Lecture 21: Forced Aligners, ASR

LING 1340/2340: Data Science for Linguists

Na-Rae Han

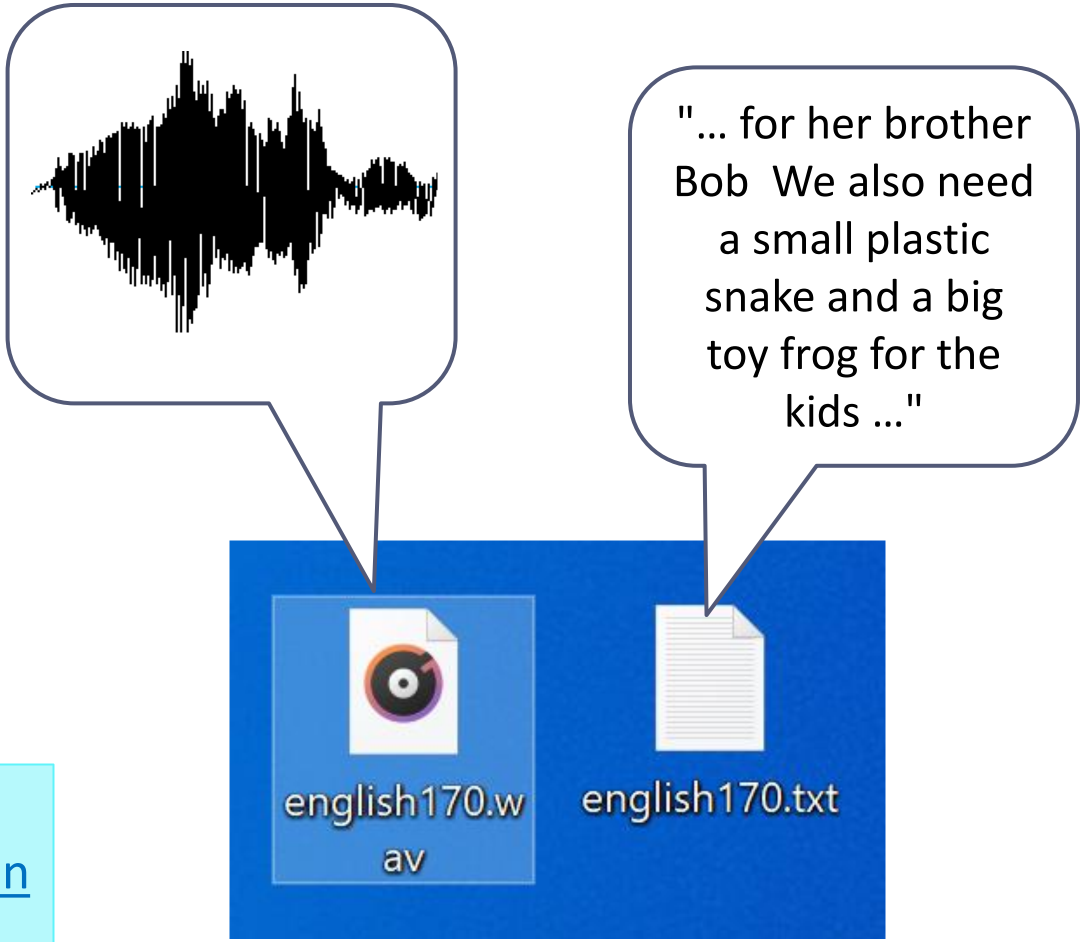
Objectives

- Forced alignment
  - Montreal Forced Aligner demo

- ASR!
  - ASR demo
  - ASR theory

Forced alignment

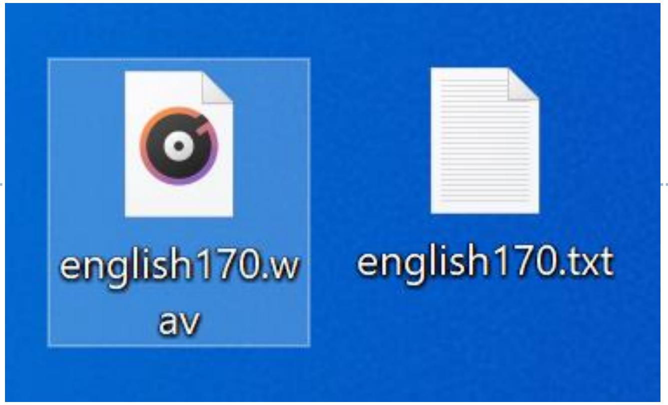
- "**Forced alignment**": automatic synchronization of a sequence of phones with an audio file.
- Purpose: speed up manual segmentation and annotation
  - Rather than doing everything manually from scratch, correct output from forced aligner
  - Makes life easier for linguists doing speech-focused research!



