基于Bench数据集的大模型测评

1. Github的使用

GitHub 是一个基于 Git 的代码托管平台，广泛用于软件开发和版本控制。

1.创建一个 GitHub 账号

访问 [GitHub 网站](https://github.com)。

点击页面右上角的 "Sign up"。

按照提示输入电子邮件、创建用户名和密码，然后完成注册过程。

2. 创建一个新的仓库

登录到你的 GitHub 账户。

点击页面右上角的 "+" 图标，选择 "New repository"。

填写仓库名称（Repository name），并选择是否公开（Public）或私有（Private）。

你可以选择初始化仓库时添加 README 文件和 .gitignore 文件。

点击 "Create repository" 完成创建。

3. 克隆仓库到本地

在你的仓库页面，点击 "Code" 按钮。

复制 "HTTPS" 或 "SSH" 链接。

打开终端或命令提示符，输入以下命令：

git clone <repository-url>

将 `<repository-url>` 替换为你复制的链接。

4. 在本地进行更改

进入克隆的仓库目录：

cd <repository-name>

创建或修改文件，保存你的更改。

5. 提交更改并推送到 GitHub

将更改添加到暂存区：

git add .

提交更改并添加提交信息：

git commit -m "你的提交信息"

将更改推送到远程仓库：

git push origin main

注意：`main` 是默认的主分支名称，可能也叫 `master`。

6. 拉取远程仓库的更新

如果其他人对仓库进行了更改，你可以使用以下命令更新你的本地仓库：

git pull origin main

接下来是针对本项目进行的步骤讲解

1. 终端的处理
2. 打开终端或命令提示符（win+R cmd）输入以下指令克隆GAOKAO-Bench仓库到本地

git clone <https://github.com/OpenLMLab/GAOKAO-Bench.git>

2. 安装项目依赖项（首先确保您已经安装了Python以及Pip）

项目的依赖项通常列在 requirements.txt 文件中，可以使用 pip 安装这些依赖项。然而该项目没有requirements.txt 文件，需要手动安装项目可能所需的依赖项。

①跳转到该目录

cd GAOKAO-Bench

②使用 pip 包管理器安装 Python 的第三方库

pip install numpy pandas scikit-learn

③检查这些库是否成功安装

pip list

3.配置Linux终端

由于源代码中有mkdir命令，所以我们需要使用Linux终端来执行脚本

打开命令提示符，输入

wsl --install

这个命令会启用必要的wsl(windows subsystem for linux)功能，并安装默认的Ubuntu发行版

安装完成后，创建用户名并输入密码，之后只需打开终端或命令提示符输入wsl 并回车即可打开

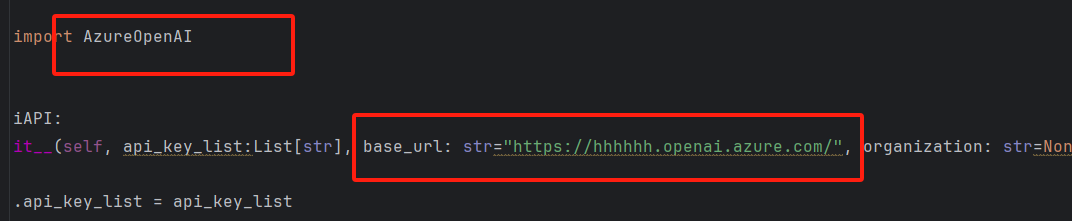
1. 调用openai及部分代码的修改
2. 使用以下命令安装 OpenAI Python 客户端库

pip install openai

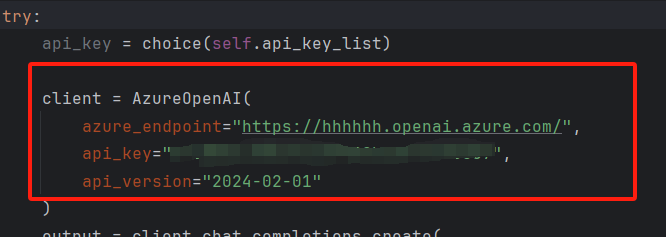
1. 转到Azure AI Studio中的资源和密钥，检索api key以及endpoint



