

VITS FINE-TUNING with KSS

☰ 태그

REFERENCE

1 SERVER 설정

- ① UBUNTU SERVER 설정
- ② JUPYTER SERVER 설정

2 PRE-REQUISITES

① 가상 환경

- 가상환경 생성
 - 가상환경 목록 조회
 - 가상 환경 활성화
 - 설치된 패키지 확인
- ② Kernel 생성
- ipykernel 설치
 - 커널 생성
 - 커널 목록 조회

JUPYTER NOTEBOOK INTERFACE에서 확인

- ③ git clone
- ④ requirements.txt 설치

- 가상 환경 활성화
- requirements.txt 설치
- 설치된 패키지 조회
- 확장 모듈 빌드
- espeak 설치
- espeak 설치 확인
- protobuf 패키지 downgrade
- jamo 패키지 설치
- ko_pron 패키지 설치

3 Preprocessing KSS Dataset

- ① Download KSS
- ② KSS 음성 Dataset 이동
- 디렉토리 1, 2, 3, 4 이동
- transcript.v.1.4.txt 이동
- ③ transcript 편집 (윈도우에서 작업)
- transcript.v.1.4.txt 을 Excel로 열기
- 필요 없는 열 삭제
- 데이터 나누기 : 1 / 2 / 3 / 4
- ④ ubuntu 서버로 편집한 transcript 이동
- ⑤ preprocessing

- 가상환경 활성화
- preprocess.py 경로로 이동
- preprocess scripts

4 config 설정

- ① korean cleaner 모듈 생성
- 경로 이동
- korean.py 파일 생성
- ② cleaners.py 수정
- 경로 확인
- cleaners.py 수정
- ③ config 파일 생성
- 경로 이동
- .json 파일 복제
- kss.json 파일 수정

5 Train Model

- ① 경로 이동
- ② training

✖ ERROR ✖

REFERENCE

GitHub - jaywalnut310/vits: VITS: Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech
VITS: Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech - GitHub - jaywalnut310/vits: Conditional Variational Autoencoder with Adversarial Learning f...
<https://github.com/jaywalnut310/vits/tree/main>

1 SERVER 설정

① UBUNTU SERVER 설정

② JUPYTER SERVER 설정

2 PRE-REQUISITES

① 가상 환경

가상환경 생성

```
conda create -n tts python=3.7 -y
```

가상환경 목록 조회

```
conda info --envs
```

가상 환경 활성화

```
source activate tts
```

설치된 패키지 확인

```
conda list
```

```
(tts) 1jhp1004@s220:~$ conda list
# packages in environment at /home/1jhp1004/anaconda3/envs/tts:
#
#      Name          Version   Build  Channel
_libgcc_mutex       0.1      main
_openmp_mutex       5.1      1_gnu
ca-certificates    2023.12.12  h06a4308_0
certifi             2022.12.7   py37h06a4308_0
ld_impl_linux-64    2.38     h1181459_1
libffi              3.4.4     h6a678d5_0
libgcc-ng           11.2.0    h1234567_1
libgomp             11.2.0    h1234567_1
libstdcxx-ng        11.2.0    h1234567_1
ncurses             6.4       h6a678d5_0
openssl             1.1.1w    h7f8727e_0
pip                 22.3.1   py37h06a4308_0
python              3.7.16    h7a1cb2a_0
readline            8.2       h5eee18b_0
setuptools          65.6.3   py37h06a4308_0
sqlite              3.41.2    h5eee18b_0
tk                  8.6.12    h1ccaba5_0
wheel               0.38.4   py37h06a4308_0
xz                  5.4.5     h5eee18b_0
zlib                1.2.13    h5eee18b_0
```

② Kernel 생성

ipykernel 설치

```
pip install ipykernel
```

커널 생성

```
python -m ipykernel install --user --name tts --display-name "tts"
```

```
(tts) 1jhp1004@S220:~$ python -m ipykernel install --user --name tts --display-name "tts"  
Installed kernelspec tts in /home/ljhp1004/.local/share/jupyter/kernels/tts
```

- ttsenv라는 가상 환경의 kernel을 “ttsenv”라는 이름으로 생성

커널 목록 조회

```
jupyter kernelspec list
```

JUPYTER NOTEBOOK INTERFACE에서 확인

- New 버튼 클릭 시 ttsenv 이름의 Kernel이 없을 경우 jupyter notebook 재실행

③ git clone

```
mkdir TTS;cd TTS
```

```
git clone https://github.com/jaywalnut310/vits.git;cd vits
```

④ requirements.txt 설치

가상 환경 활성화

```
source activate tts
```

- 이미 활성화 되어 있다면 생략

requirements.txt 설치

```
pip install -r requirements.txt
```

- 현재 디렉토리에 있는 `requirements.txt` 파일을 사용하여 Python 패키지 설치
- `-r` 옵션
 - “파일을 읽음(read)”
 - pip에게 설치할 패키지 목록을 파일에서 읽도록 지시하는 옵션
 - requirements.txt 파일에 명시된 모든 패키지를 설치하도록 한다
- 설치해야 할 항목

| | |
|------------|---------|
| Cython | 0.29.21 |
| librosa | 0.8.0 |
| matplotlib | 3.3.1 |
| numpy | 1.18.5 |
| phonemizer | 2.2.1 |

| | |
|-------------|-------|
| scipy | 1.5.2 |
| tensorboard | 2.3.0 |
| torch | 1.6.0 |
| torchvision | 0.7.0 |
| Unidecode | 1.1.1 |

설치된 패키지 조회

```
conda list
```

- 위의 항목이 다 설치 되었는지 확인한다
- 하나씩 확인하는 방법

```
conda list 이름
```

```
conda list cython
```

```
(tts) ljh1004@S220:~/TTSTEST/TTS/vits$ conda list cython
# packages in environment at /home/ljh1004/anaconda3/envs/tts:
#
#          Name           Version      Build Channel
cython     0.29.21            py37_0    pypi_0    pypi
```

- 설치 확인

```
conda list protobuf
```

```
(tts) ljh1004@S220:~/TTSTEST/TTS/vits$ conda list protobuf
# packages in environment at /home/ljh1004/anaconda3/envs/tts:
#
#          Name           Version      Build Channel
protobuf   3.20.0             py37_0    pypi_0    pypi
```

확장 모듈 빌드

```
cd monotonic_align
```

```
python setup.py build_ext --build-lib=./
```

- C/C++로 작성된 파이썬 확장 모듈을 빌드하고 그 결과물을 지정된 디렉토리에 저장한다
- `python` : Python 인터프리터를 실행하는 명령
- `setup.py` : 파이썬 패키지를 빌드하기 위한 설정 파일
- `build_ext` : 빌드 확장(extensions)를 수행하는 옵션. 확장은 C/C++ 코드로 작성된 파이썬 모듈을 의미
- `--build-lib=./` : 빌드된 확장 모듈을 저장할 디렉토리 지정. `./` 는 현재 디렉토리를 의미
- 실행 결과 생성되는 것

[빌드 전]

- `__init__.py`
- `core.pyx`
- `setup.py`

[빌드 후]

- `build`
- `core.c`
- `monotonic_align`
- `__init__.py`
- `core.pyx`
- `setup.py`

espeak 설치

espeak : 텍스트를 오디오로 변환

```
sudo apt-get install espeak -y
```

espeak 설치 확인

```
espeak --version
```

```
(ttsenv) ljhpc1004@s220:~/TTS$ espeak --version
eSpeak text-to-speech: 1.48.03 04.Mar.14 Data at: /usr/lib/x86_64-linux-gnu/espeak-data
```

