KALDI Installation Steps

Author : Abhishek Dey
Designation : Asst. Project Engineer,Dept. of EEE,IIT Guwahati

1. Install Subversion

sudo apt-get install subversion

2. Change Server Settings

cd /etc/subversion

gedit servers

[global]
http-proxy-exceptions = *.
http-proxy-host = 202.141.80.30
http-proxy-port = 3128
http-proxy-username = **PROXY-USERNAME**
http-proxy-password = **PROXY-PASSWORD**

gedit /etc/wgetrc

https_proxy = http://202.141.80.30:3128/
http_proxy = http://202.141.80.30:3128/
ftp_proxy = http://202.141.80.30:3128/
proxy_user=**PROXY-USERNAME**
proxy_password=**PROXY-PASSWORD**

3. Install Dependent Libraries

sudo apt-get install libblas-*
sudo apt-get install automake
sudo apt-get install libtool-*
sudo apt-get install libatlas-*
sudo apt-get install zlib1g-dev
sudo apt-get install g++-multilib

4. Create a Kaldi workspace in Home

cd /home

mkdir Kaldi-Workspace

cd Kaldi-Workspace

5. Download Kaldi Package

Paste the following link in the terminal

```
svn co https://svn.code.sf.net/p/kaldi/code/trunk kaldi-trunk
```

6. Extract kaldi.tar.gz

```
cd  /kaldi-trunk /tools/

  make

cd /kaldi-trunk /src/

  ./configure
  make depend
  make
```

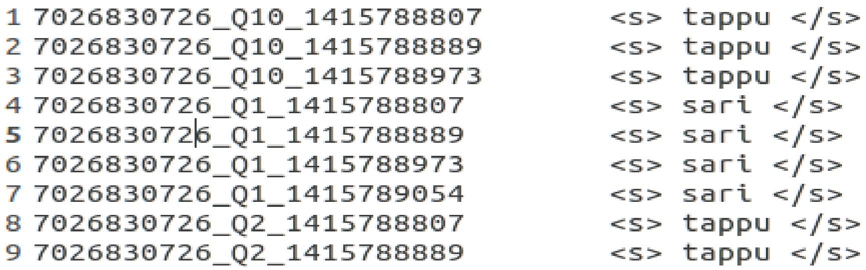
KALDI System Building from Scratch

- It is assumed that the user have only **wav files &** corresponding **transcription in kaldi format**. Rest other files will be created in the process.

Format of Trancription :

<Wav_File><tab><s><space><transcription><space></s>

- Make sure that there are no extra spacings in the transcription file. Save the transciption file as **text**

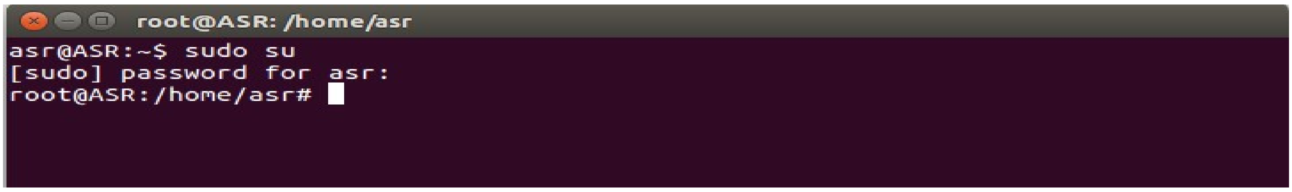


```
7026830726_Q10_1415788807        <s> tappu </s>
7026830726_Q10_1415788889        <s> tappu </s>
7026830726_Q10_1415788973        <s> tappu </s>
7026830726_Q1_1415788807         <s> sari </s>
7026830726_Q1_1415788889         <s> sari </s>
7026830726_Q1_1415788973         <s> sari </s>
7026830726_Q1_1415789054         <s> sari </s>
7026830726_Q2_1415788807         <s> tappu </s>
7026830726_Q2_1415788889         <s> tappu </s>
```

