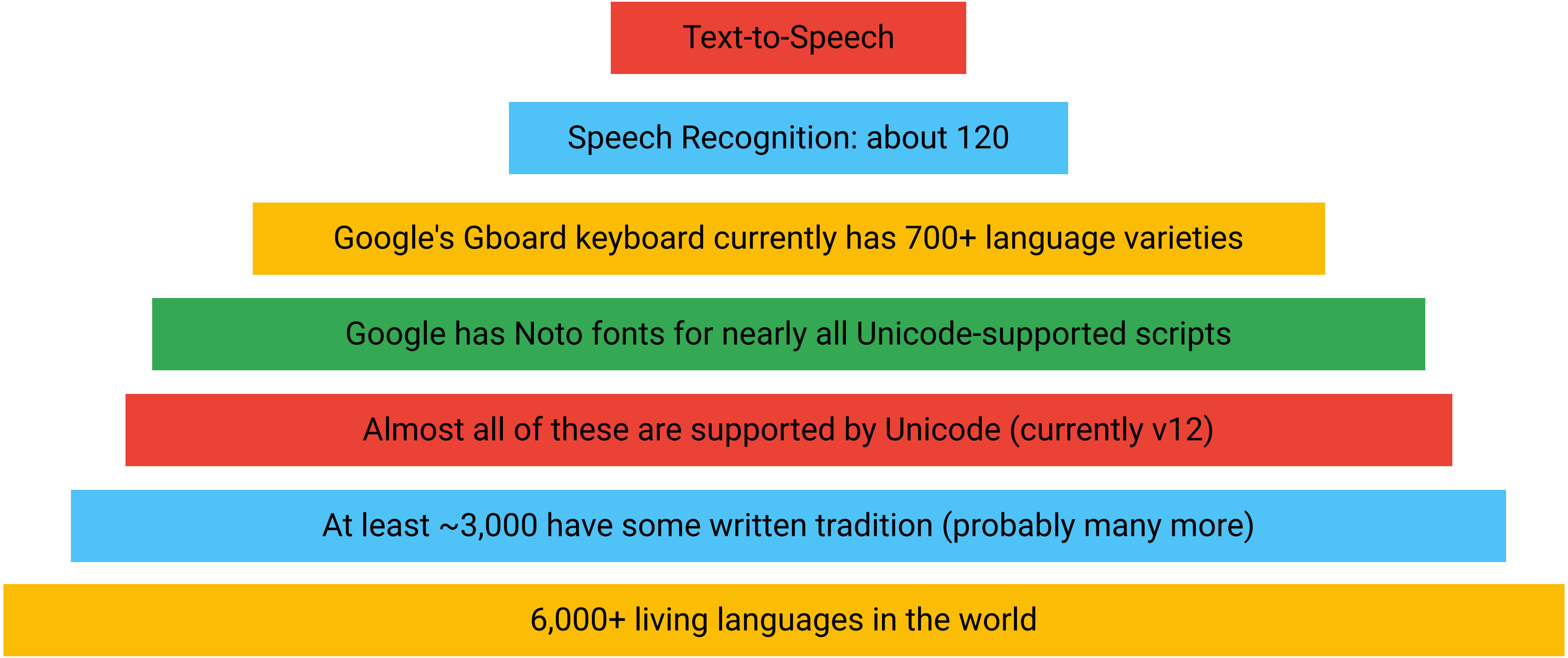
Technology for African Languages

Daan van Esch Google Speech

Talk at *Digital Humanities: the Perspective of Africa*
Lorentz Center, Leiden, the Netherlands
July 2, 2019

The Language Technology Pyramid



Text-to-Speech

Speech Recognition: about 120

Google's Gboard keyboard currently has 700+ language varieties

Google has Noto fonts for nearly all Unicode-supported scripts

Almost all of these are supported by Unicode (currently v12)

At least ~3,000 have some written tradition (probably many more)

6,000+ living languages in the world

Language Technology In Use the World Over

- Smartphones increasingly ubiquitous

- Communities everywhere are using language technology to…
  - ...**communicate** and keep in touch, e.g. on **social media**
  - ...**find** information, e.g. (voice) search
  - ...**create content**, e.g. typing or voice dictation
  - ...get things done, e.g. voice **assistants**

- But what does it mean if your language is not (yet) supported?
  - Can be significantly more challenging to use them online
  - Why is it the case that some languages are not (yet) supported?
  - Let's take a look at the technological challenges

Before We Start: What's in a Name?

