

基於生成式人工智慧與檢索增強生成技術 的校園資訊整合平台

A Campus Information Integration Platform Based on Generative Artificial Intelligence and Retrieval-Augmented Generation

王能中 李雅淳 王語瑄 張智淵

國立聯合大學資訊工程學系

苗栗市南勢里聯大二號

ncwang@nuu.edu.tw

{U1116044, U1118015, U1124019}@o365.nuu.edu.tw

摘要

在本研究中，為了解決傳統校園資訊系統面臨的資料分散和缺乏互動性等問題，我們設計並實作了「生成式 AI 與 RAG 校園資訊整合平台」。本平台的核心是將生成式人工智慧(GAI)與大型語言模型(LLM)技術結合，構建一個智慧化的資訊服務系統。本系統主要包含四個部份：「資料蒐集系統」、「知識庫」、「檢索增強生成系統」與「對話系統」。在資訊系統中，系統透過自動化網路爬蟲蒐集並預處理校園資料。接著，知識系統利用詞嵌入與知識圖譜技術，將資料轉換為可高效檢索的知識庫。在對話系統中，系統以 LLM 為基礎，並靈活運用多元增強生成策略，動態生成具備時效性、專業性與準確性的回應。

關鍵字：生成式人工智慧、知識圖譜、大型語言模型、檢索增強生成、智慧校園。

Abstract

In this study, we designed and implemented the “Generative AI and RAG-Based Campus Information Integration Platform.” The core concept of this platform is to integrate GAI with Large Language Model (LLM) technologies to construct an intelligent information service system. The system consists of four major components: the Data Collection System, the Knowledge Base, the Retrieval-Augmented Generation System, and the Dialogue System. Within the information layer, the system uses automated web crawlers to gather and preprocess campus data. Next, the knowledge layer transforms this data into an efficiently searchable knowledge base through embedding techniques and knowledge-graph construction. In the dialogue layer, the system relies on an LLM and applies diverse enhancement strategies to dynamically generate responses that are timely, domain-relevant, and accurate.

Keywords: Generative Artificial Intelligence (GAI), Knowledge Graph, Large Language Model (LLM), Retrieval-Augmented Generation (RAG), Smart Campus.

1. 簡介

1.1. 前言

隨著數位校園迅速發展以及資訊量的急遽成長，傳統校園資訊系統因仍依賴靜態資料庫與分散式管理模式，逐漸暴露出跨平台查詢困難，以及在多媒體資料整合與展示方面的不足，皆造成使用者在查詢與應用校園資訊時面臨延遲與資訊落差。

1.2. 研究動機

本研究提出一套以生成式人工智慧 (Generative Artificial Intelligence, GAI)[1]為核心的「智慧校園資訊整合平台」，整合大型語言模型 (Large Language Model, LLM)之提示工程(Prompt Engineering, PE)與檢索增強生成 (Retrieval-Augmented Generation, RAG)[2]等先進方法，建立可即時更新且具語意理解能力的資訊系統。

此外，平台更導入多種增強生成技術，如關鍵字檢索增強生成(Keyword-RAG)[3]、快取增強生成(Cache-Augmented Generation, CAG)[4]、結構化檢索增強生成(Structured-RAG)[5]、關鍵字圖形檢索增強生成(Keyword-Graph-based-RAG, KWGB-RAG)[6]與圖形檢索增強生成(GraphRAG)[7]，以應對不同層次的查詢需求與場景複雜度。最終，本研究希望透過先進技術打造具有人性化互動體驗的智慧校園資訊平台，有效解決現行資訊系統之瓶頸。

1.3. 相關研究

美國密西西比州立大學開發 BarkPlug v.2 智慧校園聊天機器人，利用 RAG 結合大型語言模型處理校園查詢，經問答任務測試證實能提升資訊取得效率[8]。

台灣東海大學與產業合作建置全棧式 LLM 平台，整合校內資料與 GAI 提供課程輔助與研究支援[9]，本研究於平台架構亦採用其資料整合與延展性思維並加入爬蟲與自動化處理提升即時性。

