# 一、项目概述

* 1. 项目名称

灵“墨”雅韵：AI古诗词创作助手

* 1. 项目负责人

队长：李虹宇 211310119

组员：刘俣哲211310218

马千瑜211310219

芮畅211380130

彭杰211310122

* 1. 项目背景

在过去，人们对古诗词的创作和欣赏需要深厚的文学功底和灵感。然而，随着AI技术的不断进步，AI生成古诗词的技术已经开始在多个领域展现出潜力。这项技术不仅可以用于增强文化传承和教育，还可以应用于创意产业、文学创作辅助、智能文化推广等方面。现在我们就在此背景下制作一个根据输入的意象关键词直接生成原创诗词的大模型。

* 1. 项目目标

应用xtuner的qlora方法，对大模型internlm-chat-7b进行微调，使其向古诗创作方向对齐。输入几个关键词即可生成符合意象的古诗。

* 1. 项目范围

数据库的导入，大模型的导入和微调，用户交互平台的构建。

* 1. 项目分工   
     ​李虹宇：项目统筹、模型微调、视频录制、数据整理

刘俣哲：PPT制作、项目报告书撰写、数据集搜集

马千瑜：前端布置、prompt增强、数据集搜集

芮畅：Github文档编写、数据集搜集

彭杰：openxlab上数据集的处理工作、数据集搜集

# 二、项目实施

## 具体过程

使用OpenXLab中的开源数据集进行微调，这里使用的是CCPM数据集(Chinese Classical Poetry Matching) 网址：<https://openxlab.org.cn/datasets/OpenDataLab/CCPM>。



