```
gpu_info = !nvidia-smi
gpu_info = '₩n'.join(gpu_info)
if gpu_info.find('failed') >= 0:
  print('Not connected to a GPU')
else:
  print(gpu_info)
```

```
Fri Nov  8 04:35:58 2024
+---------------------------------------------------------------------------------------+
| NVIDIA-SMI 535.104.05             Driver Version: 535.104.05   CUDA Version: 12.2     |
|-----------------------------------------+----------------------+----------------------+
| 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  Tesla T4                       Off | 00000000:00:04.0 Off |                    0 |
| N/A   43C    P8               9W /  70W |      0MiB / 15360MiB |      0%      Default |
|                                         |                      |                  N/A |
+-----------------------------------------+----------------------+----------------------+

+---------------------------------------------------------------------------------------+
| Processes:                                                                            |
|  GPU   GI   CI        PID   Type   Process name                            GPU Memory |
|        ID   ID                                                             Usage      |
|=======================================================================================|
|  No running processes found                                                           |
+---------------------------------------------------------------------------------------+
```

```
from psutil import virtual_memory
ram_gb = virtual_memory().total / 1e9
print('Your runtime has {:.1f} gigabytes of available RAM₩n'.format(ram_gb))

if ram_gb < 20:
  print('Not using a high-RAM runtime')
else:
  print('You are using a high-RAM runtime!')
```

```
Your runtime has 54.8 gigabytes of available RAM

You are using a high-RAM runtime!
```

```
import os
import sys

os.chdir("/content/drive/MyDrive/archive_FastSpeech2")
!git clone https://github.com/HGU-DLLAB/Korean-FastSpeech2-Pytorch.git
sys.path.append('/content/drive/MyDrive/archive_FastSpeech2')
os.chdir("/content")


# requirements.txt의 일부 패키지 버전을 최신 버전으로 설치
!pip install librosa matplotlib g2p-en inflect numba pypinyin pyworld scikit-learn scipy soundfile tensorboard tgt tqdm unidecode
```

```
fatal: destination path 'Korean-FastSpeech2-Pytorch' already exists and is not an empty directory.
Requirement already satisfied: librosa in /usr/local/lib/python3.10/dist-packages (0.10.2.post1)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (3.8.0)
Collecting g2p-en
  Downloading g2p_en-2.1.0-py3-none-any.whl.metadata (4.5 kB)
Requirement already satisfied: inflect in /usr/local/lib/python3.10/dist-packages (7.4.0)
Requirement already satisfied: numba in /usr/local/lib/python3.10/dist-packages (0.60.0)
Collecting pypinyin
  Downloading pypinyin-0.53.0-py2.py3-none-any.whl.metadata (12 kB)
Collecting pyworld
  Downloading pyworld-0.3.4.tar.gz (251 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 252.0/252.0 kB 4.8 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (1.5.2)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (1.13.1)
Requirement already satisfied: soundfile in /usr/local/lib/python3.10/dist-packages (0.12.1)
Requirement already satisfied: tensorboard in /usr/local/lib/python3.10/dist-packages (2.17.0)
Collecting tgt
  Downloading tgt-1.5-py3-none-any.whl.metadata (764 bytes)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (4.66.6)
Collecting unidecode
  Downloading Unidecode-1.3.8-py3-none-any.whl.metadata (13 kB)
Requirement already satisfied: audioread>=2.1.9 in /usr/local/lib/python3.10/dist-packages (from librosa) (3.0.1)
Requirement already satisfied: numpy!=1.22.0,!=1.22.1,!=1.22.2,>=1.20.3 in /usr/local/lib/python3.10/dist-packages (from librosa) (1.26.4)
Requirement already satisfied: joblib>=0.14 in /usr/local/lib/python3.10/dist-packages (from librosa) (1.4.2)
Requirement already satisfied: decorator>=4.3.0 in /usr/local/lib/python3.10/dist-packages (from librosa) (4.4.2)
Requirement already satisfied: pooch>=1.1 in /usr/local/lib/python3.10/dist-packages (from librosa) (1.8.2)
Requirement already satisfied: soxr>=0.3.2 in /usr/local/lib/python3.10/dist-packages (from librosa) (0.5.0.post1)
Requirement already satisfied: typing-extensions>=4.1.1 in /usr/local/lib/python3.10/dist-packages (from librosa) (4.12.2)
Requirement already satisfied: lazy-loader>=0.1 in /usr/local/lib/python3.10/dist-packages (from librosa) (0.4)
Requirement already satisfied: msgpack>=1.0 in /usr/local/lib/python3.10/dist-packages (from librosa) (1.1.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.3.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (4.54.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.4.7)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (24.1)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (3.2.0)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (2.8.2)
Requirement already satisfied: nltk>=3.2.4 in /usr/local/lib/python3.10/dist-packages (from g2p-en) (3.8.1)
Collecting distance>=0.1.3 (from g2p-en)
  Downloading Distance-0.1.3.tar.gz (180 kB)
     ──────────────────────────────────────────────────────────────────────────────── 180.3/180.3 kB 14.8 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: more-itertools>=8.5.0 in /usr/local/lib/python3.10/dist-packages (from inflect) (10.5.0)
Requirement already satisfied: typeguard>=4.0.1 in /usr/local/lib/python3.10/dist-packages (from inflect) (4.4.0)
Requirement already satisfied: llvmlite<0.44,>=0.43.0dev0 in /usr/local/lib/python3.10/dist-packages (from numba) (0.43.0)
Requirement already satisfied: cython>=0.24 in /usr/local/lib/python3.10/dist-packages (from pyworld) (3.0.11)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (3.5.0)
Requirement already satisfied: cffi>=1.0 in /usr/local/lib/python3.10/dist-packages (from soundfile) (1.17.1)
Requirement already satisfied: absl-py>=0.4 in /usr/local/lib/python3.10/dist-packages (from tensorboard) (1.4.0)
Requirement already satisfied: grpcio>=1.48.2 in /usr/local/lib/python3.10/dist-packages (from tensorboard) (1.64.1)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.10/dist-packages (from tensorboard) (3.7)
Requirement already satisfied: protobuf!=4.24.0,<5.0.0,>=3.19.6 in /usr/local/lib/python3.10/dist-packages (from tensorboard) (3.20.3)
```

```
!pip install tensorflow tensorboardX pillow pyyaml torchaudio librosa matplotlib numpy scipy tqdm soundfile
```

```
Requirement already satisfied: tensorflow in /usr/local/lib/python3.10/dist-packages (2.17.0)
Collecting tensorboardX
  Downloading tensorboardX-2.6.2.2-py2.py3-none-any.whl.metadata (5.8 kB)
Requirement already satisfied: pillow in /usr/local/lib/python3.10/dist-packages (10.4.0)
Requirement already satisfied: pyyaml in /usr/local/lib/python3.10/dist-packages (6.0.2)
Requirement already satisfied: torchaudio in /usr/local/lib/python3.10/dist-packages (2.5.0+cu121)
Requirement already satisfied: librosa in /usr/local/lib/python3.10/dist-packages (0.10.2.post1)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (3.8.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (1.26.4)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (1.13.1)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (4.66.6)
Requirement already satisfied: soundfile in /usr/local/lib/python3.10/dist-packages (0.12.1)
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=24.3.25 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (24.3.25)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.6.0)
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: h5py>=3.10.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (3.12.1)
Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (18.1.1)
Requirement already satisfied: ml-dtypes<0.5.0,>=0.3.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.4.1)
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (3.4.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from tensorflow) (24.1)
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<5.0.0dev,>=3.20.3 in /usr/local/lib/python3.10/
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (2.32.3)
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from tensorflow) (75.1.0)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.16.0)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (2.5.0)
