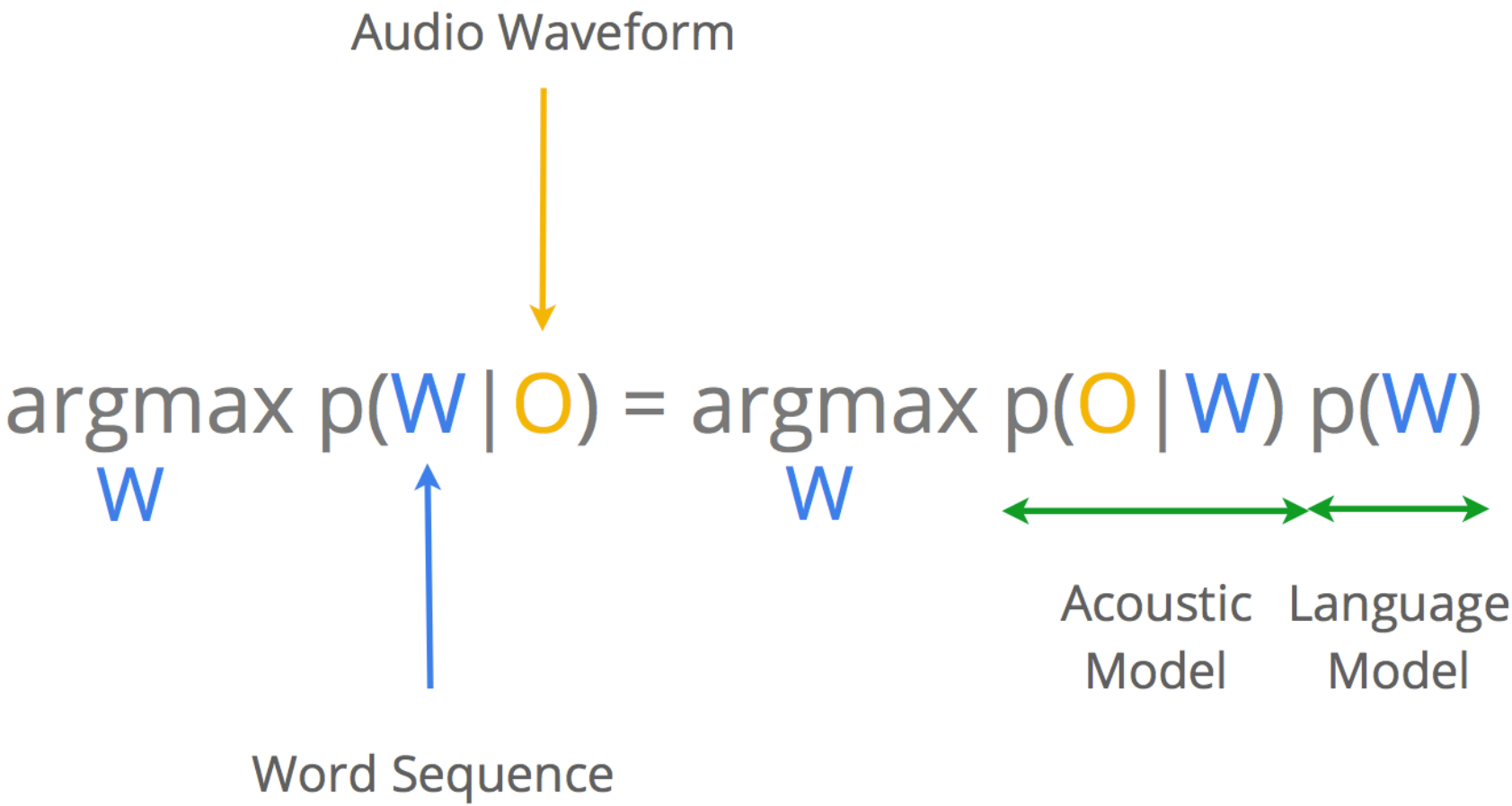
Pronunciation Modeling

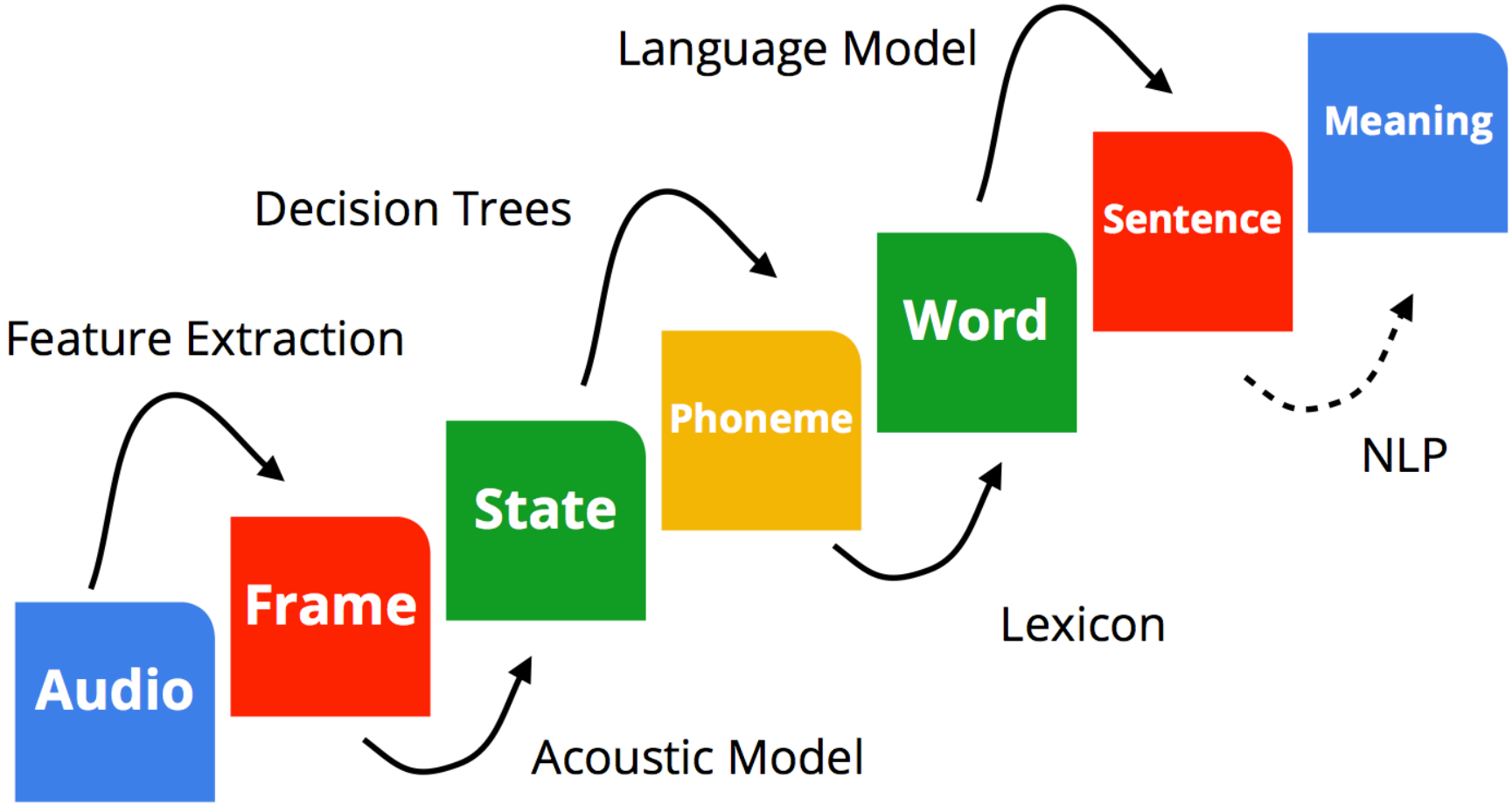
Te Rutherford

Bottom Line

- Fixing pronunciation is much easier and cheaper than LM and AM.
- The improvement from the pronunciation model alone can be sizeable.

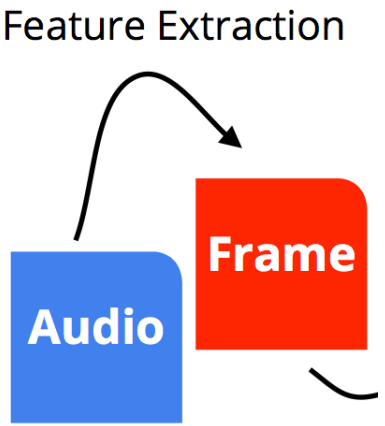

Audio Waveform
argmax p(W | O) = argmax p(O | W) p(W)
W
W
Acoustic Model
Language Model
Word Sequence

