

紡織成衣業大語言模型導入AI應用與成功案例解析

CEO & Co-Founder

Phill Kao Ph.D

MORALE AI



五大主題

1

主題一|RAG 與向量資料庫

幻覺問題、檢索→增強→生成、相似度搜尋=算距離、TextileGPT 知識庫實例

2

主題二|提示詞模組與提示工程

六大模組、CoT、提示工程 vs RAG vs Fine-tuning,加上 Trueway Agent 差/好 Prompt 實測與系統提示詞實驗

3

主題三|AI Agent 基礎

靜態模型 vs Agent、定義與核心特徵(自主性、工具使用、記憶)

4

主題四|Tools、MCP 與 Skills

M×N 整合難題、三大原語、SKILL.md 漸進式揭露、安全須知

5

主題五|Agent 框架與製造業落地 + 實證案例

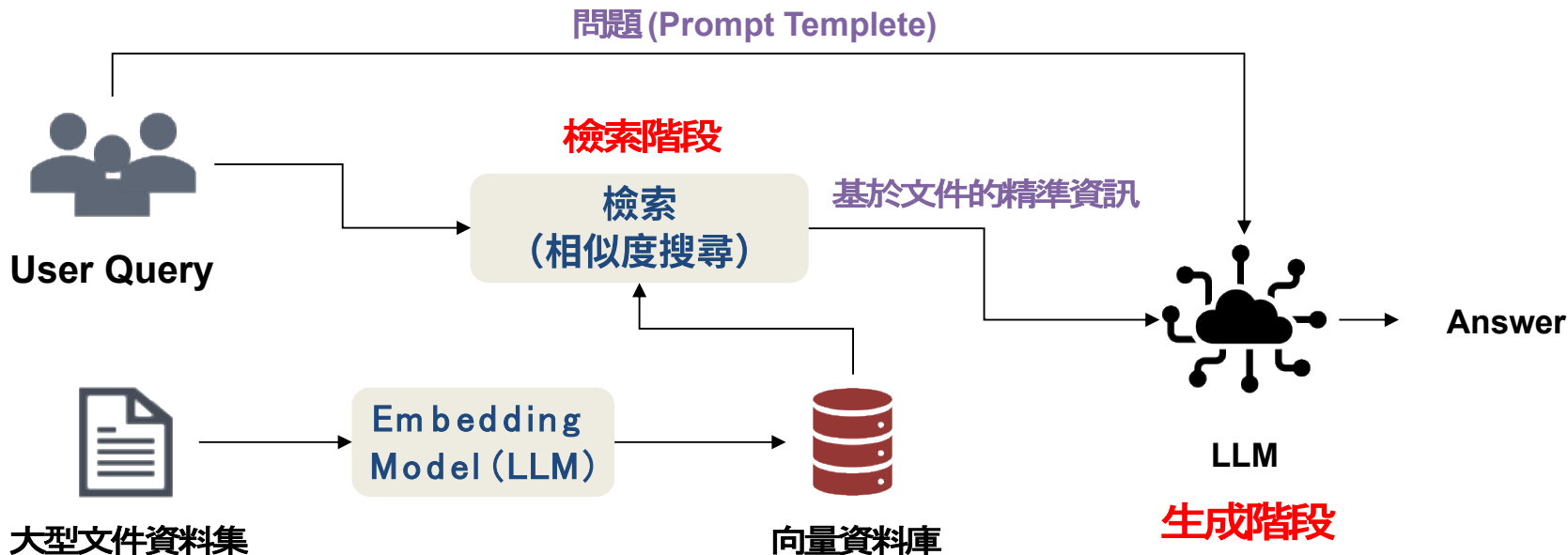
OpenClaw / NemoClaw / Hermes 定位、安全理念、學習迴圈、落地建議

RAG & VectorDB

生成式AI的問題 → 幻覺

RAG

- 在目前LLM應用中常有幻覺（hallucination）產出的問題，未必免LLM的產出錯誤， Retrieval-Augmented Generation(RAG)基於明確的大型文件資料先進行檢索，後再由LLM生成結果。
 - 檢索階段：利用檢索技術，從一個大型文件資料集合（如企業內部知識庫、技術手冊、報告等）中檢索與查詢相關的文档或片段。
 - 生成階段：利用LLM模型（如GPT-4、BERT等）將檢索到的資訊作為上下文，生成基於這些上下文的答案或內容。