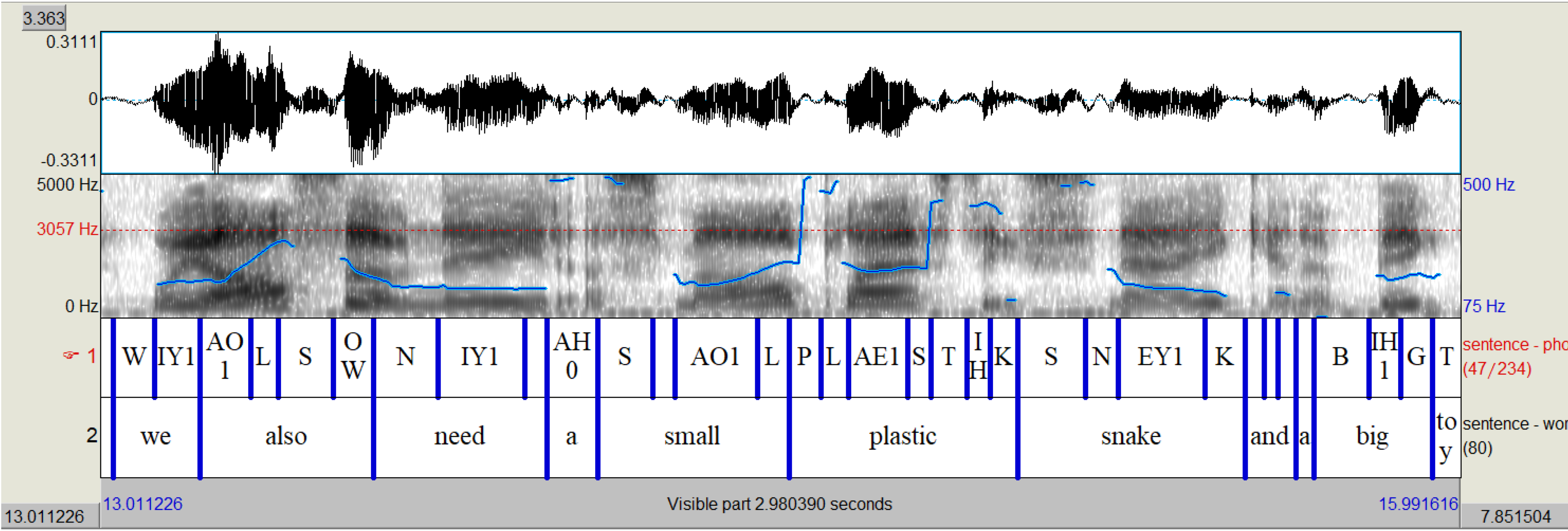
Example speech from the Speech Accent Archive: https://accent.gmu.edu/browse_language.php?function=detail&speakerid=556

Forced alignment

- You have: a speech file (.wav), a transcript file (.txt) →

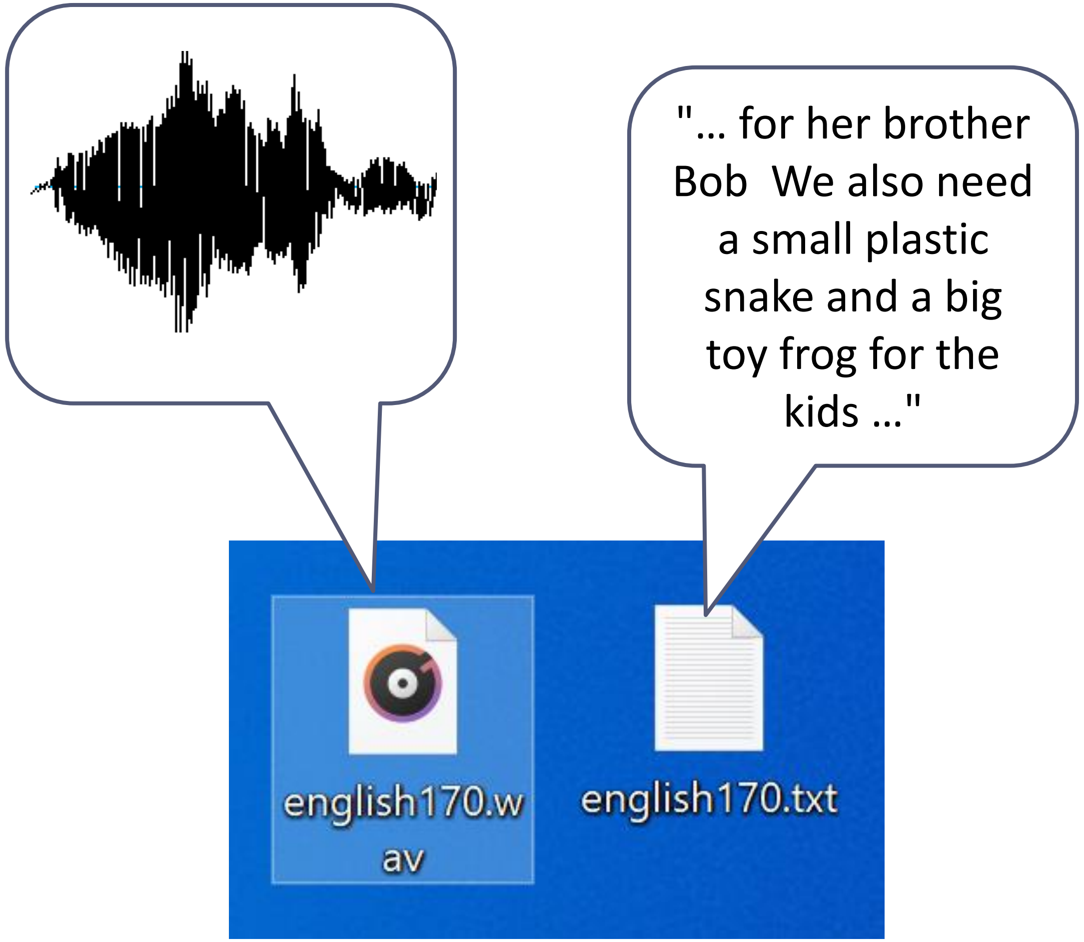


- You want:



