

[TTS] VITS-JAPANESES-FINETUNING DEMO

담당자

JUBI

REFERENCE

1 환경 설정

가상환경

git clone

pre-trained model 다운로드

preprocessing

2 .ipynb

Edit config

Check if your dataset meets the requirements (Optional)

Process the wav files of the dataset to make them meet the requirements (Optional)

Start training

TROUBLE SHOOTING

preprocessing

ModuleNotFoundError: No module named 'pyopenjtalk'

Start Training

Descriptors cannot be created directly

REFERENCE

- 가상환경 복제

conda 기존 환경을 복사하여 새로운 환경을 만들자

기존 환경에 설치된 라이브러리들을 동일하게 사용할건데 개발 환경은 분리하고 싶을 경우 다시 설치하려면 시간만 엄청 오래걸리지.. 이럴 경우 기존 환경을 복사하여 새로운 환경을 만드는 방법. 가상환경 복제하여 새로

<https://jangjy.tistory.com/341>



- vits

<https://github.com/jaywalnut310/vits>

- vits-finetuning(Japanese)

GitHub - SayaSS/vits-finetuning: Fine-Tuning your VITS model using a pre-trained model

Fine-Tuning your VITS model using a pre-trained model - GitHub - SayaSS/vits-finetuning: Fine-Tuning your VITS model using a pre-trained model

<https://github.com/SayaSS/vits-finetuning/tree/main>

SayaSS
finetun

Fine-Tuning your VITS model

1 Contributor 10

1 환경 설정

가상환경

- vits demo 환경 복제

```
conda create -n JPTTS --clone TTS
```

```
(base) ljhp1004@S220:~/TTS/vits-finetuning$ conda create -n ttsJPEnv --clone ttsenv
Source: /home/ljhp1004/anaconda3/envs/ttsenv
Destination: /home/ljhp1004/anaconda3/envs/ttsJPEnv
Packages: 20
Files: 26685
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate ttsJPEnv
#
# To deactivate an active environment, use
#
#     $ conda deactivate
#
Retrieving notices: ...working... done
```

- 가상환경 활성화

```
source activate JPTTS
```

- 커널 연결

```
python -m ipykernel install --user --name TTS --display-name "JPTTS"
```

- librosa 버전 변경

```
pip install librosa==0.9.1
```

- gradio 설치

```
pip install gradio
```

git clone

```
git clone https://github.com/SayaSS/vits-finetuning.git
```

pre-trained model 다운로드

```
mkdir checkpoints && cd checkpoints
```

```
wget https://huggingface.co/spaces/sayashi/vits-uma-genshin-honkai/resolve/main/model/G_0.p
```

```
wget https://huggingface.co/spaces/sayashi/vits-uma-genshin-honkai/resolve/main/model/D_0.p
```

preprocessing

```
cd ..
```

- 현재 경로 : ../vits-finetuning

```
python preprocessing.py --filelists filelists/miyu_train.txt filelists/miyu_val.txt
```

```
ttsJPenv) ljhpl004@S220:~/TTS/vits-finetuning$ python preprocess.py --filelists filelists/miyu_train.txt filelists/miyu_val.txt
START: filelists/miyu_train.txt
Downloading: "https://github.com/r9y9/open_jtalk/releases/download/v1.1.1/open_jtalk_dic_utf_8-1.1.1.tar.gz"
dic.tar.gz: 100%|██████████████████████████████████████████████████████████████████████████| 22.6M/22.6M [00:03<00:00, 6.38MB/s]
Extracting tar file /home/ljhpl004/anaconda3/envs/ttsJPenv/lib/python3.7/site-packages/pyopenjtalk/dic.tar.gz
START: filelists/miyu_val.txt
```

- trouble shooting

2 .ipynb

Edit config

```
#@title Edit config
import json

batchsize = 16 #@param {type:"number"}

training_files = "filelists/miyu_train.txt.cleaned" #@param {type:"string"}
validation_files = "filelists/miyu_val.txt.cleaned" #@param {type:"string"}

config = json.load(open("configs/config.json"))
config['train']['batch_size'] = batchsize
config['data']['training_files'] = training_files
config['data']['validation_files'] = validation_files

with open("configs/config.json", 'w+') as f:
    json.dump(config, f, indent=4)
```

Check if your dataset meets the requirements (Optional)

```
#@title Check if your dataset meets the requirements (Optional)
import os
import soundfile as sf
all_meet = True
wav_path = "wav/ba/miyu"##@param {type:"string"}
for file_name in os.listdir(wav_path):
    if not file_name.endswith(".wav"):
        continue
    data, sr = sf.read (rf"{wav_path}/{file_name}")
    n_channels = data.shape [1] if data.ndim > 1 else 1
    subtype = sf.info (rf"{wav_path}/{file_name}").subtype
    if sr == 22050 and n_channels == 1 and subtype == "PCM_16":
        filesize = os.path.getsize(rf"{wav_path}/{file_name}")/1024
        if filesize>500 or filesize<16:
            print(f"Warning: {file_name}: wav files larger than 500KB and smaller than 16KB will be removed")
            continue
    else:
        print(f'\x1b[31m>Error: {file_name} does not meet the criteria because:\n\x1b[0m')
        if sr != 22050:
            print(" - sample rate is " + str (sr) + " instead of 22050")
        if n_channels != 1:
            print(" - number of channels is " + str (n_channels) + " instead of 1")
        if subtype != "PCM_16":
            print(" - subtype is " + subtype + " instead of PCM_16")
        all_meet = False
if all_meet:
    print ("All files meet the requirements")
```