相似度搜尋



INPUT DATA (IMAGES)

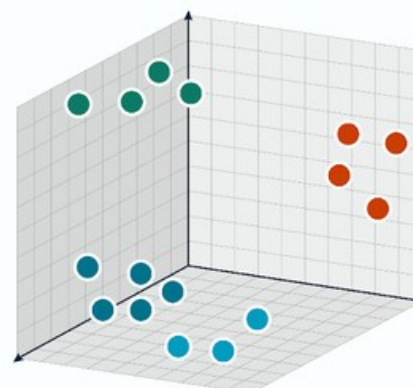


VECTOR EMBEDDING

```
[0.95, 0.49, 0.67, 0.75, 0.43, 0.00, 0.60, 0.13, 0.72, ...],
[0.73, 0.83, 0.80, 0.48, 0.55, 0.83, 0.86, 0.85, 0.06, ...],
[0.03, 0.71, 0.62, 0.07, 0.74, 0.30, 0.80, 0.08, 0.63, ...],
[0.49, 0.48, 0.38, 0.51, 0.45, 0.05, 0.10, 0.92, 0.29, ...],
[0.84, 0.38, 0.81, 0.00, 0.48, 0.61, 0.41, 0.85, 0.93, ...],
[0.24, 0.90, 0.69, 0.57, 0.43, 0.60, 0.37, 0.63, 0.28, ...],
[0.72, 0.33, 0.36, 0.98, 0.88, 0.05, 0.64, 0.91, 0.15, ...],
[0.89, 0.17, 0.44, 0.78, 0.33, 0.59, 0.29, 0.23, 0.22, ...],
[0.88, 0.75, 0.38, 0.77, 0.31, 0.98, 0.84, 0.79, 0.25, ...],
[0.11, 0.77, 0.30, 0.56, 0.62, 0.14, 0.89, 0.19, 0.62, ...],
[0.63, 0.41, 0.97, 0.73, 0.64, 0.41, 0.96, 0.36, 0.87, ...],
[0.51, 0.16, 0.26, 0.94, 0.53, 0.38, 0.39, 0.90, 0.55, ...],
[0.91, 0.99, 0.10, 0.87, 0.24, 0.91, 0.94, 0.36, 0.69, ...],
[0.89, 0.64, 0.14, 0.94, 0.51, 0.24, 0.61, 0.99, 0.11, ...],
[0.77, 0.65, 0.48, 0.94, 0.49, 1.00, 0.24, 0.10, 0.97, ...],
[0.58, 0.47, 0.01, 0.56, 0.59, 0.32, 0.79, 0.95, 0.40, ...]
```

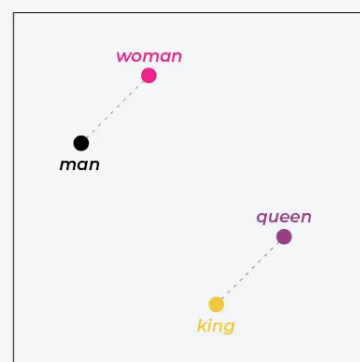


VECTOR DATABASE



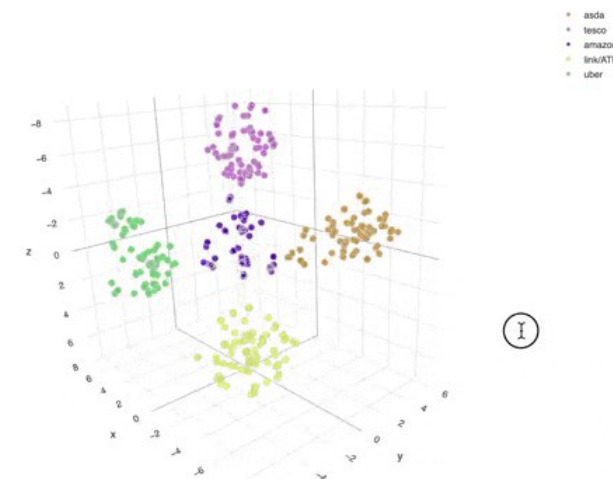
altexsoft

	living being	feline	human	gender	royalty	verb	plural
man →	0.6	-0.2	0.8	0.9	-0.1	-0.9	-0.7
woman →	0.7	0.3	0.8	-0.7	0.1	-0.5	-0.4
king →	0.5	-0.4	0.7	0.8	0.9	-0.7	-0.6
queen →	0.8	-0.1	0.8	-0.9	0.8	-0.5	-0.9
word	Word embedding						