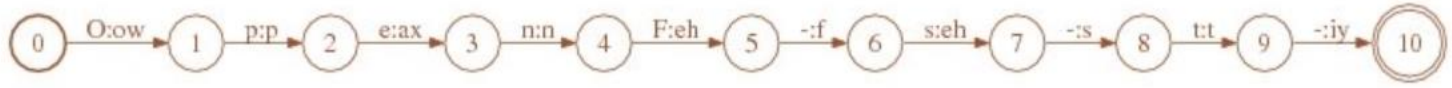
Sound wave, words, phones

- What additional linguistic information is needed?
  - Pronunciation dictionary
    - Phonemic representations for "brother", "we", "also"…
    - More broadly: orthography → phone (**G2P, "grapheme-to-phoneme"**)
    - David Mortensen's G2P library "Epitran" https://github.com/dmort27/epitran
  - Acoustic model
    - How phonemic representation relates to sound wave



Demo: Montreal Forced Aligner

- Home page:
  - https://montreal-forced-aligner.readthedocs.io/en/latest/user_guide/index.html#user-guide
- GitHub project page:
  - https://github.com/MontrealCorpusTools/Montreal-Forced-Aligner
- Builds on popular/standard libraries:
  - **Kaldi** ASR toolkit
    - [home] [GitHub repo]
  - which builds on **OpenFST**
    - [home]



Steps (latest MFA version 2.0)

- Install Kaldi, MFA
  - Windows users: For ver 2.0, you need WSL (**W**indows **S**ubsystem for **L**inux, essentially Linux on Windows!) to use full G2P functionality. Alternatively: install older ver 1.0.1 available here, which is Windows-native.
- Prepare data to align
  - Speech files (WAV format, single-channel)
  - Transcript files (.lab or .txt format; no punctuation)

We'll use TIMIT data for demo (pretend it came with audio files and .TXT transcripts only)

- Download language models (pre-trained, MFA offers many)
  - A pronunciation dictionary for the language
    - If not available: produce one by running language-specific G2P (grapheme-to-phoneme) on your transcript files
  - An acoustic model for the language
- Run:
  - `mfa align <input-dir> <pron-dict> <acoustic-model> <output-dir>`
- New TextGrid files in the output dir! Examine.

Cleaning transcript files

MINGW64:/c/Users/narae/Desktop/true_wav

```
narae@T480s MINGW64 ~/Desktop/FCJF0
$ cat *TXT
0 46797 She had your dark suit in greasy wash water all year.
0 34509 Don't ask me to carry an oily rag like that.
0 49460 Even then, if she took one step forward he could catch her.
0 45466 Or borrow some money from someone and go home by bus?
0 57856 A sailboat may have a bone in her teeth one minute and lie becalmed the next.
0 24679 The emperor had a mean temper.
0 27751 How permanent are their records?
0 23143 The meeting is now adjourned.
0 36250 Critical equipment needs proper maintenance.
0 39220 Tim takes Sheila to see movies twice a week.
```

Initial digits and punctuation need to go

```
narae@T480s MINGW64 ~/Desktop/FCJF0
$ perl -npe 's/^\d \d+ //' SA1.TXT
She had your dark suit in greasy wash water all year.

narae@T480s MINGW64 ~/Desktop/FCJF0
$ perl -npe 's/^\d \d+ //; s/\.//g;' SA1.TXT
She had your dark suit in greasy wash water all year
```

Perl + regular expressions to clean up

MINGW64:/c/Users/narae/Desktop/true_wav

```
narae@T480s MINGW64 ~/Desktop/FCJF0
$ perl -npe 's/^\d \d+ //; s/[\.,\?]//g;' *.TXT
She had your dark suit in greasy wash water all year
Don't ask me to carry an oily rag like that
Even then if she took one step forward he could catch her
Or borrow some money from someone and go home by bus
A sailboat may have a bone in her teeth one minute and lie becalmed the next
The emperor had a mean temper
How permanent are their records
The meeting is now adjourned
Critical equipment needs proper maintenance
Tim takes Sheila to see movies twice a week

narae@T480s MINGW64 ~/Desktop/FCJF0
$ for x in *TXT
> do
> perl -npe 's/^\d \d+ //; s/[\.,\?]//g;' $x > ../true_wav/$x
> echo $x completed
> done
SA1.TXT completed
SA2.TXT completed
SI1027.TXT completed
SI1657.TXT completed
SI648.TXT completed
SX127.TXT completed
SX217.TXT completed
SX307.TXT completed
SX37.TXT completed
SX397.TXT completed

narae@T480s MINGW64 ~/Desktop/FCJF0
$ cd ../true_wav/

narae@T480s MINGW64 ~/Desktop/true_wav
$ ls
SA1.TXT  SA2.TXT  SI1027.TXT  SI1657.TXT  SI648.TXT  SX127.TXT  SX217.TXT  SX307.TXT  SX37.TXT  SX397.TXT
SA1.WAV  SA2.WAV  SI1027.WAV  SI1657.WAV  SI648.WAV  SX127.WAV  SX217.WAV  SX307.WAV  SX37.WAV  SX397.WAV
```

Use bash for-loop to create cleaned-up version of all .TXT files

.WAV and .TXT files are now ready...

