**國立中山大學**

**113年度鼓勵大學部學生參與專題研究計畫申請書**

一、綜合資料:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 申 請 人 【 學 生 】 | 姓 名 | 鄔仁迪 | 系（所）及  年級 | 資訊管理系大學部  三年級 |
| 身 分 證  號 碼 |  | 學 號 |  |
| 電 話 |  | | |
| 學生研究  計畫名稱 | 自然語言導向開發程式系統驗證與評估 | | |
| 研究期間 | 113 年 7 月 1 日 起 至 114 年 2 月 28 日止 | | |
| 指  導  教  授 | 姓 名 | 康藝晃 | | |
| 系 (所) | 資訊管理學系 | | |
| 職 稱 | 專任副教授 | 電 話 | 07-5252000 |

1. 研究計畫摘要—請對研究計畫之問題依下列各項述明：

(1)摘要

調查顯示，超過70%的程式設計師使用生成式 AI 輔助編程，顯示其普及性。然而現有環境以程式碼為導向，缺乏高品質指示支持和專案結構統整規劃，限制了效率和質量的提升。本作品之核心在於研究自然語言導向開發（NLOD）系統，主要目標是建立一套融合自然語言處理技術的程式開發環境，並且能夠針對現有開發環境中的專案結構建議、文件自動建構、程式碼自動生成及指示品質進行改進等功能，期望提升開發效率、程式碼品質和降低程式開發的難度。

開發階段分為四個階段，每個階段都有明確的主題，並且承上啟下，第一階段目標為基礎問答功能，第二階段目標為圖像與音訊服務，第三階段目標為程式碼編輯區功能和第四階段代理人自動完成。

(2)研究動機與研究問題

一張含有 文字, 螢幕擷取畫面, 軟體, 多媒體軟體 的圖片

自動產生的描述

* 軟體開發已經與生成式AI無法脫離 : 根據 Survey reveals AI’s impact on the developer experience 調查收集超過 100 個北美公司員工，高達 92%受訪者 有使用生成式 AI協助開發，並有 70 % 用戶認為這項技術給予工作優勢。 [1]使用者用AI協助撰寫程式
* 現有開發環境限制：當前的程式開發環境主要以程式碼為導向，儘管一些環境已經嵌入了生成式 AI 的輔助功能，但仍然缺乏對於專案結構劃分和建議的統整支持，限制了開發效率與質量的進一步提升。
* 語言模型對變更的追蹤不足：語言模型生成內容大部份都需要額外修正後才能有效使用，但使用線上功能的語言模型無法自動追蹤用戶修正過後的程式碼，且易受聊天記錄干擾，導致需要重複給予修改指示，降低開發效率。
* 模型對當前開發狀態理解不足：軟體開發應該是有序建立檔案，並依照順序開發，語言模型很常只關注自己當前任務，而忽視應優先處理項目，例如: 應該在撰寫 API 前先建立模型定義。
* 指示好壞是依賴於當前開發階段 : 對於不同項目指示會有不同寫法，像是 Python 與 Java 開發邏輯就差異很大。即便是在同個項目中，不同組件也需要依據功能進行動態調整指示。
* 程式編寫規範無法簡單統一 : 大型專案由多人完成，程式編寫者有可能對公司或團體的規範有不清楚或遺漏，需要系統能夠在使用者的編寫過程中和最後繳交時檢驗程式是否完全符合團體規範。

(3)文獻回顧與探討

3.1 運用AI輔助程式編寫的困境

大型語言模型（LLM)是基於大量資料進行訓練的超大型深度學習模型。基礎轉換器是一組神經網絡，這些神經網絡由具有自我專注機制的編碼器和解碼器組成，可以從一系列文字中提取含義並理解其中字詞與片語間的關係。LLM可用於生成式AI，根據人類語言輸入的指示產生內容，也可產生程式碼來解決各種用自然語言表達的問題。目前此項技術的應用已相當廣泛，然而將AI用於輔助程式開發依舊會面臨一些困境與挑戰。

Adavit Sarkar(2022)提出使用AI輔助程式編碼時，可能會產生語法不正確或未定義的程式碼，並會呼叫未定義或是超出作用域的函數、變數和屬性。此外，程式碼的可讀性、完整性和安全性都可能會出現潛在的風險與問題。

