

CONTINUOUS WORD RECOGNITION SYSTEM IN KALDI

Author : Abhishek Dey

Aim : To build a Continuous Word Recognition System in Kaldi

Step 1: Create a working directory inside **kaldi-master/egs** directory. Say **Cont-ASR**.

```
cd kaldi-master/egs
```

```
mkdir Cont-ASR
```

Step 2: Copy the following scripts & folders inside Cont-ASR directory.

- steps
- local
- utils
- conf
- run.sh
- create_Bigram_LM_v1.sh
- Gen_lm_input_file.sh
- cmd.sh
- path.sh
- wav.sh

N.B :Set your **cluster options in cmd.sh** file and change the **KALDI_ROOT in path.sh** file

```
export KALDI_ROOT=/home/Kaldi_Workspace/kaldi-master
```

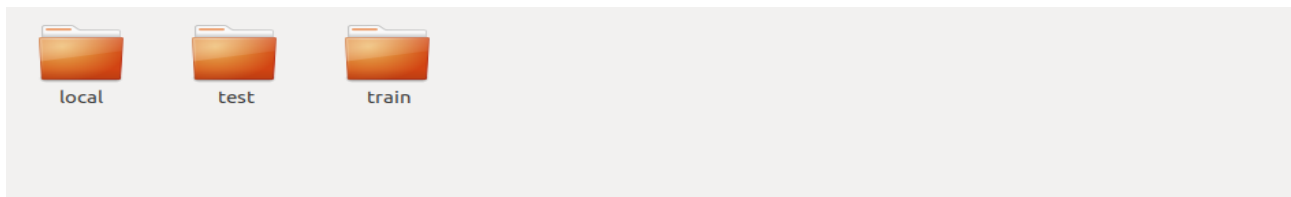
Step 3: Create a directory named **wav** inside Cont-ASR directory. Further create **two** directories named **train** and **test** inside wav directory. Copy all your training .wav files inside train directory and testing .wav files inside test directory.

```
cd Cont-ASR  
mkdir wav  
cd wav  
mkdir train test
```

Step 4: Create a directory named **data** inside Cont-ASR directory. Further create **three** directories inside data directory named **train test** and **local**.

```
cd Cont-ASR  
mkdir data  
cd data  
mkdir train test local
```

And inside **data** directory, it should look like this at this step



Step 5: You need to create 4 files inside **data/train** and **data/test** directories w.r.t training and testing sets.

- **text**—file containing the filename & their corresponding transcriptions

<Filename><tab><transcription>

```
103-1240-0008 THE ORCHARD ON THE SLOPE BELOW THE HOUSE WAS IN ,  
CALLED RACHEL LYNDE'S HUSBAND WAS SOWING HIS LATE TURNIP SEED (  
103-1240-0009 MISSUS RACHEL KNEW THAT HE OUGHT BECAUSE SHE HAD  
TURNIP SEED THE NEXT AFTERNOON
```

- **utt2spk** – utterance to speaker mapping file

<utterance><tab><speaker>

```
103-1240-0004 103  
103-1240-0005 103  
103-1240-0006 103  
103-1240-0007 103  
103-1240-0008 103  
103-1240-0009 103  
103-1240-0010 103  
103-1240-0011 103
```

```
cd data/train
```

Extract utterance list from the text file

```
cat text | awk '{print $1}' > utt
```

Extract corresponding speaker list from the text file

```
cat text | cut -d '-' -f1 > spk
```

Join utt and spk files side by side and create utt2spk file

paste utt spk > utt2spk

- **wav.scp** – file containing the filename & their corresponding wav file path

<Filename><tab><wav file path>

```
103-1240-0002 /home/Kaldi_Workspace/kaldi-master/egs/Cont-ASR/wav/train_wav/103-1240-0002.wav
103-1240-0003 /home/Kaldi_Workspace/kaldi-master/egs/Cont-ASR/wav/train_wav/103-1240-0003.wav
103-1240-0004 /home/Kaldi_Workspace/kaldi-master/egs/Cont-ASR/wav/train_wav/103-1240-0004.wav
103-1240-0005 /home/Kaldi_Workspace/kaldi-master/egs/Cont-ASR/wav/train_wav/103-1240-0005.wav
103-1240-0006 /home/Kaldi_Workspace/kaldi-master/egs/Cont-ASR/wav/train_wav/103-1240-0006.wav
103-1240-0007 /home/Kaldi_Workspace/kaldi-master/egs/Cont-ASR/wav/train_wav/103-1240-0007.wav
```

