如何写好Prompt: Chain-of-Thought

30 Jun 2023 **@lijigang**

是什么

> 在撰写 **Prompt** 时, 使用 **CoT(Chain of Thought)** 技巧是指让 **GPT** 输出一步步的中间状态, 最终完成目标.

在网上看到的各种 **Prompt** 小技巧, 很常见的一条就是 **"let's do it step by step"** 之类的, 输出质量确实有明显的提升, 其原理以及量化的对比, 可以阅读论文 **[2201.11903] Chain-of-Thought Prompting Elicits Reasoning in Large Language Models**.

怎么用

在如何写好Prompt: 结构化 的 **## Workflow** 模块, 我们除了写明交互工作流程以外, 还可以结合 **CoT** 技巧来明确其中关键环节的中间状态输出, 进一步提升 **GPT** 的输出质量.

> **[SYS]:**
>
> - **author: Arthur**
> - **version: 0.1**
> - **language:** 中文
> - **description:** 十年经验的 **Python** 开发专家, 擅长使用 **PyQT6** 开发图形软件。
>
> **## Goals:** 通过 **Python** 代码实现用户提出的需求.
>
> **## Constrains:**
>
> - 仅限于在 **macOS** 系统下提供支持；
> - 专注于使用 **PyQT6** 进行图形软件的开发。

Skills:

- 丰富的 **Python** 开发经验；
- 熟练使用 **PyQT6** 开发图形软件；
- 熟悉 **macOS** 系统的 **Python** 开发环境。

Workflows:

- 初始化：作为 **Python** 开发专家，使用中文与用户进行交流，友好地欢迎用户并介绍自己的角色和技能；
- 理解需求：根据用户的需求和问题，拆分你的代码实现思路**(**拆分为五步**)**
- 友好提示: 如有软件包依赖, 提示用户进行安装
- 输出代码: 将上一步的代码思路使用 **Python (**结合 **PyQT6)** 实现

[ME]:

1. 有一个图形界面, 支持用户录入如下几个字段:
   - 日期: <默认当日, 支持编辑>
   - 百度渠道: <整数类型>
   - 头条渠道: <整数类型>
   - 腾讯渠道: <整数类型>
2. 在同一目录下已经有一个 **Excel** 文件 **“data.xlsx”**
3. 该界面有一个"确定"按钮, 点击后, 将录入的数据添加到该 **Excel** 文件中, 同时清空图形中的数据
4. 该界面有一个"曲线"按钮, 点击后, 将读取 **Excel** 文件中**(“data.xlsx”)**, 并画出曲线图

示例

ChatGPT 3.5 Respond:

```
#!/usr/bin/env python3

import sys
from PyQt6.QtWidgets import QApplication, QWidget, QLabel, QD
from PyQt6.QtCore import Qt, QDate
from openpyxl import Workbook, load_workbook
```

```
import matplotlib.pyplot as plt

plt.rcParams["font.sans-serif"]=["SimHei"] #设置字体
plt.rcParams["axes.unicode_minus"]=False #该语句解决图像中的“-”负

class DataInputWidget(QWidget):
    def __init__(self):
        super().__init__()

        # 创建日期字段
        date_label = QLabel('日期:', self)
        date_label.move(20, 20)
        self.date_edit = QDateEdit(self)
        self.date_edit.move(100, 20)
        self.date_edit.setDate(QDate.currentDate())

        # 创建百度渠道字段
        baidu_label = QLabel('百度渠道:', self)
        baidu_label.move(20, 60)
        self.baidu_edit = QLineEdit(self)
        self.baidu_edit.move(100, 60)

        # 创建头条渠道字段
        toutiao_label = QLabel('头条渠道:', self)
        toutiao_label.move(20, 100)
        self.toutiao_edit = QLineEdit(self)
        self.toutiao_edit.move(100, 100)

        # 创建腾讯渠道字段
        tencent_label = QLabel('腾讯渠道:', self)
        tencent_label.move(20, 140)
        self.tencent_edit = QLineEdit(self)
        self.tencent_edit.move(100, 140)

        # 创建确定按钮
        confirm_button = QPushButton('确定', self)
        confirm_button.move(20, 180)
        confirm_button.clicked.connect(self.add_data_to_excel

        # 创建曲线按钮
        curve_button = QPushButton('曲线', self)
        curve_button.move(120, 180)
        curve_button.clicked.connect(self.plot_curve)

        self.setGeometry(300, 300, 300, 220)
        self.setWindowTitle('数据录入')
        self.show()
```

```
def add_data_to_excel(self):
    date = self.date_edit.date().toString("yyyy-MM-dd")
    baidu = int(self.baidu_edit.text())
    toutiao = int(self.toutiao_edit.text())
    tencent = int(self.tencent_edit.text())

    try:
        wb = load_workbook('data.xlsx')
        sheet = wb.active
    except FileNotFoundError:
        wb = Workbook()
        sheet = wb.active
        sheet.append(['日期', '百度渠道', '头条渠道', '腾讯渠道']

    row_data = [date, baidu, toutiao, tencent]
    sheet.append(row_data)

    wb.save('data.xlsx')

    # 清空数据
    self.date_edit.setDate(QDate.currentDate())
    self.baidu_edit.clear()
    self.toutiao_edit.clear()
    self.tencent_edit.clear()

def plot_curve(self):
    try:
        wb = load_workbook('data.xlsx')
        sheet = wb.active
        data = list(sheet.values)[1:]
        dates = [row[0] for row in data]
        baidu = [row[1] for row in data]
        toutiao = [row[2] for row in data]
        tencent = [row[3] for row in data]

        fig, ax = plt.subplots()
        ax.plot(dates, baidu, label='百度渠道')
        ax.plot(dates, toutiao, label='头条渠道')
        ax.plot(dates, tencent, label='腾讯渠道')
        ax.set_xlabel('日期')
        ax.set_ylabel('渠道')

        # 将纵坐标范围设置为从零开始
        ax.set_ylim(bottom=0)

        ax.legend()
        plt.show()
```

```
        except FileNotFoundError:
            print('Excel 文件不存在')
if __name__ == '__main__':
    app = QApplication(sys.argv)
    data_input = DataInputWidget()
    sys.exit(app.exec())
```