Visualization of word embedding

相似度搜尋 = 算距離



生成式AI的問題 → 幻覺

● RAG



RAG (Retrieval-Augmented Generation, 檢索增強生成) 是一種讓AI變得更聰明的方法。

它不是讓AI「自己想」全部的內容，而是讓AI在「想答案」之前，先去找資料，然後根據找到的資料來回答問題。

就像是學生考試時，先去翻課本找線索，再寫答案，而不是亂猜！

1. 檢索 (Retrieval) → 從資料庫、知識庫、文件裡，找出跟問題有關的內容。
2. 增強 (Augment) → 把找到的內容附加在問題上，讓AI可以「看到更多線索」。
3. 生成 (Generation) → AI根據這些線索，寫出更正確、更有根據的答案。



AI 不是「自己想」全部內容,而是先找資料、再依資料回答

幻覺(Hallucination)

LLM 未經檢索時,依訓練資料「生成」答案 — 流暢、有自信,但可能是錯的。

紡織場景的代價:甲醛限值、AATCC 條款、RSL 適用範圍答錯,可能導致驗貨失敗或合規風險。

訓練資料還可能過時:去年的 RSL 年版、改版前的標準。

檢索 Retrieval

1

從知識庫、文件裡找出跟問題有關的內容(RSL 條目、標準章節、布種說明)

增強 Augment

2

把找到的內容附加在問題上,讓 AI「看到更多線索」

生成 Generation

3

AI 根據線索寫出更正確、有根據的答案,並標註來源

比喻:像考試先翻課本找線索再寫答案,而不是亂猜。



RAG 檢索的核心:把文字變成向量,語意近 = 距離近

向量化(Embedding)

文字 → 轉成一串數字(向量)→ 放進向量資料庫。

查詢時,把問題也轉成向量,計算與庫中各片段的「距離」—— 距離近就是語意相近。

紡織例:搜「防潑水布料」也能命中「撥水加工」文件 —— 不靠同義詞表,不靠人工標註,靠的是向量空間中的距離。

RAG 實務導入架構

1. 文件入庫:RSL、AATCC 標準、布種手冊 → 切塊(chunk)
2. 向量化:每個片段轉成向量,建立索引
3. 檢索:使用者問題向量化,找出最相近的 top-k 片段
4. 生成:LLM 以片段為依據作答,標註來源文件與頁碼

文件品質與切塊方式,直接決定檢索命中率 —— 垃圾進,垃圾出。

Prompt Engineering

提示詞條件



- **Role Module (角色模組)**

1. 功能是設定 AI 的專業身份與視角。
2. 例如：讓模型扮演顧問、工程師、醫師、法務、資料分析師等。
3. 作用是讓回答風格、思考方式、用字更貼近指定角色

- **Context Module (情境模組)**

1. 功能是提供精確背景、場景與邊界條件。
2. 例如：產業別、使用情境、公司背景、目標對象、資料來源。
3. 作用是避免模型脫離脈絡亂答，讓輸出更符合真實需求。

- **Constraints Module (限制模組)**

1. 功能是明確列出限制條件與禁止事項。
2. 例如：不能捏造資料、不能超出預算、不能使用某些格式、不能回答敏感內容。
3. 作用是降低錯誤、風險與不符合規範的輸出。

提示詞條件



- **Format Module (格式模組)**

1. 功能是規定輸出的結構與格式。
2. 例如：JSON、Markdown、表格、固定欄位、條列式。
3. 作用是讓輸出更容易被人閱讀，或直接被系統/API使用。

