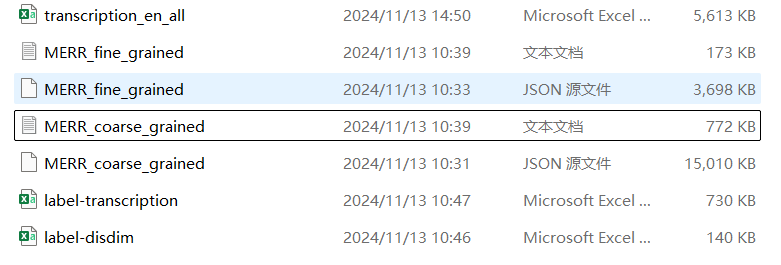
**验证实验1.第一次训练数据包括：emo-llama2中的粗粒度标签、细粒度标签以及MER2024中的train。**



由于给出的标注数据中没有subtitles因此需要从transcription\_en\_all中提取，label-transcription和label-disdim为train中的标注和subtitles。

首先需要将transcription\_en\_all中的subtitles集成到MERR\_coarse\_grained中。

代码路径：C:\Users\admin\Desktop\video-llama2数据集构造\video-llama2\_first.py

处理完数据："C:\Users\admin\Desktop\video-llama2数据集构造\json数据\updated\_data.json"

**2.构造instruction数据,因为MERR\_fine\_grained中以已经有subtitles所以不需要处理，可以直接生成。**

参考emo-llm："<video><VideoHere></video> <feature><FeatureHere></feature> The person in video says: I am feeling a bit down. [emotion] Please determine which emotion label in the video represents: happy, sad, neutral, angry, worried, surprise, fear, contempt, doubt."

使用程序：C:\Users\admin\Desktop\video-llama2数据集构造\video-llama2-instruction.py

处理后文件路径：

C:\Users\admin\Desktop\video-llama2数据集构造\**MERR\_coarse\_instruction.json**

C:\Users\admin\Desktop\video-llama2数据集构造\**MERR\_fine\_instruction.json**

**示例：**

{

        "id": 1,

        "video": "videos/MER2023/sample\_00000003.mp4",

        "conversations": [

            {

                "from": "human",

                "value": "<video>\nThe person in video says: I can forget about it. Please determine which emotion label in the video represents: happy, sad, neutral, angry, worried, surprise, fear, contempt, doubt."

            },

            {

                "from": "gpt",

                "value": "happy"

            }

        ]

    }

合并两个json文件

合并路径后路径：C:\Users\admin\Desktop\video-llama2数据集构造\**MERR\_coarse\_fine\_instruction.json**

**共33105个样本**

**3.音频的instruction：**

{

"id": 0,

"audio": "audio/MER2023/sample\_00000003.wav",

"conversations": [

{

"from": "human",

"value": "<audio>\nThe person in audio says: I can forget about it. Please determine which emotion label in the audio represents: happy, sad, neutral, angry, worried, surprise, fear, contempt, doubt."

},

{

"from": "gpt",

"value": "happy"

}

]

} 无法使用两个.json

Traceback (most recent call last):

File "/data/hongbo.xu/Datasets/MC-ERU/Video-llama2/VideoLLaMA2-audio\_visual/videollama2/train.py", line 683, in <module>

train()

File "/data/hongbo.xu/Datasets/MC-ERU/Video-llama2/VideoLLaMA2-audio\_visual/videollama2/train.py", line 660, in train

data\_module = make\_supervised\_data\_module(tokenizer=tokenizer, data\_args=data\_args)

File "/data/hongbo.xu/Datasets/MC-ERU/Video-llama2/VideoLLaMA2-audio\_visual/videollama2/train.py", line 433, in make\_supervised\_data\_module

train\_dataset = LazySupervisedDataset(

File "/data/hongbo.xu/Datasets/MC-ERU/Video-llama2/VideoLLaMA2-audio\_visual/videollama2/train.py", line 274, in \_\_init\_\_

raise NotImplementedError

NotImplementedError

TEST："C:\Users\admin\Desktop\video-llama2数据集构造\原始json备份\test\_instruction.json"

**4.参数设置**

 --num\_frames 16 \

--num\_train\_epochs 5 \

 --output\_dir "$OUTP\_DIR/${WANDB\_PROJECT}/VideoLLaMA2.1\_train\_v1-7B-AV" \

命令：

nohup bash scripts/custom/va\_joint.sh> output.txt 2>&1 &

tail -f output.txt

推理代码：/data/hongbo.xu/Datasets/MC-ERU/Video-llama2/VideoLLaMA2-audio\_visual/MER\_test\_inference.py

**推理权重2000**：/data/hongbo.xu/Datasets/MC-ERU/Video-llama2/VideoLLaMA2-audio\_visual/work\_dirs/audio\_visual\_stage3\_qwen2/VideoLLaMA2.1\_train\_v1-7B-AV/checkpoint-2000

