小红书内容审核助手 - 项目指南

基于 Qwen2-VL + LoRA SFT 的多模态内容审核系统

1. 项目概述

1.1 项目定位

这是一个多模态内容审核系统，能够审核小红书风格的图文内容，判断是否符合平台规范。

1.2 技术栈

- **模型**：Qwen2-VL-2B-Instruct
- **框架**：LLaMA-Factory
- **微调方法**：LoRA + SFT
- **硬件要求**：RTX 4060 8GB 显存

1.3 审核类别

结果	说明
通过 (pass)	内容符合规范
需要修改 (needs_edit)	轻微问题，修改后可发布
违规删除 (remove)	严重违规，直接删除
人工复核 (escalate)	需人工判断

---

2. 环境配置

2.1 创建 Conda 环境

```bash
conda create -n llama_factory python=3.10 -y
conda activate llama_factory
```

2.2 安装 LLaMA-Factory

```bash
git clone https://github.com/hiyouga/LLaMA-Factory.git
cd LLaMA-Factory
pip install -e ".[torch,metrics]" -i https://pypi.tuna.tsinghua.edu.cn/simple
```

2.3 安装 PyTorch（CUDA 版本）

```bash
pip install torch==2.5.1 torchvision torchaudio --index-url https://download.pytorch.org/whl/cu121
```

---

3. 模型下载

```bash
pip install modelscope -i https://pypi.tuna.tsinghua.edu.cn/simple

modelscope download --model Qwen/Qwen2-VL-2B-Instruct --local_dir models/Qwen2-VL-2B-Instruct
```

---

4. 数据集准备

4.1 数据格式

在 `data/content_review_sft.json` 中，格式如下：

```json
```

```
[
  {
    "messages": [
      {
        "role": "system",
        "content": "你是小红书内容审核助手，负责判断用户发布的内容是否符合平台规范。..."
      },
      {
        "role": "user",
        "content": "请审核这段文案：「xxx」"
      },
      {
        "role": "assistant",
        "content": "审核结论：通过\n\n分析：..."
      }
    ]
  }
]
```

4.2 注册数据集

在 `data/dataset_info.json` 中添加：

```
json
{
  "content_review_sft": {
    "file_name": "content_review_sft.json",
    "formatting": "sharegpt",
    "columns": {
      "messages": "messages"
    }
  }
}
```

4.3 数据分布

类型	数量	示例
虚假宣传	~200	"一周瘦20斤"
私域引流	~200	"加V: xxx"

类型	数量	示例
医疗违规	~150	"祖传秘方"
诱导互动	~100	"点赞抽奖"
正常内容	~350	日常分享

5. SFT 训练

5.1 创建训练配置

创建 `train_sft.yaml`：

```yaml
```

```
### 模型配置
model_name_or_path: models/Qwen2-VL-2B-Instruct
trust_remote_code: true

### 训练方法
stage: sft
do_train: true
finetuning_type: lora

### LoRA 配置
lora_target: all
lora_rank: 8
lora_alpha: 16
lora_dropout: 0.05

### 数据集配置
dataset: content_review_sft
template: qwen2_vl
cutoff_len: 2048

### 训练参数
output_dir: saves/qwen2vl-content-review-sft
logging_steps: 10
save_steps: 500
learning_rate: 1.0e-4
num_train_epochs: 3.0
per_device_train_batch_size: 1
gradient_accumulation_steps: 8
max_grad_norm: 1.0
warmup_ratio: 0.1
bf16: true

### 其他
plot_loss: true
```

5.2 开始训练

```
bash

llamafactory-cli train train_sft.yaml
```

5.3 训练结果

指标	值
训练样本	1,020 条
训练轮次	3 epochs
最终 Loss	0.305
训练时长	~12 分钟

---

6. 测试模型

```bash
llamafactory-cli chat \
  --model_name_or_path models/Qwen2-VL-2B-Instruct \
  --adapter_name_or_path saves/qwen2vl-content-review-sft \
  --template qwen2_vl \
  --finetuning_type lora
```

测试样例：

- `请审核这段文案：「分享今天做的午餐，番茄炒蛋，简单又好吃」` → 通过
- `请审核这段文案：「这款美白霜用了一周，皮肤白了三个色号！」` → 违规删除
- `请审核这段文案：「私我领取内部优惠券，比官方便宜50%」` → 违规删除

---

7. 导出模型

7.1 创建导出配置

创建 `export_sft.yaml`：

```yaml
```

```
model_name_or_path: models/Qwen2-VL-2B-Instruct
adapter_name_or_path: saves/qwen2vl-content-review-sft
template: qwen2_vl
finetuning_type: lora
export_dir: models/qwen2vl-content-review-sft-merged
export_size: 2
export_legacy_format: false
```

7.2 执行导出

```bash
llamafactory-cli export export_sft.yaml
```

---

8. 部署 Demo

8.1 安装 Gradio

```bash
pip install gradio qwen-vl-utils -i https://pypi.tuna.tsinghua.edu.cn/simple
```

8.2 运行 Demo

```bash
cd src
python demo.py
```

访问 http://127.0.0.1:7860

---

9. 关键命令速查

```bash
```

```
# 激活环境
conda activate llama_factory

# 训练
llamafactory-cli train train_sft.yaml

# 测试
llamafactory-cli chat --model_name_or_path models/Qwen2-VL-2B-Instruct --adapter_name_or_path saves/qwen2vl-content

# 导出
llamafactory-cli export export_sft.yaml

# 运行 Demo
python src/demo.py
```

---

10. 常见问题

Q1: 显存不够怎么办?

在配置文件中添加:

```yaml
quantization_bit: 4
per_device_train_batch_size: 1
gradient_checkpointing: true
```

Q2: 训练 Loss 不下降?

- 检查数据格式是否正确
- 尝试调低学习率（如 5e-5）
- 增加训练轮次

Q3: 模型输出格式不对?

- 确保训练数据格式统一
- 检查 system prompt 是否一致