- Virtually all operating systems use **ISO 639 codes** instead of language names
  - "eng" for English, "nld" for Dutch, "igb" for Ebira, etc.
  - Helps **order** the world's **6,000** or so languages & prevents ambiguity
  - e.g. "Swiss German": de-CH (*Sie lässt ihn nicht schlafen.*) or gsw-CH (*Si lat ne nid la schlafe.*)?

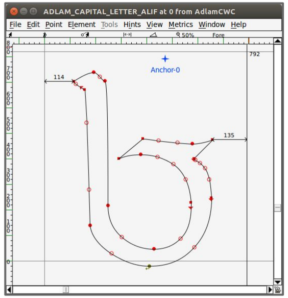
- General concept works well in languages where an accepted **standard** exists
  - Or at least a **clear demarcation** between one language and the next
  - But of course, in many situations, there are **dialect continuums**

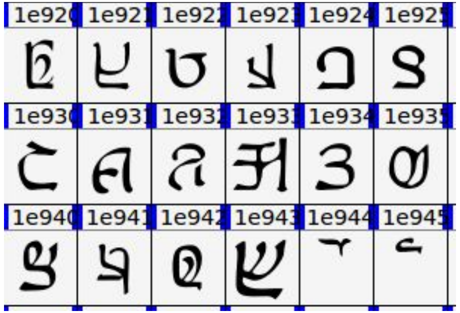
- **Mixing** languages (code-switching/translanguaging/…) also challenging
  - People mix and match from their full linguistic inventory
  - Technology finds it easier to operate on one variety at a time

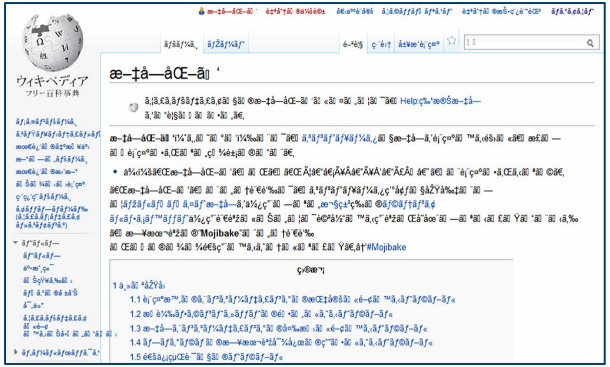
The Base of the Pyramid

- **Unicode**: encoding system for the world's writing systems
  - Computers represent everything in 0's and 1's under the hood
  - Unicode defines how to map these binary values to human writing systems
  - e.g. "DH" is 01000100 01001000

- **Fonts** are needed to determine the exact appearance (google.com/get/noto)
  - Long-standing support for the Latin alphabet, Ge'ez syllabary, etc.
  - In recent years, increasingly wide support for scripts like Vai, Adlam & Bamum




1e920
1e921
1e922
1e923
1e924
1e925
1e930
1e931
1e932
1e933
1e934
1e935
1e940
1e941
1e942
1e943
1e944
1e945



æ–‡å—åŒ–ã‘

ã,¦ã,£ã,ãƒšãƒ‡ã,£ã,¢ã § ã ®æ–‡å—åŒ–ã ' ã «ã ¤ã „ã ¦ã ¯ã€ Help:ç‰'æ®Šæ–‡å— ã,'ã "è¦§ã ã ã •ã „ã€,

æ–‡å—åŒ–ã 'ï¼ˆã,,ã "ã °ã 'ï¼‰ã "ã ¯ã€ ã,³ãƒ³ãƒ"ãƒ¥ãƒ¼ã,¿ã §æ–‡å—ã,'è¡¨ç¤ºã ™ã,‹éš›ã «ã€ æ£ã —ã è¡¨ç¤ºã •ã,Œã ªã „ç ¾è±¡ã ®ã "ã "ã€,

