

从 0 开始测评大语言模型 (LLM)

——谭亲怡

对于小白来说,对于“如何让大语言模型做题”脑海中呈现出的便是用户自己一题一题地输入给 LLM;然而,对于程序员来说,该如何让它自动地读取题库、进而测评呢?本文将通过一个项目来具体介绍。

带着测评 LLMs 的目的进行互联网搜索,本人在 GitHub 上发现一个以中国高考题目为数据集,能够测评大模型语言理解能力、逻辑推理能力的测评框架。

问题来了, GitHub 是什么?

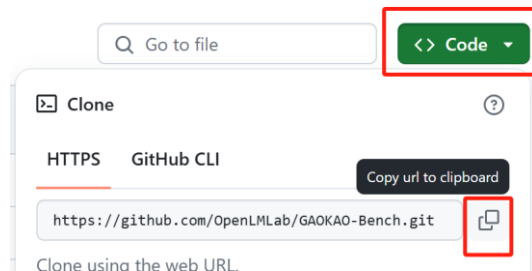
GitHub 是一个只支持 Git 作为唯一的版本库格式进行托管、面向开源及私有软件项目的托管平台。GitHub 上有大量的开源项目,开发者可以自由地浏览、使用和贡献这些项目。

那么对于一位小白来说,看见一个好玩的开源项目,该怎么办呢?以运行 GAOKAO-Bench 项目为例 (<https://github.com/OpenLMLab/GAOKAO-Bench>):

在源码区的下面,会自动显示源码中的 README.md 文件,这是项目的说明文件,我们可以通过简单示例一步一步将项目运行下去。然而在运行项目之前,有几步步骤是必备的。

1. 克隆项目仓库 (首先确保您已经安装了 Git)

① 点击“Code”获取项目的托管地址,通过这个地址可以远程拉取到这个仓库



② 打开终端 (macOS 或 Linux) 或命令提示符 (Windows) 执行以下命令克隆 GAOKAO-Bench 仓库

```
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
```

接下来可以正式地按示例操作:

在观察完代码后,本人决定用 Visual Studio code (vscode) 对代码进行润色与修改 (确保已经安装 python、配置基本环境)。由于代码中含有 mkdir 命令,我们还需要配置 Linux 终端

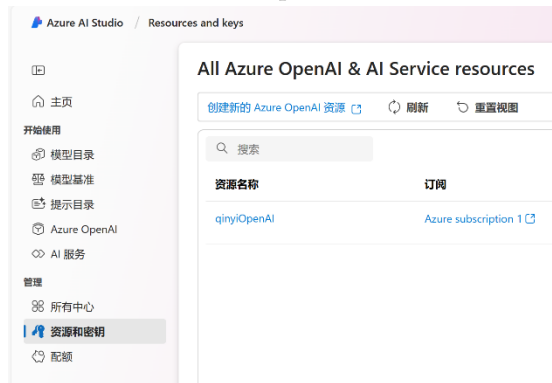
而非 windows 终端，确保能执行命令。

那么如何将各种 LLMs、Azure OpenAI 服务，和 GaoKao-Bench 配合使用？

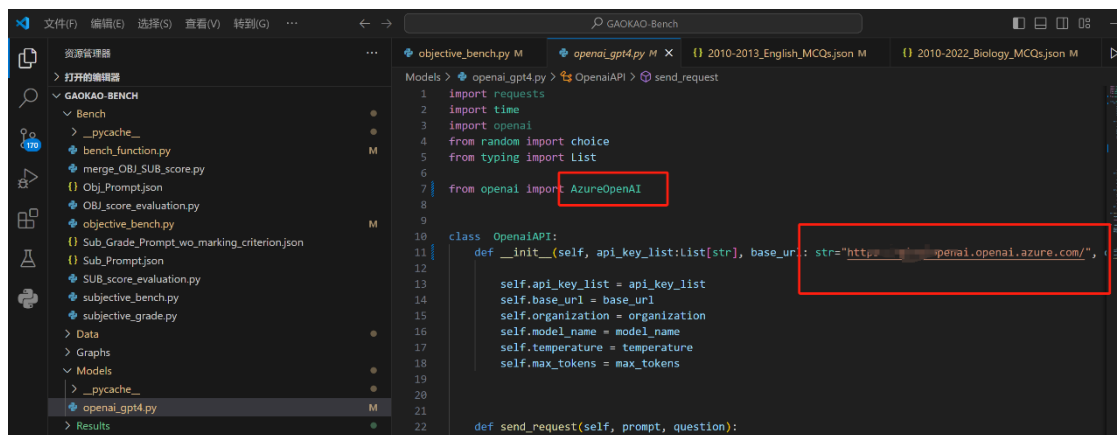
①我们需要使用以下命令安装 OpenAI Python 客户端库

```
pip install openai
```