Process the wav files of the dataset to make them meet the requirements (Optional)

```
#@title Process the wav files of the dataset to make them meet the requirements (Optional)
import os
import librosa
from tqdm import tqdm
wav_path = "wav/ba/miyu" #@param {type:"string"}
for file_name in tqdm(os.listdir(wav_path)):
    if file_name.endswith(".spec.pt"):
        os.remove(rf"{wav_path}/{file_name}")
        continue
    y, sr = librosa.load(rf"{wav_path}/{file_name}", sr=22050, mono=True)
    sf.write(rf"{wav_path}/{file_name}", y, 22050, subtype="PCM_16")
```

Start training

```
#@title Start training
!python train_ms.py -c configs/config.json -m checkpoints
```

- trouble shooting

TROUBLE SHOOTING

preprocessing

ModuleNotFoundError: No module named 'pyopenjtalk'

```
(ttsenv) ljhpl004@S220:~/TTS/vits-finetuning$ python preprocess.py --filelists filelists/miyu_train.txt filelists/miyu_val.txt
Traceback (most recent call last):
  File "preprocess.py", line 2, in <module>
    import text
  File "/home/ljhpl004/TTS/vits-finetuning/text/__init__.py", line 2, in <module>
    from text import cleaners
  File "/home/ljhpl004/TTS/vits-finetuning/text/cleaners.py", line 2, in <module>
    from text.japanese import japanese_to_romaaji_with_accent
  File "/home/ljhpl004/TTS/vits-finetuning/text/japanese.py", line 3, in <module>
    import pyopenjtalk
ModuleNotFoundError: No module named 'pyopenjtalk'
```

- pyopenjtalk
 - 일본어 음성 합성 엔진
- OpenJTalk, MeCab 설치

```
sudo apt-get install open-jtalk open-jtalk-mecab-naist-jdic
```

- pyopenjtalk 설치

```
pip install pyopenjtalk
```

- pyopenjtalk 0.3.3 설치

Start Training

Descriptors cannot not be created directly

```
In [8]: !python train_ms.py -c configs/config.json -m checkpoints
```

```
Traceback (most recent call last):
  File "train_ms.py", line 10, in <module>
    from torch.utils.tensorboard import SummaryWriter
  File "/home/ljhp1004/anaconda3/envs/ttsenv/lib/python3.7/site-packages/torch/utils/tensorboard/__init__.py", line 8, in <module>
    from .writer import FileWriter, SummaryWriter # noqa F401
  File "/home/ljhp1004/anaconda3/envs/ttsenv/lib/python3.7/site-packages/torch/utils/tensorboard/writer.py", line 14, in <module>
    from tensorboard.compat.proto.event_pb2 import SessionLog
  File "/home/ljhp1004/anaconda3/envs/ttsenv/lib/python3.7/site-packages/tensorboard/compat/proto/event_pb2.py", line 17, in <module>
    from tensorboard.compat.proto import summary_pb2 as tensorboard_dot_compat_dot_proto_dot_summary__pb2
  File "/home/ljhp1004/anaconda3/envs/ttsenv/lib/python3.7/site-packages/tensorboard/compat/proto/summary_pb2.py", line 17, in <module>
    from tensorboard.compat.proto import tensor_pb2 as tensorboard_dot_compat_dot_proto_dot_tensor__pb2
  File "/home/ljhp1004/anaconda3/envs/ttsenv/lib/python3.7/site-packages/tensorboard/compat/proto/tensor_pb2.py", line 16, in <module>
    from tensorboard.compat.proto import resource_handle_pb2 as tensorboard_dot_compat_dot_proto_dot_resource__handle__pb2
  File "/home/ljhp1004/anaconda3/envs/ttsenv/lib/python3.7/site-packages/tensorboard/compat/proto/resource_handle_pb2.py", line 16, in <module>
    from tensorboard.compat.proto import tensor_shape_pb2 as tensorboard_dot_compat_dot_proto_dot_tensor__shape__pb2
  File "/home/ljhp1004/anaconda3/envs/ttsenv/lib/python3.7/site-packages/tensorboard/compat/proto/tensor_shape_pb2.py", line 42, in <module>
    e>
    serialized_options=None, file=DESCRIPTOR),
  File "/home/ljhp1004/anaconda3/envs/ttsenv/lib/python3.7/site-packages/google/protobuf/descriptor.py", line 561, in __new__
    _message.Message._CheckCalledFromGeneratedFile()
TypeError: Descriptors cannot not be created directly.
If this call came from a _pb2.py file, your generated code is out of date and must be regenerated with protoc >= 3.19.0.
If you cannot immediately regenerate your protos, some other possible workarounds are:
  1. Downgrade the protobuf package to 3.20.x or lower.
  2. Set PROTOCOL_BUFFERS_PYTHON_IMPLEMENTATION=python (but this will use pure-Python parsing and will be much slower).

More information: https://developers.google.com/protocol-buffers/docs/news/2022-05-06#python-updates
```

TypeError: Descriptors cannot not be created directly.

If this call came from a _pb2.py file, your generated code is out of date and must be regenerated with protoc >= 3.19.0.

If you cannot immediately regenerate your protos, some other possible workarounds are:

1. Downgrade the protobuf package to 3.20.x or lower.
2. Set PROTOCOL_BUFFERS_PYTHON_IMPLEMENTATION=python (but this will use pure-Python parsing and will be much slower).

- Protocol Buffers (protobuf) 라이브러리와 관련 있는 문제
- 방법 1 : protobuf 패키지 다운그레이드

```
pip install protobuf==3.20.0
```

| | |
|-------|--------|
| 바꾸기 전 | 4.24.4 |
| 바꾼 후 | 3.20.0 |