

中原大學  
電機工程學系  
專題報告

封面

中原大學校園嚮導

Campus Guide

指導教授：王佳盈

專 題 生：11028201 高孟麟  
11028220 葉威廷  
11028224 包傾祺

中華民國 113 年 12 月



# 致謝

首先，我要感謝我的指導教授王佳盈教授，感謝您在這段期間給予我們的指導和支持。您的專業知識和耐心教導讓我們在專題研究中獲益良多，並且在使用 Langchain 和 RAG 技術方面提供了寶貴的建議。

其次，我們還要感謝我們的家人和朋友們，感謝你們在我們專題研究過程中的鼓勵和支持。你們的理解和陪伴讓我們能夠專注於研究，並且在遇到困難時給予我力量。

此外，我們也要感謝中原大學電機系的所有老師和同學們，感謝你們在學習和研究中的幫助和合作。特別感謝研究室的一個學長，感謝你在技術問題上的討論和分享，讓我們在研究中能夠更深入地理解和應用所學知識。

最後，我們要感謝所有參與和支持這個專題報告的人，感謝你們的貢獻和努力。這篇報告的完成離不開你們的幫助和支持，謝謝你們！

# 中文摘要

現今科技蓬勃發展，隨著 OpenAI 於 2022 年底公布 ChatGPT 至今，才短短的兩年時間，人們平常使用科技的方式又被徹底的改變了，從以前的查尋一個資料必須徹底的仔細查找許多資料後再去比對，現在我們已經可以針對我們要問的問題去做詢問，GPT 隨後就會給你答案了。

然而 GPT 的資料仍然是有限的，全世界有太多的人跟太多的資訊，如果要針對特定資訊去做回答普通的 GPT 是無法達成的，於是我們想要透過 Langchain 結合 GPT，讓 GPT 可以根據我們給予他的資訊去做回答，再來就是結合網頁，方便大家能有更加快速又方便的管道來了解中原的資訊，而不必去網站上看密密麻麻的文字內容。

總而言之，本專題報告介紹了一個名為「中原大學嚮導」的網頁式聊天機器人系統，該系統利用 Langchain 和 RAG 技術，旨在為中原大學的學生和訪客提供智慧化的校園導覽服務。通過自然語言處理和知識檢索技術的結合，系統能夠即時回答用戶關於校園設施、活動和歷史的各類問題，並提供精確的導航和建議。

在研究過程中，我們首先對 Langchain 和 RAG 技術進行了深入的探討和實驗，確定了其在智慧導覽系統中的應用可行性。接著，我們設計並實現了一個原型系統，通過多次測試和優化，最終達到了預期的功能和性能目標。

本報告詳細介紹了系統的設計思路、技術實現和測試結果。我們相信，這一智慧導覽系統將為中原大學的師生和訪客提供更便捷和智慧的校園體驗。

## 英文摘要

With the rapid development of technology, the way people use technology in their daily lives has been completely transformed in just two short years since OpenAI released ChatGPT at the end of 2022. Previously, finding information required thorough and meticulous searching and comparison of numerous sources. Now, we can simply ask our questions, and GPT will provide the answers.

However, GPT's data is still limited. There are too many people and too much information in the world for a standard GPT to handle specific queries effectively. Therefore, we aim to combine Langchain with GPT, allowing GPT to respond based on the information we provide. Additionally, by integrating this with a web interface, we offer a faster and more convenient way for everyone to access information about Chung Yuan Christian University without having to sift through dense text on websites.

In summary, this project report introduces a web-based chatbot system named "Chung Yuan University Guide," which utilizes Langchain and RAG technologies. The system aims to provide intelligent campus guide services for students and visitors of Chung Yuan Christian University. By combining natural language processing and knowledge retrieval technologies, the system can instantly answer various questions about campus facilities, activities, and history, and provide precise navigation and suggestions.

During the research process, we first conducted in-depth exploration and experiments on Langchain and RAG technologies to determine their feasibility in an intelligent guide system. Then, we designed and implemented a prototype system, which, after multiple tests and optimizations, achieved the expected functionality and performance goals.

This report details the system's design concepts, technical implementation, and test results. We believe that this intelligent guide system will provide a more convenient and intelligent campus experience for the faculty, students, and visitors of Chung Yuan Christian University.

致謝	.....	I
中文摘要	.....	II
英文摘要	.....	III
目錄	.....	IV
圖目錄	.....	V
表目錄	.....	VI
第一章 緒論	.....	1
1.1 研究背景	.....	1
1.2 研究動機	.....	3
1.3 研究問題	.....	6
1.4 研究目的	.....	9
第二章 研究方法	.....	10
2.1 文獻回顧	.....	10
2.2 Langchain	.....	11
2.3 RAG	.....	12
第三章 研究步驟與結果	.....	15
3.1 研究與開發步驟	.....	15
3.2 研究設置方法	.....	18
3.3 程式相關設定	.....	24
3.4 程式主要設定	.....	28
3.5 研究結果與展示	.....	30
第四章 結論與未來方向	.....	36
4.1 結論	.....	36
4.2 未來方向	.....	37
參考文獻	.....	39
授權書	.....	40

-----

## 圖目錄

圖 1	南華資管聊天機器人的起始畫面.....	10
圖 2	研究結果與展示-問題一.....	32
圖 3	研究結果與展示-問題二.....	33
圖 4	研究結果與展示-問題三.....	34
圖 5	研究結果與展示-問題四.....	35

## 表目錄

表 1	本實驗使用的工具.....	9
-----	---------------	---



# 第一章 緒論

## 1.1 研究背景

### 1.1.1 人工智慧技術的發展與應用

在近十年中，人工智慧 ( AI ) 技術取得了巨大的進步，尤其是在自然語言處理 ( NLP ) 領域。隨著深度學習 ( Deep Learning ) 的普及，許多原本只存在於理論中的模型和算法逐漸變得切實可行。如今，語言模型如 OpenAI 的 GPT 系列、Google 的 BERT 等，已經能夠在各類文本生成與理解任務中展現出強大的表現。這些模型能夠模仿人類的語言表達方式，並應用於多種場景，如客服機器人、翻譯工具、搜索引擎等。

### 1.1.2 自然語言處理技術的發展

NLP 技術是支持現代聊天機器人的核心。傳統的聊天機器人通常使用規則匹配或基於決策樹的簡單算法，這些方法雖然可以處理簡單的對話，但對於複雜的語義理解和上下文推理能力有限。隨著 Transformer 架構的出現，尤其是 BERT ( Bidirectional Encoder Representations from Transformers ) 模型的應用，語言模型能夠更好地理解上下文和長距離依賴關係。這為語言模型在對話系統中的應用奠定了基礎。

在此背景下，我們的專題報告中使用的 Langchain 和 RAG 技術正是基於這些先進的

語言模型，它們不僅能夠生成符合語法的對話，還可以在特定領域內進行知識檢索和精確回答問題。

### 1.1.3 Langchain 技術的發展

Langchain 是一種用於構建基於語言模型應用的框架。這個框架使得開發者能夠將多種功能模塊結合在一起，構建一個更強大的對話系統。Langchain 支持通過「鏈式」的方式將多個處理步驟連接在一起，這意味著用戶的每個查詢都可以經歷多個語言模型的處理步驟，而不僅僅依賴單一模型。這使得 Langchain 在應對複雜問題、處理多輪對話時具有更高的靈活性。

Langchain 的另一個特點是它與外部數據源的集成能力。傳統的聊天機器人主要依賴於訓練數據中的知識，無法應對新知識的查詢。而通過 Langchain，開發者可以將外部數據庫、API 或實時的網絡資源接入到系統中，從而實現實時的信息檢索，這對於我們的專題「中原大學嚮導」的場景尤為重要，因為它需要根據實時的校園信息為用戶提供準確的回覆。

### 1.1.4 RAG 技術的應用

RAG ( Retrieval-Augmented Generation ) 是一種結合了信息檢索與文本生成技術的系統架構。與單純的文本生成模型不同，RAG 可以根據查詢內容從外部知識庫中檢索相

關信息，並將檢索到的知識與生成模型結合，以提供更加準確和信息豐富的回覆。

RAG 系統的工作流程通常分為兩個步驟：首先，系統會基於用戶的查詢從大規模的外部資料中檢索與之相關的文檔或信息；接下來，系統會將這些檢索到的文檔作為輔助信息，生成最終的回覆。這種技術有效地解決了傳統生成模型可能面臨的「幻想」問題，即模型在缺乏真實知識的情況下產生虛構內容。

在「中原大學嚮導」中，RAG 技術可以被用來查詢校園內的具體信息，如建築位置、課程安排、校園活動等。這種方式能夠確保用戶獲得的信息不僅是語言模型生成的流暢句子，更是基於真實數據和知識的準確答覆，從而提升用戶的信任感和使用體驗。

### 1.1.5 聊天機器人的演進

早期的聊天機器人多依賴於預設的規則和模板，這些機器人的靈活性和智慧性受到限制。例如，FAQ 系統通常是基於預先設定的問題與答案匹配，但這類系統無法應對用戶提出的非預期問題。隨著語言模型和深度學習的發展，現代的聊天機器人已經能夠在更為開放的環境下進行對話，並逐漸應用於教育、醫療、客服、娛樂等多個領域。

尤其在教育領域，AI 技術逐漸被引入，用於輔助學生學習、回答問題、提供學術指導等。對於「中原大學嚮導」而言，聊天機器人的目標不僅是提供校園信息，還應當具備一定的教學輔助功能，幫助新生更快地熟悉校園環境，並為其提供更精準的指導。

此外，隨著技術的不斷演進，聊天機器人的可擴展性和交互性不斷提升，這也為學術研究和實際應用提供了更多的可能性。

### 1.1.6 「中原大學嚮導」的應用場景

「中原大學嚮導」作為一個專為中原大學校園設計的智慧型聊天機器人，具體的應用場景涵蓋了多個方面。首先，這個系統可以幫助新生更快地了解校園環境，通過查詢校園內的建築位置、設施服務等，輕鬆找到自己需要的資訊。此外，它還可以用於查詢課程信息、活動安排，甚至提供個性化的校園活動推薦，這有助於學生融入校園生活。

除此之外，該系統還可以為校園內的訪客提供導覽服務，通過即時的問答來解決訪客的需求，從而提高訪客的體驗和滿意度。系統還可以根據使用者的需求，進行特定主題的深入對話，這種靈活性是傳統查詢系統無法達到的。

## 1.1.7 技術挑戰與未來發展

儘管 Langchain 和 RAG 技術提供了強大的工具來構建智能聊天機器人，但在實際應用中仍然存在一些技術挑戰。例如，如何有效地管理和更新知識庫，以確保系統回答的準確性和及時性；如何處理用戶提問中的模糊語義，並為其提供正確的解答；以及如何進一步提升對話系統的自然度，使其更像人類對話，這些問題仍需進一步的研究和優化。

