Day22 - 使用向量資料庫作為長久記憶

沒接觸 AI 前，儲存對話只會透過 SQL 或 NoSQL 資料庫儲存對話內容，查詢資料也只能使用關鍵字搜尋，只要關鍵字不同就無法搜尋到內容，而昨天提到向量資料庫會先將文字向量化在儲存，並使用近似值搜尋的方式讓我們可以將語意相近的內容也一併找出，除了更容易找出資料外也不會有數量限制，因為查詢的內容是根據整個對話的資料搜尋

竟然這麼好用，趕緊來實作吧，程式只需要把 Day18 的 Advisor 改成 VectorStoreChatMemoryAdvisor 即可，另外因為 VectorStoreChatMemoryAdvisor 需要傳入向量資料庫的變數，所以我們需要先宣告變數讓 Spring 幫我們注入

```
@RequiredArgsConstructor
@Service
public class ChatService {
    private final ChatClient chatClient;
    private final VectorStore vectorStore;

    public String chat(String chatId, String userMessage) {
        return this.chatClient.prompt()
        .advisors(new VectorStoreChatMemoryAdvisor(vectorStor
        .user(userMessage)
        .call().content();
    }
}
```

VectorStoreChatMemoryAdvisor 的參數說明如下

- vectorStore: 向量資料庫的變數
- chatId: 對話識別 ID，查詢時會根據 ID 過濾資料
- 100: 查詢資料庫時最大的返回數量

趕快來驗證成果吧

```
Chat ID:1 User Message:幫我計算100*300
Chat ID:1 Assistant Message:100 * 300 = 30,000。
PromptTokens:47
GenerationTokens:10
TotalTokens:57
Chat ID:1 User Message:上個問題是?
Chat ID:1 Assistant Message:抱歉，我無法查看過去的問題。如果您有任何具體的問題或需要幫助的地方，請告訴我，我會很樂意幫助您！
PromptTokens:44
GenerationTokens:41
TotalTokens:85
```

咦？哪裡出了問題，透過向量資料庫反而查不到內容，上 Spring AI 的 Github 才發現原來這是 bug 阿，https://github.com/spring-projects/spring-ai/issues/800

> **Bug description**
>
> When using PGvector as vector store for **VectorStoreChatMemoryAdvisor** to save conversation history, there is org.postgresql.util.PSQLException "ERROR: syntax error at or near "conversationId"". The problem happens because in **adviseRequest** method in **VectorStoreChatMemoryAdvisor** single quotes added around **DOCUMENT_METADATA_CONVERSATION_ID** , which causes an error when parse it in JDBCTemplate.
>
> ```
> var searchRequest = SearchRequest.query(request
>             .withTopK(this.doGetChatMemoryRetri
>             .withFilterExpression(
>                     "'" + DOCUMENT_METADATA_CON
> ```

看到這個 Issue 還蠻傻眼的 **DOCUMENT_METADATA_CONVERSATION_ID** 一看也知道是常數，怎麼會被當成文字來組 Filter，不過凱文大叔還是得讓大家看到測試結果，在官方還沒更新前只好自己 Debug 了，原本想直接繼承 **VectorStoreChatMemoryAdvisor** 再覆蓋有問題的方法，不過程式中許多常數都設為 private，所以乾脆直接複製一個新的 **MyVectorStoreChatMemoryAdvisor** 完整程式碼可以在下面的 Source Code 查到，更改的內容如下

```
var searchRequest = SearchRequest.query(request.userText())
            .withTopK(this.doGetChatMemoryRetrieveSize(contex
```

```
            .withFilterExpression(
                DOCUMENT_METADATA_CONVERSATION_ID + "== '" +
```

另外針對近似值查詢凱文大叔也稍微調整一下程式

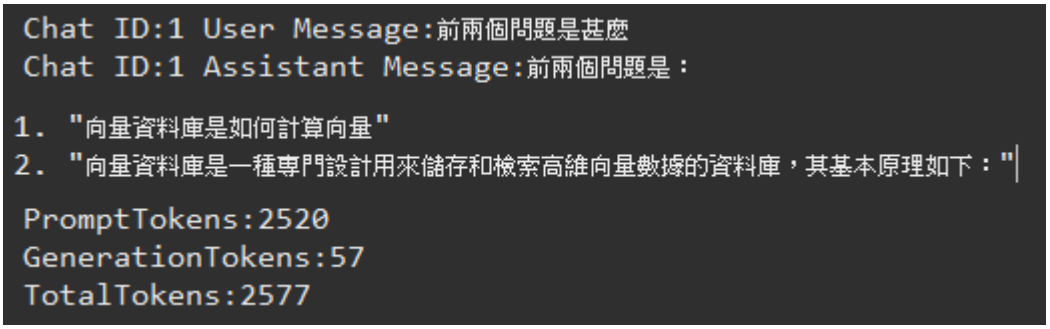
```
        List<Document> documents = this.getChatMemoryStore().

        String longTermMemory = documents.stream()
            .map(Content::getContent).distinct()
            .collect(Collectors.joining(System.lineSeparator(
```