数据集分为译文和古诗原文的四个选项，其中有一项是正确的古诗原文。具体格式如下图：



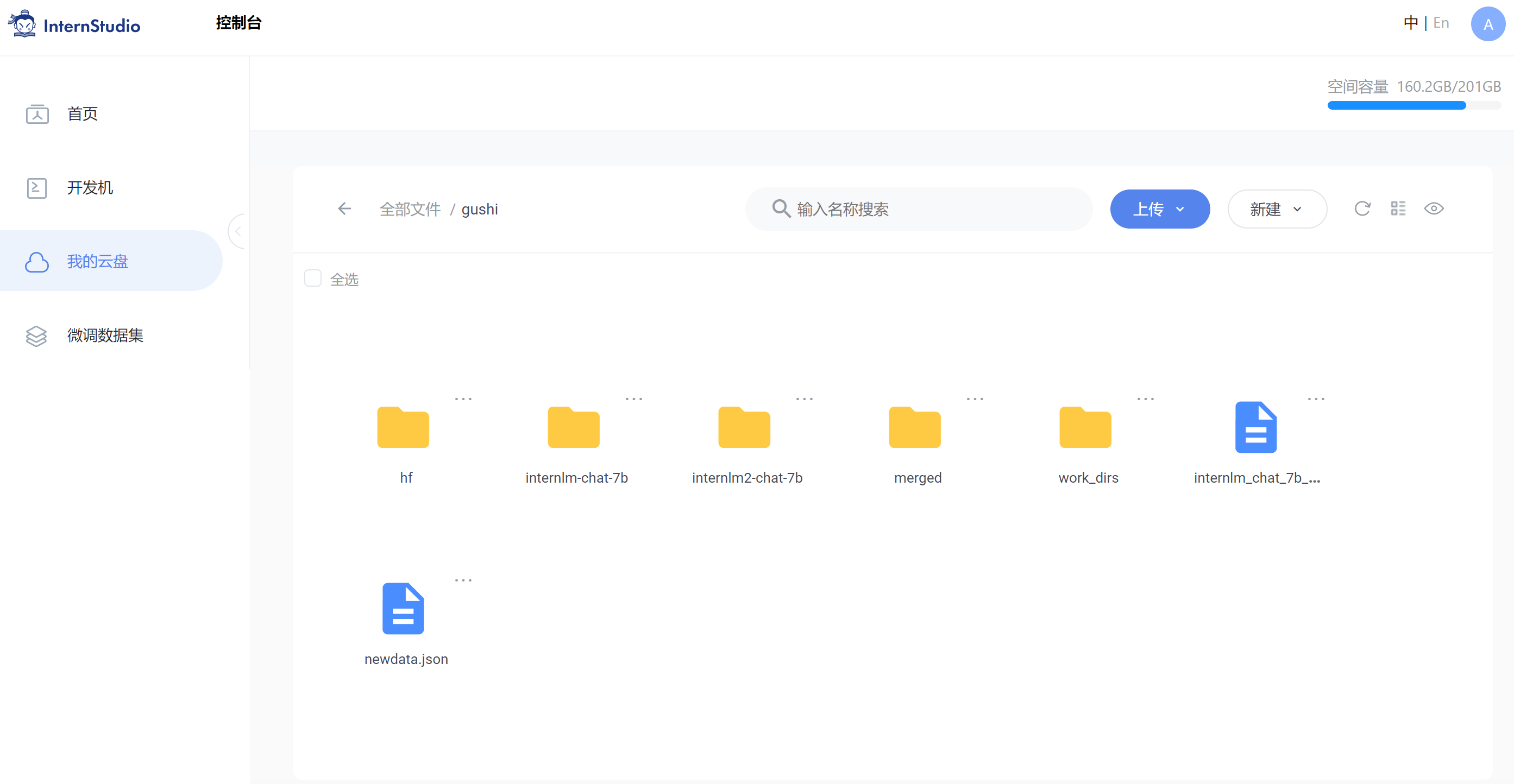


将数据集下载到本地，打开后可以发现，数据集内附带了正确的对应选项。遂编写脚本删除错误选项，对原数据集进行格式处理。

处理后数据：

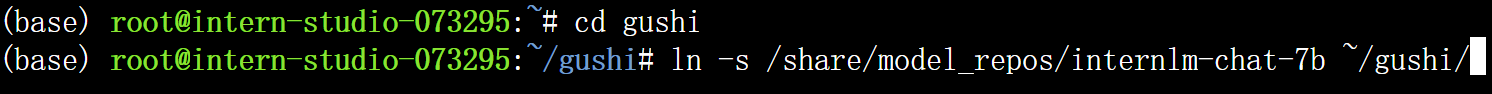


打开云盘，将数据集（newdata.json）上传至开发机：

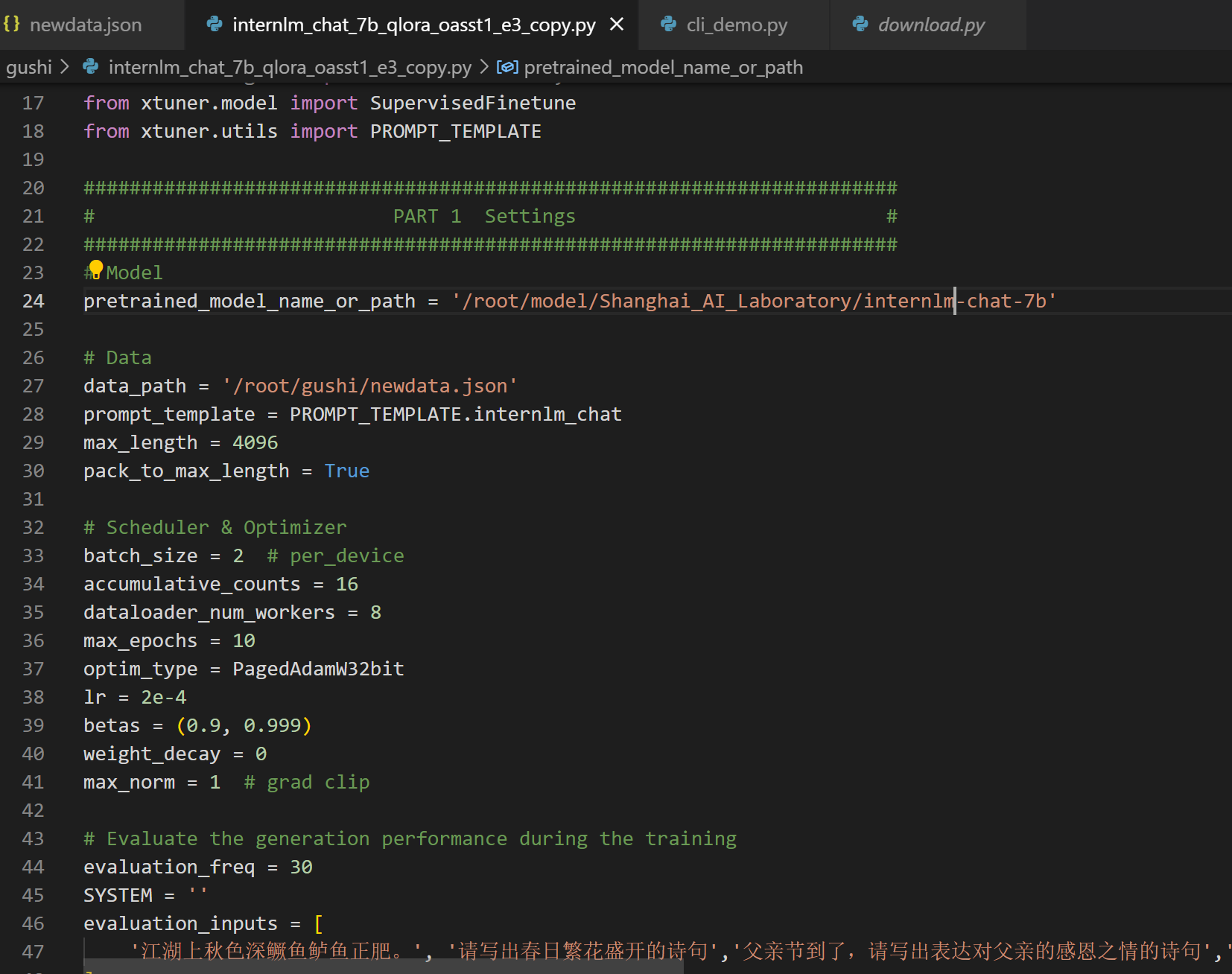