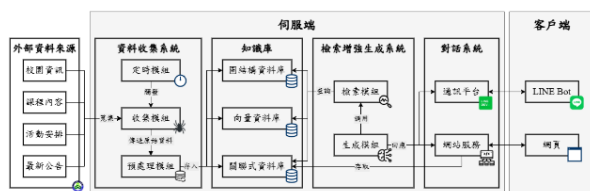
伊朗研究利用 RAG 建立大學知識檢索系統，透過公告與課程資料集並結合語意分析與 PE，使系統降低錯誤回覆率[10]，本研究依其檢索概念並加入 CAG、GraphRAG 等方法以滿足多場景查詢需求。

2. 系統內容

在本研究中，我們透過爬蟲蒐集校園資訊並存入多種資料庫，接著以不同檢索方式進行資訊查詢與整合，最終將結果送入 LLM 進行回答生成。

2.1. 系統架構

本系統架構從外部資料來源經由資訊系統進行蒐集與預處理，透過知識系統進行建構與檢索，最後結合對話系統生成回應，提供使用者完整的互動服務，系統架構圖，如圖一所示。



圖一：系統架構圖。

2.2. 開發環境

本系統的開發環境此部分包含在開發、執行、部署過程中所需的基礎環境、工具與執行平台，確保整體服務能在穩定、可重現的條件下運作，說明如下。

2.2.1. Selenium

Selenium 是一套開源的自動化測試與瀏覽器操作框架，主要用於在各種瀏覽器上模擬人類行為。它提供跨平台、跨瀏覽器的自動化能力，使開發者能以程式碼精準控制瀏覽器。藉由統一的 API 與模組化設計，Selenium 能整合多種驅動程式與測試工具。

2.2.2. PyPDF2、docx、openpyxl、pytesseract

多種檔案格式解析工具提供多類型文件的讀取、分析與文字抽取能力。它們各自負責特定類型的內容解析：PyPDF2 用來讀取並提取 PDF 文字與頁面資訊；docx 模組專門處理 Word 文件，使程式能取得段落、表格與樣式資料；openpyxl 則支援 Excel 活頁簿的讀取與操作；pytesseract 配合 Python 的影像處理函式庫(Python Imaging Library, PIL)的影像處理能力，能對圖片進行光學文字辨識(Optical Character Recognition, OCR) 辨識並轉換為可搜尋的文字。

2.2.3. PostgreSQL

PostgreSQL 是一套功能完整且高度擴充的開源關聯式資料庫系統，搭配 pgvector 擴充模組，能直接在資料庫中儲存與搜尋向量嵌入，可以高效完成語意搜尋、推薦系統 RAG 檢索任務。這種原生整合讓資料庫同時具備結構化資料管理與向量搜尋能力，使開發者能在一套系統中完成混合查詢與內容檢索串接，降低資料堆疊的複雜度並提升整體效能。

2.2.4. Weaviate

Weaviate 是一套開源的向量資料庫，能同時儲存資料物件及其向量嵌入，並提供語意搜尋功能。

其架構支援模組化整合，可串接多種向量嵌入模型，適用於 RAG、語意問答、知識庫等各類應用場景，方便開發者快速構建智能檢索與生成系統。

2.2.5. Neo4j

Neo4j 是一套圖形資料庫，透過節點與邊表示資料及其關係，並使用 Cypher 查詢語言進行圖形遍歷與模式匹配。它支援向量索引，可將節點屬性轉為向量並建立 HNSW 索引，用於語意搜尋。開發者可先用外部模型計算嵌入，再存入 Neo4j，查詢時可結合向量相似度與關鍵字搜尋，實現混合檢索。

2.2.6. Docker

Docker 是一套開源的容器化平台，能將應用程式及其相依環境打包在輕量級、可移植的容器中，確保在不同作業系統與伺服器上均能一致執行。其架構支援模組化整合，開發者可將資料庫、後端服務、前端應用與其他依賴各自封裝成容器。Docker 適用於快速部署、持續整合與測試環境建立，也方便在校園資訊平台中構建多模組系統、管理服務依賴，提升開發效率與系統穩定性。