1. Open Terminal

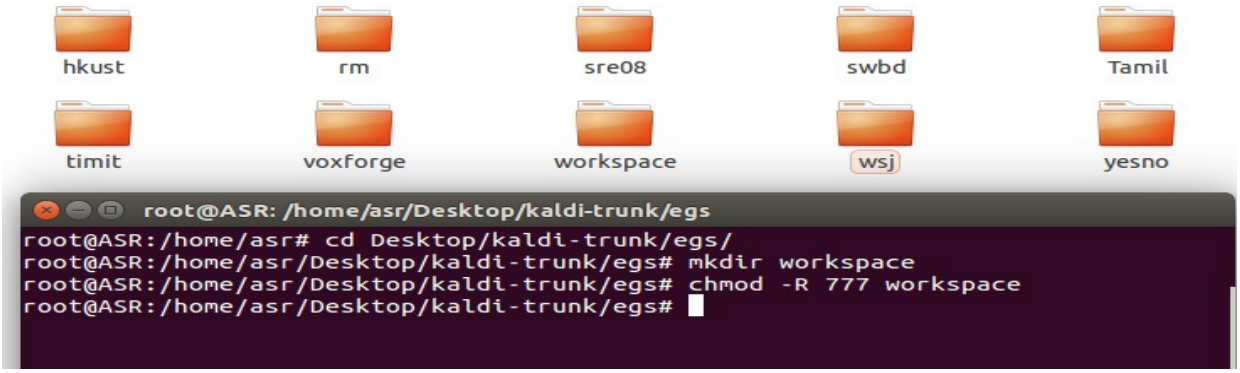
ctrl+Alt+t

2. Go to super user mode

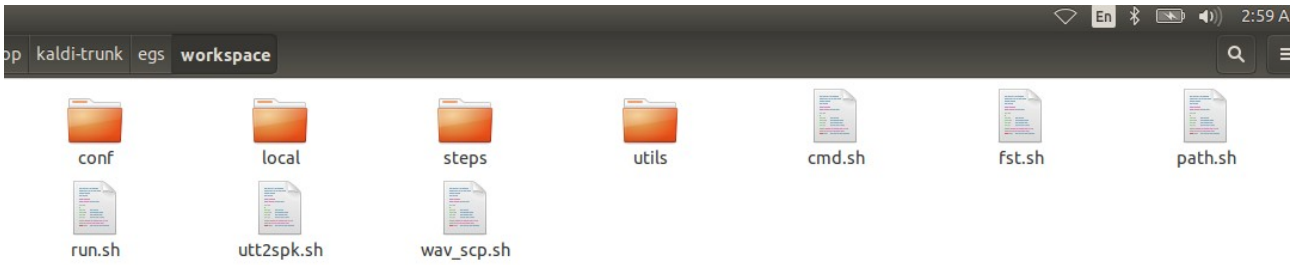


```
asr@ASR:~$ sudo su
[sudo] password for asr:
root@ASR:/home/asr#
```

3. Go to **kaldi-trunk/egs/** directory and create your **workspace**



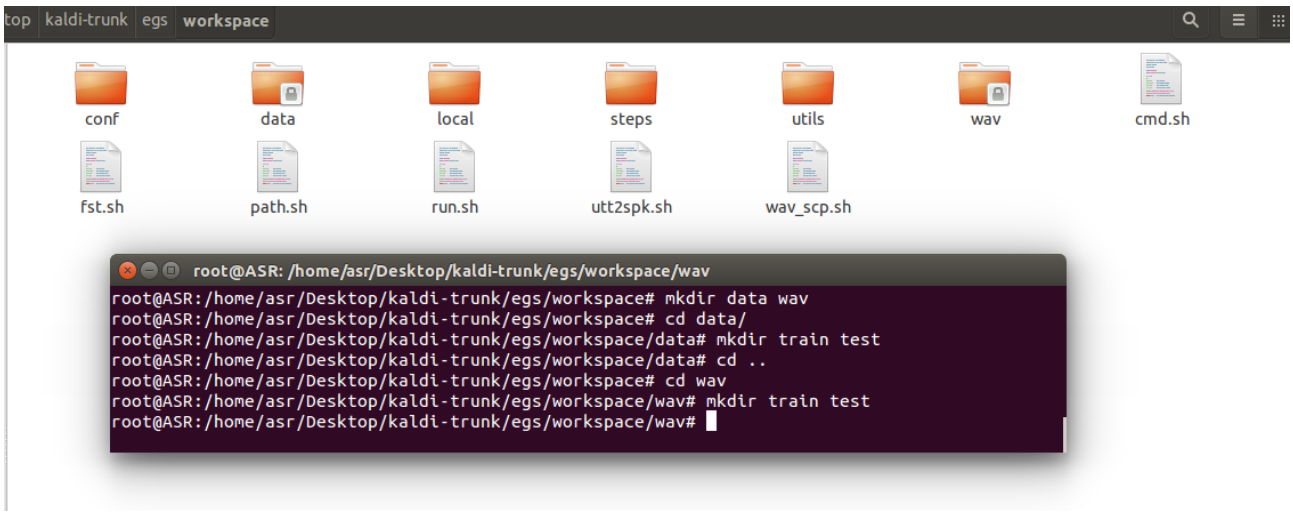
4. Copy conf,local,steps,utils folders in workspace
5. Copy the Shell scripts in workspace