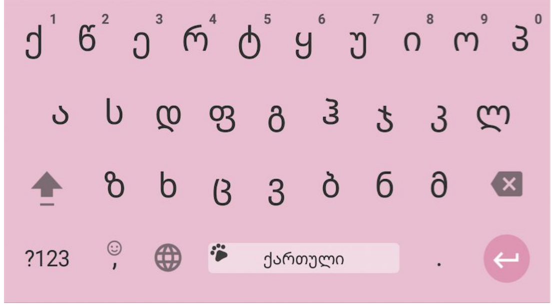
- ä¾‹ï¼šã€Œæ–‡å—åŒ–ã 'ã€ ã Œã€ ã€ŒÃ¦â€"â€¡Ã¥Ââ€"Ã¥Å'â€"Ã£Â â€˜ã€ ã "è¡¨ç¤ºã •ã,Œã,‹ã ªã ©ã€,

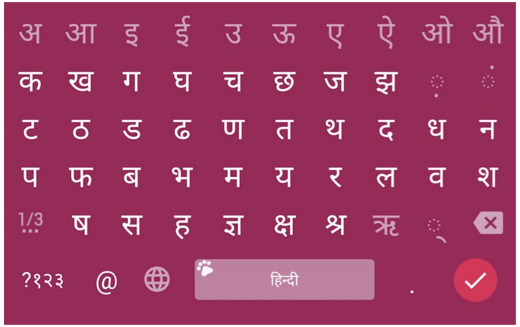
ã€Œæ–‡å—åŒ–ã 'ã€ ã "ã „ã †è"€è'‰ã ¯ã€ ã,³ãƒ³ãƒ"ãƒ¥ãƒ¼ã,¿ç'°å¢ƒã §åŽŸå‰‡ã "ã — ã ¦ãƒžãƒ«ãƒ ãƒ ã,¤ãƒ³æ–‡å—ã,'ä½¿ç"¨ã —ã ªã „æ¬§ç±³ç‰ã ®ãƒ©ãƒ†ãƒ³ã,¢ ãƒ«ãƒ•ã,¡ãƒ™ãƒƒãƒˆä½¿ç"¨è€…ã «ã Šã „ã ¦ã ¯è©²å½"ã ™ã,‹ç"¨èªžã Œå˜åœ¨ã —ã ªã ‹ã £ã Ÿã "ã "ã ‹ã,‰ ã€ æ—¥æœ¬èªžã ®"Mojibake"ã "ã „ã †è"€è'‰ ã Œã ã ®ã ¾ã ¾é€šç"¨ã ™ã,‹ã ˆã †ã «ã ªã £ã Ÿã€‚ât'#Mojibake

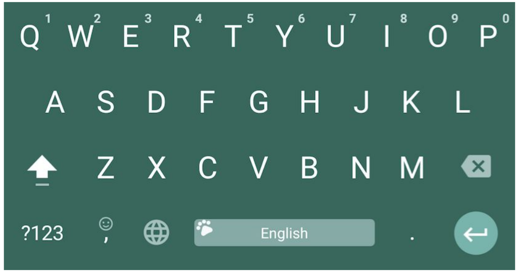
ç›®æ¬¡

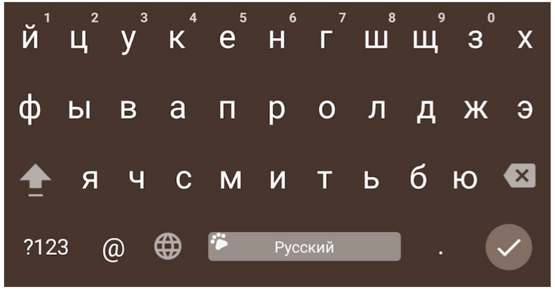
1 ä¸»ã ªåŽŸå›
1.1 è¡¨ç¤ºæ™'ã ®ã,¨ãƒ³ã,³ãƒ¼ãƒ‡ã,£ãƒ³ã,°ã ®æŒ‡å®šã «é–¢ã ™ã,‹ãƒˆãƒ©ãƒ–ãƒ«
1.2 æ è¼‰ãƒ•ã,©ãƒ³ãƒˆã,»ãƒƒãƒˆã ®é •ã „ã «ã,ˆã,‹ãƒˆãƒ©ãƒ–ãƒ«
1.3 æ–‡å—ã,¨ãƒ³ã,³ãƒ¼ãƒ‡ã,£ãƒ³ã,°ã ®å¤‰æ ›ã «é–¢ã ™ã,‹ãƒˆãƒ©ãƒ–ãƒ«
1.4 ãƒ—ãƒã,°ãƒ©ãƒ ã ®æ—¥æœ¬èªžå¯¾å¿œã ®ç"¨ã •ã «ã,ˆã,‹ãƒˆãƒ©ãƒ–ãƒ«
1.5 é€šä¿¡çµŒè·¯ã §ã ®ãƒˆãƒ©ãƒ–ãƒ«

