檢索增強生成（RAG）技術說明文件

一、RAG 技術簡介

RAG（Retrieval-Augmented Generation，檢索增強生成）是一種結合「資訊檢索」與「大型語言模型生成能力」的技術架構。其主要目的是讓語言模型在回答問題時，能夠根據指定文件內容進行推論，而非僅依賴模型訓練期間的內建知識。

RAG 特別適合用於企業內部文件問答、法務與金融分析，以及需要降低 AI 幻覺風險的應用場景。

二、為何需要 RAG？

大型語言模型雖然具備良好的語言理解能力，但仍存在以下問題：

1. 無法即時更新最新資料
2. 可能產生看似合理但實際錯誤的回答（幻覺）
3. 無法確認回答資訊的來源

RAG 透過引入外部文件檢索機制，使模型回答能「有所依據」，提升可信度與可控性。

三、RAG 系統的核心流程

RAG 系統通常包含以下幾個主要步驟：

1. 文件切塊（Chunking）

系統會先將長篇文件切割為較小的文字區塊。
常見設定為每個 Chunk 約 300 至 500 字，並保留部分重疊區段，以避免語意在邊界處中斷。

2. 向量化（Embedding）

每個文件區塊會透過 Embedding 模型轉換為數值向量，用來表示該文字的語意特徵。
此向量將用於後續的相似度搜尋。

3. 建立向量資料庫

完成向量化後，所有文件向量會被存入向量資料庫中，以支援快速且高效的語意搜尋。

四、使用者提問與檢索機制

當使用者輸入問題時，系統會將該問題轉換為向量，並與向量資料庫中的文件向量進行比對。
透過相似度計算，系統能找出與問題語意最接近的文件區塊，作為生成回答的參考依據。

五、Prompt 組合與回答生成

在取得相關文件內容後，系統會將「檢索到的文件片段」與「使用者問題」整合成 Prompt，交由大型語言模型生成最終回答。

此方式可確保模型的回答基於實際文件內容，降低錯誤與幻覺產生的風險。

六、影響 RAG 成效的因素

RAG 系統的效能會受到多項設計因素影響，包括：

- Chunk 大小與是否重疊
- 檢索數量（Top-K）設定
- 是否使用 Re-ranking 機制
- Metadata（如頁碼、文件來源、日期）的設計

適當的參數調整可有效提升系統的回答品質。

七、RAG 的應用場景

RAG 技術可應用於以下領域：

- 企業內部知識庫問答
- 客服與聊天機器人長期記憶
- 金融、法務與醫療文件輔助分析

即使未來大型語言模型支援更長的上下文長度，RAG 仍因其成本低、資料可更新與來源可控等優勢，具有高度實務價值。