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (4.12.2)
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.16.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.64.1)
Requirement already satisfied: tensorboard<2.18,>=2.17 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (2.17.0)
Requirement already satisfied: keras>=3.2.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (3.4.1)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.37.1)
Requirement already satisfied: torch==2.5.0 in /usr/local/lib/python3.10/dist-packages (from torchaudio) (2.5.0+cu121)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from torch==2.5.0->torchaudio) (3.16.1)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch==2.5.0->torchaudio) (3.4.2)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch==2.5.0->torchaudio) (3.1.4)
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (from torch==2.5.0->torchaudio) (2024.10.0)
Requirement already satisfied: sympy==1.13.1 in /usr/local/lib/python3.10/dist-packages (from torch==2.5.0->torchaudio) (1.13.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from sympy==1.13.1->torch==2.5.0->torchaudio) (
Requirement already satisfied: audioread>=2.1.9 in /usr/local/lib/python3.10/dist-packages (from librosa) (3.0.1)
Requirement already satisfied: scikit-learn>=0.20.0 in /usr/local/lib/python3.10/dist-packages (from librosa) (1.5.2)
Requirement already satisfied: joblib>=0.14 in /usr/local/lib/python3.10/dist-packages (from librosa) (1.4.2)
Requirement already satisfied: decorator>=4.3.0 in /usr/local/lib/python3.10/dist-packages (from librosa) (4.4.2)
Requirement already satisfied: numba>=0.51.0 in /usr/local/lib/python3.10/dist-packages (from librosa) (0.60.0)
Requirement already satisfied: pooch>=1.1 in /usr/local/lib/python3.10/dist-packages (from librosa) (1.8.2)
Requirement already satisfied: soxr>=0.3.2 in /usr/local/lib/python3.10/dist-packages (from librosa) (0.5.0.post1)
Requirement already satisfied: lazy-loader>=0.1 in /usr/local/lib/python3.10/dist-packages (from librosa) (0.4)
Requirement already satisfied: msgpack>=1.0 in /usr/local/lib/python3.10/dist-packages (from librosa) (1.1.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.3.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (4.54.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.4.7)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (3.2.0)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (2.8.2)
Requirement already satisfied: cffi>=1.0 in /usr/local/lib/python3.10/dist-packages (from soundfile) (1.17.1)
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.10/dist-packages (from astunparse>=1.6.0->tensorflow) (0.44.0)
```

```
import os
import sys
import yaml


os.chdir("/content")
!git clone https://github.com/JH-lee95/Fastspeech2-Korean.git
```

```
Cloning into 'Fastspeech2-Korean'...
remote: Enumerating objects: 89, done.
remote: Counting objects: 100% (45/45), done.
remote: Compressing objects: 100% (19/19), done.
remote: Total 89 (delta 27), reused 43 (delta 26), pack-reused 44 (from 1)
Receiving objects: 100% (89/89), 581.55 KiB | 21.54 MiB/s, done.
Resolving deltas: 100% (27/27), done.
```

```
!pip install jamo
```

```
Collecting jamo
  Downloading jamo-0.4.1-py3-none-any.whl.metadata (2.3 kB)
Downloading jamo-0.4.1-py3-none-any.whl (9.5 kB)
Installing collected packages: jamo
Successfully installed jamo-0.4.1
```

```
import shutil

shutil.rmtree('/content/drive/MyDrive/archive_FastSpeech2/Korean-FastSpeech2-Pytorch/data/Dataset/kss/1')
shutil.rmtree('/content/drive/MyDrive/archive_FastSpeech2/Korean-FastSpeech2-Pytorch/data/Dataset/kss/2')
```

```
import shutil

shutil.rmtree('/content/drive/MyDrive/archive_FastSpeech2/Korean-FastSpeech2-Pytorch/ckpt')
```

```
import os
import shutil
import json
from sklearn.model_selection import train_test_split

# 원본 데이터 경로
data_dirs = [
    ('/content/drive/MyDrive/archive/1/Happy', '/content/drive/MyDrive/archive/1/Happy_L'),
    ('/content/drive/MyDrive/archive/1/Neutral', '/content/drive/MyDrive/archive/1/Neutral_L')
]

# 학습 및 검증 데이터 출력 경로
train_output_dir = '/content/drive/MyDrive/archive_FastSpeech2/Korean-FastSpeech2-Pytorch/data/Dataset/kss/2'
val_output_dir = '/content/drive/MyDrive/archive_FastSpeech2/Korean-FastSpeech2-Pytorch/data/Dataset/kss/1'
transcript_path = '/content/drive/MyDrive/archive_FastSpeech2/Korean-FastSpeech2-Pytorch/data/Dataset/transcript.v.1.4.txt'

all_files = []
file_json_map = {}

# 각 디렉토리의 wav 파일과 json 파일 매핑
for audio_dir, json_dir in data_dirs:
    wav_files = [os.path.join(audio_dir, f) for f in os.listdir(audio_dir) if f.endswith('.wav')]
    json_files = [os.path.join(json_dir, f.replace('.wav', '.json')) for f in os.listdir(audio_dir) if f.endswith('.wav')]

    all_files.extend(wav_files)
    file_json_map.update({wav: json for wav, json in zip(wav_files, json_files)})

# 데이터셋을 훈련과 검증으로 나눕니다.
train_files, val_files = train_test_split(
    all_files,
    test_size=0.1,
    random_state=42
)

# 출력 디렉토리 생성
os.makedirs(train_output_dir, exist_ok=True)
os.makedirs(val_output_dir, exist_ok=True)

with open(transcript_path, 'w', encoding='utf-8') as transcript_file:
    for file_path in all_files:
        # 파일 이동
        filename = os.path.basename(file_path)
        json_path = file_json_map[file_path]  # 각 파일마다 올바른 json 경로 참조
        folder = '2' if file_path in train_files else '1'
        destination = os.path.join(train_output_dir if folder == '2' else val_output_dir, filename)

        if not os.path.exists(destination):
            shutil.copy(file_path, destination)
        # JSON 파일에서 TransLabelText 추출 및 텍스트 파일에 작성
        if os.path.exists(json_path):
            with open(json_path, 'r', encoding='utf-8') as f:
                json_data = json.load(f)
                trans_label_text = json_data['전사정보']['TransLabelText']

                # 원하는 형식으로 텍스트 파일에 추가
                transcript_file.write(f"{folder}/{filename}|{trans_label_text}₩n")

print("데이터셋 분할 및 복사가 완료되고, transcript.v.1.4.txt에 정보가 추가되었습니다.")
```

```
데이터셋 분할 및 복사가 완료되고, transcript.v.1.4.txt에 정보가 추가되었습니다.
```

```
with open(transcript_path, 'r', encoding='utf-8') as f:
    transcript_length = len(f.readlines())
print(f"transcript.v.1.4.txt 총 줄 수: {transcript_length}")

# kss/1 및 kss/2의 wav 파일 개수 확인
val_wav_count = len([f for f in os.listdir(val_output_dir) if f.endswith('.wav')])
train_wav_count = len([f for f in os.listdir(train_output_dir) if f.endswith('.wav')])

print(f"kss/1의 wav 파일 개수: {val_wav_count}")
print(f"kss/2의 wav 파일 개수: {train_wav_count}")
```

```
transcript.v.1.4.txt 총 줄 수: 3987
kss/1의 wav 파일 개수: 399
kss/2의 wav 파일 개수: 3588
```

```
meta_name = "transcript.v.1.4.txt"
base = "/content/drive/MyDrive/archive_FastSpeech2/Korean-FastSpeech2-Pytorch/data/Dataset"

def audio_text_pair(data_dir, metadata_name):
    # 메타데이터 파일 경로 정의
    meta_path = os.path.join(data_dir, metadata_name)

    with open(meta_path, "r", encoding="utf-8") as m:
        for line in m:
            # wav 파일의 위치와 문장을 분리
            wav_path = line.split("|")[0]
            content = line.split("|")[1]

            # lab 파일 경로 생성
            text_path = wav_path.replace("wav", "lab")
            lab_file_path = os.path.join(base, text_path)

            # 해당 디렉토리가 없을 경우 생성
            os.makedirs(os.path.dirname(lab_file_path), exist_ok=True)

            # lab 파일에 문장 쓰기
            with open(lab_file_path, "w", encoding="utf-8") as t:
                t.write(content)

# 함수 실행 예시
audio_text_pair(base, meta_name)
```

```
from jamo import h2j,hangul_to_jamo,j2hcj
from g2pk import G2p
import jamotools
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-7-27b758f5a700> in <cell line: 2>()
      1 from jamo import h2j,hangul_to_jamo,j2hcj
----> 2 from g2pk import G2p
      3 import jamotools

ModuleNotFoundError: No module named 'g2pk'

---------------------------------------------------------------------------
NOTE: If your import is failing due to a missing package, you can
manually install dependencies using either !pip or !apt.

To view examples of installing some common dependencies, click the
"Open Examples" button below.
---------------------------------------------------------------------------
```

