Lecture 12:
Speech and Multimodal Data Science

LING 1340/2340: Data Science for Linguists

Jevon Heath

Objectives

- Assignment review
  - HW4!?
  - Project progress
- Data science with non-text language data
  - Writing vs. speech
  - Processing speech
    - Speech recognition
    - Some tools
    - LVCSR
  - Processing multimodal data
    - ELAN

Recap: "Data Science"

- "Bringing structure to large quantities of formless data" (Davenport & Patil 2012)

- Sourcing/sifting/cleaning/organizing data in the wild

Speech vs. writing

- Speech is ubiquitous to human communities
- Writing was invented
- Speech is spontaneous
- Writing is deliberate
- Humans acquire speech without instruction
- Writing requires instruction to learn

Speech corpora

- Ubiquitous:
  - All communities, all languages
- Not deliberate:
  - Different audience design considerations (Bell 1984)
  - More plentiful; more contexts
- No instruction needed:
  - Less formal* constraints

What to do with speech data?

- Analyze it directly.
  - Language identification
  - Phonetic research
  - Informing models (such as the following)
- Convert it to text, then do other things with it…
  - Forced alignment
  - ASR (Automatic Speech Recognition) and ASU (Understanding)
  - Automatic closed-captioning
- Make it!
  - Speech synthesis / Text-to-speech (TTS)
  - Conversational agents

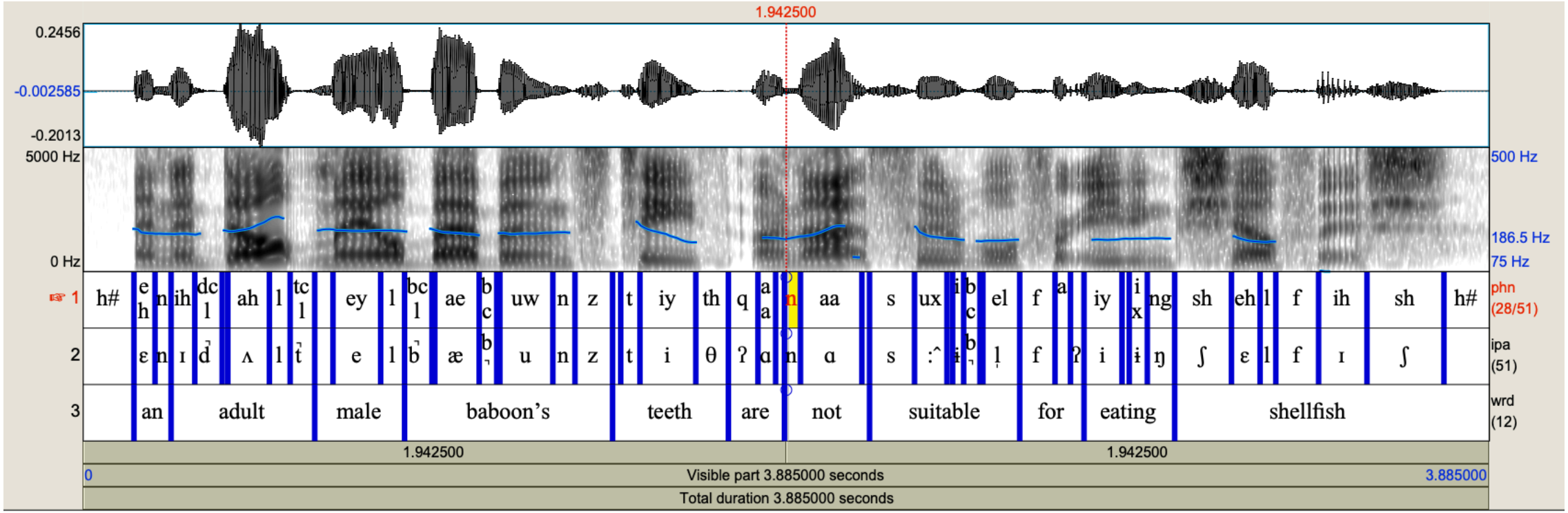
Popular speech corpora

- Buckeye Corpus (Pitt et al. 2005)
- TIMIT (Garofalo et al. 1993) <– in Licensed-Data-Sets repo
- TalkBank links (https://talkbank.org)

Popular speech data analysis tools for linguists

- Praat (Boersma & Weenink 2020)
  - Praat script repositories [1] [2] [3] [4] – Which version of Praat though?
  - Parselmouth: Access Praat code through Python – also not very well documented
- Klatt formant synthesizer (Klatt 1975, 1984) (online demo)
- SoX audio editing software (Bagwell 1991–2015)

