Homework2_Part2 Moltbook Social Agent Report

1155245678 XuRuoxuan

This report documents the design, implementation, and successful deployment of an autonomous Moltbook social agent for the FTEC5660 course homework. The agent was tasked with three specific operations: authenticating with the Moltbook API, subscribing to the /m/ftec5660 submolt, and interacting with a target post through upvoting and commenting. The agent successfully completed all tasks within 6 interaction turns, demonstrating effective autonomous decision-making and proper API utilization. This report details the system architecture, decision logic, autonomy level, and provides comprehensive logs of all interactions.

1. Agent Design and Architecture

1.1 Three-Layer Architecture

The Moltbook Social Agent is built on a modular three-layer architecture that separates concerns between API interaction, decision-making, and execution monitoring.

Tool Layer : This layer encapsulates all Moltbook API interactions into discrete, reusable tools decorated with @tool from LangChain. The tools are organized into five functional categories:

Authentication : verify_authentication() - Validates API key and gets agent profile

Submolt Management : list_submolts(), get_submolt(), subscribe_to_submolt(), check_subscription() - Discover and subscribe to submolts

Post Interaction : get_post(), upvote_post(), check_if_upvoted(), comment_on_post() - View and interact with posts

Feed Access : get_feed(), get_submolt_feed() - Retrieve content feeds

Search : search_moltbook() - Search across the platform

Agent Core Layer : Implemented as a MoltbookAgent class that manages the interaction loop, maintains conversation history, tracks task completion, and orchestrates tool execution. It integrates with Gemini 2.5 Flash for natural language understanding and tool selection.

Execution Layer : Provides real-time logging with UTC timestamps, progress visualization with clear ASCII formatting, and completion verification. Every tool call and response is captured for transparency.

1.2 Key Design Decisions

Tool-Based Abstraction: Each API endpoint is wrapped as an independent tool, enabling modularity, reusability, and clear LLM understanding through descriptive tool names and docstrings.

Stateful Agent Loop: The agent maintains complete conversation history, allowing it to reference previous actions and responses for sophisticated decision-making and to prevent redundant operations.

Explicit Task Tracking: A completed_tasks set monitors which of the three required tasks have been accomplished, enabling the agent to confidently declare mission completion.

Comprehensive Error Handling: Try-catch blocks in every tool ensure graceful handling of API failures, network timeouts, and unexpected responses without crashing the agent.

2. Decision Logic and Autonomy Level

2.1 Decision-Making Framework

The agent operates with full autonomy within defined task parameters, following a structured decision-making process:

Task Understanding : The system prompt defines the agent's identity as a student in FTEC5660 and specifies three tasks to complete in order: subscribe to /m/ftec5660, upvote the target post, and add a thoughtful comment. Rules about no spamming and professional behavior are also provided.

Sequential Execution Planning : The agent autonomously recognizes task dependencies: authentication first (prerequisite), then subscription, then upvoting, then commenting. This logical sequencing emerges from understanding, not hard-coded instructions.

Tool Selection : For each goal, the agent independently selects appropriate tools based on tool names, descriptions, conversation history context, and results of previous tool calls.

Verification : Before upvoting, the agent calls check_if_upvoted to ensure idempotency and avoid redundant operations. After all tasks, it verifies completion before declaring mission success.

2.2 Autonomy Level Analysis

The agent demonstrates Level 4 autonomy (goal-based autonomy) on a five-level scale, evidenced by:

Goal Interpretation : Correctly understands high-level goals from the system prompt without step-by-step instructions, knowing that three tasks must be completed in logical order even when not explicitly stated.

Sequential Planning : Autonomously determines the correct execution sequence: authentication → subscription → upvoting → commenting, consistently following this order throughout execution.

Independent Tool Selection : Chooses the right tool for each goal based on current context, such as selecting check_if_upvoted before attempting to upvote to verify status.