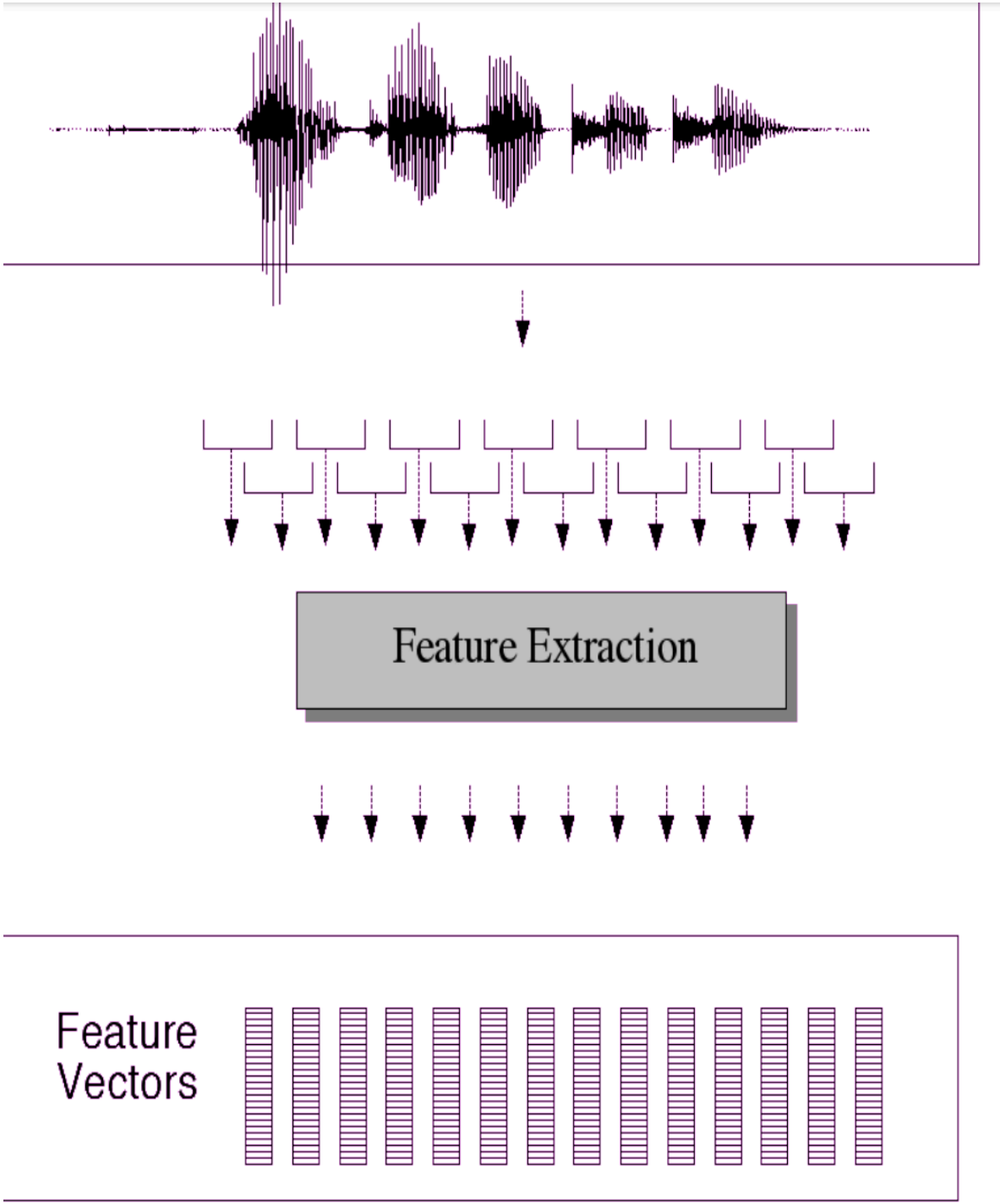
Overview of Speech Recognition



Audio to Feature Vector

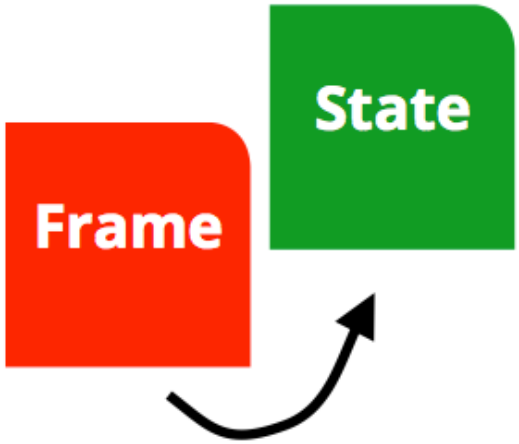
- Chop up the audio
- Take Fourier Transform
  - power of each frequency
- Featurize
  - MFCC, Cepstrum, PLP, etc.
  - Clip out the silence
  - Stacking (windowing)
    - use t-1 t-2 t-3 … and t+1 t+2 t+3 as features