配置完基础环境（此处参考：https://github.com/InternLM/Tutorial），先在家目录下创建gushi目录，并将基座模型软链挂载过来。

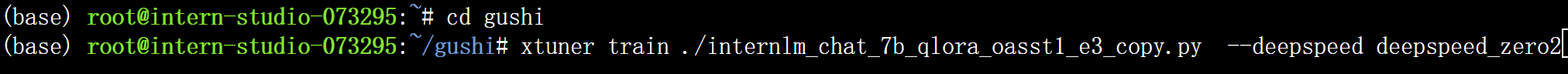




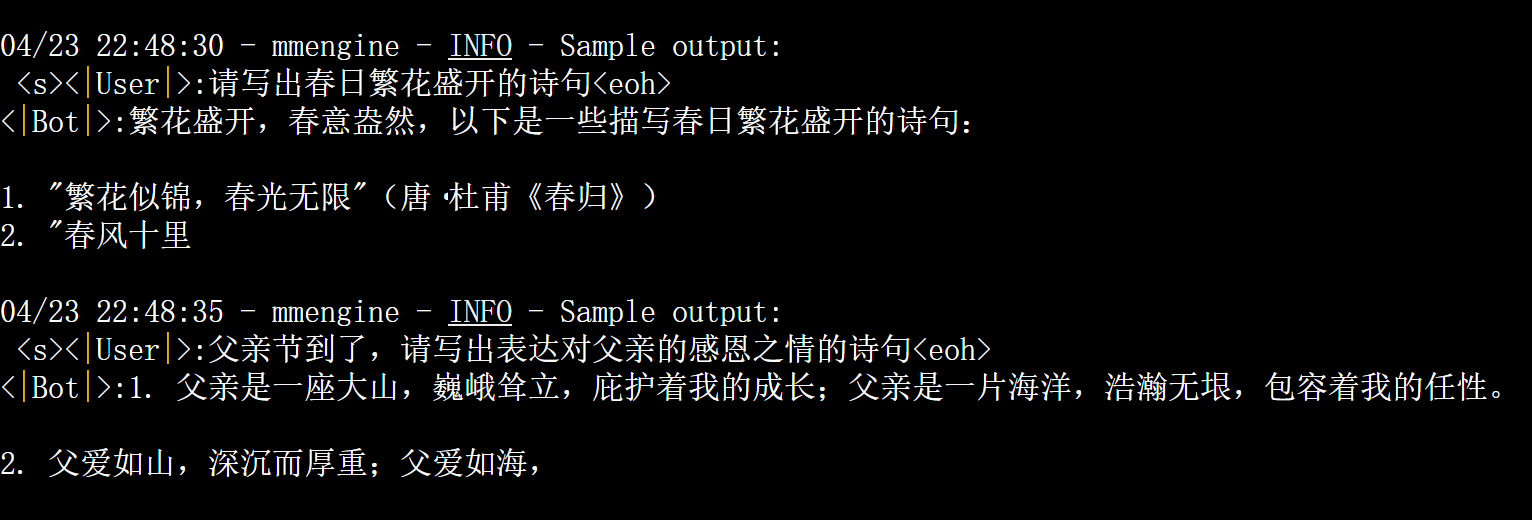
将internlm\_chat\_7b\_qlora\_oasst1\_e3\_copy.py脚本中的模型路径改为internlm-chat-7b所在的绝对路径，数据集路径改为/root/gushi/newdata.json。加大batch\_size、max\_length等参数，使得硬件效率被充分利用。