隨著 AI 技術的持續發展，未來的聊天機器人將會更加智慧，能夠進行更複雜的推理和問題解答。同時，對於具體應用場景的深度定制化和個性化也是未來發展的重要方向。對於「中原大學嚮導」這樣的系統，未來可以考慮集成更多的校園服務，甚至結合移動應用、AR 技術等，為用戶提供更沉浸式的校園導航和指導體驗。

## 1.2 研究動機

### 1.2.1 校園導覽需求的增長

隨著大學校園的規模日益擴大，學生、訪客和教職員工在日常生活和學習中，對於校園信息的需求變得越來越多樣化且即時化。特別是對於新生和校園訪客來說，熟悉校園環境和獲取相關資源可能是一個漫長而艱難的過程。傳統的校園導覽方式通常依賴於地圖或人工導覽，這些方式雖然能夠解決部分需求，但往往不夠靈活，無法實時更新，且無法為每個使用者提供個性化的解答。

基於此現狀，開發一個智慧化、可交互的校園導覽系統成為一個迫切的需求。此專題報告的核心目標是通過人工智慧技術，特別是語言模型和檢索增強生成（RAG）技術，來解決這一問題。該系統不僅能夠回答學生、教職員工和訪客的具體問題，還可以根據實時更新的信息提供精確且個性化的導覽服務，從而提升用戶在校園中的體驗。

### 1.2.2 現有校園導覽系統的局限性

目前，許多大學提供了不同形式的校園導覽工具，如靜態地圖、導航應用程式或 FAQ 問答系統等。然而，這些工具通常存在一些明顯的局限性。首先，靜態地圖無法解決用戶的具體問題，無法根據個人需求提供定制化的建議。其次，傳統的 FAQ 系統多為基於規則的系統，難以處理複雜的語義理解和多輪對話。這些系統對於訪客來說，無法提供足夠靈活的交互方式，導致用戶體驗不夠理想。

人工智慧技術的進步，尤其是自然語言處理和生成模型的發展，為解決這些問題提供了新的思路。通過集成先進的 NLP 技術，校園導覽系統不僅能夠識別並理解用戶的自然語言輸入，還可以在此基礎上提供動態的回覆和建議。這種智慧化的系統將大大改善現有導覽工具的不足，並提高校園內的服務質量。

### 1.2.3 人工智慧技術在教育領域的潛力

人工智慧已經廣泛應用於教育領域的多個方面，如智慧輔導系統、自適應學習平台和虛擬助教等。這些技術不僅能夠幫助學生更有效地學習，還能為教師提供教學上的輔助工具。智慧型聊天機器人作為人工智慧技術的一部分，具備即時性、個性化、可擴展性強等優點，特別適合用於教育場景中的輔助教學和信息查詢。

在「中原大學嚮導」專題中，我們結合了人工智慧和校園導覽服務的兩大需求，這樣的系統不僅能夠為學生提供即時的校園信息，還可以作為新生熟悉環境的助手。透過智慧對話系統，學生可以輕鬆查詢校園內的設施、課程安排、活動時間表等，這將大大提高他們的學習效率和校園生活的便利性。

此外，這樣的系統也可以為教師和管理人員提供便捷的查詢服務，簡化日常校務管理。整體而言，將 AI 技術引入校園導覽和管理服務，將對於學校整體運作和師生生活質量的提升產生積極影響。

### 1.2.4 提升校園訪客和新生的體驗

中原大學每年吸引大量的訪客、外國交換生和新入學的學生(號稱 100%註冊率)。對於這些群體而言，迅速熟悉校園環境和了解學校提供的各項服務至關重要。然而，傳統的校園介紹和導覽方式無法滿足其多樣化的需求，尤其是在面對語言障礙或個性化

需求時，這些方式顯得尤為不足。

智慧型聊天機器人技術的引入，能夠極大地提升校園訪客和新生的體驗。透過自然語言處理技術，訪客可以使用母語與系統進行交互，並得到準確且具體的回答。此外，該系統可以針對不同背景的使用者提供定制化的導覽建議，例如為外國學生提供英語或其他語言版本的導覽，或針對新生提供校園設施的詳細介紹。這不僅能增強訪客和學生對學校的好感度，也有助於學校形象的提升。

## 1.2.5 個性化信息查詢需求的提升

隨著現代教育和生活模式的多樣化，學生對於信息查詢的需求變得越來越個性化。傳統的單一查詢方式往往無法滿足每個人不同的需求。例如，一名學生可能想知道某一特定教室的開放時間，另一名學生則可能想了解某門課程的具體安排，這些問題如果由人工或靜態系統處理，將無法快速、準確地解決。

通過整合 Langchain 和 RAG 技術，智慧型校園導覽系統能夠根據用戶的具體查詢動態生成答案，並從校園內部知識庫或外部資源中檢索相關信息。這不僅能確保查詢結果的準確性，還能夠適應不同學生的個性化需求。隨著系統的不斷完善，它將能夠學習並記錄每個用戶的偏好，提供更精確的個性化建議，從而提升用戶的滿意度和系統的使用率。

## 1.2.6 學術研究與實際應用的結合

「中原大學嚮導」專題研究不僅僅是一次技術實現的嘗試，它也是將前沿人工智慧技術應用於教育場景中的一次學術探索。作為一個基於 Langchain 和 RAG 的智慧聊天機器人系統，它結合了自然語言處理、信息檢索、深度學習等多個領域的技術，這為學術研究提供了豐富的素材。

首先，該系統可以成為研究現代語言模型如何在實際應用中運作的案例。研究者可以通過該系統觀察語言模型在不同場景下的表現，並分析其優勢和不足。其次，該系統也為信息檢索與生成模型的結合應用提供了一個具體實驗場景，這對於檢索增強生成技術的發展具有重要意義。

另外，這樣的研究也能夠為教育管理的數字化轉型提供技術支持。隨著校園管理的日益數字化，傳統的人工服務模式正在逐漸被技術手段所取代。透過本研究，我們可以探討如何利用智能聊天機器人技術，實現更加高效的校園信息管理和服務。

## 1.2.7 提升用戶體驗與互動的可能性

在現代科技的推動下，互動性和用戶體驗已經成為技術應用中的關鍵因素。尤其是對於需要與人進行頻繁交互的系統來說，如何提升用戶的體驗是技術設計的重點。智慧

聊天機器人具備的高交互性為此提供了可能性。

「中原大學嚮導」不僅是一個被動回應用戶問題的系統，它還可以主動引導用戶進行多輪對話，並提供有價值的建議。通過分析用戶的查詢習慣，系統可以在適當的時機提出相關的問題，幫助用戶更快地找到答案，並且能夠為用戶推薦其他可能感興趣的校園信息。

此外，隨著語音識別技術的進步，未來的校園導覽系統甚至可以實現語音交互，讓用戶在不需要鍵盤輸入的情況下完成所有查詢和對話，這將進一步提高用戶體驗。

## 1.2.8 資訊管理與數據安全的挑戰

在開發這類智慧系統時，數據安全和資訊管理的問題始終是不可忽視的挑戰。智慧聊天機器人需要處理大量的用戶查詢數據，這些數據可能涉及個人隱私或學校內部敏感信息。因此，如何確保用戶數據的安全性，並符合相關法律法規，是系統開發中的一個重要考量。

為此，系統需要設計完善的數據保護機制，並對查詢數據進行匿名化處理，確保用戶的隱私不會被洩露。隨著 AI 技術的發展，對數據安全的需求將越來越高，因此，在本專題中，研究如何在保證數據安全的前提下實現高效的信息查詢和導覽服務，是一

個具有挑戰性但非常重要的議題。

## 1.3 研究問題

### 1.3.1 如何確保智能聊天機器人對於多樣化查詢的準確性？

隨著語言模型的進步，現代聊天機器人已經具備了一定的語言理解和生成能力。然而，對於一個校園導覽系統來說，查詢內容的多樣性是一個不可忽視的挑戰。不同的用戶可能會對同一個問題使用不同的表達方式，甚至一些問題可能包含模糊或不完整的信息。因此，如何確保「中原大學嚮導」系統能夠準確地處理各種不同表述的查詢，是一個重要的研究問題。

在處理這個問題時，我們需要考慮以下幾點：

- 1.系統能否正確識別不同的語義並理解用戶的真正意圖？
- 2.如何利用語言模型（如 GPT 或 BERT）的自然語言理解能力來應對模糊查詢？
- 3.如果系統對查詢內容不確定，能否提供合理的反問來澄清用戶需求，從而避免錯誤回答？

為了解決這些問題，研究者可以探討多種自然語言處理技術的應用，如語義分析、上下文理解等。同時，也可以考慮引入多輪對話來改善查詢的準確性，從而保證用戶

能夠得到準確且有價值的回答。

### 1.3.2 如何有效地將外部數據源與語言模型結合，實現實時信息檢索？

RAG ( Retrieval-Augmented Generation ) 技術的應用使得聊天機器人能夠結合外部數據源進行實時的知識檢索。對於「中原大學嚮導」系統來說，這是一個關鍵的功能，因為校園內部信息經常更新，如課程安排、活動通知等。因此，如何在系統中有效地將外部數據源與語言模型結合，實現即時的準確信息檢索，是需要深入探討的研究問題。

為了回答這個問題，系統需要解決以下幾個技術挑戰：

- 1.如何選擇合適的數據來源來保證系統能夠獲取到最新、最相關的信息？
- 2.如何確保檢索到的外部數據能夠與生成模型無縫結合，從而提供準確且符合上下文的回覆？
- 3.系統如何處理由外部數據源檢索來的冗餘信息或矛盾信息，並生成一個一致且準確的回答？

此研究問題涉及到對數據庫設計、索引技術以及如何與語言模型整合的深入理解。

我們可以在此問題上進行實驗，評估不同數據源的質量以及語言模型的檢索能力。

### 1.3.3 系統如何應對多輪對話中的上下文理解與持續對話問題？

智慧聊天機器人通常需要處理多輪對話，而每一輪對話往往與上一輪的內容相關聯。這意味著「中原大學嚮導」系統在回答用戶的問題時，需要能夠記住並理解對話上下文，從而在多輪對話中給出連貫的回覆。如何讓系統保持上下文的一致性，並在多輪對話中正確回應，是一個關鍵的研究問題。

具體而言，我們需要探討以下幾個問題：

1. 語言模型如何有效地記住並處理對話中的上下文信息，避免因為過多的信息丟失而導致的誤解？
2. 在多輪對話中，如何判斷何時應當回溯到之前的對話內容來提供準確的回答？
3. 當對話中的上下文變得複雜時，如何引導用戶簡化問題，並確保系統能夠處理？

為了解決這些問題，系統可能需要設計一個持續對話記憶機制，來保持對話的上下文信息。我們還可以探討如何使用框架來分段處理上下文，使系統能夠在每一輪對話中清晰地識別上下文的變化。

