Qwen3-Reranker-SFT

一个基于 Qwen3 重排器模型的监督微调（SFT）项目，用于提升文本检索和重排任务的性能。

一、项目简介

本项目实现了对 Qwen3-Reranker-0.6B 模型的监督微调，通过 LoRA（Low-Rank Adaptation）技术进行参数高效微调，用于改善查询-文档匹配的重排效果。

二、项目结构

```
Qwen3-Reranker-SFT/
├── README.md                # 项目说明文档
├── reranker_sft.sh          # 训练脚本
├── ds_stage0.json           # DeepSpeed 配置文件
├── test.py                  # 模型测试脚本
└── toy_dataset.jsonl        # 示例训练数据集
```

三、功能特点

- **高效微调**: 使用 LoRA 技术进行参数高效微调，减少计算资源需求
- **DeepSpeed 支持**: 集成 DeepSpeed 优化训练性能和内存使用
- **中文优化**: 针对中文查询-文档重排任务进行优化
- **完整流程**: 从数据准备到模型训练、测试的完整流程

四、环境配置

```
conda create -n flag_venv python=3.12 -y
conda init bash && source /root/.bashrc
conda activate flag_venv
conda install ipykernel
ipython kernel install --user --name=flag_venv

pip install -U FlagEmbedding[finetune]
```

五、数据格式

训练数据应为 JSONL 格式，每行包含一个样本：

```
{
    "query": "查询文本",
    "pos": ["正例文档1"],
    "neg": ["负例文档1", "负例文档2", "负例文档3"]
}
```

六、使用方法

1. 准备数据

将你的训练数据准备成 JSONL 格式，参考 `toy_dataset.jsonl` 的格式。

2. 配置训练参数

修改 `reranker_sft.sh` 中的参数：

```
# 数据路径
train_data="/root/autodl-tmp/Qwen3-Reranker-SFT/toy_dataset.jsonl"

# 模型路径
model_name_or_path="/root/autodl-tmp/models/Qwen3-Reranker-0.6B"

# 输出路径
output_dir="/root/autodl-tmp/models/finetuned-qwen3-reranker"
```

3. 开始训练

```
bash reranker_sft.sh
```

4. 测试模型

训练完成后，运行测试脚本：

```
python test.py
```

七、训练配置

LoRA 配置

- **Rank**: 8
- **Alpha**: 16
- **目标模块**: q_proj, k_proj, v_proj, o_proj

训练超参数

- **学习率**: 2e-4
- **训练轮数**: 3
- **批量大小**: 4
- **梯度累积步数**: 1
- **训练组大小**: 8
- **序列长度**: 512

DeepSpeed 配置

```
- 阶段：Stage 0
- ZeRO Stage 0：不进行任何分区，所有参数、梯度和优化器状态都保存在每个 GPU 上
- 适用场景：单卡训练或小规模模型训练，保持最高的训练速度

#### 混合精度训练
- FP16 配置：
  - `enabled: "auto"`：自动启用 FP16 训练
  - `loss_scale: 0`：动态损失缩放，自动调整防止梯度下溢
  - `loss_scale_window: 1000`：损失缩放调整的观察窗口
```

```
  - `initial_scale_power: 12`：初始损失缩放因子为 2^12 = 4096
  - `hysteresis: 2`：损失缩放调整的迟滞参数
  - `min_loss_scale: 1`：最小损失缩放值

- BF16 配置：
  - `enabled: "auto"`：自动启用 BF16 训练（如果硬件支持）
  - 优势：比 FP16 有更大的数值范围，减少溢出风险

#### 优化器配置
- AdamW 优化器：
  - `lr: "auto"`：学习率由训练脚本自动设置
  - `betas: "auto"`：动量参数自动设置（通常为 [0.9, 0.999]）
  - `eps: "auto"`：数值稳定性参数自动设置（通常为 1e-8）
  - `weight_decay: "auto"`：权重衰减参数自动设置

#### 学习率调度器
- WarmupDecayLR：
  - `warmup_min_lr: "auto"`：预热阶段最小学习率
  - `warmup_max_lr: "auto"`：预热阶段最大学习率
  - `warmup_num_steps: "auto"`：预热步数，通常为总步数的 10%
  - `total_num_steps: "auto"`：总训练步数

#### 训练控制参数
- 梯度处理：
  - `gradient_accumulation_steps: "auto"`：梯度累积步数，用于模拟更大的批量大小
  - `gradient_clipping: "auto"`：梯度裁剪，防止梯度爆炸

- 批量大小配置：
  - `train_batch_size: "auto"`：全局训练批量大小
  - `train_micro_batch_size_per_gpu: "auto"`：每个 GPU 的微批量大小

- 监控配置：
  - `steps_per_print: 100`：每 100 步打印一次训练日志
  - `wall_clock_breakdown: false`：不启用详细的时间性能分析

#### 配置文件说明
所有标记为 `"auto"` 的参数将由 DeepSpeed 根据训练脚本中的设置自动配置，这种设计保证了配置文件的简洁性和灵活性。
```

AI进阶学习

📺 **B站课堂**：https://space.bilibili.com/3546748815411393/pugv

🌐 **官方网站**：https://www.tgltommy.com/ (国内访问需科学上网)

如果你正在关注大模型 Agent、强化学习后训练、RLHF、DPO、GRPO、RLVR 等前沿方向，欢迎关注我最新上线的精品课程：
《大模型 Agent 强化学习实战：从后训练、奖励设计到工业级对齐系统》

这门课程围绕当前大模型 Agent 强化学习的核心技术路线展开，不只是介绍零散算法和论文概念，而是希望帮助学习者建立一套完整的技术地图：从基础原理、论文脉络、关键算法，到开源项目源码解析与工业级系统实践，逐步理解 Agent RL 为什么重要、如何演进，以及在真实应用中如何落地。

课程配套专属飞书知识库 《Agent RL 学习宝典》，包含学习路线、论文资料、项目说明、源码阅读辅助资料和后续持续更新内容，方便长期查阅和跟进前沿技术。

Agent RL 变化很快，但越是变化快，越需要一套系统的学习框架。如果你希望系统进入 Agent RL 方向，而不是停留在碎片化了解阶段，这门课会是一个很好的起点。

课程官网：https://www.tgltommy.com/p/agent-rl（国内访问需科学上网）
B站课堂：https://www.bilibili.com/cheese/play/ss842375604

更多课程详情，扫描以下二维码咨询课程助理：

B站/公众号/YouTube：唐国梁Tommy

关注微信公众号
获取更多学习资源

官方网站：TGLTommy.com(需科学上网)