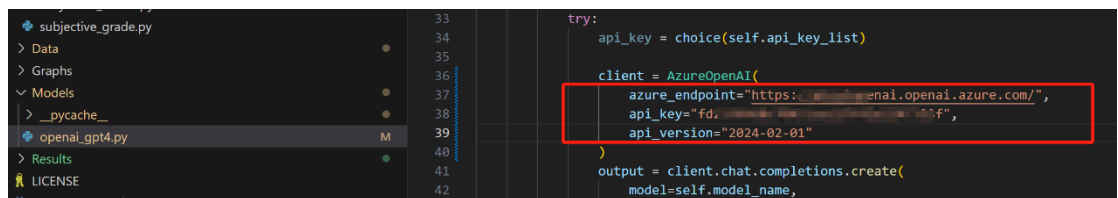
②转到 Azure AI Studio 中的资源和密钥，检索 api key 以及 endpoint（两个必要参数），使得后续能成功调用 Azure OpenAI



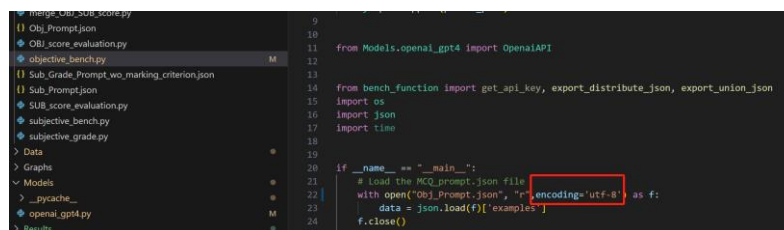
③在 vscode 中打开 GaoKao-Bench 项目，在 openai_gpt4.py 中更改引用包的函数为 AzureOpenAI、更改 base_url