### 1.3.4 如何提升系統在不同場景中的可用性和反應速度？

用戶體驗是評估一個智慧系統成功與否的關鍵指標之一。對於「中原大學嚮導」這

樣一個面向校園場景的智慧系統而言，系統的可用性和反應速度直接影響著用戶的滿意度。如果系統的回應時間過長或是在特定情境下無法正確作答，將大大降低其實用性。因此，如何在確保回覆準確的同時，提升系統的反應速度是一個重要的研究問題。

這一問題可以從以下幾個方面探討：

- 1.系統在不同場景（如校園導覽、活動查詢、設施導覽等）中的表現如何？是否存在某些場景下回覆效率低下或無法應對的情況？
- 2.如何優化數據檢索和生成的過程，縮短系統的回應時間？
- 3.是否可以設計一個自適應算法，使系統能根據查詢的複雜程度來決定是進行深度檢索還是生成較為簡單的回覆？

在解決這個問題時，研究者可以對系統進行性能測試和壓力測試，以了解系統在不同查詢負荷下的表現，並採取相應的優化策略來提升系統的效率 and 穩定性。

### **1.3.5 如何保證系統對於用戶個性化需求的適應性？**

每一位用戶的需求是不同的。對於一個智能導覽系統來說，能否根據每位用戶的個性化需求提供定制化的回答是衡量其成功與否的重要標準之一。例如，新生可能更關

注校園設施的分佈，而一位老教授可能更關注校內科研資源的最新動態。因此，如何保證系統能夠根據不同用戶的特定需求進行個性化回覆，是一個重要的研究問題。

這一問題涉及到以下幾個方面：

- 1.系統如何根據不同用戶的偏好或歷史查詢記錄進行個性化調整？
- 2.如何設計一個動態的回答生成機制，使系統能夠根據不同類型的查詢內容進行優化處理？
- 3.當用戶需求發生變化時，系統是否能夠自我學習並調整回答策略，以適應不同的查詢場景？

解決這一問題需要研究者深入了解個性化推薦算法，並考慮將其與語言模型相結合。通過分析用戶查詢行為數據，系統可以更好地理解不同用戶的需求，並提供定制化的回覆。

## 1.4 研究目的

研究目的主要集中在開發一個基於人工智慧 ( AI ) 技術的智慧校園導覽系統，利用自然語言處理 ( NLP ) 與檢索增強生成 ( RAG ) 技術來提升校園資訊查詢的效率和準確性。具體而言，本研究的目的是透過先進的 AI 技術提供動態、即時且個性化的校園

導覽服務，從而改善學生、教職員工和訪客的校園體驗。

首先，該研究的核心目的是開發一個智能化系統，使其能夠即時回應用戶的多樣化查詢。傳統的校園導覽方式往往局限於靜態的信息展示，而智能導覽系統將透過 AI 技術實現即時的回答。這不僅能夠提升用戶在校園中的自主性，還能減少他們在尋找信息時的時間成本。

其次，本研究的另一個重要目的在於探索 NLP 技術在校園導覽中的應用潛力，特別是語言模型在處理多輪對話、模糊查詢和語義理解中的表現。這將幫助研究者理解如何進一步優化語言模型，以應對具體領域內的專業需求。

最後，檢索增強生成技術的整合也是本研究的重要目標之一。該技術不僅能夠動態檢索來自外部數據源的最新信息，還能根據查詢內容生成個性化的回答，解決傳統系統無法應對動態數據的局限性。通過 RAG 技術的應用，本研究將提供更靈活、準確的導覽服務，進一步推動校園信息化管理的發展。

總的來說，本專題的研究目的是通過技術創新，提升校園導覽系統的智能程度，並為校園內不同群體提供精準、即時的信息查詢服務，進而促進校園數位化服務的進一步發展。

表一 本實驗使用的工具

工具	功用
Langchain	使開發者更容易、更有效地利用大型語言模型
RAG	擷取系統從資料庫或一組文件中提取相關信息，然後由語言模型使用這些信息來生成回應。

## 第二章 研究方法

### 2.1 文獻回顧

智慧導覽系統逐漸成為校園和公共場所中常見的應用。2018年，李勝凱碩士的製作了一款聊天機器人，此聊天機器人主要可詢問南華大學資管系的資訊，讓學生能夠透過其了解到有關資管系的基本資訊、師資等[4]，這個聊天機器人的優點為使用者一定可以查詢到想要找的問題，缺點為無法回答得更为簡單且精確的事情。



圖 1 南華資管聊天機器人的起始畫面

2002 年，沈天佐碩士指出[5]，問答系統 ( Question Answering System ) 可以簡單地理解為一種系統，它能接收使用者提出的自然語言問題，並從一組文件中檢索出相應的答案。因此，當使用者想要了解某個問題的答案時，透過這個問答系統，就能直接獲得答案，而無需自己逐一查看資訊檢索系統返回的相關文件來尋找答案。

2016 年，劉秉鈞碩士提到[6]，Eliza 程式的設計理念主要以模仿為核心。第一代的設計模仿醫生，創建了一個虛擬的診療系統，讓電腦充當心理醫生。這個系統通過向使用者 ( 病人 ) 提出問題並根據其回答進行簡單的樣式比對，從而營造出電腦與人之間的對話感。

2017 年，彭昱傑碩士指出[7]，聊天機器人的應用可以分為開放領域 ( Open-

domain) 和封閉領域 ( Closed-domain ) 。封閉領域的聊天機器人專注於特定主題的對話，例如新聞、天氣或電子商務。而開放領域的聊天機器人則能處理任何主題，並努力提供相應的回答。

2024 年，王薇婷碩士使用 LangChain 框架[8]，串接 OpenEdu 資料庫，設計了一個自然語言查詢介面，透過生成 SQL 進行數據檢索。研究探索了順序式查詢與 Agent 查詢的不同方法，並分析了其準確性和錯誤類型，結果顯示，加入表格與欄位描述的順序式查詢達到 85.7% 的準確率，顯示 LLM 在教育數據分析中的潛力。

2023 年，Ballapuram, Ravi Kumar 指出[9]，檢索增強生成 ( RAG ) 模型在自然語言處理中顯示出良好的潛力，但在特定領域 ( 如半導體產業 ) 的檢索效率和準確性仍存在不足。為改善這一問題，本研究提出了結合領域專屬本體的方法，利用本體結構與語義關係以提升資訊檢索的效果。此外，詞嵌入技術如 owl2vec 也進一步增強了查詢的精準度。實驗結果表明，本體增強的 RAG 模型在半導體研究中提高了檢索準確性和知識整合能力，展現出在科學技術創新中的應用潛力。

## 2.2 Langchain

LangChain 是一個專門用來開發語言模型相關應用的框架，旨在讓開發者能夠更容易、更有效地利用大型語言模型 ( LLM ) 來增強現有系統的功能。隨著自然語言處理和生成式 AI 的發展，LLM 已經在許多領域中被廣泛應用，例如對話系統、文本生

成、知識檢索等。LangChain 的目的是提供一個靈活且功能強大的工具集，讓開發者可以將這些模型應用到他們的特定需求上，例如創建自定義的對話機器人、問答系統或其他需要語言理解的應用。

## 2.2.1 LangChain 的基本架構

LangChain 的架構可以分為三個主要部分：文檔分割、LLM 建立和檢索系統。這些部分共同作用，以實現從大量資料中進行高效的檢索並提供準確的回應。

1. 文檔分割：這一步是指將輸入的文章或文檔進行分割，將其整理成適合語言模型處理的格式。由於 LLM 無法直接處理過於龐大的文檔，因此需要對資料進行適當的切割和整理。這個過程能夠幫助模型更好地理解資料的結構，並在需要檢索信息時提供更準確的答案。常見的分割方式可以根據段落、句子甚至是主題來進行，這樣的分割策略可以根據應用需求進行調整。

2. 建立 LLM：在這個步驟中，開發者需要設定好語言模型的架構，這包括選擇使用的模型、設定提示詞 ( prompt )，以及定義應用場景。語言模型本身是非常強大的工具，但它的效能很大程度上取決於如何設計和設定提示詞。根據應用的不同，提示詞的設計可以是引導模型生成文本，回答特定問題，或是進行語義上的推理。因此，設計合適的提示詞是確保 LLM 能夠準確理解任務的關鍵。

3. 檢索系統：檢索系統負責將文檔分割和 LLM 建立的結果結合起來。當使用者提出問

題時，檢索系統會在已整理好的文檔中搜尋相關的資料，並將這些資料餵給語言模型，讓模型生成相應的回應。這個過程稱為檢索增強生成 ( Retrieval-Augmented Generation, RAG )，它能夠確保模型生成的答案具有基於實際數據的依據，而不是僅憑模型的內部知識進行推測。這種結合方式使得系統在面對複雜問題時，能夠提供更加準確且上下文相關的回應。

### 2.2.2 LangChain 的應用

LangChain 的主要應用場景之一是問答系統。當使用者提出問題時，這個系統能夠快速地檢索相關資料，並使用語言模型生成準確的回答。這在教育、客服、技術支持等需要快速回答問題的領域中具有廣泛的應用前景。通過結合 LLM 的語言生成能力和檢索系統的資料檢索能力，LangChain 可以應用於各種需要即時生成文本的情境。

例如，在我們所開發的"中原大學嚮導"專題中，LangChain 可以幫助構建一個智慧的聊天機器人，為學生、訪客等提供即時的學校資訊。透過文檔分割，機器人能夠快速整理學校的規章制度、課程介紹、交通資訊等文檔；而透過檢索系統，當使用者詢問問題時，系統可以在這些資料中尋找答案，並使用 LLM 生成一個流暢且準確的回應。

### 2.2.3 LangChain 的技術優勢

LangChain 的一大技術優勢是它的模組化設計，這意味著開發者可以根據自己的需

求選擇不同的模組來組裝自己的應用。這樣的靈活性允許它適應各種不同的應用場景，無論是基於網頁的聊天機器人，還是嵌入到現有應用中的智慧助手，LangChain 都能提供相應的支持。此外，LangChain 的檢索增強生成技術也大大提升了語言模型的實用性，尤其是在需要準確回答問題的應用中，它能夠提供比純粹依賴語言模型內部知識更可靠的答案。

## 2.3 RAG

RAG ( Retrieval-Augmented Generation，擷取增強生成 ) 是一種結合了擷取系統與神經語言模型的技術，旨在從大量資料中擷取相關訊息，並利用語言生成模型來提供精確且流暢的回答。這項技術的興起解決了許多大型語言模型 ( LLM ) 在實際應用中的限制，尤其是在知識更新和正確性方面。RAG 系統能夠利用外部知識庫中的資料，讓語言模型的生成結果更具參考價值和即時性，適用於各種需要精確回答的場景。