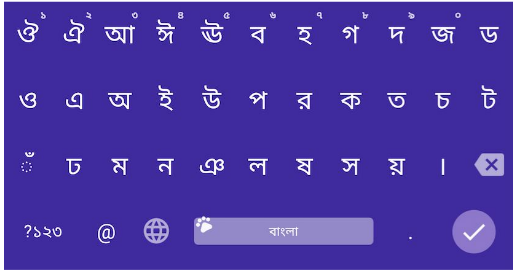
Keyboard Layouts

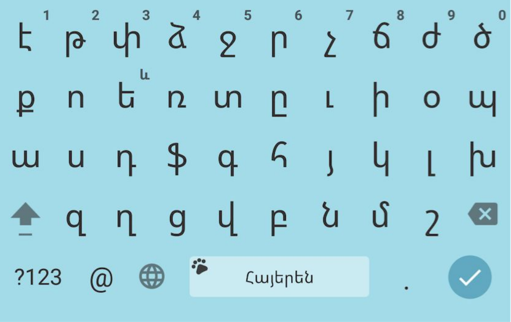












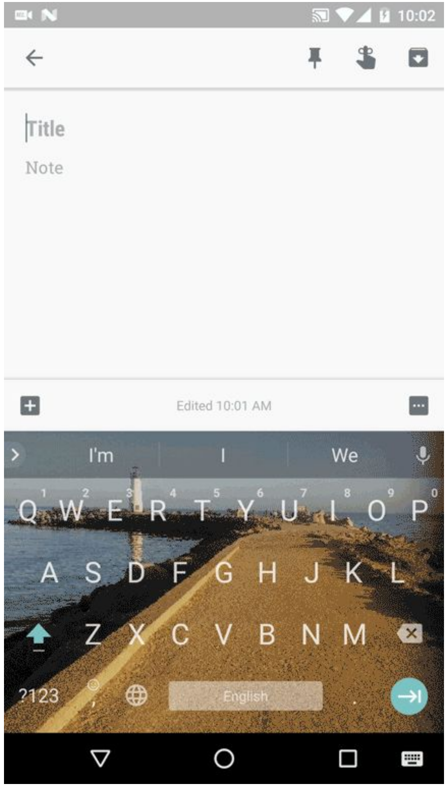
Google

Smart Keyboards

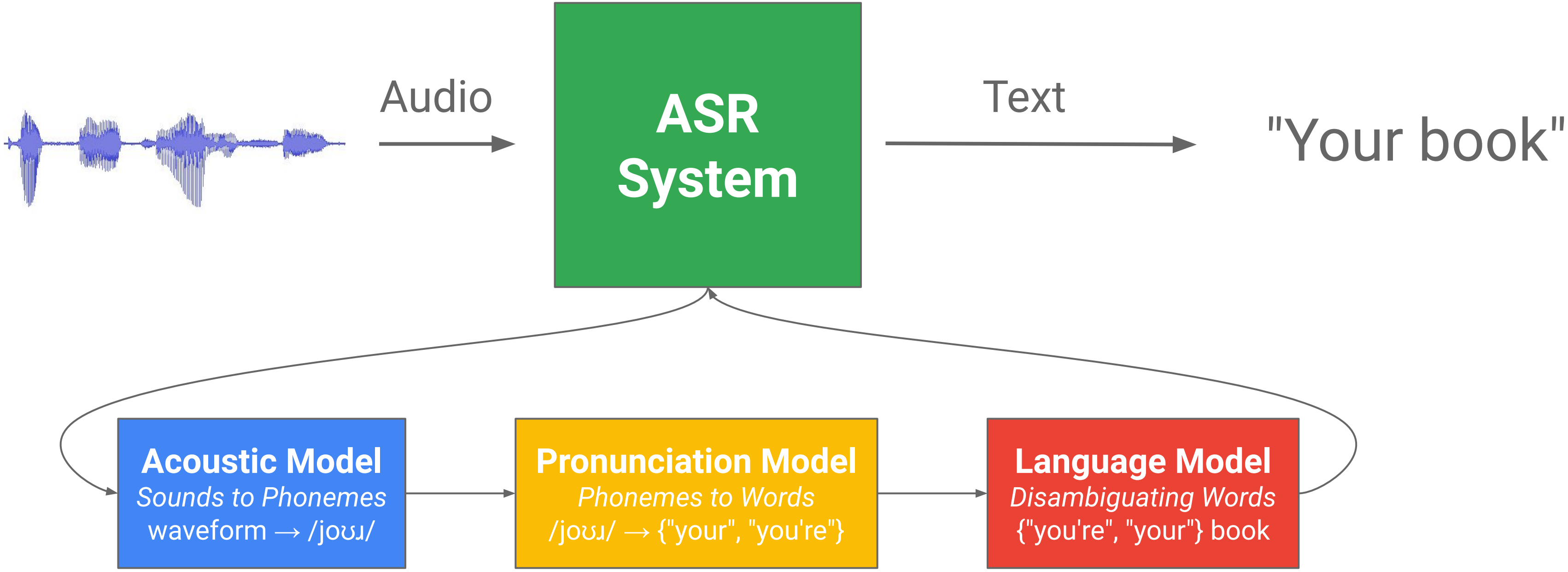
- Beyond a layout, use a **machine learning** language model
  - Trained on a **corpus** of text to predict **likely phrases** & sentences
  - Corpus can be gathered using simple elicitation questions

- Enables features to increase typing speed and accuracy