OPEN EXAMPLES

```
# 한글 문장을 초/중/종성 형태로 분리, 단독 사용 안함
g2p=G2p() #grapheme to phoneme

def jamo_split(content):

  content=g2p(content)
  jamo=h2j(content).split(" ")

  return jamo



# word_to_phoneme 딕셔너리 , lexiocn 파일을 만드는 함수
import tqdm

def make_p_dict(meta_path,position):

  p_dict={}

  with open(meta_path,"r") as f:
    for line in tqdm.tqdm(f.readlines()):
      line=line.rstrip()
      content=line.split("|")[position] #meta data 내의 텍스트가 기록된 위치
      word_list=content.split(" ")

      for idx,word in enumerate(word_list):
        print(word)
        if not word in p_dict.keys():
          p_dict[word]=" ".join(jamo_split(word)[0])


  # with open("phoneme_lexicon.txt","w") as p:
  #   for k,v in p_dict.items():
  #     p.write("{}₩t{}₩n".format(k,v))

  return p_dict
```

```
!pip install jamotools
```

```
Collecting jamotools
  Downloading jamotools-0.1.10-py2.py3-none-any.whl.metadata (8.2 kB)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from jamotools) (1.26.4)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from jamotools) (1.16.0)
Requirement already satisfied: future in /usr/local/lib/python3.10/dist-packages (from jamotools) (1.0.0)
Downloading jamotools-0.1.10-py2.py3-none-any.whl (13 kB)
Installing collected packages: jamotools
Successfully installed jamotools-0.1.10
```

```
from jamo import h2j,hangul_to_jamo,j2hcj
from g2pk import G2p
import jamotools
import re
import tqdm

g2p=G2p() #grapheme to phoneme

def jamo_split(content):

  content=g2p(content)
  jamo=h2j(content).split(" ")
```

```
        return jamo

    def make_p_dict(meta_path, position):
        p_dict = {}

        with open(meta_path, "r") as f:
            for line in tqdm.tqdm(f.readlines()):
                line = line.rstrip()
                content = line.split("|")[position]  # meta data 내의 텍스트가 기록된 위치
                word_list = content.split(" ")

                for idx, word in enumerate(word_list):
                    print(word)

                    # 원래 단어 추가
                    if word not in p_dict.keys():
                        p_dict[word] = " ".join(jamo_split(word)[0])

                    # 특수문자를 제거한 단어 추가
                    clean_word = re.sub(r"[^가-힣a-zA-Z0-9]", "", word)  # 한글, 알파벳, 숫자를 제외한 특수문자 제거
                    if clean_word and clean_word not in p_dict.keys():
                        p_dict[clean_word] = " ".join(jamo_split(clean_word)[0])

        # with open("phoneme_lexicon.txt", "w") as p:
        #     for k, v in p_dict.items():
        #         p.write("{}₩t{}₩n".format(k, v))

        return p_dict


    p_dict = make_p_dict("/content/drive/MyDrive/archive_FastSpeech2/Korean-FastSpeech2-Pytorch/data/Dataset/transcript.v.1.4.txt", 1)
```

```
  0%|          | 0/3987 [00:00<?, ?it/s]나와
같은
학생이었던
그녀가
극장의
딸이었다.
  0%|          | 2/3987 [00:00<12:11,  5.45it/s]나는
큰배추흰나비의
애벌레를
기르기
시작했습니다.
익살꾸러기와
어릿광대는
  0%|          | 3/3987 [00:00<12:44,  5.21it/s]객석으로
내려와
피노키오를
얼싸안았습니다.
그의
눈웃음에
반해
  0%|          | 4/3987 [00:00<12:53,  5.15it/s]함께
있고
싶어졌다.
핸드폰
요금
얼마
나왔지?
  0%|          | 6/3987 [00:01<12:01,  5.52it/s]가나와
맞붙어
우리나라가
이길
수
있었다.
지방선거가
  0%|          | 7/3987 [00:01<11:57,  5.55it/s]끝난
지
일주일이
지났다.
귀중한
보석이라도
  0%|          | 8/3987 [00:01<12:36,  5.26it/s]들고
가는
것
같아보였습니다.
기꺼이
태워다
  0%|          | 10/3987 [00:01<10:28,  6.33it/s]드립니다.
포르투갈의
산업
지역은
드물답니다.
끌려
  0%|          | 11/3987 [00:01<11:20,  5.84it/s]나간
선
자리에는
```

```
첫사랑의
그녀가
```

```
def make_lexicon(p_dict):
  with open("p_lexicon2.txt","w") as f:
    for k,v in p_dict.items():
      f.write("{}₩t{}₩n".format(k,v))


make_lexicon(p_dict)


import os
import sys

os.chdir("/content/drive/MyDrive/archive_FastSpeech2/Korean-FastSpeech2-Pytorch")

!python preprocess.py
```

```
스트리밍 출력 내용이 길어서 마지막 5000줄이 삭제되었습니다.
    ISFT            : Lavf58.76.100
  Stream #0:0: Audio: pcm_s16le ([1][0][0][0] / 0x0001), 22050 Hz, mono, s16, 352 kb/s
    Metadata:
      encoder         : Lavc58.134.100 pcm_s16le
size=      89kB time=00:00:02.06 bitrate= 353.2kbits/s speed= 103x
video:0kB audio:89kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: 0.085761%
ffmpeg version 4.4.2-0ubuntu0.22.04.1 Copyright (c) 2000-2021 the FFmpeg developers
  built with gcc 11 (Ubuntu 11.2.0-19ubuntu1)
  configuration: --prefix=/usr --extra-version=0ubuntu0.22.04.1 --toolchain=hardened --libdir=/usr/lib/x86_64-linux-gnu --incdir=/usr/include
  libavutil      56. 70.100 / 56. 70.100
  libavcodec     58.134.100 / 58.134.100
  libavformat    58. 76.100 / 58. 76.100
  libavdevice    58. 13.100 / 58. 13.100
  libavfilter     7.110.100 /  7.110.100
  libswscale      5.  9.100 /  5.  9.100
  libswresample   3.  9.100 /  3.  9.100
  libpostproc    55.  9.100 / 55.  9.100
Guessed Channel Layout for Input Stream #0.0 : mono
Input #0, wav, from '/content/drive/MyDrive/archive_FastSpeech2/Korean-FastSpeech2-Pytorch/data/Dataset/kss/wavs_bak/2/0050_G2A4E7S0C2_HJH_00
  Duration: 00:00:03.69, bitrate: 768 kb/s
  Stream #0:0: Audio: pcm_s16le ([1][0][0][0] / 0x0001), 48000 Hz, mono, s16, 768 kb/s
Stream mapping:
  Stream #0:0 -> #0:0 (pcm_s16le (native) -> pcm_s16le (native))
Press [q] to stop, [?] for help
Output #0, wav, to '/content/drive/MyDrive/archive_FastSpeech2/Korean-FastSpeech2-Pytorch/data/Dataset/kss/wavs/0050_G2A4E7S0C2_HJH_000760.wa
  Metadata:
    ISFT            : Lavf58.76.100
  Stream #0:0: Audio: pcm_s16le ([1][0][0][0] / 0x0001), 22050 Hz, mono, s16, 352 kb/s
    Metadata:
      encoder         : Lavc58.134.100 pcm_s16le
size=     159kB time=00:00:03.68 bitrate= 353.0kbits/s speed= 566x
video:0kB audio:159kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: 0.047932%
ffmpeg version 4.4.2-0ubuntu0.22.04.1 Copyright (c) 2000-2021 the FFmpeg developers
  built with gcc 11 (Ubuntu 11.2.0-19ubuntu1)