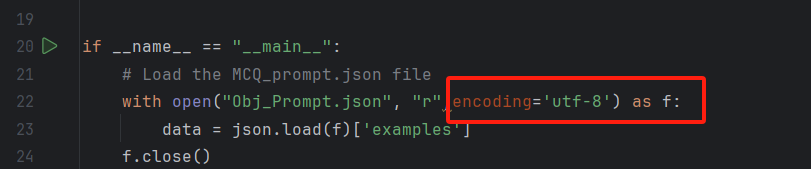
3.在pycharm中打开GaoKao-Bench项目，在openai\_gpt4.py中更改引用包的函数为AzureOpenAI、更改base\_url



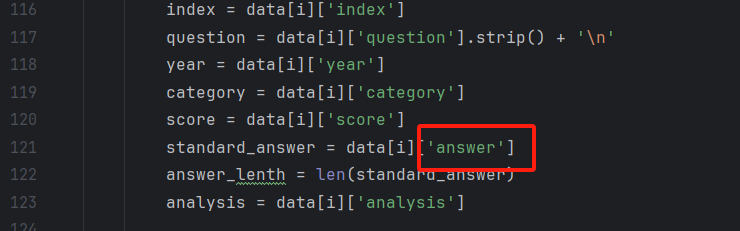
4.加入endpoint以及api key这两个参数



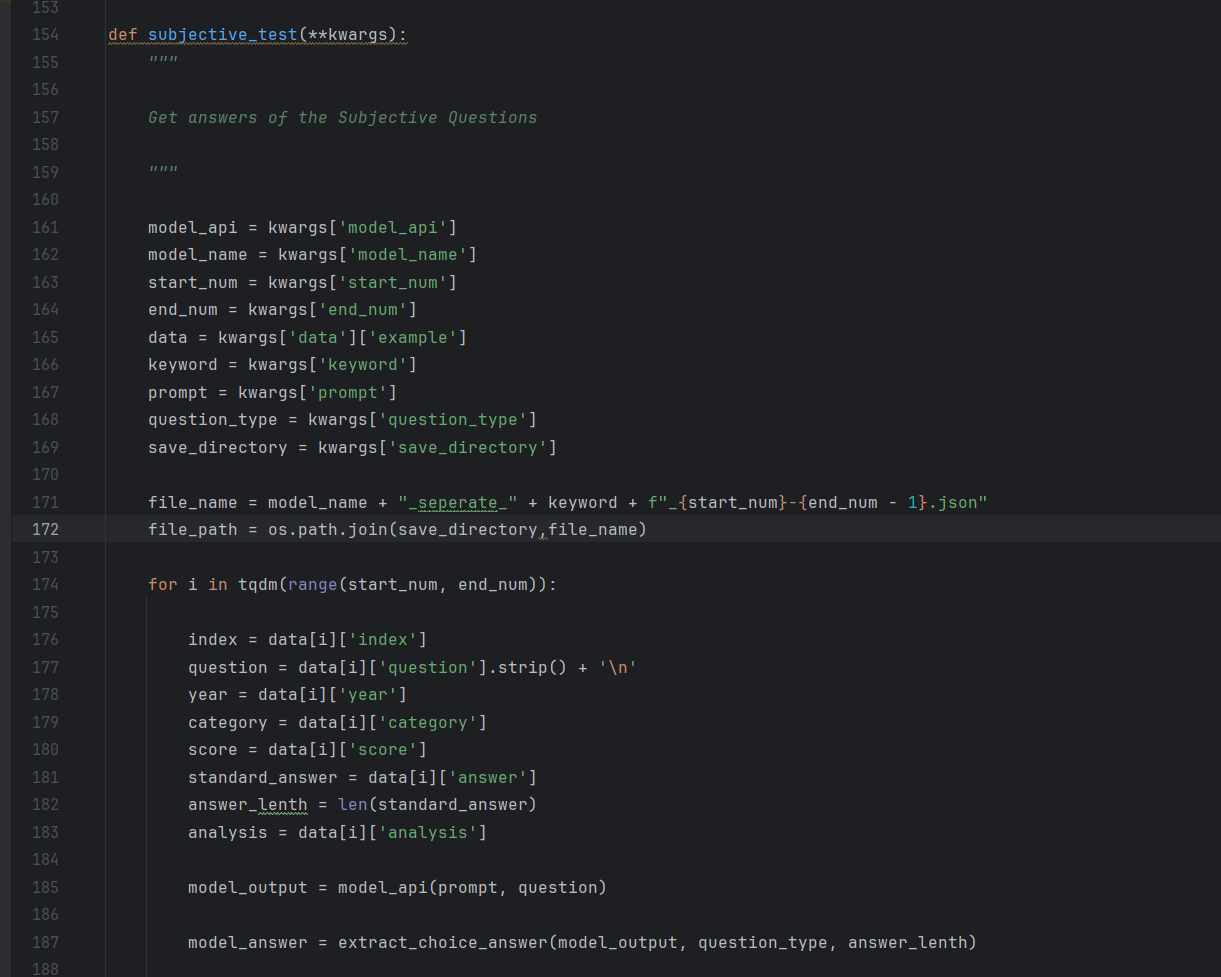
5.在objective\_bench.py中将字符编码改为utf-8，确保能成功编码



