SMU | LYLE SCHOOL OF ENGINEERING

CS 3341 FOUNDATIONS OF COMPUTING: FINAL C++ CODE ASSISTANT AGENT

Overview: Design and build an Autonomous Code Assistant Agent that can compile and fix C++ code using an LLM, FlowScript and Multithreaded job system.

Repo Link: https://classroom.github.com/a/d5tDVxWv

Due: December 9, 2023 @ 2:30pm.

Code Requirements (50%):

- Build upon the JobSystem, FlowScript and REST based LLM systems from Lab 1, 2, 3 and 4. You may modify, change, update any portion of previous labs to meet the objectives defined in the Final. Job System must be loaded as a library as in previous labs.
- Create an C++ Code Assistant Agent that will automatically resolve compile errors in c++ code, using the following properties.
    - Using an LLM, have it write FlowScript to build a compiling pipeline that will return all errors from the compilation of C++ code project (Lab 1, 2 and 4)
        - You will need to provide single/few shot training to ensure the LLM writes valid FlowScript
        - Ideally, this would be something that could be launched as a job to write FlowScript for a given prompt, but it can be customized to focus specifically on compiling
        - You will need to provide the LLM with a list of jobs that are available, so it can write the proper FlowScript to execute those jobs.
    - Using an LLM, have it submit and execute a FlowScript and obtain the results. (Lab 3 and 4)
    - Using an LLM, have it suggest and rewrite a section/all of a code file based upon compilation results (Lab 4).
    - Using an LLM, provide feedback on errors within a file with an explanation of changes that need to be made to correct the errors. (Lab 4)
    - Using an LLM, have it verify the results from suggested code improvements to verify the suggestion resolved compilation errors, and repeat the process if not (Lab 4)
    - The jobs must be launched from the JobSystem, but it can be written in any language
    - The overall automation framework can be written in any language, but must utilize a c++ job system to launch jobs.
    - Any LLM can be used, but must be documented in report on which LLM is utilized.

- The goal is to have fully working Code Assistant Agent, but the grading is not based upon the ability of the agent, rather your ability to connect all the systems together and have them work as a single project.

Report Requirements (50%):

- Document and describe all prompts utilized for the LLM
    - The writeup should provide a full description of the final prompts and reasoning for the specific wording and methodology of how the prompt works
    - Provide details on any parsing and setup for LLM context and prompt creation
    - Provide details on any zero/few shot training provided within the prompt/context
    - Provide details on any parsing and processing of LLM results sent back to user

- Document all Jobs created and used within the Agent
    - Provide details on how the job is created and executed
    - Expected outputs and write up for each of the jobs should be shown and fully documented
        - Showing inputs passed
        - Formatted inputs/prompts and context created for inputs
        - Raw results from job/LLM query
        - Final parsed output sent back/saved
- Document and discuss a complete example running through the entire automated system
    - Show all results from initial prompt to final code and report provide from system.
    - Provide a video showing the entire process running
    - Provide explanations and descriptions of final and intermediate results
        - FlowScript Prompt
        - FlowScript Produced
        - Compile Error Outputs
        - Agent Produced Report
        - Agent Produced Code
        - Agent Prompts
- Provide a final overall description of the agent and its capabilities.

All code in the "Code" folder, raw data files generated in "Data" and your final report in "Report" folder in the git repo. If your project fails the compile action upon final push to git repo, it will receive a 0. Be sure to resolve any issues with the compile action prior to deadline.