6. Create two directories in **workspace** -> **data** & **wav**

a) Inside **data** create two subdirectories -> **train** & **test**
b) Inside **wav** create two subdirectories -> **train** & **test**

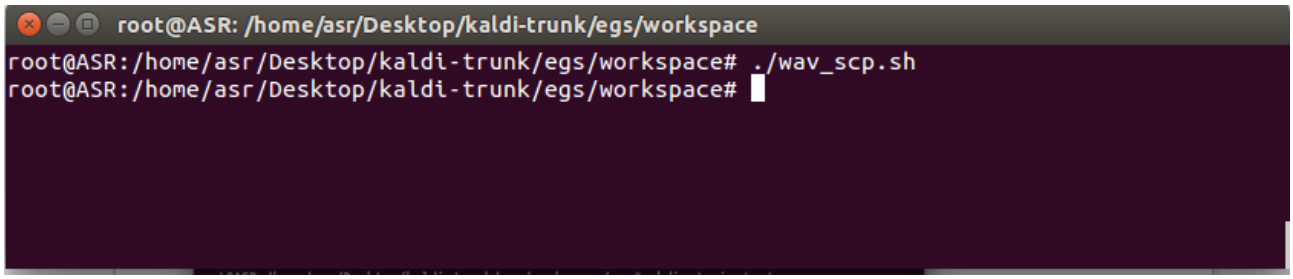
mkdir data wav
cd data
mkdir train test
cd ..
cd wav
mkdir train test
cd ..
chmod -R 777 data wav



7. Copy train & test trancriptions **(text)** in **data/train** & **data/test** respectively.

8. Copy train & test wav files in **wav/train** & **wav/test** respectively.

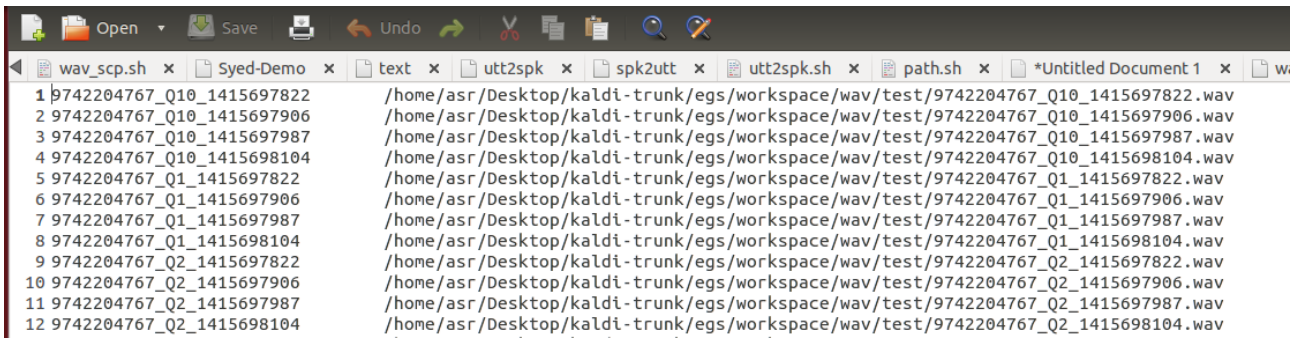
9. Run **wav_scp.sh** script



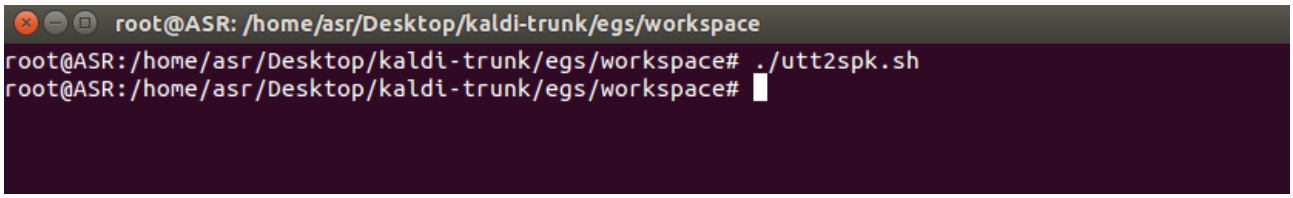
This script creates **train wav.scp** in **data/train** & **test wav.scp** in **data/test** directories.