这里epoch被设置为10轮，以增强训练效果；将dataloader\_num\_workers设为8，增加预加载命中效率；evaluation\_freq参数设置为30，并添加evaluation\_inputs测试样例，从而在训练时充分观察训练效果。



输入指令xtuner train ./internlm\_chat\_7b\_qlora\_oasst1\_e3\_copy.py --deepspeed deepspeed\_zero2，开始训练：



可以看到，刚开始训练时，模型对问题的回答并不好，更接近现代白话文。

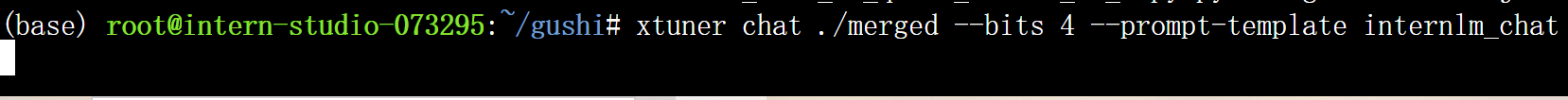
跑完训练后，输入指令xtuner convert pth\_to\_hf ./internlm\_chat\_7b\_qlora\_oasst1\_e3\_copy.py ./work\_dirs/internlm\_chat\_7b\_qlora\_oasst1\_e3\_copy/epoch\_10.pth ./hf，创建hf文件夹，将得到的 PTH 模型转换为 HuggingFace 模型：



