新舊版流程與 Schema 差異

115 資工專題 · 2026/10/06 Meeting

回答上週留下的核心問題：

自然語言進來後，系統到底憑什麼選到正確規則？

這次先釐清三件事

CORE QUESTION

自然語言進來後，系統到底憑什麼選到正確規則？

後面會先從舊版流程開始，再說明新版為什麼要拆出新的結構。

01

規則怎麼被選中？

從使用者一句話到 exact rule，中間經過哪些步驟？

02

LLM 到底負責哪裡？

哪些是模型理解，哪些是 static / deterministic 程式？

03

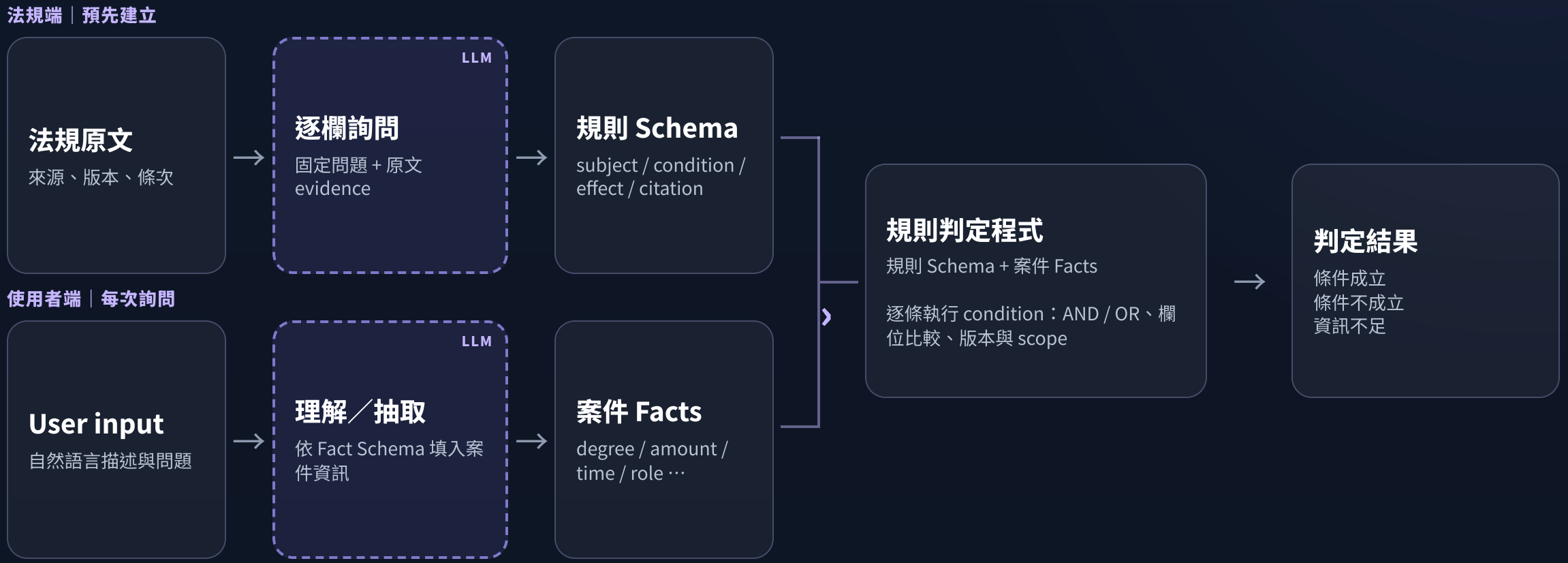
相似問題怎麼不配錯？

文字接近但法律語意不同時，不能只靠「最像」。

這頁先當導覽；後面會先從舊版流程開始，再說明新版為什麼要拆出新的結構。

新舊版共享同一個判定流程

法規與使用者輸入會分別整理成結構化資料，最後匯入同一套規則判定程式。



新舊版共同的底層：LLM 負責把兩側資料整理成結構化輸入，最後的規則條件由程式執行。

Rule Schema 與 Facts 怎麼接起來？

規則端定義「要檢查什麼」；案件端提供「這次的值」。兩邊使用同一套欄位名稱。

Rule Schema｜規則端

```
{
  "subject": {
    "role": ["博士班研究生"]
  },
  "condition": {
    "AND": [
      {
        "field": "object.degree_level",
        "operator": "eq",
        "value": "博士班"
      },
      {
        "field": "object.per_person_fee",
        "operator": "lte",
        "value": 2000
      }
    ]
  }
}
```

