Report

1. Agent Design and Architecture

1.1 Overall Architecture

This project designed and implemented an autonomous Moltbook AI agent to perform a series of tasks on a real social platform. The system adopts a modular design, consisting of three core components:

1. Tool Layer: This layer encapsulates the Moltbook platform's REST API, providing a set of tool functions that can be invoked by the agent. These tools cover core functionalities such as retrieving feeds, semantic searching, creating posts, commenting, upvoting, and subscribing to communities. Each tool interacts with the Moltbook API using an API key securely retrieved from Colab Secrets for authentication.

2. LLM Decision Layer: This layer uses the Google Gemini 2.5 Flash model as the "brain" of the agent. It is responsible for understanding task objectives, planning execution steps, and deciding which tools to call. Integrated via the langchain-google-genai library, it supports tool-calling capabilities, enabling the LLM to automatically select and invoke appropriate tools based on the conversational context.

3. Execution Loop Layer: This layer implements the moltbook_agent_loop function, which serves as the core control flow of the agent. The loop initializes the conversation, calls the LLM for the next action, executes the tool, processes the results, and feeds them back to the LLM for the next decision. This cycle continues until the task is completed or the maximum number of turns is reached.

1.2 Key Design Decisions

The selection of Gemini 2.5 Flash as the LLM was driven by its speed, cost-effectiveness, and native support for tool calling, making it ideal for autonomous agents that require rapid decision-making and multi-turn interaction. The use of LangChain's @tool decorator standardizes the API calls, making them easily understandable by the LLM and unifying error handling. Storing the API key in Colab Secrets enhances security by avoiding hardcoding, while a maximum turn limit in the execution loop prevents the agent from entering infinite loops, ensuring task completion within a controlled scope.

2. Decision Logic and Autonomy Level

2.1 Decision Logic

The agent's decision-making is driven by the LLM, following a structured process. It begins by parsing the user's instructions, which are fed into the LLM as an initial prompt. The LLM then selects the appropriate tools and their order of invocation based on the task goal and the available toolset. It automatically generates the necessary parameters for each tool call, such as the community path, post ID, and comment content. After a tool is executed, the results are added to the conversation history, allowing the LLM to adjust its strategy for subsequent steps.

2.2 Autonomy Level

The agent exhibits a high level of autonomy. It can autonomously plan the steps required to complete complex tasks without human intervention. It is context-aware, remembering previous operations and results to inform its decisions. Additionally, it handles errors robustly; when encountering API errors, such as an invalid key, the error information is relayed back to the LLM, which then decides on the next course of action, whether it be retrying or terminating the process.

3. Screenshots or Logs of Moltbook Interactions

The agent's interaction with the Moltbook platform is captured in detailed logs. These logs typically show the agent starting a new turn, calling the LLM for a decision, executing a tool call, and processing the result. For example, the logs might show the agent first subscribing to a community, then successfully upvoting a specified post, and finally leaving a comment. Each step is timestamped, providing a clear audit trail of the agent's actions and the platform's responses, which serves as concrete evidence of the system's functionality.

4. Conclusion

This project successfully built an autonomous Moltbook AI agent capable of performing tasks like authentication, community subscription, post upvoting, and commenting on a real social platform. By combining a modular tool layer, powerful LLM decision-making, and a robust execution loop, the agent demonstrates the potential for AI to operate autonomously in open digital environments. The system's design is highly extensible, allowing for easy integration of additional tools or expansion to other platforms. Security and controllability are ensured through the use of Colab Secrets and maximum turn limits, making it a reliable solution for automating social media interactions.