### 2.3.1 RAG 的運作原理

RAG 的運作流程可分為兩個主要步驟：擷取 ( Retrieval ) 與生成 ( Generation ) 。

1. 擷取 ( Retrieval )：首先，當使用者提出問題時，系統會從一個大型的知識庫中擷取出與該問題最相關的前幾個文檔或片段。這些知識庫可以是靜態的、動態更新的，或是根據具體應用需求來進行定制。擷取系統負責從這些資料中挑選出最有用的部分，這一步驟相當於為語言模型提供準確的上下文背景。

2. 生成 ( Generation ) : 接下來, 語言模型根據擷取到的資訊進行回答。這個過程中, 模型不僅僅是重複文檔中的文字, 而是將這些資訊轉化為流暢的自然語言回應, 確保回答的可讀性和邏輯性。這使得 RAG 能夠比單純的語言生成模型提供更加準確且上下文相關的答案。

### 2.3.2 RAG 的優勢

RAG 的優勢主要體現在其對於外部知識的整合能力, 這讓系統在應對複雜問題時, 能夠提供具有依據且最新的答案。以下是 RAG 的幾大優勢:

1. 提高答案的準確性: 由於 RAG 能夠將外部資料庫中的知識整合到回應中, 因此答案的準確性明顯提升。這有效地減少了語言模型生成虛假訊息的風險, 特別是在需要專業知識或複雜問題時, RAG 透過從可信資料中擷取相關內容, 讓回應更具可信度。
2. 即時性與更新性: LLM 本身依賴於訓練資料, 而這些資料在模型訓練完成後可能會過時。但透過 RAG 的擷取功能, 系統能夠使用最新的資料來生成回應, 這意味著使用者能夠獲得及時的資訊。這對於處理快速變化的領域 ( 如新聞、科技 ) 來說至關重要。
3. 高透明度與可核實性: RAG 提供了回應所引用的資料來源, 這讓使用者能夠進一步核實答案的準確性。這種透明度增強了使用者對系統輸出結果的信任, 特別是在涉及到關鍵決策或精確回答的情況下。

4. 客製化與領域專業知識支援：RAG 可以根據具體領域的需求進行客製化。透過針對不同領域的資料庫進行擷取，系統能夠提供高度專業化的回應。例如，在醫療、法律等需要專業知識的場景中，RAG 能夠從相關文獻或專業資料庫中獲取資訊，並提供準確的建議或解答。

5. 安全性與隱私保護：相比於直接微調 ( Fine-tuning ) 模型，RAG 的資料使用更加安全。資料不需要儲存在模型中，而是儲存在知識庫內，這讓資料的管理和控制變得更為靈活，特別是在處理敏感資訊或企業內部資料時，RAG 提供了更好的安全性和隱私保護措施。

6. 經濟效益與效率：RAG 的另一個重要優勢在於它不需要頻繁地更新模型參數。由於擷取系統可以隨時使用最新的資料，因此不需要像其他方法那樣頻繁對語言模型進行重新訓練或更新，這在處理大規模資料集時，能夠顯著節省成本與資源。

### 2.3.3 RAG 的應用場景

RAG 的應用場景十分廣泛，尤其在需要處理大量資訊並提供精確回應的情況下。

以下是幾個典型應用場景：

1. 問答系統：在企業客服系統或技術支持中，RAG 可以用來即時回答客戶的問題。當客戶詢問某些產品的技術細節時，RAG 能夠從技術文件中提取相關資訊，並生成一個

簡潔、易懂的回應。

2. 教育領域：在教育應用中，RAG 可以用來解答學生的問題，特別是在查詢學術資料、課程內容或某些具體領域知識時，RAG 能夠通過擷取相關文獻或教材來提供具體的解釋。

3. 醫療診斷輔助：醫療領域中，RAG 可以從醫學文獻或臨床指南中擷取出最新的研究結果，並為醫生提供參考意見或輔助診斷，這大大提升了醫療決策的精確性。

4. 新聞與資訊更新：RAG 在新聞領域的應用也很廣泛，當使用者查詢最新的新聞事件時，系統能夠從最新的報導中擷取資訊，並生成符合事實的回應，確保資訊的時效性。

### 2.3.4 RAG 的未來展望

隨著人工智慧技術的不斷發展，RAG 作為一種結合了資料擷取與生成技術的框架，將會在更多領域中發揮重要作用。隨著知識庫的擴展與擷取技術的進步，RAG 將能夠提供更快、更準確的回答，並應用於更加複雜的場景中。此外，在隱私保護與安全性方面，RAG 也具有良好的潛力，特別是在處理敏感資料時，RAG 的資料控制能力將成為企業和用戶的理想選擇。

總的來說，RAG 為語言模型提供了一種有效的增強手段，透過結合擷取系統與語

言生成模型，能夠在不同領域中提供專業化的、高效且準確的回應。

## 第三章 研究步驟與結果

### 3.1 研究與開發步驟

#### 3.1.1 HTML 檔案 文件結構與基本設定

```
!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>中原大學報你知</title>
```

- <!DOCTYPE html> 宣告這是 HTML5 文件。
- <html lang="en"> 指定文件語言為英文。
- <meta charset="UTF-8"> 設定字符編碼為 UTF-8，支持多語言字符，確保中文字能正常顯示。
- <meta name="viewport" content="width=device-width, initial-scale=1.0"> 設定視窗寬度，以便在不同裝置上更好地適配，提升響應式設計效果。
- <title>中原大學報你知</title> 設定網頁標題，在瀏覽器標籤中顯示為「中原大學報你知」。

#### 3.1.2 外部資源連結

```
link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css">
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
<link rel="stylesheet" href="style_test.css">
```

- 第一行引入 Font Awesome 圖示庫，以使用各種圖示，例如月亮和飛機圖示。
- 第二行引入 Bootstrap 4.5，提供排版、按鈕和表單等的樣式支持，使布局更簡便和美觀。
- 第三行連結了 style\_test.css，可以自定義 CSS 樣式來覆蓋或擴展上述外部資源，並設置特定樣式。

### 3.1.3 網頁主體結構 (<body>)

```
<body>
<div class="background"></div> <!-- 背景元素 -->
<div class="container">
  <div class="chat-container">
    ...
  </div>
</div>
<script src="script_test.js"></script>
</body>
</html>
```

- <div class="background"></div> 用於顯示背景樣式，具體效果應在 style\_test.css 中定義。

- `<div class="container">` 使用 Bootstrap 的 `container` 類別，使內容在較大屏幕上居中並自動適應屏幕大小。

- `<script src="script_test.js"></script>` 引入 JavaScript 文件 `script_test.js`，處理互動功能，例如發送訊息、切換主題等。

### 3.1.4 聊天容器 (`<div class="chat-container">`)

```
<div class="chat-container">
  <div class="chat-header">
    
    <h2>中原大學報你知</h2>
    <button class="theme-toggle" id="theme-toggle">
      <i class="fas fa-moon"></i> 切換主題
    </button>
  </div>
  <div class="chat-box" id="chat-box">
    <!-- 對話內容將動態插入 -->
  </div>
  ...
</div>
```

- `<div class="chat-header">` 包含聊天室的標題、圖片和主題切換按鈕。

- `` 顯示標題圖片，替代文字為「Chat Header Image」，放在根目錄或指定位置。

- `<h2>中原大學報你知</h2>` 為聊天室標題。

- <button class="theme-toggle" id="theme-toggle"> 主題切換按鈕，內含月亮圖示，點擊後切換深淺主題。

- <div class="chat-box" id="chat-box"> 主要對話框，內容將透過 JavaScript 動態插入。

### 3.1.5 輸入區域 (<div class="input-area">)

```
<div class="input-area">
```

```
  <input type="text" id="user-input" placeholder="請輸入您的問題..." class="form-control">
```

```
  <button id="send-btn" class="btn btn-primary"><i class="fas fa-paper-plane"></i></button>
```

```
  <button id="voice-btn" class="btn btn-secondary"><i class="fas fa-microphone"></i></button>
```

```
</div>
```

- <input type="text" id="user-input" ...> 文本輸入框，供用戶輸入問題。placeholder 設定提示文本，form-control 來自 Bootstrap，提供基本樣式。

- <button id="send-btn" class="btn btn-primary"><i class="fas fa-paper-plane"></i></button>

文字發送按鈕，帶有紙飛機圖示。JavaScript 中應綁定點擊事件來發送消息。

- <button id="voice-btn" class="btn btn-secondary"><i class="fas fa-

microphone"></i></button> 語音輸入按鈕，帶有麥克風圖示，可能用於錄音或啟動語音

識別。

### 3.1.6 加載動畫 (<div id="loading" class="loading">)

```
<div id="loading" class="loading">
  <i class="fas fa-spinner fa-spin"></i> 正在加載...
</div>
```

•這個區域顯示「正在加載」提示。<i class="fas fa-spinner fa-spin"></i> 使用 Font Awesome 的旋轉圖示來呈現加載效果。當系統等待回應時可顯示此區域，平時則隱藏。

### 3.1.7 JavaScript (script\_test.js)

JavaScript 文件 script\_test.js 主要處理聊天室互動邏輯。此文件包含：

- 發送訊息：監聽發送按鈕和輸入框事件。
- 主題切換：監聽 theme-toggle 按鈕，切換網頁主題。
- 語音輸入：監聽 voice-btn 按鈕以啟用語音輸入。
- 動態對話框更新：在 chat-box 中插入用戶和機器人的對話內容。

## 3.2 研究設置方法

### 3.2.1 CSS 檔案全局設置

```
* {
  box-sizing: border-box;
}
```

- box-sizing: border-box; 設置所有元素的寬度和高度包含內邊距與邊框，方便控制元素

大小。

### 3.2.2 body 的樣式

```
body {  
  font-family: 'Arial', sans-serif;  
  margin: 0;  
  padding: 0;  
  display: flex;  
  justify-content: center;  
  align-items: flex-end;  
  height: 100vh;  
  position: relative;  
  overflow: hidden;  
}
```

- 設定字體為 Arial，並清除 margin 和 padding，使網頁不受瀏覽器預設樣式的影響。
- display: flex; justify-content: center; align-items: flex-end; 則將內容水平置中，並靠近頁面底部對齊。
- height: 100vh; 讓 body 高度填滿整個視窗。
- overflow: hidden; 隱藏超出視窗範圍的內容。

### 3.2.3 背景層 background

```
.background {  
  position: absolute;  
  top: 0;  
  left: 0;  
  width: 100%;
```

```
height: 100%;
background: linear-gradient(135deg, #74ebd5, #acb6e5);
z-index: 1;
}
```

- 背景使用漸層效果，並設置為全螢幕的絕對定位，固定在網頁的最底層。

### 3.2.4 容器 container

```
.container {
width: 100%;
height: 100%;
margin: 0;
position: relative;
z-index: 2;
padding: 0;
}
```