- 음성 파일 만들고 재생해보기

```
espeak "Hello, this is a test" -w output.wav
```

- 재생이 안될 수도 있음. 재생 안되면 일단 PASS

protobuf 패키지 downgrade

```
pip install protobuf==3.20
```

```
(tts) ljhpc1004@s220:~/TTSTEST/TTS/vits$ pip install protobuf==3.20
Collecting protobuf==3.20
  Using cached protobuf-3.20.0-cp37-cp37m-manylinux_2_5_x86_64.manylinux1_x86_64.whl (1.0 MB)
Installing collected packages: protobuf
  Attempting uninstall: protobuf
    Found existing installation: protobuf 4.24.4
    Uninstalling protobuf-4.24.4:
      Successfully uninstalled protobuf-4.24.4
Successfully installed protobuf-3.20.0
```

jamo 패키지 설치

```
pip install jamo
```

```
(tts) ljhpc1004@s220:~/TTSTEST/TTS/vits$ pip install jamo
Collecting jamo
  Using cached jamo-0.4.1-py3-none-any.whl (9.5 kB)
Installing collected packages: jamo
Successfully installed jamo-0.4.1
```

ko_pron 패키지 설치

```
pip install ko_pron
```

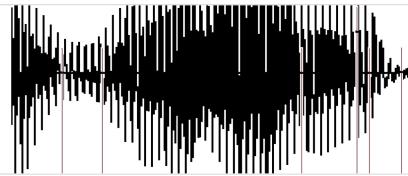
```
(tts) ljhpc1004@s220:~/TTSTEST/TTS/vits$ pip install ko_pron
Collecting ko_pron
  Using cached ko_pron-1.3-py3-none-any.whl (12 kB)
Installing collected packages: ko_pron
Successfully installed ko_pron-1.3
```

3 Preprocessing KSS Dataset

① Download KSS

Korean Single Speaker Speech Dataset
KSS Dataset: Korean Single Speaker Speech Dataset

<https://www.kaggle.com/datasets/bryanhpark/korean-single-speaker-speech-dataset>



- Kaggle에 로그인 후 다운로드

- Archive.zip 다운로드

- Archive.zip 구조

- kss
 - 1
 - 많은 .wav 파일들
 - 2
 - 많은 .wav 파일들
 - 3
 - 많은 .wav 파일들
 - 4
 - 많은 .wav 파일들
- transcript.v1.4.txt

② KSS 음성 Dataset 이동

디렉토리 1, 2, 3, 4 이동

- 경로 : TTS/vits/

- 결과

- TTS/vits/1
 - TTS/vits/2
 - TTS/vits/3
 - TTS/vits

```
(base) ljhpi004@S220:~/TTSTEST/TTS/vits$ ls
1 LICENSE           config             kss               modules.py      resources      transforms.py
2 README.md         datasets_vits.py   losses.py       monotonic_align  test.py        utils.py
3 attentions.py     filelists          mel_processing.py preprocess.py    train.py
4 commons.py        inference.ipynb   models.py      requirements.txt train_ms.py
```

| Name | Last Modified | File size |
|-------------------|---------------|-----------|
| 1 LICENSE | 몇 초 전 | |
| 2 README.md | 2분 전 | |
| 3 attentions.py | 2분 전 | |
| 4 commons.py | 1분 전 | |
| config | 한 시간 전 | |
| datasets_vits.py | 몇 초 전 | |
| filelists | 1분 전 | |
| inference.ipynb | 한 시간 전 | 6.1 kB |
| losses.py | 한 시간 전 | 11.8 kB |
| mel_processing.py | 한 시간 전 | 4.78 kB |
| monotonic_align | 한 시간 전 | 15 kB |
| models.py | 한 시간 전 | 1.07 kB |
| modules.py | 한 시간 전 | 1.31 kB |
| resources | 한 시간 전 | 3.83 kB |
| test.py | 한 시간 전 | 19.4 kB |
| transforms.py | 한 시간 전 | 13.2 kB |

transcript.v1.4.txt 이동

- 경로 : TTS/vits/filelists/

- 결과

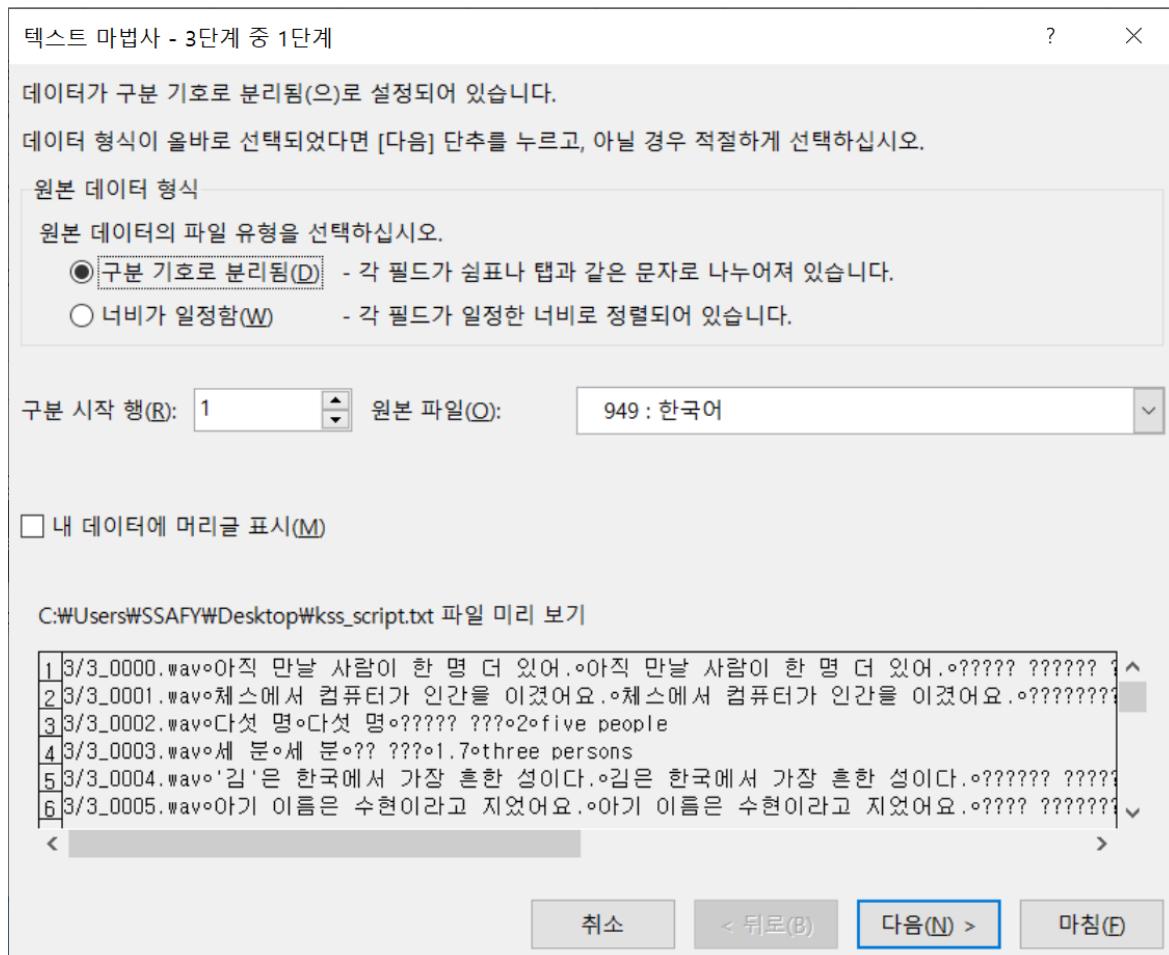
- TTS/vits/filelists/transcript.v1.4.txt

③ transcript 편집 (윈도우에서 작업)

| Excel 프로그램이 있는 환경에서 작업 (ex. 윈도우)

transcript.v.1.4.txt 을 Excel로 열기

- 구분 기호로 분리됨 선택 → 다음



- 구분 기호 : 기타 선택 → | 입력 → 다음

텍스트 마법사 - 3단계 중 2단계

데이터의 구분 기호를 설정합니다. 미리 보기 상자에서 적용된 텍스트를 볼 수 있습니다.

