University of Pennsylvania Department of Linguistics

Penn Phonetics Lab Forced Aligner Toolkit (P2FA)

http://www.ling.upenn.edu/phonetics/p2fa/

==

A. Introduction

The Penn Phonetics Lab Forced Aligner (P2FA) is **an automatic phonetic alignment toolkit based on HTK**. It contains the acoustic models of American English, a Python script that can be used to do forced alignment, as well as this readme file and some examples.

There is also an online processing system on the P2FA website with which you can submit a WAV file and transcript and get back a Praat TextGrid file by email.

The acoustic models included in the toolkit are **GMM-based monophone HMMs**.

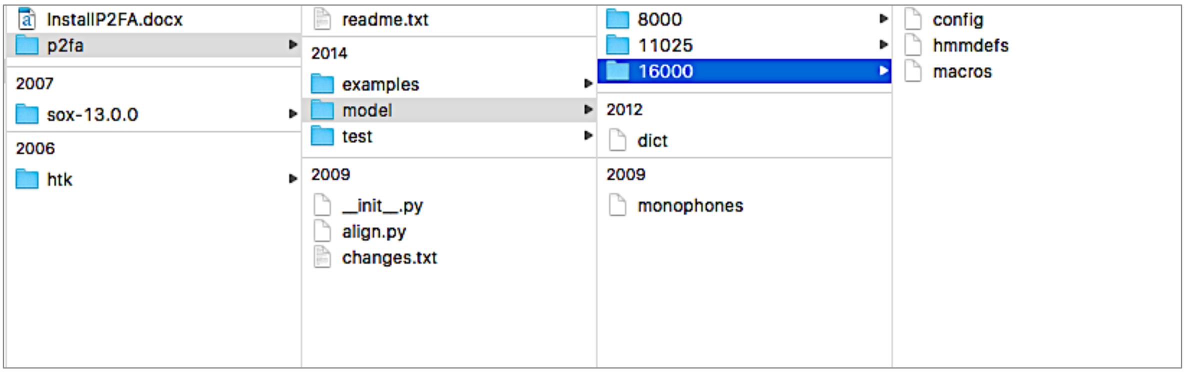
- **Each HMM state** has **32 Gaussian mixture components** on **39 PLP coefficients**.
- Separate models were created for speech sampled at 8 KHz, 11,025 Hz, and 16 KHz.
- The acoustic model includes a robust short-pause ("**sp**") HMM inserted optionally between words which greatly improves alignment accuracy.

< Citation >

P2FA can be cited as:

Jiahong Yuan and Mark Liberman. 2008. Speaker identification on the SCOTUS corpus. Proceedings of Acoustics '08.

B. Included in the Toolkit



./models/:

- the acoustic models by sampling frequency (‘hmmdefs’, ‘macros’)
- parameter files (‘config’)
- CMU pronunciation dictionary (‘dict’, ‘monophones’)

./align.py:

a python script that automates the procedure of doing forced alignment

- creates a **Praat TextGrid** file from a WAV file
- creates an **orthographic transcript**

./test/:

- files for testing the Toolkit

./examples/:

- the example files.

C. Prerequisites

You will need the **HTK toolkit version** 3.4 already installed to do forced alignment. HTK can be found at http://htk.eng.cam.ac.uk/. Only version 3.4 is supported by the align.py script. The newer version 3.4.1 has a problem aligning the 'sp' model.

You need to have Python 2.5/2.6 (earlier/later versions have not been tested) for the align.py script.

D. Using **align.py**

1

Run the followings in terminal:

```
python align.py wavfile trsfile output_file
```

(If you are not running align.py from the toolkit directory, you will need to specify its path.)