- container 佔滿整個頁面，用於放置聊天室的主要內容，並確保內容在背景之前顯示。

### 3.2.5 聊天容器 chat-container

```
.chat-container {
background-color: rgba(255, 255, 255, 0.85);
border-radius: 20px;
box-shadow: 0 8px 30px rgba(0, 0, 0, 0.2);
padding: 20px;
backdrop-filter: blur(10px);
transition: all 0.3s ease;
display: flex;
flex-direction: column;
height: 100%;
justify-content: space-between;
}
```

- `chat-container` 是聊天框的主要區域，使用 `rgba(255, 255, 255, 0.85)` 半透明的背景。
- `backdrop-filter: blur(10px);` 添加模糊效果，帶來半透明玻璃質感。
- `display: flex; flex-direction: column;` 讓聊天內容和輸入區域呈垂直排列。
- `height: 100%; justify-content: space-between;` 保證內容填滿可用空間，並使聊天框與輸入區域在垂直方向上分開。

### 3.2.6 聊天框 `chat-box`

```
.chat-box {  
  background-color: #f9f9f9;  
  height: 400px;  
  overflow-y: scroll;  
  padding: 10px;  
  border-radius: 10px;  
  margin-bottom: 15px;  
  border: 1px solid #ced4da;  
}
```

- `chat-box` 是聊天訊息顯示區域，設置為 `400px` 高度並允許垂直滾動，讓訊息能夠顯示更多。
- `border-radius: 10px;` 添加圓角效果，使設計更柔和。

### 3.2.7 消息 `message`

```
.message {  
  padding: 10px;  
  margin: 5px 0;
```

```
border-radius: 10px;
max-width: 80%;
display: flex;
align-items: center;
}
```

```
.user-message {
  background-color: #007bff;
  color: white;
  text-align: right;
  margin-left: auto;
}
```

```
.bot-message {
  background-color: #e2e3e5;
  color: black;
  text-align: left;
}
```

• `.message` 是每條消息的基礎樣式，包含內邊距、圓角和最大寬度限制，讓每條消息不會超過聊天室的 80% 寬度。

• `.user-message` 為用戶消息設置藍色背景並靠右對齊，`.bot-message` 則將機器人消息設置為灰色背景並靠左對齊。

### 3.2.8 輸入區域 `input-area`

```
.input-area {
  display: flex;
  gap: 10px;
  margin-left: 10%;
  margin-right: 10%;
  margin-bottom: 20px;
}
```

- `.input-area` 用於輸入框和按鈕，設定 `gap: 10px`; 為子元素間添加間距。
- `margin-left: 10%; margin-right: 10%`; 將左右邊界留出空間，使設計更居中且不緊貼邊緣。

### 3.2.9 暗黑模式 `.dark-theme`

```
.dark-theme {  
  background-color: #121212;  
}  
  
.dark-theme .chat-container {  
  background-color: #2a2a2a;  
}  
  
.dark-theme .chat-box {  
  background-color: #3b3b3b;  
}  
  
.dark-theme .user-message {  
  background-color: #1e90ff;  
}  
  
.dark-theme .bot-message {  
  background-color: #444444;  
}  
  
.dark-theme h2, .dark-theme .theme-toggle {  
  color: white;  
}
```

- `dark-theme` 類別為暗黑模式設定，主要調整背景和文字顏色，使之適合低亮度環境。

- .dark-theme .chat-container 和 .dark-theme .chat-box 調整為深色系背景。
- .user-message 和 .bot-message 在暗黑模式下使用較亮的藍色和灰色來區分消息。

### 3.2.10 頭像圖片 .header-image

```
.header-image {  
  width: 50px;  
  height: auto;  
  margin-right: 10px;  
  border-radius: 5px;  
}
```

- .header-image 設置頭像寬度 50px，height: auto; 保持比例縮放，並加上圓角效果讓圖片邊角更柔和。

### 3.2.11 響應式設計

```
@media (max-width: 768px) {  
  .chat-container {  
    width: 100%;  
    padding: 10px;  
    margin: 0;  
    border-radius: 0;  
  }  
  .chat-box {  
    height: 50vh;  
  }  
  .input-area {  
    flex-direction: column;  
    gap: 5px;  
    margin-left: 0;  
    margin-right: 0;  
  }  
}
```

```
#user-input, #send-btn, #voice-btn {
  width: 100%;
  margin: 0;
}
}
```

- 在小於 768px 的螢幕上，對佈局進行調整：
- chat-container 滿版，取消圓角。
- 將輸入框和按鈕排列為縱向佈局，並調整間距。

### 3.3 程式相關設定

#### 3.3.1 Javascript 初始化與歡迎訊息

```
window.onload = function() {
  setTimeout(function() {
    addMessage('bot', '哈囉！可以問我任何關於中原大學的問題 ~ 😊😊');
  }, 2000);
};
```

- window.onload 在頁面加載完成後執行。
- setTimeout 延遲 2 秒後調用 addMessage 函式，向使用者顯示歡迎訊息（模擬加載效果）。

#### 3.3.2 設定按鈕與 Enter 鍵的事件監聽器

```
document.getElementById('send-btn').addEventListener('click', sendMessage);
document.getElementById('user-input').addEventListener('keydown', function (event) {
```

```
    if (event.key === 'Enter') {
      sendMessage();
    }
  });
```

- 監聽 send-btn 按鈕的點擊事件，點擊後執行 sendMessage 函式來發送訊息。
- 監聽 user-input 輸入框的 keydown 事件，當使用者按下 Enter 鍵後，也會調用 sendMessage 函式。

### 3.3.3 發送訊息的主函式 sendMessage

```
async function sendMessage() {
  const userInput = document.getElementById('user-input').value;

  if (userInput.trim() !== "") {
    addMessage('user', userInput);
    document.getElementById('user-input').value = "";

    showLoading();

    try {
      const response = await fetch('http://localhost:8000/ask', {
        method: 'POST',
        headers: {
          'Content-Type': 'application/json',
        },
        body: JSON.stringify({ question: userInput }),
      });

      const data = await response.json();

      hideLoading();

      if (response.ok) {
```

```

        addMessage('bot', data.response);
    } else {

        addMessage('bot', '伺服器回應錯誤：' + data.error);

    }
} catch (error) {

    addMessage('bot', '無法連接到伺服器，請稍後再試。');

    console.error('Error:', error);

    hideLoading();
}
}
}
}

```

- `sendMessage` 函式主要處理使用者訊息的發送流程：
- 讀取 `user-input` 的值並檢查內容是否為空。
- 使用 `addMessage` 函式顯示使用者的輸入訊息，並清空輸入框。
- 調用 `showLoading()` 顯示加載動畫。
- 使用 `fetch` 發送 HTTP POST 請求到後端 API，將使用者的問題以 JSON 格式發送。
- 回傳的 JSON 資料存於 `data`，若回應成功 (`response.ok`)，則顯示回覆訊息，否則顯示錯誤訊息。
- 若發送失敗（例如伺服器無回應），則顯示連線錯誤並隱藏加載動畫。

### 3.3.4 新增訊息到對話框的函式 `addMessage`

```

function addMessage(sender, message) {
    const chatBox = document.getElementById('chat-box');
    const messageDiv = document.createElement('div');

```

```
messageDiv.classList.add('message', sender + '-message');

const textNode = document.createTextNode(message);
messageDiv.appendChild(textNode);

chatBox.appendChild(messageDiv);
chatBox.scrollTop = chatBox.scrollHeight;
}
```

- addMessage 用來將訊息插入到聊天室視窗 (chat-box) 中。
- 根據 sender ( user 或 bot ) 設定不同的訊息樣式，並插入到對話框。
- chatBox.scrollTop = chatBox.scrollHeight; 確保對話框自動滾動至最新訊息位置。

### 3.3.5 加載動畫顯示與隱藏函式

```
function showLoading() {
  document.getElementById('loading').style.display = 'block';
}
```

```
function hideLoading() {
  document.getElementById('loading').style.display = 'none';
}
```

- showLoading 與 hideLoading 函式用於控制加載動畫的顯示與隱藏。
- 當使用者發送訊息時會顯示加載動畫，當接收到回覆時則隱藏。

### 3.3.6 發送訊息處理

```
async function sendMessage() {
  const userInput = document.getElementById('user-input').value;
```

- 獲取使用者的輸入，並判斷是否為有效內容（非空字串）。

### 3.3.7 像後端 API 發送請求

```
const response = await fetch('https://94de-113-14-188-97.ngrok-free.app/ask', {  
  method: 'POST',  
  headers: { 'Content-Type': 'application/json' },  
  body: JSON.stringify({ question: userInput }),  
});
```

- 發送 POST 請求到指定的後端 API。
- URL 使用 ngrok 公網地址。
- 傳遞的資料格式為 JSON，其中包含 question 參數。

### 3.3.8 處理伺服器回應

```
Const data = await response.json();  
hideLoading();  
  
if (response.ok) {  
  addMessage('bot', data.response);  
} else {  
  addMessage('bot', '伺服器回應錯誤：' + data.error);  
}
```

- 隱藏“加載中”動畫。
- 若伺服器成功回應，顯示回應內容；否則顯示錯誤訊息。

### 3.3.9 捕捉例外錯誤

```
} catch (error) {  
  
    addMessage('bot', '無法連接到伺服器，請稍後再試。');  
  
    console.error('Error:', error);  
    hideLoading();  
}
```

- 若請求失敗，顯示連線失敗訊息，並隱藏“加載中”動畫。

## 3.4 程式主要設定

### 3.4.1 主函式 環境設定

```
from dotenv import load_dotenv  
import os  
  
load_dotenv()  
openai_api_key = os.getenv("OPENAI_API_KEY")  
if not openai_api_key:  
  
    raise ValueError("OpenAI API 金鑰未正確設定。請確保 .env 檔案中包含  
OPENAI_API_KEY")
```

- 使用 `dotenv` 讀取 `.env` 檔案中的環境變數（特別是 OpenAI API 金鑰）。
- 程式碼在 `.env` 檔案未提供 API 金鑰時會引發錯誤提醒。

### 3.4.2 載入 PDF 文件

```
pdf_loader = PyPDFLoader("/Users/gaomenglin/Desktop/university-query-  
platform/cycu.pdf")  
documents = pdf_loader.load()
```

- 使用 PyPDFLoader 將指定的 PDF 文件載入，並提取出文字內容。
- documents 包含了該 PDF 文件的文本內容，可用於向量化處理和後續檢索。

### 3.4.3 建立向量資料庫

```
embedding = OpenAIEmbeddings(openai_api_key=openai_api_key)
vector_store = FAISS.from_documents(documents, embedding)
```

- 使用 OpenAI 的嵌入模型將文本內容轉換為向量表示。
- FAISS 向量資料庫將這些向量儲存，並提供快速的檢索能力。

### 3.4.4 設置檢索器和對話記憶

```
retriever = vector_store.as_retriever()
memory = ConversationBufferMemory(memory_key="chat_history", return_messages=True)
```

