Kaldi ToolKit

A Brief Overview

Contents

Session 1:

1. Introduction
2. Why Kaldi ?? Advantages of Kaldi over different toolkits
3. Kaldi supported Algorithms
4. Acoustic Models
5. Language model overview
6. Training a GMM-HMM model with Mini librispeech dataset
7. Force Alignment
8. Performance Evaluation

Session 2:

1. GPU configuration (CUDA installation)
2. Training a DNN-Hmm & TDNN model
3. Distributed Computing in GPU
4. Testing of DNN & TDNN models
5. Performance Evaluation

Introduction

Kaldi is a open-source toolkit written in C++. It is wrapped with Bash and Python scripts.

It focuses mainly on Speech Recognition. It is also used in the development of other tasks such as Speaker Recognition and Speaker Diarization.

Kaldi was developed by Daniel Povey and others.

Why Kaldi ???

Features like speaker identification, gmm-based VAD etc are easy to implement in Kaldi as compared to HTK.

Various recipes available in Kaldi. Unpack the data in the right place and you are good to go!!!

Kaldi has a huge supportive community. Help-forum to address issues.

The license is a bane for people from using HTK in commercial software.

Algorithms supported

Kaldi provides support for different preprocessing algorithms like Feature extraction (MFCC,PLP), Voice Activity Detection (VAD).

For development of Automatic Speech Recognition (ASR), Gaussian Mixture Model (GMM), Hidden Markov Model (HMM), Deep Neural Networks (DNN), Time delay Neural network(TDNN), Long-short term Memory (LSTM), Recurrent Neural networks (RNN) etc are available.

For Speaker Recognition, algorithms like I-Vectors, X-Vector are also available.

Acoustic Models

Acoustic model is a file that contains statistical representations of each of the distinct sounds that makes up a word. Each of these statistical representations is assigned a label called a phoneme.

The English language has about 40 distinct sounds that are useful for speech recognition, and thus we have 40 different phonemes.

An acoustic model is created by taking a large database of speech and using special training algorithms to create statistical representations for each phoneme in a language.

These statistical representations are called Hidden Markov Models ("HMM"s). Each phoneme has its own HMM.

For example, if an ASR system has to recognize the word "Education " (whose phonemes are: "eh jh ah k ey sh ah n"), here are the (simplified) steps that the speech recognition engine might take:

The speech decoder listens for the distinct sounds spoken by a user and then looks for a matching HMM in the Acoustic Model. In our example, each of the phonemes in the word education has its own HMM:

eh jh ah k ey sh ah n

- When it finds a matching HMM in the acoustic model, the decoder takes note of the phoneme. The decoder keeps track of the matching phonemes until it reaches a pause in the users speech.
- When a pause is reached, the decoder looks up the matching series of phonemes it heard (i.e. "eh jh ah k ey sh ah n") in its Pronunciation Dictionary to determine which word was spoken. In our example, one of the entries in the pronunciation dictionary is EDUCATION:
    - ...
    - EDUCATES EH JH Y UW K EY T S
    - EDUCATING EH JH AH K EY T IH NG
    - EDUCATION EH JH AH K EY SH AH N
    - ...

The decoder then looks in the Grammar file for a matching word or phrase. Since our grammar in this example only contains one word ("EDUCATION"), it returns the word "EDUCATION".

Language Model

Used to constrain the search in a decoder by limiting the number of possible words. Consequence is faster execution and higher accuracy.

LM constrain search probabilistically (by computing a likelihood for each possible successor word).

Kaldi supports different language models like IRSTLM, SRILM, Kaldi-LM etc.

GMM-HMM Training

Data preparation

Basic files required by Kaldi:

- **Text** - Contains the transcriptions of each utterance.
  spk001_utt001 I AM LEARNING KALDI
- **Utt-** Contains utterance id's of all speakers
  utt001
  utt002 ...
- **Spk** - Contains speaker id's of all speakers
  spk001
  spk002 ...
- **utt2spk** - Maps the utterance id's to the speaker.
  utt001 spk001
  utt002 spk001
- **spk2utt** - Maps the speaker id to the utterance id's.
  spk001 utt001 utt002 ...