- **Few-Shot Module (少樣本示例模組)**

1. 功能是提供輸入與輸出的範例對齊。
2. 透過範例讓模型理解「你要的答案長什麼樣子」。
3. 作用是提升一致性，尤其適合格式嚴格或任務較複雜的情境。

- **Task Module (任務模組)**

1. 功能是定義核心指令與目標。
2. 通常用明確動詞開頭，例如：分析、整理、比較、翻譯、生成、萃取。
3. 作用是直接告訴模型「現在到底要做什麼」。

提示工程類型與介紹



技術	複雜度	Token 效率	最佳應用場景	失敗風險
Zero-Shot			簡單分類、常見翻譯	幻覺、格式不穩定
Few-Shot			嚴格格式輸出、模式識別	過度擬合範例
Chain-of-Thought (CoT)			數學計算、邏輯推演	思考鏈中斷／跑題
Tree of Thoughts (ToT)			創意寫作、多重路徑探索	運算成本過高
ReAct (Reason+Act)			代理工作流程、工具調用	陷入無限迴圈

複雜度／Token 效率以五級表示，橘色圓點越多代表程度越高。

Chain-of-Thought

STEP 1 | 任務拆解 Instructional Planning

先將複雜任務拆解為多個子問題

子問題 ①

子問題 ②

子問題 ③

STEP 2 | 逐步推理執行 Step-by-Step Execution

解 ① → 得到中間答案

- 上一步答案帶入下一題

- 先想再答，減少跳步

帶入 ② 接續推理

- 建議限定 3-5 步，避免跑題

- 步驟可在輸出時隱藏

彙整 ③ 產出最終答案

為什麼有效？

- 引導模型輸出推理過程，而非直接給出答案
- 適用數學計算、邏輯推演等多步驟任務
- 風險：思考鏈中斷或跑題，需限定步驟範圍

實例：3 分鐘路程，走路還是開車？

我要去洗車場洗車，走路 3 分鐘到，請推薦我走路 or 開車去

Thought for a couple of seconds

走路去。

因為只有 3 分鐘路程，開車反而有幾個缺點：

到洗車場前車又多沾一點灰，還要發動、停車、排位置，整體不一定比較省事。

走路去最輕鬆，洗完也不用煩惱怎麼把人跟車分開處理。

準確率提升高達 **61.4%**，冗餘 Token 減少 **13.9%**

為什麼 Prompt 很重要?



AI 不會讀心。

你沒說清楚的事, AI 只能:

- 自行猜測(可能猜錯方向)
- 給出籠統的通用答案
- 反問你,多耗一輪往返

模糊的 Prompt

「AATCC 22有改什麼?」→ AI 自己猜版本、猜範圍,答案冗長發散,最後還要反問你要什麼。

結構化的 Prompt

指定角色、範圍、格式與防護要求 → 一次得到可直接使用的表格與行動建議。

差 Prompt 的常見地雷



! 太模糊

「有改什麼?」「可以用嗎?」— 缺主詞、缺版本、缺對象, AI 只能全面撒網。

! 沒給範圍

不指定章節、品牌、產品類別, 回應就會又長又發散, 重點被稀釋。

! 沒指定格式

沒說要表格還是條列、要不要來源頁碼, 拿到的東西無法直接放進報告。

! 沒設防護

沒要求「查無資料要說明」, AI 可能在資料不足時給出看似肯定的答案。



案例一 / 差 Prompt 實測

Agent: 檢測技術標準改版前後比對助手

送出的 Prompt

「AATCC 22有改什麼？」

實際回應狀況

- AI 只能自行假設比對版本(2017e vs 2024)
- 沒有範圍 → 全部章節撒網: 安全條款、試片尺寸、hoop、校正... 重點分散
- 結尾反問: 「請告訴我你最關心的範圍」 → 還需要第二輪對話

回應節錄

「就目前命中的 AATCC 22...2017e vs 2024 片段/索引內容來看: 改版重點多為『安全注意事項章節的重整』與『章節重編號/文字編修』...其餘是否還有更多實質條件變更,需再補足 2024 原文片段才能全面確認。...

如果你想要『完整列出哪些章節是重編號、哪些是文字實質改動』,請告訴我你最關心的範圍(例如 Procedure/Report/Specimens/Apparatus)...