2.2.7. LangChain

LangChain 是一個開源框架，旨在簡化 LLM 在應用程式開發中的使用流程。它提供一個統一的介面，使開發者能方便地連接不同的語言模型，無論是替換模型版本還是測試不同模型的效能，都能快速實現。

2.2.8. LangGraph

LangGraph 是一個開源框架，專為構建以 LLM 為核心的多模組應用而設計。它提供可視化與程式化的流程管理工具，使開發者能將資料檢索、知識庫整合、生成策略和對話邏輯等多個模組串接成靈活的工作流。

2.2.9. LM Studio

LM Studio 是一款本地化的 LLM 操作平台，用於在個人電腦上管理、運行與測試各種類型的開源模型。它支援多種推論後端與量化格式，使模型能在 CPU 或 GPU 上高效運作。透過內建的聊天介面、API 端點、模型監控與日誌功能，開發者可以直接在本機環境測試模型效能、調整參數，或將其整合至應用程式工作流中。

2.2.10. Django

Django 是一套開源的高階 Python Web 框架，專注於快速開發與乾淨、實用的設計。其架構支援模組化整合，開發者可利用內建的物件關聯對應、路由系統、模板引擎與認證機制，快速構建後端服務、管理資料庫操作與使用者介面。Django 適用於校園資訊平台、內容管理系統與各類 Web 應用場景，能高效整合資料庫、前端與第三方服務。

2.2.11. LINE Bot SDK

LINE Bot SDK 是一個專門為 LINE Bot 開發者設計的軟體開發工具包，其中包含了 LINE Bot API 所支援的全部功能與介面，也提供各種常見程式語言的函式庫及範例程式。

2.3. 相關技術

2.3.1 生成式人工智慧(Generative AI, GAI)

生成式人工智慧(GAI)基於深度學習與神經網路架構,透過大規模語料進行預訓練與微調,使模型具備語意理解與自然語言生成能力,提升靈活性與語意精準度。

2.3.2. 檢索增強生成 (Retrieval-Augmented Generation, RAG)

RAG 結合向量資料庫(Vector Database)技術,透過語意相似度檢索獲取相關文本,再交由 LLM 生成回應。此技術能動態連結外部知識來源,適用於需要即時更新的查詢情境。

2.3.3. 關鍵字檢索增強生成(Keyword-Retrieval-Augmented Generation, Keyword-RAG)

Keyword-RAG 是一種結合關鍵字檢索與生成式 LLM 的增強技術。它首先從輸入查詢中抽取關鍵字,利用這些關鍵字在資料庫或知識庫中進行精準檢索,獲取高度相關的文本資訊,再交由 LLM 生成回應。此方法能減少無關資訊干擾。

2.3.4. 快取增強生成 (Cache-Augmented Generation, CAG)

CAG 透過預載關鍵知識進入上下文,減少即時檢索延遲,能在查詢集中或知識範圍固定的情境下發揮最佳效益。其主要優勢是低延遲與流程簡化,適合校園基本知識與重複查詢場景。

2.3.5. 結構化檢索增強生成(Structured-RAG)

Structured-RAG 結合向量檢索與結構化知識庫查詢。系統先將問題向量化送入 Weaviate 搜尋,取得對應 Neo4j 識別碼,再查詢 Neo4j 知識庫獲取結構化資訊(如聯絡方式、組織架構)。此架構適合需要特定結構化資訊的查詢場景。

2.3.6. 關鍵字擴充圖形檢索增強生成(Keyword-Graph-based Retrieval-Augmented Generation, KWGB-RAG)

KWGB-RAG 是一種結合向量語意檢索、關鍵字圖譜擴展的增強生成技術。它透過並行檢索同時抓取語意相關與關鍵字相關資訊,再從檢索結果中提取新關鍵字進行圖譜擴展,最後由 LLM 驗證並整合所有資訊生成最完整的答案。此方法能同時兼顧精準性與全面性。

2.3.7. 圖形檢索增強生成(GraphRAG)