Dictionary Preparation

Create a dictionary (say dict) inside data/local directory

- extra_questions.txt
- lexicon.txt (word & its phone level break up)
- nonsilence_phones.txt (all the phones excluding silence)
- optional_silence.txt (silence phone)
- silence_phones.txt (silence phone including additional fillers such as bgnoise, chnoise)

Language Preparation

A Language directory is created with the below files :

- ***L.fst***, FST form of lexicon.
- ***L_disambig.fst***, L.fst but including the disambiguation symbols.
- ***oov.int***, mapped integer of out-of-vocabulary words.
- ***oov.txt***, out-of-vocabulary words.
- ***phones.txt***, maps phones with integers.
- ***topo***, the topology of the HMMs we use.
- ***words.txt***, maps words with integers.
- ***phones/***, specifies various things about the phone set.

Feature Extraction

- Raw MFCC features are extracted from the audio files. This generates feats.scp file in data folder.

Cepstral Mean and Variance Normalization (CMVN)

- Feature normalization is performed after feature extraction to normalise the background noise.
- This generates cmvn.scp file in the data folder.

Acoustic Model Training

Monophone Training

- A monophone model is an acoustic model that does not include any contextual information about the preceding or following phone.
- Used as a building block for the triphone models, which do make use of contextual information.

Triphone Training

As phonemes vary depending on their particular context. The triphone models represent a phoneme variant in the context of two other (left and right) phonemes.

As not all triphone units are present (or will ever be present) in the dataset.

A phonetic decision tree groups these triphones into a smaller amount of acoustically distinct units, thereby reducing the number of parameters.

Different Training Algorithms

Delta+delta-delta training computes delta and double-delta features, or dynamic coefficients, to supplement the MFCC features

LDA-MLLT stands for Linear Discriminant Analysis – Maximum Likelihood Linear Transform.

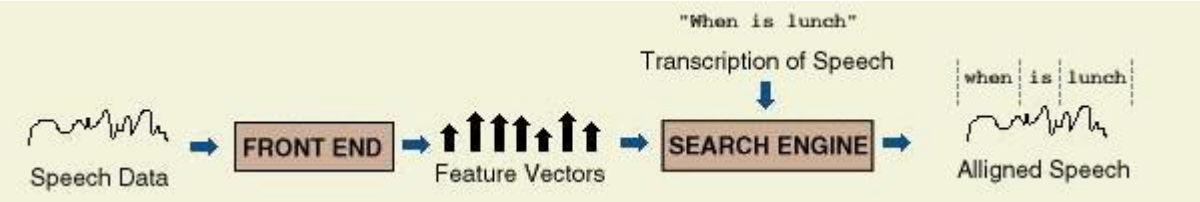
SAT stands for Speaker Adaptive Training. SAT also performs speaker and noise normalization by adapting to each specific speaker with a particular data transform

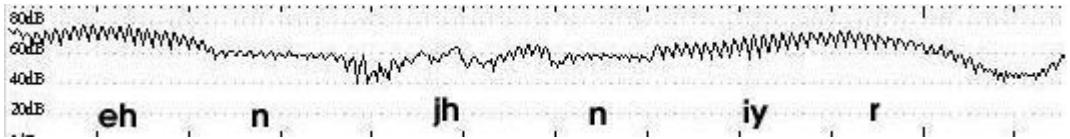
Force Alignment

Determining where in time particular words of the text transcription occurs in the speech segment.

The system then aligns the transcribed data with the speech data.

Aligns the phonemes of the transcription data to the speech data given, although with more explicitly defined boundaries on where each phoneme begins and ends.





Performance Evaluation

The Word Error Rate (WER) is a way to measure performance of an ASR.

WER=(S+D+I)/N

where

- S is the number of substitutions,
- D is the number of deletions,
- I is the number of insertions and
- N is the number of words in the reference

Accuracy is calculated as:

$$Recognition\ Accuracy = (1 - WER) \times 100.$$

Example:

REF: I **** am going to the college

HYP: I can of going to college

Eval I S D

WER = 100 (1+1+1)/6 = 50%

Accuracy= 50%