Lecture 17: Speech Data and Phonetic Representations

LING 1340/2340: Data Science for Linguists

Na-Rae Han

Objectives

- Speech data
  - Working with phonetic representations and IPAs

Speech vs. Writing

- Ubiquitous to human communities
- Spontaneous
- Humans acquire speech without instruction

- Invented, many communities without
- Deliberate
- Requires instruction to learn

en chart showing the location and usage of all the writing systems of the world

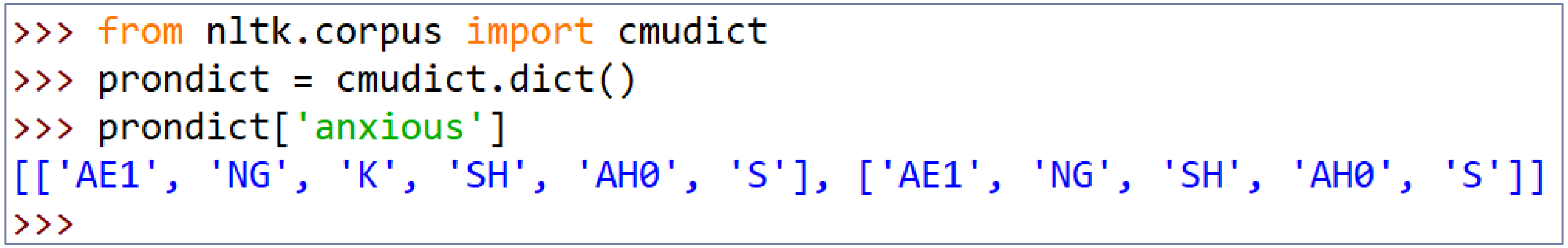
As digital data, fundamentally different representations!

What to do with speech data?

- Directly analyze acoustic signals
  - Language identification
  - Phonetics research
  - Informing models (example below)
- Convert audio to text, then text-process for downstream tasks
  - ASR (Automatic Speech Recognition) and ASU (… Understanding)
  - Automatic closed-captioning
- The other direction: text -> Sound
  - Speech Synthesis / Text-to-Speech (TTS), Conversational Agents
  - **Interim step**: text to phonetic *representation*

Speech sounds: how to encode/represent?

- **IPA**, ɒbvɪəsli…
  - IPA chars are Unicode characters, cumbersome to input directly (especially in the olden days!)
  - ASCII-based coding systems have been commonly used for English
- DISC phonemic alphabet, used by APLS:
  - https://djvill.github.io/APLS/doc/phonemic-transcription
- Do you remember the CMU Pronouncing Dictionary?



```
>>> from nltk.corpus import cmudict
>>> prondict = cmudict.dict()
>>> prondict['anxious']
[['AE1', 'NG', 'K', 'SH', 'AH0', 'S'], ['AE1', 'NG', 'SH', 'AH0', 'S']]
>>>
```

  - Uses **ARPABET**: https://en.wikipedia.org/wiki/ARPABET
  - CMU pronouncing dict is used in all sorts of English speech technologies…
  - Also: https://heardle.glitch.me/

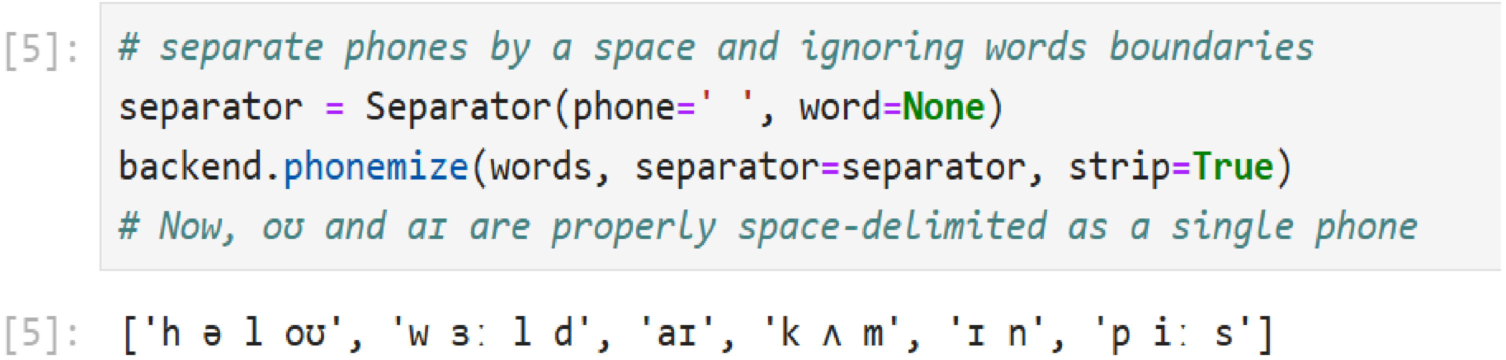
Working with phonetic representations

- **Grapheme-to-Phoneme** (= **G2P**)
  - Methods for transforming orthographic text into sound representation, often IPA
  - 'father' → 'fɑːðər', '学校' → 'ɕyɛɜ ɕjɑu5' , etc.
  - We'll look at the `phonemizer` Python library

