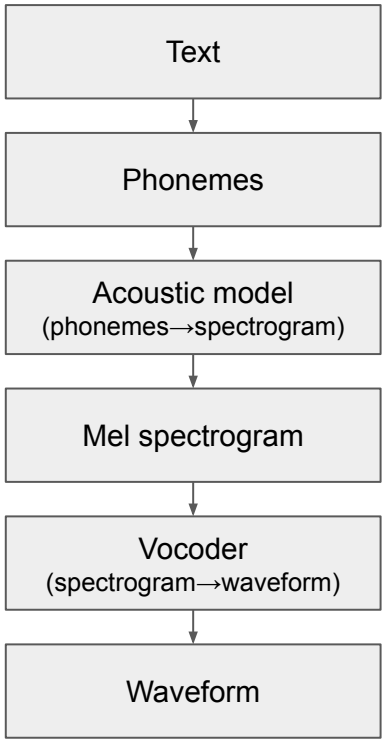
3. How machines process text

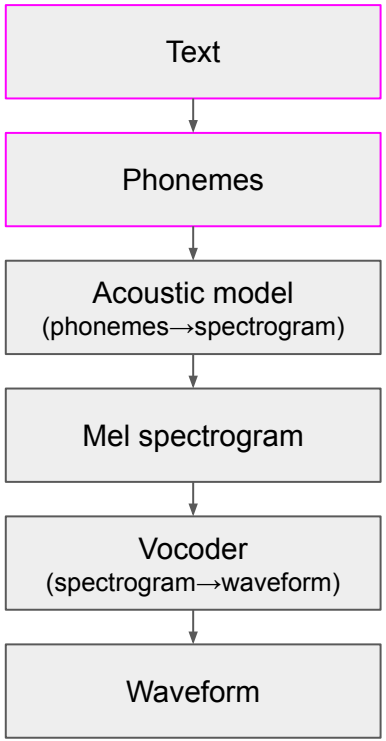
The Monster Text to Speech & Voice Cloning Course

THE SOUND OF AI

TTS pipeline



TTS pipeline



From characters to phonemes

Dr. Smith has 2 cats

/ˈdɑktər smɪθ hæz tuː kæts/

From characters to phonemes

Dr. Smith has 2 cats

?