- retriever 為檢索器，幫助從向量資料庫中查找相關內容。
- ConversationBufferMemory 用於保留對話記憶，保持上下文，使應用程式可以基於先前的對話提供相關回覆。

### 3.4.5 建立問答鏈

```
qa_chain = ConversationalRetrievalChain.from_llm(
    llm=ChatOpenAI(model="gpt-4", openai_api_key=openai_api_key),
    retriever=retriever,
    memory=memory
)
```

- 使用 ConversationalRetrievalChain 結合 ChatOpenAI 和檢索器，創建一個問答鏈，並使用 GPT-4 作為語言模型來生成回覆。

### 3.4.6 問答 API 路由

```
@app.route("/ask", methods=["POST"])
def ask_question():
    data = request.json
    question = data.get("question")

    if not question:
        return jsonify({"error": "問題不能為空"}), 400

    try:
        result = qa_chain({"question": question})
        answer = result["answer"]
        return jsonify({"response": answer})
    except Exception as e:
        return jsonify({"error": str(e)}), 500
```

- 提供 /ask 路由，用於接收 POST 請求。
- 檢查請求的 JSON 資料是否包含 question 欄位，若無則回傳錯誤。
- 若有問題則使用 qa\_chain 生成回覆，並回傳回覆結果。

### 3.4.7 啟動 Flask 應用程式

```
if __name__ == "__main__":
    app.run(host="0.0.0.0", port=8000)
```

- 啟動應用程式並設定埠號，使其在指定埠上接收請求。

### 3.4.8 Ngrok

```
public_url = ngrok.connect(8080)
print(f" * Ngrok tunnel \'{public_url}\'' -> http://127.0.0.1:8080")
app.run(host="0.0.0.0", port=8080)
```

- 使用 pyngrok 在外網暴露本地伺服器。

## 3.5 研究結果與展示

開發一個校內專用的 GPT 系統，旨在模仿 ChatGPT 風格的問答 AI，以更全面、詳細的方式整理特定主題的內容資料，並且透過與 Langchain、RAG 以及網域的結合，以日常、熟悉的方式呈現給使用者。

這個系統將會是校園中的智慧助手，為學生、教師和校園管理者提供便捷、準確的資訊和支援。例如，當一個學生需要解答關於化學實驗的問題時，他可以通過此網頁向校內專用 GPT 提問，而 GPT 會根據學生提供的問題，提供相關的化學知識、實驗步驟、安全注意事項等。

此外，校內專用 GPT 還可以與學生、教師和校園管理者進行對話互動，以更好地理解他們的需求和提供更加個性化的服務。例如，當一位教師需要安排一個會議時，他可以直接與校內專用 GPT 對話，GPT 會根據教師提供的時間和地點，幫助教師安排

好會議的時間、地點以及相關的事項。

透過與網頁的結合，校內專用 GPT 不僅可以在電腦上使用，還可以在手機上隨時隨地進行操作，讓使用者能夠更加方便地使用這個系統。無論是學生在學習上遇到問題，還是教師需要管理校園事務，都可以通過此與校內專用 GPT 進行溝通和互動。

校內專用 GPT 的開發將為校園帶來更高效、更個性化和更智慧化的學習和管理環境，同時透過與網頁的結合，使得這個系統更加貼近日常生活，為使用者提供更好的服務和支援。

### 3.5.1 問題一.

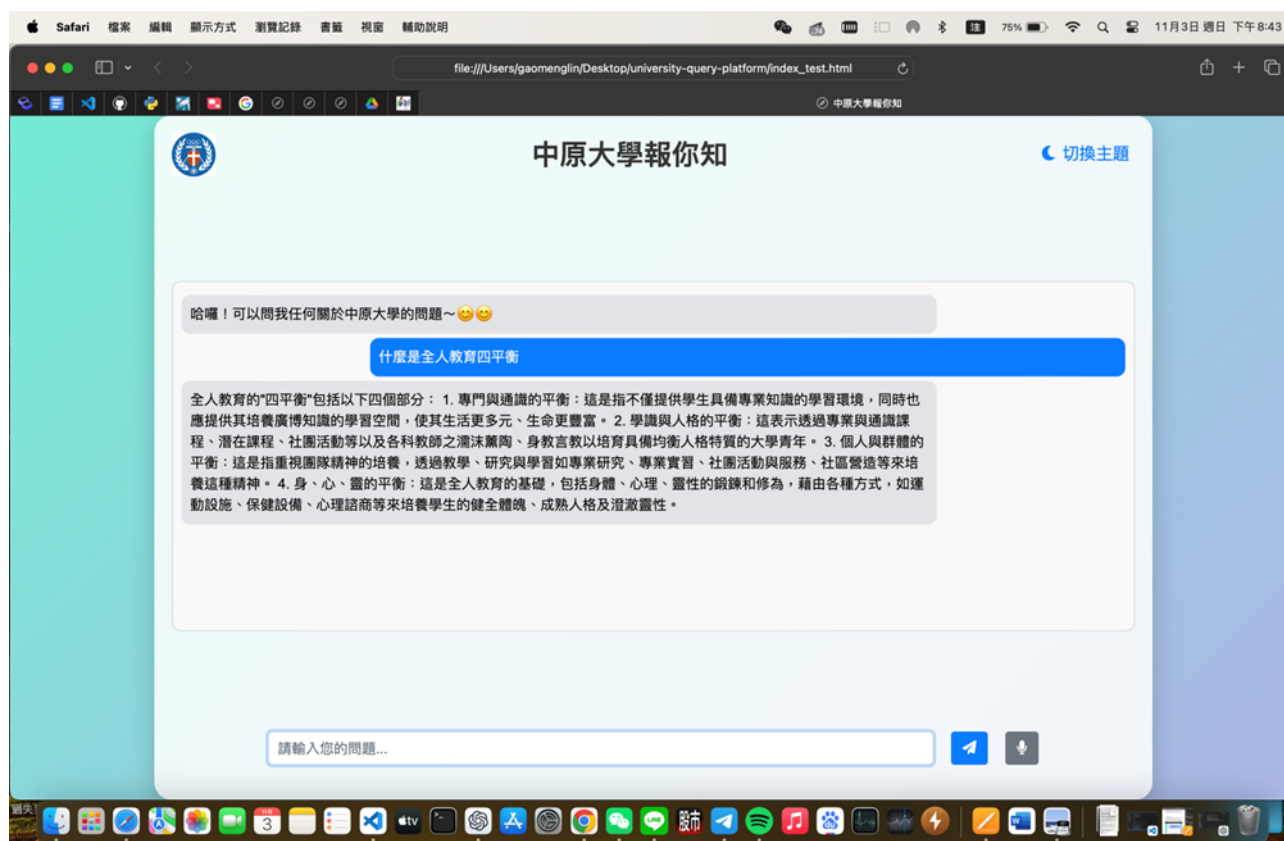


圖 2 研究結果與展示-問題一

問題：什麼是全人教育四平衡

答：全人教育的"四平衡"包括以下四個部分：1. 專門與通識的平衡：這是指不僅提供學生具備專業知識的學習環境，同時也應提供其培養廣博知識的學習空間，使其生活更多元、生命更豐富。2. 學識與人格的平衡：這表示透過專業與通識課程、潛在課程、社團活動等以及各科教師之濡沫薰陶、身教言教以培育具備均衡人格特質的大學青年。3. 個人與群體的平衡：這是指重視團隊精神的培養，透過教學、研究與學習如專業研究、專業實習、社團活動與服務、社區營造等來培養這種精神。4. 身、心、靈