구분 기호

탭(I)
 세미콜론(M)
 쉼표(C)
 공백(S)
 기타(O):

연속된 구분 기호를 하나로 처리(R)

텍스트 한정자(Q): "

데이터 미리 보기(P)

```
3/3_0000.wav◦아직 만날 사람이 한 명 더 있어.◦아직 만날 사람이 한 명 더 있어.◦????? ?????? ???  
3/3_0001.wav◦체스에서 컴퓨터가 인간을 이겼어요.◦체스에서 컴퓨터가 인간을 이겼어요.◦?????????  
3/3_0002.wav◦다섯 명◦다섯 명◦????? ???◦five people  
3/3_0003.wav◦세 분◦세 분◦?? ???◦1.7◦three persons  
3/3_0004.wav◦'김'은 한국에서 가장 흔한 성이다.◦김은 한국에서 가장 흔한 성이다.◦????? ??????  
3/3_0005.wav◦아기 이름은 수현이라고 지었어요.◦아기 이름은 수현이라고 지었어요.◦????? ????????
```

< >

취소 < 뒤로(B) 다음(N) > 마침(F)

- 열 데이터 서식 : 일반 선택 → 마침

텍스트 마법사 - 3단계 중 3단계

각 열을 선택하여 데이터 서식을 지정합니다.

열 데이터 서식

일반(G)
 텍스트(I)
 날짜(D): 년월일
 열 가져오지 않음(건너뜀)(I)

[일반]을 선택하면 숫자 값은 숫자로, 날짜 값은 날짜로, 모든 나머지 값은 텍스트로 변환됩니다.

고급(A)...

데이터 미리 보기(P)

| 일반 | 일반 | 일반 |
|--------------|---------------------------|-------------------------|
| 1/1_0000.wav | 그는 괜찮은 척하려고 애쓰는 것 같았다. | 그는 괜찮은 척하려고 애쓰는 것 같았... |
| 1/1_0001.wav | 그녀의 사랑을 얻기 위해 애썼지만 헛수고였다. | 그녀의 사랑을 얻기 위해 애썼지만 헛... |
| 1/1_0002.wav | 용돈을 아껴 써라. | 용돈을 아껴 써라. |
| 1/1_0003.wav | 그는 아내를 많이 아낀다. | 그는 아내를 많이 아낀다. |
| 1/1_0004.wav | 그 애 전화번호 알아? | 그 애 전화번호 알아? |
| 1/1_0005.wav | 차에 대해 잘 아세요? | 차에 대해 잘 아세요? |

취소 < 뒤로(B) 다음(N) > 마침(E)

필요 없는 열 삭제

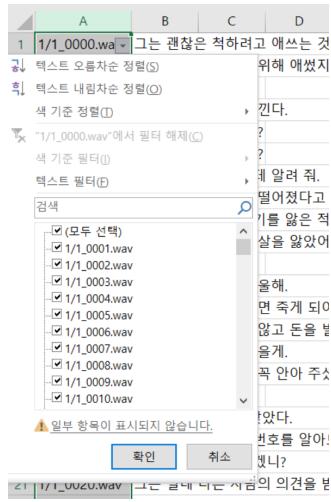
- C, D, E, F 열 삭제

[삭제 전]

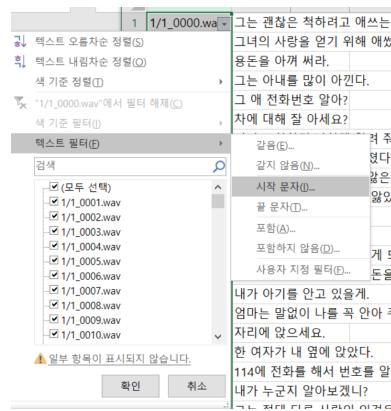
[삭제 후]

데이터 나누기 : 1 / 2 / 3 / 4

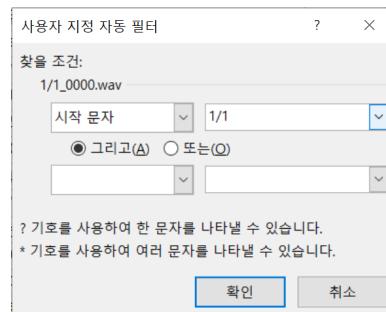
- A열 선택
- 데이터 탭 → 필터
- A열 1행에 생긴 필터 드롭다운 클릭



- 텍스트 필터 → 시작 문자



- 시작 문자에 1/1 입력 → 확인



- 전체 선택 → 복사 (ctrl + c)
- 새로운 sheet 생성 후 붙여넣기 (ctrl + v)
- 새로운 sheet 내용 전체 선택 (ctrl + a) → 복사 (ctrl + c)
- 메모장에 붙여넣기 (ctrl + v)

```

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
1/1,0000.wav 그는 괜찮은 척하려고 애쓰는 것 같았다.
1/1,0001.wav 그녀의 사랑을 얻기 위해 애썼지만 헛수고였다.
1/1,0002.wav 용돈을 아껴 써라.
1/1,0003.wav 그는 아내를 많이 아낀다.
1/1,0004.wav 그에 전화번호 알아?
1/1,0005.wav 차에 대해 잘 아세요?
1/1,0006.wav 거기 도착하면 나한테 알려 줘.
1/1,0007.wav 그들은 내가 사랑에 빠졌다고 알려 왔다.
1/1,0008.wav 나는 상대으면서 강기를 얻은 적이 한 번도 없다.
1/1,0009.wav 사랑을 인심하게 몸살을 앓았어요.
1/1,0010.wav 요즘 컴퓨터가 안돼요.
1/1,0011.wav 장거리 절 안돼서 우울해.
1/1,0012.wav 아무 것도 먹지 않으면 죽게 되어 있다.
1/1,0013.wav 그녀는 이유를 물지 않고 돈을 벌려 주었다.
1/1,0014.wav 내가 이돈을 안고 있었습니다.
1/1,0015.wav 엄마는 말없이 나를 꼭 안아 주셨다.
1/1,0016.wav 자본에 있으세요.
1/1,0017.wav 한 여자가 내 옆에 앉았다.
1/1,0018.wav 114에 전화를 해서 번호를 알아보시지 그래세요?
1/1,0019.wav 내가 누군지 알아보겠나?
1/1,0020.wav 그는 절대 다른 사람의 의견을 받아들이지 않는다.
1/1,0021.wav 아이들도 언젠가는 그녀를 엄마로 받아들일 거야.
1/1,0022.wav 어디에서 영어를 배우셨어요?
1/1,0023.wav 동료에게서 수화를 배우고 있습니다
Ln 6, Col 26 100% Windows (CRLF) UTF-8