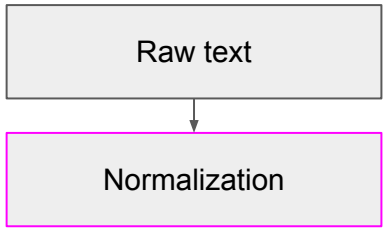
/ˈdɑktər smɪθ hæz tuː kæts/

Text processing

Raw text

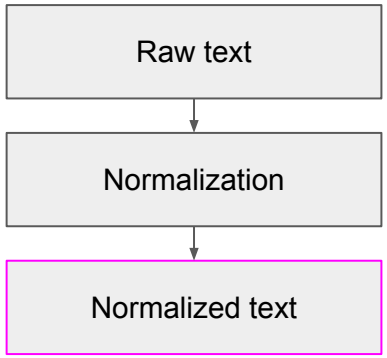
Dr. Smith has 2 cats

Text processing



Dr. Smith has 2 cats

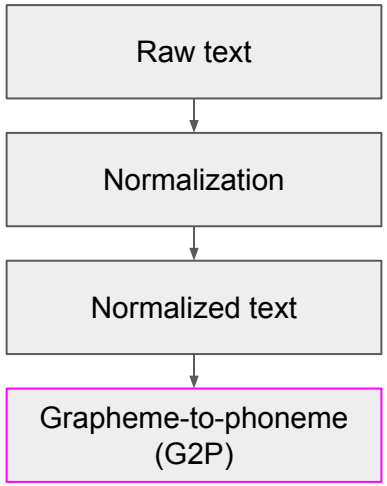
Text processing



Dr. Smith has 2 cats

Doctor Smith has two cats

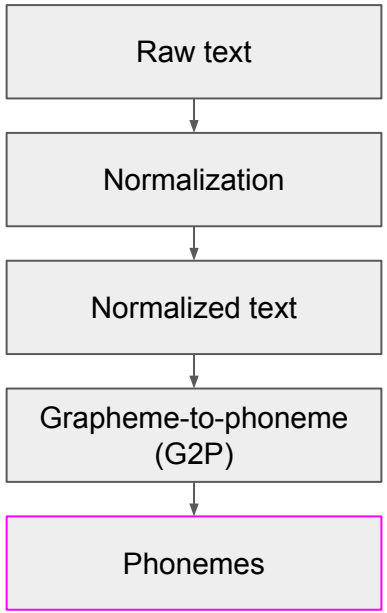
Text processing



Dr. Smith has 2 cats

Doctor Smith has two cats

Text processing

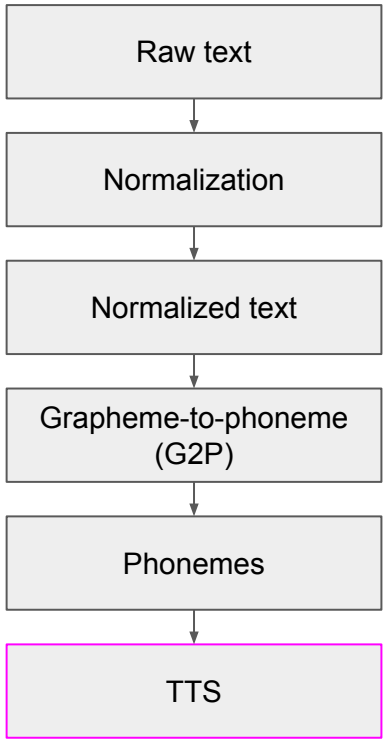


Dr. Smith has 2 cats

Doctor Smith has two cats

/ˈdɑktər smɪθ hæz tuː kæts/

Text processing

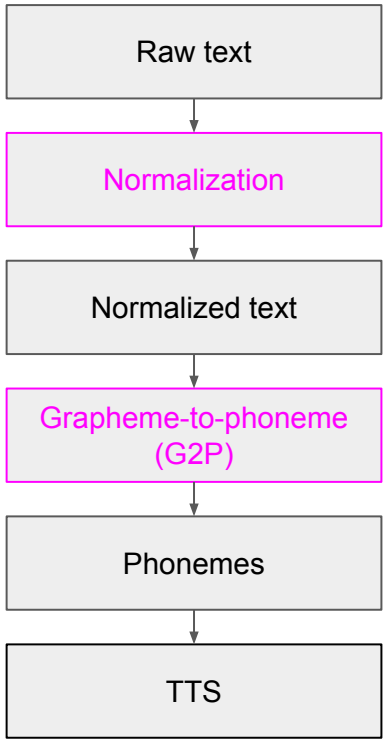


Dr. Smith has 2 cats

Doctor Smith has two cats

/ˈdɑktər smɪθ hæz tuː kæts/

Text processing



Dr. Smith has 2 cats

Doctor Smith has two cats

/ˈdɑktər smɪθ hæz tuː kæts/

Why two steps?

- *Normalization* handles symbols, numbers, formatting
- *G2P* handles pronunciation

Normalization: The problem

- “Dr.” → “Doctor" or “Drive”?

Normalization: The problem

- “Dr.” → “Doctor" or “Drive”?
- “123” → “one hundred twenty-three” or “one two three”?