Open **wav_scp.sh** script

set **wav_path** as path containing the train/test wav files and **data_path** as path of data/train or data/test directories.

```
#!/bin/bash
# Author : Abhishek Dey

wav_path=/home/Kaldi_Workspace/kaldi-master/egs/Cont-ASR/wav/train_wav
data_path=/home/Kaldi_Workspace/kaldi-master/egs/Cont-ASR/data/train
cat $data_path/utt | awk '{printf "%s\t%s%s\n",$1,"'$wav_path'/"$1,".wav"}' > $data_path/wav.scp
```

Then execute this script.

cd Cont-ASR

./wav_scp.sh

- **spk2utt** – speaker to utterance mapping file

<Speaker><space><Utterance><space><Utterance>

```
103-1240-0012 103-1240-0013 103-1240-0014 103-1240-0015 103-1240-0016 103-1240-0017 103-1240-0018 103-1240-0019 103-1240-0020 103-1240-0021 103-1240-0022 103-1240-0023 103-1240-0024
103-1240-0025 103-1240-0026 103-1240-0027 103-1240-0028 103-1240-0029 103-1240-0030 103-1240-0031 103-1240-0032 103-1240-0033 103-1240-0034 103-1240-0035 103-1240-0036 103-1240-0037
103-1240-0038 103-1240-0039 103-1240-0040 103-1240-0041 103-1240-0042 103-1240-0043 103-1240-0044 103-1240-0045 103-1240-0046 103-1240-0047 103-1240-0048 103-1240-0049 103-1240-0050
103-1240-0051 103-1240-0052 103-1240-0053 103-1240-0054 103-1240-0055 103-1240-0056 103-1240-0057 103-1241-0000 103-1241-0001 103-1241-0002 103-1241-0003 103-1241-0004 103-1241-0005
103-1241-0006 103-1241-0007 103-1241-0008 103-1241-0009 103-1241-0010 103-1241-0011 103-1241-0012 103-1241-0013 103-1241-0014 103-1241-0015 103-1241-0016 103-1241-0017 103-1241-0018
103-1241-0019 103-1241-0020 103-1241-0021 103-1241-0022 103-1241-0023 103-1241-0024 103-1241-0025 103-1241-0026 103-1241-0027 103-1241-0028 103-1241-0029 103-1241-0030 103-1241-0031
103-1241-0032 103-1241-0033 103-1241-0034 103-1241-0035 103-1241-0036 103-1241-0037 103-1241-0038 103-1241-0039 103-1241-0040 103-1241-0041 103-1241-0042 103-1241-0043
1034 1034-121119-0000 1034-121119-0001 1034-121119-0002 1034-121119-0003 1034-121119-0004 1034-121119-0005 1034-121119-0006 1034-121119-0007 1034-121119-0008 1034-121119-0009
1034-121119-0010 1034-121119-0011 1034-121119-0012 1034-121119-0013 1034-121119-0014 1034-121119-0015 1034-121119-0016 1034-121119-0017 1034-121119-0018 1034-121119-0019
1034-121119-0020 1034-121119-0021 1034-121119-0022 1034-121119-0023 1034-121119-0024 1034-121119-0025 1034-121119-0026 1034-121119-0027 1034-121119-0028 1034-121119-0029
1034-121119-0030 1034-121119-0031 1034-121119-0032 1034-121119-0033 1034-121119-0034 1034-121119-0035 1034-121119-0036 1034-121119-0037 1034-121119-0038 1034-121119-0039
1034-121119-0040 1034-121119-0041 1034-121119-0042 1034-121119-0043 1034-121119-0044 1034-121119-0045 1034-121119-0046 1034-121119-0047 1034-121119-0048 1034-121119-0049
1034-121119-0050 1034-121119-0051 1034-121119-0052 1034-121119-0053 1034-121119-0054 1034-121119-0055 1034-121119-0056 1034-121119-0057 1034-121119-0058 1034-121119-0059
1034-121119-0060 1034-121119-0061 1034-121119-0062 1034-121119-0063 1034-121119-0064 1034-121119-0065 1034-121119-0066 1034-121119-0067 1034-121119-0068 1034-121119-0069
1034-121119-0070 1034-121119-0071 1034-121119-0072 1034-121119-0073 1034-121119-0074 1034-121119-0075 1034-121119-0076 1034-121119-0077 1034-121119-0078 1034-121119-0079
1034-121119-0080 1034-121119-0081 1034-121119-0082 1034-121119-0083 1034-121119-0084 1034-121119-0085 1034-121119-0086 1034-121119-0087 1034-121119-0088 1034-121119-0089
1034-121119-0090 1034-121119-0091 1034-121119-0092 1034-121119-0093 1034-121119-0094 1034-121119-0095
```

```
cd data/train
```

```
../../utils/utt2spk_to_spk2utt.pl utt2spk > spk2utt
```