GraphRAG 是一種結合知識圖譜與生成式 LLM 的增強技術。系統先從輸入查詢中提取關鍵字,並在 Neo4j 圖譜中查詢共現關鍵字以發現語義相關文檔;同時將問題向量化,在 Weaviate 進行相似度檢索。兩路檢索結果合併後,提供 LLM 更全面的上下文。此方法能捕捉資料間的多跳關聯,提高檢索完整性與覆蓋率,特別適合跨部門或跨主題的校園查詢。

2.3.8. 知識圖譜(Knowledge Graph)

知識圖譜技術將多來源校園資料轉換為節點與邊的結構化關聯,能捕捉不同資訊單元間的語意與邏輯關係,支援多跳查詢與語意推理,是 GraphRAG 的關鍵基礎。

3. 系統功能

3.1. 介面設計

本系統提供網頁與 LINE Bot 兩種介面;其中網頁以 Django 架構整合前後端互動,LINE Bot 則提供快速查詢入口。

3.2. 校園資訊查詢

系統支援文字與語音輸入,並可回答校園資訊、課程與活動相關問題,同時提供建議詢問項目以提升使用體驗。

3.2.1. 方法選擇

系統提供 RAG、Keyword-RAG、CAG、KWGB-RAG、GraphRAG 等多種檢索模式,以因應不同查詢需求與精準度。

3.2.2. 聊天歷史管理

所有對話記錄皆儲存於資料庫,並可以在側欄瀏覽或刪除。

3.2.3. 行事曆功能

系統自動擷取文章日期並呈現在行事曆中,使用者可查看事件摘要並將重要事件釘選。

3.2.4. 登入與帳號管理

提供電子郵件驗證碼登入、密碼重設與帳號資訊修改功能。

3.2.5. 意見回饋

使用者可透過彈跳視窗提交回饋意見。

3.2.6. 系統介紹

平台提供基本操作說明以降低使用門檻。

4. 系統實作成果

本系統的系統實作成果包含:「核心檢索機制實作」、「使用者介面」、「管理者介面」及「模型效率分析」,說明如下。

4.1. 核心檢索機制實作

本系統的檢索機制分為兩個層級,分為基礎檢索層與進階圖譜增強層。基礎層提供穩固的檢索能力,結合不同檢索方法互補優勢;進階層則引入知識圖譜的深度應用,以提升檢索完整性與智能性。

4.1.1. 基礎檢索機制

基礎檢索層構成了整個系統的核心。本層級採用兩種互補的檢索機制,分別從不同維度提升檢索品質:第一種機制聚焦於「精確度與語義理解的平衡」,第二種專注於「文本與結構化資訊的整合」。這兩種機制相輔相成,為後續的進階圖譜增強提供穩固的基礎。

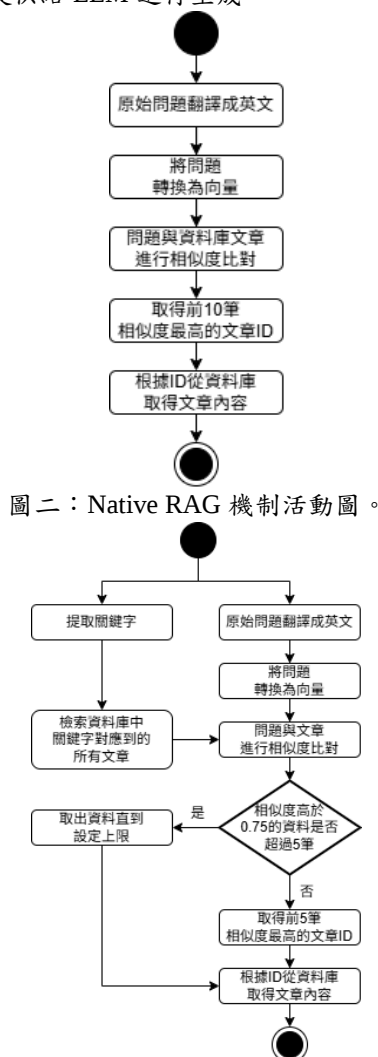
A. Native RAG 機制

Native RAG 機制是一種結合外部知識檢索與生成式模型的混合式推理流程。如圖二所示,系統會先接收使用者的自然語言問題,並將問題向量化後送入向量資料庫進行語意檢索,以取得與問題最