  - Auto-correction {defunitely, definately} → definitely
  - Next-word prediction How are → you
  - Completion superca → supercalifragilisticexpialidocious
  - Post-correction new York → New York
  - Glide typing
  - "I am going to my grandma's" → don't predict funeral!

- Available in **150+ African languages** on Android today



Speech Recognition



Audio

ASR System

Text

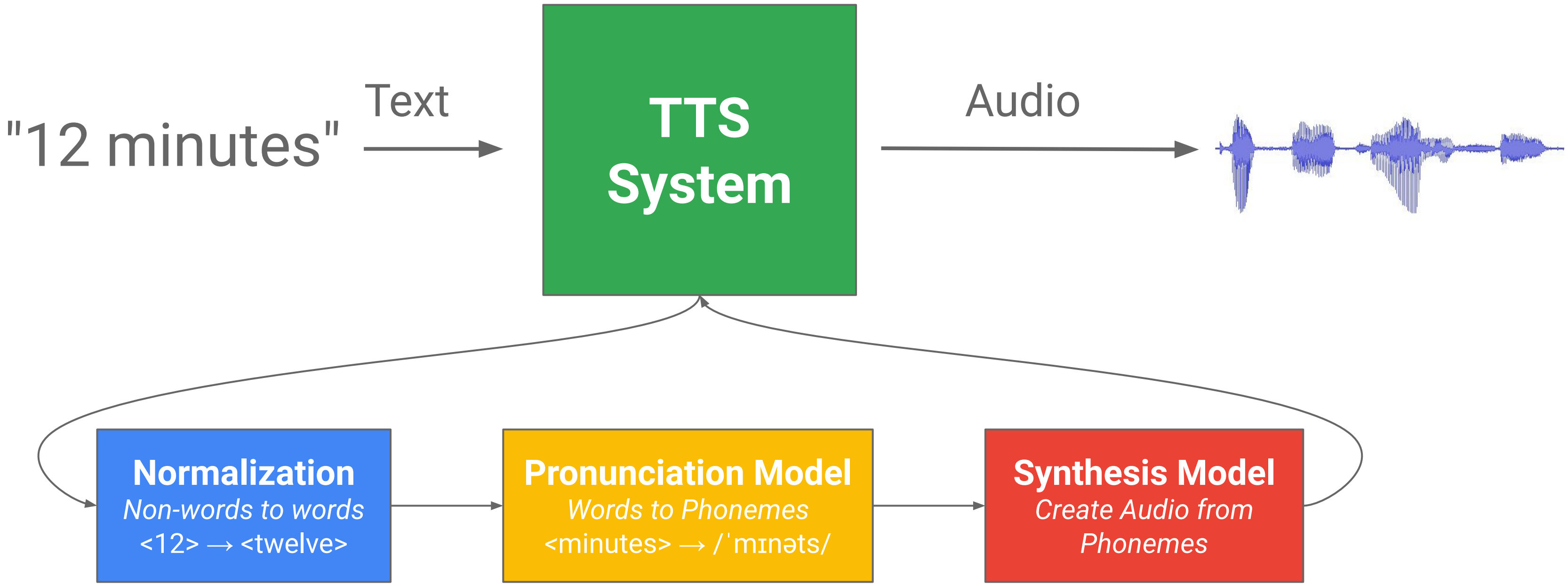
"Your book"