```

- 편집 탭 → 바꾸기(R)...

```

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
1/1,0  실행 위치(N) Ctrl+Z 하라고 애쓰는 것 같았다.
1/1,0  상대나이거(I) Ctrl+X 얻기 위해 애썼지만 헛수고였다.
1/1,0  파일열기(P) Ctrl+N 라.
1/1,0  새문서(N) Ctrl+Y 이 아낀다.
1/1,0  Bing으로 검색(S)... Ctrl+F 알아?
1/1,0  찾기(F) Ctrl+G 가세요?
1/1,0  다음 찾기(N) F3 나한테 알려 줘.
1/1,0  이번 찾기(V) Shift+F3 험에 빠졌다고 알려 왔다.
1/1,0  바꾸기(R) Ctrl+H 서 강기를 얻은 적이 한 번도 없다.
1/1,0  이동(G)... Ctrl+O 게 몸살을 앓았어요.
1/1,0  모두 선택(A) Ctrl+A 됐네요.
1/1,0  시간/날짜(D) F5 서 우울해.
1/1,0012.wav 아무 것도 먹지 않으면 죽게 되어 있다.
1/1,0013.wav 그녀는 이유를 물지 않고 돈을 벌려 주었다.
1/1,0014.wav 내가 아기를 안고 있개.
1/1,0015.wav 엄마는 말없이 나를 꼭 안아 주셨다.
1/1,0016.wav 자본에 있으세요.
1/1,0017.wav 한 여자가 내 옆에 앉았다.
1/1,0018.wav 114에 전화를 해서 번호를 알아보시지 그래세요?
1/1,0019.wav 내가 누군지 알아보겠나?
1/1,0020.wav 그는 절대 다른 사람의 의견을 받아들이지 않는다.
1/1,0021.wav 아이들도 언젠가는 그녀를 엄마로 받아들일 거야.
1/1,0022.wav 어디에서 영어를 배우셨어요?
1/1,0023.wav 동료에게서 수화를 배우고 있습니다
Ln 7, Col 27 100% Windows (CRLF) UTF-8

```

- 찾을 내용에 공백 복사해서 붙여넣기

[복사]

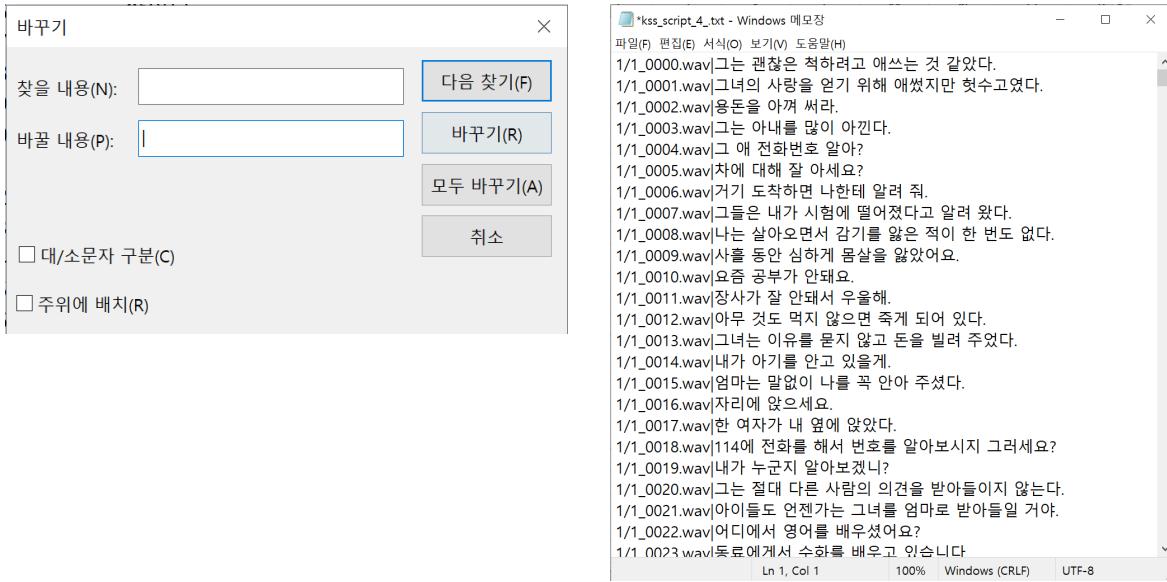
[붙여넣기]

```

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
1/1,0000.wav 그는 괜찮은 척하려고 애쓰는 것 같았다.
1/1,0001.wav 그녀의 사랑을 얻기 위해 애썼지만 헛수고였다.
1/1,0002.wav 용돈을 아껴 써라.
1/1,0003.wav 그는 아내를 많이 아낀다.
1/1,0004.wav 그에 전화번호 알아?
1/1,0005.wav 차에 대해 잘 아세요?
1/1,0006.wav 거기 도착하면 나한테 알려 줘.
1/1,0007.wav 그들은 내가 사랑에 빠졌다고 알려 왔다.
1/1,0008.wav 나는 상대으면서 강기를 얻은 적이 한 번도 없다.
1/1,0009.wav 사랑을 인심하게 몸살을 앓았어요.
1/1,0010.wav 요즘 컴퓨터가 안돼요.
1/1,0011.wav 장거리 절 안돼서 우울해.
1/1,0012.wav 아무 것도 먹지 않으면 죽게 되어 있다.
1/1,0013.wav 그녀는 이유를 물지 않고 돈을 벌려 주었다.
1/1,0014.wav 내가 아기를 안고 있개.
1/1,0015.wav 엄마는 말없이 나를 꼭 안아 주셨다.
1/1,0016.wav 자본에 있으세요.
1/1,0017.wav 한 여자가 내 옆에 앉았다.
1/1,0018.wav 114에 전화를 해서 번호를 알아보시지 그래세요?
1/1,0019.wav 내가 누군지 알아보겠나?
1/1,0020.wav 그는 절대 다른 사람의 의견을 받아들이지 않는다.
1/1,0021.wav 아이들도 언젠가는 그녀를 엄마로 받아들일 거야.
1/1,0022.wav 어디에서 영어를 배우셨어요?
1/1,0023.wav 동료에게서 수화를 배우고 있습니다
Ln 1, Col 13 100% Windows (CRLF) UTF-8

```

- 바꿀 내용 : | 입력 → 모두 바꾸기(A) → X



- 메모장 저장
 - 파일명 : `kss_script_1.txt`
- 엑셀 데이터 필터 단계부터 2/2 3/3 4/4 반복하기
 - 새로운 sheet에 붙여넣기 한 후 첫 번째 행이 1/1인지 아닌지 잘 확인하기. 1/1이면 삭제하기
- 최종 생성 파일 : 4개
 - `kss_script_1.txt`
 - `kss_script_2.txt`
 - `kss_script_3.txt`
 - `kss_script_4.txt`