相關的文件片段。RAG 機制的核心流程包含「檢索」與「生成」兩階段：在檢索階段，向量資料庫根據語意相似度排序文件並返回最相關的內容；在生成階段，LLM 會根據檢索到的內容作為外部知識上下文，產生具備事實性、可推論性與可解釋性的答案。

B. Keyword-RAG 機制

Keyword-RAG 機制是一種混合檢索流程，旨在結合關鍵字精確性與向量語義理解。如圖三所示，系統首先從中文問題中提取關鍵字來檢索 PostgreSQL 資料庫中的文章，同時將問題轉換為向量對檢索到的文章進行相似度比對。最終，系統根據這些 ID 從 PostgreSQL 提取原文文章內容，作為上下文提供給 LLM 進行生成。

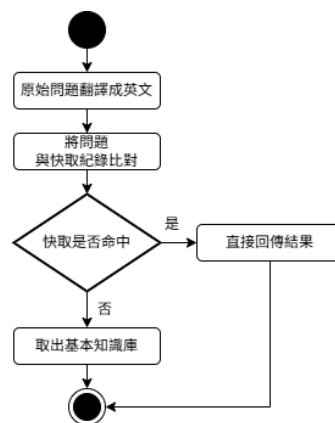


圖二：Native RAG 機制活動圖。

圖三：Keyword-RAG 機制活動圖。

C. CAG 機制

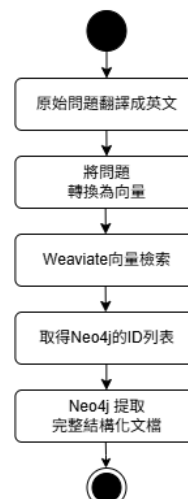
CAG 機制以固定知識庫為基礎運作，先載入校園基本知識庫作為唯一依據，並在收到使用者問題後將其向量化，透過 PostgreSQL 向量索引搜尋是否命中相似度達閾值的快取資料；若命中則直接返回歷史答案以提升速度與一致性，若未命中則自動構建 CAG Prompt，將問題與知識庫內容結合送入 LLM 生成回應，最後把該次問答與向量嵌入寫入 PostgreSQL 作為後續快取來源。



圖四：CAG 機制活動圖。

D. Structured-RAG 機制

結構化增強檢索實作了一套雙知識庫協作的流程。如圖五所示，系統首先將問題向量化，送往 Weaviate 進行相似度搜尋以取得 Neo4j 識別碼。隨後，系統利用此識別碼查詢 Neo4j 知識庫，獲取結構化資訊(如聯絡方式、組織架構)來增強檢索內容。最終，系統會將從 Neo4j 擷取的結構化資訊整理後作為生成階段的背景知識，提供給 LLM 參考。



圖五：Structured-RAG 機制活動圖。

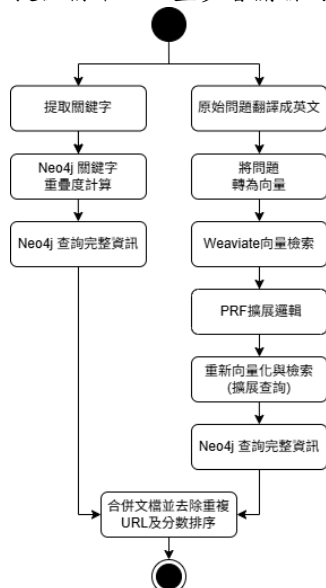
4.1.2. 進階知識圖譜加強

在基礎檢索的基礎上，進階層級引入知識圖譜的深度應用，進一步提升檢索的完整性與智能性。本層級包含兩個主要創新：KWGB-RAG 架構透過發現語義共現關係來擴展檢索範圍，提高召回率；GraphRAG 採用自適應檢索策略，對宏觀問題進行社群層級全局搜尋，對具體實體問題則先向量檢索再擴充三元組，依查詢類型自動選擇最合適的檢索與生成方式。