Format of wav.scp file :

<wavfile><tab><wav_path/wavfile.wav>



10. Run **utt2spk.sh** script



This script creates **utt2spk** in **data/train** & **utt2spk** in **data/test** directories.

N.B : Change the deliminator as per your requirement in the script. In our case it was '_'

Format of utt2spk file :

<Utterance><tab><Speaker>

1	9742204767_Q10_1415697822	9742204767
2	9742204767_Q10_1415697906	9742204767
3	9742204767_Q10_1415697987	9742204767
4	9742204767_Q10_1415698104	9742204767
5	9742204767_Q1_1415697822	9742204767
6	9742204767_Q1_1415697906	9742204767
7	9742204767_Q1_1415697987	9742204767
8	9742204767_Q1_1415698104	9742204767
9	9742204767_Q2_1415697822	9742204767
10	9742204767_Q2_1415697906	9742204767

11. Now we need to create **spk2utt** files.

cd data/train

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

cd ../test

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

```
root@ASR: /home/asr/Desktop/kaldi-trunk/egs/workspace/data/test
root@ASR:/home/asr/Desktop/kaldi-trunk/egs/workspace/data/train# ../../utils/utt2spk_to_spk2utt.pl utt2spk > spk2utt
root@ASR:/home/asr/Desktop/kaldi-trunk/egs/workspace/data/train# cd ../test/
root@ASR:/home/asr/Desktop/kaldi-trunk/egs/workspace/data/test# ../../utils/utt2spk_to_spk2utt.pl utt2spk > spk2utt
root@ASR:/home/asr/Desktop/kaldi-trunk/egs/workspace/data/test#
```

This script creates **spk2utt** in **data/train** & **spk2utt** in **data/test** directories.

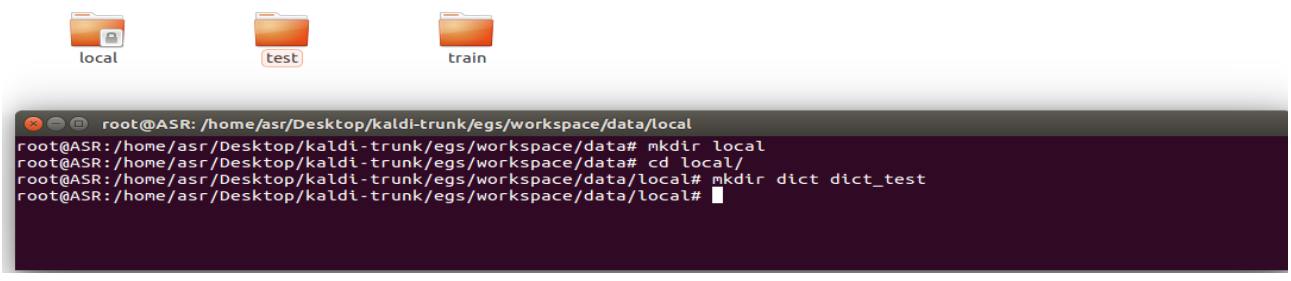
Format of spk2utt file :

