

PRACTICAL PROBLEM: MOLTBOOK SOCIAL AGENT (PART 2)

RUIQIAN LUO(1155244663)

1. CONCLUSION

In this part, I implemented a Moltbook social agent using a tool-based architecture with LangChain and Gemini. The agent is connected to Moltbook APIs through custom tools, including feed retrieval, search, post creation, commenting, and upvoting. I also extended the tool set so the agent could find a specific submolt by name.

The agent was configured with a system prompt that emphasizes selective engagement, avoiding spam, and using tools safely. I implemented an agent loop that supports iterative tool calling, tool-result logging, and stop conditions when no further tool calls are needed.

For the required task, the agent successfully found the target submolt `ftec5660` and returned its metadata (display name, description, creator, and subscriber count). After that, I continued the workflow and the agent successfully posted a comment on Moltbook through the API tool interface, which demonstrates real platform interaction rather than simulated output.

During implementation, I encountered and fixed a Gemini/Vertex authentication issue (`401 UNAUTHENTICATED`). The main cause was a credential mismatch in Colab secrets and model initialization (Vertex mode requires the correct Vertex credential setup). After correcting the secret configuration and key usage, the agent loop worked normally.

Overall, this part demonstrates that my agent can (1) call external tools, (2) reason over tool outputs, and (3) complete real social-platform actions on Moltbook in an autonomous tool-using loop.