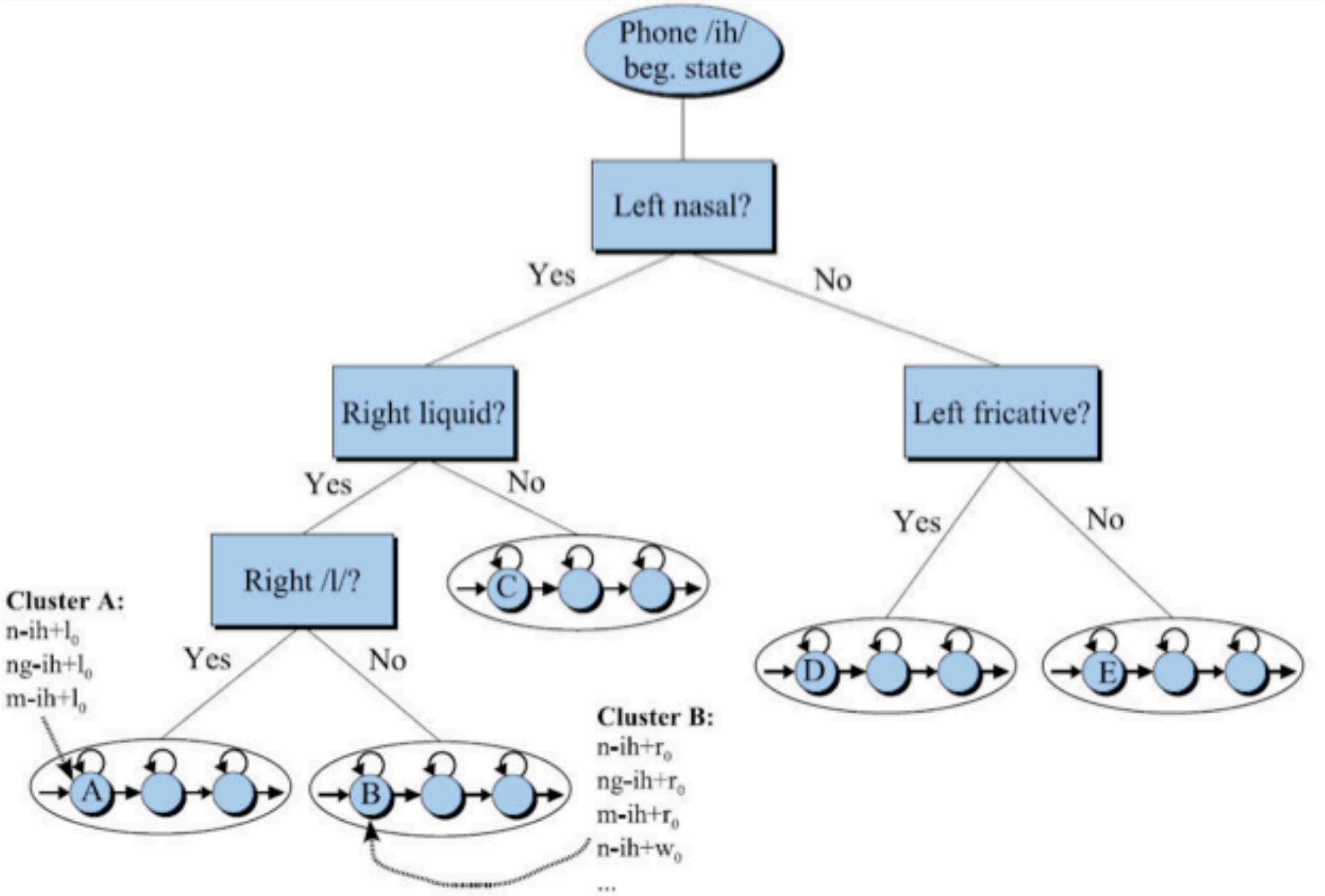



Feature Extraction
Feature
Vectors

Frame to State

- Acoustic model
  - a classifier (e.g. Neural Network)
    - input : a feature vector
    - output : a triphone state
- What is a triphone state?
  - type 1 /t/ : {vowel} + /t/ + {vowel}
  - type 2 /t/ : {nasal} + /t/ + {vowel}
- Use decision trees to cluster /t/ sounds from different environment.



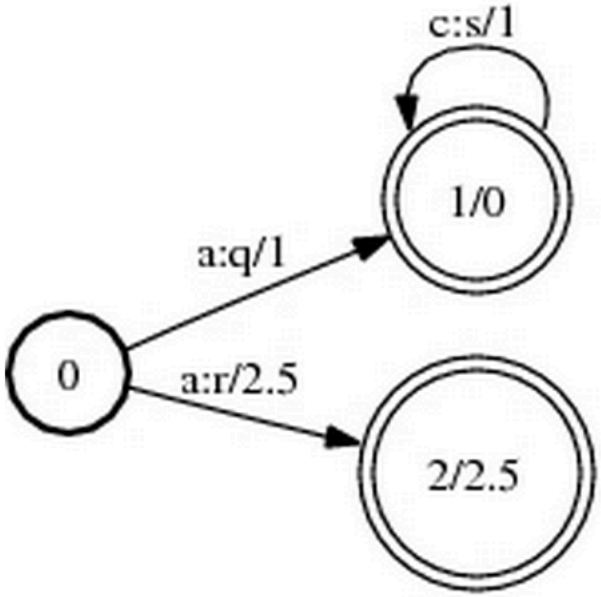

Phone /ih/
beg. state
Left nasal?
Yes
No
Right liquid?
Left fricative?
Yes
No
Yes
No
Right /l/?
C
D
E
Yes
No
Cluster A:
n-ih+l0
ng-ih+l0
m-ih+l0
A
B
Cluster B:
n-ih+r0
ng-ih+r0
m-ih+r0
n-ih+w0
...