Facts｜案件端

```
{
  "object.degree_level": "博士班",
  "object.oral_exam_committee_count": 6,
  "object.per_person_fee": 2000
}
```

未提供的欄位維持 `null`／UNKNOWN。

舊版流程：兩次 LLM semantic decision

使用者原文
自然語言問題

→

LLM
選 topic
從 closed topic list 選最符合的主題

→

決定所需欄位
`required_facts_for_topic()`

→

LLM
抽取 facts
只填 fact schema 允許的欄位

→

Rule engine
逐條 deterministic 執行

舊版主要資料結構

`topic` + `facts`

關鍵風險

如果 topic / fact semantic mapping 先錯了，後面只是**很精確地執行錯的規則**。

新版：LLM 只提名證據，語意要逐層取得 authority

模型可以指出「哪段原文值得看」，但不能直接決定「因此應該套哪條規則」。

> **「這是國科會計畫，博士論文口試費的金額標準是多少？」**

LLM

① Evidence nomination

只提名 exact span + offset

博士論文口試費　金額標準

不輸出 topic、scope 或 verdict。

→

STATIC

② Semantic grammar

reviewed grammar 才把文字轉成 typed meaning：

口試費 → 論文口試

金額標準 → 金額問題

→

STATIC

③ Ownership / provenance

還要證明 observation：

- 來自這段原文
- 屬於目前 case
- 屬於目前 request

他案／引述不能直接借用。

→

STATIC

④ Static applicability

每個精確 scope 跑固定 predicate：

MATCH
NO_MATCH
UNKNOWN

真正的大改：LLM 可以說「這裡可能有證據」，但不能直接說「所以套這條規則」。

Finite grammar 從哪裡來？它又犧牲了什麼？

這些 mapping 不是 LLM 臨時發明，而是人工 review、版本化的有限語意文法。

同一句自然語言，怎麼變成 typed meaning？

> **「博士論文口試費的金額標準是多少？」**

博士	→	`degree_level = 博士班`
論文口試費	→	`target = THESIS_ORAL_DEFENSE`
金額標準	→	`question = AMOUNT_STANDARD`

grammar 不只列詞形，也定義角色、允許組合、順序與 guard；只有通過 reviewed pattern 的 mapping 才能取得 authority。

