chains. It can be in a Python notebook or python code directly.
2. A video recording demonstrating your chains and the application. Videos should not be more than 10 minutes long. Focus on important parts of the code, and explain things in your own words.

Grading Criteria

Your assignment will be graded based on the following criteria:

- Successful implementation of three different ICEL chain(45%)
- Implementation of one of the pre-built chains from scratch using basic components (15%)
- Video demonstration clearly explaining the code and demonstrating understanding of LangChain API and different components (30%)
- Discretionary (10%)