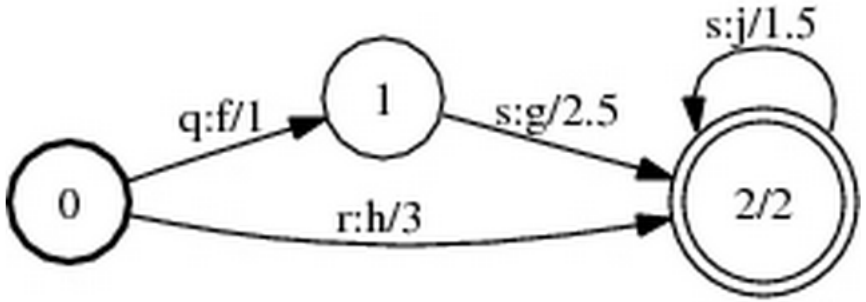
Phone Lattice

	time 1	time 2	time 3	time 4	…
type 1 t - beginning	0.2	0.11	0.001		
type 1 t - middle	0.1	0.23	0.001		
type 1 t - end	0.1	0.06	0.2		
type 2 t - beginning	0.05	0.02	0.2		
type 2 t - middle	0.05	0.05	0.005		
type 2 t - end	0.1	0.001	0.05		
type 1 k - beginning	0.03	0.06	0.12		
type 2 k - middle	0.02	0.02	0.42		
type 3 k – end	0.03	0.001	0.001		
…					

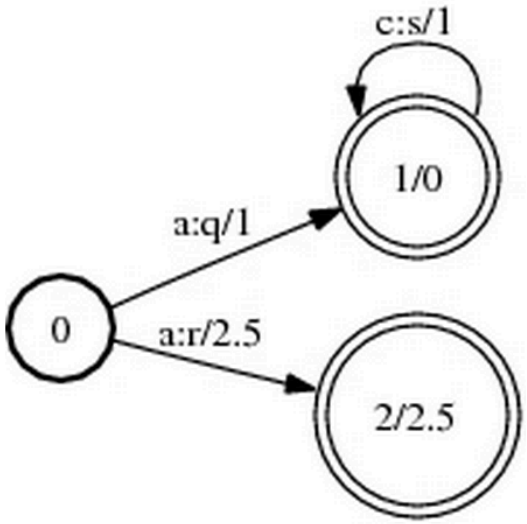
Weighted Finite State Transducer

- Essentially a search graph
- In speech, FST composition is used to prune the search graph and score the results.