A. KWGB-RAG 機制

KWGB-RAG 機制基於 Neo4j 的關鍵字關聯網路，其核心目標是結合向量的語意理解、關鍵字的精準檢索以及圖譜的關聯性，以找出最完整的答案。系統採用並行檢索與多階段擴展的混合策略：首先將使用者問題向量化，送入 Weaviate 進行語意相似

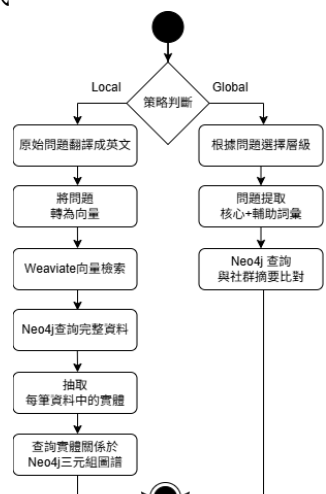
度搜尋，同時從問題中提取關鍵字，送入 Neo4j 查詢直接相關資訊；接著從向量檢索結果中擷取新的核心關鍵字，再透過 Neo4j 查詢其分類與同義詞以擴展相關資訊；最後將原始問題與擴展後的資訊結合，交由 LLM 進行生成，整合各來源資料形成完整上下文，提供精確且涵蓋多層關聯的回答。



圖六：KWGB-RAG 機制活動圖。

B. GraphRAG 機制

GraphRAG 機制採用自適應的雙策略檢索模式。如圖七所示，系統首先使用 Mistral 模型對問題進行策略判斷，決定採用「全局搜尋 (Global)」或「局部搜尋 (Local)」。全局搜尋針對需要跨域整體理解的宏觀問題，系統選擇適當的社群層級，提取多個社群摘要進行 Map-Reduce 式多階段生成，最終綜合形成全面回答；局部搜尋針對指向具體實體或微觀細節的問題，系統會先進行向量檢索，然後對檢索到的資料進行三元組關係擴充，以補充更多相關資訊，結合社群背景進行精確生成。這種分層策略確保系統能針對不同問題與範圍選擇最優的檢索與生成方式。



圖七：GraphRAG 機制活動圖。

4.2. 使用者介面

使用者介面包含網頁及 LINE Bot。

4.2.1. 校園資訊平台網站

校園資訊平台網站的實作結果包含了平台首頁及登入、聊天視窗、聊天紀錄、使用者設定、行事曆、回饋系統及系統介紹七個部分。

A. 聊天視窗

使用者可使用鍵盤打字與 Bot 進行對話，對話內容會形成一筆對話串於對話視窗上。使用者的對話會顯示在視窗右側，Bot 的回應會顯示在左側，如圖八所示。另外點選加號圖案可切換方法，如圖九所示。



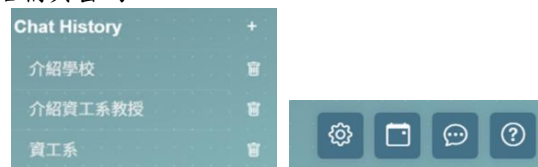
圖八：對話視窗。 圖九：切換方法。

B. 聊天記錄

聊天記錄功能會顯示過去對話的標題，每筆紀錄可新增或删除，方便使用者快速存取歷史對話，如圖十所示。

C. 功能選單

功能選單由左至右分別包含「設定」、「行事曆」、「回饋系統」及「系統介紹」，如圖十一所示。使用者點選功能選單上方的齒輪圖示，即可進入「使用者設定」頁面，在此可修改個人資料，如名稱與密碼。



圖十：聊天紀錄。 圖十一：功能選單。

D. 行事曆

點選功能選單上的日曆圖案，即可進入到行事曆頁面，如圖十二所示。行事曆會從資料庫找尋有標註完整日期的事件放至行事曆中，點選事件即可查看該事件的詳細內容，如圖十三所示。另外，點選針頭圖案可將該事件釘選。點選上方的左右鍵可切換月份。