④ ubuntu 서버로 편집한 transcript 이동

- 경로 : TTS/vits/filelists/
- 결과

- TTS/vits/filelists/ `kss_script_1.txt`
- TTS/vits/filelists/ `kss_script_2.txt`
- TTS/vits/filelists/ `kss_script_3.txt`
- TTS/vits/filelists/ `kss_script_4.txt`

```
(base) 192.168.0.52:~/TTSTEST/TTS/vits/filelists$ ls
kss_script_1.txt          ljs_audio_text_val_filelist.txt.cleaned
kss_script_2.txt          transcript.v.1.4.txt
kss_script_3.txt          vctk_audio_sid_text_test_filelist.txt
kss_script_4.txt          vctk_audio_sid_text_test_filelist.txt.cleaned
ljs_audio_text_test_filelist.txt    vctk_audio_sid_text_train_filelist.txt
ljs_audio_text_train_filelist.txt.cleaned  vctk_audio_sid_text_train_filelist.txt.cleaned
ljs_audio_text_train_filelist.txt    vctk_audio_sid_text_val_filelist.txt
ljs_audio_text_train_filelist.txt.cleaned  vctk_audio_sid_text_val_filelist.txt.cleaned
ljs_audio_text_val_filelist.txt
```

| Name | Last Modified | File size |
|---|---------------|-----------|
| ... | ... | ... |
| <code>kss_script_1.txt</code> | 12분 전 | 62.6 kB |
| <code>kss_script_2.txt</code> | 12분 전 | 73.3 kB |
| <code>kss_script_3.txt</code> | 12분 전 | 309 kB |
| <code>kss_script_4.txt</code> | 12분 전 | 1.42 MB |
| <code>ljs_audio_text_test_filelist.txt</code> | 한 시간 전 | 61.3 kB |
| <code>ljs_audio_text_test_filelist.txt.cleaned</code> | 한 시간 전 | 3.4 kB |
| <code>ljs_audio_text_train_filelist.txt</code> | 한 시간 전 | 1.54 MB |
| <code>ljs_audio_text_train_filelist.txt.cleaned</code> | 한 시간 전 | 2.22 MB |
| <code>ljs_audio_text_val_filelist.txt</code> | 한 시간 전 | 12 kB |
| <code>ljs_audio_text_val_filelist.txt.cleaned</code> | 한 시간 전 | 17.3 kB |
| <code>transcript.v.1.4.txt</code> | 4년 전 | 3.3 MB |
| <code>vctk_audio_sid_text_test_filelist.txt</code> | 한 시간 전 | 34.3 kB |
| <code>vctk_audio_sid_text_test_filelist.txt.cleaned</code> | 한 시간 전 | 44.9 kB |
| <code>vctk_audio_sid_text_train_filelist.txt</code> | 한 시간 전 | 2.96 MB |
| <code>vctk_audio_sid_text_train_filelist.txt.cleaned</code> | 한 시간 전 | 3.86 MB |
| <code>vctk_audio_sid_text_val_filelist.txt</code> | 한 시간 전 | 6.62 kB |
| <code>vctk_audio_sid_text_val_filelist.txt.cleaned</code> | 한 시간 전 | 8.62 kB |

⑤ preprocessing

가상환경 활성화

```
source activate tts
```

- 이미 활성화되어 있다면 생략

preprocess.py 경로로 이동

- 경로 : TTS/vits/
- preprocess.py 파일 확인

```
ls
```

```
(base) ljhpc1004@S220:~/TTSTEST/TTS/vits$ ls
1 LICENSE           configs      kss                   modules.py      resources    transforms.py
2 README.md         data_utils.py losses.py      monotonic_align   text        utils.py
3 attentions.py     filelists    mel_processing.py preprocess.py  train.py
4 commons.py        inference.ipynb models.py     requirements.txt train_ms.py
```

preprocess scripts

```
python preprocess.py --text_index 1 --filelists filelists/kss_script_1.txt filelists/kss_script_2.txt filelists/kss_script_3.txt filelists/kss_script_4.txt
```

```
(tts) ljhpc1004@S220:~/TTSTEST/TTS/vits$ python preprocess.py --text_index 1 --filelists filelists/kss_script_1.txt filelists/kss_script_2.txt filelists/kss_script_3.txt filelists/kss_script_4.txt
START: filelists/kss_script_1.txt
START: filelists/kss_script_2.txt
START: filelists/kss_script_3.txt
START: filelists/kss_script_4.txt
```

- 생성되는 파일

- TTS/vits/filelists/`kss_script_1.txt.cleaned`
- TTS/vits/filelists/`kss_script_2.txt.cleaned`
- TTS/vits/filelists/`kss_script_3.txt.cleaned`
- TTS/vits/filelists/`kss_script_4.txt.cleaned`

```
(tts) ljhpc1004@S220:~/TTSTEST/TTS/vits/filelists$ ls
kss_script_1.txt           kss_script_4.txt.cleaned          transcript.v1.4.txt
kss_script_1.txt.cleaned    kss_script_4.txt.list            vctk_audio_sid_text_test_filelist.txt
kss_script_2.txt           kss_script_4.txt.list.cleaned    vctk_audio_sid_text_test_filelist.txt.cleaned
kss_script_2.txt.cleaned    ljs_audio_text_test_filelist.txt  vctk_audio_sid_text_train_filelist.txt
kss_script_3.txt           ljs_audio_text_train_filelist.txt vctk_audio_sid_text_train_filelist.txt.cleaned
kss_script_3.txt.cleaned    ljs_audio_text_val_filelist.txt  vctk_audio_sid_text_val_filelist.txt
kss_script_4.txt           ljs_audio_text_val_filelist.txt.cleaned vctk_audio_sid_text_val_filelist.txt.cleaned
```

4 config 설정

① korean cleaner 모듈 생성

jupyter notebook interface에서 생성하는 것을 권장

경로 이동

- 경로 : TTS/vits/text/

korean.py 파일 생성

- TTS/vits/text/korean.py

```
import re
from jamo import h2j, j2hcj
import ko_pron

# This is a list of Korean classifiers preceded by pure Korean numerals.
_korean_classifiers = '군데 권 개 그루 님 대 두 마리 모 모금 못 발 발짝 방 번 별 보루 살 수 술 시 쌈 을
```

```

# List of (hangul, hangul divided) pairs:
_hangul_divided = [(re.compile('%s' % x[0]), x[1]) for x in [
    ('ㄱ', 'ㄱ'),
    ('ㄴ', 'ㄴ'),
    ('ㄷ', 'ㄷ'),
    ('ㄹ', 'ㄹ'),
    ('ㅁ', 'ㅁ'),
    ('ㅂ', 'ㅂ'),
    ('ㅅ', 'ㅅ'),
    ('ㅇ', 'ㅇ'),
    ('ㅈ', 'ㅈ'),
    ('ㅊ', 'ㅊ'),
    ('ㅋ', 'ㅋ'),
    ('ㅌ', 'ㅌ'),
    ('ㅍ', 'ㅍ'),
    ('ㅎ', 'ㅎ'),
    ('ㅏ', 'ㅏ'),
    ('ㅑ', 'ㅑ'),
    ('ㅓ', 'ㅓ'),
    ('ㅕ', 'ㅕ'),
    ('ㅗ', 'ㅗ'),
    ('ㅘ', 'ㅘ'),
    ('ㅓ', 'ㅓ'),
    ('ㅕ', 'ㅕ'),
    ('ㅜ', 'ㅜ'),
    ('ㅛ', 'ㅛ'),
    ('ㅡ', 'ㅡ'),
    ('ㅣ', 'ㅣ'),
    ('ㅔ', 'ㅔ'),
    ('ㅖ', 'ㅖ'),
    ('ㅚ', 'ㅚ'),
    ('ㅟ', 'ㅟ'),
    ('ㅢ', 'ㅢ'),
    ('ㅤ', 'ㅤ'),
    ('ㅦ', 'ㅦ'),
    ('ㅪ', 'ㅪ')
]]]

# List of (Latin alphabet, hangul) pairs:
_latin_to_hangul = [(re.compile('(%s)' % x[0], re.IGNORECASE), x[1]) for x in [
    ('a', '애이'),
    ('b', '비'),
    ('c', '시'),
    ('d', '디'),
    ('e', '이'),
    ('f', '에프'),
    ('g', '지'),
    ('h', '에이치'),
    ('i', '아이'),
    ('j', '제이'),
    ('k', '케이'),
    ('l', '엘'),
    ('m', '엠'),
    ('n', '엔'),
    ('o', '오'),
    ('p', '피'),
    ('q', '큐'),
    ('r', '아르'),
    ('s', '에스'),
    ('t', '티'),
    ('u', '유'),
    ('v', '브이'),
    ('w', '더블유'),
    ('x', '엑스'),
    ('y', '와이'),
    ('z', '제트')
]]]