  configuration: --prefix=/usr --extra-version=0ubuntu0.22.04.1 --toolchain=hardened --libdir=/usr/lib/x86_64-linux-gnu --incdir=/usr/include
  libavutil      56. 70.100 / 56. 70.100
  libavcodec     58.134.100 / 58.134.100
  libavformat    58. 76.100 / 58. 76.100
  libavdevice    58. 13.100 / 58. 13.100
  libavfilter     7.110.100 /  7.110.100
  libswscale      5.  9.100 /  5.  9.100
  libswresample   3.  9.100 /  3.  9.100
  libpostproc    55.  9.100 / 55.  9.100
Guessed Channel Layout for Input Stream #0.0 : mono
Input #0, wav, from '/content/drive/MyDrive/archive_FastSpeech2/Korean-FastSpeech2-Pytorch/data/Dataset/kss/wavs_bak/2/0050_G2A4E7S0C2_HJH_00
  Duration: 00:00:04.99, bitrate: 768 kb/s
  Stream #0:0: Audio: pcm_s16le ([1][0][0][0] / 0x0001), 48000 Hz, mono, s16, 768 kb/s
Stream mapping:
  Stream #0:0 -> #0:0 (pcm_s16le (native) -> pcm_s16le (native))
Press [q] to stop, [?] for help
Output #0, wav, to '/content/drive/MyDrive/archive_FastSpeech2/Korean-FastSpeech2-Pytorch/data/Dataset/kss/wavs/0050_G2A4E7S0C2_HJH_000780.wa
  Metadata:
    ISFT            : Lavf58.76.100
  Stream #0:0: Audio: pcm_s16le ([1][0][0][0] / 0x0001), 22050 Hz, mono, s16, 352 kb/s
    Metadata:
      encoder         : Lavc58.134.100 pcm_s16le
```

```
import os
import sys
os.chdir("/content/drive/MyDrive/archive_FastSpeech2/Korean-FastSpeech2-Pytorch")

!python train.py
```

```
스트리밍 출력 내용이 길어서 마지막 5000줄이 삭제되었습니다.
Epoch [290/1000], Step [60130/208000]:
Total Loss: 0.4062, Mel Loss: 0.0714, Mel PostNet Loss: 0.0708, Duration Loss: 0.0654, F0 Loss: 0.1048, Energy Loss: 0.0937;
Time Used: 22665.951s, Estimated Time Remaining: 30132.217s.
```

```
Epoch [290/1000], Step [60140/208000]:
Total Loss: 0.4292, Mel Loss: 0.0641, Mel PostNet Loss: 0.0635, Duration Loss: 0.0674, F0 Loss: 0.1296, Energy Loss: 0.1046;
Time Used: 22668.151s, Estimated Time Remaining: 28551.832s.

Epoch [290/1000], Step [60150/208000]:
Total Loss: 0.3830, Mel Loss: 0.0689, Mel PostNet Loss: 0.0683, Duration Loss: 0.0596, F0 Loss: 0.0955, Energy Loss: 0.0909;
Time Used: 22672.850s, Estimated Time Remaining: 32437.432s.

Epoch [290/1000], Step [60160/208000]:
Total Loss: 0.3278, Mel Loss: 0.0577, Mel PostNet Loss: 0.0571, Duration Loss: 0.0536, F0 Loss: 0.0803, Energy Loss: 0.0791;
Time Used: 22674.606s, Estimated Time Remaining: 24606.724s.

Epoch [290/1000], Step [60170/208000]:
Total Loss: 0.3763, Mel Loss: 0.0683, Mel PostNet Loss: 0.0676, Duration Loss: 0.0578, F0 Loss: 0.0975, Energy Loss: 0.0850;
Time Used: 22679.452s, Estimated Time Remaining: 31727.761s.

Epoch [290/1000], Step [60180/208000]:
Total Loss: 0.3816, Mel Loss: 0.0714, Mel PostNet Loss: 0.0708, Duration Loss: 0.0579, F0 Loss: 0.0973, Energy Loss: 0.0841;
Time Used: 22684.018s, Estimated Time Remaining: 35409.734s.

Epoch [290/1000], Step [60190/208000]:
Total Loss: 0.3404, Mel Loss: 0.0617, Mel PostNet Loss: 0.0612, Duration Loss: 0.0531, F0 Loss: 0.0786, Energy Loss: 0.0859;
Time Used: 22685.938s, Estimated Time Remaining: 31895.525s.

Epoch [290/1000], Step [60200/208000]:
Total Loss: 0.3888, Mel Loss: 0.0692, Mel PostNet Loss: 0.0685, Duration Loss: 0.0565, F0 Loss: 0.0979, Energy Loss: 0.0966;
Time Used: 22690.673s, Estimated Time Remaining: 36543.053s.

Epoch [290/1000], Step [60210/208000]:
Total Loss: 0.4078, Mel Loss: 0.0710, Mel PostNet Loss: 0.0703, Duration Loss: 0.0644, F0 Loss: 0.1044, Energy Loss: 0.0977;
Time Used: 22695.171s, Estimated Time Remaining: 31796.202s.

Epoch [290/1000], Step [60220/208000]:
Total Loss: 0.3640, Mel Loss: 0.0655, Mel PostNet Loss: 0.0648, Duration Loss: 0.0561, F0 Loss: 0.0873, Energy Loss: 0.0903;
Time Used: 22697.407s, Estimated Time Remaining: 32695.791s.

Epoch [290/1000], Step [60230/208000]:
Total Loss: 0.4162, Mel Loss: 0.0736, Mel PostNet Loss: 0.0729, Duration Loss: 0.0641, F0 Loss: 0.1045, Energy Loss: 0.1011;
Time Used: 22702.147s, Estimated Time Remaining: 33528.898s.

Epoch [290/1000], Step [60240/208000]:
Total Loss: 0.3334, Mel Loss: 0.0585, Mel PostNet Loss: 0.0579, Duration Loss: 0.0508, F0 Loss: 0.0834, Energy Loss: 0.0827;
Time Used: 22703.851s, Estimated Time Remaining: 26009.251s.

Epoch [290/1000], Step [60250/208000]:
Total Loss: 0.3754, Mel Loss: 0.0665, Mel PostNet Loss: 0.0659, Duration Loss: 0.0596, F0 Loss: 0.0937, Energy Loss: 0.0897;
Time Used: 22708.629s, Estimated Time Remaining: 31299.720s.

Epoch [290/1000], Step [60260/208000]:
Total Loss: 0.4208, Mel Loss: 0.0738, Mel PostNet Loss: 0.0732, Duration Loss: 0.0650, F0 Loss: 0.1101, Energy Loss: 0.0985;
Time Used: 22713.417s, Estimated Time Remaining: 33121.578s.

Epoch [290/1000], Step [60270/208000]:
```

```
import os
import sys
os.chdir("/content/drive/MyDrive/archive_FastSpeech2/Korean-FastSpeech2-Pytorch")

!python train.py --restore_step 140000
```

```
스트리밍 출력 내용이 길어서 마지막 5000줄이 삭제되었습니다.
Time Used: 40932.211s, Estimated Time Remaining: -2570.110s.

Epoch [390/1000], Step [220940/208000]:
Total Loss: 0.2856, Mel Loss: 0.0627, Mel PostNet Loss: 0.0617, Duration Loss: 0.0308, F0 Loss: 0.0648, Energy Loss: 0.0655;
Time Used: 40934.254s, Estimated Time Remaining: -2699.269s.

Epoch [390/1000], Step [220950/208000]:
Total Loss: 0.3250, Mel Loss: 0.0706, Mel PostNet Loss: 0.0693, Duration Loss: 0.0374, F0 Loss: 0.0703, Energy Loss: 0.0774;
Time Used: 40938.961s, Estimated Time Remaining: -2941.513s.

Epoch [390/1000], Step [220960/208000]:
Total Loss: 0.2716, Mel Loss: 0.0574, Mel PostNet Loss: 0.0562, Duration Loss: 0.0328, F0 Loss: 0.0613, Energy Loss: 0.0639;
Time Used: 40940.661s, Estimated Time Remaining: -2627.249s.

Epoch [390/1000], Step [220970/208000]:
Total Loss: 0.2896, Mel Loss: 0.0653, Mel PostNet Loss: 0.0642, Duration Loss: 0.0326, F0 Loss: 0.0616, Energy Loss: 0.0660;
Time Used: 40945.403s, Estimated Time Remaining: -3263.507s.

Epoch [390/1000], Step [220980/208000]:
Total Loss: 0.3462, Mel Loss: 0.0722, Mel PostNet Loss: 0.0709, Duration Loss: 0.0376, F0 Loss: 0.0837, Energy Loss: 0.0819;
Time Used: 40949.927s, Estimated Time Remaining: -3002.271s.

Epoch [390/1000], Step [220990/208000]:
Total Loss: 0.2898, Mel Loss: 0.0597, Mel PostNet Loss: 0.0585, Duration Loss: 0.0319, F0 Loss: 0.0669, Energy Loss: 0.0729;
Time Used: 40951.862s, Estimated Time Remaining: -2681.618s.

Epoch [390/1000], Step [221000/208000]:
```

```
Total Loss: 0.3266, Mel Loss: 0.0692, Mel PostNet Loss: 0.0680, Duration Loss: 0.0343, F0 Loss: 0.0759, Energy Loss: 0.0791;
Time Used: 40956.434s, Estimated Time Remaining: -2718.954s.
done

FastSpeech2 Step 221000,
Duration Loss: 0.5958324847397981
F0 Loss: 0.7345797077373222
Energy Loss: 0.6360382526009171
Mel Loss: 0.33839005287046786
Mel Postnet Loss: 0.33811924468587945

Epoch [390/1000], Step [221010/208000]:
Total Loss: 0.3306, Mel Loss: 0.0727, Mel PostNet Loss: 0.0713, Duration Loss: 0.0351, F0 Loss: 0.0805, Energy Loss: 0.0710;
Time Used: 40967.088s, Estimated Time Remaining: -9548.813s.

Epoch [390/1000], Step [221020/208000]:
Total Loss: 0.3423, Mel Loss: 0.0628, Mel PostNet Loss: 0.0616, Duration Loss: 0.0420, F0 Loss: 0.0989, Energy Loss: 0.0770;
Time Used: 40969.131s, Estimated Time Remaining: -3879.670s.

Epoch [390/1000], Step [221030/208000]:
Total Loss: 0.3273, Mel Loss: 0.0684, Mel PostNet Loss: 0.0672, Duration Loss: 0.0360, F0 Loss: 0.0773, Energy Loss: 0.0783;
Time Used: 40973.881s, Estimated Time Remaining: -3074.246s.

Epoch [390/1000], Step [221040/208000]:
Total Loss: 0.3087, Mel Loss: 0.0577, Mel PostNet Loss: 0.0565, Duration Loss: 0.0311, F0 Loss: 0.0701, Energy Loss: 0.0933;
Time Used: 40975.682s, Estimated Time Remaining: -2253.710s.

Epoch [390/1000], Step [221050/208000]:
Total Loss: 0.3031, Mel Loss: 0.0657, Mel PostNet Loss: 0.0645, Duration Loss: 0.0360, F0 Loss: 0.0639, Energy Loss: 0.0731;
Time Used: 40980.401s, Estimated Time Remaining: -2944.846s.
```

```
!pip install g2pk
```

```
Collecting g2pk
  Downloading g2pK-0.9.4-py3-none-any.whl.metadata (7.5 kB)
Requirement already satisfied: jamo in /usr/local/lib/python3.10/dist-packages (from g2pk) (0.4.1)
Requirement already satisfied: nltk in /usr/local/lib/python3.10/dist-packages (from g2pk) (3.8.1)
Collecting konlpy (from g2pk)
  Downloading konlpy-0.6.0-py2.py3-none-any.whl.metadata (1.9 kB)
Collecting python-mecab-ko (from g2pk)
  Downloading python_mecab_ko-1.3.7-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (3.4 kB)
Collecting JPype1>=0.7.0 (from konlpy->g2pk)
  Downloading JPype1-1.5.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (4.9 kB)
Requirement already satisfied: lxml>=4.1.0 in /usr/local/lib/python3.10/dist-packages (from konlpy->g2pk) (5.3.0)
Requirement already satisfied: numpy>=1.6 in /usr/local/lib/python3.10/dist-packages (from konlpy->g2pk) (1.26.4)
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from nltk->g2pk) (8.1.7)
Requirement already satisfied: joblib in /usr/local/lib/python3.10/dist-packages (from nltk->g2pk) (1.4.2)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.10/dist-packages (from nltk->g2pk) (2024.9.11)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from nltk->g2pk) (4.66.6)
Collecting python-mecab-ko-dic (from python-mecab-ko->g2pk)
  Downloading python_mecab_ko_dic-2.1.1.post2-py3-none-any.whl.metadata (1.4 kB)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from JPype1>=0.7.0->konlpy->g2pk) (24.1)
Downloading g2pK-0.9.4-py3-none-any.whl (27 kB)
Downloading konlpy-0.6.0-py2.py3-none-any.whl (19.4 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 19.4/19.4 MB 103.2 MB/s eta 0:00:00
Downloading python_mecab_ko-1.3.7-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (577 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 577.1/577.1 kB 37.8 MB/s eta 0:00:00
Downloading JPype1-1.5.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (488 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 488.6/488.6 kB 40.0 MB/s eta 0:00:00
Downloading python_mecab_ko_dic-2.1.1.post2-py3-none-any.whl (34.5 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 34.5/34.5 MB 65.7 MB/s eta 0:00:00
Installing collected packages: python-mecab-ko-dic, python-mecab-ko, JPype1, konlpy, g2pk
Successfully installed JPype1-1.5.0 g2pk-0.9.4 konlpy-0.6.0 python-mecab-ko-1.3.7 python-mecab-ko-dic-2.1.1.post2
```

```
import os
import sys
os.chdir("/content/drive/MyDrive/archive_FastSpeech2/Korean-FastSpeech2-Pytorch")
```

```
!python synthesize.py --step 70000
```

```
/content/drive/MyDrive/archive_FastSpeech2/Korean-FastSpeech2-Pytorch/synthesize.py:49: FutureWarning: You are using `torch.load` with `weights_(
  model.load_state_dict(torch.load(checkpoint_path)['model'])
/content/drive/MyDrive/archive_FastSpeech2/Korean-FastSpeech2-Pytorch/utils.py:116: FutureWarning: You are using `torch.load` with `weights_only=
  checkpoint = torch.load(ckpt_path)
/usr/local/lib/python3.10/dist-packages/torch/nn/utils/weight_norm.py:143: FutureWarning: `torch.nn.utils.weight_norm` is deprecated in favor of
  WeightNorm.apply(module, name, dim)
which sentence do you want?
1.eval_sentence 2.train_sentence 3.test_sentence 4.create new sentence
4
you went for mode 4
input sentence
안녕하세요, 오늘도 화이팅,
sentence that will be synthesized:
안녕하세요, 오늘도 화이팅,
after g2p:  안녕하세요, 오늘도 화이팅,
after h2j:  안녕하세요, 오늘도 화이팅,
phone:  {ㅇ}{ㅏ}{ㄴ}{ㄴ}{ㅕ}{ㅇ}{ㅎ}{ㅏ}{ㅅ}{ㅔ}{ㅇ}{ㅛ}{,}{ㅇ}{ㅗ}{ㄴ}{ㅡ}{ㄹ}{ㄷ}{ㅗ}{ㅎ}{ㅘ}{ㅇ}{ㅣ}{ㅌ}{ㅣ}{ㅇ}{,}
after re.sub:  {ㅇ}{ㅏ}{ㄴ}{ㄴ}{ㅕ}{ㅇ}{ㅎ}{ㅏ}{ㅅ}{ㅔ}{ㅇ}{ㅛ}{sp}{ㅇ}{ㅗ}{ㄴ}{ㅡ}{ㄹ}{ㄷ}{ㅗ}{ㅎ}{ㅘ}{ㅇ}{ㅣ}{ㅌ}{ㅣ}{ㅇ}{sp}
```

```
|{ㅇ ㅏ ㄴ ㄴ ㅕ ㅇ ㅎ ㅏ ㅅ ㅔ ㅇ ㅛ sp ㅇ ㅗ ㄴ ㅡ ㄹ ㄷ ㅗ ㅎ ㅘ ㅇ ㅣ ㅌ ㅣ ㅇ sp}|
```

```
import shutil

shutil.rmtree('/content/drive/MyDrive/archive_FastSpeech2/Korean-FastSpeech2-Pytorch/ckpt')
shutil.rmtree('/content/drive/MyDrive/archive_FastSpeech2/Korean-FastSpeech2-Pytorch/log')
```