Download language models

- MFA's pre-trained models:
  - https://mfa-models.readthedocs.io/en/latest/

Pretrained acoustic models

As part of using the Montreal Forced Aligner in our own research, we have trained acoustic models for a number of languages. If you would like to use them, please download them below. Please note the dictionary that they were trained with to see more information about the phone set. When using these with a pronunciation dictionary, the phone sets must be compatible. If the orthography of the language is transparent, it is likely that we have a G2P model that can be used to generate the necessary pronunciation dictionary.

Any of the following acoustic models can be downloaded with the command `mfa download acoustic <language_id>`. You can get a full list of the currently available acoustic models via `mfa download acoustic`. New models contributed by users will be periodically added. If you would like to contribute your trained models, please contact Michael McAuliffe at michael.e.mcauliffe@gmail.com.

Language	Link	Corpus	Number of speakers	Audio (hours)	Phone set
Arabic	Arabic acoustic model	GlobalPhone	80	19.0	GlobalPhone
Bulgarian	Bulgarian acoustic model	GlobalPhone	79	21.4	GlobalPhone
Croatian	Croatian acoustic model	GlobalPhone	94	15.9	GlobalPhone
Czech	Czech acoustic model	GlobalPhone	102	31.7	GlobalPhone
English	English acoustic model	LibriSpeech	2484	982.3	Arpabet (stressed)
French (FR)	French (FR) acoustic model	GlobalPhone	100	26.9	GlobalPhone

Available pronunciation dictionaries

Any of the following pronunciation dictionaries can be downloaded with the command `mfa download dictionary <language_id>`. You can get a full list of the currently available dictionaries via `mfa download dictionary`. New dictionaries contributed by users will be periodically added. If you would like to contribute your dictionaries, please contact Michael McAuliffe at michael.e.mcauliffe@gmail.com.

Language	Link	Orthography system	Phone set
English	English pronunciation dictionary	Latin	Arpabet (stressed)
French	French Prosodylab dictionary	Latin	Prosodylab French
German	German Prosodylab dictionary	Latin	Prosodylab German