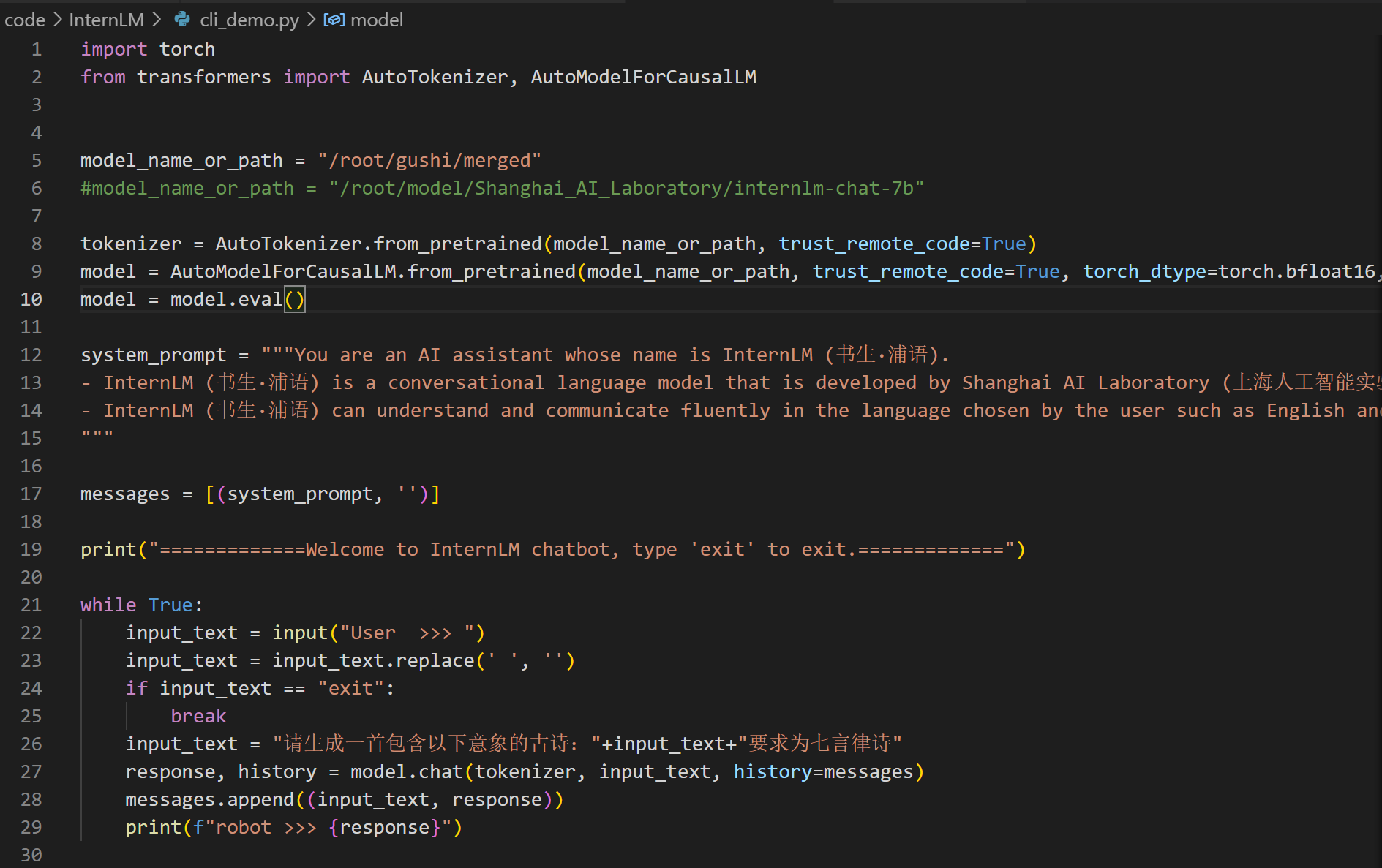
此时，hf 文件夹即为所谓 “qLoRA 模型文件”，即Adapter模型。

接下来输入xtuner convert merge ./internlm-chat-7b ./hf ./merged --max-shard-size 2GB指令，将Adapter与原大模型合并。得到merged文件夹。

这样模型就微调结束了，4比特量化加载模型：xtuner chat ./merged --bits 4 --prompt-template internlm\_chat



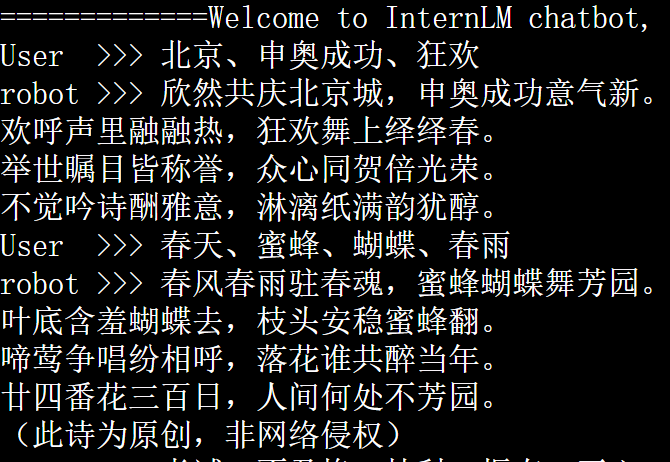
也可使用cli\_demo.py脚本加载模型对话，修改模型路径为/root/gushi/merged。



