Human in a loop: Feedback based learning - Math Routing Agent

Your goal is to design an Agentic-RAG architecture system that replicates a mathematical professor. When presented with a mathematical question, the system should understand and generate a step-by-step solution, simplifying it for the student. The response must first check the existing knowledge base for relevant information. If the solution is not found in the knowledge base, the system should perform a web search and then generate the step-by-step solution.

The Math-agent must have a solid routing pipeline:

- **AI Gateway mainly guardrails**
    - **Integrate the routing agent with AI gateway by adding the Input and Output guardrails. Do proper research on how the guardrails is incorporated.**
    - The app should be focused on only delivering education based content mainly: Mathematics.
    -
- **Knowledge Base Creation**
    - To build an Agentic RAG, the system must adapt to a knowledge base to retrieve relevant questions effectively.
    - You can choose any dataset or math module as the **knowledge base and store it in a VectorDB.**
    - If the user's question exists in the knowledge base, the system should retrieve it and generate a step-by-step solution to provide simplified results.

- **Web Search or Using MCP**
    - If the user's question is not found in the knowledge base, **the system should perform a web search to fetch the result and generate a response.**
    - However, if the question is not available online, the system must ensure it does not provide incorrect results.
    - Furthermore, for this step, you need to have a pipeline for **Web Search extraction**.
    - **Usage of Model Context Protocol (MCP) is MUST**.

- **Human-in-the loop mechanism**
    - Incorporate an evaluation or feedback agent layer to enhance the agent's performance through self-learning capabilities.
    - **Note: You need to take human feedback and refine the response accordingly.**
    - The final response is subjective and requires a Human-in-the-Loop mechanism for feedback and validation.
    - **Bonus**: If you use the DSPy library for this.

- **Bonus: Good to have: Benchmark the solution or develop results on JEE Bench**
    - It would be great if you could develop and provide a script to execute the benchmark for the Agentic approach you choose against the JEE Bench dataset.

Final Proposal

- Input & Output guardrails used for privacy. What approach did you take and why?
- Knowledge Base - Dataset used and details on it: including 2-3 questions to try the system.
- Web Search capabilities or MCP Setup: - include 2-3 questions that are not from the knowledge base. Need to include the strategy taken for Web extraction or MCP Setup.
- A detailed report on the Human-in-a-loop routing for the Agentic workflow taken to build the Math Agent.
- [Bonus] If the JEE Bench is carried out, do mention the results in the proposal.

Deliverables

- **PDF file: with the above Final Proposal**
- **Source code:** Submit the source code of your Math Agent, along with any necessary scripts, configuration files, and documentation/proposal.
- **Demo Video**: A video showcasing the architecture flowchart and the output generated.
- Develop the application using **FastAPI and React app**. Once you develop the code logic as per the Final proposal, you can develop the **FastAPI application** for the given code.

Assignment Evaluation Criteria:

- Your agent will be evaluated based on the following parameters:
  - If the system is able to route efficiently between knowledge base and search.
  - Functionality of guardrails and feedback mechanism capabilities.
  - Feasibility and practicality of implementing the proposed solutions.
  - Quality and clarity of the final proposal, including actionable insights.

Suggested Tools & Frameworks

- Agent Frameworks: Use LangGraph, LLamaIndex, Autogen, or CrewAI for building AI agents. You can check DeepLearning AI short courses for quick tutorials
- Search & Retrieval:
  - Leverage Tavily or Exa or Serper for advanced web search
  - Qdrant or Weaviate for better search retrieval.
  - Check Awesome MCP server for the relevant MCP server for Search.
- For feedback: Utilize the framework like: DSPy [**get bonus advantage**]

Feel free to use any other tools you're comfortable with.

Learning Resources

- Deep learning Short sources:
  - RAG- Langchain:
    - https://youtube.com/playlist?list=PLiBGj68xqdPj5bECy3Xk3IG_y_PVR6Zwo&si=9MGuC5dGs5I5L5He
    - https://www.deeplearning.ai/short-courses/multi-ai-agent-systems-with-crewai/
    - https://www.deeplearning.ai/short-courses/ai-agents-in-langgraph/
  - Agents
    - https://youtube.com/playlist?list=PLiBGj68xqdPhUlghO14TABwx-QlQjfIXm&si=yocZsv8nai1B1uqP
  - MCP:
    - https://learn.deeplearning.ai/courses/mcp-build-rich-context-ai-apps-with-anthropic/lesson/fkbhh/introduction
    - https://youtube.com/playlist?list=PLiBGj68xqdPhWidhMK_IN6tDQguUKHEcB&si=h_wWlR0J-9thvZMB