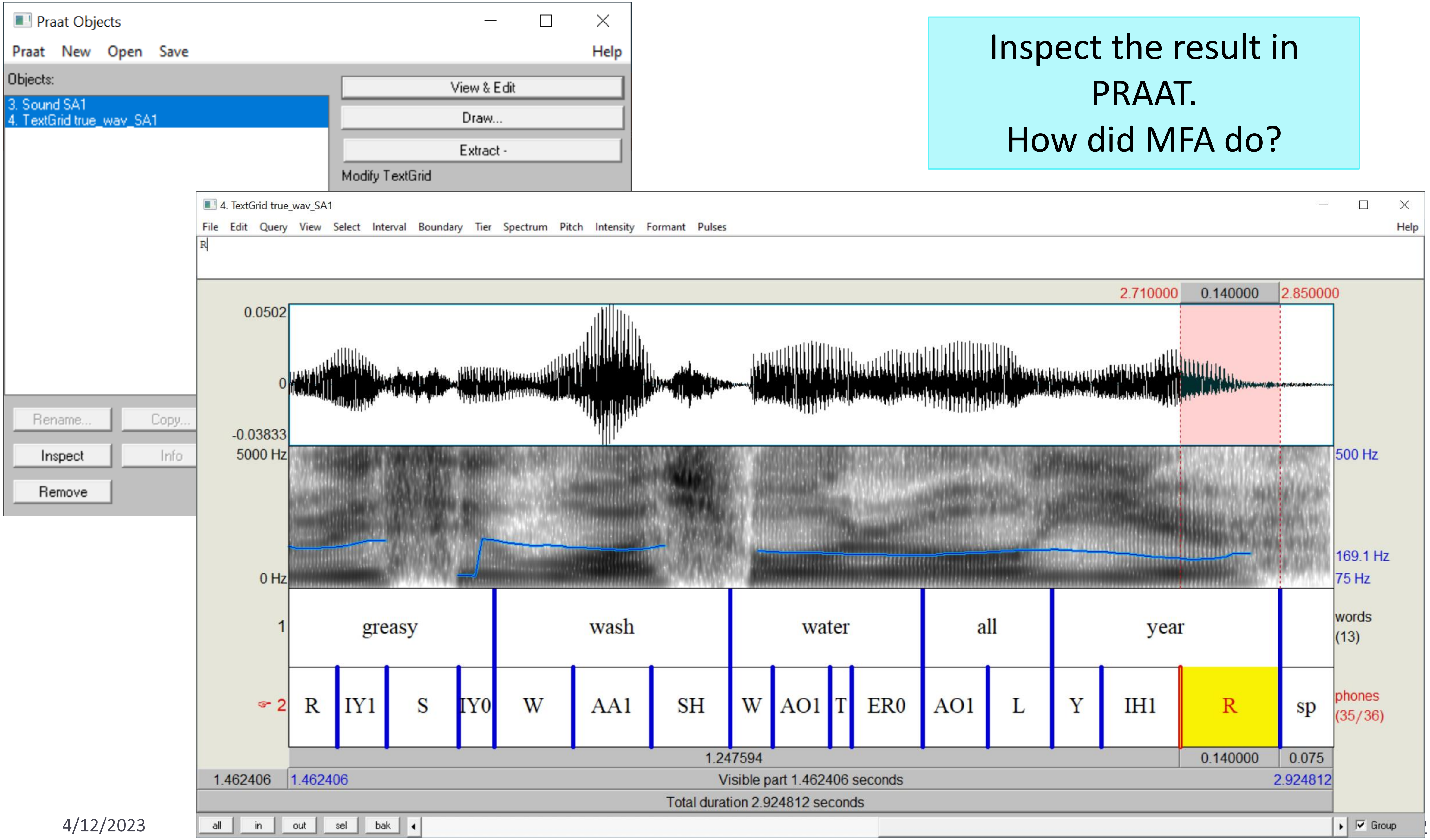
CMU pronouncing dictionary

narae@T480s: /mnt/c/Users/narae/Desktop

```
(base) narae@T480s:~$ conda activate aligner
(aligner) narae@T480s:~$ cd /mnt/c/Users/narae/Desktop/
(aligner) narae@T480s:/mnt/c/Users/narae/Desktop$ ls
FCJF0  desktop.ini  english.dict  english.zip  tidy  true_wav
(aligner) narae@T480s:/mnt/c/Users/narae/Desktop$ mkdir mfa_out
(aligner) narae@T480s:/mnt/c/Users/narae/Desktop$ mfa align true_wav/ english.dict english.zip mfa_out/
All required kaldi binaries were found!
/home/narae/Documents/MFA/true_wav/align.log
INFO - Setting up corpus information...
loading from source
INFO - Number of speakers in corpus: 1, average number of utterances per speaker: 10.0
INFO - Number of speakers in corpus: 1, average number of utterances per speaker: 10.0
INFO - Parsing dictionary without pronunciation probabilties without silence probabilties
INFO - Creating dictionary information...
INFO - Setting up training data...
INFO - Generating base features (mfcc)...
INFO - Calculating CMVN...
INFO - Done with setup!
INFO - Performing first-pass alignment...
INFO - Calculating fMLLR for speaker adaptation...
INFO - Performing second-pass alignment...
INFO - All done!
(aligner) narae@T480s:/mnt/c/Users/narae/Desktop$ ls mfa_out/
true_wav_SA1.TextGrid     true_wav_SI1657.TextGrid  true_wav_SX217.TextGrid  true_wav_SX397.TextGrid
true_wav_SA2.TextGrid     true_wav_SI648.TextGrid   true_wav_SX307.TextGrid
true_wav_SI1027.TextGrid  true_wav_SX127.TextGrid   true_wav_SX37.TextGrid
(aligner) narae@T480s:/mnt/c/Users/narae/Desktop$
```

