

jumbokh /  
ml-class2025

&lt;&gt; Code

Issues

Pull requests

Actions

Projects

Wiki

Security



main



ml-class2025 / lab / Lab1.md



jumbokh change file name

d1720b2 · 1 minute ago



165 lines (130 loc) · 5.83 KB

設計這樣一個實驗是一個非常有意義的教學項目！讓學生通過下指令讓AI生成程序並執行，可以幫助他們更好地理解算法的概念、數據結構的應用以及如何與AI進行互動。以下是為你提供的一個詳細方案：

## 實驗目標

1. 學生能夠通過自然語言（例如中文）向AI提出需求。
2. AI根據學生的需求生成Python代碼，實現數據結構或算法。
3. 學生可以在Visual Studio Code中執行生成的代碼並觀察結果。

## 實驗準備

### 1. 硬件與軟件

- 設備：每位學生需要一台手機（安裝微信小程序）和一台電腦（安裝Visual Studio Code）。
- 軟件：
  - 微信（手機端，用於運行小程序）。
  - Visual Studio Code（電腦端，用於編輯和執行Python代碼）。
  - Python環境（確保已安裝Python解釋器）。

### 2. AI服務

- 選擇一個適合的AI服務提供商，例如：
  - **DeepSeek**：提供自然語言處理和代碼生成能力。
  - **ChatGPT 或 GPT-3.5/Turbo**：OpenAI提供的強大語言模型，可以生成Python代碼。

- **Google Bard** 或其他類似工具。

### 3. 微信小程序開發

- 使用微信小程序開發工具（基於JavaScript）來實現以下功能：
  - 收集學生的指令（例如：“請生成一個二分查找算法”）。
  - 將指令傳送至AI服務，獲取Python代碼。
  - 顯示生成的代碼並提供下載或複製選項。

## 實驗步驟

### 第1步：設計小程序界面

在微信小程序中設計簡單的用戶界面，讓學生可以輸入指令並查看結果。以下是基本功能模塊：

- **文本框**：供學生輸入需求（例如：“請生成一個冒泡排序算法”）。
- **按鈕**：點擊後將指令傳送至AI服務。
- **顯示區域**：展示AI返回的Python代碼。

### 第2步：連接AI服務

選擇一個API接口，讓小程序能夠與AI模型交互。以下是示例（假設使用OpenAI的ChatGPT API）：

1. 註冊並獲取API密鑰。
2. 在小程序中發送HTTP請求，將學生的指令傳送至AI模型。
3. AI返回Python代碼後，在小程序界面中顯示。

### 第3步：設計數據結構與算法實驗

根據教學目標，設計一些典型的數據結構或算法題目。例如：

- **二分查找**：讓AI生成二分查找的Python實現。
- **冒泡排序**：讓AI生成冒泡排序的代碼並解釋其實現原理。
- **樹狀結構遍歷**：讓AI生成一棵二叉樹，並實現前序、中序和後序遍歷。

### 第4步：執行代碼

在小程序中提供一個“複製代碼”按鈕，將生成的Python代碼複製到剪貼板。學生可以將其粘貼到Visual Studio Code中，然後執行並觀察結果。

## 具體實現細節

## 1. 微信小程序代碼示例

以下是小程序中連接AI服務的基本代碼框架（基於JavaScript）：

```
// 小程序頁面
Page({
  data: {
    inputText: '',
    generatedCode: ''
  },

  // 學生輸入指令
  bindInput: function(e) {
    this.setData({
      inputText: e.detail.value
    });
  },

  // 連接AI服務並生成代碼
  generateCode: async function() {
    const input = this.data.inputText;

    try {
      // 發送HTTP請求至後端或直接調用API
      const response = await wx.request({
        url: 'YOUR_API_ENDPOINT', // 替換為你的API地址
        method: 'POST',
        data: { prompt: input },
        header: {
          'Content-Type': 'application/json',
          'Authorization': 'Bearer YOUR_API_KEY' // 替換為你的API密
        }
      });

      this.setData({
        generatedCode: response.data.code
      });
    } catch (error) {
      console.error('請求失敗：', error);
    }
  },

  // 複製代碼到剪貼板
  copyToClipboard: function() {
    wx.setClipboardData({
      data: this.data.generatedCode,
      success: function(res) {
        wx.showToast({
          title: '已複製到剪貼板',
          icon: 'success'
        });
      }
    });
  }
});
```



```
    });  
  }  
});
```

## 2. AI代碼生成示例

以下是AI可能返回的Python代碼示例（以二分查找為例）：

```
def binary_search(arr, target):  
    left = 0  
    right = len(arr) - 1  
  
    while left <= right:  
        mid = (left + right) // 2  
        if arr[mid] == target:  
            return mid  
        elif arr[mid] < target:  
            left = mid + 1  
        else:  
            right = mid - 1  
    return -1  
  
# 測試數據  
sorted_array = [1, 3, 5, 7, 9, 11]  
target_value = 7  
result = binary_search(sorted_array, target_value)  
  
print(f"目標值 {target_value} 的索引是 : {result}")
```



## 3. 在Visual Studio Code中執行代碼

學生可以將生成的Python代碼粘貼到Visual Studio Code中，然後按 **F5** 或點擊“運行”按鈕來執行程序。觀察輸出結果，驗證算法的正確性。

## 實驗總結

1. 學生通過小程序下達指令，與AI進行交互。
2. AI根據需求生成Python代碼，實現數據結構或算法。
3. 學生在Visual Studio Code中執行代碼並觀察結果。
4. 教師可以引導學生分析代碼的邏輯和複雜度，進一步強化對數據結構與算法的理解。

## 教學建議

1. 提供多個實驗題目，讓學生熟悉不同的算法（例如排序、搜索、樹狀結構等）。
2. 要求學生在執行程序後，總結代碼的功能和實現原理。
3. 鼓勵學生修改代碼參數或數據，觀察結果的變化，進一步理解算法的工作機制。

希望這些步驟能夠幫助你設計一個有趣且有意義的實驗！如果有其他需求或問題，隨時告訴我！ 😊