Step 6 : Create a dictionary directory (**say dict**) inside data/local directory

- **extra_questions.txt** (This file is kept blank)
- **extra_phones.txt** (This file is kept blank)
- **lexicon.txt** (Word & its phone level break up)

```
ABANDON AH B AE N D AH N
ABANDONED AH B AE N D AH N D
ABANDONING AH B AE N D AH N IH NG
ABANDONMENT AH B AE N D AH N M AH N T
ABASHED AH B AE SH T
ABATED AH B EY T IH D
ABATEMENT AH B EY T M AH N T
ABATEMENTS AH B EY T M AH N T S
ABATING AH B EY T IH NG
ABBAYE AE B EY
ABBE AE B IY
ABBE AE B EY
ABBE'S AE B IY Z
ABBEY AE B IY
ABDUCT AE B D AH K T|
```

- **nonsilence_phones.txt** (All the phones excluding silence)

```
OW
G
DH
OY
TH
UH
ZH
```

- **phones.txt** (All the phones including silence)

```
OW
G
DH
OY
TH
UH
ZH
SIL
```

- **optional_silence.txt**(silence phone)
- **silence_phones.txt** (silence phone including additional fillers such as bgnoise,chnoise)

In our case **optional_silence.txt** & **silence_phones.txt** are same since we haven't used additional fillers such as background noise, channel noise etc.

Step 7: Now we will create n-gram language model

Note that you should be a root user.

- sudo su mode
- You should have a **trans** file inside data/train directory that contains only the transcription excluding the filenames

THE ORCHARD ON THE SLOPE BELOW THE HOUSE WAS IN A BRIDAL FLUSH OF PINKY WHITE BLOOM HUMMED OVER BY A MYRIAD OF BEES THOMAS LYNDE A MEER LITTLE MAN WHOM AVONLEA PEOPLE CALLED RACHEL LYNDE'S HUSBAND WAS SOWING HIS LATE TURNIP SEED ON THE HILL FIELD BEYOND THE BARN
MISSUS RACHEL KNEW THAT HE OUGHT BECAUSE SHE HAD HEARD HIM TELL PETER MORRISON THE EVENING BEFORE IN WILLIAM J BLAIR'S STORE OVER AT CARMODY THAT HE MEANT TO SOW HIS TURNIP SEED THE NEXT AFTERNOON
PETER HAD ASKED HIM OF COURSE FOR MATTHEW CUTHBERT HAD NEVER BEEN KNOWN TO VOLUNTEER INFORMATION ABOUT ANYTHING IN HIS WHOLE LIFE AND YET HERE WAS MATTHEW CUTHBERT AT HALF PAST THREE ON THE AFTERNOON OF A BUSY DAY PLACIDLY DRIVING OVER THE HOLLOW AND UP THE HILL
AND HIS BEST SUIT OF CLOTHES WHICH WAS PLAIN PROOF THAT HE WAS GOING OUT OF AVONLEA AND HE HAD THE BUGGY AND THE SORREL MARE WHICH BETOKENED THAT HE WAS GOING A CONSIDERABLE DISTANCE NOW WHERE WAS MATTHEW CUTHBERT GOING AND WHY WAS HE GOING THERE

- Open the Gen_lm_input_file.sh shell script
- Change the Path according to your directory

```
#!/bin/bash

# Author : Abhishek Dey

path="/home/Kaldi_workspace/kaldi-master/egs/Cont-ASR/data/train"

while read line
do
echo "<s> $line </s>" >> $path/lm_train.txt
done <$path/trans
```

Now Run

Gen_lm_input_file.sh script

./Gen_lm_input_file.sh

This will add sentence markers <s> and </s> and create lm_train.txt file. Using this file, n-gram language model will learn the probabilities.