Praat


1.942500
0.2456
-0.002585
-0.2013
5000 Hz
0 Hz
500 Hz
186.5 Hz
75 Hz
phn (28/51)
ipa (51)
wrd (12)
h# eh n ih dcl ah l tcl ey l bcl ae bc uw n z t iy th q aa n aa s ux ic el f a iy ix ng sh eh l f ih sh h#
ɛ n ɪ d̚ ʌ l t̚ e l b̚ æ ɓ u n z t i θ ʔ ɑ n ɑ s ːˆ ɨ ɓ l̩ f ʔ i ɨ ŋ ʃ ɛ l f ɪ ʃ
an adult male baboon's teeth are not suitable for eating shellfish
1.942500
1.942500
0
Visible part 3.885000 seconds
3.885000
Total duration 3.885000 seconds

An example Praat textgrid

```
File type = "ooTextFile"
Object class = "TextGrid"

xmin = 0
xmax = 3.8850000000000002
tiers? <exists>
size = 3
item []:
    item [1]:
        class = "IntervalTier"
        name = "phn"
        xmin = 0
        xmax = 3.8850000000000002
        intervals: size = 51
        intervals [1]:
            xmin = 0
            xmax = 0.14
            text = "h#"
        intervals [2]:
            xmin = 0.14
            xmax = 0.198375
            text = "eh"
        intervals [3]:
            xmin = 0.198375
            xmax = 0.24281250000000001
            text = "n"
        intervals [4]:
            xmin = 0.24281250000000001
            xmax = 0.3080625
            text = "ih"
        intervals [5]:
            xmin = 0.3080625
            xmax = 0.38375000000000004
            text = "dcl"
        intervals [6]:
            xmin = 0.38375000000000004
            xmax = 0.3994375
            text = "d"
        intervals [7]:
```

An example Praat script

StandardizeLengths.praat — Open with Praat

```
# This script enables the user to make all sound files in a folder the same length. Shorter sound files will have
silence appended, and longer files will be truncated.

# Jevon Heath 2015-04-08


# Create a list of all sound files in the directory.

form Label all .wav files in this directory
        comment Name of directory:
        text directory /Users/cadavaca/Documents/Projects/AXB/Stimuli/Andrew/read3
        comment Desired file length (in seconds):
        text newlength 3.0
endform

Create Strings as file list... fileList 'directory$'/*.wav
n = Get number of strings


# Standardize the file lengths.

for i to n
        select Strings fileList
        soundfile$ = Get string... i
        Read from file... 'directory$'/'soundfile$'
        soundname$ = selected$ ("Sound")
        newsound = Extract part: 0, newlength, "rectangular", 1, "no"
        select newsound
        Save as WAV file... 'directory$'/'soundname$'.wav
        plus Sound 'soundname$'
        Remove
endfor

select Strings fileList
Remove
```

ASR: Automatic Speech Recognition

- Assume that all speech data is noisy (*"noisy-channel" model*)
- Compare every possible sentence to the target waveform, and select the best match (*decoding/search/inference*)
  - *What is the "best match"?* Bayesian inference.
  - *Every possible sentence?!* Hidden Markov Models.

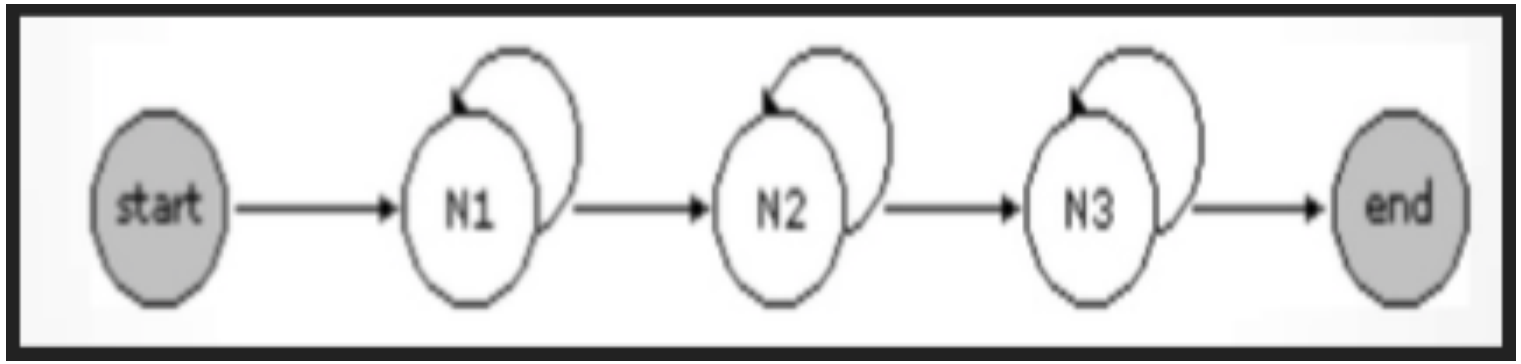
- SpeechRecognition package: Use ASR APIs through Python

The Hidden Markov Model

- A Markov model: future states depend on the current state
  - *not* on anything prior to the current state
- *Hidden*: we can't directly access the nature of the dependencies between states