MFA is installed on **WSL**, need to bring out **Ubuntu console**

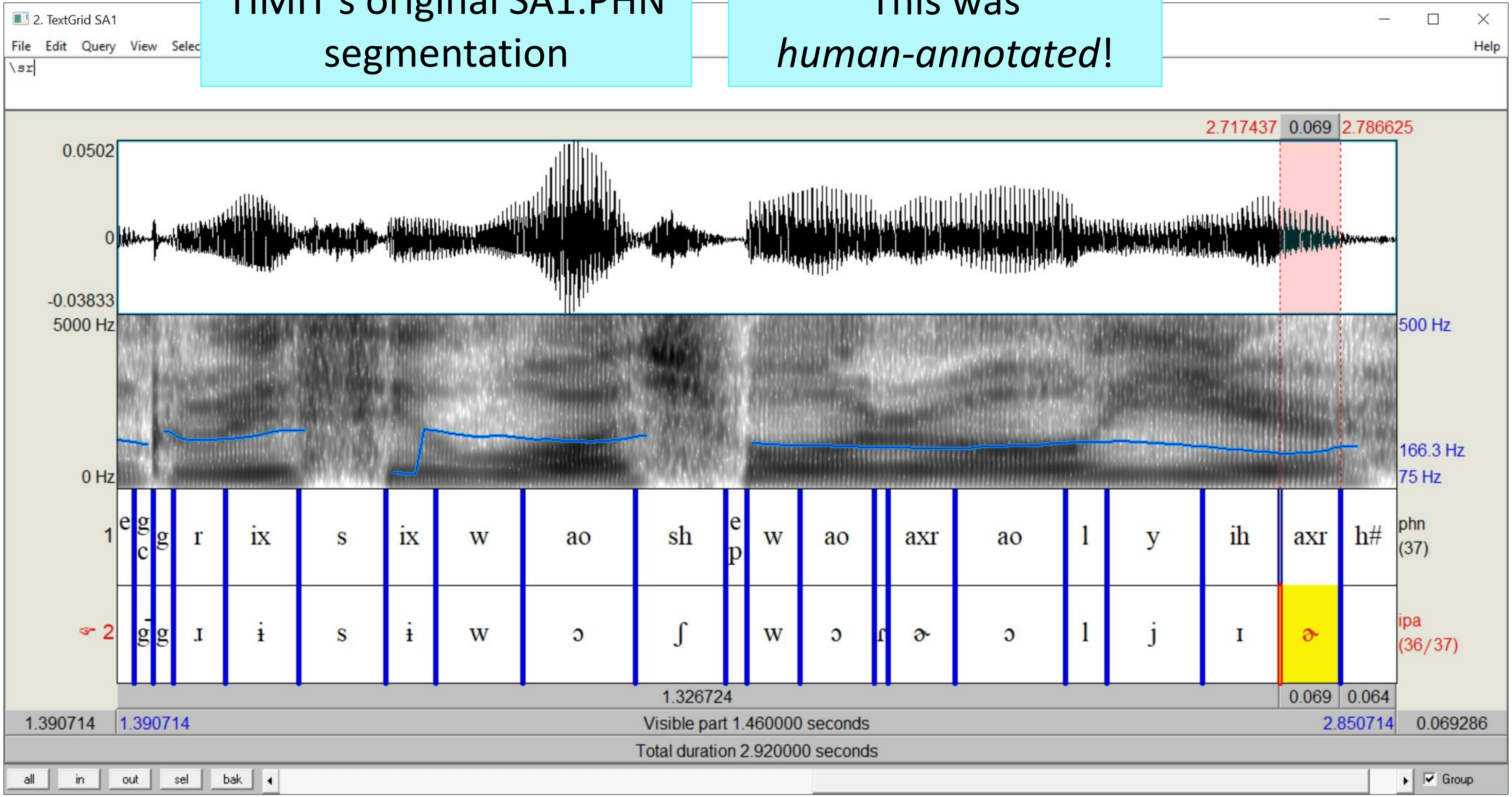
SUCCESS!
New crop of TextGrid files



Inspect the result in PRAAT.
How did MFA do?

Compare with TIMIT's original SA1.PHN segmentation

This was *human-annotated*!



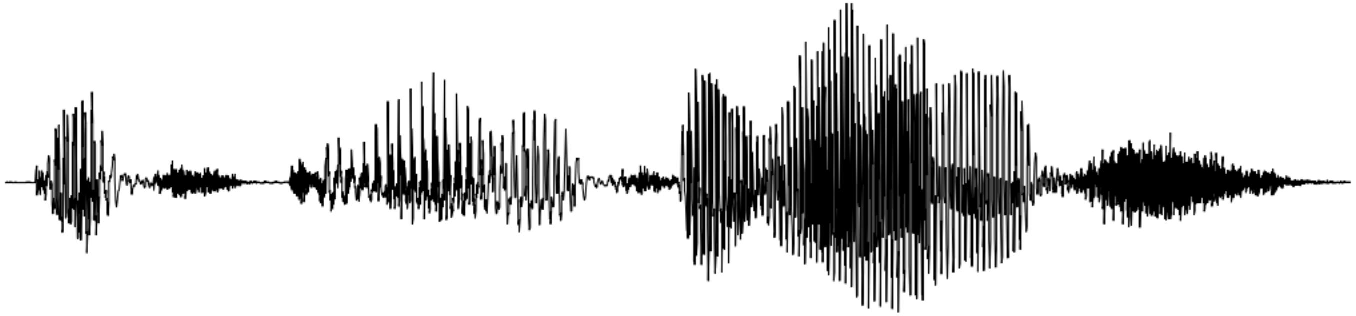
No transcripts?

- But wait! What if we don't even have a transcript file?
- We can auto-transcribe… ASR!
- ASR demo using SpeechRecognition Python library
  - In Jupyter Notebook

Backing up: ASR

SLP, Jurafsky & Martin

- Forced alignment is based on ASR technology.
- This is NOT an NLP class, but we should at least have some sense of how ASR works…



```
It's time for lunch
```

Is **processing speech** going to be entirely different from **text processing technologies**?

In Which We Skim Through Blog Articles (Again) In Lieu of Proper Academic Textbook

- Proper academic textbook chapter on ASR/TTS:
  - Jurafsky & Martin (2020) *Speech and Language Processing* Ch. 26 Automatic Speech Recognition and Text-to-Speech
- More accessible:
  - Speech Recognition – ASR Model Training (by Jonathan Hui)
  - Introduction to ASR (by Maël Fabien, with IPA!!)