圖十二：行事曆頁面。



圖十三：事件詳細內容。

E. 回饋系統

使用者點選功能選單中的對話按鈕，會跳出回饋系統的視窗，使用者可輸入回饋回傳給開發者。

F. 系統介紹

使用者還不清楚平台的操作方式，點選功能選單問號圖案，可進入到系統介紹的頁面。

4.2.2. LINE Bot

LINE Bot 提供了一個互動式圖文選單，每一個區塊代表每一個功能，使用者可以藉此快速且直觀地進行操作。LINE Bot 的實作成果包含了方法更改、意見信箱、前往平台及使用者指引四個部分，如圖十四所示。



圖十四：LINE Bot 圖文選單。

A. 方法更改

使用者可點選方法更改的區域進行檢索方法的更改，如圖十五所示。



圖十五：方法更改。

B. 意見信箱

使用者可按下意見信箱的區塊，此時 LINE Bot 會轉為接收回饋的模式。使用者輸入完訊息後可回傳至資料庫，如圖十六所示。



圖十六：回饋信箱。

C. 前往平台

使用者點選前往平台區塊即可前往校園資訊平台網站。

D. 使用者指引

使用者可點選使用指引的區塊獲取此 LINE Bot 的使用者指引。

4.3. 模型效率分析

本研究比較五種生成模型(LLM、Native RAG、Keyword-RAG、Structured-RAG、KWGB-RAG)，並以 RAGAs 框架評估其正確率(Content Precision, CP)、召回率(Content Recall, CR)、忠誠度

(Faithfulness, F)、整體準確率(Answer Accuracy Rate, AR)與綜合效能分數(F1 Measure, F1 Score)。

結果顯示，KWGB-RAG 的 CP、F 與 F1 Score 最高，顯示結合關鍵字擴展與圖譜關聯能有效支援跨主題檢索並維持高正確性。Structured-RAG 在 CP 最佳，反映結構化資訊補強有助於提升回答的事實對應性；Keyword-RAG 的 AR 最高，顯示關鍵詞篩選能有效提升問題貼合度。Native RAG 整體表現居中，可作為基準模型。LLM 在無檢索輔助下表現最弱，但在 F1 Score 仍顯示具備基本語意理解能力。整體而言，Structured-RAG 與 KWGB-RAG 雖具優異事實性與覆蓋度，但因回答內容較為完整，於回答聚焦度上略有不足。

表一：不同 RAG 方法針對不同效能參數的比較表

評估項目\指標	CP	CR	F	AR	F1 score
LLM	0	0	0.35	0.31	0.10
Native RAG	0.49	0.65	0.47	0.54	0.35
Keyword-RAG	0.51	0.80	0.80	0.74	0.60
Structured-RAG	0.79	0.8	0.67	0.62	0.73
KWGB-RAG	0.68	0.83	0.86	0.57	0.75

4.3.1. 評估限制說明

本研究未將 CAG 與 GraphRAG 納入 RAGAs 評估，原因在於 CAG 基於快取與固定知識庫運作，並不執行動態檢索；而 GraphRAG 著重全域語意與多跳推理，其能力無法以傳統檢索型指標衡量。此外，現有 RAG 評估依賴黃金答案比對，難以完整反映 GraphRAG 的多跳推理與資訊擴展特性，亦無法評估 CAG 的即時性與回應穩定度。

4.4. 檢索效率與特性比較

本研究分析五種檢索機制的效率與運作特性，CAG 採預先快取匹配方式，平均耗時最低，適合即時問答但精準度有限，一般 RAG 採標準向量檢索流程，速度快且通用，Keyword-RAG 先進行資料庫關鍵字過濾再向量檢索，耗時 2.24 秒，檢索精度提升，Structured-RAG 處理結構化資料，平均耗時 5.30 秒，平衡精準度與完整性，KWGB-RAG 透過多階段並行檢索與關鍵字圖譜擴展，平均耗時 13.40 秒，以換取深度與多跳推理能力，GraphRAG 涉及全局摘要與 Map-Reduce 運算，耗時最高，但提供最完整的跨主題與多跳查詢結果，整體顯示檢索速度與資料處理深度存在明顯權衡，快取與基礎檢索適合即時互動，進階檢索耗時較長但提升答案完整性、精準度與多跳推理能力，適用於複雜或跨主題校園查詢場景。