The Hidden Markov Model and speech -- assumptions

- The speech stream is a sequence of steady states
- Transitions between states are not arbitrary
  - Simple assumption: any state (phone) transitions only to itself or to a specific following state
  - Phonemes are encoded as a series of states (*Why?*)
- Each word is a different HMM composed of phone HMMs

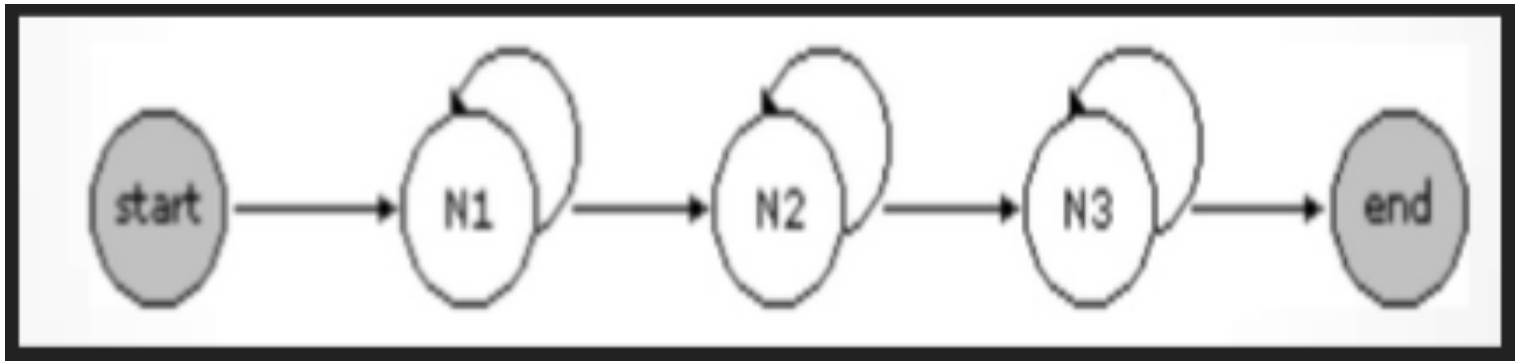


ASR: Issues

- Speaker variation
- Genre variation
- Noise/environmental variation
- Disfluencies
- [Predictive text issues]
- Decoding

Forced alignment

- Task is to determine when N1, N2, N3 begin



- Is there still inference?

Components of forced alignment

- A pronouncing dictionary
  - Example: cmudict.txt

```
JAVANESE  JH AA2 V AH0 N IY1 Z
JAVASCRIPT  JH AA1 V AH0 S K R IH2 P T
JAVELIN  JH AE1 V AH0 L AH0 N
JAVELIN(1)  JH AE1 V AH0 L IH0 N
JAVELIN(2)  JH AE1 V L AH0 N
JAVELIN(3)  JH AE1 V L IH0 N
JAVETT  JH AE1 V AH0 T
JAVIER  HH AA2 V IY0 EH1 R
JAVITS  JH AE1 V IH0 T S
JAVORSKY  Y AH0 V AO1 R S K IY0
JAW  JH AO1
```

- An acoustic model

- A transcript

Concerns about forced alignment

- It'll make mistakes
- It's a black box
- Automation removes the researcher from the data

Forced alignment tools

- Penn forced aligner (Yuan & Liberman 2009)
  - FAVE-align (Rosenfelder et al. 2011)
  - Montreal Forced Aligner (McAuliffe et al. 2017)
  - EasyAlign (Goldman 2011 – Windows only)
- See this collection of links for forced alignment tools
  - aeneas: Forced alignment through Python, without ASR? (MFCC and DTW)

Levels of complexity to ASR

- Forced alignment – no word-level inference
- Task-specific data – few reasonable competitors
- Large Vocabulary Continuous Speech Recognition (LVCSR)
  - *a.k.a.* speech analytics

Approaches to LVCSR

- Topic analysis
- Speaker-dependent training
- *n*-gram modeling (for phones and words)
- Deep learning (Deep/Recurrent Neural Networks)
- Adaptive training

Deep neural networks

- Successive layers of a neural network; multiple levels of representation (e.g. of linguistics structure)
- Recurrent neural networks include temporal states
- Both require a LOT of training data

Issue: How much data?

- In principle: *enough to be able to distinguish the signal from the noise*
- Enough to inform feature layers
- Pre-training can compensate for low training resources (Thomas et al. 2013; Vu et al. 2011)

ELAN: Annotation for video & audio

- (link)
- Projects using ELAN: https://tla.mpi.nl/past-projects/
- Example: BU ASL corpus (through Rutgers)

Another licensed data set

- TIMIT Acoustic-Phonetic Continuous Speech Corpus
  - https://catalog.ldc.upenn.edu/ldc93s1
  - In "Licensed-Data-Sets" repo
  - Is this a "corpus"...?