案例一 / 好 Prompt 這樣寫

同一個 Agent、全新對話,改用結構化 Prompt

你是紡織檢測實驗室的標準管理專員。

請比對 AATCC 22 2017e 版與 2024 版的差異,

範圍聚焦在 Procedure 與 Calibration 章節。

請以表格呈現,欄位包含:章節編號、2017e 內容、2024 內容、差異類型(實質變更/重編號)、對實驗室 SOP 的影響。

最後請給出 3 點實驗室應採取的行動建議,

若資料不足請明確標示。

角色

標準管理專員 → 回答以「實驗室要不要改 SOP」為導向

任務+範圍

鎖定兩個版本、兩個章節,不再全面撒網

格式

指定表格欄位 + 3 點行動建議,產出可直接使用

防護

要求標示資料不足,避免 AI 硬給結論



案例一 / 回應比較

同一個 Agent, 回應品質差在哪?

差 Prompt: 「AATCC 22有改什麼？」

- 版本、範圍全靠 AI 猜(自行假設 2017e vs 2024)
- 回應冗長, 安全條款、試片、校正混在一起, 焦點分散
- 格式由 AI 自訂, 無法直接貼進改版評估表
- 結尾反問使用者要範圍 → 至少要多一輪往返

好 Prompt: 角色 + 範圍 + 格式 + 防護

- 直接鎖定 Procedure 與 Calibration, 無關內容不進來
- 按指定欄位輸出表格: 章節、兩版內容、差異類型、SOP 影響
- 資料不足處明確標示「資料不足」, 不亂下結論
- 附 3 點可執行行動建議(校正不改條件、啟動原文核對、文件控制防呆) → 一次到位

案例一 / 實際問答畫面

TextileGPT 對話截圖(標註為本次觀察重點)

差 Prompt 對話

Prompt 只有一句:沒版本、沒範圍

AATCC 22有改什麼?

1) 結論

就目前命中的 AATCC 22 (Water Repellency: Spray) 2017e vs 2024 片段/索引內容來看: 改版重點多為「安全注意事項章節的重整 (新增/移位)」與「章節重編號/文字編修」; 而噴灑試驗儀的校正條件 (250 mL、27±1°C、25-30 s) 在命中證據中屬於實質不變。其餘是否還有更多實質條件變更, 需再補足 2024 原文片段才能全面確認。

AI 只能自行假設 2017e vs 2024、全章節撒網

判斷	範圍/章節	改版前依據 (2017e)	改版後依據 (2024)	影
實質		命中摘要顯示: 原有「Good	命中摘要顯示: 2024 出現/調	使 者 文 中 找 全 求 時

詢問任何問題

(結尾還反問「你最關心的範圍」→ 需第二輪)

好 Prompt 對話

角色+版本+範圍+表格欄位+防護,一次給齊

你是紡織檢測實驗室的標準管理專員。請比對 AATCC 22 2017e 版與 2024 版的差異, 範圍聚焦在 Procedure 與 Calibration 章節。請以表格呈現, 欄位包含: 章節編號、2017e 內容、2024 內容、差異類型(實質變更/重編號)、對實驗室 SOP 的影響。最後請給出 3 點實驗室應採取的行動建議, 若資料不足請明確標示。

1. 結論 (就目前命中的 context/片段)

就目前命中的 Calibration (校正) 章節片段 (Section 8.1/8.1.1), AATCC 22 2017e 與 2024 的校正條件沒有可確認的實質差異 (250 mL、27±1°C、25-30 s 皆一致); 但 Procedure (操作程序) 章節在本輪檢索中未取得足夠的 2017e/2024 兩邊原文對照, 因此除校正段落外, 其餘程序差異多只能先標示為「資料不足/可能重編號」, 需回查原文確認。

回應直接聚焦 Procedure / Calibration

2. 差異表 (聚焦 Procedure 與 Calibration)

章節編號	2017e 內容	2024 內容	差異類型(實質變更/重編號)	對實驗室 SOP 的影響
	兩版皆要求以 "250 mL distilled	兩版皆要求以 "250 mL distilled		SOP 的校正用水量/溫度條件可維持不

詢問任何問題

按指定欄位輸出差異表,可直接使用

MORALE AI

案例一 | 8



進階:系統提示詞(System Prompt)

使用者 Prompt 管「單次提問」,系統提示詞管「每一次對話」

是什麼?