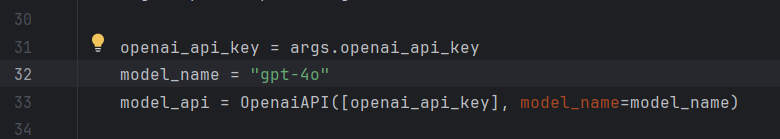
1. 在bench\_function.py中修改原字段的错误（将standard\_answer改为answer），使得可以读取answer的json格式



7.为避免因为一个卡顿而无法获得输出的情况，在bench\_function.py中修改写入文件的方式为每循环一次写一次，（下图所示为修改后的字段）



1. 在objective\_bench.py中写入需要测试的LLM（可以任意选择Azure AI Studio中的基本模型），以gpt-4o为例



四、结果的输出和处理

1.代码修改完成后，在Linux终端输入指令，成功运行

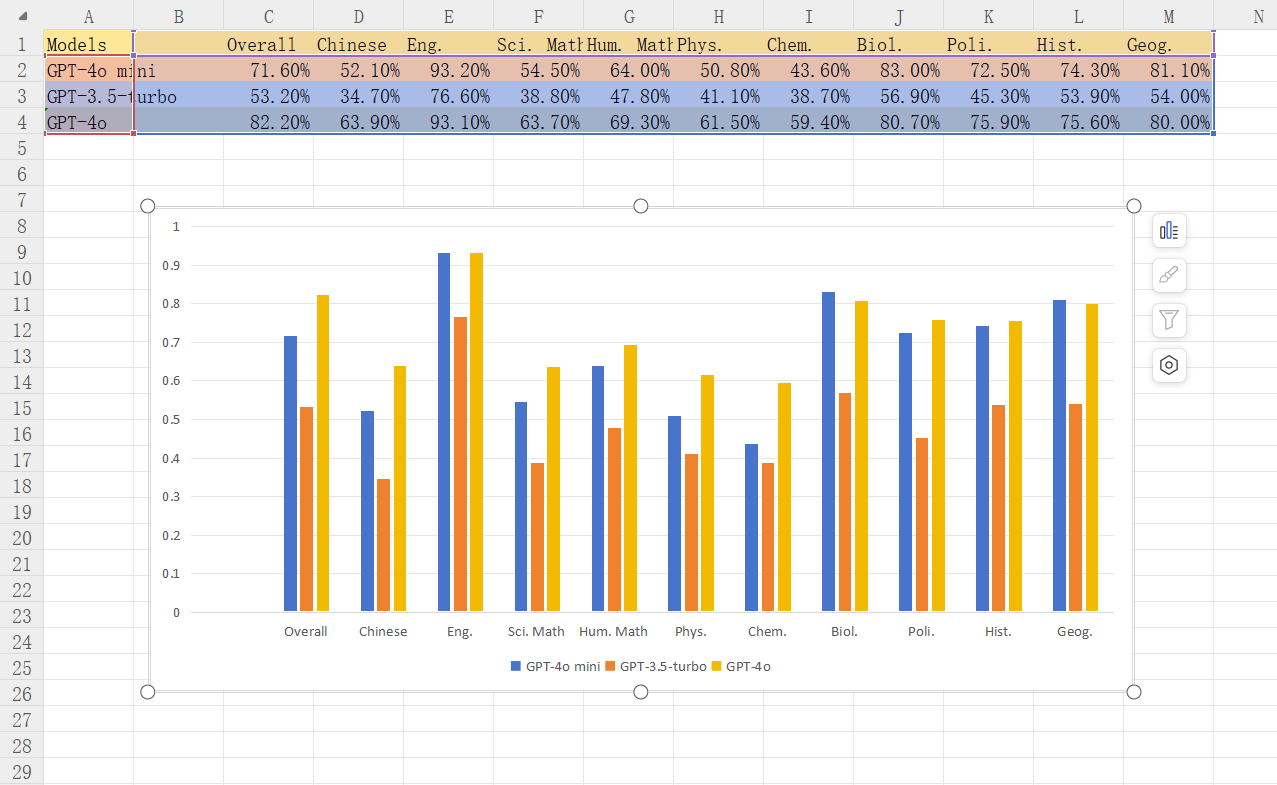
cd /mnt/c/Users/川普/Desktop/GAOKAO-Bench-main/Bench/

