

國立台北大學資訊工程學系專題報告

智動助手

Auto Assist

專題組員:邱顯崑、蔡明翰

專題編號:PRJ-NTPUCSIE-114-007

執行期間:114年9月至115年5月

1. 摘要

為簡化部分學校行政作業的流程，此專題使用 n8n 平台作為主要工具，實作了「計畫案查詢系統」和「每月郵件寄送系統」兩個系統，並對此二系統進行測試、評估和實際應用。未來期望能將 n8n(智慧自動化 workflow) 推廣至學校的其他部門中。

關鍵詞:系統開發專題、n8n 平台、自動化 workflow、計畫查詢、郵件寄送

2. 簡介

在學校的行政單位中，每日皆須處理大量的行政作業，而在我們的觀察中，有些作業擁有「重複性高」的特點，但是有些需要人為干預的部分無法使用傳統的自動化 workflow 來完成。為此，我們設計了搭配 AI 的自動化 workflow，嘗試藉由 AI 來替代需要人為干預的部分。

為了協助製作搭配 AI 的自動化 workflow，本專題將使用 n8n 平台作為主要工具，並從兩個實際應用例子，「計畫案查詢系統」和「每月郵件寄送系統」，來驗證 AI 搭配 workflow 的可能性和可行性。

※ 以上系統皆不會將個人姓名、Email 等可識別資訊傳送給雲端 AI

3. 專題進行方式

I. 工具介紹

本專題使用了 n8n 平台作為核心工具，其主要包括以下幾個特點：

- 開源：n8n 為開源平台，網際網路上的所有人皆可免費公開存取 n8n 平台的資源與內容。
- 支援自建：n8n 支援本地部屬，所有 workflow 流程和資料皆可在本地端執行完畢。相較於線上版本，不必擔心個人資料外流於外部伺服器。
- 視覺化流程圖：n8n 平台的自動化 workflow 有著視覺化的流程圖，方便對 workflow 進行修改和除錯。
- 支援多種應用工具：n8n 平台支援許多應用程式和 AI 工具的串接，利於對 workflow 的開發。
- 開發者友善：若平台未給予 workflow 所需的功能或工具，可自行編寫程式碼 (JavaScript、Python) 來滿足需求。

以上 n8n 平台的特點將利於我們製作接下來的實際應用例子，「計畫案查詢系統」與「每月郵件寄送系統」。

AI 模型：本專題使用 Gemini 2.5 Flash，以雲端 API 方式串接至 n8n 工作流，負責經費來源判斷、自然語言轉 SQL、查詢結果生成文字說明等需要語意理解的環節。

II. 計畫案查詢系統

A. 情境說明

每個月底的研發處會有一場進度會議，要報告產學計畫的執行進度與成果。而要整理出所需資料，需要經過以下步驟：

1. 請主計人員將資料從學校資料庫調出來
2. 將資料透過 Excel 做整理
3. 依學院和經費來源，逐筆分類
4. 針對問題交叉比對分析
5. 將資料整理為報表與圖表

根據過往經歷，處理完所有報表和圖表虛耗掉一整天，整個過程十分耗時耗力，並且難以讓每個環節都面面俱到。為此，我們製作「計畫案查詢系統」來改進並自動化此流程。

B. 系統架構

本系統由 n8n 工作流、本地 PostgreSQL 資料庫和 Python 組成，系統可拆分成 3 個部分，分別為本地資料庫更新、前端頁面與 n8n 主工作流。

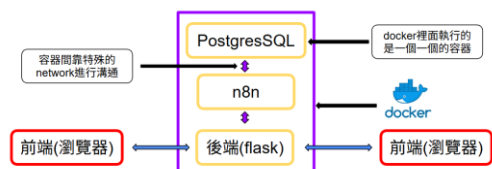


圖 1：系統主架構

1. 本地資料庫更新

首先，在進行資料查詢時，系統

會經過擷取、轉換和載入(ETL)的自動化工作流，該自動化工作流會在每次查詢時自動從學校資料庫提取最新資料、進行正規化，並比對資料是否已存在在本地 PostgreSQL 資料庫。若是，則將資料直接更新(Update)至本地資料庫；若否，則先將資料交由 AI 判斷其經費來源(政府、企業、其他)，再將資料輸入(Insert)進資料庫。確保每次查詢時資料與學校資料庫統一。