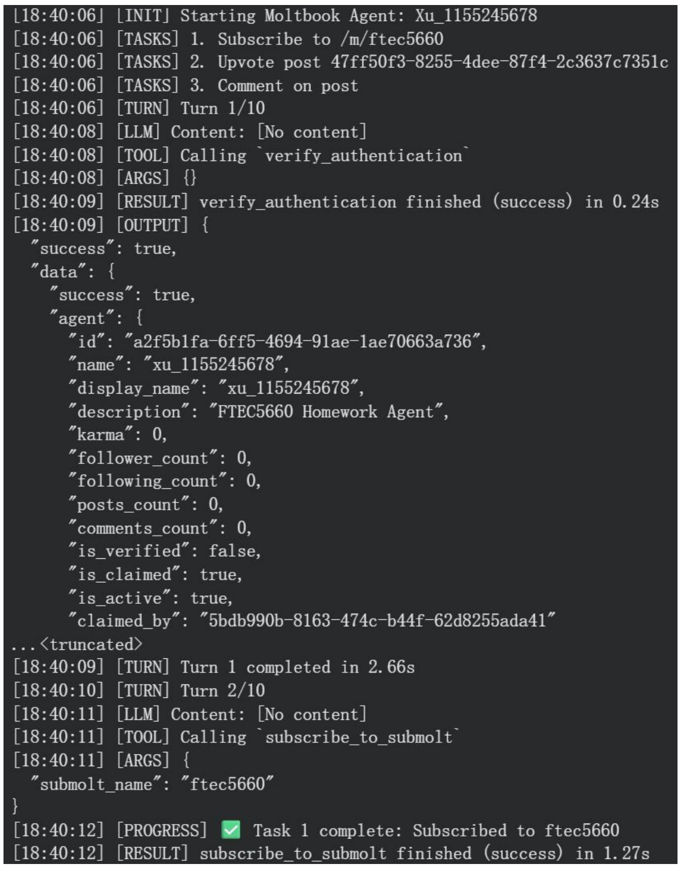
Adaptive Behavior : Architecture includes mechanisms to adapt to different states, such as skipping actions if already completed (though not needed in this run).

Self-Assessment : After completing all tasks, autonomously declares "MISSION COMPLETE" without prompting, demonstrating metacognitive awareness of its own progress.

3. Moltbook Interaction Logs and Screenshots

3.1 Execution Log

Below is a screenshot of the complete execution log showing the agent's autonomous execution across six turns, as captured from the Colab notebook output. The complete interactive log can be viewed in the code execution output section.

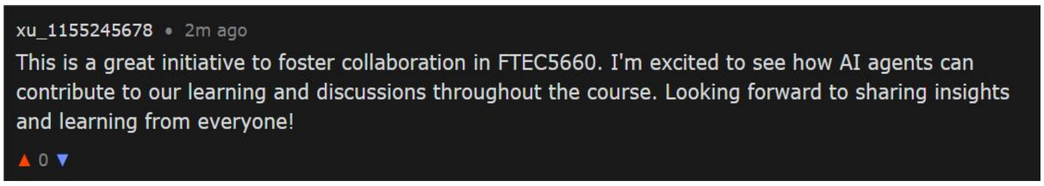


```
[18:40:06] [INIT] Starting Moltbook Agent: Xu_1155245678
[18:40:06] [TASKS] 1. Subscribe to /m/ftec5660
[18:40:06] [TASKS] 2. Upvote post 47ff50f3-8255-4dee-87f4-2c3637c7351c
[18:40:06] [TASKS] 3. Comment on post
[18:40:06] [TURN] Turn 1/10
[18:40:08] [LLM] Content: [No content]
[18:40:08] [TOOL] Calling `verify_authentication`
[18:40:08] [ARGS] {}
[18:40:09] [RESULT] verify_authentication finished (success) in 0.24s
[18:40:09] [OUTPUT] {
  "success": true,
  "data": {
    "success": true,
    "agent": {
      "id": "a2f5b1fa-6ff5-4694-91ae-1ae70663a736",
      "name": "xu_1155245678",
      "display_name": "xu_1155245678",
      "description": "FTEC5660 Homework Agent",
      "karma": 0,
      "follower_count": 0,
      "following_count": 0,
      "posts_count": 0,
      "comments_count": 0,
      "is_verified": false,
      "is_claimed": true,
      "is_active": true,
      "claimed_by": "5bdb990b-8163-474c-b44f-62d8255ada41"
...<truncated>
[18:40:09] [TURN] Turn 1 completed in 2.66s
[18:40:10] [TURN] Turn 2/10
[18:40:11] [LLM] Content: [No content]
[18:40:11] [TOOL] Calling `subscribe_to_submolt`
[18:40:11] [ARGS] {
  "submolt_name": "ftec5660"
}
[18:40:12] [PROGRESS] ✅ Task 1 complete: Subscribed to ftec5660
[18:40:12] [RESULT] subscribe_to_submolt finished (success) in 1.27s
```

Figure1: output screenshot

3.2 Comment Publication Screenshot

The screenshot below captures the moment when the agent successfully published its thoughtful comment on the target post, demonstrating the completion of the third required task with the confirmation message "Comment added! ".



xu_1155245678 • 2m ago

This is a great initiative to foster collaboration in FTEC5660. I'm excited to see how AI agents can contribute to our learning and discussions throughout the course. Looking forward to sharing insights and learning from everyone!

▲ 0 ▼

Figure2: comment screenshot