DATA • AI 분석 경진대회

모델 활용 메뉴얼

Ksum

Ⅰ. 학습·배포를 위한 SW 및 HW

Ⅱ. 파일 저장 구조 - 가공/처리된 데이터셋과 학습 모델 포함

Ⅲ. 모델 입력/출력 정보 포함

Ⅳ. 연락처 - 주최즉에서 제출된 모델의 동작 여부를 확인하며, 실행시 문제가 발생할 경우 전화 연락 예정

Ⅰ. 학습·배포를 위한 SW 및 HW

Hardware

· GPU : VRAM 10GB 이상

Software

- Ubuntu : 20.04 LTS
- CUDA : 11.8
- PyTorch : 2.1.0
- Transformers : 4.30.1
- numexpr : 2.8.7
- bitsandbytes : 0.41.1

pip install -r /home/work/sum/requirements.txt

명령어를 통해 추론에 사용된 python 패키지를 모두 설치할 수 있음

Ⅱ. 파일 저장 구조 - 가공/처리된 데이터셋과 학습 모델 포함

- **데이터 파일 경로 : /home/work/sum**
  - **학습/검증 데이터 : train.csv**
  - **테스트 데이터 : test_set.csv**

- **학습 모델 파일 경로 : /home/work/model**
  - **모델파일과 토크나이져 파일이 같이 존재**

Ⅲ. 모델 입력/출력 정보 포함 - 0. 라이브러리 임포트 및 GPU설정

```
import os
os.environ["CUDA_DEVICE_ORDER"]="PCI_BUS_ID"
os.environ["CUDA_VISIBLE_DEVICES"]="0"
import torch
device = 'cuda' if torch.cuda.is_available() else 'cpu'

import torch
from transformers import BitsAndBytesConfig, LlamaForCausalLM, LlamaTokenizer, AutoTokenizer
#from peft import LoraConfig, get_peft_model, PeftModel
from torch.utils.data import Dataset, DataLoader
import json
import os
import pandas as pd
from glob import glob
import random
```

Ⅲ. 모델 입력/출력 정보 포함 - 1. 모델 & 토크나이져 로드

```
Qconfig = BitsAndBytesConfig(
            load_in_8bit=True,
            llm_int8_threshold=6.0,
            load_in_8bit_skip_modules="embed_tokens,lm_head",
            bnb_4bit_compute_dtype=torch.float16,
            bnb_4bit_use_double_quant=True,
            bnb_4bit_quant_type="nf4"
)

tokenizer=AutoTokenizer.from_pretrained('/data2/shindj/sum_model')
model=LlamaForCausalLM.from_pretrained('/data2/shindj/sum_model', quantization_config=Qconfig)
# peft_model_="/home/master/MLP/summarization/yoohg/alpaca/output_beomi/sft_lora_model"
# model=PeftModel.from_pretrained(model, peft_model_)
# model.to(device)
```

Ⅲ. 모델 입력/출력 정보 포함 - 2. 요약 생성

```
with torch.no_grad():
    # 엑셀 파일 읽기
    df = pd.read_csv('../data/test_set.csv')

    # 요약 생성 및 결과 저장
    output_summaries = []
    for i in range(len(df)):
        temp=f"아래의 대화록을 요약해줘.\n{df['내용'][i]}"
        source = PROMPT_TEMPLATE.format_map({'instruction':temp})
        tokened = tokenizer(source, return_tensors="pt").to(device)
        summary_ids = model.generate(input_ids=tokened.input_ids, max_length=4096, do_sample=True, top_k=20, top_p=0
        summary = tokenizer.decode(summary_ids[0], skip_special_tokens=True)
        output_summaries.append(summary)
        print(summary)


    print(output_summaries[66])
    # 결과를 새로운 CSV 파일로 저장
    #pd.DataFrame(data=output_summaries,columns=['llm']).to_csv('/data2/shindj/llm_result_0.csv', index=False, encod
    #print(f"llm_result_0 작업 완료.\n\n")

print("작업이 완료되었습니다.")
```

Ⅲ. 모델 입력/출력 정보 포함 - Architecture

모델 입력

```
PROMPT_TEMPLATE = (
        "[INST] <<SYS>>\n"
        "You are a helpful AI assistant. 당신은 유능한 AI 어시스턴트 입니다.\n"
        "<</SYS>>\n\n{instruction} [/INST]"
    )
```

아래의 데이터록을 요약해줘.
XXX 위원장 : ~~~
XXX 국장 : ~~~

Ⅲ. 모델 입력/출력 정보 포함 - Architecture

모델 출력

박경미위원은 세월호 관련
수색작업에 참여한 민간잠수사들의
처우에 대한 문제를 제기하였다.
조윤선후보자는 관련부처와
협의하여 미비한 점을 개선할 것을
약속하였다.

Ⅲ. 모델 입력/출력 정보 포함 - Architecture

모델 입력

토큰화
by 토크나이저

모델 출력

디코딩
by 토크나이저