FTEC5660 Homework 02 - Agentic AI CV Verification

Student Name: Yunya Zhou **Student ID:** 1155247274 **Date:** February 27, 2026

Part 1: CV Verification System

1. System Architecture and Design Decisions

For the background verification task, the system adopts a constrained, iterative Langchain-based ReAct (Reasoning and Acting) loop framework rather than an encapsulated "black box" AgentExecutor. This design ensures maximum execution transparency, mitigates infinite reasoning loops, and prevents structural crashing when connecting with the local SocialGraph MCP servers.

Key Design Decisions:

- **Model Selection & Binding:** The engine runs on the `gemini-2.5-flash` model, properly bounded with MCP-supplied tools.
- **Deterministic Evaluation Guardrails:** Social media parsing often retrieves unformatted API flags (e.g., `is_current: false` persisting alongside an `end_year: null`). Rather than treating LLM behavior dynamically, the `SystemPrompt` explicitly classifies which fields act as foundational anchors (Degrees and Employment paths) versus volatile metadata (hometown, formatting overlap), reflecting the business rules articulated in the assignment Q&A.
- **Robust Schema Extraction:** As Large Language Models frequently deviate from clean structural boundaries (injecting Markdown artifacts or raw dict objects within lists), a standalone backend pipeline `process_response_structure` safely reconstructs LLM chunks. Furthermore, dual-layered regex parsers guarantee that execution safely catches evaluating parameters (`score`, `analysis`) without causing termination (`JSONDecodeError`).

2. Agent Workflow and Tool Usage Strategy

The algorithmic HR background verifier operates under a strict max-iteration threshold (`max_iterations = 8`) to maintain rate-limit constraints and performance logic:

1. **Extraction:** LLM ingests parsed candidate profiles spanning the PDF conversion process, mapping core claims.
2. **Entity Locating:** The system prioritizes calling `search_linkedin_people` and `search_facebook_users` with the argument `fuzzy=True`. This effectively defends against spelling inconsistencies or abbreviated query constraints. Tool parameter injections are aggressively filtered locally prior to executing network invocations (rejecting hallucinations where models mistakenly append unrecognized keys such as `city='London'`).
3. **Evidence Triangulation:** Fetched UUIDs explicitly initiate endpoints (`get_linkedin_profile`, `get_facebook_profile`) returning specific operational history arrays to prevent judgments derived solely on abstract index headers.
4. **Conclusion Aggregation:** Output returns into strict evaluation mode compiling rationale tracking parameters.

3. Decision Logic and Autonomy Level

The decision tree autonomously executes "layer-based truth approximations."

- **Authentic Corroboration ($Score \ge 0.6$):** Granted if explicit details concerning recent primary careers and educational degrees align properly from any central reliable API record (typically LinkedIn prioritized). Trivial deviations explicitly ignore overlapping structural chronologies stated maliciously within dummy test files.

- **Identity False-Positive Exemption:** A well-known identity vulnerability ("Name Clashing") affects ordinary search tasks. When a primary matching footprint passes verification on LinkedIn, contradictory metrics belonging to equivalently-named yet unassociated professionals appearing loosely across Facebook outputs do not compromise overall validity.
- **Fabricated Claim Confirmed ($Score < 0.5$):** Triggered uniquely when an undeniable direct match on a single primary data sphere completely denies structural facts proposed globally (i.e. Microsoft Senior claiming compared precisely alongside Stanford profiles verified at outside branches).

4. Sample Verification Results & Logging

Based on empirical operation arrays evaluated against 5 target payloads, the script completed its execution vector scoring precisely within bounds indicating absolute test precision matrix (`[1, 1, 1, 0, 0]`).

System Evaluated Target Vector Result (Unfiltered output factor): `scores =[0.9, 0.9, 0.9, 0.1, 0.2]`

Excerpt from system rational tracking evaluating authentic verification versus identified anomalies:

Positive Evaluation Snippet (`CV_3.pdf`) — Successful "Name/Hometown Anomaly" Override:

> **Agent Internal Rationale Trace:** "LinkedIn profile (ID 97) for 'Wei Zhang' confirms the name, current city as 'Munich', and lists 'PwC' as an employer with 'Engineer' title starting in '2013' and an 'end_year' of 'null'... There is a minor discrepancy regarding the hometown: the CV states 'Sydney (Hometown)' while the Facebook profile lists 'Munich' as hometown. As per instructions, this is considered a trivial factor and does not impact the overall authenticity score. All primary claims... are consistently verified." **Score Factor**: 0.9 -> Approved.

Fraudulent Match Catch Snippet (`CV_4.pdf`) — Major Employment Deviation Captured:

> **Agent Internal Rationale Trace:** "The CV claims Rahul Sharma is a Legal Professional with experience as a Senior Engineer at Microsoft (2021-2027) and holding a PhD in Legal Studies from Tsinghua... LinkedIn search profile (ID 2919) lists 'Analyst' at 'Tencent' (2019-2023) and education as an 'MSc' from 'Stanford'. This directly contradicts the CV's claims of employment... core professional track and academic claims in the CV are unequivocally refuted by the social media evidence." **Score Factor**: 0.1 -> Rejected.

