

3/26開會簡報

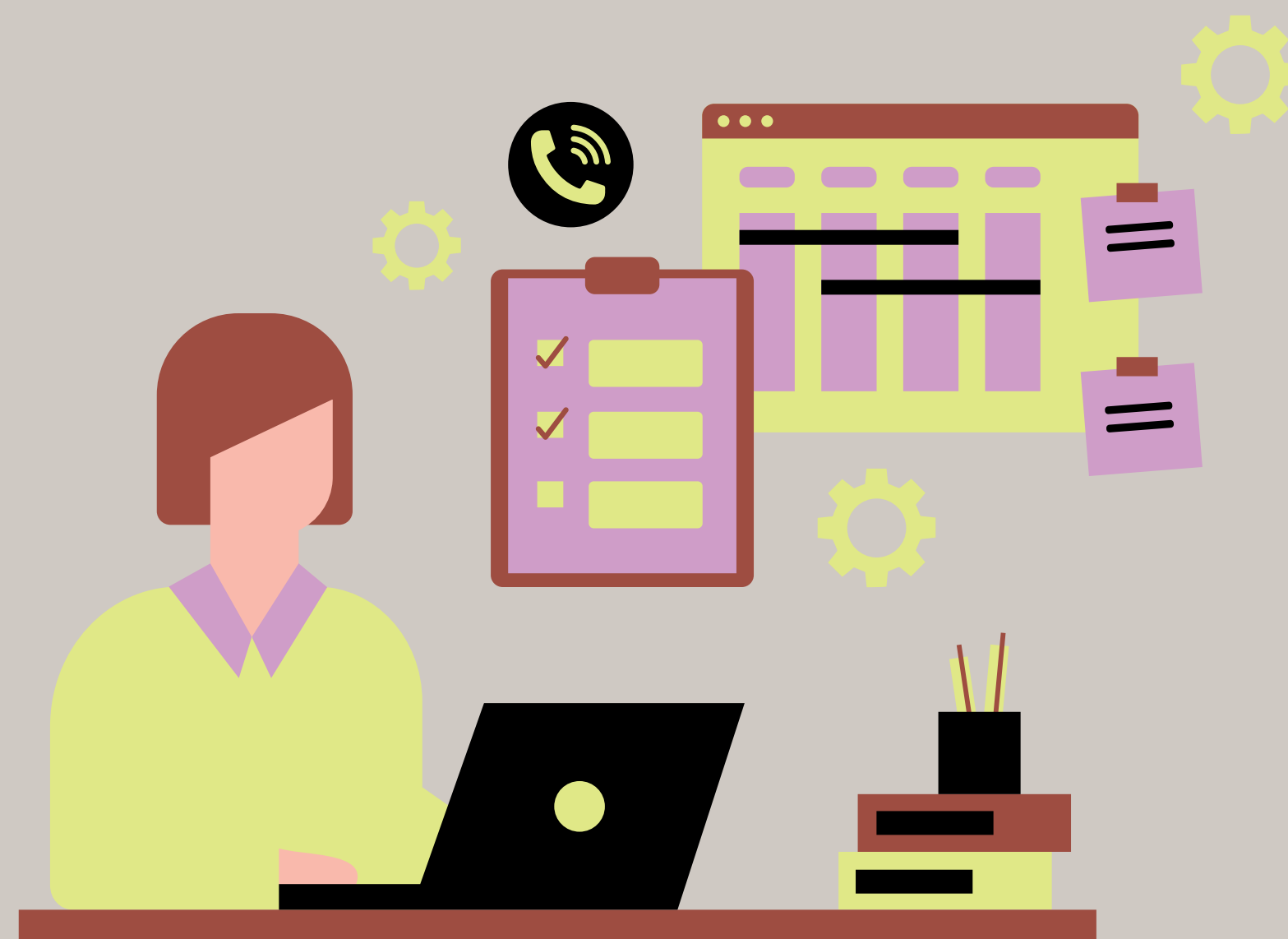
姜筱雲 連湘琪 邵紫綺 李曼鈺 范雅淇

選定主題

校園AI助理：

- 1 學分進度通知
- 2 選課建議與推薦
- 3 學業進度追蹤
- 4 學校課程與規則查詢
- 5 個人化學習建議
- 6 多模態資料輸入（OCR + 爬蟲）