- Breaking down IPA into **phonological features**
  - Decompose IPA into their articulatory features
    - /s/ → [-syl, -son, +cons, +cont, -delrel, -lat, -nas, 0strid, -voi, -sg, -cg, +ant, +cor, -distr, -lab, -hi, -lo, -back, -round, -velaric, 0tense, -long, 0hitone, 0hireg]
  - We'll look at the panphon Python library

→ Demo in Jupyter Notebook

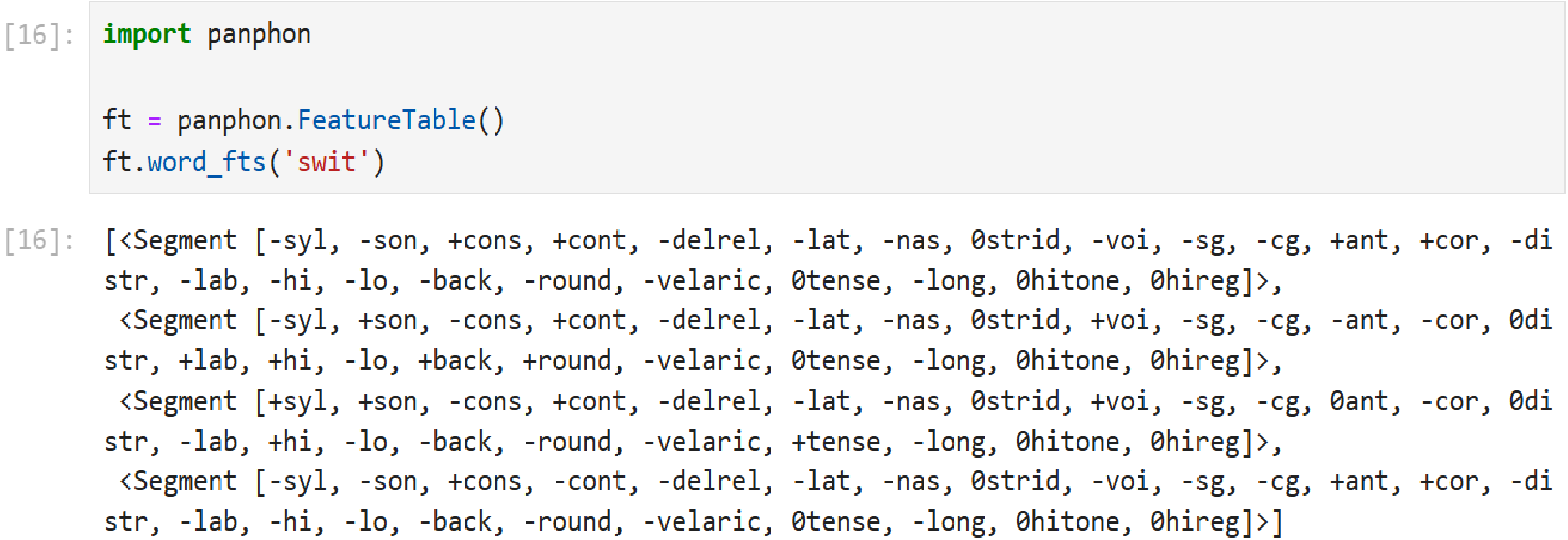
Phonemizer:



```
[5]: # separate phones by a space and ignoring words boundaries
     separator = Separator(phone=' ', word=None)
     backend.phonemize(words, separator=separator, strip=True)
     # Now, oʊ and aɪ are properly space-delimited as a single phone

[5]: ['h ə l oʊ', 'w ɜː l d', 'aɪ', 'k ʌ m', 'ɪ n', 'p iː s']
```

Panphon:



```
[16]: import panphon

      ft = panphon.FeatureTable()
      ft.word_fts('swit')

[16]: [<Segment [-syl, -son, +cons, +cont, -delrel, -lat, -nas, 0strid, -voi, -sg, -cg, +ant, +cor, -di
      str, -lab, -hi, -lo, -back, -round, -velaric, 0tense, -long, 0hitone, 0hireg]>,
       <Segment [-syl, +son, -cons, +cont, -delrel, -lat, -nas, 0strid, +voi, -sg, -cg, -ant, -cor, 0di
      str, +lab, +hi, -lo, +back, +round, -velaric, 0tense, -long, 0hitone, 0hireg]>,
       <Segment [+syl, +son, -cons, +cont, -delrel, -lat, -nas, 0strid, +voi, -sg, -cg, 0ant, -cor, 0di
      str, -lab, +hi, -lo, -back, -round, -velaric, +tense, -long, 0hitone, 0hireg]>,
       <Segment [-syl, -son, +cons, -cont, -delrel, -lat, -nas, 0strid, -voi, -sg, -cg, +ant, +cor, -di
      str, -lab, -hi, -lo, -back, -round, -velaric, 0tense, -long, 0hitone, 0hireg]>]
```

Wrapping up

- Next class:
  - Speech corpora, datasets
  - praat and speech data format
  - Audio file format conversion
  - Forced alignment overview
- 3rd progress report due Monday!
- Also coming up: project presentations. Dates/presenters fixed.