Normalization: The problem

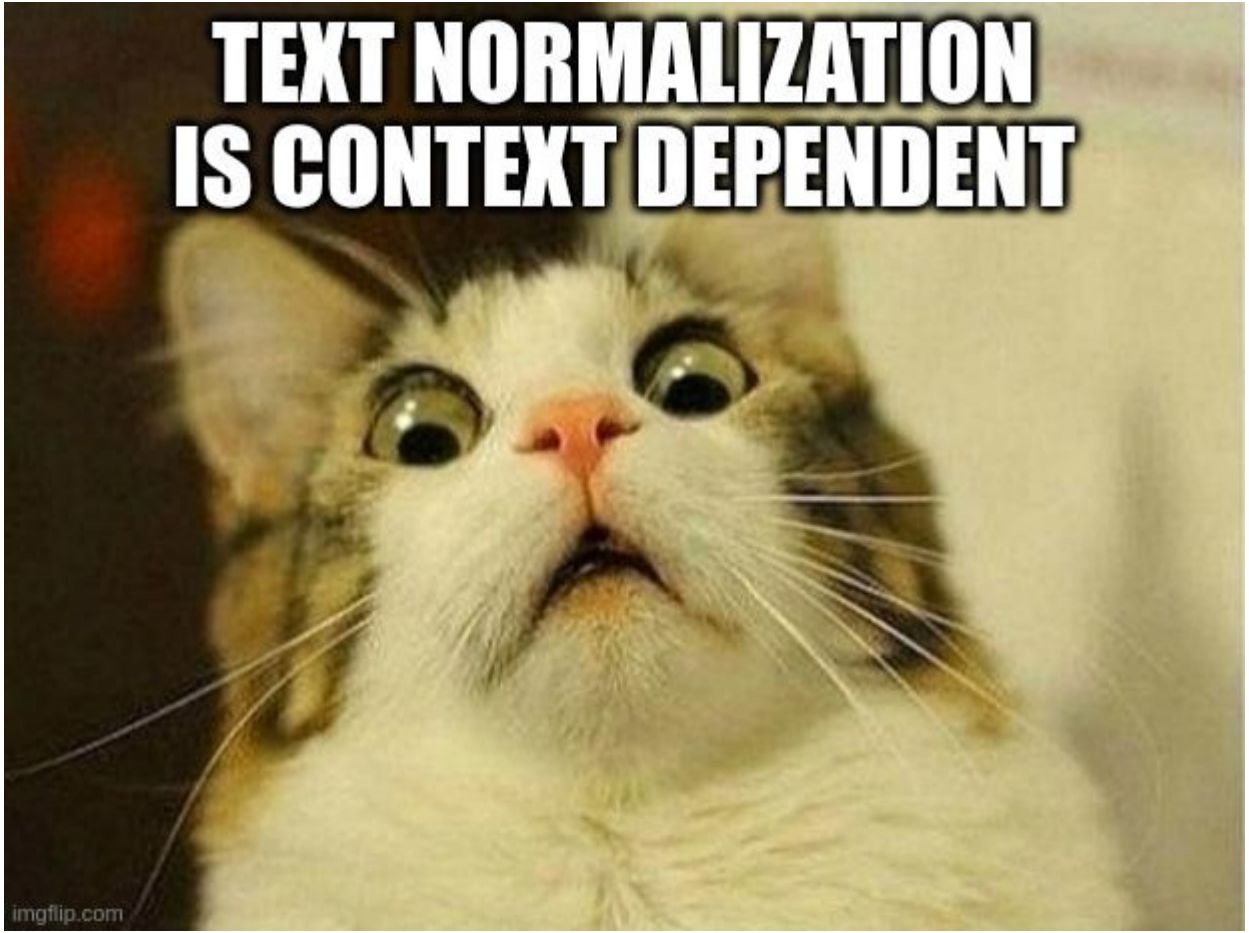
- “Dr.” → “Doctor" or “Drive”?
- “123” → “one hundred twenty-three” or “one two three”?
- “$5.99” → "five dollars ninety-nine cents" or "five point ninety-nine dollars"?

Normalization: The problem

- “Dr.” → “Doctor" or “Drive”?
- “123” → “one hundred twenty-three” or “one two three”?
- “$5.99” → "five dollars ninety-nine cents" or "five point ninety-nine dollars"?
- “12/5/2024” → “December fifth” (US) or “twelfth of May” (UK)?

Normalization

Input	Normalized Output
“Dr. Smith has 2 cats”	“Doctor Smith has two cats”
“It costs $5.99”	“It costs five dollars and ninety nine cents”
“Meet at 3:30pm”	“Meet at three thirty P M”
“I live at 123 Main St.”	“I live at one twenty three Main Street”


TEXT NORMALIZATION
IS CONTEXT DEPENDENT
imgflip.com

Normalization tools

- NeMo Text Processing (NVIDIA) - production-grade normalization
- num2words - number to word conversion
- unidecode - unicode to ascii
- Custom regex + tools


Normalization
G2P

Why Grapheme-to-Phoneme?

Spelling ≠ Pronunciation

English is chaotic!

- “ough”: tough /ʌf/, through /uː/, though /oʊ/, cough /ɔf/

English is chaotic!

- “ough”: tough /ʌf/, through /uː/, though /oʊ/, cough /ɔf/
- “read”: present /riːd/ vs past /rɛd/

English is chaotic!

- “ough”: tough /ʌf/, through /uː/, though /oʊ/, cough /ɔf/
- “read”: present /riːd/ vs past /rɛd/
- “bow”: bow tie /boʊ/ vs bow down /baʊ/

Why G2P?