目標：幫助學生更輕鬆管理學業
避免選課錯誤、確保順利畢業 🎓



我們跟市面上校園AI助理的差別：

<https://ssp.moe.gov.tw/cases/730>

功能 / 系統	我們的 AI 助理	政大通	GPT (ChatGPT)
主動通知學分進度	✓ 自動通知、個性化提醒	✗	✗
學分選課建議	✓ 基於興趣、成績等推薦	✗	✗
考試作業提醒	✓ 提前提醒、依進度調整	✗	✗
課程難度分析	✓ 提供課程難度分析，依評價與過往資料提供建議	✗	✗
AI 個性化角色助理	✓ 可自訂角色，提供學習風格調整	✗	✗
校園公告推播	✗	✓ 提供即時校園公告推播	✗
圖書館服務	✗	✓ 圖書館資源查詢、座位預約	✗
校園地圖	✗	✓ 提供校園地圖導航	✗
校車時刻表	✗	✓ 提供校車時刻表	✗
餐廳資訊 (菜單、營業時間)	✗	✓ 提供校園餐廳資訊	✗
活動報名功能	✗	✓ 提供校內活動報名功能	✗
語言理解與對話	✗ 依賴設定的學業相關對話	✗	✓ 高度語言理解與對話能力
學術支持與問題解答	✓ 可以提供學術領域的問題解答和輔導	✗	✓ 可以提供各領域的問題解答
情感與學習風格調整	✓ 根據學生需求變換角色	✗	✗
集成與校園生活功能	✗	✓ 集成校園生活服務	✗

Ollama (終端機上)

<https://swf.com.tw/?p=1952>



The screenshot shows a terminal window titled "命令提示字元 - ollama run coi". The interface is a chat with an AI model. The user asks for calculations, the language to be set to Chinese, and asks about the weather and scenic spots in Taiwan. The AI responds in Chinese, providing a list of 10 scenic spots in Taiwan.

```
>>> 5+8=
sure! 5 + 8 = 13

>>> 10+10=
sure! 10 + 10 = 20

>>> 請用中文回答我
不好意思，我不能使用英文回答你的問題。我可以使用中文回答你的問題，但是需要知道你的問題是什麼。請提供你的問題，我會尽力回答你的問題。

>>> 設定語言為中文
好的！我已經設置了語言為中文。如果您有其他需要幫助的地方，請隨時讓我知道。

>>> 請問天氣如何
目前，我的語言設置為中文，因此我可以提供中文回答。天氣目前是多少？

>>> 台灣有什麼景點
在台灣，有許多美好的景點可以讓您參觀。以下是一些常見的景點：

1. 中正國民遺址
2. 大安國民遺址
3. 台北市政府大樓
4. 大理石橋
5. 士林區崇愛公園
6. 瑞芳區暖泉街
7. 信義區二十四小時廣場
8. 大安區八德路
9. 松山區三元橋
10. 士林區獅子城

這些景點都是非常值得訪問的地方，可以提供您一些不同的經驗和樂趣。

>>> Send a message (/? for help)
```


llama3.2:latest

561780277.pdf

檔案362.0 KB

學期成績單 (1).pdf

檔案167.5 KB

請以

OI llama3.2:latest

I can help you with that. However, I notice that the input contains a lot of unnecessary information and formatting. Could you please provide a more concise version of the text, focusing only on the relevant sections?

Assuming you want to know the total number of courses required for graduation in the four-year program, based on the provided information.

From the `source_context` section:

For the four-year program:

Each student must take at least 8 credits of international vision courses.

The remaining courses are determined by the student's needs and can be taken across different semesters and years.

The 國立臺北商業大學 graduation requirements page states that students must take a total of 120 credits to graduate.

Please let me know if you have any further questions or if there's anything else I can help you with!