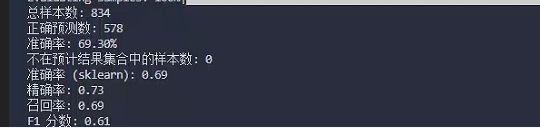


**推理权重3000:**/data/hongbo.xu/Datasets/MC-ERU/Video-llama2/VideoLLaMA2-audio\_visual/work\_dirs/audio\_visual\_stage3\_qwen2/VideoLLaMA2.1\_train\_v1-7B-AV/checkpoint-3000

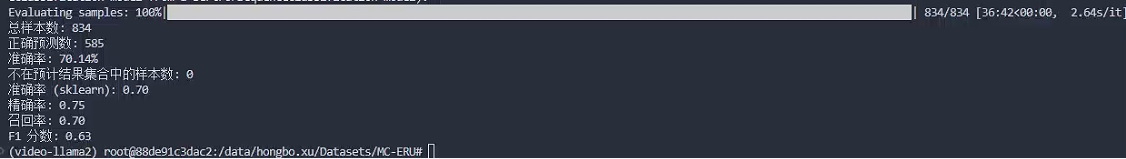


修改后的json：/data/hongbo.xu/Datasets/MC-ERU/Video-llama2/VideoLLaMA2-audio\_visual/datasets/MERR\_coarse\_fine\_instruction\_video\_R.json

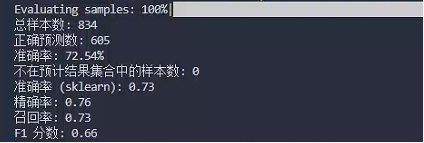
**500：**



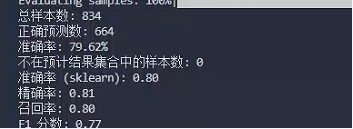
**1000:**



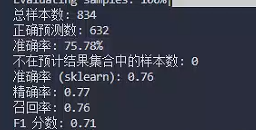
**1500：**



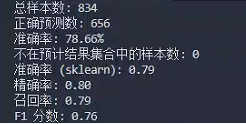
**2000：**



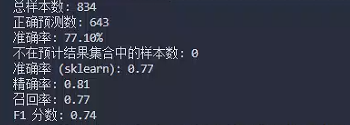
**2500：**



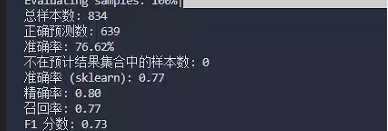
**3000：**



**4000:**

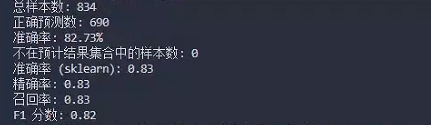


**5000:**

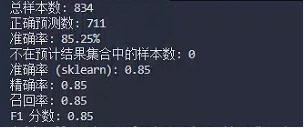


**二阶段：**

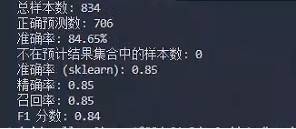
**Stage200：**



**Stage400:**



**600:**



conda env create -f environment.yaml

*依赖如果安装失败*

conda activate llama

Pip install -i <https://pypi.tuna.tsinghua.edu.cn/simple>

torch==2.0.0

torchaudio

torchvision

huggingface-hub==0.18.0

matplotlib==3.7.0

psutil==5.9.4

iopath

pyyaml==6.0

regex==2022.10.31

tokenizers==0.13.2

tqdm==4.64.1

transformers==4.30.0

timm==0.6.13

webdataset==0.2.48

omegaconf==2.3.0

opencv-python==4.7.0.72

decord==0.6.0

peft==0.2.0

sentence-transformers

gradio==3.47.1

accelerate==0.20.3

bitsandbytes==0.37.0

scikit-image

visual-genome

wandb

sudo apt-get install libgl1-mesa-glx

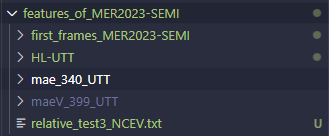
注意：提取特征路径是否正确

train复现

需要自己导入：

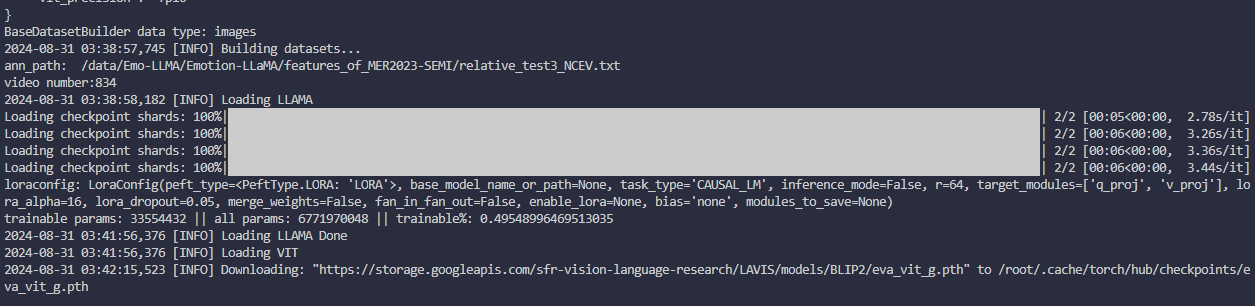
1.[camenduru/MiniGPT-v2 at main (huggingface.co)](https://huggingface.co/camenduru/MiniGPT-v2/tree/main)

2.