得到什麼？

可驗證性 ↑

可重播、可測試、可版本化

錯配較難無證據取得 authority

代價是什麼？

任意自然語言 coverage ↓

finite grammar 不可能窮舉所有 paraphrase；未支援的說法先停在 `UNKNOWN`。

下一個研究問題：能不能保留 LLM 的 semantic proposal／泛化能力，但不讓 proposal 本身直接取得 authority？

問題不在「大類」，而在精確法律問題

三句都屬於「論文口試費」大類；真正困難的是：使用者此刻到底在問哪一個 exact legal issue？

只是多提到「國科會」｜問題其實沒有變

「博士論文口試費的金額標準是多少？」

「這是國科會計畫，博士論文口試費的金額標準是多少？」

兩句都指向 → 博士口試費・金額標準

真的在問 funding｜exact scope 才改變

「國科會計畫經費能不能支付博士論文口試費？」

指向 → 國科會經費支付資格

36 / 36
大類 family
全部判對

→

31 / 36
exact scope
精確判對

→

5 / 5
應保持 scope 未決
模型卻全部多推一步

核心錯誤：「文字裡有提到」≠「使用者正在問」；family match 也不等於 exact legal applicability。

第一個修法：把 scope 判斷拆成 requirements

第一個假設：也許不是 LLM 不能判斷，而是「一次選 exact scope」太難。

我們先把任務拆細

① **LLM 不直接選 exact scope**

② **逐條判斷 requirement**
博士 / 口試 / 金額 / 人數 / funding

③ **Server 再組 scope**
requirements 全成立才建立

但原本的問題沒有消失

31 / 36 → 26 / 36
exact scope accuracy 反而下降

5 / 5
該停下來的案例，仍全部多推一步

問題不是 prompt 不夠細：requirement 的 semantic yes / no 仍由 LLM 拍板。

實驗後真正改的：把 semantic authority 往後移

真正要改的不是「怎麼問 LLM」，而是「LLM 的判斷能不能直接取得 authority」。

第一次修法

LLM 判斷：
「這個 requirement 成立嗎？」

Server 只負責組 scope；
semantic yes / no 仍由模型決定。

semantic authority 仍在 LLM

現在的做法

LLM 只說：
「這些原文片段可能有用。」

typed meaning / ownership / applicability
→ **reviewed / static proof**

proposal 本身沒有 legal authority

LLM 沒被拿掉：只是從 semantic decider 變成 evidence proposer。

Schema 為什麼要拆？

舊版 **topic + facts**　兩次 LLM 理解後，直接形成後段 rule engine 的結構化輸入

↓ 新版把前段原本容易混在一起的三種 semantic responsibility 分開

request_intent

使用者現在在問什麼？

```
target = THESIS_ORAL_DEFENSE
question = AMOUNT_STANDARD
```

有提到 2000 元 ≠ 正在問金額標準

scenario_facts

這個案件本身是什麼？

```
degree_level = 博士班
per_person_fee = 2000
```

案件裡的事實，**不直接決定 query intent**

ownership proof

這個值真的屬於誰？

```
value_proven
case_owner_proven
request_owner_proven
```

原文裡出現 ≠ 屬於目前 case / request

同一句話裡，「提到什麼」、「案件是什麼」、「正在問什麼」不能再視為同一個 semantic decision。

同一句話，新版怎麼避免配錯相似規則？

「這是國科會計畫，博士論文口試費的金額標準是多少？」

target：論文口試　degree：博士班　question：金額標準

博士口試費｜金額／人數標準

需要：論文口試 ＋ 博士班
＋ 金額或人數問題

✓ MATCH

國科會經費｜支付資格

需要：論文口試
＋「國科會經費能否支付？」

— NO_MATCH

不是選「最像哪條」；而是 每個 exact scope 各自證明必要條件
只出現「國科會」不會自動取得 funding authority。

最後：新版目前能保證什麼？

現在可以保證的邊界

- ✓ LLM 提議 不能直接變成具權威性的事實或適用範圍
- ✓ 「最相似」不能直接取得精確法律適用權威
- ✓ 證明不足就維持 UNKNOWN，不猜最接近的規則
- ✓ 值、案件與請求的歸屬分開證明

現在還不能保證

- — 任意自然語言都已完整覆蓋
- — LLM 不會漏掉應提名的原文證據
- — 新版端到端正確率已證明高於舊版
- — 185 條規則都已完成「法規原文 → 結構化規則」的語意等價證明

舊版：選定規則後才是確定性判定
新版：連「為什麼能選這條規則」也要有可審核、可重播的證據鏈

下一個研究問題：可驗證性與語意泛化怎麼兼得？

LLM 擁有語意判定權

自然語言泛化能力較強

但 ↓

相似語意可能被模型多推一步，
難以限制 semantic overreach。

Reviewed finite grammar

可審核、可重播、可 fail closed

但 ↓

自然語言 coverage 有限；
一直補 pattern 可能變成語句窮舉。

想討論的下一步

能不能保留 LLM 的 semantic proposal／泛化能力，
但 proposal 必須經過獨立驗證，才能取得 authority？

新的 failure 不應直接變成新的 pattern；先問它能否抽象成**可泛化、可審核的 semantic mechanism**。

下一步：先讓真實使用告訴我們哪裡會失敗

01

下週 Demo

先驗證現有 development prototype
是否已足以自然使用

→

02

小規模測試

內部 → 受控外部
收集真實問法與 failure

→

03

決定研究方向

從 failure taxonomy 判斷
下一個 bounded checkpoint

不是「錯一題就補一條 grammar」；先取得真實 failure distribution，再決定什麼值得固化。