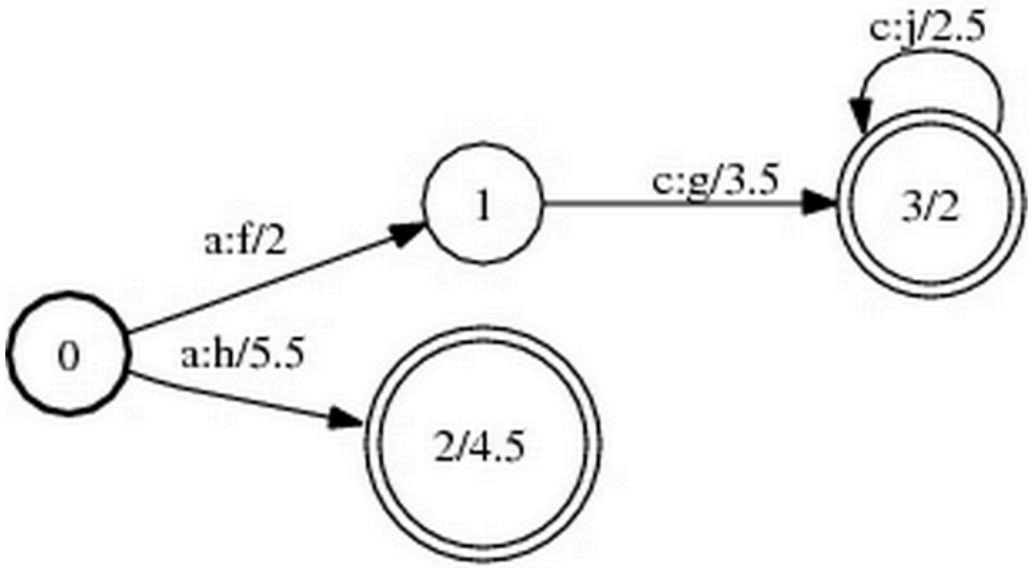




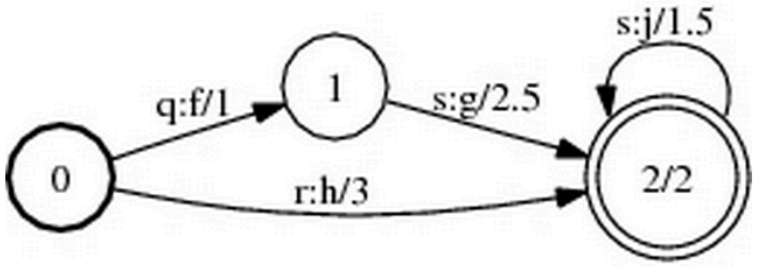
A:


c:s/1
1/0
a:q/1
0
a:r/2.5
2/2.5

A o B:

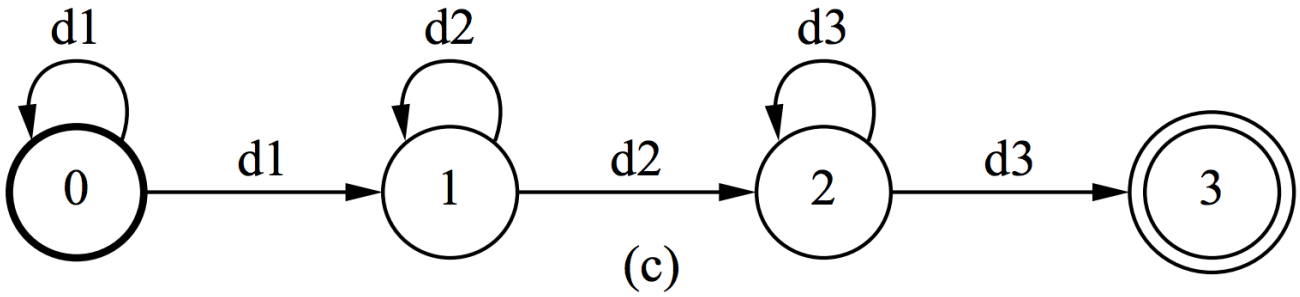

c:j/2.5
1
c:g/3.5
3/2
a:f/2
0
a:h/5.5
2/4.5

B:


s:j/1.5
1
q:f/1
s:g/2.5
0
r:h/3
2/2

HMM conversion

- Limiting what sequence of states is valid.
  - k1 k1 k3 k3 ao1 ao2 ao2 ao3 l1 l2 l3 l3 – not okay
  - k1 k1 k2 k2 l1 l1 l2 l3 l3 – not okay
  - k1 k2 k2 k2 k3 k3 ao1 ao2 ao3 ao3 l1 l1 l2 l3 – okay
- Use FST composition to prune the phone lattice



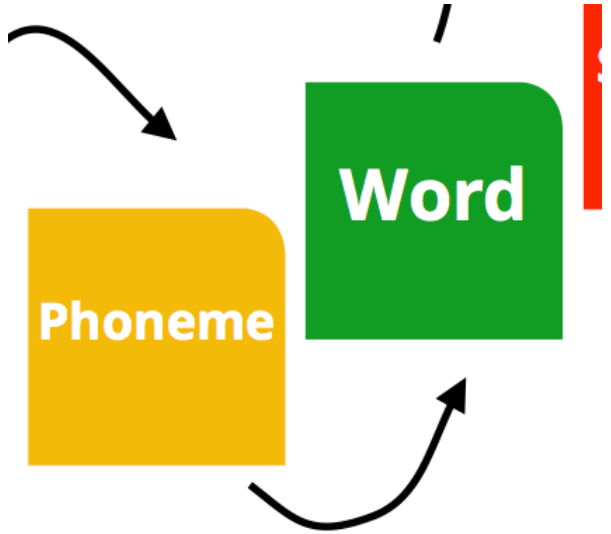
State to Phoneme

- Converting context-dependent triphone state back into phoneme
  - type 1 /t/ : {vowel} + /t/ + {vowel} → /t/
  - type 2 /t/ : {nasal} + /t/ + {vowel} → /t/

- Use FST Composition operation to convert (transduce)