< Description >

wavfile

- the **path** to a WAV file containing the audio to be aligned
- If it was not sampled at one of the three sampling rates used for the acoustic models, it will be automatically **resampled to 11,025 Hz**. This sampling rate is recommended.

trsfile

- a **text file** containing the **transcript**
- Spaces or newlines should separate words.
- If a word is not found in the CMU pronouncing dictionary, an error will occur, but you can edit the file model or dict and add new pronunciations as needed.
- You may include the following labels in the transcript:
  - '{SL}' for silence, '{LG}' for laughter, '{NS}' for noise, '{CG}' for cough, '{BR}' for breath, and '{LS}' for lipsmack.

output_file

- a ***Praat TextGrid*** file containing the rest of the forced alignment.

- TESTING -

The toolkit contains example files for testing. You can align the test files with:

```
python align.py ./test/BREY00538.wav ./test/BREY00538.txt \
    ./test/BREY00538.TextGrid
```

(cf. the slash denotes that the single line is supposed to continue unbroken)

The output file may contain 'sp' intervals between words where there was a pause.

2

If you have **more than one file to align**, you can **write a shell script** and call align.py in a loop. You can also follow the instructions below.

3

Several command-line **options** can also be included.
They must precede the specification of the WAV file. They are:

```
-r sampling_rate   override which sample rate model to use, one of 8000, 11025, and 16000.
                   The default is the sampling rate of the WAV file if it is one of
                   the three, otherwise 11025.

-s start_time      start of portion of wavfile to align (in seconds, default 0)
-e end_time        end of portion of wavfile to align (in seconds, default to end)
```

E. Doing **Forced Alignment** The Hard Way

1	Please refer to the files under **./examples/** for the right formats for the files described in the next steps.
2	Prepare **speech files** and their **word transcription** as described below. You can do forced alignment for very long speech files (e.g., one hour), or you can also align many files in one step.
3	Create the reference transcription file, **transcript.mlf**. ‘transcript.mlf’ is a **HTK "master label file"** containing the transcripts of all the files to be force-aligned. Below are the steps to generate the file: I. **Capitalize** all the letters of the words. If a word doesn’t appear in the CMU pronunciation dictionary, you can either manually add it to the dictionary or exclude it from forced alignment, depending on your goal. The alignment will fail if there is an unknown word in transcript.mlf. II. You may include the following labels in transcript.mlf: '{SL}' for silence, '{LG}' for laughter, '{NS}' for noise, '{CG}' for cough, '{BR}' for breath, and '{LS}' for lipsmack. III. You may want to insert an '**sp**' between every two words. 'sp' stands for small pause. It can have zero length (no pause) from forced alignment.
4	Create **code.scp** and **test.scp**. They contain the names of the files to be coded and aligned respectively.
5	**Extract acoustic features**: `HCopy -T 1 -C ./models/your-sampling-rate/config -S code.scp`
	< Description > The file **config** contains **parameter settings** for the speech files (.wav, .raw, sampling rate, etc.) and for the acoustic features (mfcc, plp, etc.). If your speech files have a different sampling rate other than 8,000Hz, 11,025Hz, or 16,000Hz, you can downsample or upsample to 11,025 Hz (which by far has the best performance). In our training procedure, we downsampled from 44,100 Hz to the target sampling rate using 'sox -polyphase'.

6	**Forced alignment**:
	
	`HVite -T 1 -a -m -I transcript.mlf -H ./model/11025/macros -H ./model/11025/hmmdefs -S test.scp -i ./align.mlf -p 0.0 -s 5.0 ./model/dict ./model/monophones`
	`(cf. Replace ‘11025’ with your own sampling rate)`

	< Description > **align.mlf** is the forced alignment results. You can convert it to **label files**, e.g., Praat TextGrids. Please note that: a) the time unit in align.mlf is **100 ns** (0.00000001 second) b) there is a rounding issue when the sampling rate is 11,025 Hz and the time step is 10 milliseconds, so you need to correct the time stamps in alilgn.mlf. You can use the following formula to convert the time stamp x into seconds: y = (x/10000000 + 0.0125)*(11000/11025)