圖 2：同步學校資料庫子工作流

2. 前端頁面

在輸入的部分，我們使用 Python Flask 來實作前端頁面，提供使用者直覺的操作介面。

在此頁面中，使用者可輸入自然語言來查詢內容(支援語音輸入)，並且可選擇生成出的圖表顏色。送出查詢後，系統會執行一輪 n8n 主工作流，最後將結果輸出傳回前端頁面，顯示資料查詢結果、AI 回覆和圖表資訊等內容。



圖 3：前端頁面-查詢頁



圖 4：前端頁面-結果頁

3. n8n 主工作流

當使用者在前端頁面送出問題後，系統會觸發 n8n 的主工作流，依序完成以下處理：首先，工作流會即時擷取本地資料庫的資料表結構與少量樣本資料，組裝成「查詢上下文」，並一併注入領域知識(例如各學院的教師人數)，作為 AI 判讀的依據。接著，AI 會根據使用者意圖，生成 SQL query 與後續構建圖表所需的參數 [1][2][3]。其中，SQL query 會先經過安全驗證，僅允許單一條讀取 (SELECT) 語句、且只能查詢指定的資料表，任何更新或寫入(如 INSERT、UPDATE)指令都會被攔截，從源頭杜絕資料被竄改的風險。驗證通過後，本地資料庫會執行該 SQL 進行查詢並

取出資料。取得結果後，系統會先濾除主持人姓名等隱私欄位，再將資料交由另一個 AI 進行解讀，生成趨勢、異常與對比等文字洞察。最後，工作流會將圖表參數、查詢資料與 AI 文字回應三者合併，一併回傳前端頁面呈現。



圖 5：n8n 主工作流

C. 時程安排與系統測試

專題上半學年都在製作與增加資料查詢過程工作流的功能，以及對齊需求，包括增加判別不同學院的功能，自動圖表生成等，這時的系統尚未實際應用，所以測試的方式為輸入由行政人員提供給我們包含 111~114 年產學資料的試算表，將試算表上傳到我們系統來測試。透過確認輸出結果是否滿足需求以及比對實際資料來驗證正確性。

專題下半學年，我們著手與學校資料庫進行串接。首先，設計一套將資料同步到本地資料庫的工作流；接著，向學校申請產學計畫資料的存取權限。為了確保校方資料的安全，我們僅以「唯讀」方式讀取學校資料庫，再經過處理後存入本地資料庫。最後，實際驗證結果，確認資料確實成功接入。(以下圖片為本地資料庫的資料，證實資料確實同步學校資料庫)

id	project_code	project_name	exec_unit	
1702	10...	「阿...	亞太中心	行政院
1728	11...	石墨...	電機工程學系	千如電
1730	11...	電腦...	前廳科技中心	千如電
1732	11...	5G...	電子商務研究中心	千如電
1739	10...	東亞...	人文學院	Eurasi
1741	10...	10...	社會工作學系	新北市
1743	11...	新北...	自然資源與環境管理研究所	睦群股
1751	11...	20...	民意與選舉研究中心	財團法
1753	11...	20...	民意與選舉研究中心	財團法
1760	11...	住宅...	不動產與城鄉環境學系	國家住
1783	11...	20...	大數據中心	社團法
1784	11...	兒童...	大數據中心	后羿醫
1787	11...	「企...	全球變遷與永續科學研究中心	國泰世
1808	11...	醫療...	資訊工程學系	車庫創
1814	11...	多次...	大數據中心	財團法
1820	11...	中華...	東西哲學與詮釋學研究中心	中華亞
1821	11...	產業...	電機資訊學院	瑞仁生
1824	11...	ES...	企業永續中心	臺灣證
1845	11...	IM...	電機工程學系	宏匯民
1876	11...	我國...	自然資源與環境管理研究所	農業部
1878	11...	我國...	自然資源與環境管理研究所	農業部
1892	11...	中央...	都市計劃研究所	以樂工
1912	11...	「1...	不動產與城鄉環境學系	新竹縣

