國立陽明交通大學
NATIONAL YANG MING CHIAO TUNG UNIVERSITY

[IIAI30003] Digital Speech Processing

Homework 4

Po-Chuan, Chen
Student ID: 311511052
`present90308.ee11@nycu.edu.tw`

National Yang Ming Chiao Tung University

November 27, 2023

Contents

1 Build espnet

```
sudo apt-get install cmake
sudo apt-get install sox
sudo apt-get install libsndfile1-dev
sudo apt-get install ffmpeg
sudo apt-get install flac

git clone https://github.com/espnet/espnet
```

Listing 1: Install some package for espnet

```
conda create --name espnet python=3.10
conda activate espnet
```

Listing 2: Using anaconda to build enivorment

```
cd espnet/tools
CONDA_TOOLS_DIR=$(dirname ${CONDA_EXE})/..
./setup_anaconda.sh ${CONDA_TOOLS_DIR} espnet 3.10
make -j 40
make kenlm.done
```

Listing 3: Set conda's enivorment

```
cd espnet/egs2
./TEMPLATE/asr1/setup.sh ./taiwanese/asr1
```

Listing 4: Create new recipes

Copy other recipe's file, for example, `aishell`

- `run.sh`
- `local/data.sh`
- `conf`

2 Overview

All the file in the `egs2/taiwanese/asr1`

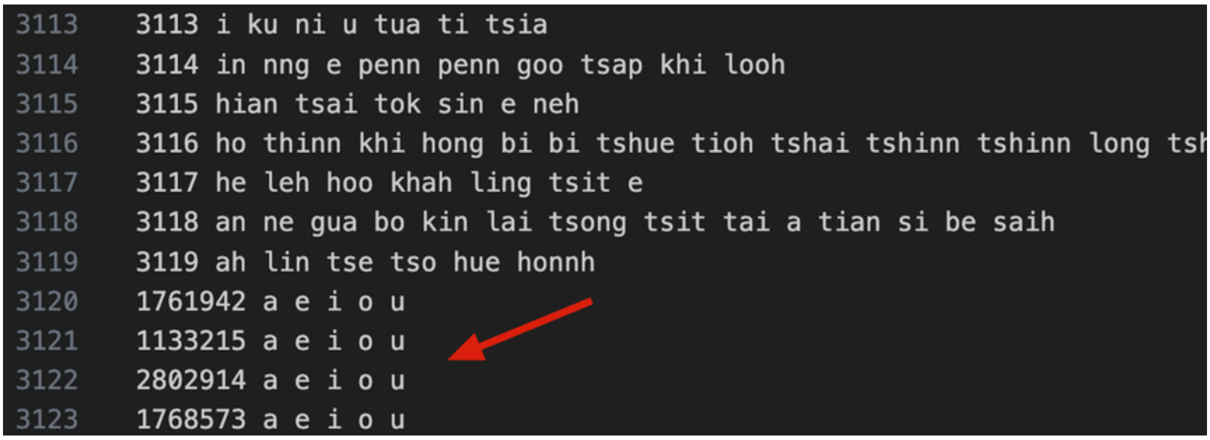
1. `conf`
2. `local`
    - `data.sh`
3. `steps`
4. `utils`
5. `pyscripts`
6. `scripts`
7. `cmd.sh`
8. `path.sh`
9. `asr.sh`
10. `db.sh`
11. `downloads`
12. `run.sh`

If all the file are in the folder, we can run `yesno` task, from `espnet/eg2/yesno/asr1`, using `bash run.sh`.

3 Data

```
data_aishell/
   resource_aishell/
      --speaker.info
      --lexicon.txt (no need)
   wav/
      train/000/audiofile
      test/000/...
      dev/000/...
   transcript/
      --aishell_transcript_v0.8.txt
```

Listing 5: aishell's file architecture



```
3113 i ku ni u tua ti tsia
3114 in nng e penn penn goo tsap khi looh
3115 hian tsai tok sin e neh
3116 ho thinn khi hong bi bi tshue tioh tshai tshinn tshinn long tsh
3117 he leh hoo khah ling tsit e
3118 an ne gua bo kin lai tsong tsit tai a tian si be saih
3119 ah lin tse tso hue honnh
1761942 a e i o u
1133215 a e i o u
2802914 a e i o u
1768573 a e i o u
```

Figure 1: test id need to add below the original text

```
mkdir new_train
mkdir new_test

for file in train/*.wav; do
  new_file="new_train/$(basename "$file")"
  sox "$file" -r 16000 -e signed-integer -b 16 "$new_file"
done

for file in test/*.wav; do
  new_file="new_test/$(basename "$file")"
  sox "$file" -r 16000 -e signed-integer -b 16 "$new_file"
done
```

Listing 6: Set Wav to 16kHz

```
import csv

with open('text.txt', 'w') as outfile:
    with open('train-toneless.csv', 'r', errors='ignore') as infile:
        [outfile.write(" ".join(row) + "\n") for row in csv.reader(infile)]
    outfile.close()
```

Listing 7: Transfering csv file to txt file

```python
import csv

# Read the text file
with open('your_text_file.txt', 'r') as file:
    lines = file.readlines()

# Parse the text data and convert it into a list of dictionaries
data = []
for line in lines:
    parts = line.strip().split()
    id, *words = parts
    data.append({'id': id, 'text': ' '.join(words)})

# Sort the data by the 'id' field
sorted_data = sorted(data, key=lambda x: int(x['id']))

# Write the sorted data to a CSV file
with open('output.csv', 'w', newline='') as csvfile:
    fieldnames = ['id', 'text']
    writer = csv.DictWriter(csvfile, fieldnames=fieldnames)

    writer.writeheader()
    for row in sorted_data:
        writer.writerow(row)

print("Data has been transferred to output.csv and sorted by ID.")
```

Listing 8: Transfering txt file to csv file

4 Run Different Model

```bash
#!/usr/bin/env bash
set -e
set -u
set -o pipefail

train_set=train
valid_set=dev
test_sets=test

asr_config=conf/tuning/train_asr_ctc_conformer_e15_linear1024.yaml
inference_config=conf/decode_asr_transformer.yaml

lm_config=conf/train_lm_transformer.yaml
use_lm=false
use_wordlm=false

# speed perturbation related
# (train_set will be "${train_set}_sp" if speed_perturb_factors is
    specified)
speed_perturb_factors="0.9 1.0 1.1"

./asr.sh \
    --nj 32 \
    --inference_nj 32 \
    --ngpu 1 \
    --lang zh \
    --audio_format "flac.ark" \
    --feats_type raw \
    --token_type char \
    --use_lm ${use_lm}                                   \
    --use_word_lm ${use_wordlm}                          \
    --lm_config "${lm_config}"                           \
    --asr_config "${asr_config}"                         \
    --inference_config "${inference_config}"             \
    --train_set "${train_set}"                           \
    --valid_set "${valid_set}"                           \
    --test_sets "${test_sets}"                           \
    --speed_perturb_factors "${speed_perturb_factors}" \
    --asr_speech_fold_length 512 \
    --asr_text_fold_length 150 \
    --lm_fold_length 150 \
    --lm_train_text "data/${train_set}/text" "$@" \
```

Listing 9: run.sh

- branchformer fast selfattn e24 amp: 34.7864
- transformer: 38.3398
- **branchformer: 33.59223**
- conformer: 36.47572
- sample: 56.33009