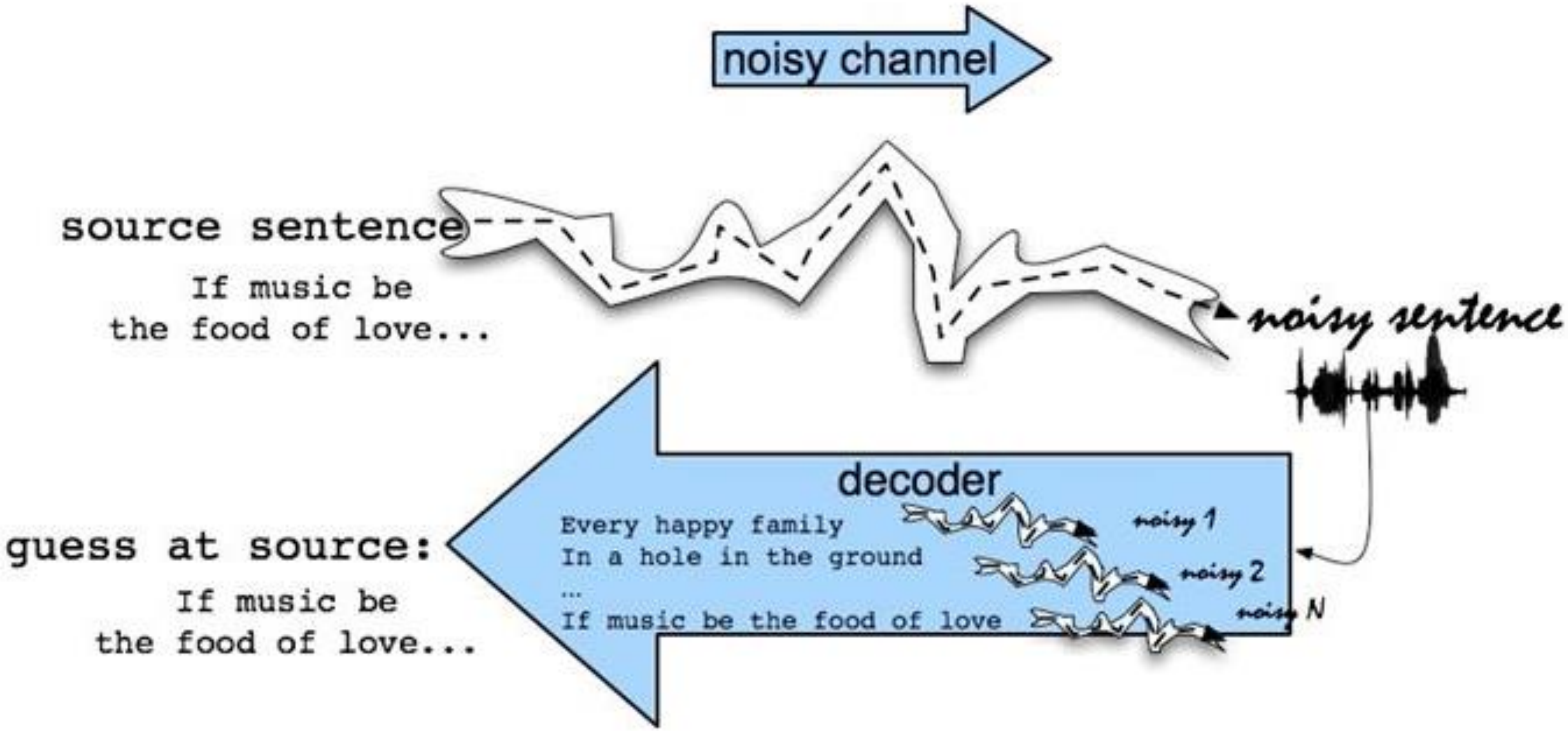
All the building blocks…

- English:
  - ARPAbet
  - CMU Pronouncing Dictionary
- World languages:
  - G2P (grapheme-to-phoneme)
- HMM (Hidden Markov Model), HTK (HMM ToolKit)
- Kaldi (ASR toolkit, built on HTK)
- Finite-State Transducer (OpenFST)
- N-gram language models

Many of them look familiar…
from LING 1330
Intro to CompLing!