- Machines have hard time reading
- Unambiguous mapping

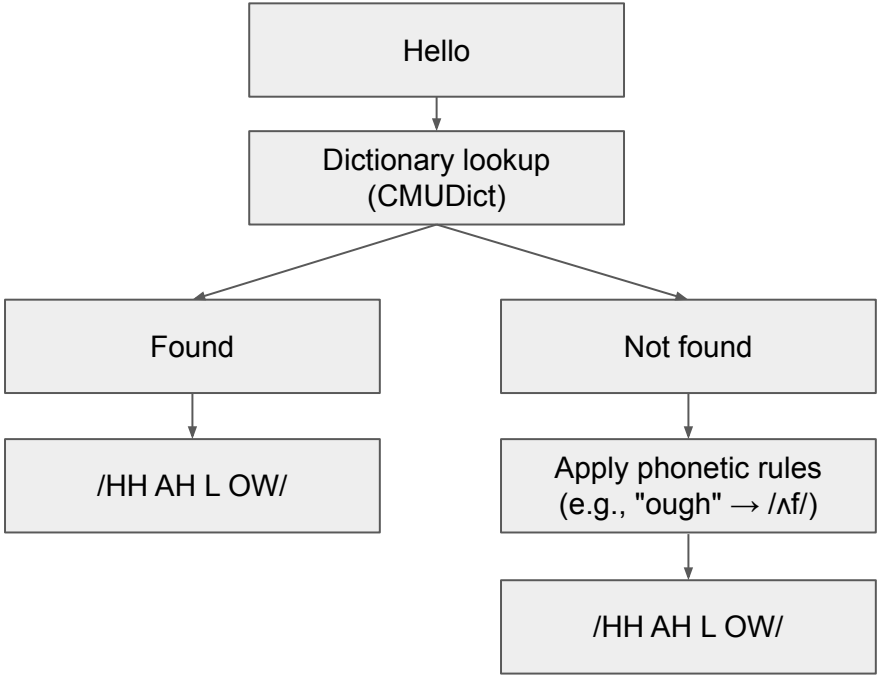
G2P approaches

- Rule-based
- Learned

Rule-based G2P

- Hand-crafted pronunciation rules
- Dictionary lookup
  - (CMUDict: 120K+ words, ARPABET)
- Fallback rules for unknown words

