人工智能基础大作业报告

陈凯丰[1] 队员姓名 2[2] 队员姓名 3[2] 队员姓名 4[3] 队员姓名 5[3]

Abstract

很短的项目摘要

1. 主题

项目的主题和需求来自朱汶宣同学安装 Jupiter Notebook 的经历，由于一些依赖和 WSL 的问题，他希望能够有一款轻量级的产品能够帮助他定制化地精细分析运行命令过程中的问题。

随后我们将项目主题定为了 LLM 集成的轻量级终端。

1.1. 需求分析

为了帮助用户解决在命令行下的问题，首先应当监控命令行的输出，然后在接到用户指令之后将这些信息处理并调用 LLM 获得建议，最后将建议呈现在用户界面上，供用户选择采用。

我们决定将开发目标平台定在 windows 上，因为小组几位同学使用的都是 windows 电脑，并且 windows 原生的命令行生态并不成熟，相对于 Linux 和 Mac 缺少统一的管理器，遇到的环境和配置问题可能更多。

1.2. 技术选型

为了能够在短时间内进行开发，我们小组决定使用 python 进行开发，优先采用已有的开源库来提高开发效率。

在查阅资料之后，我们决定使用 pywinpty 库来实现 windows 下和终端的交互，这个封装了系统调用，维护了一个伪终端（Pseudo Terminal）对象，使得程序可以和运行当中的命令行进行交互。

对于和 LLM 交互的部分，我们决定使用较为成熟的 openai 库来实现和远程 API 的交互。

由于实现一个 GUI 过于复杂，需要实现的终端渲染内容过多（如虚拟控制字符等），我们决定实现一个 CLI，直接利用 windows 已经实现好的终端应用（如 windows terminal 等）。

1.2.1. 总体架构

项目总体包括如下模块：

- 模拟终端模块：封装和命令行的交互。
- CLI 模块：实现用户交互，维护命令行历史内容。
- 记忆模块：实现和 LLM 交互的短期和长期记忆。
- LLM 模块：封装和 LLM 的交互。
- agent 模块：总结和处理信息，将上下文、记忆和问题交给 LLM 模块。
- 部署模块：针对 git 仓库生成部署计划
- 安全模块：离线检测命令安全性。
- utils 模块：工具模块。

1.2.2. 模块简介

CLI 模块是整个程序的入口以及交互界面，其中维护了一个内建的模拟终端对象和 Agent 对象，CLI 负责捕获用户的输出，判断是否是询问指令，并且将输入交给模拟终端或 Agent，同时异步地监听终端的输出。

模拟终端模块实现了一个模拟终端类，在设定启动命令后异步地读取内建终端的输出，将输出异步地将输出回传给 CLI。

LLM 模块，封装了获取 API Key 已经和远端 API 的调用。

记忆模块封装了读取中期和长期记忆的过程，以及总结生成新记忆的 prompt 工程。

agent 模块封装了和记忆以及 LLM 的交互，并通过 prompt 工程实现格式化回复。

utils 模块实现了一些独立的工具和功能，如行内刷新输出等。

1.3. 实现细节

1.3.1. CLI 实现细节

1.3.2. LLM 接口处理

1.3.3. Prompt 工程

1.3.4. 安全检查

1.3.5. 历史记忆与用户画像

1.3.6. API 参数调优

1.4. 评估对比

1.5. 反思

参考文献

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), Proceedings of the 17th International Conference on Machine Learning (ICML 2000), pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

A. 附录

可以将一些额外的内容放在这里