FTEC5660 Homework 02 (Part 1) Report

Dai jiaxin 1155241401

Section 1: System Architecture and Design Decisions

The CV Verification System is built using a highly decoupled architecture integrating LangChain, Model Context Protocol (MCP), and the Gemini-2.5-Flash Large Language Model.

Key Design Decisions:

Resilient Tool Invocation: Interacting with external MCP tools (social graph endpoints) is inherently unstable. To address this, an asynchronous wrapper (_call_tool) was designed with timeout protection (10s) and exponential backoff retries for network-related errors. Concurrency was strictly managed using asyncio.Semaphore to avoid triggering API rate limits.

Strict Scoring Policy via Prompt Engineering: Initially, LLMs tend to over-penalize common names when they retrieve mismatched profiles. To counter this, the cognitive prompt was engineered with a strict, logical policy distinguishing between a "Search Miss" (ambiguous/sparse CV with unrelated profiles = Innocent/Pass) and "Proven Fraud" (synthetic profiles containing impossible chronological facts = Fail).

Robust JSON Extraction: Instead of relying on the LLM to natively output pure JSON without markdown wrappers, a regex-based strict JSON extractor was implemented to capture the evidence payload perfectly during every execution round.

Section 2: Agent Workflow and Tool Usage Strategy

The verification pipeline follows a multi-stage sequential workflow:

Extraction & Parsing: The system uses markitdown and PyPDF2 to extract raw text from the CV PDFs. A heuristic parser then identifies core claims: Candidate Name, Education details, Experience strings, and Date ranges.

Search Strategy (Candidate Discovery), Tools Used: search_linkedin_people and search_facebook_users.

Strategy: The agent sanitizes the candidate's name and queries the MCP server with fuzzy matching enabled. It deduplicates IDs and limits candidates per source to prevent context overflow.

Fetch Strategy (Profile Deep-Dive):

Tools Used: get_linkedin_profile and get_facebook_profile.

Strategy: The system fetches detailed profiles using the discovered IDs. A local caching mechanism (_PROFILE_CACHE) is implemented to prevent redundant network requests for overlapping candidate IDs.

Hard-Fail Detection Layer: Before invoking the LLM, the system performs a deterministic check for obvious red flags, such as future dates (e.g., year 2027) or extreme name mismatches.

LLM Judgment: An evidence pack containing overlapping terms and detected contradictions is sent to the LLM. The agent applies the defined scoring policy to return a final confidence score, rationale, and findings list.

Section 3: Sample Verification Results

The pipeline successfully verified the 5 provided sample CVs, achieving a 100% accuracy rate (5/5) against the ground truth labels [1, 1, 1, 0, 0].

CV_1.pdf (John Smith): Score 0.95 (PASS)

Rationale: LinkedIn profile exactly matches the BSc Marketing degree from McGill University (2009) and the experience start year (2020). Minor profile glitches were bypassed by the matching policy.

CV_2.pdf (Minh Pham): Score 0.95 (PASS)

Rationale: Strong match. Education (BSc Design, HKU) and multiple professional milestones (Tencent 2013, BCG 2022) were fully corroborated by a fetched LinkedIn profile.

CV_3.pdf (Wei Zhang): Score 0.60 (PASS)

Rationale: Evaluated as a "Search Miss". The CV is sparse, and the fetched profiles clearly belong to different individuals (KAIST/NTU vs. Univ. of Tokyo). Per the lenient baseline policy, the lack of verifiable data without hard contradiction results in a passing baseline.

CV_4.pdf (Rahul Sharma): Score 0.20 (FAIL)

Rationale: Hard fail triggered by the deterministic layer. The CV contains an implausible future-dated timeline (e.g., working until 2027).

CV_5.pdf (Rahul Sharma): Score 0.15 (FAIL)

Rationale: Proven Fraud. Not only does the claimed "PhD in AI" contradict all fetched profiles, but the associated social profiles contain fabricated future dates (e.g., end year 2025), indicating a synthetic identity cluster.