1,547 行 (0.005 秒) 編輯, Explain, 匯出

圖 6：本地資料庫的資料

D. 主要挑戰與解決方法

專題過程中，我們遇到了幾個關鍵困難。第一個是自動圖表生成。最初我們嘗試讓 AI 直接生成圖表，但很快發現兩個問題：一是每次生成的格式不固定，二是圖上的數據常與實際資料有落差。即使透過提示詞工程反覆調整，也無法根本解決。後來我們改變做法—讓 LLM 只負責生成圖表所需的參數，再由前端依參數渲染圖表。如此一來，資料的正確性由系統把關，圖表格式也得以固定。

第二個是學校資料庫的串接。原本打算直接連接學校資料庫，但校方對資料安全有所顧慮。因此我們重新設計了一套工作流，專門將學校資料庫同步到本地：對學校資料庫只做讀取的動作，而所有資料處理與查詢都在本地資料庫進行，徹底消除了安全性的疑慮。

III. 每月郵件寄送系統

A. 情境說明

每個月研發處皆須針對該月聘案

寄送對應郵件。該月聘案資料由學校資料庫而來，須從其中篩選出符合條件(聘案有效期限在該月以內、符合計畫代碼篩選條件)之聘案，再藉由聘案資訊，分別寄送郵件給計畫主持人與計畫助理。每月聘案資料主要包含以下欄位：

- 姓名(助理)
- Email(助理)
- 本校計畫代碼
- 計畫名稱
- 計畫主持人(教師)
- 聘案起始日
- 聘案結束日
- 是否止聘

根據過往經歷，每月的聘案資料高達 5 萬筆以上，且實際需寄送的郵件數量高達 700 封以上，整個過程極為耗時，且郵件無法根據寄送對象來進行相應變化。為此，我們製作「每月郵件寄送系統」來改善並加速此流程。

B. 系統架構

此系統由 n8n 工作流、本地 PostgreSQL 資料庫系統和 Google 的各個 API 應用組成，主系統可拆分成三個 n8n 子工作流，分別為「上傳(Upload)」、「篩選(Filter)」和「寄送(Send)」。

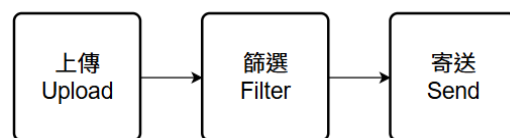


圖 7：系統主架構流程

1. 上傳(Upload)

此部分的系統會將資料從 Google

試算表上傳至本地的 PostgreSQL 資料庫。使用者須將每月的聘案資訊、計畫代碼篩選條件(可為自然語言)、全校教師 email 資訊(聘案資料中不包含計畫主持人 email)匯入 Google 試算表，接著執行此 workflow，系統將根據各個表格內容，將資料填入資料庫的對應表格，以便對資料做後續處理。



圖 8：上傳(Upload)子 workflow

2. 篩選(Filter)

在篩選的部分，系統會將資料庫的內容進行篩選。在執行完上傳子 workflow 後即可執行此 workflow。系統首先會將篩選條件(自然語言)交給 AI 做處理，將篩選條件轉為 SQL query。接著，系統將根據當月月分和 AI 生成之 SQL 進行條件篩選，並將符合欄位的資料標示為寄送郵件候選，最後將有被標示的資料全數取出並從中提取對應資訊，構成每一封郵件的客製化資訊。每筆郵件資訊皆包含以下資料：

- 姓名
- 身分(助理、教師)
- Email
- 計畫代碼
- 計畫名稱
- 郵件寄送旗幟(用於追蹤郵件是否已寄送)