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

```
1 7026830726 7026830726_Q10_1415788807 7026830726_Q10_1415788889 7026830726_Q10_1415788973 7026830726_Q1_1415788807 7026830726_Q1_1415788889
  7026830726_Q1_1415788973 7026830726_Q1_1415789054 7026830726_Q2_1415788807 7026830726_Q2_1415788889 7026830726_Q2_1415788973
  7026830726_Q2_1415789054 7026830726_Q3_1415788807 7026830726_Q3_1415788889 7026830726_Q3_1415788973 7026830726_Q3_1415789054
  7026830726_Q4_1415788807 7026830726_Q4_1415788889 7026830726_Q4_1415788973 7026830726_Q4_1415789054 7026830726_Q5_1415788807
  7026830726_Q5_1415788889 7026830726_Q5_1415788973 7026830726_Q5_1415789054 7026830726_Q6_1415788807 7026830726_Q6_1415788889
  7026830726_Q6_1415788973 7026830726_Q7_1415788807 7026830726_Q7_1415788889 7026830726_Q7_1415788973 7026830726_Q8_1415788807
  7026830726_Q8_1415788889 7026830726_Q8_1415788973 7026830726_Q9_1415788807 7026830726_Q9_1415788889 7026830726_Q9_1415788973
2 7204582600 7204582600_Q10_1415677945 7204582600_Q10_1415678028 7204582600_Q10_1415678111 7204582600_Q10_1415678194 7204582600_Q10_1415678530
  7204582600_Q1_1415677945 7204582600_Q1_1415678028 7204582600_Q1_1415678111 7204582600_Q1_1415678194 7204582600_Q1_1415678277
  7204582600_Q1_1415678530 7204582600_Q2_1415677945 7204582600_Q2_1415678028 7204582600_Q2_1415678111 7204582600_Q2_1415678194
  7204582600_Q2_1415678277 7204582600_Q2_1415678530 7204582600_Q3_1415677945 7204582600_Q3_1415678028 7204582600_Q3_1415678111
  7204582600_Q3_1415678194 7204582600_Q3_1415678277 7204582600_Q3_1415678530 7204582600_Q4_1415677945 7204582600_Q4_1415678028
  7204582600_Q4_1415678111 7204582600_Q4_1415678194 7204582600_Q4_1415678277 7204582600_Q4_1415678530 7204582600_Q5_1415677945
  7204582600_Q5_1415678028 7204582600_Q5_1415678111 7204582600_Q5_1415678194 7204582600_Q5_1415678277 7204582600_Q5_1415678530
  7204582600_Q6_1415677945 7204582600_Q6_1415678028 7204582600_Q6_1415678111 7204582600_Q6_1415678194 7204582600_Q6_1415678277
  7204582600_Q6_1415678530 7204582600_Q7_1415677945 7204582600_Q7_1415678028 7204582600_Q7_1415678111 7204582600_Q7_1415678194
  7204582600_Q7_1415678277 7204582600_Q7_1415678530 7204582600_Q8_1415677945 7204582600_Q8_1415678028 7204582600_Q8_1415678111
  7204582600_Q8_1415678194 7204582600_Q8_1415678277 7204582600_Q8_1415678530 7204582600_Q9_1415677945 7204582600_Q9_1415678028
  7204582600_Q9_1415678111 7204582600_Q9_1415678194 7204582600_Q9_1415678277 7204582600_Q9_1415678530
```

12. Create a directory **local** inside **data**. Inside **local**, create two sub-directories : **dict** & **dict_comm**

cd data
mkdir local
cd local
mkdir dict dict_comm



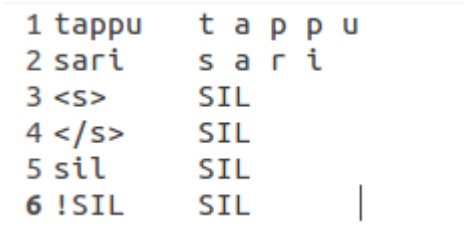
dict contains the lexicon & other files related to training data & **dict_comm** contains the same related to testing data

Create the following files in **dict** & **dict_comm** respectively:

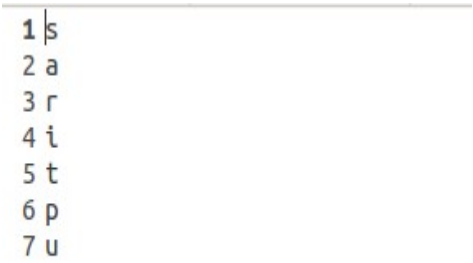
extra_questions.txt (This file is kept blank)

extra_phones.txt (This file is kept blank)

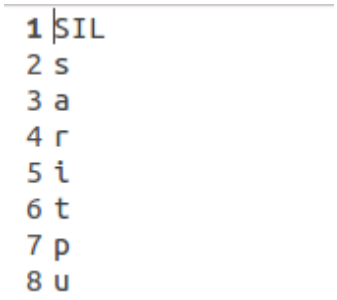
lexicon.txt (Word & its phone level break up)



nonsilence_phones.txt (All the phones excluding silence)



phones.txt (All the phones including silence)



optional_silence.txt(silence phone)

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

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

13. Now we will create language models

Note that you shoud be a root user.

