CAMPUS AGENT / WEEKLY RESEARCH MEETING

把 Schema 與抽取流程變成可檢查的程式

115 資工專題 · 2026/08/07 Meeting

回應 07/31：MECE · 固定抽取 · 可驗證 evidence

先把「情境」定義清楚，再限制規則能寫什麼

依老師建議，先用 **人、事、時、地、物** 建立 `fact_schema.json`。

關鍵設計不是再加幾個欄位，而是訂一條約束：

> **規則的 condition 欄位，只能使用情境 schema 已定義的 key。**

這樣 MECE 不再只是人工閱讀時的感覺，而能變成 validator 可以檢查的規格。

情境 schema → 限制規則條件 → 判定時也只能讀同一組案件事實

抽取流程改成固定問題，而不是自由生成一整條規則

原本：

把一段法規交給 LLM → 請它自己整理成規則

改成：

程式固定問 6 類問題 → 每一格回答封閉選項 → 附 `evidence_span` → 程式驗證原文真的存在

其中 `evidence_span` 必須能逐字回到原始 PDF / 文字來源。

> **LLM 從「規則作者」降成「逐格標註者」；流程本身決定哪些問題一定要問。**

全量抽取已經被切成一張可以跑完的矩陣

當時規劃：

- **1,183 條**待抽取單位
- 每條固定問 **6 類問題**
- 共 **7,098 格**
- 先跑數十條確認格式與流程
- 全量 API 成本估計約 **US$10**

這次重點不是模型一次答得多漂亮，而是：

> **每一格有沒有問、答案能不能被驗證、哪一格需要人工複核，都能被記錄。**

上層判定流程也第一次接起來

使用者自由文字

↓ AI 只做語意轉換

案件 facts

↓ deterministic

`judge.py` **比對規則 / 取嚴**

↓

判定 + 依據 + 還缺的資訊

現場仍用「國科會 3 個月 vs 中央大學 45 天」展示，並補上資訊不足時反問使用者的雛形。

下一步原本想先用本地模型做便宜的流程健檢

計畫先用實驗室 GPU 上的本地模型：

- `qwen3:8b`
- `llama3.1:70b`

先確認：

- JSON schema 強制輸出是否能正常工作
- pipeline 不會卡在錯誤迴圈
- 再用 GPT API 跑正式全量

> 核心思路：**小模型先驗證流程健壯性，大模型再換品質。**

老師這次主要改的是「呈現」與「執行風險」

1. **UI 要先給結論**：判定、依據、還要做什麼先出現；完整解釋預設收合。
2. **本地 / 小模型先跑一輪**：先確保不會有 infinite loop 在燒 API 成本，再進正式抽取。
3. 整體方向持續成立；schema 本身這次沒有再被質疑。

> **系統正確性之外，使用者看不看得懂、願不願意用，開始成為同等重要的驗收條件。**