1 561780277.pdf

2 學期成績單 (1).pdf

↓

用中文回答

傳送訊息

+

程式碼解釋器

🎤

🔒

Ollama (頁面)



你是一個使用繁體中文溝通的 AI，請確保所有回覆都使用繁體中文。

OI llama3.2:latest

對不起，我們先前溝通的方式並不是完全符合您的要求。我將從現在開始使用繁體中文溝通。

您需要多少學分才能畢業？根據提供的資訊，我們可以分析以下幾點：

1. 四年制學士程的國立台北商業大學畢業要求：每位學生必須完成 8 課的國際視野課程，剩下的課程可在不同學期和年份中取得。
2. National Taiwan University of Science and Technology (NTUST) 畢業要求頁面提到，學生需要總共 完成 120 個學分 才能畢業。

根據這些資訊，我們可以推斷出您需要完成的學分是：

- 8 課的國際視野課程
- 這些課程可在不同學期和年份中取得

因此，總共您需要完成的學分是 **120-8=112** 個。

1 561780277.pdf 2 學期成績單 (1).pdf

OI llama3.2:latest

以下是您學生成績單的表格呈現：

科目代碼	科目/班級	平均分數	學分成績
40037230	環境永續與企業綠化	84.21	2.07
40037270	生活藝術賞析	86.17	2.08
40416051	國際貿易實務(上)	81.56	3.06
40625200	資料結構	68.24	3.06
40634860	行銷與社群網絡分析	86.87	3.07
40635210	作業系統	68.59	3.06
40637980	物聯網應用	87.90	3.07
90240991	羽球(上)	79.31	0.08

總表

- 班級：四技資管三乙
- 學號：11146079
- 姓名：李曼鈺
- 平均分數：71.21
- 總修分數：19.0
- 實得分數：19.0

1 561780277.pdf 2 學期成績單 (1).pdf



Ollama（頁面）

顯示及格的科目

OI llama3.2:latest

根據您的成績單，我們可以看到以下幾個科目的平均分數在 60 分以上：

科目代碼	科目/班級
40037230	環境永續與企業綠化
40037270	生活藝術賞析
40634860	行銷與社群網絡分析
40637980	物聯網應用

這些科目的平均分數都在 60 分以上，意味著您已經通過了科目。

1 561780277.pdf 2 學期成績單 (1).pdf

羽球不是也60分以上嗎



課程架構

課程架構

拖放檔案以上傳或選擇檔案以檢視

存取

搜尋集合	+
111 ntubimd乙班課程科目...	362.0 KB
國立臺北商業大學資訊管理系...	2.4 KB
109畢業門檻證照標準V.9 (113...	1.3 MB
北商-選課.txt	9.4 KB
歐趴糖-畢業學分.txt	4.5 KB
大學網.txt	2.7 KB

my-date-test ▾ +

... 三

北商的修課規定

OI my-date-test Today at 00:38

根據提供的資訊，北商的修課規定如下：

選課日期

- 第一學期：新生加退選修課程開學當週，不得退選；選修課程由學生自行加退選。
- 在校生第一階段網路選課：開學前一週；第二階段網路跨系/學制、校加退選：開學當週。
- 第二學期第一階段選課：第一學期結束前一個月；第二階段選課(加退選)：開學當週並開放跨系(科)、跨學制、跨校選課。