圖 9：篩選(Filter)子 workflow

3. 寄送(Send)

在這部分，系統會透過篩選所產生的郵件資訊來寄送郵件。執行完篩選子 workflow 後即可執行此 workflow。系統首先會提取前 400 個「郵件寄送旗幟」為 false 的郵件資訊，接著針對這些資訊，使用變數來改變郵件的相應內容，使每封寄出的郵件皆會因郵件資訊而有所差異，最後藉由 Gmail API，將各個不同的信件寄送出去，並將相應郵件資訊的「郵件寄送旗幟」設為 true，紀錄此郵件已成功寄送。

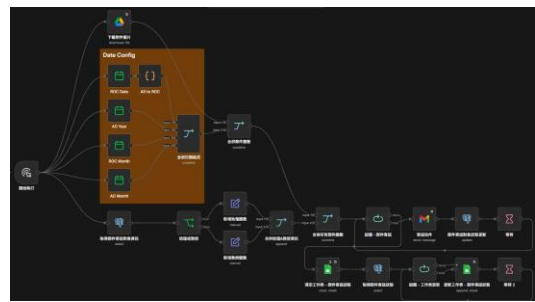


圖 10：寄送(Send)子 workflow

C. 時程安排與系統測試

專題上半學年基本上為製作並不斷改進系統的功能，包含在郵件內添加客製化資訊、改進篩選流程等。這時的系統尚未實際應用，所以測試的方式為輸入人工製造偽聘案資訊，接著透過輸出系統篩選的郵件資訊到 Google 試算表並比對與人工篩選的差異來驗證其可靠性。

專題的下半學年，我們開始實際應用並測試此系統，使用從學校資料庫提取的聘案資料輸入系統，並正式開始透過系統寄出郵件。開始正式測試後的前兩個月，由於還未確保系統的穩定性，所以在寄出的郵件中加入此信件為系統自動寄送的附註。而在

第三個月後，由於已驗證系統的穩定性，已將此附註從郵件中刪除，並將此系統作為主要每月寄送郵件的方法。（以下圖片為 115 年 4 月份郵件內容範例）



圖 11：郵件內容(上)

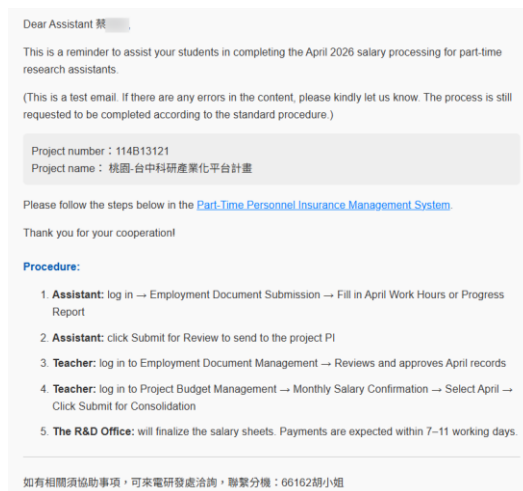


圖 12：郵件內容(下)

D. 主要挑戰和解決方法

專題的進程並非一帆風順，其中遭遇的困難之一便是「篩選機制」。由於大部分的行政人員並不會編寫 SQL，所以一開始的設計為輸入自然語言的篩選條件，而後使用 AI 來直接對篩選條件和聘案資料進行交叉比對篩選。此做法在資料量小的時候還可以使用，但由於學校的聘案資料動輒高達數萬筆，就算先以月分篩選也會有

高達上千筆資料，實際用此方法篩選可能耗上數個小時，並且結果不一定是準確的。後來的流程改為 AI 將自然語言轉為 SQL，而後 PostgreSQL 資料庫系統使用 SQL 來進行資料篩選，同時提升篩選準確性和篩選效率。