```

```

# List of (ipa, lazy ipa) pairs:
_ipa_to_lazy_ipa = [(re.compile('%s' % x[0], re.IGNORECASE), x[1]) for x in [
    ('t̪', 'tʃ'),
    ('d̪z', 'dʒ'),
    ('n̪', 'n^'),
    ('š', 'ʃ'),
    ('w̪', 'w'),
    ('l̪', 'l`'),
    ('r̪', 'r'),
    ('y̪', 'ŋ'),
    ('ψ', 'w'),
    ('j̪', 'j'),
    ('ʌ', 'ə'),
    ('g̪', 'g'),
    ('\u031a', '#'),
    ('\u0348', '='),
    ('\u031e', ''),
    ('\u0320', ''),
    ('\u0339', '')
]]
]]
```

```

def latin_to_hangul(text):
    for regex, replacement in _latin_to_hangul:
        text = re.sub(regex, replacement, text)
    return text
```

```

def divide_hangul(text):
    text = j2hcj(h2j(text))
    for regex, replacement in _hangul_divided:
        text = re.sub(regex, replacement, text)
    return text
```

```

def hangul_number(num, sino=True):
    '''Reference https://github.com/Kyubyong/g2pK'''
    num = re.sub(',', '', num)

    if num == '0':
        return '영'
    if not sino and num == '20':
        return '스무'

    digits = '123456789'
    names = '일이삼사오육칠팔구'
    digit2name = {d: n for d, n in zip(digits, names)}

    modifiers = '한 두 세 네 다섯 여섯 일곱 여덟 아홉'
    decimals = '열 스물 서른 마흔 쉰 예순 일흔 여든 아흔'
    digit2mod = {d: mod for d, mod in zip(digits, modifiers.split())}
    digit2dec = {d: dec for d, dec in zip(digits, decimals.split())}

    spelledout = []
    for i, digit in enumerate(num):
        i = len(num) - i - 1
        if sino:
            if i == 0:

```

```

        name = digit2name.get(digit, '')
    elif i == 1:
        name = digit2name.get(digit, '') + '십'
        name = name.replace('일십', '십')
    else:
        if i == 0:
            name = digit2mod.get(digit, '')
        elif i == 1:
            name = digit2dec.get(digit, '')
    if digit == '0':
        if i % 4 == 0:
            last_three = spelledout[-min(3, len(spelledout)):]
            if ''.join(last_three) == '':
                spelledout.append('')
                continue
            else:
                spelledout.append('')
                continue
        if i == 2:
            name = digit2name.get(digit, '') + '백'
            name = name.replace('일백', '백')
        elif i == 3:
            name = digit2name.get(digit, '') + '천'
            name = name.replace('일천', '천')
        elif i == 4:
            name = digit2name.get(digit, '') + '만'
            name = name.replace('일만', '만')
        elif i == 5:
            name = digit2name.get(digit, '') + '십'
            name = name.replace('일십', '십')
        elif i == 6:
            name = digit2name.get(digit, '') + '백'
            name = name.replace('일백', '백')
        elif i == 7:
            name = digit2name.get(digit, '') + '천'
            name = name.replace('일천', '천')
        elif i == 8:
            name = digit2name.get(digit, '') + '억'
        elif i == 9:
            name = digit2name.get(digit, '') + '십'
        elif i == 10:
            name = digit2name.get(digit, '') + '백'
        elif i == 11:
            name = digit2name.get(digit, '') + '천'
        elif i == 12:
            name = digit2name.get(digit, '') + '조'
        elif i == 13:
            name = digit2name.get(digit, '') + '십'
        elif i == 14:
            name = digit2name.get(digit, '') + '백'
        elif i == 15:
            name = digit2name.get(digit, '') + '천'
    spelledout.append(name)
    return ''.join(elem for elem in spelledout)

def number_to_hangul(text):
    '''Reference https://github.com/Kyubyong/g2pK'''
```

```

tokens = set(re.findall(r'(\d[\d,]*)([\uac00-\ud71f]+)', text))
for token in tokens:
    num, classifier = token
    if classifier[:2] in _korean_classifiers or classifier[0] in _korean_classifiers:
        spelledout = hangul_number(num, sino=False)
    else:
        spelledout = hangul_number(num, sino=True)
    text = text.replace(f'{num}{classifier}', f'{spelledout}{classifier}')
# digit by digit for remaining digits
digits = '0123456789'
names = '영일이삼사오육칠팔구'
for d, n in zip(digits, names):
    text = text.replace(d, n)
return text

def korean_to_lazy_ipa(text):
    text = latin_to_hangul(text)
    text = number_to_hangul(text)
    text=re.sub('([\uac00-\ud71f]+)',lambda x:ko_pron.romanise(x.group(0),'ipa').split(']') ~
    for regex, replacement in _ipa_to_lazy_ipa:
        text = re.sub(regex, replacement, text)
    return text

def korean_to_ipa(text):
    text = korean_to_lazy_ipa(text)
    return text.replace('tʃ', 'tʃ').replace('dʒ', 'dʒ')

```

② cleaners.py 수정

| jupyter notebook interface에서 수정하는 것을 권장

경로 확인

- 경로 : TTS/vits/text/
- cleaners.py 파일 클릭

cleaners.py 수정

▼ 전체 코드

```

""" from https://github.com/keithito/tacotron """

...
Cleaners are transformations that run over the input text at both training and eval time.

Cleaners can be selected by passing a comma-delimited list of cleaner names as the "cleaners"
hyperparameter. Some cleaners are English-specific. You'll typically want to use:
1. "english_cleaners" for English text
2. "transliteration_cleaners" for non-English text that can be transliterated to ASCII using
   the Unidecode library (https://pypi.python.org/pypi/Unidecode)
3. "basic_cleaners" if you do not want to transliterate (in this case, you should also
   use the symbols in symbols.py to match your data).
...

import re
from unidecode import unidecode

