比奇堡冲浪小组

作品报告

小组成员:高茂庭

# 目录

[（1）微调算法介绍，包含使用的微调数据集规模的预处理方式](#_微调算法介绍，包含使用的微调数据集规模的预处理方式)

[（2）超参配置介绍说明](#_超参配置介绍说明)

[（3）微调后的权重文件链接](#_微调后的权重文件链接)

[（4）运行环境说明](#_运行环境说明)

[（5）模型微调后原有能力评估得分；](#_模型微调后原有能力评估得分；)

## 微调算法介绍

**使用 LORA 进行微调**

在比赛中，我们使用了 LORA（Low-Rank Adaptation）技术对预训练模型进行微调。LORA 是一种高效的微调方法，通过在模型权重矩阵上添加低秩近似矩阵，实现对模型参数的有效更新，从而减少微调过程中对计算资源和存储空间的需求。

**使用 ALPACA 数据集**

我们选择了 ALPACA 数据集作为微调的基础数据。ALPACA 数据集以其高质量和多样性，广泛应用于自然语言处理任务的训练和评估。

**数据预处理**

为了使数据集适应我们的微调需求，我们将 ALPACA 数据集转换为 MindRecord 格式。以下是数据预处理的主要步骤：

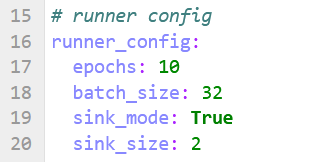
1. **读取 ALPACA 数据集**：参考MindFormers官网的数据前处理，使用以下指令进行下载

wget https://raw.githubusercontent.com/tatsu-lab/stanford\_alpaca/main/alpaca\_data.json

1. **将原始数据集转换为多轮对话格式**：执行alpaca\_converter.py(附件数据集处理数据夹中)，使用fastchat工具添加prompts模板，将原始数据集转换为多轮对话格式。
2. **数据预处理、Mindrecord数据生成**：执行llama\_preprocess.py(附件数据集处理资料夹中)，进行数据预处理、Mindrecord数据生成，将带有prompt模板的数据转换为mindrecord格式。