The Noisy Channel Model

SLP, Jurafsky & Martin
https://web.stanford.edu/~jurafsky/slp3/B.pdf

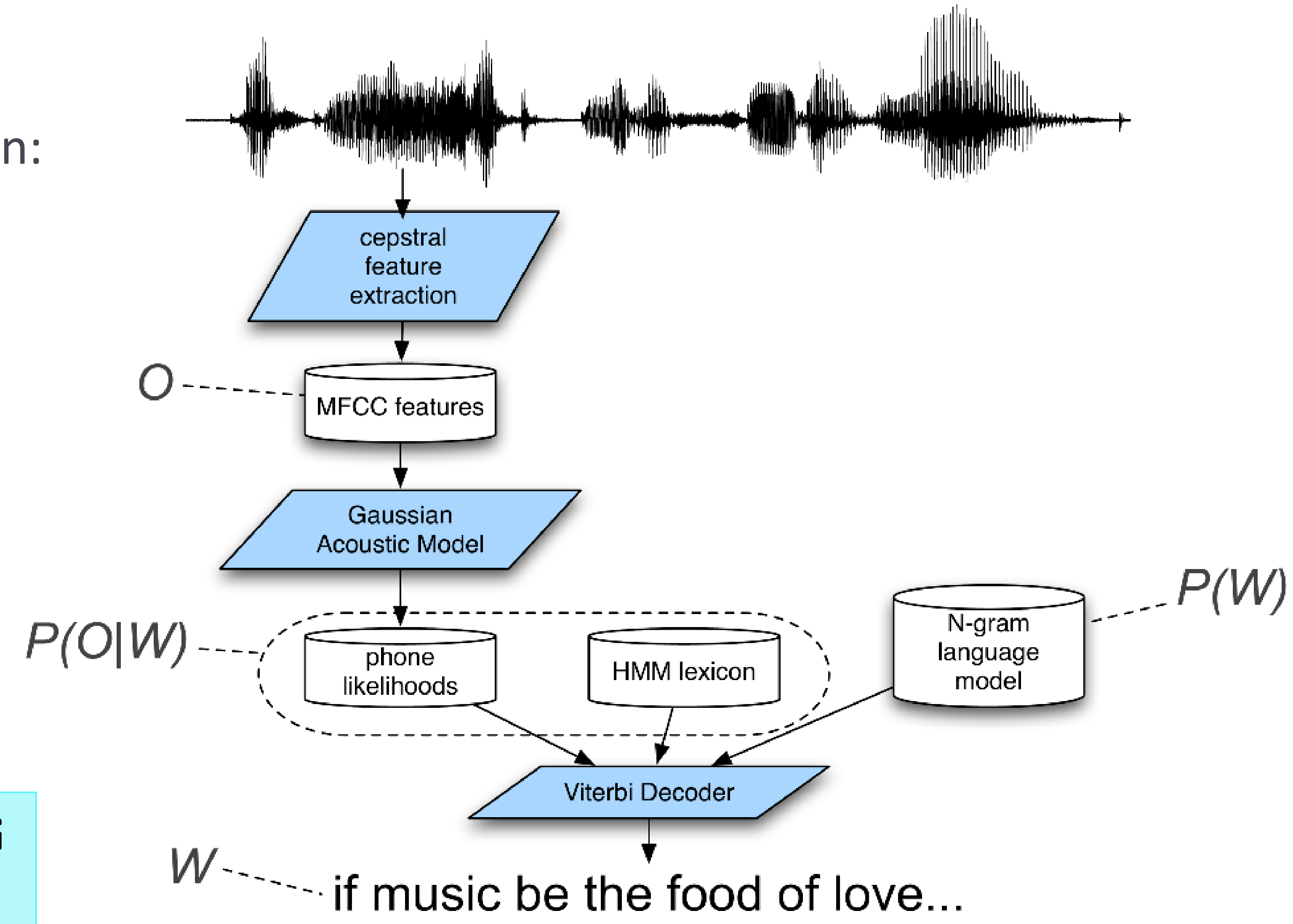


Speech recognition architecture (classic)

SLP, Jurafsky & Martin

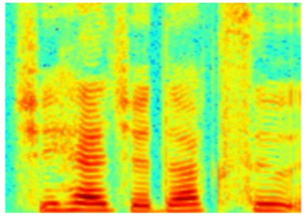
- ASR components
  - Lexicons and pronunciation:
    - Hidden Markov Models
  - Feature extraction
  - Acoustic modeling
  - Decoding
  - Language modeling:
    - N-gram models
- But: why "classic"?

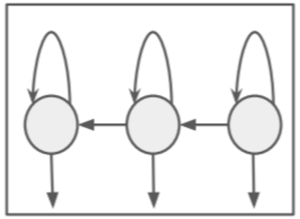
Because **DEEP LEARNING**
(what else?)



Speech recognition architecture (classic)

- Inference: Given audio features $\mathbf{X} = x_1 x_2 \ldots x_T$ infer most likely text sequence $\mathbf{Y}^* = y_1 y_2 \ldots y_L$ that caused the audio features





$\mathbf{X} = x_1\ x_2\ \ldots\ x_T$

Cat: /kæt/
Sits: /sɪts/
On: /ɔ:n
A: /ə/
Mat: /mæt/

Y = "*cat sits on a mat*"

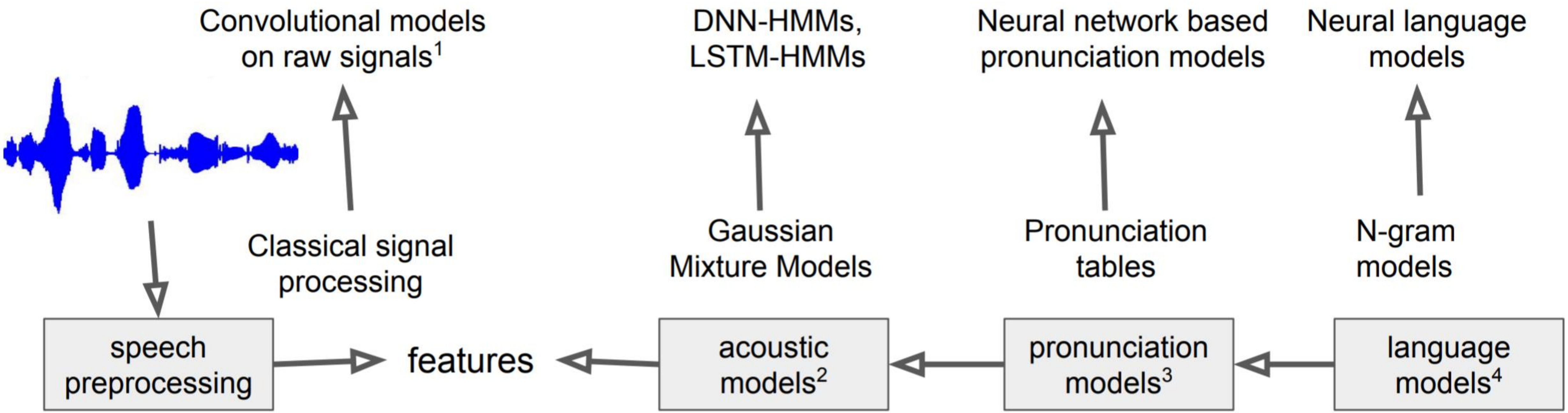
speech preprocessing → features ← acoustic models ← pronunciation models ← language models

$$\mathbf{Y}^* = \arg\max_{\mathbf{Y}} p(\mathbf{X}|\mathbf{Y})\, p(\mathbf{Y})$$

Speech recognition architecture (neural net)

SLP, Jurafsky & Martin

- Each of the components seems to be better off with a neural network



Wrapping up

- One last To-do!
  - Visit your classmates, final round
- Next class:
  - Emma presents ELAN demo
  - Project presentation: Sen