透過問題重塑以及微調LLM提升程式碼補全能力

專題編號：114-1-CSIE-S008
執行期限：113年第1學期至114年第1學期
指導教授：孫勤昱
專題參與人員： 111590018 黃政豪
111590019 廖峰瑞
111590037 陳奕翔
111590045 張峻誠

一、 摘要

近期大語言模型在程式碼生成方面展現了卓越的性能，然而，其龐大的硬體需求限制了可用性與可擴展性。這種能力與可部署性之間的差距，凸顯了我們需要一種能平衡性能與效率的方法。

為應對此挑戰，我們提出一種策略，旨在增強整合agentic pipelines中與適合消費級硬體的小語言模型的程式碼編寫能力。我們引入一個輕量級的兩階段工作流程，該流程結合了prompt engineering與fine-tuning。在第一階段，我們將prompt重塑為一種結構化的表示形式，使其與微調資料的格式對齊。在第二階段，一個特化的模型會根據這些重塑後的prompt來生成程式碼。

結果顯示，我們的兩階段流程在性能上比肩大模型，且能運作在消費級電腦上。更廣泛地說，在代理流程中結合提示工程與微調，為建構強大的程式碼編寫助理提供了一種有效且經濟的方法。

關鍵字：Large language model, Fine-tune, Prompt Engineering, Agentic AI

二、緣由與目的

隨著語言模型在程式碼生成領域展現出巨大潛力，對於保障隱私與降低對雲服務商依賴的需求也日益提升。因此，開發一個能在本地運行的模型具有重要意義。本專題的目標為：

(一)、結合微調與prompt engineering來提升的小語言模型程式碼生成能力。

(二)、替換agentic pipeline基礎模型，並且比較其程式碼生成的表現。

(三)、開發Web UI，方便使用者與模型互動。

三、使用技術方法

本章節將說明本專題的核心實作內容與模型選擇，探討微調方法對模型的影響、agentic pipeline的設計與評估方式：

(一)、Agentic pipeline

1. 第一階段：

使用我們訓練好的模型將prompt重塑為一種結構化的表示形式，此結構符合第二階段模型微調時的輸入。

輸入：function signature

輸出：重塑過的輸入

2. 第二階段：

使用我們訓練好的模型將根據重塑過的prompt產生程式碼。

輸入：第一階段模型的輸出

輸出：完整的function程式碼

(二)、微調方法

模型:

1、Gemma-3-270m-it[1]

2、Llama3.2-1B-Instruct[2]

3、Qwen2.5-Coder-1.5B-Instruct[3]

資料集:OpenCodeInstruct[4]

硬體:8張NVIDIA H100 SXM

(三)、評估指標

評估指標為HumanEval[5]。

(四)、 Web UI

我們的Web UI透過Python的Gradio函式庫加速建構流程，避免耗費人力與時間。

四、實驗結果

本章節將概述本專題達成的成果，包括微調後模型的改進程度、agentic pipeline與模型本地化三大面向，具體如下：

(一)、微調後模型的改進程度

透過HumanEval指標，比較原始模型與微調後模型的生成品質差異。其中，微調後的Qwen2.5-Coder-1.5B結合agentic pipeline後在指標中達到79.26分，超過原模型的64.63分

下表為模型在**微調前**套用在Agentic pipeline中的pass@1分數

2nd-stage / 1st-stage	Llama3.2-1B Instruct	Qwen2.5 Coder-1.5B
Llama3.2 1B Instruct	40.24	60.36
Qwen2.5 Coder-1.5B	46.95	64.63

下表為模型在**微調後**套用在Agentic pipeline中的pass@1分數

2nd-stage / 1st-stage	Llama3.2-1B Instruct	Qwen2.5 Coder-1.5B
Llama3.2 1B Instruct	60.36	76.82
Qwen2.5 Coder-1.5B	60.97	79.26

(二)、Agentic pipeline

結合兩個小模型，在效率與表現上均勝過大模型。

(三)、模型本地化

透過實現可在本地部屬的小型程式碼生成模型，使開發者可不依賴雲端大模型，保護隱私。

五、 參考文獻

[1]Gemma Team, “Gemma 3 Technical Report,” arXiv preprint, arXiv:2503.19786, 2025. [Online]. Available: https://arxiv.org/abs/2503.19786

[2] A. Dubey et al., “The LLaMA 3 herd of models,” arXiv preprint, arXiv:2407.21783, 2024. [Online]. Available: https://arxiv.org/abs/2407.21783

[3] Binyuan Hui et al., “Qwen2.5-Coder Technical Report,” arXiv preprint, arXiv:2409.12186, 2024. [Online]. Available: https://arxiv.org/abs/2409.12186

[4]Wasi Uddin Ahmad et al., “OpenCodeInstruct: A Large-scale Instruction Tuning Dataset for Code LLMs,” arXiv preprint, arXiv:2504.04030, 2025. [Online]. Available: https://arxiv.org/abs/2504.04030

[5] M. Chen et al., “Evaluating large language models trained on code,” arXiv preprint, arXiv:2107.03374, 2021. [Online]. Available: https://arxiv.org/abs/2107.03374