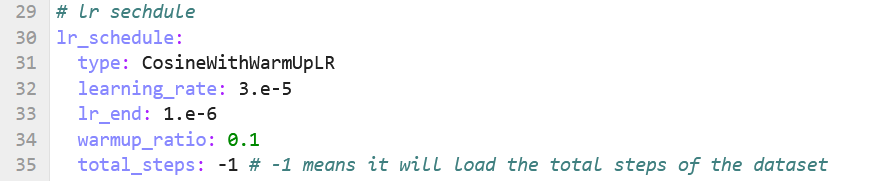
## 超参配置介绍说明

超参配置链接: https://bucket-8869.obs.cn-southwest-2.myhuaweicloud.com/config.yaml



增加训练周期，确保模型有足够时间学习数据

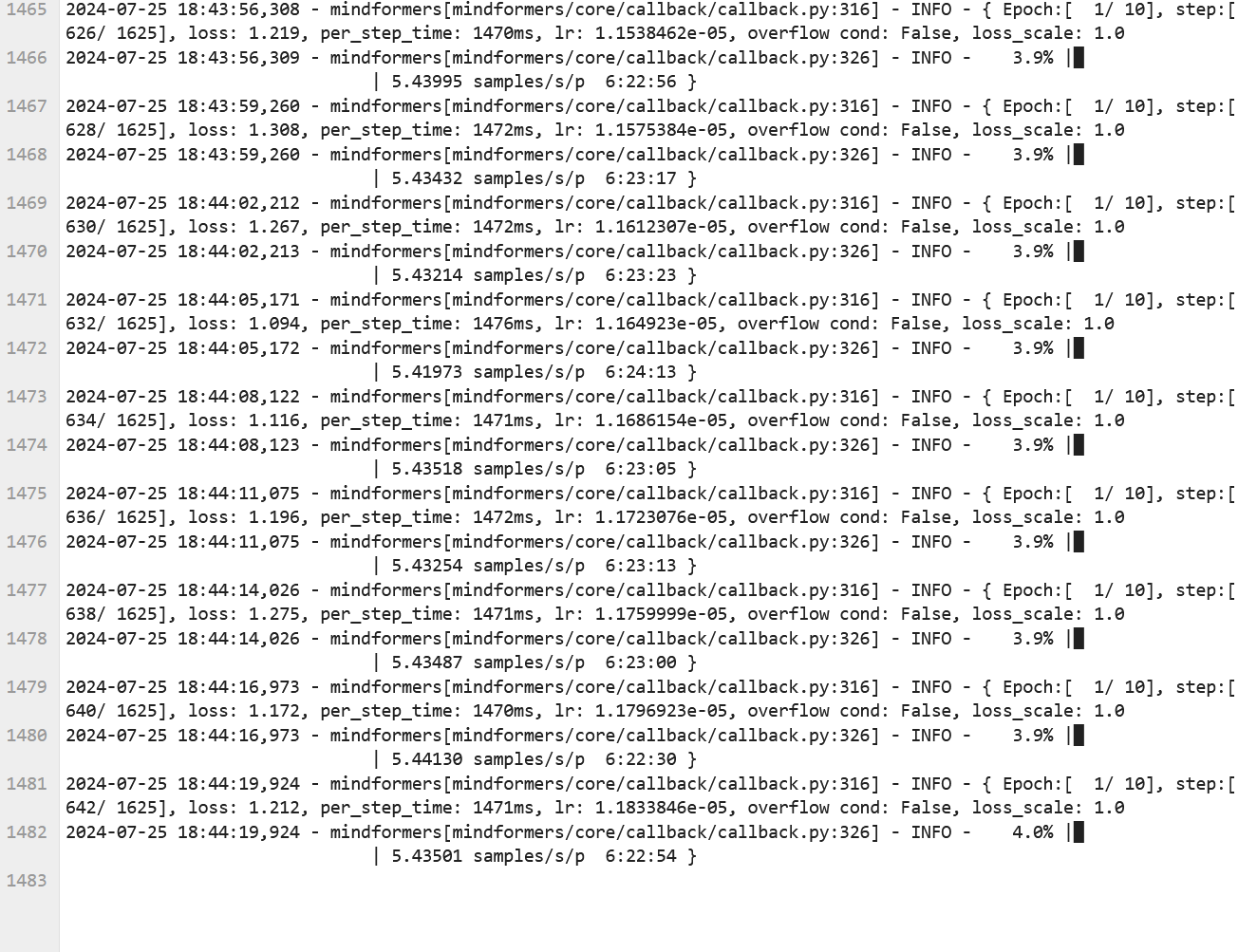
调整批次大小以平衡内存使用和训练速度



调整学习率以确保训练稳定性

增加warmup比例以确保平稳过渡

开始进行训练:



## 微调后的权重文件链接

## <https://bucket-8869.obs.cn-southwest-2.myhuaweicloud.com/best_llama3-8B.ckpt>

## 运行环境说明

Notebook 规格: Ascend: 4\*ascend-snt9b(32G)|ARM: 96核 768GB

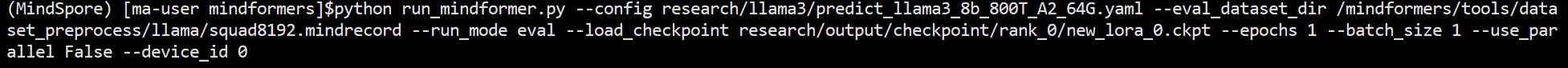
## 模型微调后原有能力评估得分；

首先是低参比例，见下图:

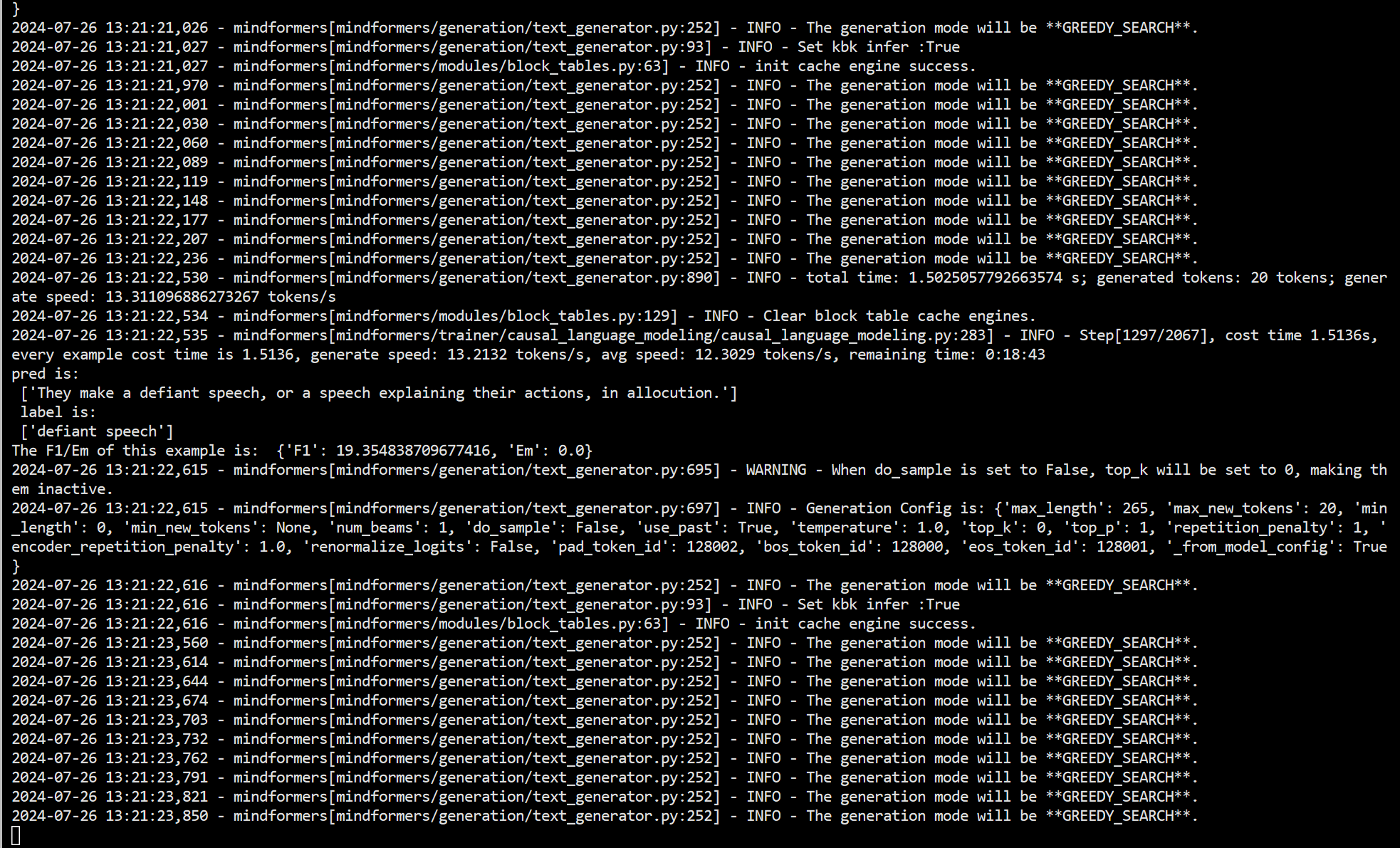


参数: 3407872

运行评估



经过漫长的测试等待之后:



得到:



**结果: F1 score: 71.01277924546918, Em score: 53.07208514755685**

**大幅超过原有要求了**

**其他代码在附件中**