3.2 自然語言指示的整合開發環境

Prompt Engineering是指設計和建構一個提示或問題，以指導機器生成所需的內容或回答。此技術主要應用於自然語言處理的領域，用於控制語言模型的輸出。一個良好設計的指示可以引導模型生成精確的回答，而不良的指示可能會導致不準確或無意義的結果。

Stephen H. Bach(2022)提出三個建構自然語言指示庫的關鍵點。首先，採用模板語言來表示指示，使指示的運用更加靈活，無需編寫和讀取程式碼也增加開發環境的靈活性。其次，建構管理指示的工具，使開發人員能根據自己的需求找到適合的指示。最後，社區驅動的品質標準使指示的應用範圍更加廣泛。

3.3 Diffusion Models 在影像合成的應用

擴散模型（Diffusion Models）是一種生成模型，被廣泛應用於生成圖片的領域。對資料集中的圖片不斷加上高斯雜訊，讓原本的圖片逐漸變成完全的雜訊，而模型主要的工作就是將雜訊修復回原圖，並透過輸入隨機雜訊來生成新的圖片。Dayin Wang(2023)在研究中訓練模型利用LLM使文字指示更加完善，並且透過輸入文字指示生成出更加精美的圖片，還可透過指示為圖片增添細節。

(4)研究方法及步驟

4.1 確立研究主題:

透過檢閱課堂調查與相關研究，可以了解到學生和工程師越來越依靠於生成式AI的輔助撰寫程式碼，但是現行的生成式AI輔助仍有開發環境限制、語言模型對變更的追蹤不足等問題，會造成開發效率及體驗不佳，因此才希望能開發自然語言導向開發環境改善問題。

4.2 確認系統需求:

進行使用者需求分析，基於文獻探討、網路上相關的開發經驗和自身的使用體驗，深入了解開發者在現行程式開發環境中所遇到的限制和挑戰，以這些去明確定義此自然語言程式開發環境的功能需求。這可能包含高品質提示支持、專案結構建議、文檔自動建構等功能。

4.3 系統設計:

在此階段，將基於確定的系統需求進行系統設計，包含選擇適當的程式語言、演算法和相關的技術架構，這一部分也將考慮語言模型的整合、自然語言處理技術的應用，以及系統組件之間的協同作業，除此之外，確認開發日程、開發順序和介面也是此階段的工作，目前設想專案分成五個區塊:

第一區塊，開發專案環境建置，例如推薦使用框架或軟體套件以達成該目標，選擇與建議程式語言區。

第二區塊，建議應先處理哪個區塊，與用戶交互改區塊作用，並撰寫樣本。

第三區塊，完成樣板程式，並同時撰寫測試組件程式，並處理報錯信息，完成該組件。

第四區塊，當用戶移動到新的檔案總結已經完成函數以及哪些共有函數後回到第三步驟。

額外區塊，參考程式碼生成角色造型並依照造型寫適配腳本。

4.4系統開發:

在程式開發實際階段，我們將按照設計階段規劃，進行自然語言導向的程式開發環境的模組化開發，包括專案環境建置、用戶交互、樣板程式和測試組件程式。同時，系統開發過程中，我們致力構建自然語言導向的程式開發環境，包括核心模型、生成指示和建議的演算法，以及整合自然語言處理和AI技術的程式碼生成。這將提供開發者直覺、高效且質量優良的程式開發體驗。

4.5 系統測試與優化

完成系統開發後，進行綜合性的測試階段，包括功能性、性能和用戶體驗測試，以確保系統各方面運作正常且符合使用者期望。測試結果將成為系統優化的基礎，透過修復可能出現的錯誤和提升效能，致力確保系統的穩定運行。這個階段不僅涵蓋單一面向的測試，還注重整體系統表現，以提供使用者高品質的程式開發環境。

4.6 成果評估

設計實驗和建立評估標準，以檢驗自然語言導向的程式開發環境對降低程式開發門檻和提升開發效率的影響。特別關注經驗較少的開發者，評估系統在提供幫助和改善開發體驗方面的效果。

收集使用者的反饋，根據實際使用情況進行調整和改進，確保系統的實用性和可持續性。

(5)預期結果

5.1 預期成果

一套以自然語言為基礎的程式開發環境，融合對高品質指示的支持、專案結構規劃建議和文件自動建構功能，能夠理解和處理自然語言指示，並提供高品質的程式碼生成和修改建議，也使語言模型能夠在整個開發過程中實時追蹤和理解程式碼的生成與用戶進行的修改，增強模型對於用戶修改的適應性和靈活性。

