ASR Tutorial 1
Speech Recognition using Sphinx3

1 Introduction

Automatic speech recognition is a process of converting a spoken utterance to text. In this tutorial we use a set of tools developed at Carnegie Mellon University to set up a speech recognition system. The speech decoder can be set up using the sphinx3 and the sphinxbase tools.

The recognition task is performed by the decode script provided by sphinx3 (sphinx3 requires some of the libraries from sphinxbase). Some set of input parameters are to be given to the sphinx3 decoder. The input parameters include paths of the files and directories given below. In this tutorial we would assume that all the existing files are already provided to us, i.e we would be using some already available models and dictionary files.

1. Acoustic Models
2. Language Model file
3. Word Dictionary
4. Filler Dictionary

The tutorial is structured into three sections: [2]Setting up the Workspace which gives an outline on what all packages are required along with their download links and also setting up the install directories, [3]Installations which gives guidelines for installing the packages and [4]Recognition which explains the scripts and the parameters required for running the sphinx3 decoder.

2 Setting up Workspace

Assume that all the installations and setting up the tools is done in the following directory */unused/asrws/workspace*. Create the following directories in the workspace as follows:

```
$ cd ~
$ mkdir workspace
$ cd workspace
$ mkdir downloads source tools
```

Download the sphinx3 and sphinxbase packages from the given links into the downloads folder in the workspace:
sphinx3 download
sphinxbase download

3 Installations

3.1 Data Preparation

Now extract the sphinx3-0.6.3.tar.gz and sphinxbase-0.6.tar.gz files and move them into the source directories as shown below

```
$ cd downloads
$ tar -xzvf sphinx3-0.6.3.tar.gz
$ tar -xzvf sphinxbase-0.6.tar.gz
$ mv sphinx3-0.6 sphinxbase-0.6 ../source/
```

3.2 Installation of SphinxBase

Go into the source folder for the installations. The installations is done via three steps - configure, make and then make install. The steps for the installations for both the packages is shown below. It is necessary to install the sphinxbase before installing the sphinx3. Before installing create two directories in *tools/* namely sphinx3 and sphinxbase

```
$ cd ../source
$ mkdir ../tools/sphinx3
$ mkdir ../tools/sphinxbase
$ cd sphinxbase-0.6
$ ./configure  --prefix=/unused/asrws/workspace/tools/sphinxbase
$ make
$ make install
```

In case of errors due to the python site-packages folder you can configure the installation without python if you only want to run a simple decoder as follows:

```
$ ./configure --prefix=/unused/asrws/workspace/tools/sphinxbase  --without-python
$ make
$ make install
```

3.3 Installation of Sphinx3

After the installation of sphinxbase install the sphinx3 as follows:

```
$ cd ../sphinx3-0.6
$ ./configure --prefix=/unused/asrws/workspace/tools/sphinx3 --with-sphinxbase=../
sphinxbase-0.6
$ make
make install
```

3.4 Defining ENV Variables

Once the installation is complete add the path of the sphinx3 and the sphinxbase build directories to the variables SPHINXDIR and SPHINXBASE as follows in the .bashrc file

1. $ vim ~/.bashrc
2. add the following line at the end of the file:
   export SPHINXDIR=/unused/asrws/workspace/tools/sphinx3
   export SPHINXBASE=/unused/asrws/workspace/tools/sphinxbase and then save and exit.
3. $ source ~/.bashrc

4 Recognition

4.1 Data Preparation

Sphinx3 decoder would also requires some more files which include acoustic models, language model, word dictionary and filler dictionary. We would also need some test samples to run the recognition. Test samples and the language model and the dictionary can be obtained from an4 data resource which is freely available. Download the an4 raw files from the below link into the *downloads/* folder and then extract the contents of the folder with the tar command: an4 raw database

```
$ cd ~/workspace/downloads
$ tar -xzvf an4_raw.littleendian.tar.gz
```

Create another directory called *decode/* in the *workspace/* directory with the following directory structure

```
$ cd ~/workspace
$ mkdir decode
$ mkdir decode/models decode/feats decode/wav
$ mkdir decode/models/hmm decode/models/lm
```

4.2 Acoustic Models

We place the acoustic models into the *decode/models/hmm/* directory and the language model files and the dictionary into the *decode/models/lm/* directory. Sphinx3 comes with some sample acoustic model files. Hub4 acoustic models are copied into the *decode/models/hmm/* as follows

```
$ cd ~/workspace/source/sphinx3-0.6/model/hmm/hub4_cd_continuous_8gau_1s_c_d_d
$ cp mdef means mixture_weights transition_matrices variances ~/workspace/decode/models/hmm
```

4.3 Language Model, Word, Filler Dictionaries and Test Audio Data

Now copy the language model and the dictionary files into its corresponding folder in the *decode/* folder. The an4 database that we have downloaded provides a language model and the dictionary files along with audio files (raw format) for decoding purpose. The following files are copied as shown

```
$ cd ~/workspace/downloads/an4/etc
$ cp an4.dic an4.filler an4.ug.lm an4.ug.lm.DMP ~/workspace/decode/models/lm
$ cd ../wav/an4_clstk/fash
$ cp *.raw ~/workspace/decode/wav
```

The audio files can be played using the play command as shown below

```
$ play -t raw -r 16000 -s -w filename.raw
```

4.4 Sphinx3 Decode

Now all the necessary files have been copied into the decode directory. The decoding is done in two steps. Firstly we do feature extraction on the raw files and then perform the decode process on the feature files. sphinxbase provides scripts to extract the feature files. These are called the mfc files. We need to create a file called *raw_files* which consists the list of raw files for which we doing feature extraction. *raw_files* consists of only the file names (not the complete path) without the file extension (i.e we write the file entry as an254-fash-b for the file an254-fash-b.raw). The feature extraction is done as follows

```
$ cd ~/workspace/decode
$ $SPHINXBASE/bin/sphinx_fe -c raw_files -di wav/ -do feats/ -ei raw -raw yes
-eo mfc
```

Once the feature files are extracted, we create another file called *feat.ctl* which consists name of the feature files without their extensions. After creating the *feat.ctl* we then use *sphinx3_decode* for recognition. The steps for decoding is shown below:

```
$ $SPHINXDIR/bin/sphinx3_decode -hmm models/hmm/ -lm models/lm/an4.ug.lm.DMP
-dict models/lm/an4.dic -fdict models/lm/an4.filler -ctl feat.ctl -logfn log.txt
-hyp out.txt -cepdir feats
```

The decoded output is written into a text file *out.txt*. Another file would be also created called *log.txt*. This file consists of the log of the *sphinx3_decode*. It is an optional file but it is always good to have a output log file.

5 Summary

1. This tutorial is to get hands on experience on the sphinx3 decoder. We assume that we already have all the required acoustic model files, language model files and the dictionaries.
2. We have used existing Hub4 models in the sphinx3 source directory.
3. Language model and the dictionary files were obtained from the an4 database which is freely available online.
4. Sample audio files have been taken from the an4 database for decoding purpose.
5. User may some times required to create his or her own acoustic models and language models based on application at hand. More about this would be covered in Tutorial 2.