FTEC5660 · Agentic AI for Business and FinTech

CV Verification System

Individual Homework 02 — Report

Name	Xu Zhuocheng
Student ID	1155244383
Due Date	27 February, Midnight

1. System Architecture and Design Decisions

The CV Verification System is a multi-agent pipeline that automates KYC-style background checks by cross-referencing CV claims against public LinkedIn and Facebook profiles. The system leverages the LangChain/LangGraph ecosystem, a ReAct agent pattern, and six MCP tools exposed by the SocialGraph server.

1.1 High-Level Architecture

Layer	Component	Technology
Input	PDF parsing	MarkItDown (markitdown[pdf])
Orchestration	ReAct Agent loop	LangGraph create_react_agent
LLM	Reasoning & analysis	Google Gemini 2.5 Pro
Tools	Social media queries	SocialGraph MCP Server (6 tools)
Output	Score + discrepancy list	JSON block parsed by regex

Table 1. System layer breakdown.

1.2 Design Decisions

- **ReAct agent pattern** — allows the LLM to autonomously decide which tools to call and in what order, adapting to partial or ambiguous CV data without hardcoded branching logic.
- **MCP (Model Context Protocol) over direct API calls** — the MCP adapter abstracts transport details and lets the agent discover tools at runtime, making the system extensible.
- **Fuzzy search enabled by default** — CV names often contain romanisation differences; fuzzy matching reduces missed matches.
- **Best-match, no-rejection policy** — following Q&A guidance, the agent always selects the most similar social profile rather than aborting on no exact match.
- **CV content treated as ground truth** — discrepancies are measured relative to social media data; internal CV inconsistencies are flagged separately.
- **Structured JSON output** — score + discrepancies list enables programmatic evaluation via evaluate() without manual inspection.

2. Agent Workflow and Tool Usage Strategy

2.1 End-to-End Workflow

Each CV is processed independently through the following sequential pipeline:

Step 1 — PDF Parsing	All five CV PDFs are loaded via MarkItDown, converting each to plain-text Markdown stored in all_cvs.
Step 2 — CV Extraction (LLM)	The agent extracts: candidate name, location(s), job title(s), company names, date ranges, education, degree, graduation year, and skills.
Step 3 — LinkedIn Search	Tool: search_linkedin_people. Searches by name + optional location. Agent selects the best-matching result by comparing name, headline, and location.
Step 4 — LinkedIn Profile Retrieval	Tool: get_linkedin_profile. Retrieves full work history, education records, skills with proficiency, and current employment status.
Step 5 — Facebook Search	Tool: search_facebook_users. Parallel search by name with fuzzy matching to handle display-name variations.
Step 6 — Facebook Profile Retrieval	Tool: get_facebook_profile. Retrieves current job, company, city, country, education level, and recent posts.
Step 7 — Discrepancy Analysis (LLM)	Compares CV fields against LinkedIn and Facebook data across 7 dimensions: job title, company, dates, education, location, skills, and internal consistency.
Step 8 — Score & JSON Output	Assigns a reliability score in [0,1] and outputs {"score": ..., "discrepancies": [...]} parsed by regex into the evaluation pipeline.

2.2 Tool Call Summary

#	Tool	Purpose	When Called
1	search_linkedin_people	Find candidate LinkedIn ID	Once per CV
2	get_linkedin_profile	Full work/edu/skills history	Once per CV
3	search_facebook_users	Find candidate Facebook ID	Once per CV
4	get_facebook_profile	Current job, company, location	Once per CV
5	get_linkedin_interactions	Network engagement check	Optional
6	get_facebook_mutual_friends	Verify social connections	Optional

Table 2. MCP tool usage strategy.

3. Sample Verification Results

The agent was run on the five provided sample CVs. Scores below 0.5 trigger a FAIL decision in evaluate().

CV	Score	Decision	# Issues
CV_1.pdf	0.20	FAIL	5
CV_2.pdf	0.75	PASS	1
CV_3.pdf	0.60	PASS	2
CV_4.pdf	0.15	FAIL	6
CV_5.pdf	0.25	FAIL	4

Table 3. Final scores and decisions for all 5 sample CVs.

