這個專案是一個基於 Cohere 的 RAG (Retrieval Augmented Generation) 問答機器人，它整合了 Gradio 用於網頁使用者介面、FastAPI 用於 API 服務，並支援 LINE Bot 串接。

**功能說明**

* **文件上傳與向量化**:
  + 使用者可以透過 Gradio 介面一次上傳多個文件，支援 .doc, .docx, .xls, .xlsx, .pdf, .txt, .csv 等多種文件格式.
  + 上傳的文件會被處理、切塊，並使用 Cohere 的 embed-multilingual-v3.0 模型轉換成向量.
  + 向量會儲存在本地的 FAISS 向量資料庫中，用於快速檢索.
  + 專案會自動偵測文件變動（新增或更新），並支援增量更新向量資料庫，無需每次都重新建立.
* **智慧問答 (RAG)**:
  + 使用 Cohere 的 command-r-plus 模型作為大型語言模型 (LLM).
  + 當使用者提問時，系統會從向量資料庫中檢索相關的文件片段.
  + 將檢索到的資訊與問題一起傳送給 LLM，以生成更精確和有根據的回答.
* **多介面支援**:
  + **Gradio 網頁介面**: 提供直觀的網頁使用者介面，方便使用者上傳文件、提問並查看回答.
  + **FastAPI Webhook**: 提供一個 /webhook API 端點，允許其他應用程式透過 HTTP POST 請求傳送訊息並接收 RAG 回答.
  + **LINE Bot 串接**: 支援 LINE 訊息機器人功能，可以透過 LINE 平台與 RAG 機器人互動.

**所需安裝步驟**

1. **環境準備**:
   * 確保您的系統已安裝 Python 3.8 或更高版本。
   * 建議使用 venv 或 conda 建立虛擬環境，以避免套件衝突。
2. **安裝專案依賴**:
   * 在專案根目錄下，您需要一個 requirements.txt 檔案來列出所有必要的 Python 套件。根據 main.py 的內容，這些套件應包含：
     + gradio
     + fastapi
     + uvicorn (用於啟動 FastAPI 應用)
     + langchain
     + langchain-cohere
     + langchain-community
     + faiss-cpu (或 faiss-gpu 如果您有 CUDA 支援)
     + python-dotenv (用於管理環境變數，雖然程式碼中直接使用 os.getenv)
     + unstructured (用於處理 PDF, Word, Excel 等文件，其下可能包含多個子套件，例如 unstructured-inference, unstructured-pytesseract, detectron2 等，可能需要額外安裝系統依賴)
     + openpyxl (用於 Excel 文件)
     + python-docx (用於 Word 文件)
     + pypdf (用於 PDF 文件，或 unstructured 會自行處理)
     + httpx (用於 LINE Bot 的 HTTP 請求)
   * 運行以下命令安裝：

Bash

pip install -r requirements.txt

如果沒有 requirements.txt，您可以手動安裝上述列出的套件：

Bash

pip install gradio fastapi "langchain>=0.2.0" "langchain-cohere>=0.0.3" langchain-community faiss-cpu python-dotenv unstructured openpyxl python-docx pypdf httpx uvicorn

*注意*：unstructured 及其相關依賴在安裝時可能會比較複雜，可能需要額外安裝系統套件，例如 poppler (用於 PDF) 和 tesseract (用於 OCR)。

1. **申請 LINE Channel (LINE Bot)**:
   * 前往 [LINE Developers 網站](https://developers.line.biz/zh-hant/) 並登入您的 LINE 帳號。
   * 點擊「Console」進入控制台。
   * 點擊「Create a new provider」建立一個新的服務提供者 (如果您還沒有的話)，並輸入提供者名稱。
   * 在提供者下方，點擊「Create a new channel」。
   * 選擇「Messaging API」作為 Channel type。
   * 填寫 Channel 資訊，包括：
     + **Channel icon**: 選擇您的 Bot 頭像。
     + **Channel name**: 您的 Bot 名稱。
     + **Channel description**: 您的 Bot 描述。
     + **Category** 和 **Subcategory**: 選擇適合您的 Bot 類別。
     + **Email address**: 輸入您的 Email。
   * 同意相關條款後，點擊「Create」。
   * 進入您剛建立的 Channel 設定頁面。在「Basic settings」頁籤下，記下您的 **Channel ID** 和 **Channel secret**。
   * 切換到「Messaging API」頁籤。在這裡您會找到 **Channel access token (long-lived)**。點擊「issue」來生成一個新的 Access Token，並將其記下。
2. **設定環境變數**:
   * 您需要設定 Cohere API Key、LINE Channel Secret 和 LINE Channel Access Token。
   * 在專案根目錄下建立一個 .env 檔案，並填入您的 API Keys:
   * COHERE\_API\_KEY="您的 Cohere API Key"
   * LINE\_CHANNEL\_SECRET="您在 LINE Developers 取得的 Channel secret"
   * LINE\_CHANNEL\_ACCESS\_TOKEN="您在 LINE Developers 取得的 Channel access token"
   * *重要*：請勿將 .env 檔案提交到版本控制系統 (如 Git)。
3. **準備文件**:
   * 在專案根目錄下建立一個名為 docs 的資料夾.
   * 將您希望用於 RAG 的文件 (例如 .txt, .pdf, .docx, .xlsx, .csv) 放入 docs 資料夾中.
4. **啟動應用程式**:
   * 在終端機中，導航到專案根目錄.
   * 運行以下命令啟動 FastAPI 應用（Gradio 介面會被掛載在其上）:

Bash

uvicorn main:app --host 0.0.0.0 --port 8000 --reload

* + - main:app: 指定 main.py 檔案中的 app 實例.
    - --host 0.0.0.0: 允許從任何 IP 位址訪問.
    - --port 8000: 在 8000 端口啟動服務.
    - --reload: 在程式碼變更時自動重啟服務（開發時使用）.