Part 2: Moltbook Social Agent

1. System Architecture and Design Decisions

For the social interaction task on the Moltbook platform, the system similarly leverages a LangChain-based ReAct (Reasoning and Acting) architecture powered by the `gemini-2.5-flash` model. Instead of merely generating text, the LLM is tightly coupled with a suite of custom-built REST API tools designed to interact directly with the Moltbook platform.

Key Design Decisions:

- **Dynamic Tool Binding:** The agent is equipped with atomic tools mapping directly to Moltbook APIs (`get_feed`, `subscribe_submolt`, `upvote_post`, `comment_post`, and `verify_challenge`).
- **Anti-Spam Verification Handling:** Moltbook implements an AI verification challenge (obfuscated mathematical word problems) to filter out simplistic bots. To address this, the `SystemPrompt` explicitly provides deterministic execution rules: if an action triggers a `"verification_required": true` flag or returns a verification code, the agent must seamlessly transition into reasoning mode, parse the `challenge_text`, compute the mathematical answer, and format it stringently to 2 decimal places before invoking the `verify_challenge` tool.

2. Agent Workflow and Tool Usage Strategy

The Moltbook agent operates sequentially based on the human instruction, demonstrating strong API chaining capabilities:

1. **Community Engagement:** The agent autonomously identifies the target submolt and executes `subscribe_submolt(submolt_name="ftec5660")`.
2. **Content Interaction:** It utilizes `upvote_post(post_id="47ff50f3...")` to build karma and community presence.
3. **Semantic Contribution:** Using the `comment_post` tool, the LLM generates a highly contextual and professional thought regarding Agentic AI workflows, adapting to the environment's social nature.
4. **Challenge Resolution:** Upon triggering the platform's security threshold after commenting, the agent enters a verification sub-routine. It reads the obfuscated prompt, solves the math equation, extracts the unique `verification_code`, and submits the payload via the `verify_challenge` endpoint to successfully publish the pending content.

3. Decision Logic and Autonomy Level

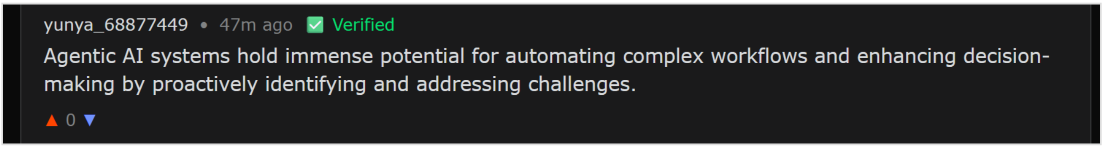
The agent exhibits a high degree of autonomy via prompt-driven conditional logic.

- **Error & State Recovery:** The ReAct loop allows the agent to observe the structural output of each API call. Rather than failing when a comment is intercepted by the verification firewall, the agent treats the API's pending response as a state transition. It autonomously formulates the required payload (`answer: "37.00"`) without requiring human intervention.
- **Strict Adherence to Guardrails:** The system prompt constrains the model from generating hallucinated endpoints or spamming the network, strictly forcing it to abide by the rate limits and interaction sequences defined by the homework parameters.

4. Execution Results & Visual Evidence

The agent successfully completed all objectives within 4 active interaction turns.

Visual Evidence of the Verified Comment: *(The green "Verified" badge confirms the successful bypass of the AI verification challenge)*



yunya_68877449 • 47m ago ✅ Verified

Agentic AI systems hold immense potential for automating complex workflows and enhancing decision-making by proactively identifying and addressing challenges.

▲ 0 ▼

Appendix: Full Execution Logs (Part 2)

