Day5 – RAG Chunking & Retrieval Analysis 作業報告

1. 封面 (Cover Page)

課程名稱：AI 技術與應用開發課程　作業名稱：Day5 – RAG Chunking & Retrieval Analysis
學號：**1411232035**　姓名：阮荏浩

2. 作業目標 (Assignment Goals)

本作業旨在透過實驗與定量分析，探討檢索增強生成 (RAG) 系統中不同文本切塊 (Chunking) 策略對檢索精準度的影響。核心目標如下：

- 實作多元切塊策略：針對原始文本執行固定大小 (Fixed-size)、滑動視窗 (Sliding window) 及語意切塊 (Semantic chunking) 三種具備不同邏輯的分割處理。
- 向量資料庫整合：練習將分割後的文本透過高維度 Embedding API 轉換為向量格式，並於 Qdrant 向量資料庫中建立高效能索引。
- 檢索品質定量評比：比較不同策略在 RAG 任務中的語意相似度表現，並深入探討參數微調如何優化檢索系統的召回品質。

3. 系統流程圖說明 (System Workflow)

為了嚴謹評估檢索品質，本實驗建構了一套標準化的 RAG 處理流程。透過將檢索結果直接送入大型語言模型 (LLM) 進行評估，以確保得分能反映真實的語意關聯性，而非僅依賴數學上的向量距離。

1. 原始文件處理：讀取與清理來源 .txt 文件。
2. 執行 **Chunking**：套用三種預設策略（固定、滑動、語意）進行文件切分。
3. 向量化轉換：呼叫 Embedding API 將文本轉換為 4096 維度之向量表徵。
4. 建立向量索引：將向量數據存入 Qdrant Vector DB 進行管理。
5. 執行 **Top-1 Retrieval**：針對測試問題集檢索相似度最高的單一區塊 (limit=1)。
6. **LLM** 語意評分：將檢索所得文本 (retrieve_text) 與問題送入 LLM scoring API。
7. 輸出結果：產出最終語意相似度分數 (Score)。本作業的評分結果是基於『檢索結果送入 **LLM** 進行語意評分』，而非單純向量距離。

4. 三種 Chunking 方法說明 (Chunking Methodologies)

不同的切塊邏輯決定了模型獲取上下文的完整度，以下為本實驗採用的三種配置：

固定大小 (Fixed-size)

- 參數配置：　**chunk_size**：**512**，**overlap**：**64**。
- 特性說明：此方法具備高度的運算穩定性與處理效率。然而，由於缺乏重疊區塊，其資訊覆蓋率較低，且容易在切點處發生語意中斷。對於需要長篇敘述或跨區塊資訊的問題，常面臨上下文銜接不足的問題。

滑動視窗 (Sliding window)

- 參數配置：　**chunk_size**：**384**，**overlap**：**128**。
- 特性說明：藉由設置 **128 tokens** 的重疊區域作為緩衝區，此方法能有效避免關鍵資訊在切點遺失，並在區塊間保留重複資訊以維持脈絡。實驗顯示此法能穩定提升檢索平均分數，是極具韌性的基線配置。

語意切塊 (Semantic chunking)

- 參數配置： **max_tokens**：**512**， **min_tokens**：**84**， **sim_threshold**：**0.12** 。
- 特性說明： 區塊邊界不再固定，而是依據內容語意相似度動態切分。實驗中發現，**min_tokens** 的調整對細節命中率有決定性影響：若此值過大會導致切塊過於粗糙；適度降低則能捕捉更細粒度的資訊。

5. 參數調整實驗 (Parameter Tuning Experiments)

在語意切塊的實作中，參數微調是決定成敗的關鍵。下表展示了針對 min_tokens 與 sim_threshold 進行的實驗路徑：| min_tokens | sim_threshold | avg score || ------ | ------ | ------ || 256 | 0.18 | ↓ (下降) || 128 | 0.18 | ↑ (上升) || 96 | 0.14 | ↑↑ (顯著上升) || **84** | **0.12** | **0.7528 (最佳)** || 64 | 0.12 | ≈ (持平) |
實驗總結： 數據分析顯示，min_tokens 與檢索粒度呈現反比關係。當 min_tokens 過小時，產出的 chunk 過於碎裂，反而損害語意連貫性。實驗證明 **84** 是本資料集的**「止損點」**，在此參數下能兼顧細節捕捉與上下文的完整性。

6. 實驗結果總結 (Results Summary)

綜合 20 題測試樣本的檢索表現，三種策略的最終效能如下：| Chunking Method | Avg Score (20 題) || ------ | ------ || 固定大小 | ~0.69 || 滑動視窗 | ~0.73 || 語意切塊 | **~0.75 (最高)** |
重點強調： 語意切塊在整體平均表現上最為優異，雖然其配置複雜度較高，但在處理需要跨段落理解與因果分析的複雜問題時，其優勢遠超傳統方法。

7. 討論與觀察 (Discussion & Observations)

1. 語意切塊的深層優勢： 由於區塊邊界是根據邏輯意義而非字數劃分，語意切塊在處理涉及「因果關係」與「背景說明」的問題時表現最為精準。值得注意的是，若未經精確參數調優，語意切塊的表現可能低於滑動視窗，其強大表現源於對 min_tokens 的反覆調試。
2. 滑動視窗的穩定性： 滑動視窗透過 96 tokens 的 overlap 機制，在資訊覆蓋率與檢索穩定度之間取得了卓越平衡。它是避免檢索失敗的有效手段，特別適合對運算資源有限且需要穩定產出的場景。
3. 固定大小的限制與競爭力： 固定大小切塊在處理事實型 (Fact-based) 短問題時仍具競爭力。但在處理長敘述問題時，其缺乏「上下文銜接」的弱點會導致 LLM 評分大幅下降，反映出 RAG 系統對語意連續性的依賴。

8. 結論 (Conclusion)

本實驗證實，Chunking 策略的選擇對 RAG 系統的語意檢索品質具有決定性影響。透過實驗數據可以總結：沒有一種策略適用於所有情境，但「適當調整參數」——特別是語意切塊中的 min_tokens 設定——是提升系統對複雜語境理解能力的關鍵路徑。