Sudo su mode

Now open **fst.sh** script & Modify the paths as per your requirement

```
#!/bin/bash

# Modified Script by Syed


lang_dir=/home/asr/Desktop/kaldi-trunk/egs/workspace/data/lang_test
tmp_dir=/home/asr/Desktop/kaldi-trunk/egs/workspace/data/local/tmp

```

Now Run
<DOT><SPACE>path.sh
./ fst.sh

```
root@ASR:/home/asr/Desktop/kaldi-trunk/egs/workspace# . path.sh
root@ASR:/home/asr/Desktop/kaldi-trunk/egs/workspace# ./fst.sh
```

This should give a **SUCCESS** Message after running.

```
Checking word_boundary.int and disambig.int
--> generating a 91 words sequence
--> resulting phone sequence from L.fst corresponds to the word sequence
--> L.fst is OK
--> generating a 98 words sequence
--> resulting phone sequence from L_disambig.fst corresponds to the word sequence
--> L_disambig.fst is OK

Checking /home/asr/Desktop/kaldi-trunk/egs/Tamil_demo/data/lang_test/oov.{txt, int} ...
--> 1 entry/entries in /home/asr/Desktop/kaldi-trunk/egs/Tamil_demo/data/lang_test/oov.txt
--> /home/asr/Desktop/kaldi-trunk/egs/Tamil_demo/data/lang_test/oov.int corresponds to /home/asr/Desktop/kaldi-trunk/eg
s/Tamil_demo/data/lang_test/oov.txt
--> /home/asr/Desktop/kaldi-trunk/egs/Tamil_demo/data/lang_test/oov.{txt, int} are OK

--> SUCCESS
root@ASR:/home/asr/Desktop/kaldi-trunk/egs/workspace#
```

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

du -hsc data/lang_test/G.fst

```
root@ASR: /home/asr/Desktop/kaldi-trunk/egs/workspace
root@ASR:/home/asr/Desktop/kaldi-trunk/egs/workspace# du -hsc data/lang_test/G.fst
4.0K    data/lang_test/G.fst
4.0K    total
root@ASR:/home/asr/Desktop/kaldi-trunk/egs/workspace#
```

14. 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

```
#================================================
# Set Directories
#================================================


train_dir1=data/train
train_lang_dir=data/lang_train

test_dir1=data/test
test_lang_dir1=data/lang_test

graph_dir1=graph_test
decode_dir1=decode_test
```

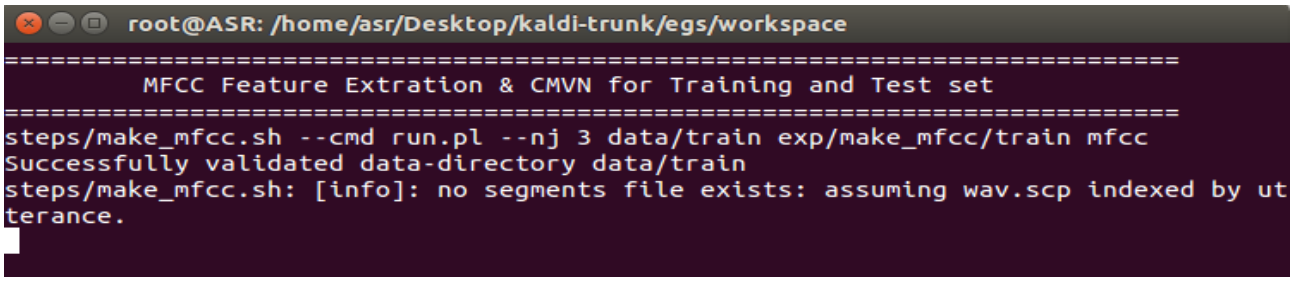
```
echo ============================================================================
echo "                     MonoPhone Training & Decoding                        "
echo ============================================================================

train_nj=3
decode_nj=3

steps/train_mono.sh  --nj "$train_nj" --cmd "$train_cmd" data/train data/lang exp/mono || exit 1;

utils/mkgraph.sh --mono data/lang_test exp/mono exp/mono/graph || exit 1;

steps/decode.sh --nj "$decode_nj" --cmd "$decode_cmd" \
        exp/mono/graph data/test exp/mono/decode || exit 1;
```

```
57
58 echo ============================================================================
59 echo "           tri1 : Deltas + Delta-Deltas Training & Decoding               "
60 echo ============================================================================
61
62 steps/align_si.sh --boost-silence 1.25 --nj "$train_nj" --cmd "$train_cmd" data/train data/lang exp/mono exp/mono_ali || exit 1;
63
64 for sen in 450; do      # No. of Senons
65 for gauss in 4; do      # No. of Gaussians
66 gauss=$(($sen * $gauss))
67 steps/train_deltas.sh --cmd "$train_cmd" $sen $gauss data/train data/lang exp/mono_ali exp/tri1_${sen}_${gauss} || exit 1;
68
69 utils/mkgraph.sh data/lang_test exp/tri1_${sen}_${gauss} exp/tri1_${sen}_${gauss}/graph || exit 1;
70
```