④加入 endpoint 以及 api key 这两个参数

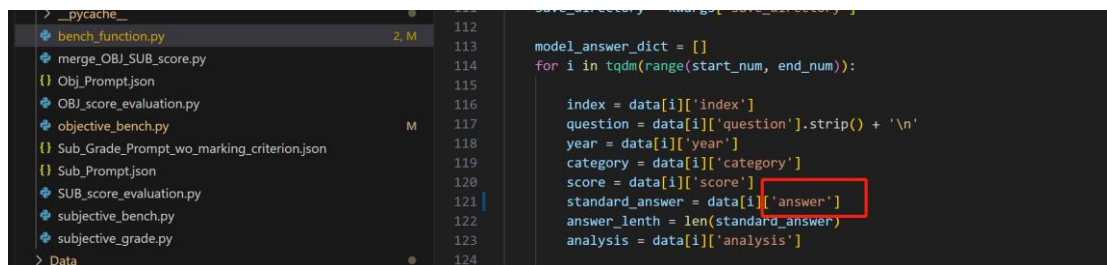


⑤在 objective_bench.py 中将字符编码改为 utf-8，确保能成功编码

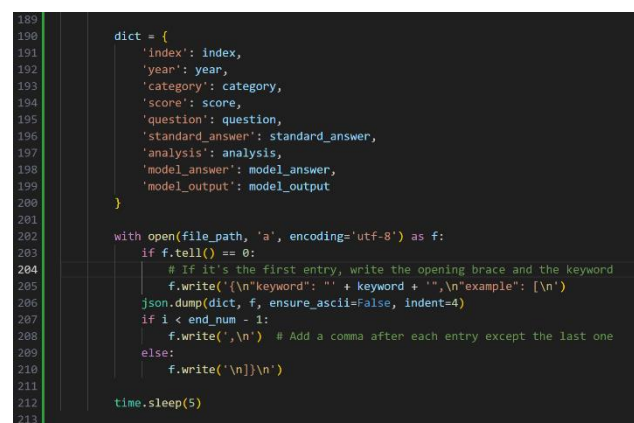
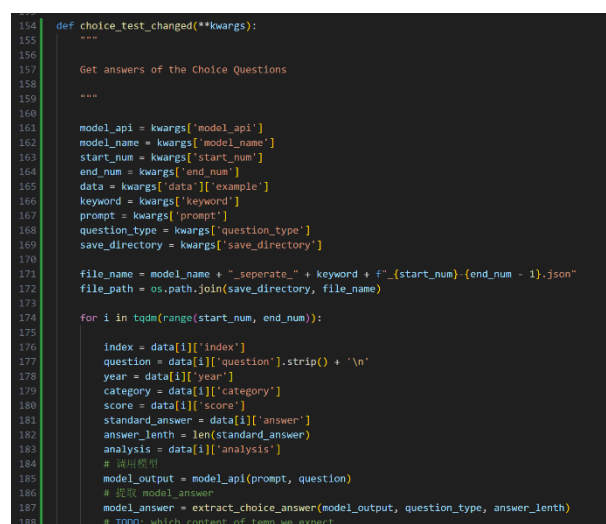


⑥ 在

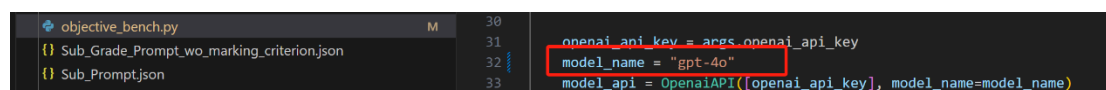
bench_function.py 中修改原字段的错误(将 standard_answer 改为 answer),使得可以读取 answer 的 json 格式




⑦在 bench_function.py 中修改写入文件的方式为每循环一次写一次,使得程序员可以同步做题情况,代替了原代码写完所有的题再输出的形式,避免了因为一个卡顿而无法获得输出的情况(下图所示为修改后的字段)



⑧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



⑨修改完成, 在 Linux 终端输入指令, 成功运行



⑩获得输出

gpt-4o_2010-2013_English_MCQs	2024/8/9 13:48	文件夹
gpt-4o_2010-2022_Biology_MCQs	2024/8/9 13:07	文件夹
gpt-4o_2010-2022_Chemistry_MCQs	2024/8/9 13:38	文件夹
gpt-4o_2010-2022_History_MCQs	2024/8/9 13:05	文件夹
gpt-4o_2010-2022_Math_I_MCQs	2024/8/9 12:54	文件夹
gpt-4o_2010-2022_Math_II_MCQs	2024/8/9 12:44	文件夹
gpt-4o_2010-2022_Physics_MCQs	2024/8/9 13:27	文件夹
gpt-4o_2010-2022_Political_Science_MCQs	2024/8/9 13:17	文件夹

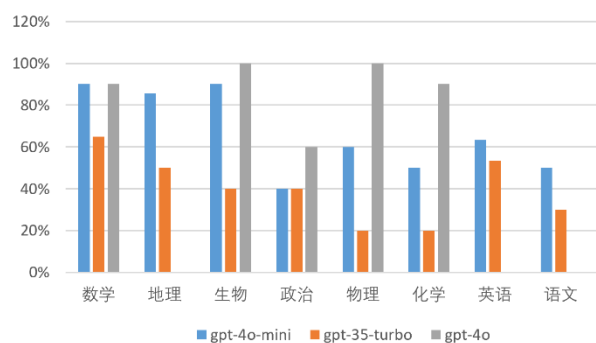
接下来本人更换了大语言模型做相同的题库(已将原题库缩小),对比 gpt-4o-mini 与 gpt-35-turbo 的做题能力 (gpt-4o 也用作测试, 然而数据不全不作详细对比)。