3.

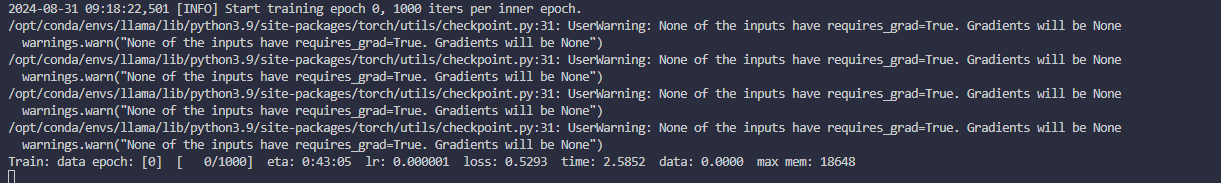


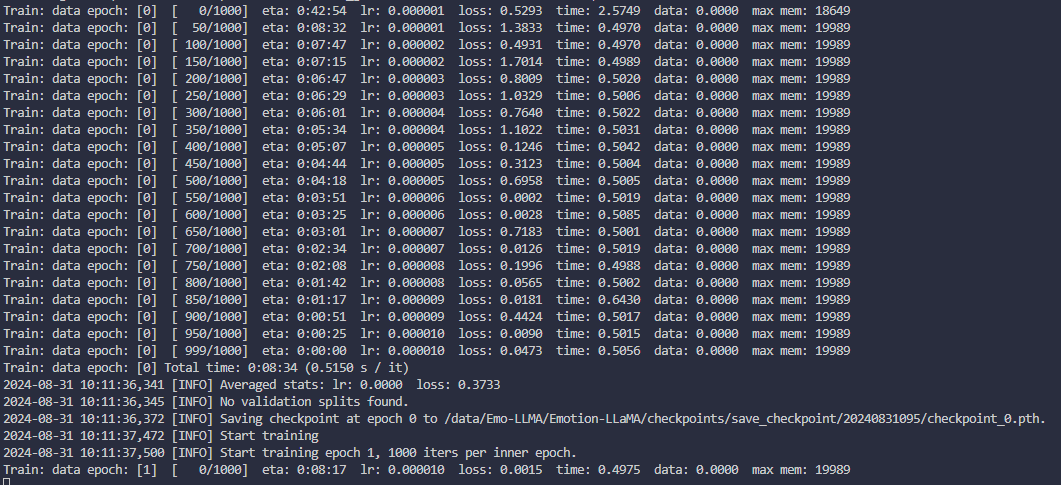


train图片路径修改：/data/Emo-LLMA/Emotion-LLaMA/minigpt4/configs/datasets/firstface/featureface.yaml

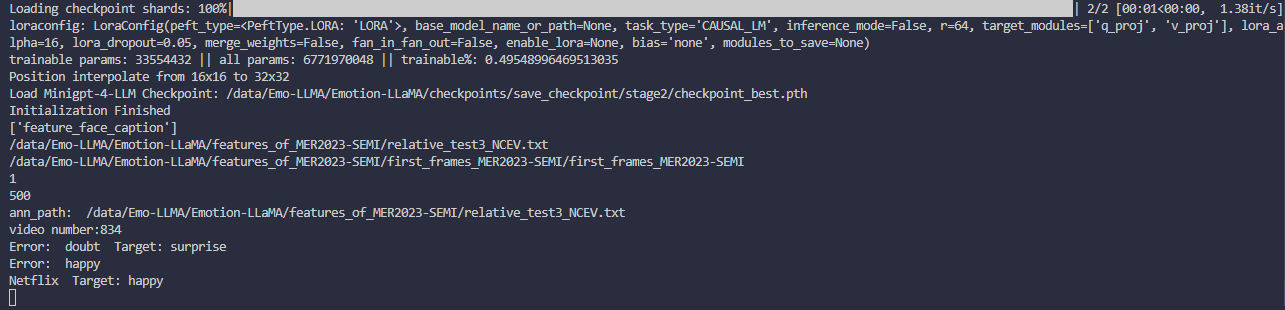
evaluate图片路径修改：/data/Emo-LLMA/Emotion-LLaMA/eval\_configs/eval\_emotion.yaml

每一个epoch：/data/Emo-LLMA/Emotion-LLaMA/minigpt4/tasks/base\_task.py





CUDA\_VISIBLE\_DEVICES=0,1,2,3 torchrun --nproc-per-node 4 train.py --cfg-path train\_configs/Emotion-LLaMA\_finetune.yaml



torchrun --nproc\_per\_node 1 eval\_emotion.py --cfg-path eval\_configs/eval\_emotion.yaml --dataset feature\_face\_caption