的平衡：這是全人教育的基礎，包括身體、心理、靈性的鍛鍊和修為，藉由各種方式，如運動設施、保健設備、心理諮商等來培養學生的健全體魄、成熟人格及澄澈靈性。

### 3.5.2 問題二

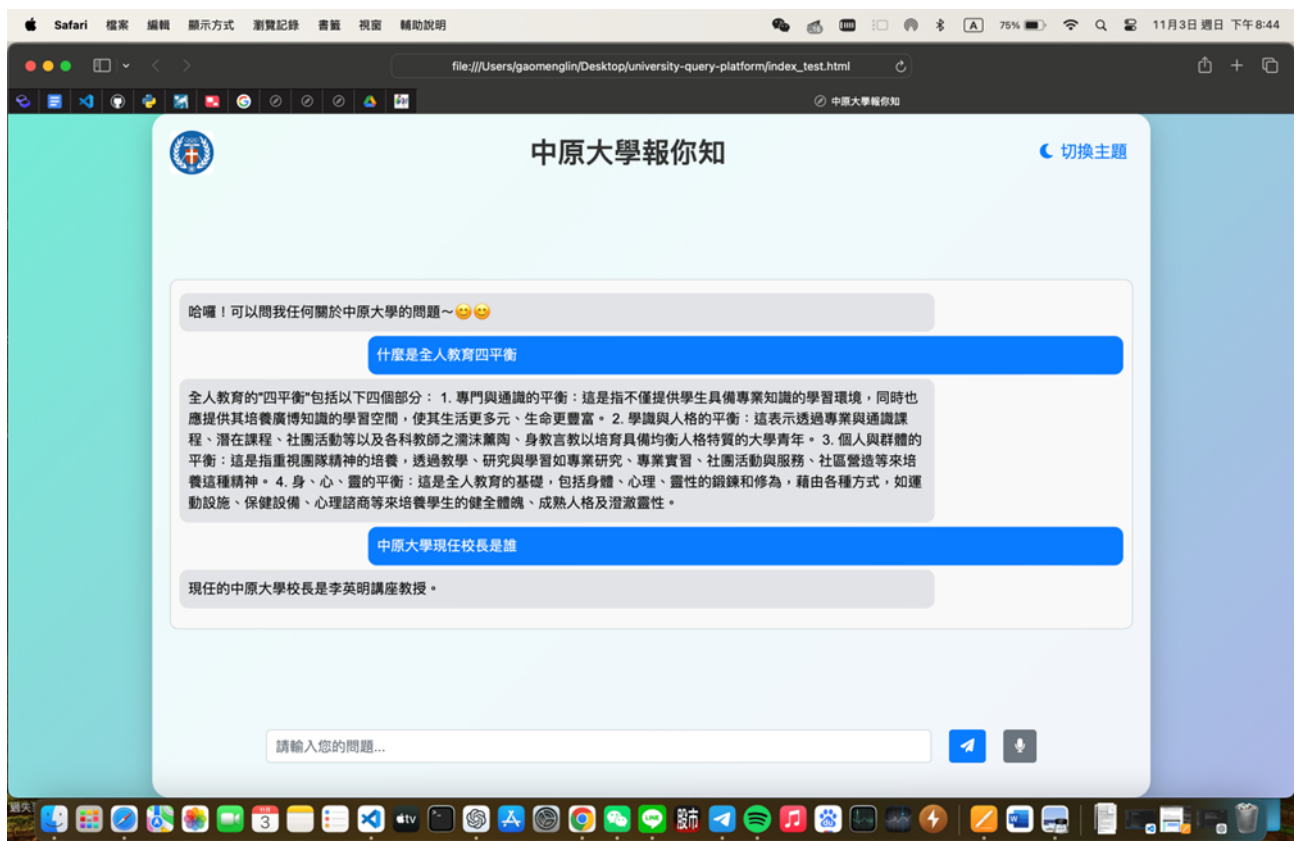


圖 3 研究結果與展示-問題二

問題：中原大學現任校長是誰

答：現任的中原大學校長是李英明講座教授。

### 3.5.3 問題三

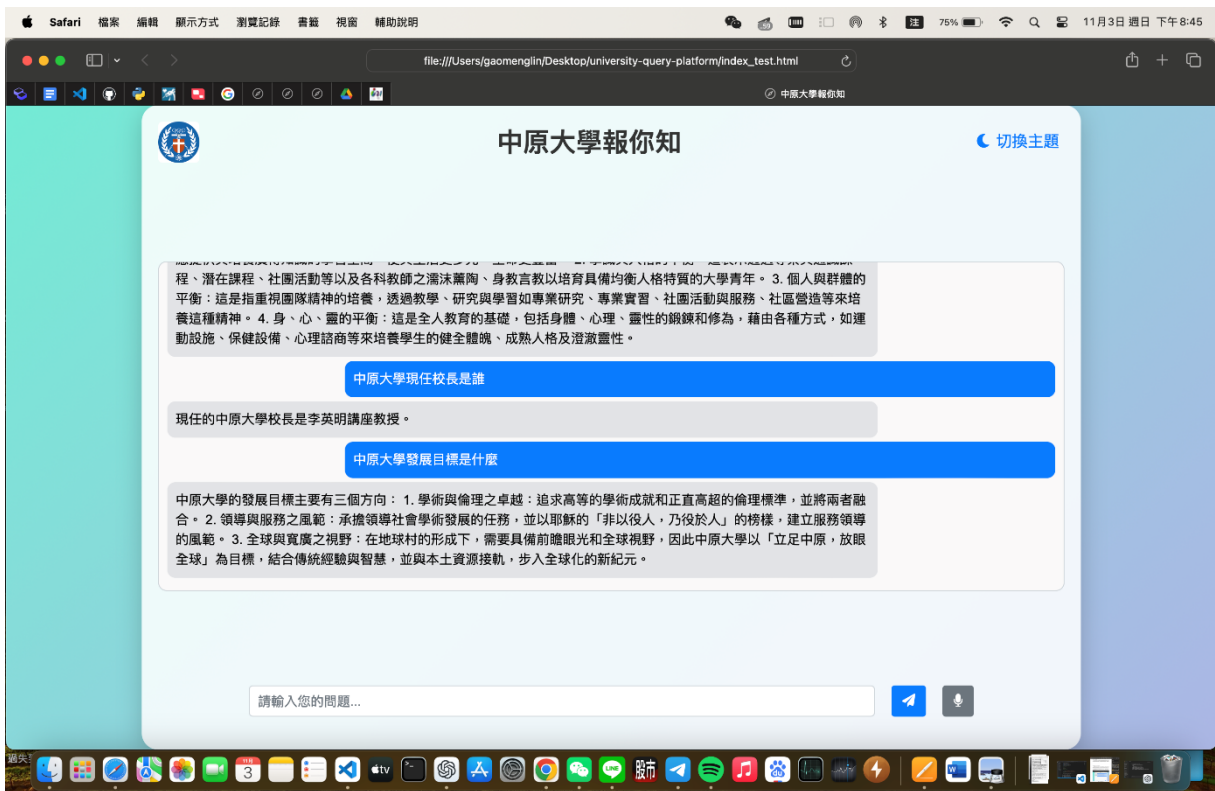


圖 4 研究結果與展示-問題三

問題：中原大學發展目標是什麼

答：中原大學的發展目標主要有三個方向：1.學術與倫理之卓越：追求高等的學術成就和正直高超的倫理標準，並將兩者融合。2.領導與服務之風範：承擔領導社會學術發展的任務，並以耶穌的「非以役人，乃役於人」的榜樣，建立服務領導的風範。3.全球與寬廣之視野：在地球村的形成下，需要具備前瞻眼光和全球視野，因此中原大學以「立足中原，放眼全球」為目標，結合傳統經驗與智慧，並與本土資源接軌，步入全球化的新紀元。

#### **3.5.4 問題四（網頁版頁面）**

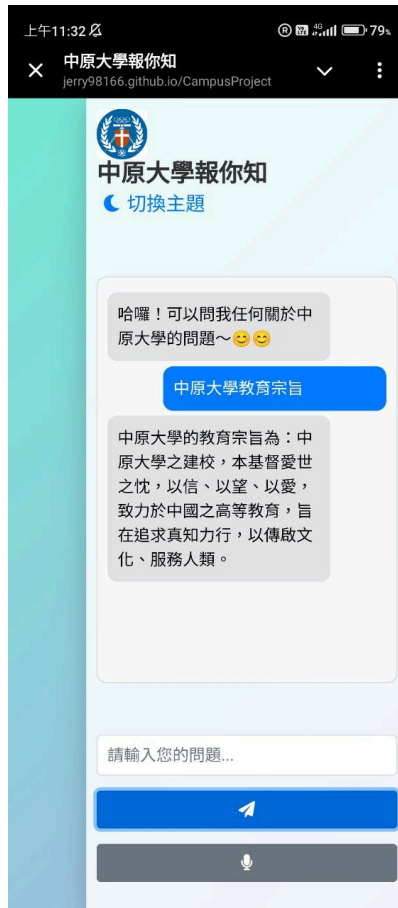


圖 5 研究結果與展示-問題四

問題：中原大學教育宗旨

答：中原大學教育宗旨為：中原大學之建校，本基督教愛世之忱，以信，以望，以愛，致力於中國之高等教育，旨在追求真知力行，以傳啟文化，服務人類。

## 3.6 研究建議與修正

### 3.6.1 VectorStore 的處理加強

PDF 文件與向量資料庫的路徑

```
pdf_path = "/Users/gaomenglin/Desktop/university-query-platform/cycu_merged.pdf"
vector_store_path = "/Users/gaomenglin/Desktop/university-query-platform/vector_store"
```

# 檢查是否有已保存的向量資料庫

```
if os.path.exists(vector_store_path):
```

```
    print("從檔案載入向量資料庫...")
```

```
    vector_store = FAISS.load_local(vector_store_path,
    OpenAIEmbeddings(openai_api_key=openai_api_key))
else:
```

```
    print("沒有找到向量資料庫，重新生成...")
```

# 載入 PDF 文件並生成向量

```
pdf_loader = PyPDFLoader(pdf_path)
```

```
documents = pdf_loader.load()
```

```
embedding = OpenAIEmbeddings(openai_api_key=openai_api_key)
```

```
vector_store = FAISS.from_documents(documents, embedding)
```

# 儲存向量資料庫

```
vector_store.save_local(vector_store_path)
```

```
print("向量資料庫已儲存至:", vector_store_path)
```

### 3.6.2 語音功能加入

```
// 確認瀏覽器是否支持 Web Speech API

if ('webkitSpeechRecognition' in window) {
  const recognition = new webkitSpeechRecognition();

  recognition.continuous = false; // 是否連續辨識

  recognition.interimResults = false; // 是否返回中間結果

  recognition.lang = 'zh-TW'; // 設定語言 ( 繁體中文 )

  const voiceBtn = document.getElementById('voice-btn');
  const userInput = document.getElementById('user-input');

  // 當點擊語音按鈕時，開始語音辨識

  voiceBtn.addEventListener('click', () => {
    recognition.start();

    voiceBtn.innerHTML = '<i class="fas fa-microphone-slash"></i> 辨識中...';

    voiceBtn.disabled = true; // 防止重複啟動
  });

  // 當辨識結果返回時，將語音轉換成文字

  recognition.onresult = (event) => {
    const transcript = event.results[0][0].transcript; // 獲取辨識到的文字

    userInput.value = transcript; // 將文字顯示在輸入框中

    recognition.stop();

    voiceBtn.innerHTML = '<i class="fas fa-microphone"></i> 語音輸入';
  }
}
```

```

        voiceBtn.disabled = false; // 重新啟用按鈕
    };

    // 當語音辨識結束
    recognition.onend = () => {
        voiceBtn.innerHTML = '<i class="fas fa-microphone"></i> 語音輸入!';
        voiceBtn.disabled = false;
    };

    // 處理語音辨識錯誤
    recognition.onerror = (event) => {
        console.error('語音辨識錯誤:', event.error);

        alert('語音辨識失敗，請重試。');
        recognition.stop();

        voiceBtn.innerHTML = '<i class="fas fa-microphone"></i> 語音輸入!';
        voiceBtn.disabled = false;
    };
} else {
    alert('您的瀏覽器不支援語音辨識功能，請嘗試使用 Chrome 瀏覽器。');
}

```

### 語音輸入的流程

- 按下「語音輸入」按鈕後，會啟動語音辨識功能，辨識的語音會自動填入輸入框。
- 使用者可以手動編輯語音辨識的內容，然後點擊「發送」按鈕提交問題。
- 若瀏覽器不支援 Web Speech API，會提示使用者切換到支持的瀏覽器（如 Google Chrome）。

### 3.6.3 利用爬蟲技術幫助資料搜尋或語音輸入

```
import requests
from bs4 import BeautifulSoup

# 發送請求並獲取網頁內容

url = "https://example.com/faq"
response = requests.get(url)

# 解析網頁內容

soup = BeautifulSoup(response.text, 'html.parser')

# 擷取特定的資料

faq_items = soup.find_all('div', class_='faq-item')

# 存儲資料

faqs = []
for item in faq_items:
    question = item.find('h3').text.strip()
    answer = item.find('p').text.strip()
    faqs.append({"question": question, "answer": answer})

# 儲存擷取的資料至資料庫或向量資料庫
```

#### 語音輸入與爬蟲結合：

- 若用戶透過語音輸入問題，系統可以先識別問題，然後再利用爬蟲技術從網頁或其他來源搜尋相關答案。例如，語音輸入的是一個學校的問題，爬蟲可以搜尋學校的網站來找到該問題的答案，並回應用戶。

## 第四章 結論與未來方向

### 4.1 結論

#### 4.1.1 自然語言處理技術的準確性與優化

在實驗中，Langchain 作為核心自然語言處理引擎，展現了其強大的語言理解與生成能力。在校園導覽系統中，聊天機器人能夠自動理解用戶提出的問題並生成合理的回應。大多數問題，尤其是簡單的查詢，如校園地點介紹、課程查詢等，系統能夠快速且準確地給出答案。然而，系統在處理更複雜、多層次的查詢時，如涉及歷史背景或需要持續上下文交互的問題，表現出了一定的不足。例如，用戶可能會詢問某一地點的歷史背景或者活動時間，這時候系統可能無法給出準確的答案或者需要多輪的對話才能最終解決。