Rule-based G2P



Rule-based G2P

Pros

- Fast
- Interpretable
- Works well for common words

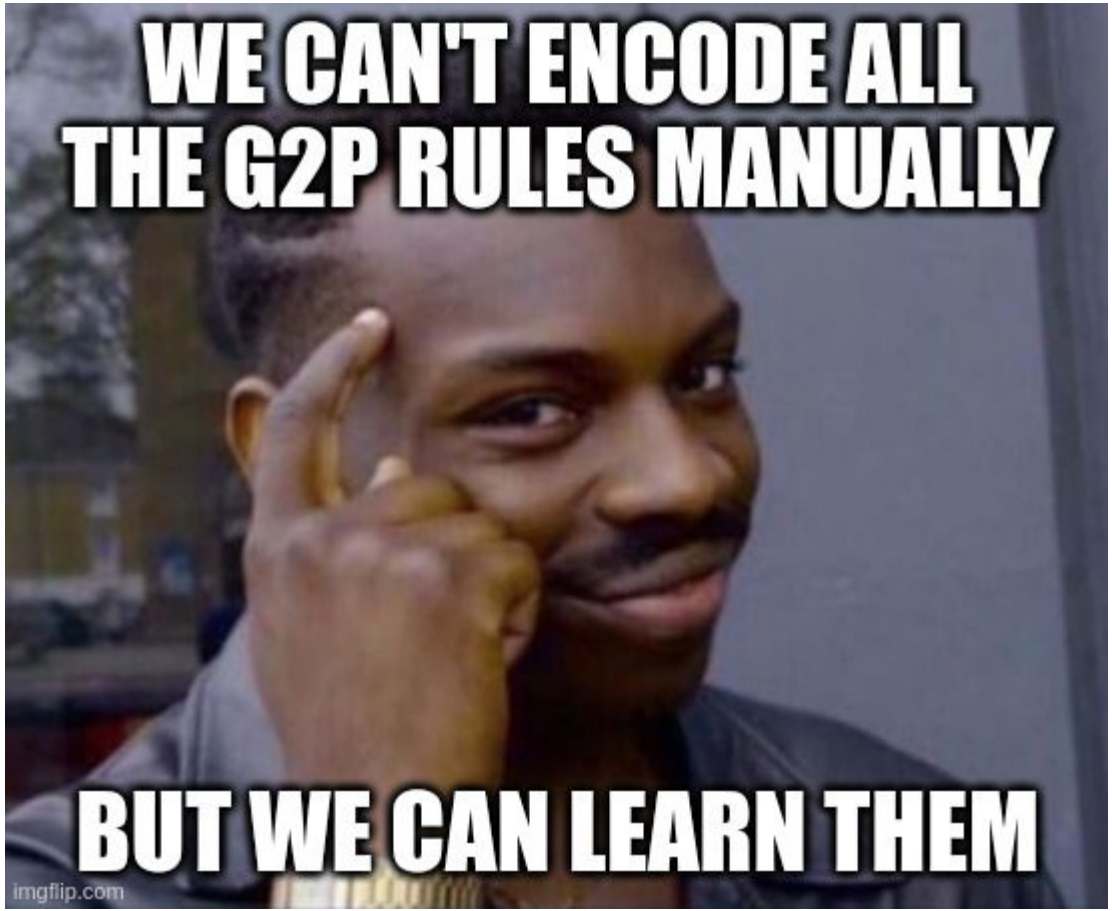
Rule-based G2P

Pros

- Fast
- Interpretable
- Works well for common words

Cons

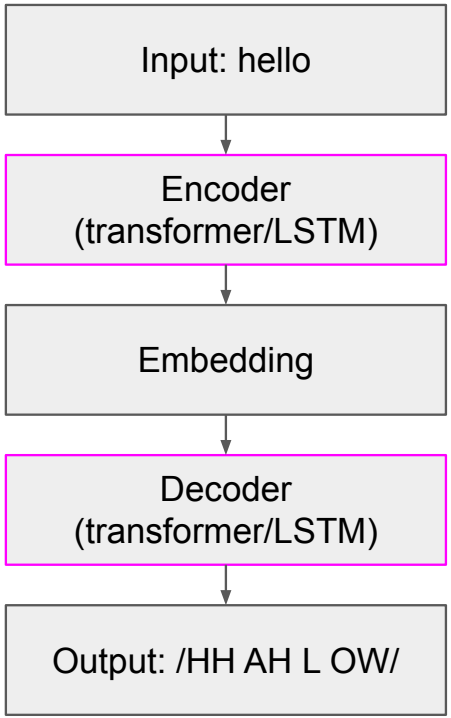
- Brittle
- Massive manual effort
- Poor generalization


WE CAN'T ENCODE ALL
THE G2P RULES MANUALLY
BUT WE CAN LEARN THEM
imgflip.com

Learned G2P

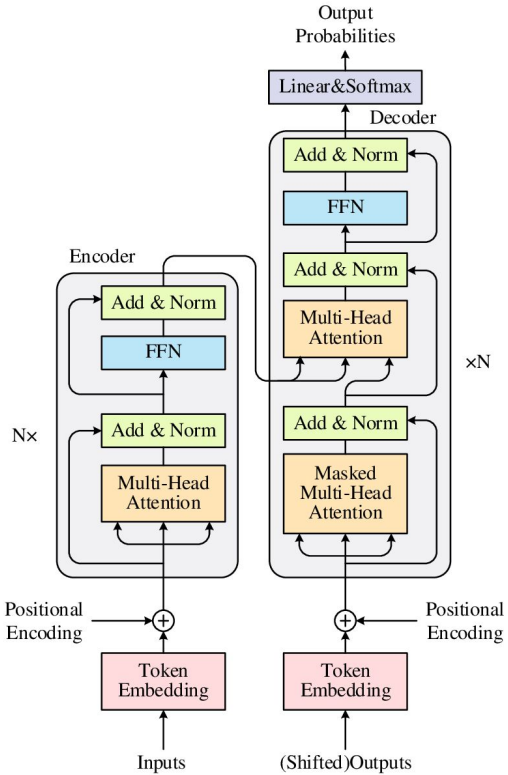
- Treat it like translation: graphemes→ phonemes
- Sequence-to-sequence models
- Train on words / phonemes pairs

Learned G2P



- Treat it like translation: graphemes→ phonemes
- Sequence-to-sequence models
- Train on words / phonemes pairs

