Execute CHiME-5 Baseline in ESPNet

WANG Wenbo

1.Introduction about ESPNet

ESPNet (end-to-end speech processing toolkit) uses Chainer and Pytorch as its learning engine, and uses Kaldi to do Kaldi style data processing, feature extraction/format[1]. It is mainly focused on Hybrid CTC/Attention architecture to do the speech recognition[2]. The steps below are a brief overview of the ESPNet when running CHiME-5 recipe.

Step0 Data Preparation (Kaldi style)
Step1 Feature Generation (Kaldi style)
Step2 Dictionary & Json Data Preparation
Step3 LM Preparation
Step4 Network Training (based on Chainer or Pytorch)
Step5 Decoding (Recognition and scoring)

2.Preparation for Installation

The basic installation can be found in [1], before your installation, the follow requirements need to be checked.

Python2.7+
Cuda 8.0 (for the use of GPU)
Cudnn 6 (for the use of GPU)
NCCL 2.0+ (for the use of multi-GPUs)
virtualenv

2.1 Set paths for cuda

To use cuda (and cudnn), make sure to set paths in your .bashrc appropriately. And note that only cuda-9.0 and cuda-8.0 can meet the requirements.

```
CUDAROOT=/path/to/cuda
export PATH=$CUDAROOT/bin:$PATH
export LD_LIBRARY_PATH=$CUDAROOT/lib64:$LD_LIBRARY_PATH
export CUDA_HOME=$CUDAROOT
export CUDA_PATH=$CUDAROOT
```

2.2 Install virtualenv

If you already installed Anaconda, you can change your paths in your .bashrc, and use Python in your Anaconda as default setting, which have already installed the virtualenv (a tool to create isolated Python environments). Use the code below to install virtualenv.

```
$ pip install virtualenv
```

3.Installation

3.1 Download ESPNet

Download the latest version in [1] as follow code.

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

3.2 Makefile

You can use the follow code to install ESPNet.

```
$ cd tools
$ make -j
```

After make file, you should probably check whether it is successful or not. You can check the return value of the last executed command.

```
$ echo $?
```

And you can also check the Makefile content to make sure whether the files and directories below exist or not. The kaldi below links to the directory of kaldi_github, if you used kaldi before you can also link kaldi here to yours.

chainer_ctc
kaldi
kaldi_github
kaldi-io-for-python
nkf
subword-nmt
venv
warp-ctc

3.3 Install SRILM

The SRILM (SRI Language Modeling Toolkit) is needed in the recipe, you can download from [3]. Run the code below to install it.

```
$ cd espnet/tools/kaldi/tools
$ ./install_srilm.sh
```

You may also need to set the SRILM path.

```
export SRILM=$KALDI_ROOT/tools/srilm
export PATH=${PATH}:${SRILM}/bin:${SRILM}/bin/i686-m64
```

After installation success, you can run the recipe in CHiME-5.

4.Introduction about CHiME-5

The 5th CHiME Speech Separation and Recognition Challenge (CHiME-5) is mainly about the problem of distant multi-microphone conversational speech recognition in real

home environments. The audio data collected from house party has real conversation from relaxed people, and noise backgrounds from the kitchen appliances, air conditioning and so on[4].

4.1 Tracks

From [4] you can get there are 2 tracks in this challenge.

•Single-array

•Multiple-array

In the single-array track, only one reference array can be used to recognize a given evaluation utterance, in the multiple-array track, all arrays can be used.

For each track, this challenge will produce two separate rankings:

•Conventional systems

•All other systems (including end-to-end ASR)

In this challenge, manual modification of the data or annotations is forbidden. But we're entirely free in the development of system.

For every tested system, we should report 2 WERs (%):

•The WER on the development set

•The WER on the evaluation set

4.2 Data

Table 1 is the detail about the data. We can get the evaluation data in June/July, 2018.

Table 1 Detail about the CHiME-5 data

Dataset	Parties	Speakers	Hours	Utterances
Train	16	32	40:33	79,980
Dev	2	8	4:27	7,440
Eval	2	8	5:12	11,028

All the data is divided into session, each session includes 4 participants' conversation, and each conversation is recorded by a set of recording devices in Figure 1 and Figure 2.

Figure 1 Soundman A3 adapter onto Tascam

DR-05 stereo recorders will be worn by the participants

Figure 2 Microsoft Kinect devices set in the house

Each session has 4 participants wearing binaural microphones, and 6 microphone arrays with 4 microphones each are set in different room of the house (see Figure 3), you

can see it in Figure 2, each microphone arrays has 4 channel (CH1-CH4). Therefore, the total number of microphones per session is 32 (2 x 4 + 4 x 6)[4].