通過 RAG 技術，系統可以在回答用戶問題時檢索相關的外部資料，從而提高準確性。這一技術彌補了 Langchain 單靠模型生成回答時可能出現的知識局限性。實驗結果表明，RAG 技術在處理需要實時或動態更新的數據時尤其有效，例如查詢當前的校園

活動或課程變更。然而，RAG 技術也存在一些挑戰，如在網絡不穩定或外部數據源信息過於廣泛的情況下，可能會影響檢索速度和結果的精確度。

### 4.1.2 系統的回應速度與用戶體驗

在實驗過程中，系統的回應速度被認為是影響用戶體驗的重要因素之一。對於大多數簡單查詢，系統能夠在不到一秒的時間內給出回應。然而，在涉及到多輪對話、語義分析複雜度較高的情況下，系統的回應速度有所減緩，尤其是在使用 RAG 技術進行檢索的時候，檢索和生成的過程會增加一定的延遲。

在一些用戶測試中，當用戶提出較複雜的問題或期望系統給出精細的建議時，系統需要檢索更多的資料或跨數據庫查找答案，導致回應時間延長。這種延遲會影響用戶的整體體驗，特別是在一個互動頻繁的導覽系統中，用戶期望得到即時的反饋。因此，未來的優化方向之一就是如何在不影響準確性的前提下縮短系統的反應時間，或通過增加預緩存技術來提前儲備可能的回答。

### 4.1.3 聊天機器人的適應性與擴展性

實驗結果表明，基於 Langchain 與 RAG 技術的系統具備很好的適應性和擴展性。隨著系統所覆蓋的校園導覽數據集不斷擴展，系統能夠迅速適應新的查詢需求。例如，當校園內新增了建築物或活動場地時，只需要在後端更新相關數據，系統便能夠快速反映這些變化，無需對整個系統架構進行大幅修改。

此外，RAG 技術為系統的擴展性提供了額外的靈活性。該技術能夠動態檢索外部資料，使系統不僅限於校園內部數據，還可以回答涉及更廣泛背景的信息查詢，如查詢某個專業的就業前景、相關課程的外部資料等。這意味著系統不僅限於校園導覽的範疇，還可以逐步發展成一個多功能的資訊平台，為用戶提供更多樣化的服務。

#### 4.1.4 多輪對話與上下文跟蹤的挑戰

在多輪對話中，上下文跟蹤被認為是系統的一大挑戰。實驗結果顯示，儘管 Langchain 具備一定的上下文理解能力，但在一些較長的對話中，系統有時會遺忘前面的對話信息，導致回應不夠連貫。例如，在涉及到連續查詢某個地點的信息時，如果對話過程較長，系統有時會混淆用戶的具體需求或忘記前幾輪的對話上下文。

為了解決這一問題，未來的優化方向包括引入更先進的對話管理模塊，例如利用強化學習技術來動態學習用戶行為和對話模式，或者加強對話記憶功能，以便系統能夠在更長的對話中保持連貫性和準確性。

## 4.2 未來方向

### 4.2.1 提升自然語言處理與上下文跟蹤能力

首先，系統的自然語言處理能力仍有很大的提升空間。目前，系統能夠處理簡單的查詢和部分多輪對話，但在面對更複雜的用戶需求或長對話時，系統容易出現上下文遺忘或語義誤解的問題。未來可以進一步優化 Langchain 的語言模型，增強其在多輪

對話中的上下文記憶和處理能力。例如，通過引入更先進的對話管理技術，系統能夠更好地追蹤對話中的關鍵訊息，並在長時間交互中保持語意的一致性和連貫性。此外，可以考慮引入基於強化學習的對話管理系統，根據用戶的回應調整回答策略，從而提高回應的準確性和靈活性。

### 4.2.2 優化 RAG 技術的檢索效率

RAG 技術在系統中起到了關鍵作用，它能夠通過檢索外部資料來補充系統內部知識的不足。然而，系統目前在檢索大量數據時，檢索效率仍然有限，特別是當查詢涉及動態更新或跨數據庫檢索時，系統的回應速度會顯著降低。為了解決這一問題，未來可以考慮優化 RAG 技術的檢索算法，減少不必要的數據檢索時間。此外，通過預先緩存常用數據，系統能夠在處理高頻查詢時提供更快的響應。進一步的改進還包括設計更精細的資料庫結構，使數據的存儲和檢索更加高效，從而縮短回應時間，並提升用戶的整體體驗。

### 4.2.3 強化個性化推薦功能

個性化是現代聊天機器人的一個重要發展方向。未來可以在「中原大學嚮導」系統中引入更加靈活的個性化推薦功能。通過分析用戶的歷史查詢記錄、行為數據和偏好，系統能夠自動推薦與用戶相關的信息，如校園活動、學術講座或課程安排等。這一功能不僅能夠提升系統的實用性，還能提供更具針對性的服務，從而提高用戶滿意度。個性化功能的實現可以依賴於先進的機器學習算法，並通過不斷學習用戶行為來

精確匹配其需求。長期來看，這將使系統不僅僅是一個被動回答問題的工具，而是一個主動為用戶提供有價值資訊的智能助理。

#### 4.2.4 拓展系統的應用場景

最後，「中原大學嚮導」未來可以逐步拓展其應用場景，超越傳統的校園導覽功能。系統可以結合學術諮詢、新生輔導、課程建議等多種功能，成為學生和校園工作人員日常使用的多功能助手。例如，系統可以為學生提供課程規劃建議，根據其學習進度和個人目標推薦相關的選修課程和學術資源。對於校園活動，系統也可以提供實時更新和提醒，確保學生不會錯過任何重要活動。隨著功能的擴展，系統不僅可以服務於在校生，還能為即將入學的新生、畢業生甚至校友提供支持，從而全面覆蓋校園社區的需求。這一發展將極大提升系統的實用價值，使其成為中原大學校園生活中不可或缺的一部分。

## 參考文獻

- [1] bit jun, "實作練習 langchain3". HackMD.io, Mar.30.2024.
- [2] The Langchain community, "Langchain introduction", 2022.
- [3] ChiChieh Huang, "RAG (Retrieval Augmented Generation):為自然語言處理揭開新篇章", 2024.
- [4] 李勝凱(2018), 聊天機器人應用之探討—以南華大學資管系為例, 碩士論文, 南華大學。
- [5] 沈天佐(2002), 以網際網路內容為基礎之問答系統“為什麼”問句之研究, 碩士論文, 國立臺灣大學資訊工程系。
- [6] 劉秉鈞(2016), 人工智慧於模擬人類對話系統設計, 碩士論文, 逢甲大學自動控制工程學系。
- [7] 彭昱傑(2017), 聊天機器人系統設計與實作, 碩士論文, 國立中正大學資訊工程系。
- [8] 王薇婷(2024), 基於 LangChain 架構之學習歷程查詢系統之開發與評估, 碩士論文, 逢甲大學資訊工程系。

[9] Ballapuram, Ravi Kumar (2023), 利用基於本體增強的語義理解來改進 RAG 模型中的資料檢索, 碩士論文, 亞洲大學資訊工程系。

## 著作權授權書

本授權書所授權之專題報告為學生\_\_\_\_\_、  
\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_

在 中原大學 113 學年度第 一 學期所撰寫

專題名稱：\_\_\_\_\_

1. 本人具有著作財產權之專題全文，同意授予中原大學電機系，得重製成電子資料檔後收錄於該單位之網路，並與台灣學術網路及科技網路連線，得不限地域時間與次數以光碟或紙本重製發行。如有因涉及專利等智慧財產權之申請，請將本專題全文延至民國 年 月 日後再公開。
2. 本人具有著作財產權之專題內容文字與實驗數據，係本人所原創、實驗所得，如涉有智慧財產權等相關爭議與法律責任，概由本人承擔。

學生簽名：  
(所有同學親筆正楷)

指導老師：

中華民國 年 月 日

## 專題論文集書背格式說明

書背：裝訂時須做書背，字體大小以能辨識為主。包括上、中、下三部份

上寫：中原大學 電機工程系

中寫： (專題題目名稱)

下寫： 年 月 (口試時間之年與月)

(格式範例如下頁↓)

中原大學

電機工程系

(專題題目)

112  
年  
12  
月

65

## 注意事項:

1. 總頁數須**超過 30 頁**。
2. 格式可參考研究室或圖書館歷屆研究生論文。
3. 每份專題完整稿中，內容要有「**致謝**」、中英文摘要、目錄、內文、及「**授權書**」...等
4. 請務必依照格式撰寫，並存成 PDF 檔。**格式不符者一率退件重寫**，請於 **12/1(日)前**將論文報告電子檔依照下述命名方式，在 Google Drive 雲端硬碟上與帳號 [cveeproject@ee.cycu.edu.tw](mailto:cveeproject@ee.cycu.edu.tw) 共用，依照系統時間為準，逾期不接受。
  - a. 共上傳 5 個檔案如下
  - b. 檔案命名規則：
    - [第 87 組]專題論文報告 30 頁(Word 檔)v.1
    - [第 87 組]專題論文報告 30 頁(PDF 檔)v.1
    - [第 87 組]專題海報
    - [第 87 組]專題中英文摘要 v.1
    - [第 87 組]專題實作成品照片 v.1 (.jpg 檔, 需小於 2MB)
  - c. 因權限未開放預覽、刪除、下載，若有更新檔案請多上傳 1 份並更改檔名依照 v.2, v.3, v.4, v.5 順序上傳，依此類推。
  - d. 中英文摘要及照片需展示並公告至系網頁，同學們可參考系網頁「專題成果」：  
<https://reurl.cc/my2enA>
5. 封面請依照第一頁範例完成。
6. 中文字體一律標楷體，英文字體為 Times New Roman。  
標題(EX: 第一章 緒論) 字體大小為 20，粗體 章節  
(EX: 1.1 研究背景) 字體大小為 16，粗體內文字體  
大小為 12

實例:

## 第一章 緒論

### 1.1 研究背景

隨著人類科技與工業蓬勃發展，隱私一直是...。

### 1.2 研究動機

.圖與表格標示請依據圖下表上之原則。

實例:



表 1-1 實驗數據

圖 1-1 電機 logo

	條件一	條件二
1.		
2.		
3.		

7. 專題書面報告(30 頁)需利用「Turnitin 論文原創性比對系統」進行比對，請於 12/2(一)前將最後「文獻比對結果」最後一頁列印出紙本，經指導老師簽名後，繳交回系辦。(紙本請寫上班級、姓名)
8. 完成所有程序後，請交一份紙本報告含「書背」裝訂成冊給指導老師，如果要多印的同學，請自行影印。