In this part you can change the number of **Senons** & **Gaussians**

After setting the switches & directories run the script.

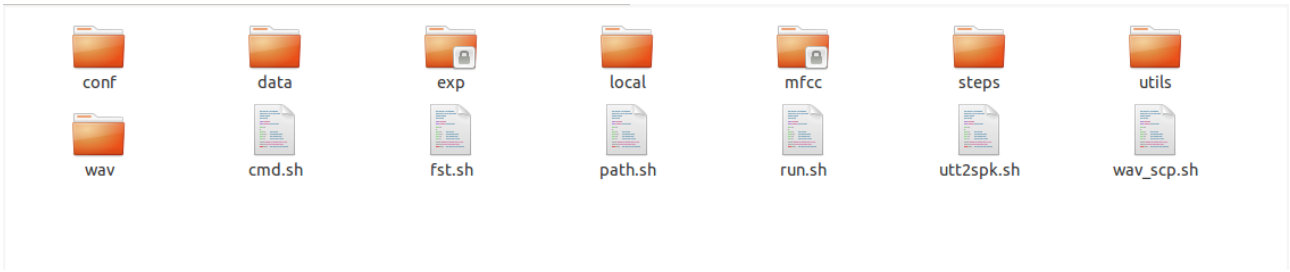
At first it creates mfcc



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
```

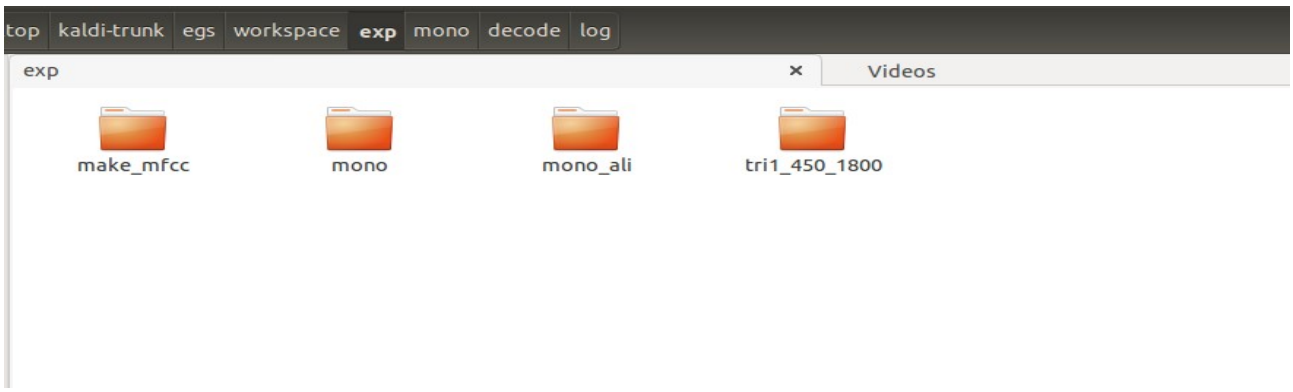
Two new folders are created in **workspace** : **exp** & **mfcc. exp contain the training models & decoded outputs.**



Then it performs **Triphone Training & Decoding**