Learned G2P



Learned G2P

Pros

- Learns complex patterns
- Generalizes to unseen words

Learned G2P

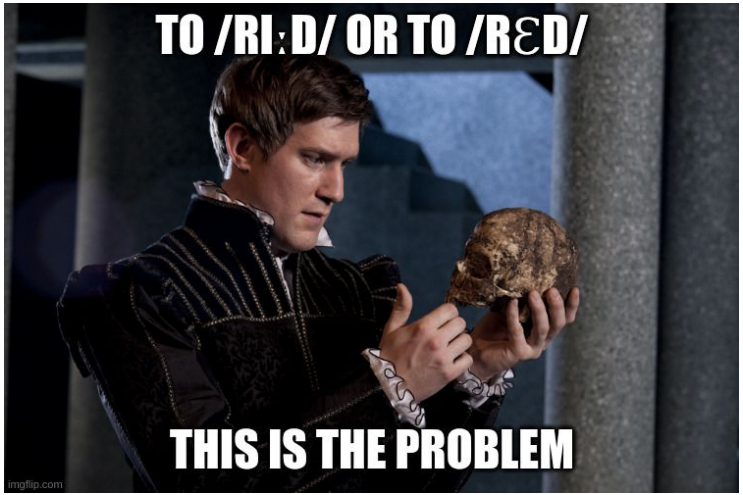
Pros

- Learns complex patterns
- Generalizes to unseen words

Cons

- Requires a lot of data
- Slower
- Difficult to interpret

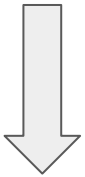
The ambiguity problem



- “I **read** books every day”→ /riːd/
- “I **read** a book yesterday”→ /rɛd/

The ambiguity problem

Same spelling, different sounds



Homographs

Solving ambiguity: Rule-based

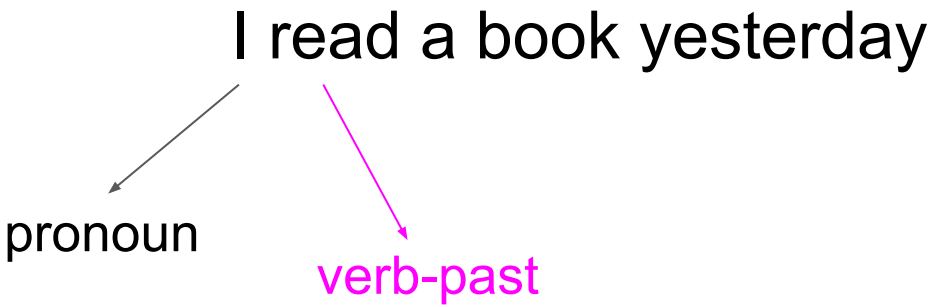
I read a book yesterday

Solving ambiguity: Rule-based

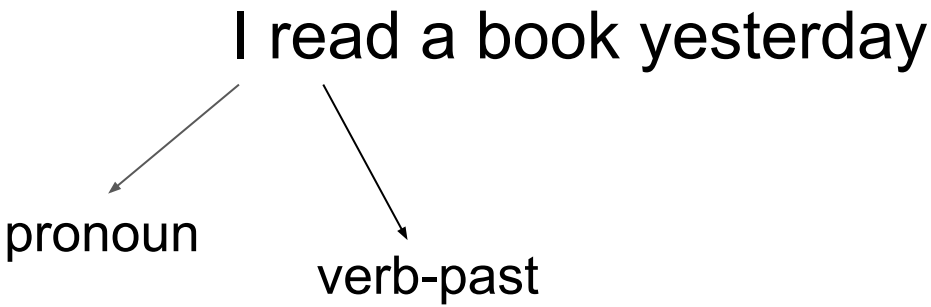
I read a book yesterday

pronoun

Solving ambiguity: Rule-based



Solving ambiguity: Rule-based



Rule: verb-past “read” → /rɛd/

Solving ambiguity: Rule-based

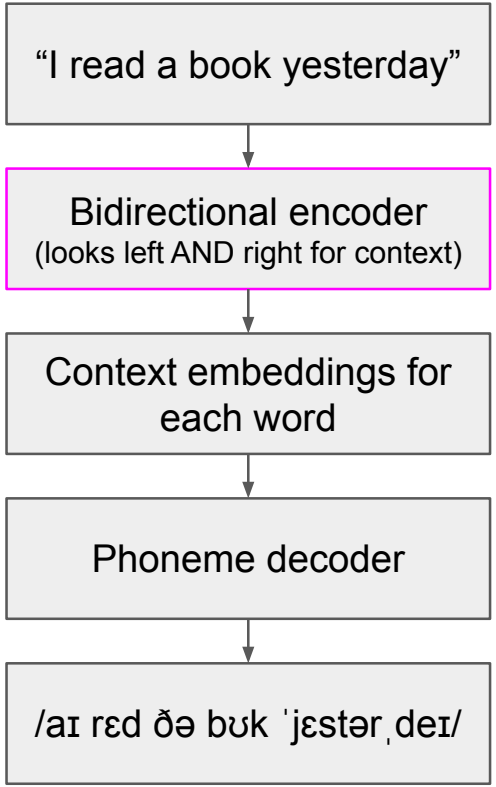
- Part-of-Speech (POS) tagging
- Requires separate POS tagger
- Accuracy for homographs isn’t great
- Simple: Pick most common pronunciation


