NATIONAL TECHNICAL UNIVERSITY OF ATHENS
SCHOOL OF ELECTRICAL & COMPUTER ENGINEERING
Speech & Language Processing
Summer Semester
2nd Preparatory Lab: Speech Recognition with Kaldi toolkit

1 Περιγραφή

The purpose of this exercise is to implement a speech processing and recognition system with the Kaldi toolkit, which is widely used in research and industry for the training of state-of-the-art speech recognition systems.

More specifically, the system you will develop concerns Phone Recognition using USC-TIMIT recordings. You will be given audio data from 4 different speakers, with the corresponding transcriptions, to train and evaluate your system.

The system design process can be divided into 4 parts.

The first part aims to extract appropriate acoustic features from the voice data (Mel-Frequency Cepstral Coefficients). These features are essentially a number of cepstrum coefficients that are extracted after analyzing the signals with a specially designed filter array (Mel filterbank). This array is inspired by the non-linear way in which the human ear perceives sound and is inspired by psychoacoustic studies.

The second part concerns the creation of language models from the transcripts of the data set, which will give an a-priori bias to the final system.

The third part concerns the training of acoustic models using the acoustic features that have been exported.

Finally, by combining the above units, the final speech recognition system can be constructed, which, given a voice signal, it produces the audio features and uses them to decode the signal into a sequence of sounds or words or phonemes.

2 Background

During the preparation you should get acquainted with specific concepts that will be used during the lab. Specifically, we would like you to know about the following concepts:

1. Mel-frequency Cepstral Coefficients (MFCCs)
2. Language Models
3. Acoustic Models

In your final report, you are tasked to briefly develop the above concepts and comment on their performance. Do not stay in the steps of the basic algorithms but try to suggest / evaluate how the voice recognition system you have developed could be improved.

In preparation for the workshop you can refer to the following material:

- Chapter 14 of the textbook **[R & S] Theory and Applications of Digital Speech Processing** of Lawrence R. Rabiner and Ronald W. Schafer (Pearson, 2011), on Automatic Speech Recognition (ASR) systems.
- Mel-frequency Cepstral Coefficients (MFCCs)
- GMM-HMM for acoustic modeling
- Kaldi tutorial 1
- Kaldi tutorial 2
- Kaldi tutorial 3

3 Preparatory Steps

1. Install Kaldi on your computer according to the instructions given in https://helios.ntua.gr/.

2. Familiarize yourself with the Kaldi tool. The language in which it was developed is C++, but the main functions we are interested in are called from bash scripts. There exist implemented recipes for speech recognition models for common datasets inside the Kaldi *egs* folder. For the dataset we will provide you will have to mix existing scripts from the base recipe (*wsj*) with your own to create a new recipe.

3. Download the data from the following link:

   `https://drive.google.com/file/d/1_mIoioHMeC2HZtIbGs1LcL4kkIF696nB/view?usp=sharing`

   The data includes recordings from 4 speakers with names: m1, m3 (male) and f1, f5 (female). Each speaker has produced 460 utterances (Note: speaker m1 lacks utterances 231 to 235 due to a recording error).

   The audio files are located in the *wav* folder. Filenames are prefixed by speaker name and followed by the utterance id. In the file *transcription.txt* you will find the text spoken by the speakers in each utterance (1st line → 1st utterance, 2nd line → 2nd utterance etc.) and in the folder *filesets* you will find which utterances correspond to the training set, the validation set and the testing set (training, validation, testing).

4. Recipe construction:

   - Inside the *egs* folder, create a *usc* subfolder, which will be your working directory.
   - Create the folder *data* and the subfolders *data/train*, *data/dev*, *data/test* inside *usc*, in which you will create index files that will describe the training, validation and testing data respectively.
   - In each of these 3 folders you must create the following files:
     - *uttids*: contains in each line a unique symbolic name for each utterance in the corresponding training/validation/tes subset (i.e. the contents of the files in the folder *filesets*) which from now on will be referred to as utterance_ids
     - *utt2spk*: contains on each line the speaker that corresponds to each utterance and is of the form:

       utterance_id_1 <*space*>*speaker_idutterance_id*_2 <space> speaker_id etc.

       where as speaker_id we select m1, m3, f1, f5 respectively
     - *wav.scp*: contains the location of the audio file that corresponds to each sentence and is of the format:

       utterance_id_1 <space> /path/to/wav1
       utterance_id_2 <space> /path/to/wav2
       etc.
     - *text*: contains the text that corresponds to each utterance and is of the form:

       utterance_id_1 <space> <utterance 1 text>
       utterance_id_2 <space> <utterance 2 text>
       etc.
   - Finally, for each *text* file you create, you must replace the words in the sentences with the corresponding phoneme sequences. For this reason you are given along with the rest of the data the dictionary (lexicon.txt), which maps each word of the English language to the corresponding sequence of phonemes. Be careful in this step to first convert all characters to lower case, as well as remove special characters (e.g. periods, dashes, etc.) except for single quotes ('). Also, at the beginning and at the end you have to add the silence phoneme (sil). The following example is given for the 1st utterance of speaker f1 (f1_001):

     This was easy for us.

     Will be converted to:

     sil dh ih s w ao z iy z iy f r er ah s sil