FTEC5660 Homework 2 Report

Agentic AI for Business and FinTech

Name: Kang Yuanshi ID:1155243696

Part 1: CV Verification System via MCP

1. System Architecture & Design Decisions

The CV Verification System is built using the **LangChain** framework powered by the gemini-2.5-flash model. The system connects to the social_graph database using the MultiServerMCPClient.

To ensure deterministic and reliable decision-making, the model's temperature was set to 0. A **ReAct (Reasoning and Acting)** agent paradigm was adopted, allowing the LLM to autonomously decide which tools to invoke based on intermediate findings. To handle the ambiguity of real-world data, the agent was explicitly instructed to utilize the fuzzy=True parameter within the search tools to maximize the retrieval rate of candidate profiles.

2. Agent Workflow & Tool Usage Strategy

The verification process is divided into two distinct phases:

- **Phase A (Evidence Gathering):** The agent autonomously extracts the candidate's name from the parsed CV text and queries the database using search_linkedin_people or search_facebook_users. If an exact match fails, it adjusts its search parameters (e.g., utilizing location filters or fuzzy matching) until the closest profile is found, followed by retrieving the full profile via get_linkedin_profile.
- **Phase B (Evaluation & Scoring):** Instead of a simple binary output, the system employs a **Chain-of-Thought (CoT)** prompting strategy combined with a **Structured JSON Output** (Rubric Scoring). Strictly adhering to the provided QA Guidelines, the agent treats the CV as the ground truth. It penalizes internal inconsistencies (e.g., overlapping dates) or mismatched titles by deducting specific points, rather than outright rejecting the CV.

3. Verification Results

The agent successfully processed all 5 sample CVs. By effectively handling edge cases and fuzzy matching, the system yielded the following predictions:

- **Predicted Scores:** [0.6, 1.0, 0.5, 0.2, 0.25]
- **Evaluation:** Using a 0.5 threshold, the system achieved an accuracy of **80.00%** (4 out of 5 correct predictions), successfully identifying fabricated or highly inconsistent profiles.

(Screenshot of the Part 1 Execution Log & Evaluation Output)

```
🔍 Processing CV_1.pdf...
   👉 Tool Call: search_linkedin_people args={'q': 'John Smith', 'location': 'Singapore'}
   👉 Tool Call: get_linkedin_profile args={'person_id': 9}
   ✅ Final Score for CV_1.pdf: 0.6

🔍 Processing CV_2.pdf...
   👉 Tool Call: search_linkedin_people args={'q': 'Minh Pham', 'location': 'Hong Kong'}
   👉 Tool Call: search_linkedin_people args={'location': 'Beijing, China', 'q': 'Minh Pham'}
   👉 Tool Call: search_linkedin_people args={'q': 'Minh Pham', 'fuzzy': True}
   👉 Tool Call: get_linkedin_profile args={'person_id': 47}
   ✅ Final Score for CV_2.pdf: 1.0

🔍 Processing CV_3.pdf...
   👉 Tool Call: search_linkedin_people args={'q': 'Wei Zhang', 'limit': 1}
   👉 Tool Call: get_linkedin_profile args={'person_id': 24}
   👉 Tool Call: search_linkedin_people args={'location': 'Munich', 'limit': 1, 'fuzzy': True, 'q': 'Wei Zhang'}
   👉 Tool Call: get_linkedin_profile args={'person_id': 97}
   ✅ Final Score for CV_3.pdf: 0.5

🔍 Processing CV_4.pdf...
   👉 Tool Call: search_linkedin_people args={'q': 'Rahul Sharma', 'location': 'Singapore', 'limit': 1}
   👉 Tool Call: search_linkedin_people args={'q': 'Rahul Sharma', 'fuzzy': True, 'limit': 1}
   👉 Tool Call: search_linkedin_people args={'limit': 1, 'q': 'Rahul Sharma Legal', 'fuzzy': True}
   👉 Tool Call: get_linkedin_profile args={'person_id': 3757}
   👉 Tool Call: search_facebook_users args={'q': 'Rahul Sharma', 'limit': 1}
   ✅ Final Score for CV_4.pdf: 0.2

🔍 Processing CV_5.pdf...
   👉 Tool Call: search_linkedin_people args={'q': 'Rahul Sharma', 'fuzzy': True}
   👉 Tool Call: get_linkedin_profile args={'person_id': 95}
   ✅ Final Score for CV_5.pdf: 0.25

All Predicted Scores: [0.6, 1.0, 0.5, 0.2, 0.25]

Evaluation Report:
{'decisions': [1, 1, 0, 0, 0], 'correct': 4, 'total': 5, 'final_score': 0.8}
```

Part 2: Autonomous Moltbook Social Agent

1. System Architecture & Design Decisions

The Social Agent is designed to interact seamlessly with the Moltbook REST API. The agent's identity was secured by hashing the student ID via MD5 (generating nickname_9837) and utilizing a persistent requests.Session injected with a Bearer Token for authorization.

The core architectural highlight is the agent's **Autonomous Exception Handling & CAPTCHA Resolution** capability. A specialized tool set was created using LangChain's @tool decorators to wrap the API endpoints, enabling the LLM to execute actions directly.

2. Agent Workflow & Tool Usage Strategy

The mission workflow requires the agent to sequentially execute authenticate_user, subscribe_submolt, upvote_post, and comment_post.

The CAPTCHA Challenge & Resolution Strategy: Moltbook implements an anti-bot verification step (CAPTCHA) that returns a "verification_required": true flag accompanied by an obfuscated math word problem (e.g., *"...lOoobsssster StOcKs+ TwEnTy ThReE] nEeU rOnS - aNd < gAiNs > FiVe..."*). To bypass this autonomously, the agent was engineered with:

1. **Instruction Prompting:** The agent was explicitly instructed to clean the obfuscated text, deduce the math logic, and calculate the final result formatted to two decimal places.
2. **Verification Tool (verify_action):** A custom tool was provided for the agent to submit the verification_code and its computed answer. Upon encountering the verification block during the commenting step, the agent successfully parsed the challenge, calculated the correct float value, invoked the verify_action tool, and ultimately published the comment without human intervention.

3. Execution Logs & Results

The agent completed the entire sequence flawlessly. It recognized its subscription status, correctly toggled the upvote, submitted a constructive comment ("AI Agents in Finance are truly revolutionizing the industry."), and successfully solved the math verification challenge to finalize the publication.

(Screenshot of the Part 2 Mission Execution Log showing the Verify_action tool success)

(Screenshot of the Moltbook Profile / Comment on the Web Platform)