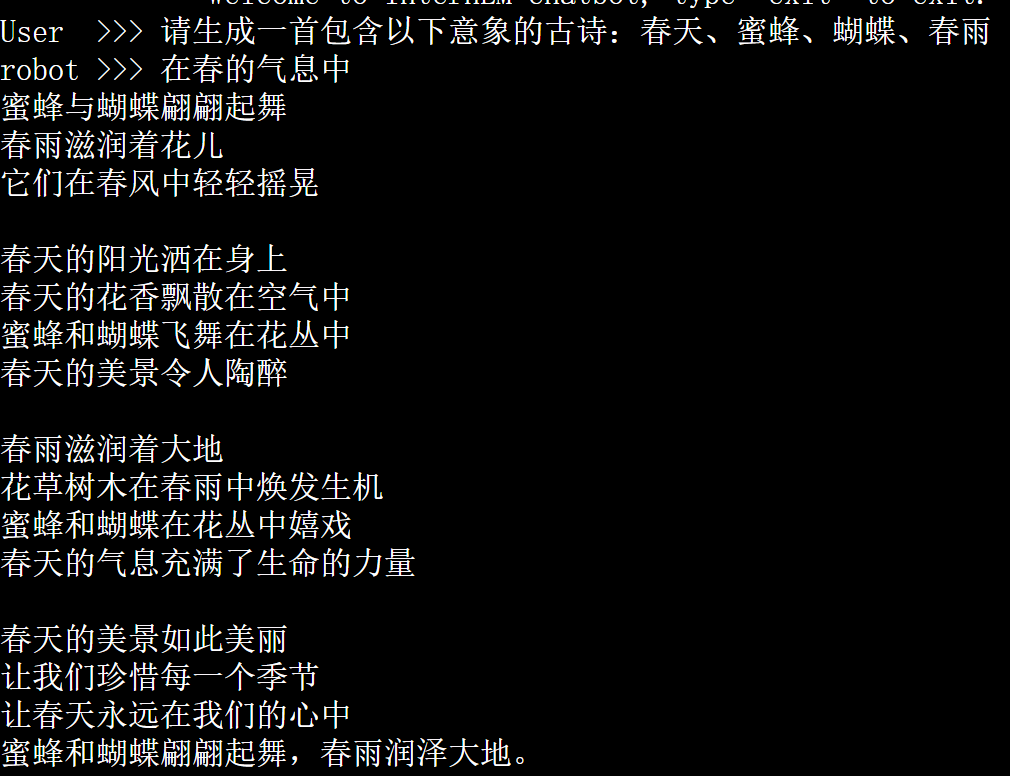
为了增强效果，增加对模型的提示词：



对话效果如下：



可以对比训练前的基座模型：



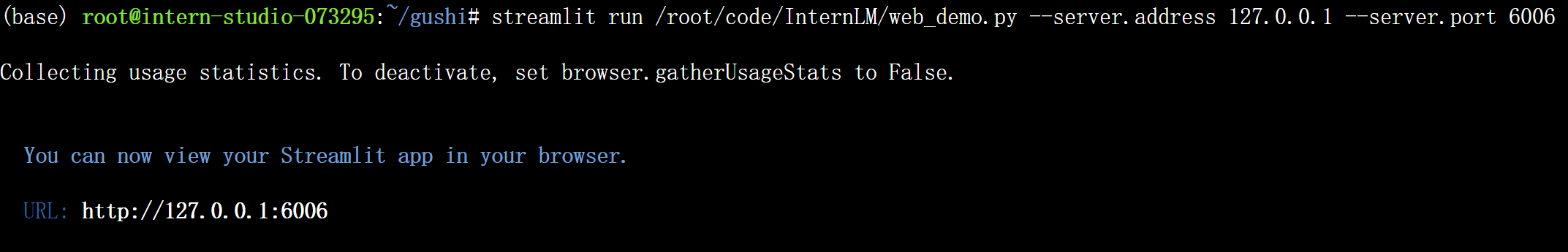
可以看出，相比微调前模型只会写现代诗歌，微调后模型学会了古诗的基本写法。

之后，运行网页demo，生成交互型页面。同样用提示词对模型进行增强：



ssh连接本地端口，输入命令启动web\_demo.py





效果如下：



# 三、项目总结

3.1 项目成功原因

1）组内成员的紧密合作，在遇到困难时相互帮助

2）组长与组员们一起制定合理的项目计划与风险措施

3）通过老师给予的文档资料与视频，我们提前熟悉了大模型技术的大概内容

3.2 项目的经验教训

1）在项目初期，一切确定之后最好尽早开始工作，为后续可能出现的难题留足解决时间。

2）在项目中期，应该主动确定好每个进度的截止时间，以防出现拖沓现象。

3）对核心重点项目和难点项目进行区分，先完成基础，再攻坚克难。

3.3 对未来的展望

1）后续在有条件的情况下，可以就生成诗词的质量进行进一步的提升。例如使用chatGPT生成更高质量的数据集进行训练，以及使用internlm2-chat-7b等新版本模型训练。

2）扩展此项目的应用场景，使得此项目能够在不同条件下都能产生一定作用。

3）使用高水平AI对微调后的模型进行主观评测，从而对模型的训练效果有更直观的了解。