```

```

from phonemizer import phonemize

# Korean Cleaner
from text.korean import latin_to_hangul, number_to_hangul, divide_hangul


# Regular expression matching whitespace:
_whitespace_re = re.compile(r'\s+')

# List of (regular expression, replacement) pairs for abbreviations:
_abbreviations = [(re.compile('\b%s\b' % x[0], re.IGNORECASE), x[1]) for x in [
    ('mrs', 'missess'),
    ('mr', 'mister'),
    ('dr', 'doctor'),
    ('st', 'saint'),
    ('co', 'company'),
    ('jr', 'junior'),
    ('maj', 'major'),
    ('gen', 'general'),
    ('drs', 'doctors'),
    ('rev', 'reverend'),
    ('lt', 'lieutenant'),
    ('hon', 'honorable'),
    ('sgt', 'sergeant'),
    ('capt', 'captain'),
    ('esq', 'esquire'),
    ('ltd', 'limited'),
    ('col', 'colonel'),
    ('ft', 'fort'),
]]
]]
```

```

def expand_abbreviations(text):
    for regex, replacement in _abbreviations:
        text = re.sub(regex, replacement, text)
    return text
```

```

def expand_numbers(text):
    return normalize_numbers(text)
```

```

def lowercase(text):
    return text.lower()
```

```

def collapse_whitespace(text):
    return re.sub(_whitespace_re, ' ', text)
```

```

def convert_to_ascii(text):
    return unidecode(text)
```

```

def basic_cleaners(text):
    '''Basic pipeline that lowercases and collapses whitespace without transliteration.'''
    text = lowercase(text)
    text = collapse_whitespace(text)
```

```

    return text

def transliteration_cleaners(text):
    '''Pipeline for non-English text that transliterates to ASCII.'''
    text = convert_to_ascii(text)
    text = lowercase(text)
    text = collapse_whitespace(text)
    return text

# Korean Cleaners
def korean_cleaners(text):
    '''Pipeline for Korean text'''
    text = latin_to_hangul(text)
    text = number_to_hangul(text)
    text = divide_hangul(text)
    text = re.sub(r'([\u3131-\u3163])$', r'\1.', text)
    return text

def english_cleaners(text):
    '''Pipeline for English text, including abbreviation expansion.'''
    text = convert_to_ascii(text)
    text = lowercase(text)
    text = expand_abbreviations(text)
    phonemes = phonemize(text, language='en-us', backend='espeak', strip=True)
    phonemes = collapse_whitespace(phonemes)
    return phonemes

def english_cleaners2(text):
    '''Pipeline for English text, including abbreviation expansion. + punctuation + stress'''
    text = convert_to_ascii(text)
    text = lowercase(text)
    text = expand_abbreviations(text)
    phonemes = phonemize(text, language='en-us', backend='espeak', strip=True, preserve_punc=True)
    phonemes = collapse_whitespace(phonemes)
    return phonemes

```

- korean 모듈에서 함수 호출

```
from text.korean import latin_to_hangul, number_to_hangul, divide_hangul
```

- korean_cleaners 함수 생성

```

def korean_cleaners(text):
    '''Pipeline for Korean text'''
    text = latin_to_hangul(text)
    text = number_to_hangul(text)
    text = divide_hangul(text)
    text = re.sub(r'([\u3131-\u3163])$', r'\1.', text)
    return text

```

③ config 파일 생성

경로 이동

- 경로 : TTS/vits/configs/

```
(tts) ljhpc1004@S220:~/TTSTEST/TTS/vits$ cd configs/
(tts) ljhpc1004@S220:~/TTSTEST/TTS/vits/configs$ ls
ljs_base.json  ljs_nosdp.json  vctk_base.json
```

.json 파일 복제

```
cp ljs_nosdp.json kss.json
```

```
(tts) ljhpc1004@S220:~/TTSTEST/TTS/vits/configs$ cp ljs_nosdp.json kss.json
(tts) ljhpc1004@S220:~/TTSTEST/TTS/vits/configs$ ls
kss.json  ljs_base.json  ljs_nosdp.json  vctk_base.json
```

kss.json 파일 수정

jupyter notebook interface에서 수정하는 것을 권장

- jupyter notebook interface에서 수정
 - TTS/vits/configs/ 디렉토리 이동 후 kss.json 파일 클릭하여 수정 후 저장
- terminal에서 수정

```
vi kss.json
```

◦ i : insert mode

- 수정 사항 : “train” 항목
 - epochs**

✖ TROUBLE SHOOTING ✖ CUDA out of memory

[설정 전]

20000,

[설정 후]

1,

◦ **batch_size**

✖ TROUBLE SHOOTING ✖ CUDA out of memory

[설정 전]

64,

[설정 후]

16,

- 수정 사항 : “data” 항목

◦ **training_files**

[설정 전]

“filelists/ljs_audio_text_train_filelist.txt.cleaned”,

[설정 후]

“filelists/kss_script_1.txt.cleaned”,

◦ **validation_files**

[설정 전]

“filelists/ljs_audio_text_val_filelist.txt.cleaned”,

[설정 후]

“filelists/kss_script_2.txt.cleaned”,

◦ **text_cleaners**

[설정 전]

[설정 후]

["english_cleaners2"], ["korean_cleaners"],

◦ **sampling_rate**

[설정 전]

22050,

[설정 후]

44100,



◦ **n_speakers**

[설정 전]

0

[설정 후]