CONTEXT
LIGHTYEAR
CONTEXT EVERYWHERE
imgflip.com

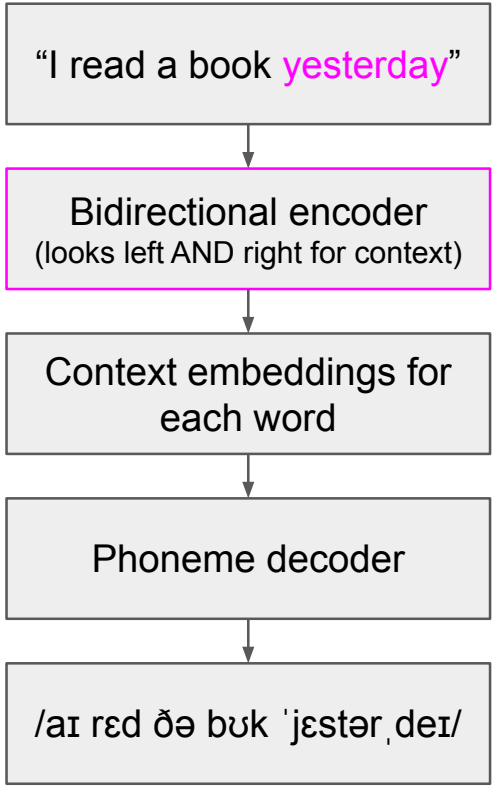
Solving ambiguity: Learned

Context-aware neural models

Bidirectional context



Bidirectional context



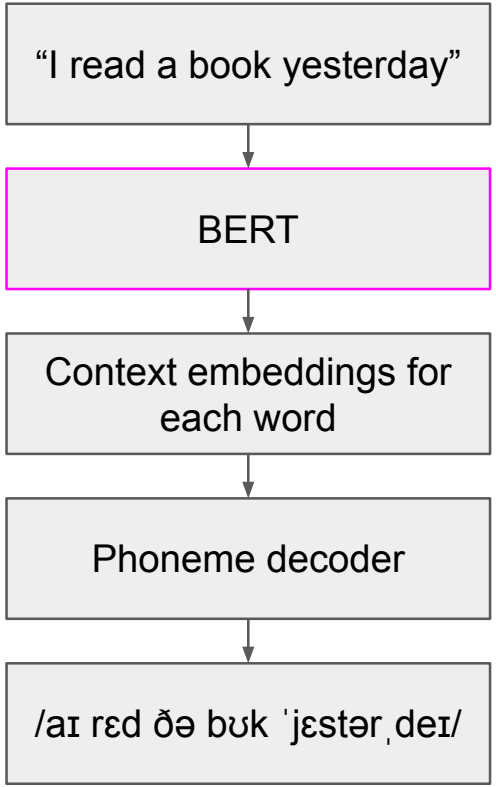
“yesterday” signals past tense

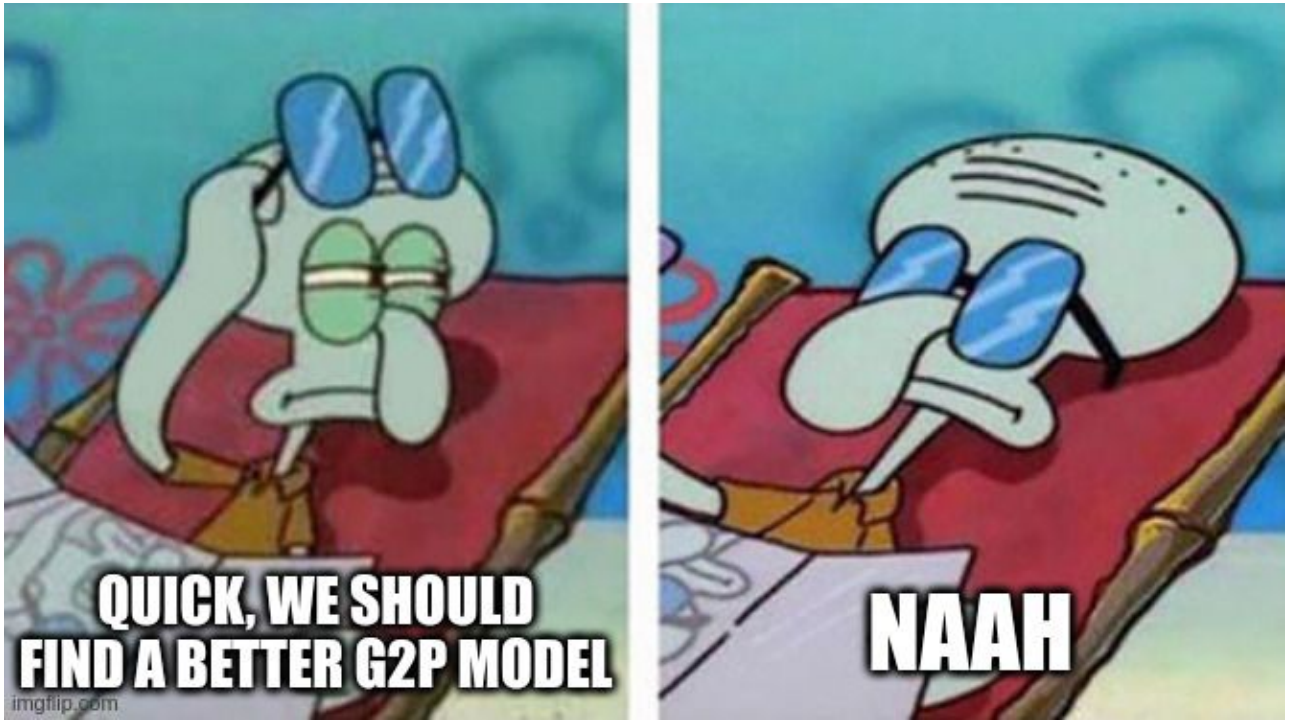
Solving ambiguity

Use pre-trained language models (BERT, RoBERTa, …)

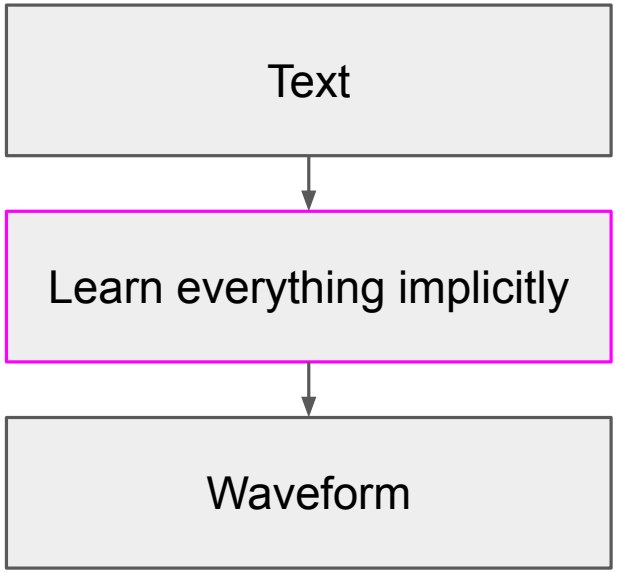
- Bidirectional architecture
- Understands context, meaning, relationships
- Text search, classification, …

Solving ambiguity

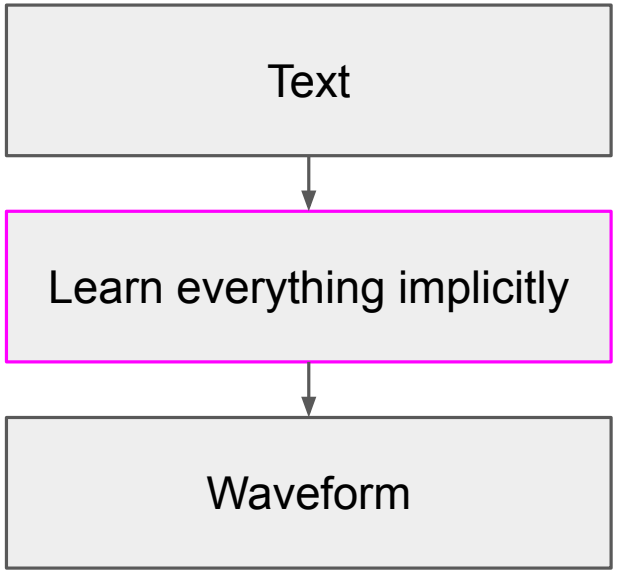



QUICK, WE SHOULD
FIND A BETTER G2P MODEL
imgflip.com
NAAH

Modern end-to-end TTS



Modern end-to-end TTS



Tacotron 2

G2P tools

- CMUdict: 120K+ words
- Phonemizer: multiple backends
- g2p_en: dictionary lookup + neural
- Epitran: 60+ languages
- DeepPhonemizer: transformer based

Takeaways

- Text processing:
  - Normalization (symbols → words)
  - G2P (words → phonemes)

Takeaways

- Text processing:
  - Normalization (symbols → words)
  - G2P (words → phonemes)
- Two approaches:
  - Rule-based: Fast, rigid, dictionary-driven
  - Learned: Flexible, context-aware, data-driven

Takeaways

- Text processing:
  - Normalization (symbols → words)
  - G2P (words → phonemes)
- Two approaches:
  - Rule-based: Fast, rigid, dictionary-driven
  - Learned: Flexible, context-aware, data-driven
- Ambiguity resolution:
  - POS tags, heuristics
  - Bidirectional context, language models