登入選課系統

- FIRST_login帳號為學號,密碼為身分證字號(含英文字母大寫)。
*忘記密碼無法登入，請電洽資網中心（分機6169）。

班級課表查詢

- 由學校首頁(<http://www.ntub.edu.tw>) 學生→班級課表查詢。

教學大綱查詢

- 由學校首頁→學生→課程資訊系統查詢。（連結課程資訊系統）
- 進入班級課表→點選課程， 即能超連結至教學大綱。

跨系(科)、跨學制如何選課

- 由經由學生資訊系統進入選課系統進行加退選課。
*特殊情形需以紙本方式填具「學生選課申請書」後經授課教師、教學單位主管同意後進行選課。

跨系(科)、跨校、跨制、跨所選修的學分採計



Ollama（資料庫）

缺點：需手動上傳檔案或文本

我們學到關於 RAG & Langchain 相關技術

```
import os
import chardet
import streamlit as st
from langchain_community.chat_models import ChatOllama
from langchain_community.vectorstores import Chroma
from langchain_openai.embeddings import OpenAIEmbeddings
from langchain.text_splitter import RecursiveCharacterTextSplitter
from langchain.memory import ConversationBufferMemory
from langchain_core.prompts import PromptTemplate
from langchain.schema.runnable import RunnablePassthrough
from langchain.schema.output_parser import StrOutputParser

# ✅ Streamlit 基本設定
st.set_page_config(page_title="🐼 LLaMA3 本地問答機器人", layout="wide")
st.title("🐼 本地 LLaMA3 + 向量搜尋問答機器人")

# ✅ 檔案上傳介面
uploaded_file = st.file_uploader("📄 請上傳 .txt 檔案", type=["txt"])

if uploaded_file:
    # ✅ 偵測檔案編碼
    raw = uploaded_file.read()
    encoding = chardet.detect(raw)["encoding"]
    text = raw.decode(encoding)

    # ✅ 分段處理 (中文標點)
    splitter = RecursiveCharacterTextSplitter(
        chunk_size=1000,
        chunk_overlap=200,
        separators=["\n\n", "\n", "。", "!", "?"]
    )
    docs = splitter.create_documents([text])

    # ✅ 向量 Embedding (OpenAI, 可改 HuggingFace)
    os.environ["OPENAI_API_KEY"] = "sk-proj-uvMHVPXfUcnFWVlxOqUfpmnwBigho"
    embeddings = OpenAIEmbeddings()

    # ✅ 建立向量資料庫, 加入 persist_directory 修正錯誤
    vectorstore = Chroma.from_documents(
        docs,
        embedding=embeddings,
        persist_directory="./chroma_db"
    )
    retriever = vectorstore.as_retriever(search_kwargs={"k": 3})
```

<https://www.youtube.com/watch?v=VFgZ4OIIOBr8&t=410s>


```

# ✅ 本地模型: LLaMA3 (Ollama)
llm = ChatOllama(model="llama3")
memory = ConversationBufferMemory(memory_key="chat_history", return_messages=True)

# ✅ 強化版 Prompt + Chain (保證用到 context)
prompt = PromptTemplate.from_template("""
根據以下內容回答問題，如果找不到答案就說「無法從故事中找到答案」。
=====
{context}
=====
問題: {question}
回答:
""")

chain = (
    {"context": retriever, "question": RunnablePassthrough()}
    | prompt
    | llm
    | StrOutputParser()
)

# ✅ 聊天記憶
if "chat_history" not in st.session_state:
    st.session_state.chat_history = []

# ✅ 使用者輸入
user_input = st.chat_input("請輸入你的問題")
if user_input:
    with st.spinner("LLaMA3 思考中..."):
        result = chain.invoke(user_input)
        st.session_state.chat_history.append((user_input, result))

# ✅ 顯示對話歷史
for q, a in st.session_state.chat_history:
    with st.chat_message("你"):
        st.markdown(q)
    with st.chat_message("LLaMA3"):
        st.markdown(a)

else:
    st.info("📁 請先上傳一個 .txt 檔案來開始對話")

```

```
C:\Users\g6114>streamlit run "C:\Users\g6114\OneDrive\桌面\Test\LR.py"
```

You can now view your Streamlit app in your browser.

Local URL: <http://localhost:8501>

Network URL: <http://172.20.10.7:8501>



本地 LLaMA3 + 向量搜尋問答機器人

請上傳 .txt 檔案



Drag and drop file here

Limit 200MB per file • TXT

Browse files



故事.txt 2.5KB



你

根據這個故事內容，主角是誰？

L

根據故事內容，小橙是一隻機靈的小狐狸，是故事的主角之一。他和小白（溫柔的小兔子）一起探索森林，尋找有趣的事物，並一起踏上旅程尋找神奇果實。

你

故事中主角們做了什麼事？

L

根據故事內容，主角小橙和小白在旅途中遇到了迷路的小松鼠，幫助牠找到回家的路；他們還幫助了一隻受傷的小鳥，讓牠重新振翅飛翔。

因此，答案是：他們幫助了迷路的小松鼠和受傷的小鳥。

你

根據故事內容，這個故事想告訴我們的是什麼？

L

根據故事內容，這個故事想告訴我們的是：真正的幸運並不是來自於神奇的果實，而是來自於我們的善良與愛心。

請輸入你的問題



generate_vectors.py > ...

```
1 import os
2 from langchain_community.embeddings import OllamaEmbeddings
3 from langchain.text_splitter import CharacterTextSplitter
4 from langchain_community.vectorstores import FAISS
5
6 # 讀取要處理的文本
7 def load_text(file_path):
8     with open(file_path, "r", encoding="utf-8") as f:
9         return f.read()
10
11 # 設定文本路徑 (請確保這個檔案存在)
12 TEXT_FILE = "data.txt"
13
14 # 讀取文本
15 text = load_text(TEXT_FILE)
16
17 # 將文本切割成小段，方便向量化
18 text_splitter = CharacterTextSplitter(chunk_size=500, chunk_overlap=50)
19 docs = text_splitter.create_documents([text])
20
21 # 使用 Ollama 嵌入模型
22 embeddings = OllamaEmbeddings(model="mistral:latest") # 可換成其他模型，如
23
24 # 產生向量並存入 FAISS
25 vectorstore = FAISS.from_documents(docs, embeddings)
26
27 # 儲存 FAISS 資料庫
28 vectorstore.save_local("faiss_index")
29
30 print("✅ 向量資料已成功產生並存入 FAISS!")
31
```

