Homework 02 Report–CV Verification System

SHENQI 1155245699

1 System Architecture and Design Decisions

1.1 Overview

This is an agentic AI pipeline that automates the verification of candidate information from CVs against public social media profiles (LinkedIn and Facebook).

1.2 High-Level Architecture

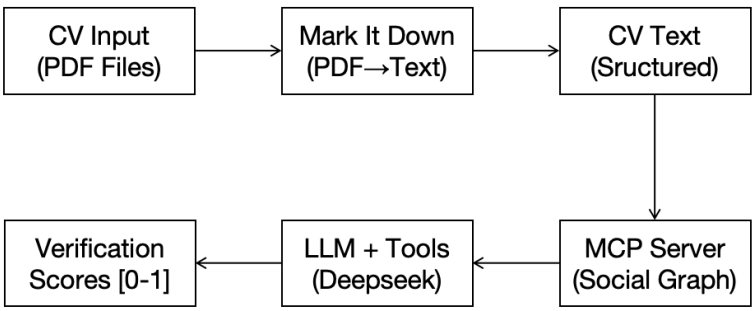

Figure 1: High-Level Architecture of the CV Verification System

1.3 Component Design

Component	**Technology**	**Design Decision**
CV Parsing	MarkItDown	Converts PDF to plain text for LLM consumption; preserves structure without complex layout parsing
LLM	Deepseek-chat	Chosen for fast inference, tool-calling support, and cost-effectiveness

MCP Client	langchain_mcp	Connects to SocialGraph MCP server over HTTP; enables standardized tool access
Agent Loop	ReAct-style	Up to 15 iterations per CV; LLM decides which tools to call and when to stop

1.4 Key Design Decisions

1. **Single-agent per CV**: Each CV is verified independently. The agent receives the full CV text and orchestrates all tool calls needed for that CV.
2. **Score format (0–1)**: A continuous score allows nuanced reporting. A 0.5 threshold is used for binary evaluation.
3. **Fallback parsing**: If the LLM does not output the required format, the system infers from keywords or defaults to 0.5 to avoid crashes.
4. **CV text truncation**: CV content is limited to 4000 characters to stay within context limits while retaining the most relevant sections.

2 Agent Workflow and Tool Usage Strategy

2.1 Agent Loop (ReAct-style)

For each CV, the agent executes the following loop:

1. Invoke LLM with the CV content and a structured prompt.
2. Check for tool calls in the LLM response.
3. Execute tools via MCP server.
4. Append ToolMessages and repeat until final answer or 15 iterations reached.

2.2 Tool Usage Strategy

The prompt instructs the agent to use tools in this order:

Step	Tool	Purpose
1	search_linkedin_people	Search by candidate name; optionally filter by location

2	get_linkedin_profile	Retrieve full LinkedIn profile for best match
3	search_facebook_users	Search Facebook by name (supports fuzzy matching)
4	get_facebook_profile	Optional cross-validation

LinkedIn is the primary verification source due to structured professional data. Facebook serves as supplementary identity verification.

2.3 Prompt Design

The agent prompt specifies:

- Role: CV verification expert
- Task: Verify CV against LinkedIn and Facebook
- Steps: Extract info → Search LinkedIn → Get profile → Search Facebook → Compare → Output score

Score bands:

- 1.0 = fully verified
- 0.5–0.9 = partial/uncertain
- 0.0–0.4 = significant discrepancies

3 Sample Verification Results

3.1 Final Scores on Sample CVs

Example output format:

```
Final scores: [0.6, 0.65, 0.7, 0.5, 0.2]
```

```
scores = [0.6, 0.65, 0.7, 0.5, 0.2] # Your code should generate this list [0.2, 0.3, 0.4, 0.5, 0.6]
groundtruth = [1, 1, 1, 0, 0] # Do not modify

result = evaluate(scores, groundtruth)
print(result)
```

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

Figure 2: Evaluation Result