近似值查詢出來的格式為 List<Document> documents，由於 Document 除了 context 還包含 meta，如果要將歷史對話組成字串可透過 map 的方式快速重組，上面這段內容凱文大叔只加了 .distinct()，做這個小小調整主要是程式剛上線或是測試時依定會問很多重複資料，透過這小小的修改就能將重複的內容過濾掉，減少我們送給 AI 的 Token 數量

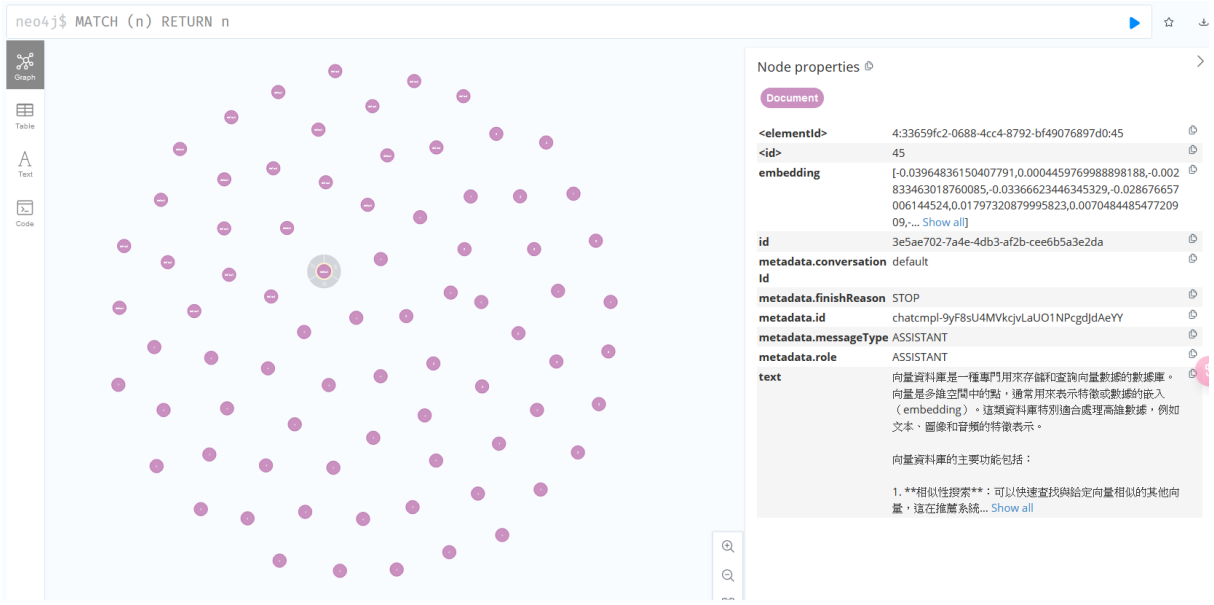
最後將 ChatService 中的 Advisor 改成自己改寫的就大功告成了

```
.advisors(new VectorStoreChatMemoryAdvisor(vectorStore, chatI
```

再來驗收一下結果



透過 Neo4j 網址也能看到每個對話節點: http://localhost:7474/browser/



問題討論

使用向量資料庫儲存聊天訊息，凱文大叔測試完有以下心得

1. 不適合回答順序性問題: 例如問上個問題是甚麼
2. 無法要求記住你提的要求: 例如請 AI 之後都用英文回答
3. 無法分辨是用戶問題還是 AI 的答案

利用向量資料庫儲存對話內容更適合用來查詢聊天訊息而不是用作對話記憶

改善方式

1. 目前向量資料庫儲存對話送給 AI 時會將查到的內容都轉為文字在加在 Prompt 上，導致無法區分用戶還是 AI 訊息，這裡可以改成 List<Message> 的方式傳送就能區分
2. 向量資料庫改為只儲存跟查詢，另外還是用短期記憶輔助將對話記憶傳送給 AI
3. 在企業使用可做為知識庫的內容，例如將問答做成客服系統的資料來源，當資料越多回答問題的結果也會越精準