提升模型理解當前開發狀態的系統性整理機制，使模型更好地理解和追蹤大型專案的多檔案結構和開發進度。

5.2 學術貢獻

在程式開發環境中有效整合defusion機制，探討開發流程的融合策略，為程式開發者提供更豐富的開發工具和素材，也降低開發門檻並優化用戶體驗，有助於改善程式開發過程中的效率和品質，推動自然語言處理技術在軟件開發領域的應用和發展。

5.3 社會貢獻

設計實驗和建立評估標準，檢驗開發環境對於降低程式開發門檻和提升開發效率的影響，尤其針對經驗較少的開發者，評估系統能夠在多大程度上提供幫助和改善開發體驗，進而為軟件開發領域提供新的理論和實證。

(6)參考文獻

1. Inbal shani. (2023, June 13). Survey Reveals AI’s Impact on the Developer Experience. Github Blog. [https://github.blog/2023-06-13-survey-reveals-ais-impact-on-the-developer-](https://github.blog/2023-06-13-survey-reveals-ais-impact-on-the-developer-experience/#why-developer-experience-matters)experience/#why-developer-experience-matters
2. Jules white, Sam hays, Quchen fu, Jesse spencer-smith, & Douglas c. schmidt. (2023). ChatGPT Prompt Patterns for Improving Code Quality, Refactoring, Requirements Elicitation, and Software Design. ArXiv. [https://doi.org/https://arxiv.org/pdf/2303.07839.pdf](https://doi.org/https:/arxiv.org/pdf/2303.07839.pdf)
3. 什麼是大型語言模型 (LLM)？. (n.d.). Amazon Web Services. <https://aws.amazon.com/tw/what-is/large-language-model/>
4. Advait sarkar, Andrew d. gordon, Carina negreanu, Christian poelitz, Sruti srinivasa ragavan, & Ben zorn. (2022). What Is It like to Program with Artificial Intelligence? ArXiv. [https://doi.org/https://arxiv.org/pdf/2208.06213.pdf](https://doi.org/https:/arxiv.org/pdf/2208.06213.pdf)
5. Alpha camp. (2023, October 2). Prompt Engineering 提示工程是什麼？新手必學指南. ALPHA Camp. <https://tw.alphacamp.co/blog/prompt-engineering>
6. Stephen h. bach, Victor sanh, & Zheng-xin yong. (2022). PromptSource: An Integrated Development Environment and Repository for Natural Language Prompts.ArXiv. [https://doi.org/https://arxiv.org/pdf/2202.01279.pdf](https://doi.org/https:/arxiv.org/pdf/2202.01279.pdf)
7. Tu32. (2023, May 26). 〈 Diffusion Model 論文研究與實作心得 Part.1 〉 前言與圖片雜訊前處理. HackMD. <https://hackmd.io/@Tu32/B1-m6Tuai>
8. Dayin wang, Chong ma, & Siwen sun. (2023). Novel Paintings from the Latent Diffusion Model through Transfer Learning. *Applied Sciences*. [https://doi.org/file:///C:/Users/USER/Downloads/applsci-13-10379%20.pdf](https://doi.org/file:/C:/Users/USER/Downloads/applsci-13-10379%20.pdf)

(7)需要指導教授指導內容

有關 VS Code 插件環境編寫方式

需要協助指導開發以程式開發環境，這將涉及使用VS Code提供的API來建立和設置插件環境，從建立設定檔保存與讀取文件以及與終端機交互溝通的方式。還有用戶操作追蹤相關的API，包括如何設置默認配置，如何處理用戶自定義配置，以及如何確保配置文件的一致性和完整性。

語言模型指示

如何構築有效指示，這將包括設計和實現指示解析器，以解析和理解用戶提供的指示和如何處理各種指示的例外情況。此外，如何處理與檔案連結相關的指示輸入，以便系統能夠理解和跟踪當前用戶的狀態。

分析現有的程式專案開發流程

歸納並建議一個完好的系統適配特定任務，包括評估需求收集、設計、開發、測試和部署等各個階段的流程和方法，這會需要比較有經驗開發大型項目的人執導。以上都需要教授與學生一起探討各種流程中可能出現的問題和挑戰，並提出解決方案和改進建議。

**國立中山大學鼓勵大學部學生參與專題研究計畫**

**指導教授初評意見表**

1. 學生潛力評估：
2. 對學生所提研究計畫內容之評述：
3. 指導方式：

承諾指導教授簽名：＿＿＿＿＿＿

年 月 日