1

④⑤

5 Train Model

① 경로 이동

- train.py 파일이 있는 곳으로 이동
- 경로 : TTS/vits/

② training

```
python train.py -c configs/kss.json -m kss
```

```
(tts) ljhlp1004@S220:~/TTSTEST/TTS/vits$ python train.py -c configs/kss.json -m kss
[INFO] {'train': {'log_interval': 200, 'eval_interval': 1000, 'seed': 1234, 'epochs': 20000, 'learning_rate': 0.0002, 'beta_s': [0.8, 0.99], 'eps': 1e-09, 'batch_size': 64, 'fp16_run': True, 'lr_decay': 0.999875, 'segment_size': 8192, 'init_lr_ratio': 1, 'warmup_epochs': 0, 'c_mel': 45, 'c_kl': 1.0}, 'data': {'training_files': 'filelists/kss_script_1.txt.cleaned', 'validation_files': 'filelists/kss_script_2.txt.cleaned', 'text_cleaners': ['korean_cleaners'], 'max_wav_value': 32768.0, 'sampling_rate': 22050, 'filter_length': 1024, 'hop_length': 256, 'win_length': 1024, 'n_mel_channels': 80, 'mel_fmin': 0.0, 'mel_fmax': None, 'add_blank': True, 'n_speakers': 0, 'cleaned_text': True}, 'model': {'inter_channels': 192, 'hidden_channel_s': 192, 'filter_channels': 768, 'n_heads': 2, 'n_layers': 6, 'kernel_size': 3, 'p_dropout': 0.1, 'resblock': '1', 'resblock_dilation_sizes': [[1, 3, 5], [1, 3, 5], [1, 3, 5]], 'upsample_rates': [8, 8, 2, 2], 'upsample_initial_channel': 512, 'upsample_kernel_sizes': [16, 16, 4, 4], 'n_layers_q': 3, 'use_spectral_norm': False, 'use_sd': False}, 'model_dir': './logs/kss'}
/home/ljhlp1004/TTSTEST/TTS/vits/utils.py:134: WavFileWarning: Chunk (non-data) not understood, skipping it.
    sampling_rate, data = read(full_path)
/home/ljhlp1004/TTSTEST/TTS/vits/utils.py:134: WavFileWarning: Chunk (non-data) not understood, skipping it.
    sampling_rate, data = read(full_path)
/home/ljhlp1004/TTSTEST/TTS/vits/utils.py:134: WavFileWarning: Chunk (non-data) not understood, skipping it.
    sampling_rate, data = read(full_path)
/home/ljhlp1004/TTSTEST/TTS/vits/utils.py:134: WavFileWarning: Chunk (non-data) not understood, skipping it.
    sampling_rate, data = read(full_path)
/home/ljhlp1004/TTSTEST/TTS/vits/utils.py:134: WavFileWarning: Chunk (non-data) not understood, skipping it.
    sampling_rate, data = read(full_path)
/home/ljhlp1004/TTSTEST/TTS/vits/utils.py:134: WavFileWarning: Chunk (non-data) not understood, skipping it.
    sampling_rate, data = read(full_path)
/home/ljhlp1004/TTSTEST/TTS/vits/utils.py:134: WavFileWarning: Chunk (non-data) not understood, skipping it.
    sampling_rate, data = read(full_path)
/home/ljhlp1004/TTSTEST/TTS/vits/utils.py:134: WavFileWarning: Chunk (non-data) not understood, skipping it.
    sampling_rate, data = read(full_path)
/home/ljhlp1004/TTSTEST/TTS/vits/utils.py:134: WavFileWarning: Chunk (non-data) not understood, skipping it.
    sampling_rate, data = read(full_path)
/home/ljhlp1004/TTSTEST/TTS/vits/utils.py:134: WavFileWarning: Chunk (non-data) not understood, skipping it.
    sampling_rate, data = read(full_path)
/home/ljhlp1004/TTSTEST/TTS/vits/utils.py:134: WavFileWarning: Chunk (non-data) not understood, skipping it.
    sampling_rate, data = read(full_path)
/home/ljhlp1004/TTSTEST/TTS/vits/utils.py:134: WavFileWarning: Chunk (non-data) not understood, skipping it.
    sampling_rate, data = read(full_path)
```

✖ ERROR ✖

RuntimeError: **CUDA out of memory**. Tried to allocate 20.00 MiB (GPU 0; 5.79 GiB total capacity; 4.09 GiB already allocated; 21.06 MiB free; 4.16 GiB reserved in total by PyTorch)

▼ 현재 CUDA MEMORY : 6GB 밖에 할당이 안되어 있는 상태

```
nvidia-smi
```

```
(tts) ljhpi004@S220:~/TTSTEST/TTS/vits$ nvidia-smi
Wed Feb  7 01:54:29 2024
+-----+
| NVIDIA-SMI 450.248.02 Driver Version: 450.248.02 CUDA Version: 11.0 |
+-----+
| GPU Name Persistence-M Bus-Id Disp.A Volatile Uncorr. ECC |
| Fan Temp Perf Pwr:Usage/Cap| Memory-Usage | GPU-Util Compute M. |
|                            |             |             | MIG M. |
+-----+
| 0 GeForce RTX 2060 Off 00000000:01:00.0 On N/A |
| N/A 39C P8 9W / N/A 736MiB / 5926MiB 6% Default N/A |
+-----+
+-----+
| Processes:
| GPU GI CI PID Type Process name GPU Memory Usage |
| ID ID          |
+-----+
| 0 N/A N/A 941 G /usr/lib/xorg/Xorg 35MiB |
| 0 N/A N/A 8227 G /usr/lib/xorg/Xorg 35MiB |
| 0 N/A N/A 9541 G /usr/lib/xorg/Xorg 167MiB |
| 0 N/A N/A 9671 G /usr/bin/gnome-shell 165MiB |
| 0 N/A N/A 17332 G /usr/lib/firefox/firefox 3MiB |
| 0 N/A N/A 20252 G ....variations-seed-version 201MiB |
+-----+
```

```
Traceback (most recent call last):
  File "train.py", line 290, in <module>
    main()
  File "train.py", line 50, in main
    mp.spawn(run, nprocs=n_gpus, args=(n_gpus, hps,))
  File "/home/ljhpi004/anaconda3/envs/tts/lib/python3.7/site-packages/torch/multiprocessing/spawn.py", line 200, in spawn
    return start_processes(fn, args, nprocs, join, daemon, start_method='spawn')
  File "/home/ljhpi004/anaconda3/envs/tts/lib/python3.7/site-packages/torch/multiprocessing/spawn.py", line 158, in start_processes
    while not context.join():
  File "/home/ljhpi004/anaconda3/envs/tts/lib/python3.7/site-packages/torch/multiprocessing/spawn.py", line 119, in join
    raise Exception(msg)
Exception:

-- Process 0 terminated with the following error:
Traceback (most recent call last):
  File "/home/ljhpi004/anaconda3/envs/tts/lib/python3.7/site-packages/torch/multiprocessing/spawn.py", line 20, in _wrap
    fn(i, *args)
  File "/home/ljhpi004/TTSTEST/TTS/vits/train.py", line 117, in run
    train_and_evaluate(rank, epoch, hps, [net_g, net_d], [optim_g, optim_d], [scheduler_g, scheduler_d], scaler, [train_loader, eval_loader], logger, [writer, writer_eval])
  File "/home/ljhpi004/TTSTEST/TTS/vits/train.py", line 168, in train_and_evaluate
    y_d_hat_r, y_d_hat_g, _, _ = net_d(y, y_hat.detach())
  File "/home/ljhpi004/anaconda3/envs/tts/lib/python3.7/site-packages/torch/nn/modules/module.py", line 722, in _call_impl
    result = self.forward(*input, **kwargs)
  File "/home/ljhpi004/anaconda3/envs/tts/lib/python3.7/site-packages/torch/nn/parallel/distributed.py", line 511, in forward
    output = self.module(*inputs[0], **kwargs[0])
  File "/home/ljhpi004/anaconda3/envs/tts/lib/python3.7/site-packages/torch/nn/modules/module.py", line 722, in _call_impl
    result = self.forward(*input, **kwargs)
  File "/home/ljhpi004/TTSTEST/TTS/vits/models.py", line 380, in forward
    y_d_g, fmap_g = d(y_hat)
  File "/home/ljhpi004/anaconda3/envs/tts/lib/python3.7/site-packages/torch/nn/modules/module.py", line 722, in _call_impl
    result = self.forward(*input, **kwargs)
  File "/home/ljhpi004/TTSTEST/TTS/vits/models.py", line 326, in forward
    x = l(x)
  File "/home/ljhpi004/anaconda3/envs/tts/lib/python3.7/site-packages/torch/nn/modules/module.py", line 722, in _call_impl
    result = self.forward(*input, **kwargs)
  File "/home/ljhpi004/anaconda3/envs/tts/lib/python3.7/site-packages/torch/nn/modules/conv.py", line 419, in forward
    return self._conv_forward(input, self.weight)
  File "/home/ljhpi004/anaconda3/envs/tts/lib/python3.7/site-packages/torch/nn/modules/conv.py", line 416, in _conv_forward
    self.padding, self.dilation, self.groups)
RuntimeError: CUDA out of memory. Tried to allocate 20.00 MiB (GPU 0; 5.79 GiB total capacity; 4.09 GiB already allocated;
21.06 MiB free; 4.16 GiB reserved in total by PyTorch)
```

③

⑥⑦⑧