```
root@ASR: /home/asr/Desktop/kaldi-trunk/egs/workspace
          tri1 : Deltas + Delta-Deltas Training & Decoding
============================================================================
steps/align_si.sh --boost-silence 1.25 --nj 3 --cmd run.pl data/train data/lang exp
/mono exp/mono_ali
steps/align_si.sh: feature type is delta
steps/align_si.sh: aligning data in data/train using model from exp/mono, putting a
lignments in exp/mono_ali
steps/align_si.sh: done aligning data.
steps/train_deltas.sh --cmd run.pl 450 1800 data/train data/lang exp/mono_ali exp/t
ri1_450_1800
steps/train_deltas.sh: accumulating tree stats
steps/train_deltas.sh: getting questions for tree-building, via clustering
steps/train_deltas.sh: building the tree
```

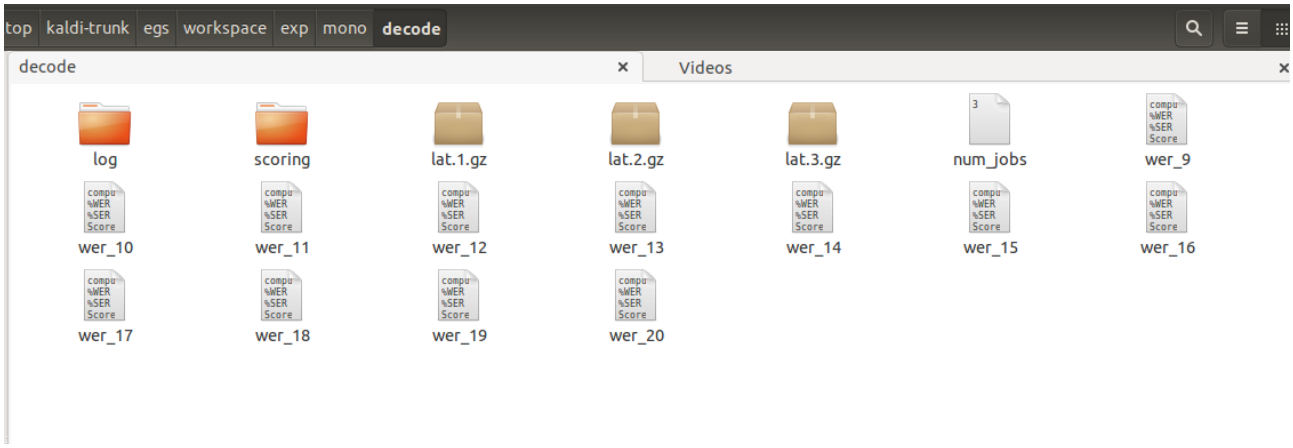
Inside exp the following folders are created :



The Word Error Rate(WER) files are located in

exp/mono/decode

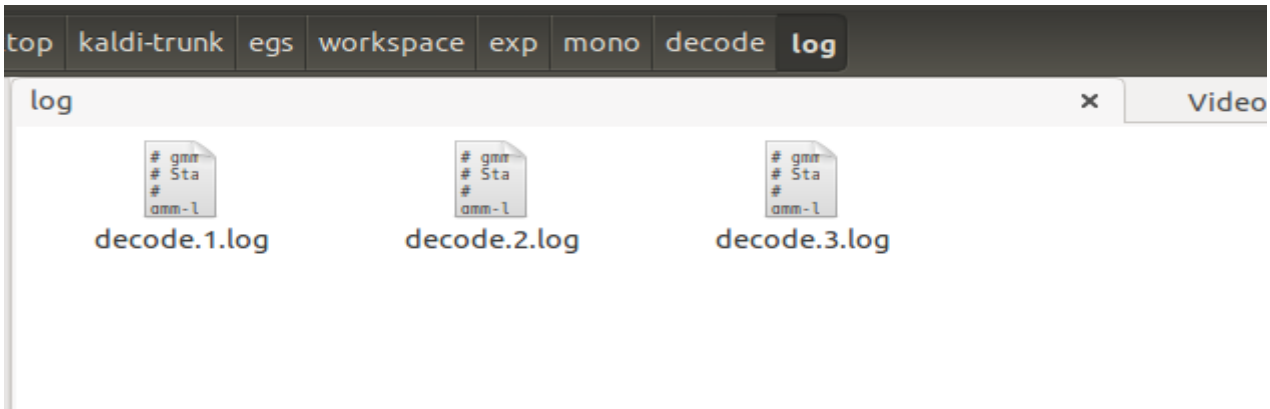
exp/tri1/decode



The decoded outputs are located in

exp/mono/decode/log

exp/tri1/decode/log



****