FTEC5660 Homework2 Part1

Shi Zhan
Student ID: 1155209445

February 19, 2026

1 System Architecture and Design Decisions

1.1 High-Level Architecture

The solution is built as an Agentic AI system that automates the 'Know Your Customer' (KYC) workflow for verifying curriculum vitae (CV) claims against public social media data. The architecture consists of three main layers:

1. **Orchestration Layer (LangChain):** The core controller uses `LangChain`'s `AgentExecutor` to manage the reasoning loop. I utilize a **ReAct (Reason + Act)** paradigm, where the Large Language Model (LLM) iteratively generates thoughts, selects tools, observes outputs, and refines its verification strategy.
2. **Reasoning Engine (LLM):** The system leverages DeepSeek, a tool-enabled LLM. The model is responsible for parsing unstructured CV text, formulating search queries, interpreting JSON responses from the social graph, and calculating a final trust score.
3. **Tool Interface (MCP Client):** Connectivity to external data is handled via the **Model Context Protocol (MCP)**. The `langchain_mcp_adapters` library connects to the provided `SocialGraph` server, exposing six specific tools for LinkedIn and Facebook data retrieval.

1.2 Key Design Decisions

1.2.1 Asynchronous Execution for MCP Tools

Problem: The standard 'invoke()' method in LangChain is synchronous. However, the MCP tools provided are network-bound and implemented asynchronously. Initial tests resulted in 'StructuredTool does not support sync invocation' errors.

Decision: The system was designed using Python's `asyncio` library. I implemented the verification workflow using 'await agent.ainvoke()'. This ensures non-blocking calls to the MCP server, improving stability and performance when processing multiple CVs.

1.2.2 Robust System Prompt Engineering

Problem: LLMs can 'hallucinate' tool arguments. For example, during testing, the model attempted to pass a 'location' argument to `search_facebook_users`, which caused a validation error because that specific tool only accepts 'q', 'limit', and 'fuzzy'.

Decision: A 'Critical Tool Rules' section was added to the system prompt. This explicitly constrains the model:

- `search_facebook_users`: *ONLY accepts arguments q, limit, and fuzzy.*
- *NEVER use location or city as arguments for this tool.*

This prompt engineering effectively stopped the argument validation errors.

1.2.3 Fuzzy Matching Strategy

Decision: The prompt instructs the agent not to give up if an exact name match fails. It is explicitly directed to retry using the ‘fuzzy=True’ parameter and to look for the ‘most similar’ profile. This mimics human investigator behavior where minor typos or name variations (e.g., ‘Alx’ vs ‘Alex’) should not result in immediate rejection.

2 Agent Workflow and Tool Usage Strategy

The agent follows a structured investigation workflow defined in the system prompt.

2.1 Workflow Steps

1. **Analysis Phase:** The agent reads the raw text extracted from the CV PDF to identify key entities: Candidate Name, Current Company, University, and Location.

2. **Primary Search (LinkedIn):**

   - **Tool:** `search_linkedin_people`
   - **Strategy:** The agent queries using the candidate’s name and location. If the result list is empty, it retries with broader criteria (e.g., removing location filter or enabling fuzzy search).

3. **Deep Dive (Profile Verification):**

   - **Tool:** `get_linkedin_profile`
   - **Strategy:** Once a candidate ID is found, the full profile is retrieved. The agent compares the ‘Experience’ and ‘Education’ sections JSON against the CV text. It checks for discrepancies in dates, job titles, and degree types.

4. **Secondary Cross-Check (Facebook):**

   - **Tool:** `search_facebook_users`
   - **Strategy:** This is used as a fallback or confirmatory step, particularly to verify location or identity if the LinkedIn data is sparse.

5. **Scoring and Reporting:** The agent calculates a final confidence score (0.0 to 1.0) based on defined rubrics:

   - **1.0:** Perfect match.
   - **0.7–0.9:** Minor discrepancies (e.g., dates off by 1 year).
   - **<0.5:** Major fabrication (fake degree, non-existent job).

2.2 Prompt Implementation

The core logic is encapsulated in the following system prompt structure:

```
system_prompt = ’’’
You are an expert Background Verification Agent...
### CRITICAL TOOL RULES:
1. search_facebook_users: ONLY accepts ‘q‘, ‘limit‘, ‘fuzzy‘.
   NEVER use ‘location‘...

### TOOLS USAGE STRATEGY:
- Search LinkedIn (Primary).
- If multiple candidates appear, use Industry/Education to pick match.
- Compare Education (University, Degree).
- Note: Small date discrepancies are minor issues.

### OUTPUT FORMAT:
Must end response with: FINAL_SCORE: <float>
’’’
```

Listing 1: System Prompt Excerpt

3 Sample Verification Results

3.1 Results of the Sample CVs

The system was tested on 5 sample CVs. Below are the verification results based on the agent’s reasoning process.

Table 1: Verification Results for Sample CVs

CV ID	Candidate Name	Agent Analysis / Discrepancy Found	Final Score
CV_1	John Smith	**Match.** Profile found on LinkedIn matching ‘ByteDance’ and ‘McGill University’. Dates align closely.	0.95
CV_2	Minh Pham	**Match.** Profile confirms ‘Manager at BCG’ and previous experience at Tencent. Education at HKU confirmed.	1.0
CV_3	Wei Zhang	**Match.** Found ‘Engineer at PwC’ profile. Education and skills align with CV claims.	0.85
CV_4	Rahul Sharma	**Major Discrepancy.** CV claims ”Senior Engineer (2021-2027)” while simultaneously being a ”Consultant” and doing a PhD. Locations ”Singapore/Philippines” are suspicious. No consistent social profile supports these overlapping timelines.	0.0
CV_5	Rahul Sharma	**Mismatch/Fabrication.** While the name matches CV_5, this CV claims ‘AI Professional’ at EY. The agent identified inconsistencies in the timeline compared to available ground truth profiles.	0.2

Using the evaluation method in the provided Jupyter Notebook to evaluate the performance of the agent, the output is:

```
{’decisions’: [1, 1, 1, 0, 0], ’correct’: 5, ’total’: 5, ’final_score’: 1.0}
```

Listing 2: Final Evaluation

which means the agent performed well on the given CVs.

3.2 Analysis of Results

- **Robustness:** The agent successfully handled common names (e.g., ‘John Smith’) by filtering results based on the employer (‘ByteDance’) found in the CV text.
- **Fraud Detection:** For CV_4, the agent correctly identified internal logical errors (working dates in the future ‘2027’) and lack of social proof, assigning a low trust score.
- **Format Compliance:** All outputs adhered to the ‘FINAL_SCORE: float’ format, allowing for automated parsing and evaluation.