Acoustic Model
Sounds to Phonemes
waveform → /joʊɹ/

Pronunciation Model
Phonemes to Words
/joʊɹ/ → {"your", "you're"}

Language Model
Disambiguating Words
{"you're", "your"} book

Speech Synthesis



"12 minutes"

Text

TTS System

Audio

Normalization
Non-words to words
<12> → <twelve>

Pronunciation Model
Words to Phonemes
<minutes> → /ˈmɪnəts/

Synthesis Model
Create Audio from Phonemes

Speech Technology Needs

- Audio + transcriptions
  - **Less data** than for languages like English, but **more data exists** than you might think
  - And you can usually share data across languages: **transfer learning**
  - Initiatives like **SADiLaR** and Mozilla's **Common Voice** have awesome databases
  - For speech-to-text, voices should be as diverse as possible; for text-to-speech, target voice?

- Pronunciation lexicon
  - For most African languages, **grapheme-to-phoneme** relationships pretty straightforward
  - But sometimes **tone** is not marked in the orthography

- Text corpus
  - Can be **elicited**, mined from the **web**, created via **OCR** for paper archives, etc.

Language Research & Linguistic Engineering

- Some systems are **rule-based**
  - Linguists may write verbalization rules ("£5" → "five pounds")
  - For shallow orthographies, grapheme-to-phoneme ("G2P") mappings expressed in rules
  - What does a valid word look like in the orthography of the target language?
  - Human lexical knowledge, e.g. the place name <Reading> is pronounced /ˈɹɛdɪŋ/

- Others are more **data-driven**
  - Linguists commonly write **data annotation guidelines**, and supervise/do data annotation work
  - Important to address linguistic edge cases for **consistent, clean** data
  - Used to **train machine learning models**

- Hybrids are quite common
  - Many systems consist of both to some extent

Technology & Language Documentation

- Language documentation typically involves **many hours of recordings**
  - Transcribing these recording can be slow and arduous
- Can technology help?
  - We think so! To some extent!
- Working with ARC Centre of Excellence for the Dynamics of Language (CoEDL)
  - Built the Elpis toolkit → connects with ELAN, Praat & Transcribe, trains ASR on transcribed data
  - Automatically proposes candidate transcriptions for everything else
  - Designed to be easy to use for fieldwork linguists
- Already trained by CoEDL linguists on 10+ languages, more to come
  - Open-source project on GitHub & see also SLTU 2018 paper

Learning More

- MOOCS: Look for Machine Learning, Natural Language Processing
  - But typically quite heavy on the math, more so than is needed for linguistic applications
  - Lots of content at https://ai.google/education/

- Conferences: ACL, NAACL, EMNLP, LREC, Interspeech, SLTU, ComputEL
  - Mostly open-access proceedings, published online
  - Another good feed of papers is arXiv cs.cl (Computation & Language)

- Books
  - Natural Language Annotation for Machine Learning by James Pustejovsky & Amber Stubbs
  - Natural Language Processing with Python by Steven Bird, Ewan Klein & Edward Loper
  - Speech and Language Processing by Daniel Jurafsky & James Martin

Thank you!