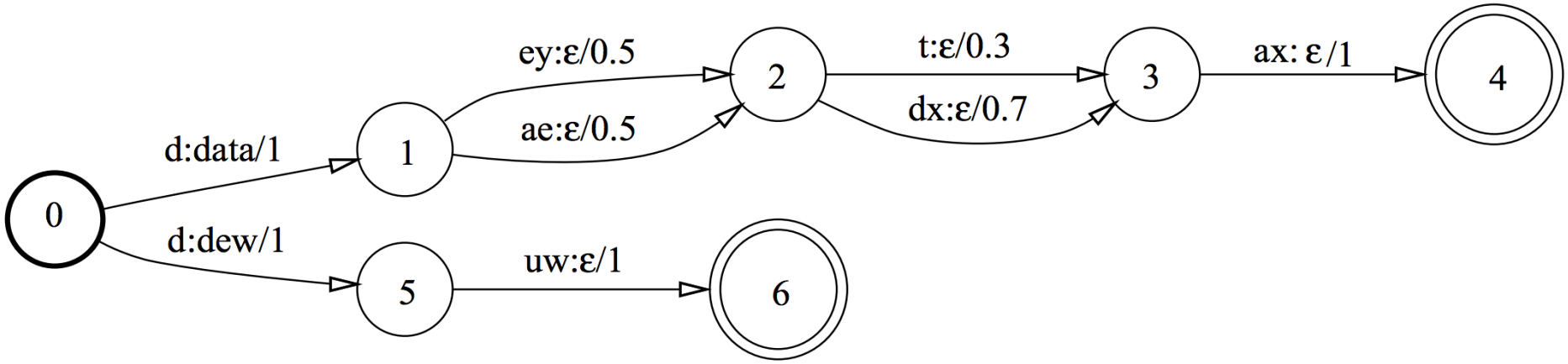
Phoneme to Word

- List of valid phoneme sequences
- Example pronunciation dictionary:
  - call : k ao l
  - dad : d ae d
  - mom : m aa m
- Backbone of the recognizer
  - the best phoneme sequence might not make a word…



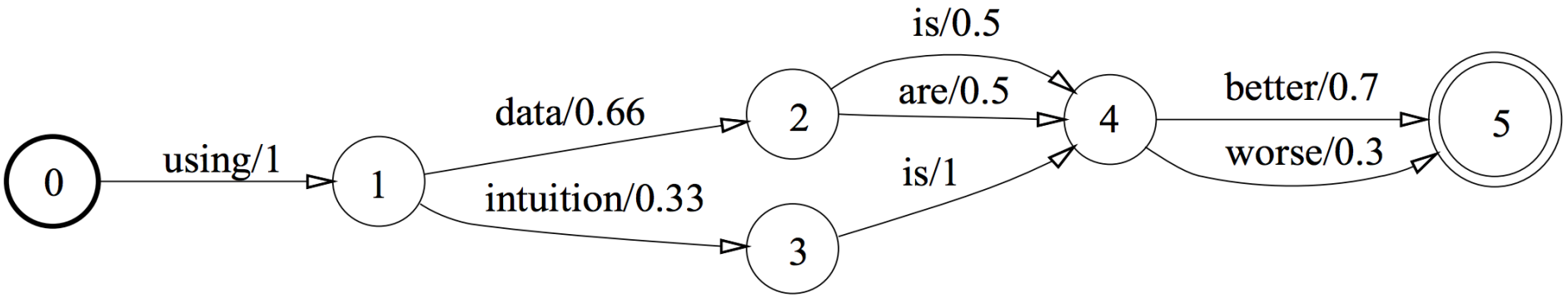
Pronunciation FSTs

- d ey t ax
  d ey dx ax
  d ae t ax
  d ae dx ax



Recognizing sequence of words

- Language model and grammar are a finite state transducers.
  - P(is | intuition)
  - P(is | data)