python3 objective\_bench.py --openai\_api\_key <your api key>

2.获得输出

3.之后更换模型做相同的题库，对比gpt-4o-mini与gpt-35-turbo的做题能力

由于主观题得分标准问题，本项目只对客观题进行了测试



数据分析

1. 总体表现(Overall)：

GPT-4o 在总体表现中表现最好，得分为82.2%。

GPT-4o mini 的表现次之，得分为71.6%。

GPT-3.5-turbo 的表现最差，得分为53.2%。

2. 学科表现：

语文(Chinese)：GPT-4o 依然表现最好，得分为63.9%，而 GPT-3.5-turbo 仅为34.7%。

英语(English)：三种模型的表现都相对较好，尤其是 GPT-4o mini 和 GPT-4o 均为93.2% 和 93.1%。GPT-3.5-turbo 稍逊，但也达到76.6%。

理科数学 (Sci.Math)：在科学和数学方面，GPT-4o 的表现继续领先，得分分别为63.7% 和 69.3%。GPT-4o mini 次之，尤其是在科学上表现为54.5%，但在数学方面略低，只有64.0%。

文科数学 (Humanities Math)：GPT-4o 的表现最佳，得分为69.3%，而 GPT-3.5-turbo 表现最差，为47.8%。

物理(Physics)：GPT-4o mini 表现较差，得分为50.8%，而 GPT-4o 在物理学科中领先，得分为61.5%。

化学(Chemistry)：GPT-4o mini 和 GPT-4o 的得分分别为43.6% 和 59.4%，而 GPT-3.5-turbo 的得分最低，仅为38.7%。

生物(Biology)：所有模型在生物学科的表现都较好，特别是 GPT-4o mini 和 GPT-4o，分别得分为83.0% 和 80.7%。

政治(Politics)：GPT-4o 在政治学科中的表现最佳，得分为75.9%。GPT-4o mini 和 GPT-3.5-turbo 的得分分别为72.5% 和 45.3%。

历史(History)：GPT-4o 和 GPT-4o mini 的得分较为接近，分别为75.6% 和 74.3%，但 GPT-3.5-turbo 仅得53.9%。

地理(Geography)：三个模型的地理学科表现都相对较好，特别是 GPT-4o mini 和 GPT-4o 的得分分别为81.1% 和 80.0%。

总结：

GPT-4o 在大部分学科中的表现明显优于另外两个模型，尤其是在整体表现、科学、数学和人文学科上。

GPT-4o mini 在整体表现上也相当不错，尤其是在生物和地理学科中表现突出。

GPT-3.5-turbo 的表现最弱，特别是在中文、科学和物理等学科中。