

项目申请书

项目名称：基于langchain和ReAct实现智能shell命令执行

项目主导师：尹泽浩

申请人：史浩楠

邮箱：hxzsty233@gmail.com





-
- 1. 项目背景：LangChain
 - 1.LangChain简介
 - 2.shell语言
 - 3.ReAct Agent
 - 2. 项目详细需求
 - 1.自然语言处理
 - 2.支持中文输入
 - 3.强制人工确认
 - 3. 技术方法及可行性
 - 1.Python 相关
 - 2.LLM 相关
 - 1.api接口使用
 - 2.大模型生成技术
 - 3.LangChain相关
 - 4. 规划
 - 1.项目开发第一阶段（7.01 - 8.15）
 - 2.项目开发第二阶段（8.15 - 9.30）
-

1. 项目背景：LangChain

1.LangChain简介

LangChain 是一款流行的大语言模型应用框架，可以整合多家大模型的 `api` 接口，搭建大模型应用。
LangChain 拥有 LangSmith, LangServer等生态系统，LangSmith 是开发平台，提供了 Debug 等功能，同时使项目开发更好的可视化；LangServer 可以为搭建的应用构建外部 `api` 接口，开箱即用。

2.shell语言

Shell 是一个命令解释器，它通过接受用户输入的 `Shell` 命令来启动、暂停、停止程序的运行或对计算机进行控制。早期计算机没有图形界面，用户通过在命令行输入命令来操作计算机，这些命令发展成为了  `Shell` 语言。现在的 shell 根据操纵系统的不同，拥有不同的语法，类 Unix 系统（如  Linux  MacOS）中支持 `bash`，而  Windows 系统中则是 `powershell`。

3.ReAct Agent

ReAct 是 **LangChain** 社区中预训练的人工智能助手，可以为用户提供更便捷、更智能的问答体验。

2. 项目详细需求

1.自然语言处理

项目要求：支持自然语言输入，用户可以使用自然语言描述要执行的 `Shell` 命令，系统将自动解析用户的意图生成并执行相应的 `Shell` 命令

根据操作系统不同，同一操作具有不同的 `shell`命令形式，如类Unix系统中的 `which`和Windows系统中的 `where`，一个简短的对照可以参见 [powershell 和 bash 对照表](#)

就算是同一操作系统环境(如 Windows)，也有 `cmd`和 `powershell`两个命令行工具，分别有不同的`shell`命令，因此首先需要询问用户使用的操作系统环境和命令行工具

目前LangChain的官网只给出了类 Unix 系统的`shell`命令工具，Windows 系统的 `shell` 自动化工具仍需开发

2.支持中文输入

由于目前流行的大语言模型普遍支持多语言输入，一些本地大模型可能需要增加语言插件。不过，国外的许多大模型，如 openai, claude 等，普遍缺乏中文语料，模型输出质量不高，可以考虑更换为 kimi, 文心一言等国内大模型的 `api` 接口，更好的支持中文。


3.强制人工确认

大模型生成具有一定的不可控性，如果生成如`Remove-Item -Recurse -Force`或者`rm -rf`这样的破坏性命令，可能对用户系统造成巨大损失。为了避免工具执行的风险，可以调用 [HumanApprovalCallbackHandler](#)等回调工具进行人工确认

考虑到部分应用程序需要管理员/root权限，可以在询问时加上授权选项，如以管理员权限运行等

3. 技术方法及可行性

1.Python 相关

使用  [VSCode](#) 中的 `jupyter notebook` 插件，`pip` 包管理工具搭建项目开发环境，加载 `langchain_community` 等开发包

2.LLM 相关

1.api接口使用

如使用 `openai` 接口实现简单的问答：

```
import os
import openai

openai.api_key = "sk-fjz7IXZBmRaQfguU365483042a4949C79347D70cDaA94149"

openai.base_url = "https://api.gpt.ge/v1/"
openai.default_headers = {"x-foo": "true"}

completion = openai.chat.completions.create(
    model="gpt-3.5-turbo",
    messages=[
        {
            "role": "user",
            "content": "Hello world!",
        },
    ],
)
print(completion.choices[0].message.content)
```

2.大模型生成技术

使用检索增强生成(RAG)提升模型智能 [RAG详解 — AWS](#)

3.LangChain相关

根据 `LangChain`框架(`ReAct`模型)搭建人工智能助手，详见：

4. 规划

暑假有小学期，但是每天课不多，可以保证每天4小时以上的开发时间

1.项目开发第一阶段（7.01 - 8.15）

- ☐ 完成项目框架搭建
- ☐ 基本完成项目需求

2.项目开发第二阶段（8.15 - 9.30）

- ☐ 编写项目测试，仓库 pr