1. **配置 LINE Bot Webhook**:
   * 在您的 LINE Developers Channel 設定頁面 (Messaging API 頁籤) 中，找到「Webhook settings」。
   * 將 **Webhook URL** 設定為您的伺服器可公開訪問的 URL，例如：https://您的伺服器IP或域名:8000/line/webhook。
     + *注意*：如果您的伺服器位於防火牆後方，或者您沒有固定的公共 IP，您可能需要使用 **ngrok** 或其他穿透工具來將本地端口暴露到公共網路。例如，如果您使用 ngrok 運行在 8000 端口，ngrok 會給您一個像 https://xxxxxx.ngrok-free.app 的 URL，您就將此 URL 填入 Webhook URL，並在後面加上 /line/webhook。
   * 點擊「Verify」按鈕來驗證您的 Webhook URL 是否正確。
   * 確保將 **Use webhook** 設定為「Enabled」。
   * 在「Messaging API」頁籤下，找到「LINE Official Account features」。
   * 建議關閉「Auto-reply messages」和「Greeting messages」，以避免 LINE 平台的預設回復與您的 Bot 衝突.
2. **訪問介面**:
   * 應用程式啟動後，您可以在瀏覽器中訪問 http://localhost:8000/gradio 來使用 Gradio 網頁介面.
   * API 服務會運行在 http://localhost:8000/webhook.
   * LINE Bot Webhook 會運行在 http://localhost:8000/line/webhook.
   * 您可以掃描您 LINE Bot 的 QR Code (在 Channel 設定頁面的「Basic settings」頁籤下找到) 來將 Bot 加入您的 LINE 帳號，並開始測試訊息。

**LINE官方帳號訊息計費制度**

LINE官方帳號目前主要分為三種方案：

* **輕用量方案 (免費)**
  + **月費：** 免費
  + **免費訊息則數：** 每月200則
  + **加購訊息：** 無法加購，若超過200則則需升級方案。
* **中用量方案 (付費)**
  + **月費：** 新台幣800元 (未稅)
  + **免費訊息則數：** 每月3,000則
  + **加購訊息：** 無法加購，若超過3,000則則需升級方案。
* **高用量方案 (付費)**
  + **月費：** 新台幣1,200元 (未稅)
  + **免費訊息則數：** 每月6,000則
  + **加購訊息：** 可加購，每則費用從0.2元起跳，採階梯式計價（發送越多單價越低）。

**訊息則數的計算方式：**

訊息則數是以「**主動發送的訊息**」給一位好友，就算一則訊息。例如，您群發一則訊息給100位好友，即使內容只有一行文字，也會計算為100則訊息。

您可以登入LINE官方帳號後台，在「設定」>「帳務專區」中查看您目前的方案和已發送的訊息則數。

**BOT自動回復訊息是否計入訊息額度？**

BOT自動回復功能，答案是：

**自動回復訊息原則上不列入訊息額度計算。**

LINE官方將訊息分為「**主動發送**」和「**被動發送**」：

* **主動發送：** 指由LINE官方帳號主動推播給用戶的訊息，例如**群發訊息**、**排程訊息**、透過**Push API**發送的訊息等。這些訊息會計入您的訊息額度。
* **被動發送：** 指由用戶行為「觸發」的訊息，例如：
  + **加入好友時的歡迎訊息：** 用戶加入您的官方帳號後，系統自動發送的歡迎訊息。
  + **自動回應訊息：** 您設定的**關鍵字回應**，當用戶輸入特定關鍵字時，BOT自動回覆的訊息。
  + **一對一聊天訊息：** 管理員在後台手動與用戶進行一對一對話的訊息。
  + **透過Reply API回覆的訊息：** 您的BOT根據用戶傳送的訊息，透過Reply API自動回覆的內容。

**因此，BOT自動回復（如關鍵字回覆、歡迎訊息等）是屬於被動發送，通常**不會列入**您的訊息額度計算。**

這是一個非常有利的機制，您可以充分利用BOT的自動回復功能來解答用戶常見問題、提供資訊或引導流程，而無需擔心額外的訊息費用，有效地節省行銷成本。

總結來說，若您主要使用BOT進行自動回復，且不頻繁進行大規模群發，那麼您選擇**輕用量方案**或**中用量方案**可能就足夠了。但如果您的業務需要經常主動推播訊息給大量好友，則應考慮**高用量方案**，並妥善規劃您的訊息發送策略。