基於語意嵌入與 Git 紀錄之開發知識搜尋系統

專題編號 : 114-1-CSIE-S015
執行期限 : 113 年第 1 學期至 114 年第 1 學期
指導教授 : 劉建宏
專題參與人員：111AB0040 杜坤翰
111332012 吳峻丞

一、摘要

本專題設計並實作一套程式碼歷史查詢系統，旨在提升軟體開發過程中對歷史修改紀錄的檢索效率與準確度。系統結合自然語言語意嵌入與程式碼語法結構分析，查詢流程整合多種語意嵌入模型與向量索引技術，並建立本地版本控制系統以維護資料版本安全與一致性。系統利用 LLM 輔助搜尋結果比對，加速、簡化用戶驗證搜尋結果、猜測的流程。

關鍵詞: 程式碼查詢 、自然語言處理 、語意嵌入 、抽象語法樹 、版本控制 、向量搜尋

二、緣由與目的

本專題源自 Sunbird 產學合作計劃在後端開發過程中所面臨的實際挑戰。我們觀察到在使用 Git 進行版本管理時，常出現以下問題：

- 紀錄描述不一致：Git commit 訊息的格式不一，標籤亦不具有辨識性。
- 缺乏開發脈絡:背景資訊不完整,導致難以理解修改原因；多版本並存時更容易誤判。
- 知識未被有效保存：需求與設計的演進多仰賴資深成員言傳，降低團隊效率。

雖然已有如 Sourcegraph 等工具結合語意搜尋與 Git 歷史，但其語意搜尋多僅限於 commit 層級，且依賴 DSL 描述結構，對缺乏後端經驗的開發者並不友善。

而 Git 歷史包含設計者的設計脈絡與需求背景，原本應是理解系統演進與評估新功能的重要參考。然而，這些紀錄常因缺乏有效、免費的查詢工具，使得開發者難以有效利用，降低了其實用性。

本專題因此提出一套程式碼歷史查詢系統。旨在協助開發者快速找出歷史紀錄及掌握修改背景，進而提升需求判讀與設計決策的效率與準確性。

三、研究範圍與目標

本專題聚焦於結合靜態程式碼結構與動態版本控制資料，設計一套適用於大型專案的智能查詢系統。系統的核心特色包括:

1. **自然語言向量搜尋輔助查詢**

   透過語句級向量模型對 PR 描述進行語意嵌入，並結合詞彙向量聚合 commit 群，實現以自然語言查詢觸發相關程式碼節點的檢索，減少對 DSL 的依賴，降低使用門檻。

2. **輔助路徑追蹤**

   當開發者對需求具有先備知識時，能輔助追蹤多層 Spring 標籤及事件驅動等非同步流程。

3. **跨版本與多入口點的脈絡分析方法**

   透過歷史 PR 與 commit 訊息重建設計脈絡，協助開發者理解不同入口點的設計背景與依賴關係。

4. **系統整合**

   開發模組化本地查詢系統，整合語意索引、LLM 與 Git 歷史擷取，並透過實際開發場景驗證系統在查詢速度、語意準確度與結構解釋能力等面向的整體表現。

四、使用技術方法

本系統結合語意理解、結構分析與互動驗證的混合式查詢架構提升歷史程式碼查詢的效果與易用性。整體流程如圖 1 所示，涵蓋資料同步、語意檢索、結構驗證、輔助搜尋：

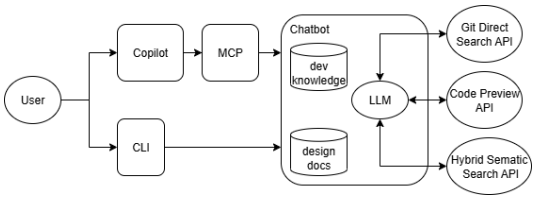


圖 1　混合式查詢流程圖

（一）資料同步與嵌入維護

- **版本一致性檢查**：每次用戶查詢時，系統自動比對本地資料與專案版本，不一致則觸發中介資料、嵌入資料的自動更新，確保查詢結果的時效性與正確性。
- **資料層更新**：利用 gitpython 與 GitHub API 同步 PR 與 commit 資料，並自動化維護語意嵌入（SBERT、FastText)。

