

基於人工智慧的程設遊樂園

AI-Assisted Coding Playground

指導教授：高宏宇

專題成員：杜雲笙、陳詠翰

開發工具：Python、Pytorch

測試環境：Ubuntu 22.04

一、簡介：

近年來，大量的預訓練語言模型興起，而生成式預訓練模型甚至能夠根據指令產生相對應的程式碼。我們所熟知的 Github CoPilot、Amazon Codewhisperer、OpenAI GPT4模型皆具有非常大量的參數 e.g. CoPilot 使用1.75T（1.75兆）個參數。參數量的多寡會嚴重的影響模型表現，而且需要花費企業等級的資源，如 A100 GPU 與 TPU，既耗費硬體資源又耗費時間。因此，有許多人會選擇使用較輕量的模型並進行 finetune。而本專題提供一個嶄新的模型 finetune 流程（Fig. 1）。

本專題想探討在眾多厲害的模型並存的環境下，是否能讓這些模型相互學習進而提升他們的表現，使用較成熟的 AI 模型提供的服務來輔助訓練較弱勢的 AI 模型。為了驗證我們的猜測，我們透過 ChatGPT 給予步驟提示與回饋來輔助輕量版預訓練模型產生正確的結果並進一步的根據這些正確的結果來 finetune 輕量版的模型。本專題提出的流程特點如下：一、能夠完全自動化。二、Finetune 後的模型生成程式碼品質超過原來的模型。

我們專題所挑選的輕量版預訓練模型為 HuggingFace 所提供的 CodeGen 模型。CodeGen 一系列的模型有高至16B（160億）低至350M（3.5億）個參數。我們用來實驗的模型為 CodeGen-350M-mono。

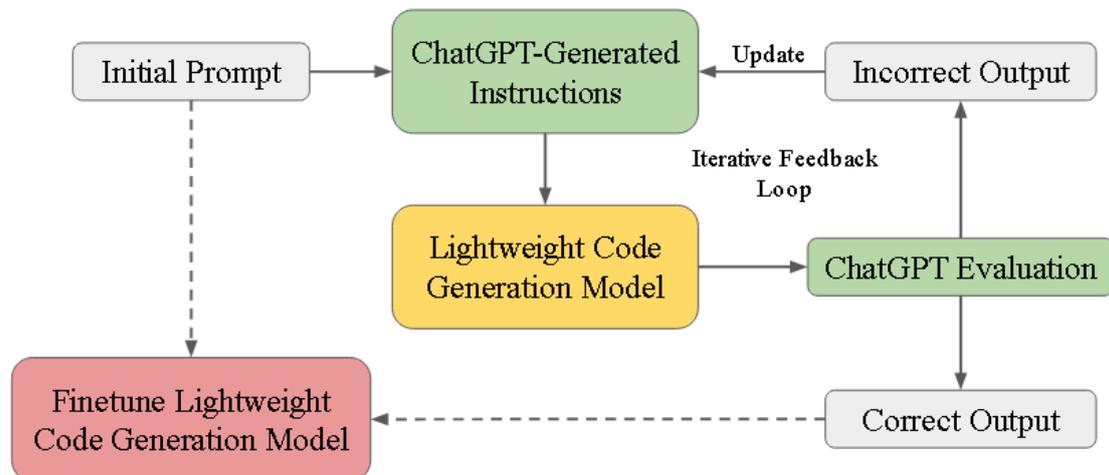


Fig. 1：流程圖

二、 測試結果：

以下呈現本專題流程的其中一個案例：

Initial Prompt：Write a program to sort a list of strings in lexicographical order using the Insertion Sort algorithm

- ChatGPT 步驟提示與回饋第一輪結果：

- ChatGPT 步驟提示所生成的程式碼

```
def insertion_sort(list):
    for i in range(1, len(list)):
        key = list[i]
        j = i - 1
        while j ≥ 0 and key < list[j] :
            list[j + 1] = list[j]
            j -= 1
        list[j + 1] = key
    return list
```

- ChatGPT 回饋：程式碼幾乎正確，但是在 python 變數命名不能使用 list。

ChatGPT 根據評估結果更新提示並進入第二輪。

- ChatGPT 步驟提示與回饋第二輪結果：

- ChatGPT 更新的步驟提示所生成的程式碼

```
def insertion_sort(lst):
    for i in range(1, len(lst)):
        key = lst[i]
        j = i - 1
        while j ≥ 0 and key < lst[j] :
            lst[j + 1] = lst[j]
            j -= 1
        lst[j + 1] = key
    return lst
```

- ChatGPT 回饋: 程式碼正確！

經過迭代，模型成功生成正確的程式碼，而結果也會被存下來被用來進一步 finetune 輕量版模型。