*修改后的数据集为数学 (单选 20 题)、生物 (单选 10 题)、政治 (单选 10 题)、物理 (多选 10 题)、化学 (单选 10 题)、英语 (单选 10 题+七选五 50 题)、语文 (单选 10 题)、地理 (单选 14 题), 以及历史 (单选 2 题)

经人工统计后, 获得不同大语言模型做不同学科的题目正确率为:

	数 学	地 理	生 物	政 治	物 理	化 学	英 语	语 文	历 史
gpt-4o-mini	90%	85.71%	90%	40%	60%	50%	63.33%	50.00%	50%
gpt-35-turbo	65%	50%	40%	40%	20%	20%	53.33%	30%	100%
gpt-4o	90%	/	100%	60%	100%	90%	/	/	50%
不同大语言模型做不同学科的题目正确率									

不同大语言模型做不同学科的题目正确率



1. 总体趋势

· gpt-4o-mini 在所有学科的表现上均优于 gpt-35-turbo, 但与自己相比, 在个别学科 (如政治和化学) 中表现也有短板

· gpt-35-turbo 表现较为均衡, 但正确率普遍较低, 尤其在某些学科上正确率特别低 (物理和化学)

2. 学科比较

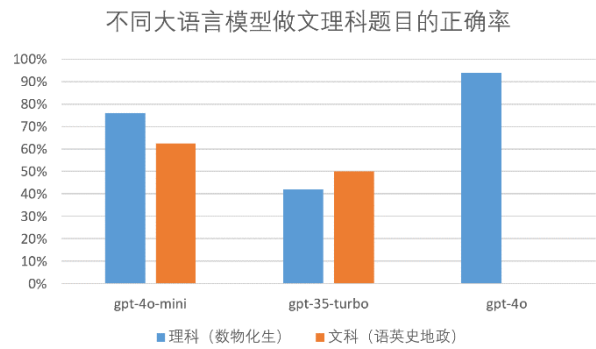
· 数学、地理、生物、英语: 这四个学科中, gpt-4o-mini 的正确率都高于 gpt-35-turbo。特别是在数学和地理学科上, 它的正确率超过 85%, 而 gpt-35-turbo 的正确率只有约 60%

· 物理、化学: 这两个学科中, 两个模型都表现不佳, 而 gpt-35-turbo 的表现最弱 (两个学科都只有 20%)

· 政治、语文: 这两个学科两个模型表现都不太理想, 然而 gpt-4o-mini 的表现仍然优于 gpt-35-turbo

将学科分类为文理科后，获得数据

	理科（数物化生）	文科（语英史地政）
gpt-4o-mini	76%	62.50%
gpt-35-turbo	42%	50%
gpt-4o	94%	/
不同大语言模型做文理科题目的正确率		



1. gpt-4o-mini
 - 理科：正确率 76%，表现相对较好，具有较高正确性
 - 文科：正确率略低于理科，为 62.5%，但也相对稳定
2. gpt-35-turbo
 - 理科与文科的表现都较差，正确率不高于 50%，但相比较而言文科正确率略高
3. gpt-4o
 - 理科：正确率高达 94%，是所有模型中在理科题目上表现最好的，非常突出
 - 文科：虽然没有得到文科数据，但从理科的表现来看，可以推测它在文科的表现上不会太差
4. 模型比较
 - gpt-4o-mini 在理科题目上的表现优于文科，且均优于 gpt-35-turbo
 - 总体来看，gpt-4o 在理科方面表现最佳，gpt-4o-mini 次之，最后是 gpt-35-turbo