（二）語意檢索與三因子猜測

- **PR 訊息語意搜尋**：採用 SBERT 將 PR 描述轉換為高維語意向量，並以 Milvus 提供向量索引、相似度搜尋，實現自然語言查詢與 PR 語意匹配。
- **壓縮 Commit 三因子搜尋**：FastText 分別對“Action”與“Target”進行詞向量嵌入，SBERT 用於“For”因子的語句級嵌入，組成三因子猜測模組，提升對 commit 語意的搜尋能力。
- **混合決策機制**：綜合語意檢索與三因子猜測，產生最終候選的 PR 列表，提升查詢的召回率與精確度。

（三）LLM 輔助搜尋

- **輔助關鍵字**：透過 few shot 將用戶問題自動組合不同搜尋引擎的關鍵字，方便使用於 Copilot。
- **開發經驗輔助**：當搜尋需求與過往的開發經歷相關時，透過 RAG 從開發文件與筆記中擷取相關關鍵字，並以此輔助搜尋流程。
- **過濾雜訊**：透過類似 Rubric-based judging 的方式對搜尋結果做過濾及評分。

（四）系統架構與穩定性

- **分層模組化設計**：資料同步、語意嵌入、語意檢索等模組獨立部署，使系統易於維護與擴充。
- **資料一致性保障**：確保查詢流程與版本管理同步，避免資料延遲造成的查詢問題。

五、進行方式與階段成果

- 預計完成目標 F1 score ≥ 0.6

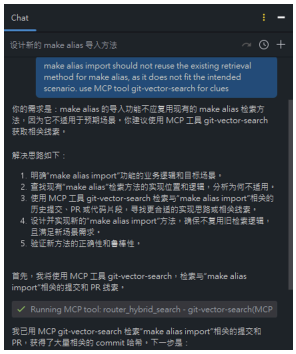
表 1　專題實作成果展示

Precision	Recall	F1 score
0.53	0.39	0.45

月份	目標與預計完成項目	完成程度
四	完成基本Git提取的Prototype	完成
	完成基本資料前處理的Prototype	完成
	完成基本文字嵌入的Prototype	完成
	完成基本向量搜尋的Prototype	完成
	完成基本版本控制的Prototype	完成
五	優化提取流程	完成
	優化版本管理方式	完成
	對專案Git訊息進行統計分析	完成
	優化前處理	完成
	完成基本圖資料庫的Prototype	效能問題，棄用
六	優化圖資料庫前處理流程	效能問題，棄用
	加入Git資訊	完成
七~八	實際開發時測試使用效果	延後
	紀錄測試發現的問題	延後
	整合 LLM 輔助搜尋	50%
九~十	整合 LLM 輔助搜尋	完成
	實際開發時測試使用效果	完成
	紀錄測試發現的問題	完成

圖 2　工作計畫表

六、實驗結果



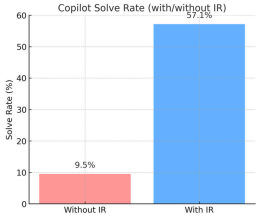


圖 3　實際完成結果　圖 4　copilot 導入成效

用戶可以透過 copilot 使用語義搜尋的服務，copilot 取得候選集後依照優先度驗證並取得詳細 commit 訊息，最終透過上述搜尋步驟取得線索並完成用戶需求。

雖然在搜尋指標上表現不太穩定，但在實驗的測試內有近 6 成問題(如圖 4 所示)在 copilot 使用了我們的搜尋系統後得到了部分甚至完全正確的解決方向。

七、結論

期待本搜尋系統能提升 Sunbird 專案的開發效率與程式碼查詢便利性，讓知識搜尋、跨團隊開發變得更有效率，並能實際幫助到學生學習。