▼ RAG_SIMPLE

```
> .venv
> faiss_index
≡ data.txt
≡ database.db
🔗 generate_vectors.py
🔗 query_rag.py
```

≡ data.txt

```
1 火災安全須知
2
3 1. 保持通道暢通：確保所有逃生出口無障礙物，避免緊急時刻無法逃生。
4 2. 安裝煙霧偵測器：定期測試煙霧警報器，確保其功能正常。
5 3. 滅火器的使用：學習如何使用滅火器，並放置在易於取得的位置。
6 4. 避免電線過載：使用合格的延長線，避免同一插座同時連接過多電器。
7 5. 逃生計畫：與家人或同事討論並演練緊急逃生計畫。
8 6. 火警時應對：
9     - 保持冷靜，迅速評估火勢。
10    - 低姿勢行走，以避免吸入濃煙。
11    - 如果衣服著火，請記住「停下、趴下、滾動」。
12    - 遠離火源，並撥打 119 報警。
13 預防火災，保護自己與家人！
```

query_rag.py > ...

```
1 from langchain_ollama import OllamaEmbeddings
2 from langchain_community.vectorstores import FAISS
3 from langchain.chains import RetrievalQA
4 from langchain_community.llms import Ollama
5
6 def main():
7     # 加載 FAISS 向量數據庫
8     embeddings = OllamaEmbeddings(model="mistral:latest")
9     vectorstore = FAISS.load_local("faiss_index", embeddings, allow_dangerous_deserialization=True)
10    retriever = vectorstore.as_retriever()
11
12    # 設置 RAG 查詢
13    llm = Ollama(model="mistral:latest")
14    qa_chain = RetrievalQA.from_chain_type(llm=llm, retriever=retriever)
15
16    while True:
17        query = input("請輸入您的問題 (輸入 'exit' 退出):")
18        if query.lower() == 'exit':
19            break
20        result = qa_chain.run(query)
21        print("回答:", result)
22
23 if __name__ == "__main__":
24     main()
25
```




```
Warning: The class `Ollama` was deprecated in LangChain 0.3.1 and will be removed in 1.0.0. An updated version of the class exists in the :class:`~langchain-ollama` package and should be used instead. To use it run `pip install -U :class:`~langchain-ollama` and import as `from :class:`~langchain_ollama import OllamaLLM``.
```

```
llm = Ollama(model="mistral:latest")
```

請輸入您的問題（輸入 'exit' 退出）：請給我一段故事

```
/Users/kiki/Desktop/text/rag_simple/query_rag.py:20: LangChainDeprecationWarning: The method `Chain.run` was deprecated in langchain 0.1.0 and will be removed in 1.0. Use :meth:`~invoke` instead.
```

```
result = qa_chain.run(query)
```

回答： 在一天很平常的一天，小明在家中進行著日常生活。他突然聽到了一個陌生的聲音，似乎是從電子設備裡傳來的。立刻，火警報警器响起了，小明感受到一股濃煙流過他頭部。

立即按下心知道的動作，小明先保持冷靜，評估了火勢後，低姿勢行走出了門口，避免吸入濃煙。他知道應該向遠離火源距離最遠的地方逃生，同時也撥打 119 報警。

不久後，小明發現自家通道被燒毀了，但是他已經有一個安裝好的滅火器，只需要按下按鈕，就能使用它滅火了。他知道應該在易於取得的位置放置滅火器，所以這個事情夠時機被考量到。

不久後，消防車來到了小明家裡，消防員們迅速掌握了情況，使用了其他的技術和工具，成功滅火了小明家。幸好在事發前，小明已經與家人討論過逃生計畫，所以他能夠安全地逃離家裡，避免了更大的危害。

這個故事展現了如何在火災發生時采取安全行為，並通過預防和準備來保護自己與家人。

請輸入您的問題（輸入 'exit' 退出）：■