- Open create_Bigram_LM_v1.sh shell script.
- Change the Path according to your directory

```
basepath='/home/Kaldi_Workspace/kaldi-master/egs/ASR_WORKSHOP_IIT_DHARWAD/Cont-ASR'
lm_arpa_path=$basepath/data/local/lm
train_dict=dict
train_lang=lang_bigram
train_folder=train
training_sw=1

n_gram=2 # This specifies bigram or trigram. for bigram set n_gram=2 for tri_gram set n_gram=3
```

This Creates **G.fst** in **data/lang_bigram** directory. To Check the memory of **G.fst** do the following:

du -hsc data/lang_bigram/G.fst

Step 8: Now open the script. **run.sh** . You need to set the switches.

```

) train_nj=10
) decode_nj=8
.
:
: #=====
: #          SET SWITCHES
: #=====
:
) mfcc_extract_sw=0
)
) mono_train_sw=0
) mono_test_sw=1
.
: tri1_train_sw=0
: tri1_test_sw=1
:
: tri2_train_sw=0
: tri2_test_sw=1
,
) tri3_train_sw=0
) tri3_test_sw=1
)
. sgmm_train_sw=0
: sgmm_test_sw=1
:
: dnn_train_sw=0
: dnn_test_sw=1
:

```

train_nj & **decode_nj** indicate the **number of jobs** during **training** & **decoding** respectively. Here train_nj 10 means the whole job will be divided into 10 parts. Based on the Processor you can change these parameters.

You need to set the directories

```
train_dir1=data/train
train_lang_dir=data/lang_bigram

test_dir1=data/test
test_lang_dir1=data/lang_bigram

graph_dir1=graph_bigram
decode_dir1=decode_bigram
```

At first the script computes mfcc

```
root@ASR: /home/asr/Desktop/kaldi-trunk/egs/workspace
=====
MFCC Feature Extration & CMVN for Training and Test set
=====
steps/make_mfcc.sh --cmd run.pl --nj 3 data/train exp/make_mfcc/train mfcc
Successfully validated data-directory data/train
steps/make_mfcc.sh: [info]: no segments file exists: assuming wav.scp indexed by ut
terance.
```

- Then it performs **Monophone Training & Decoding**

```
Succeeded creating CMVN stats for train
steps/make_mfcc.sh --cmd run.pl --nj 3 data/test exp/make_mfcc/test mfcc
Successfully validated data-directory data/test
steps/make_mfcc.sh: [info]: no segments file exists: assuming wav.scp indexed by ut
terance.
Succeeded creating MFCC features for test
steps/compute_cmvn_stats.sh data/test exp/make_mfcc/test mfcc
Succeeded creating CMVN stats for test
=====
MonoPhone Training & Decoding
=====
steps/train_mono.sh --nj 3 --cmd run.pl data/train data/lang exp/mono
steps/train_mono.sh: Initializing monophone system.
steps/train_mono.sh: Compiling training graphs
steps/train_mono.sh: Aligning data equally (pass 0)
steps/train_mono.sh: Pass 1
steps/train_mono.sh: Aligning data
steps/train_mono.sh: Pass 2
steps/train_mono.sh: Aligning data
```

- Two new folders are created in **workspace** : **exp** & **mfcc**. **exp** contain the training models & decoded outputs.
- The Word Error Rate(WER) files are located in
exp/mono/decode_bigram

- **The decoded outputs are located in**
exp/mono/decode_bigram/log

- **Command to check all the WER**

```
cat/exp/mono/decode_bigram/wer_*| egrep WER | sort -n
```