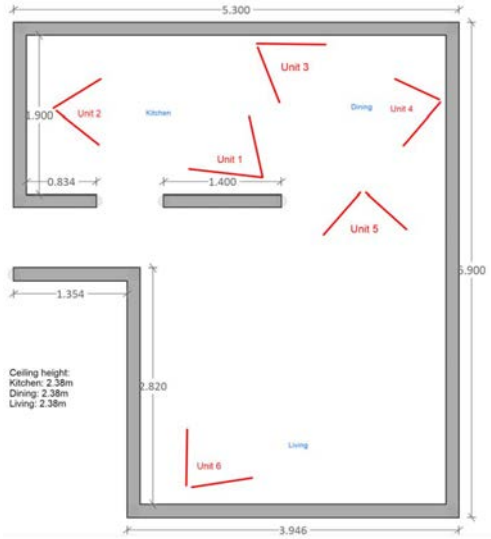

Figure 3 The setting of microphone arrays from S02 example

The wav files are named as follow:

•Binaural microphones

<session ID>_<speaker ID>.wav , e.g., S02_P05.wav

•Array microphone

<session ID>_<array ID>.CH<channel ID>.wav , e.g., S02_U05.CH1.wav

The transcriptions are made in JSON files for each session as <session ID>.json, e.g., S02.json.

5.Run CHiME-5 in ESPNet

The CHiME-5 data directory is /net/callisto/storage1/teisk/Chime5, also you can download from [4], it's 151GB (the eval set is not include).

The follow details show an example about how to run the CHiME-5 in ESPNet:

1.Move to CHiME-5 directory.

```
$ cd /espnet/egs/chime5/asr1
```

2.Set the paths in run.sh to the CHiME-5 data directory.

```
chime5_corpus=/your/path/to/CHiME-5_data
```

3.Set gpu in run.sh to use gpu.

```
gpu=0
```

4.Adjust batchsize, the original batchsize maybe too big, to avoid out of memory error, you can set as batchsize less than 30.

```
batchsize=28
```

5.Execute run.sh in the background.

```
$ nohup ./run.sh > run.log 2>&1 &
```

If you are using gpu run the recipe, you may get this error when train network.

```
catastrophic error: cannot open source file “cuda_fp16.h”
```

If so, you can copy the cuda_fp16.h file to your own virtual environment.

```
$cp /usr/local/cuda-9.0/include/cuda_fp16.h /espnet/tools/venv/lib/python2.7/ \
site-packages/cupy/core/include/cupy/cuda_fp16.h
```

Also don’t forget change the header file search priority in the file which includes cuda_fp16.h. Just like the code below, it can search the current directory first.

```
#include "cuda_fp16.h"
```

6.Results of CHiME-5 in ESPNet

Now, I’ve got some results of different batchsize in Table 2 and Table 3. The official data is from [4].

Table 2 Official results and results of different batchsize (beamformit)

	CER (%)	WER (%)
baseline(end to end)	N/A	94.7
dev_beamformit(beamsize=20)	72.5	93.0
dev_beamformit(beamsize=26)	75.2	97.0
dev_beamformit(beamsize=28)	73.4	94.3

Table 3 Official results and results of different batchsize (worn)

	CER (%)	WER (%)
baseline(end to end)	N/A	67.2
dev_worn(beamsize=20)	47.3	66.6
dev_worn(beamsize=26)	52.2	73.1
dev_worn(beamsize=28)	48.3	68.0

The results are two parts, dev_beamformit and dev_worn, dev_beamformit is the result of enhanced speech data of array microphone using beamformit, while dev_worn is the result of binaural microphones data. As a matter of fact, WER of dev_worn is listed, but it is not regarded as the challenge result.

Finally, here I give some time spent on training and decoding in Table 4 for reference.

Table 4 Time spent on training and decoding

Pikaia	Beamsize	Time on training(h)	Time on decoding(h)
Pikaia5	20	60.3	7.8
Pikaia10	26	31.5	9.7
Pikaia3	28	45.5	9.8

Reference

[1] *https://github.com/espnet/espnet*

[2] T. Hori, S. Watanabe, Y. Zhang and W. Chan, “Advances in Joint CTC-Attention based End-to-End Speech Recognition with a Deep CNN Encoder and RNN-LM,” in *Interspeech*, 2017, pp. 949-953.

[3] *http://www.speech.sri.com/projects/srilm/*

[4] *http://spandh.dcs.shef.ac.uk/chime_challenge/index.html*