表二：不同 RAG 方法的速度比較表

檢索方法\時間	平均耗時 (秒)
CAG	0.730 s
Native RAG	1.338 s
Keyword-RAG	2.238 s
Structured-RAG	5.301 s
KWGB-RAG	13.404 s
Graph-RAG	16.078 s

5. 結論

本研究建置一套「基於生成式人工智慧與檢索增強生成技術的校園資訊整合平台」，以 Django 為核心框架，結合多模組 RAG 技術與自動化爬蟲，實現校園知識的智能檢索與自然語言互動。系統整合校內外公開資料來源，透過語意分割、詞向量嵌入與資料正規化，建立可持續更新的知識庫，並以多層次檢索策略強化回答的精準度與廣度。前端則透過即時互動式聊天介面與視覺化呈現，提升使用者查詢體驗。整體而言，本系統不僅實現了校園資訊的智慧化整合，也展示生成式人工智慧在教育領域中應用的可行性與發展潛力。

6. 專題作品影音平台網址

<https://youtu.be/-gbsN91M2hE>

參考文獻

- [1] M. Jovanovic and M. Campbell, “Generative Artificial Intelligence: Trends and Prospects,” *Computer*, Vol. 55, No. 10, pp. 107-112, October 2022.
- [2] P. Lewis, E. Perez, A. Piktus, and F. Petroni, “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” *Advances in Neural Information Processing Systems*, 33, pp. 9459-9474, 2020.
- [3] Y. Huang, S. Zhang, and X. Xiao, “KET-RAG: A Cost-Efficient Multi-Granular Indexing Framework for Graph-RAG,” *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Toronto, Canada, pp. 1003-1012, August 3-7, 2025.
- [4] B. J. Chan, C. T. Chen, J. H. Cheng, and H. H. Huang, “Don’t Do RAG: When Cache-Augmented Generation is All You Need for Knowledge Tasks,” *arXiv, Computer Science, Computation and Language*, arXiv:2412.15605, pp. 1-5, December 2024.
- [5] T. Lin, Y. Zhu, Y. Luo, and N. Tang, “SRAG: Structured Retrieval Augmented Generation for Multi Entity Question Answering over Wikipedia Graph,” *arXiv, Computer Science, Computation and Language*, arXiv:2503.01346, pp. 1-17, March 2025.
- [6] Y. Xia, J. Wu, S. Kim, T. Yu, R.A. Rossi, H. Wang, and J. McAuley, “Knowledge Aware Query Expansion with Large Language Models for Textual and Relational Retrieval,” *arXiv, Computer Science, Information Retrieval*, arXiv:2410.13765, pp. 1-15, October 2024.
- [7] H. Han, Y. Wang, H. Shomer, K. Guo, J. Ding, Y. Lei, M. Halappanavar, R.A. Rossi, and S. Mukherjee, “Retrieval-Augmented Generation with Graphs (GraphRAG),” *arXiv, Computer Science, Information Retrieval*, arXiv:2501.00309, pp. 1-88, December 2024.
- [8] M. Borah, et al. From Questions to Insightful Answers: Building an Informed Chatbot for University Resources (BarkPlug v.2). *arXiv preprint arXiv:2405.08120*, 2024.
- [9] 東海大學，「東海大學攜手 HPE、NVIDIA、AMD 建構生成式 AI 全棧平台」，2024。取自 https://www.thu.edu.tw/web/news/news_detail.php?cid=22&id=4307
- [10] A. Khodayari, et al. Leveraging RAG for Persian University Knowledge Retrieval. *arXiv preprint arXiv:2411.06237*, 2024.