CAMPUS AGENT / WEEKLY RESEARCH MEETING

# Baseline 找得到規則，卻還不會判斷

**115 資工專題 · 2026/07/31 Meeting**

第一次把「找到規則」與「做出正確判定」拆開來看

# Demo：兩份規則都撈到了，答案還是錯

問題：計畫執行期滿後，什麼時候要完成結報？

| 方法 | 找到的資訊 | 最後結果 |
| --- | --- | --- |
| Baseline RAG | 國科會 3 個月 + 中央大學 45 天 | 直接採用較寬鬆的 **3 個月** |
| `judge.py` | 同樣兩條規則 | 比較後輸出 **45 天** |

> **Retrieval 成功，不代表 decision 成功。**

這是第一次用可以現場跑的程式，具體展示規則判定層和 baseline RAG 的差異。

# 當時的資料流程已經能跑，但還不夠受控

法規 PDF → 純文字 → 結構化 JSON → 人工複核 → 規則判定

當時最大的弱點在「PDF → 結構化規則」：

- 主要還是讓 AI 看內容後自行抓欄位。
- 有人工複核，但沒有一套保證每個面向都被檢查過的固定程序。
- 規則欄位是從現有文件「看到什麼就加什麼」，完整性仍靠人的直覺。

> **引擎可以是 deterministic，但如果進去的規則本身漏了條件，最後仍然會穩定地判錯。**

# 覆蓋率也開始暴露另一個問題：資訊不完整時怎麼辦？

測試題當時分成 A–E 類：

- A–D：大多已經有規則可判。
- **E 類：使用者情境不完整，需要系統反問。**

這類還沒有完成。

下一步不只是增加更多規則，而是讓系統知道：

> **目前缺哪個案件事實？這個事實會不會改變判定？如果會，就先問。**

# 老師的核心要求一：Schema 必須先做到 MECE

不能再用「PDF 裡出現什麼，就決定 schema 有什麼」。

老師建議改成 top-down：用 **人、事、時、地、物** 系統化檢查情境。

| 維度 | 要問的事 |
| --- | --- |
| **人** | 身分 / 角色會不會影響資格？ |
| **事** | 這筆支出屬於什麼事件 / 程序？ |
| **時** | 計畫年度、生效日、付款日是否重要？ |
| **地** | 國內外、校內外是否改變規則？ |
| **物** | 金額、科目、品項或文件是什麼？ |

> 空值可以存在，但「有沒有問過這個維度」不能靠運氣。

# 老師的核心要求二：抽取流程不能整包交給 LLM

老師區分了兩件事：

- `judge.py`：已經是固定規則邏輯。
- **PDF → 結構化規則**：仍太像讓 LLM 自由發揮。

要改成：

**程式決定每一步一定要檢查什麼 → LLM 只回答這一步 → 程式再驗證答案**

> 不是相信模型「應該有想到」，而是讓流程本身保證每個檢查都被執行。

# 這場之後的工作順序被重新排過

1. **先把 schema 與抽取流程打磨紮實。**
2. 做前端，讓老師與同學先試，找出模型和流程的漏洞。
3. 最後再找真正的系辦承辦人驗證，不要太早消耗對方時間。
4. 同時要確認「跨來源取嚴」這個核心假設在實務上是否真的成立。

> **下一週的任務：把 MECE 與「每一步一定執行」從口號變成可被程式檢查的設計。**