Agent 管理者在「Agent 設定 → 系統提示詞設定」中定義的常駐指令,每一次對話都會自動套用。

定義 Agent 的角色、知識庫使用規則、回答結構與風格。

使用者不必每次重打「你是...請用表格...」— 把好 Prompt 的要素固化到 Agent 本身。

可在 Agent 編輯頁的「除錯與預覽」面板即時測試,並參考歷史對話驗證改版效果。

本次實驗(紡織專家 Agent)

測試問題(取自歷史對話):「推薦藍球衣適合的布料,並提供時間」

改版前:純文字回應,把保溫用鋪棉布也推薦給藍球衣,理由照抄知識庫;知識庫裡的布樣圖片完全沒用上,時間只回 UTC。

改版方式:直接修改「角色/系統提示詞」,加入【布料推薦規則】— 先分析用途機能需求、不符用途不可推薦、每項附推薦理由,並要求「圖文混排:每項布料插入知識庫中的布樣圖片」、時間以台灣時間呈現。

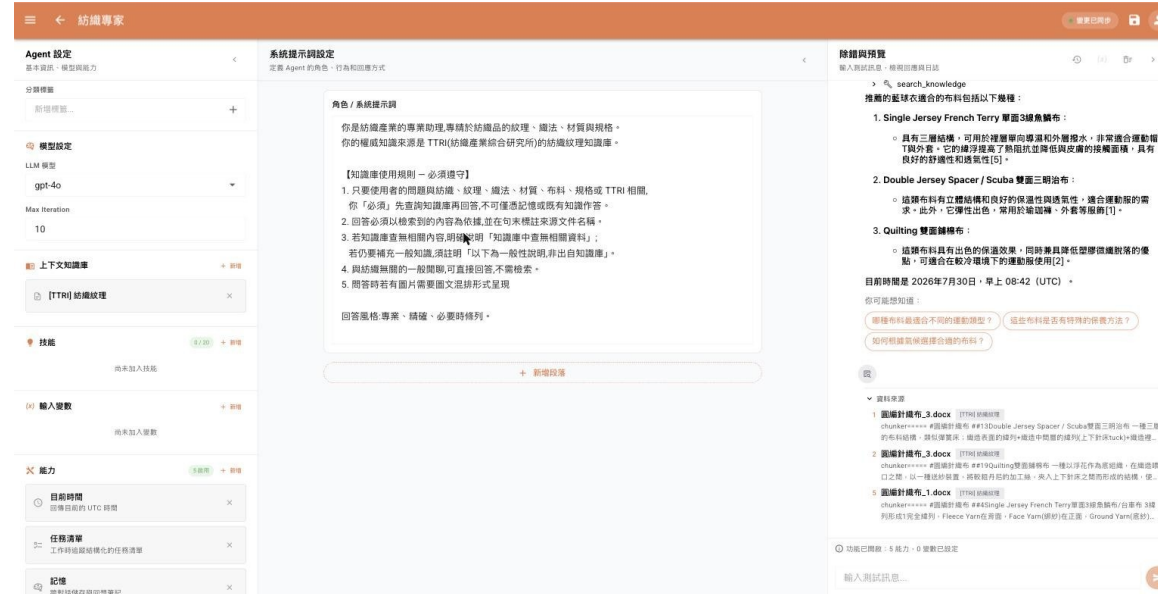
改版後:每項推薦布料下方直接嵌入實際布樣照片與織法圖,圖文混排一目了然;推薦理由對應籃球需求,時間為台灣時間。



系統提示詞 / 改版前後畫面

同一個問題「推薦藍球衣適合的布料,並提供時間」— 改的是 Agent 的系統提示詞

改版前(原系統提示詞):純文字



改版後(主提示詞加入【布料推薦規則】):圖文混排



- 純文字回應:知識庫裡的布樣圖片完全沒被用上
- 推薦「保溫用鋪棉布」給籃球衣 — 與用途矛盾,理由照抄知識庫
- 「提供時間」只回 UTC,沒轉換成使用者時區

- 图文混排:每項布料下方嵌入實際布樣照片與織法圖
- 推薦附「特點+理由」,對應籃球需求(支撐/彈性/透氣)
- 時間自動轉為台灣時間(UTC+8)呈現