Stanford CS329T

Trustworthy Machine Learning:
Building and Evaluating Agentic Systems

Homework 1: Measuring and Improving Agent GPA

1. Assignment Overview

The objective of this assignment is to build and evaluate a multi-agent system using LangGraph and TruLens. By completing a series of tasks within a provided Jupyter Notebook (linked from this document below), you will measure and improve the agent's performance, referred to as **Agent GPA (Goal-Plan-Action Alignment)**. GPA is a key metric that assesses how effectively the agent's actions and plan align with the user's goal. Your final deliverables will be: (1) a one-page summary of your findings and proposed improvements, in the format described below, and (2) a printout of your completed notebook with outputs included. You will combine these two deliverables into a single combined PDF for Gradescope.

In this assignment you will learn:

1. **How agents are implemented in LangGraph** - LangGraph is a library for building stateful, multi-actor applications with LLMs. It provides a graph-based framework where each node represents a different agent or function, making it easier to orchestrate complex workflows. While there are other ways to implement agents, LangGraph is an excellent starting point because it provides clear structure, built-in state management, and visual workflow representation.
2. **How to connect agents to the web** - You'll use Tavily, a search API designed for AI agents that allows them to access real-time web information. This enables your agents to retrieve current data and perform web-based research as part of their workflows.
3. **How to use evaluation tools** - You'll work with TruLens, an open-source evaluation framework designed specifically for LLM applications. While there are other evaluation tools available, TruLens provides comprehensive tracing and evaluation capabilities that help you understand agent behavior.
4. **The typical workflow for creating reliable agents** - You'll experience the standard iterative process: first curating a small dataset, letting the agent perform its tasks, then annotating errors and exploring where improvements are needed.
5. **How to validate automated evaluations** - You'll check whether automated evaluations catch the same errors you identify manually, providing a sanity check on whether the evaluation system is working correctly and aligns with human judgment.
6. **What types of improvements make meaningful differences in agent performance** - You'll be tasked with addressing the results of your evaluations and making improvements to the agent.

2. Assignment Steps

Complete the following tasks. You'll prepare a 1-page summary document addressing certain questions set out below.

1. **Complete Building and Evaluating Data Agents** from deeplearning.ai. One of the instructors should look familiar. This should take around 2 hours to complete.
2. **Work through the assignment notebook.**
   - Here is the notebook. Two API keys are needed (Tavily and OpenAI). Tavily is free. To get an OpenAI key, you will need to give your credit card info, but the total cost to complete this homework should be very low (< $2.00). If this poses a problem to you, please reach out to us

on ed with a private post! **Redact your API keys when you print your notebook and submit your PDF.** Alternatively, you may change the method for loading your API keys into the notebook such that they are not shown in plain text. Any modern coding assistant can guide you through this process.

- Work through the notebook through and including step 8 to understand the implementation of the `planner_node`, `executor_node`, and other agents.
- **Note:** The initial iteration of the agent is fully implemented and there are no "to do" style coding tasks in the notebook, although you will need to add human annotations in step 9 (see below).

3. **Complete human annotations.**
    - Work through step 9 of the notebook, following the instructions. This will involve running the provided test examples and examining the traces in TruLens using `run_dashboard()`. You will need to annotate the traces with failure modes aligned with the Goal-Plan-Act Evaluations that you identify.
    - **Section 1 of your one-page summary should consist in a brief rationale for each of your human annotations.**
4. **Run agent & analyze metrics.**
    - Continue following the instructions in the notebook. You will run the agent again with the LLM judge metrics now included.
    - Continue through `session.get_leaderboard()` to view the computed metrics.
    - **In section 2 of your submission**, summarize the key metrics and discuss how they relate to the concept of Goal-Plan-Action Alignment.
    - Examine the evaluation metrics on a trace-level, and **include in your section 2** an analysis of the extent to which the LLM judge metrics agree or disagree with your human annotations. Explain in this section what your investigation reveals about the failure modes of the agent.
5. **Propose improvements, implement them and re-run the agent.**
    - Based on your analysis of the metrics, propose one or more specific modifications to the agent's prompts or architecture.
    - Implement these improvements in the notebook (for example, you might make improvements to the prompts).
    - In **section 3 of your submission**, justify your improvements by explaining how they address an identified failure mode and improve the agent's GPA.

3. Homework Deliverables

You will submit the homework through Gradescope. You will need to combine your 1-page report with your notebook and submit them together as a single PDF.

1. The first page of your submission should be the one-page summary document with sections 1, 2, and 3 as described above, all clearly labeled.
2. This should be followed by the PDF printout of your notebook with outputs included, including your final human annotations added in step 9, and the prompt or architecture updates you made to the agent.

4. Grading Rubric

- **Completion of Tasks (40%):** Successful execution of the notebook with and without the improvements.
- **Annotation Quality (20%):** Thoughtfulness and quality of trace annotations.
- **Clarity of Metrics Analysis (30%):** Insightful summary of results and identified failure modes.
- **Quality of Proposed Improvements (10%):** Well-reasoned and justified suggestions for agent improvements.

5. Getting Started with AI Assistance

To learn more agent construction and the libraries used, you are allowed and encouraged to use AI assistants such as Cursor, or Claude Code. You can ask questions such as the following:

- Describe the architecture of the agent system created in this notebook
- Explain what langgraph is and how it is used in this notebook
- Help me understand how the tracing and logging is done with trulens
- How do I add human feedback to annotate the agent traces?
- Help me understand how the evaluation is done with trulens
- What improvements should I make to the agent to improve <insert your failure mode description here>