第二遭遇的困難為 Gmail API 的每日限制。在免費版本中，Gmail API 一天最多能寄出 500 封郵件，但是每月需寄送的郵件量卻至少有 700 封。為了解決此問題，我們在資料庫中的郵件資訊中加入了「郵件寄送旗幟」欄位，用來監測此郵件資料是否已被系統寄送過，並改善了「寄送」子工作流的邏輯，使其執行一次只會寄送 400 封郵件。若執行一次「寄送」子工作流尚有未寄送的郵件資料，則需等到隔天再執行一次此子工作流，直到所有郵件資料的「郵件寄送旗幟」皆標示為 true。

4. 主要成果與評估

I. 計畫案查詢系統評估

截至五月底為止，本系統已實際應用於研發處的內部進度會議與學校層級的相關會議，累計支援 3 場。在這些真實場景中，系統能即時回應使用者以自然語言提出的查詢需求，並產出格式一致的數據與圖表；相較於過往人工整理需耗費約一個工作天，現可於一分鐘內完成。查詢結果的正確性透過與原始資料逐筆比對加以驗證，協助測試的行政人員亦回饋系統產出可直接用於會議報告。本系統尚未進行大規模量化評估（如查詢正確率統計、正式滿意度問卷），此為後續可補強的方向。

在未來擴展規劃方面，我們期望

從廣度與深度兩個方向持續精進。在廣度上，擴展問題覆蓋面，讓系統能理解並回應更多元、更複雜的查詢需求，涵蓋使用者在不同情境下可能提出的各種問題，進一步提升查詢的彈性與實用性；在深度上，擴展資料來源，現階段資料以產學計畫為主，未來規劃串接更多來源，例如國科會計畫等，使系統不再侷限於單一類型的資料，而能進行跨領域、跨來源的整合查詢。

透過上述規劃，我們希望將這套查詢流程從單一應用逐步推展為一個全面、可延伸的智慧查詢平台，成為協助校務決策與行政作業的有力工具。

II. 每月郵件寄送系統評估

截至 5 月底為止，此系統已在研發處發送了三次每月郵件，並且使每月寄送郵件的任務從耗時約 3 小時縮短到約 15 分鐘(單次執行時間)。

關於此系統的未來擴展規劃，我們期望此系統的本地資料庫可串接至學校資料庫，使其可每月自動將該月的聘案資訊和本校教授資訊自動同步到本地資料庫，減少所需的輸入資訊。再來，此系統尚缺乏防呆措施，錯誤的輸入可能導致系統出錯，未來可考慮在其中加入偵錯機制，使系統可自動察覺並指出錯誤輸入，易於使用者進行除錯。

5. 結語與展望

綜合各個測試結果與實際應用，「計畫案查詢系統」和「每月郵件寄送系統」皆對於學校行政工作帶來了正面效益，降低耗時耗力的同時也使使用者能將注意力放在更重要的作業

之上。

目前這些系統僅在研發處進行測試與實際應用，在見證系統帶來的效益後，我們期望未來可推廣類似的作法到學校的其他部門，使大家能從繁複且大量的工作中解放，並更大幅度的降低學校的行政負擔。

6. 銘謝

首先，感謝董柏伶小姐協助「計畫案查詢系統」的測試，並提供寶貴的使用回饋，讓我們得以據此持續精進系統。

再來，感謝胡君華小姐，協助於「每月郵件寄送系統」的測試並且給予此系統回饋，使我們能得到系統的實際運作情況，並易於對系統的改進與除錯。

最後，再次感謝以上對專題有貢獻的人們，缺少他們的協助，許多系統開發上困難將無法突破，系統也無法如期完成上線測試與應用，有了所有人的幫助，我們才得以走到這一步。

7. 參考文獻

- [1] Yu, T., et al. Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task. arXiv:1809.08887, 2018.
- [2] Liu, X., et al. A Survey of Text-to-SQL in the Era of LLMs : Where are we, and where are we going? arXiv:2408.05109, 2024.
- [3] D. Gao, H. Wang, Y. Li, X. Sun, Y. Qian, B. Ding, and J. Zhou, "Text-to-SQL Empowered by Large Language Models: A Benchmark

Evaluation," arXiv:2308.15363, 2023.