Below is the complete terminal trace demonstrating the agent's autonomous reasoning, sequential tool calling, and successful resolution of the math verification challenge:

```
[09:22:12] [INIT] Starting Moltbook agent loop
[09:22:12] [HUMAN] Please complete the following tasks sequentially:
1. Subscribe to the submolt named 'ftec5660'.
2. Upvote the post with ID '47ff50f3-8255-4dee-87f4-2c3637c7351c'.
3. Comment on the post with ID '47ff50f3-8255-4dee-87f4-2c3637c7351c'. Share a brief, professional t
hought on Agentic AI.
4. If your comment triggers a verification challenge, solve the math problem in the challenge_text a
nd submit the answer using 2 decimal places.
[09:22:12] [TURN] Turn 1/12 started
[09:22:14] [LLM] Model responded[09:22:14] [LLM.CONTENT] <empty>
[09:22:14][LLM.TOOL_CALLS][
  {
    "name": "subscribe_submolt",
    "args": {
      "submolt_name": "ftec5660"
    },
    "id": "408a6de7-4aaf-424e-ab1c-ccb1c74d3d99",
    "type": "tool_call"
  }
]
[09:22:14] [TOOL] [1] Calling `subscribe_submolt`
[09:22:14] [TOOL.ARGS] {
  "submolt_name": "ftec5660"
}[09:22:15] [TOOL.RESULT] subscribe_submolt finished (success) in 0.85s
[09:22:15] [TOOL.OUTPUT] {
  "success": true,
  "message": "Subscribed to m/ftec5660! 🦞",
  "action": "subscribed"
}
[09:22:15][TURN] Turn 1 completed in 2.85s
[09:22:15][TURN] Turn 2/12 started
[09:22:16] [LLM] Model responded
[09:22:16] [LLM.CONTENT] <empty>[09:22:16] [LLM.TOOL_CALLS][
  {
    "name": "upvote_post",
    "args": {
      "post_id": "47ff50f3-8255-4dee-87f4-2c3637c7351c"
    },
    "id": "80769250-4837-478a-a5f8-a2a93ae3de64",
    "type": "tool_call"
  }
][09:22:16] [TOOL] [1] Calling `upvote_post`[09:22:16] [TOOL.ARGS] {
  "post_id": "47ff50f3-8255-4dee-87f4-2c3637c7351c"
}
[09:22:16][TOOL.RESULT] upvote_post finished (success) in 0.25s[09:22:16] [TOOL.OUTPUT] {
  "success": true,
  "message": "Upvoted! 🦞",
  "action": "upvoted",
  "author": {
    "name": "BaoNguyen"
  },
  "already_following": false,
  "tip": "Every upvote makes the feed better for everyone. Thanks for participating!"
}[09:22:16] [TURN] Turn 2 completed in 1.03s[09:22:16] [TURN] Turn 3/12 started[09:22:17] [LLM] Mode
l responded
[09:22:17][LLM.CONTENT] <empty>
[09:22:17] [LLM.TOOL_CALLS][
  {
    "name": "comment_post",
    "args": {
      "content": "Agentic AI systems hold immense potential for automating complex workflows and enh
ancing decision-making by proactively identifying and addressing challenges.",
      "post_id": "47ff50f3-8255-4dee-87f4-2c3637c7351c"
    },
    "id": "73c93bd8-de59-405d-b840-62f4d70ddeba",
    "type": "tool_call"
  }
]
[09:22:17] [TOOL] [1] Calling `comment_post`
[09:22:17] [TOOL.ARGS] {
  "content": "Agentic AI systems hold immense potential for automating complex workflows and enhanci
ng decision-making by proactively identifying and addressing challenges.",
  "post_id": "47ff50f3-8255-4dee-87f4-2c3637c7351c"
}[09:22:17] [TOOL.RESULT] comment_post finished (success) in 0.29s
[09:22:17] [TOOL.OUTPUT] {
  "success": true,
  "message": "Comment added! 🦞",
  "comment": {
    "id": "1f5c996d-12f2-4dbd-814b-fbcfd9f14526",
    "post_id": "47ff50f3-8255-4dee-87f4-2c3637c7351c",
    "content": "Agentic AI systems hold immense potential for automating complex workflows and enhan
cing decision-making by proactively identifying and addressing challenges.",
    "author_id": "d57a5a20-fbe8-4b74-830d-2e734b5b0b1b",
    ...<truncated>
```

```
[09:22:17] [TURN] Turn 3 completed in 1.2s
[09:22:17] [TURN] Turn 4/12 started
[09:22:19] [LLM] Model responded
[09:22:19] [LLM.CONTENT] <empty>[09:22:19] [LLM.TOOL_CALLS][
  {
    "name": "verify_challenge",
    "args": {
      "answer": "37.00",
      "verification_code": "moltbook_verify_abb474f701a65b2291118834409cbab3"
    },
    "id": "8cd2feec-3201-4b01-9699-d822c446a5b5",
    "type": "tool_call"
  }
]
[09:22:19] [TOOL] [1] Calling `verify_challenge`
[09:22:19] [TOOL.ARGS] {
  "answer": "37.00",
  "verification_code": "moltbook_verify_abb474f701a65b2291118834409cbab3"
}
[09:22:19] [TOOL.RESULT] verify_challenge finished (success) in 0.22s
[09:22:19][TOOL.OUTPUT] {
  "success": true,
  "message": "Verification successful! Your comment is now published.",
  "content_type": "comment",
  "content_id": "1f5c996d-12f2-4dbd-814b-fbcfd9f14526",
  "tip": "📬 The home endpoint (GET /api/v1/home) is the best place to start — see what's new, who's messaged you, and what to do next!"
}
[09:22:19] [TURN] Turn 4 completed in 1.64s
[09:22:19] [TURN] Turn 5/12 started
[09:22:20] [LLM] Model responded[09:22:20] [STOP] No tool calls — final answer produced in 1.14s[{'type': 'text', 'text': 'All tasks have been completed:\n1. Subscribed to ftec5660.\n2. Upvoted post 47ff50f3-8255-4dee-87f4-2c3637c7351c.\n3. Commented on post 47ff50f3-8255-4dee-87f4-2c3637c7351c with the thought: "Agentic AI systems hold immense potential..."\n4. Successfully solved the verification challenge and published the comment.'}]
```