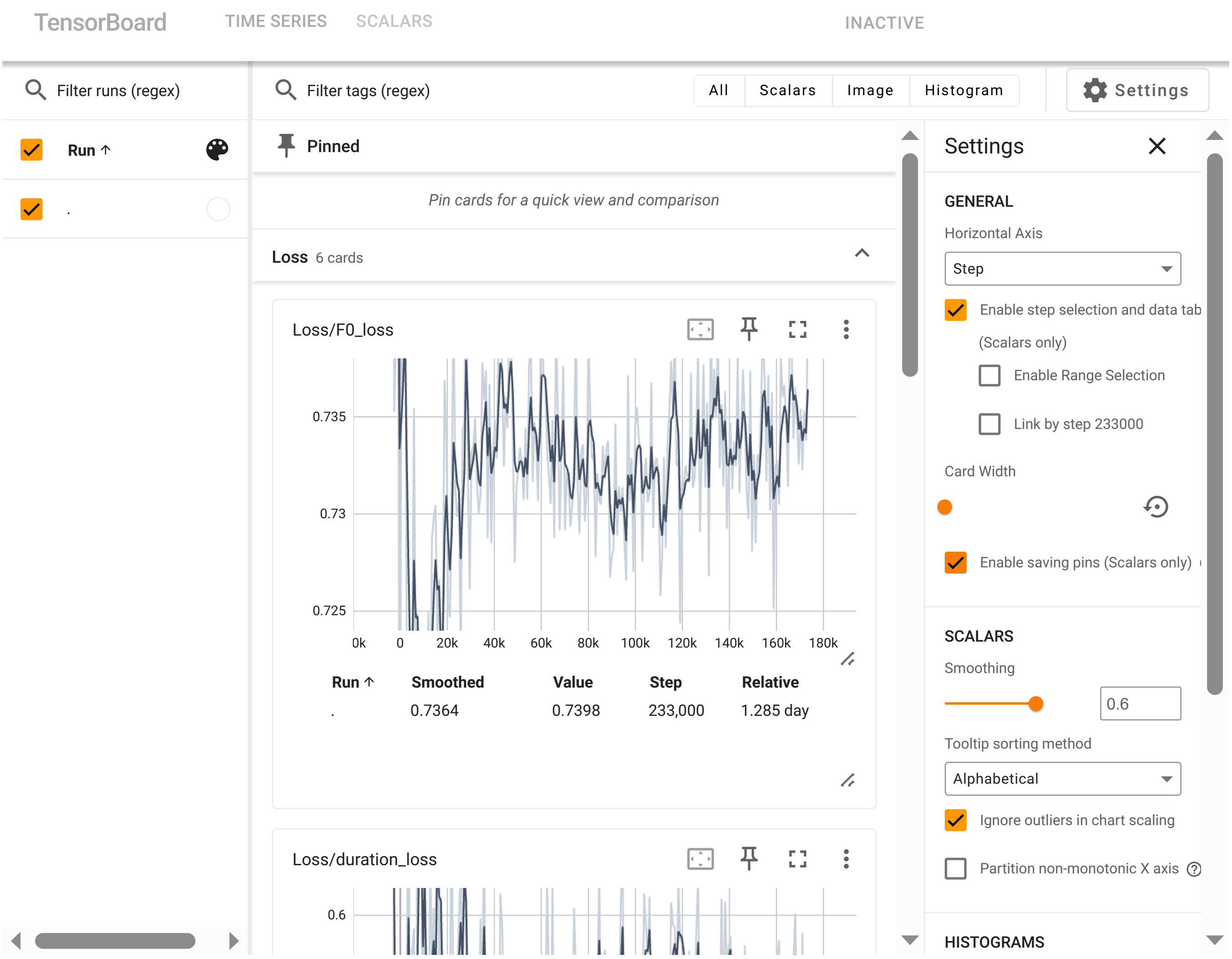
```
import shutil

shutil.rmtree('/content/drive/MyDrive/archive_FastSpeech2/Korean-FastSpeech2-Pytorch/results')
```

```
!free -h
```

```
              total        used        free      shared  buff/cache   available
Mem:           50Gi       1.7Gi        32Gi       1.0Mi        16Gi        48Gi
Swap:            0B          0B          0B
```

```
%load_ext tensorboard
%tensorboard --logdir /content/drive/MyDrive/archive_FastSpeech2/Korean-FastSpeech2-Pytorch/log/kss/validation
```

```
The tensorboard extension is already loaded. To reload it, use:
  %reload_ext tensorboard
```