CV_1.pdf — John Smith

Score: **0.20 · FAIL**

- Internal CV inconsistency: headline 'Marketing Professional' with marketing skills conflicts with the only listed role 'Engineer'.
- Employment status: CV claims ByteDance role is current (2020-Present); LinkedIn shows it ended and status is 'student'.
- Job title mismatch: Facebook profile lists current job as 'Scientist', not 'Engineer'.
- Company mismatch: Facebook profile lists current employer as 'Traveloka', not 'ByteDance'.
- Location ambiguity: CV lists both Singapore and Kowloon (Hong Kong) with no clear primary location.

CV_2.pdf — Minh Pham

Score: **0.75 · PASS**

- Minor date gap: LinkedIn shows no record bridging the Tencent Analyst role ending in 2017 and BCG Manager role starting in 2022 — a 5-year unexplained gap.

CV_3.pdf — Wei Zhang

Score: **0.60 · PASS**

- Degree field mismatch: CV claims 'BSc in Consulting' from University of Tokyo; LinkedIn records a different field of study.
- Job title discrepancy: CV states 'Engineer' at PwC; LinkedIn headline describes a consulting-focused role.

CV_4.pdf — Rahul Sharma

Score: **0.15 · FAIL**

- Future employment date: Senior Engineer at Microsoft listed as 2021-2027; 2027 is in the future.
- Overlapping roles: Senior Engineer (2021-2027) and Consultant (2020-2023) overlap by 2 years.
- Degree-profession mismatch: PhD in Legal Studies, but all experience is in engineering/quantum computing.
- Skills mismatch: CV mixes 'Compliance, Litigation' with 'Web3, Machine Learning, Quantum Computing' — incoherent profile.
- Invalid location: 'Singapore / Philippines' is not a standard single location.
- No social profile found matching both legal PhD and engineering background simultaneously.

CV_5.pdf — Rahul Sharma

Score: **0.25 · FAIL**

- Overlapping employment: Senior Analyst at DataForge (2016-Present) and Lead Scientist at UrbanFlow (2010-2017) overlap.
- Overlapping employment: Senior Engineer at EY (Current) and Senior Analyst at DataForge (Present) are simultaneous full-time roles.
- Date inconsistency: Multiple concurrent positions create an implausible work history.
- Education timeline: PhD completed in 2012 potentially conflicts with UrbanFlow start date of 2010.

4. Evaluation Results

After running the agent on all 5 CVs, scores were passed to evaluate() with ground-truth labels [1, 1, 1, 0, 0].

4.1 evaluate() Output

```
scores = [0.20, 0.75, 0.60, 0.15, 0.25]
```

```
{'decisions': [0, 1, 1, 0, 0], 'correct': 4, 'total': 5, 'final_score': 0.80}
```

CV	Score	Prediction (>0.5=1)	Ground Truth	Correct?
CV_1.pdf	0.20	0 (FAIL)	1	x
CV_2.pdf	0.75	1 (PASS)	1	v
CV_3.pdf	0.60	1 (PASS)	1	v
CV_4.pdf	0.15	0 (FAIL)	0	v
CV_5.pdf	0.25	0 (FAIL)	0	v

Table 4. Per-CV evaluation results. Final score = 4/5 = 0.80.

5. Conclusion

The implemented CV Verification System successfully demonstrates an end-to-end agentic pipeline for automated KYC-style background checking. By combining PDF parsing, LLM-based reasoning, and real-time MCP tool calls to LinkedIn and Facebook data, the agent correctly identified 4 out of 5 sample CVs, achieving a final evaluation score of 0.80.

The single misclassification (CV_1, John Smith) occurred because the agent detected multiple genuine discrepancies and assigned a low score of 0.20, but the ground-truth label was 1 (valid). This suggests the ground-truth may incorporate CVs that appear inconsistent on the surface but are nonetheless labelled legitimate, pointing to a potential area for threshold tuning or more nuanced scoring logic.

Future improvements could include: (1) incorporating get_linkedin_interactions to verify professional network authenticity, (2) using get_facebook_mutual_friends to cross-validate claimed colleagues, and (3) applying a calibrated scoring model rather than a single LLM-assigned float to improve consistency across borderline cases.