☑ Run ↑

☑ .

Run ↑	Smoothed	Value	Step	Relative
.	0.7364	0.7398	233,000	1.285 day

Settings

GENERAL

Horizontal Axis: Step

☑ Enable step selection and data tab (Scalars only)

□ Enable Range Selection

□ Link by step 233000

Card Width

☑ Enable saving pins (Scalars only)

SCALARS

Smoothing: 0.6

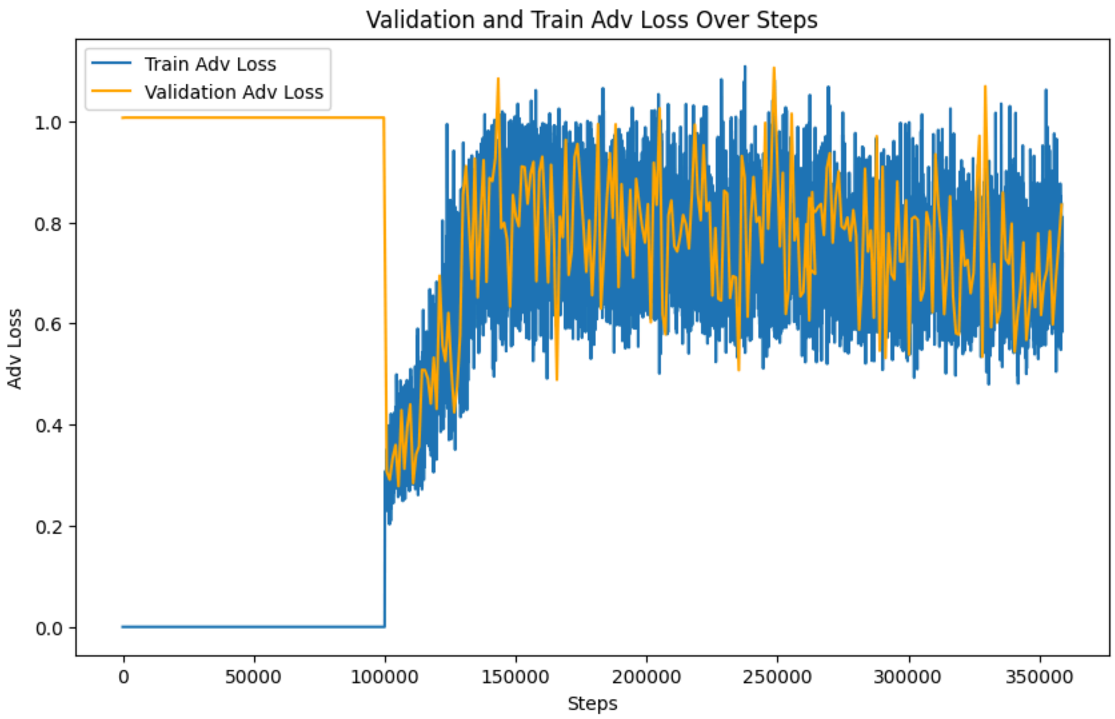
Tooltip sorting method: Alphabetical

☑ Ignore outliers in chart scaling

□ Partition non-monotonic X axis

```
import matplotlib.pyplot as plt
from tensorboard.backend.event_processing import event_accumulator

# 이벤트 파일 로드
event_file = "/content/drive/MyDrive/archive_VocGan/datasets/logs/vocgan_Aihub_pretrained_model_epoch"  # 파일 경로를 지정하세요
event_acc = event_accumulator.EventAccumulator(event_file)
event_acc.Reload()

# g_loss 값 추출
validation_g_loss = event_acc.Scalars('validation.adv_loss')
train_g_loss = event_acc.Scalars('train.adv_loss')

# 스텝 및 손실 값 추출
```

```
validation_steps = [x.step for x in validation_g_loss]
validation_values = [x.value for x in validation_g_loss]
train_steps = [x.step for x in train_g_loss]
train_values = [x.value for x in train_g_loss]

# 그래프 그리기
plt.figure(figsize=(10, 6))
plt.plot(train_steps, train_values, label='Train Adv Loss')
plt.plot(validation_steps, validation_values, label='Validation Adv Loss', color='orange')
plt.xlabel("Steps")
plt.ylabel("Adv Loss")
plt.legend()
plt.title("Validation and Train Adv Loss Over Steps")
plt.show()
```



```
!python '/content/drive/MyDrive/archive_VocGan/VocGAN/inference.py' -p '/content/drive/MyDrive/archive_VocGan/datasets/chkpt/vocgan_Aihub_pretrained_
```

```
Traceback (most recent call last):
  File "/content/drive/MyDrive/archive_VocGan/VocGAN/inference.py", line 10, in <module>
    from denoiser import Denoiser
  File "<frozen importlib._bootstrap>", line 1027, in _find_and_load
  File "<frozen importlib._bootstrap>", line 1006, in _find_and_load_unlocked
  File "<frozen importlib._bootstrap>", line 688, in _load_unlocked
  File "<frozen importlib._bootstrap_external>", line 879, in exec_module
  File "<frozen importlib._bootstrap_external>", line 975, in get_code
  File "<frozen importlib._bootstrap_external>", line 1074, in get_data
KeyboardInterrupt
^C
```

```
!rm -rf /content/drive/MyDrive/archive_FastSpeech2/Korean-FastSpeech2-Pytorch/log/kss/train/logs/*
!rm -rf /content/drive/MyDrive/archive_FastSpeech2/Korean-FastSpeech2-Pytorch/log/kss/validation/logs/*
%tensorboard --logdir_spec=train:/content/drive/MyDrive/archive_FastSpeech2/Korean-FastSpeech2-Pytorch/log/kss/train,validation:/content/drive/MyDriv
```

```
def get_alignment(tier):
    sil_phones = ['sil', 'sp', 'spn', '']

    phones = []
    durations = []
    start_time = 0
    end_time = 0
    end_idx = 0
    for t in tier._objects:
        s, e, p = t.start_time, t.end_time, t.text

        if not p:
            p = 'sp'

        print(f"[{p}]")
        # Trimming leading silences
        if phones == []:
            if p in sil_phones:
                continue
            else:
                start_time = s
        if p not in sil_phones:
            phones.append(p)
            end_time = e
            end_idx = len(phones)
        else:
            phones.append(p)
        durations.append(int(e*hp.sampling_rate/hp.hop_length)-int(s*hp.sampling_rate/hp.hop_length))

    # Trimming tailing silences
    phones = phones[:end_idx]
    durations = durations[:end_idx]

    return phones, np.array(durations), start_time, end_time


textgrid = tgt.io.correct_start_end_times_and_fill_gaps('/content/drive/MyDrive/archive_FastSpeech2/Korean-FastSpeech2-Pytorch/preprocessed/kss/TextG
#textgridX = tgt.io.correct_start_end_times_and_fill_gaps(textgrid)

get_alignment(textgridX.get_tier_by_name('phones'))
```

```
([ 'ㅇ',
  'ㅏ',
  'ㅃ',
  'ㅏ',
  'ㅇ',
  'ㅔ',
  'ㄱ',
  'ㅔ',
  'sp',
  'ㄱ',
  'ㅡ',
  'ㄹ',
  'ㅓ',
  'ㅋ',
  'ㅔ',
  'ㅊ',
  'ㅣ',
  'ㄴ',
  'ㅈ',
  'ㅓ',
  'ㄹ',
  'ㅎ',
  'ㅏ',
  'ㄱ',
  'ㅔ',
  'ㄷ',
  'ㅐ',
  'ㅎ',
  'ㅐ',
  'ㅈ',
  'ㅜ',
  'ㅅ',
  'ㅕ',
  'ㅅ',
  'ㅓ',
  'sp',
  'ㄱ',
  'ㅏ',
  'ㅁ',
  'ㅅ',
  'ㅏ',
  'ㄷ',
  'ㅡ',
  'ㄹ',
  'ㅣ',
  'ㅁ',
  'ㄴ',
  'ㅣ',
  'ㄷ',
  'ㅏ'],
 array([ 1,  7, 12, 10,  4,  5,  6, 12,  8,  8,  3,  3,  7,  9,  6, 11,  2,
         6,  4,  5,  3,  5,  4,  5,  6,  8,  9,  3,  4,  6,  4,  6,  8,  7,
        17, 16,  5,  5,  6,  7,  6,  6,  3,  3,  6,  3,  3,  7,  5, 19]),
 0.34,
 4.1)
```

```
import numpy as np
import os
import tgt
from scipy.io.wavfile import read
import pyworld as pw
import torch
import audio as Audio
from utils import get_alignment, standard_norm, remove_outlier, average_by_duration
import hparams as hp
from jamo import h2j
import codecs

def process_utterance():

    textgrid = tgt.io.read_textgrid('/content/drive/MyDrive/archive_FastSpeech2/Korean-FastSpeech2-Pytorch/preprocessed/kss/TextGrid/0050_G2A4E1S0C1_
    phone, duration, start, end = get_alignment(textgrid.get_tier_by_name('phones'))
    print(phone)
    print(len(phone))
    print(duration)
    print(len(duration))
    print(start)
    print(end)
    text = '{'+ '}{'.join(phone) + '}' # '{A}{B}{$}{C}', $ represents silent phones
    print(text)
    text = text.replace('{$}', ' ')    # '{A}{B} {C}'
    print(text)
    text = text.replace('}{', ' ')     # '{A B} {C}'

    if start >= end:
        return None


process_utterance()
```

```
['ㅇ', 'ㅏ', 'ㅃ', 'ㅏ', 'ㅇ', 'ㅔ', 'ㄱ', 'ㅔ', 'ㄱ', 'ㅡ', 'ㄹ', 'ㅓ', 'ㅋ', 'ㅔ', 'ㅊ', 'ㅣ', 'ㄴ', 'ㅈ', 'ㅓ', 'ㄹ', 'ㅎ', 'ㅏ', 'ㄱ', 'ㅔ',
48
[ 1  7 12 10  4  5  6 12  8  3  3  7  9  6 11  2  6  4  5  3  5  4  5  6
  8  9  3  4  6  4  6  8  7 17  5  5  6  7  6  6  3  3  6  3  3  7  5 19]
48
0.34
4.1
{ㅇ}{ㅏ}{ㅃ}{ㅏ}{ㅇ}{ㅔ}{ㄱ}{ㅔ}{ㄱ}{ㅡ}{ㄹ}{ㅓ}{ㅋ}{ㅔ}{ㅊ}{ㅣ}{ㄴ}{ㅈ}{ㅓ}{ㄹ}{ㅎ}{ㅏ}{ㄱ}{ㅔ}{ㄷ}{ㅐ}{ㅎ}{ㅐ}{ㅈ}{ㅜ}{ㅅ}{ㅕ}{ㅅ}{ㅓ}{ㄱ}{ㅏ}
{ㅇ}{ㅏ}{ㅃ}{ㅏ}{ㅇ}{ㅔ}{ㄱ}{ㅔ}{ㄱ}{ㅡ}{ㄹ}{ㅓ}{ㅋ}{ㅔ}{ㅊ}{ㅣ}{ㄴ}{ㅈ}{ㅓ}{ㄹ}{ㅎ}{ㅏ}{ㄱ}{ㅔ}{ㄷ}{ㅐ}{ㅎ}{ㅐ}{ㅈ}{ㅜ}{ㅅ}{ㅕ}{ㅅ}{ㅓ}{ㄱ}{ㅏ}
```

```
!cp -r /content/drive/MyDrive/archive_FastSpeech2/Korean-FastSpeech2-Pytorch/data_First/Dataset/kss/wavs_bak/* /content/drive/MyDrive/archive_FastSpe
```