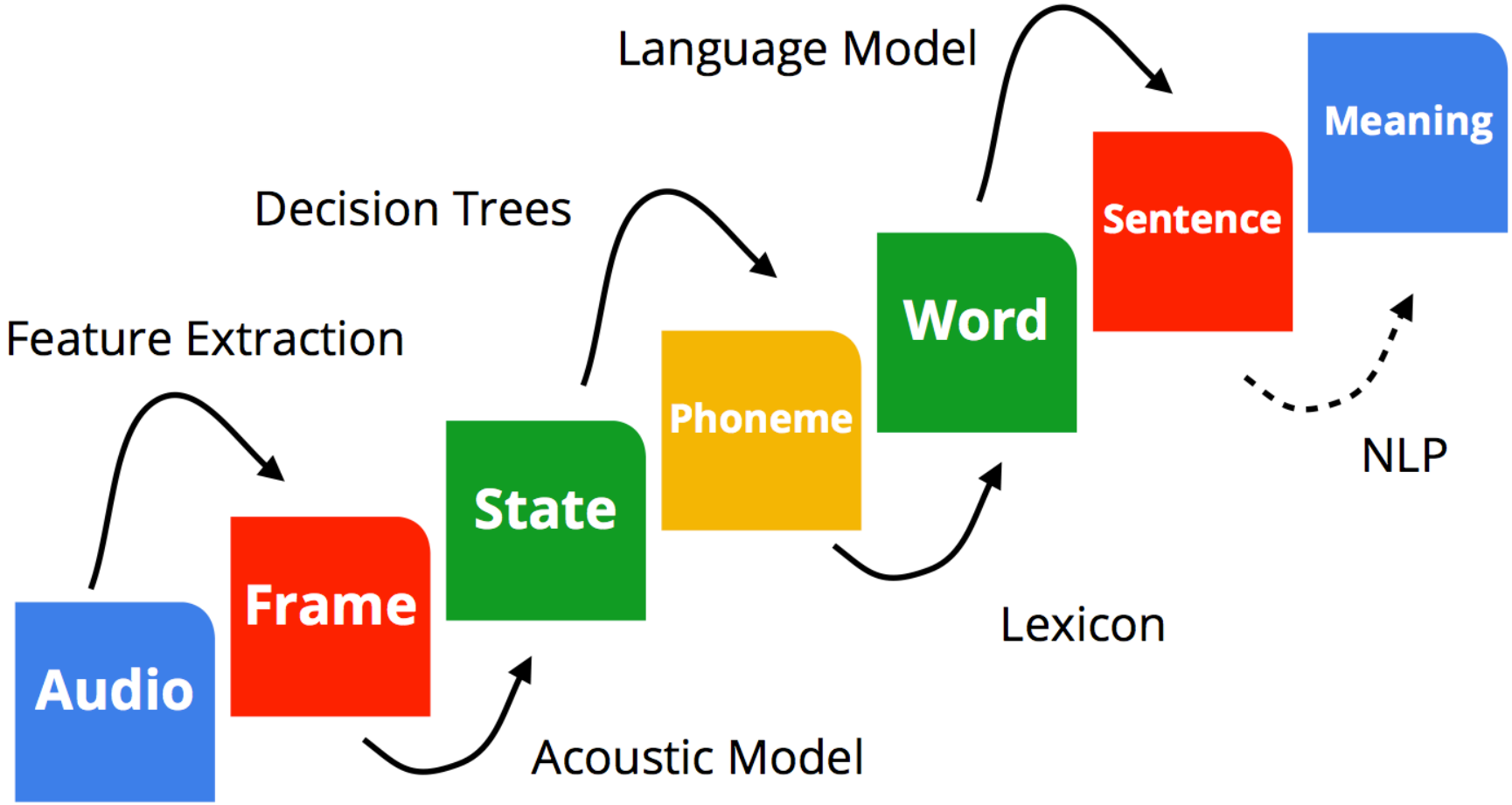
Search meets machine learning

- Compose a well-pruned search graph (HCLG)
  - HMM FSTs→ pruned phone graph
  - Context-independent FSTs → phoneme graph
  - Lexicon FSTs→ word graph
    (heavily pruned phoneme graph)
  - Grammar and LM FSTs → pruned and scored word graph

Search meets machine learning

- Compose a well-pruned search graph (HCLG)
  - HMM FSTs→ pruned phone graph
  - Context-independent FSTs → phoneme graph
  - Lexicon FSTs→ word graph
    (heavily pruned phoneme graph)
  - Grammar and LM FSTs → pruned and scored word graph
- Classify a sequence of feature vector (by acoustic model) → phone lattice
- Compose the phone lattice with HCLG and search

Overview of Speech Recognition



Lexicon is the key component

- The lexicon makes training and decoding possible.
  - limiting the size of the search graph
- The lexicon determines the words that can possibly be recognized.

Motivations for Pronunciation Modeling

- Suppose you are making a speech recognizer for a new language.
- Base dictionary and specialized dictionary
  - How is a dictionary made?
  - Can we automate it? semi-automate it?

Motivations for Pronunciation Modeling

- Suppose you are making a speech recognizer for a new language.
- Base dictionary and specialized dictionary
  - How is a dictionary made?
  - Can we automate it? semi-automate it?
- The errors from pronunciation modeling can be isolated and fixed relatively easily.

Pronunciation can be hard…

- For a dictation task, we face new words all the time.
  - "Play a song by Ke$ha"
  - "Install Spotify"
  - "Who sings Gangnam Style?"
  - "Tell me about Zoe Deschanel"

Ask linguists for help

- These new words are usually hard and might require several pronunciations
  - Gotye
  - Rihanna
  - Gangnam Style
  - Ellie Goulding

Ask linguists for help

- These new words are usually hard and might require several pronunciations
  - Gotye
  - Rihanna
  - Gangnam Style
  - Ellie Goulding
- Disadvantages
  - Expensive
  - “Too accurate”
  - Slow(er) turn-around time

Refresh the pronunciation dictionary

- Apply Grapheme-to-Phoneme conversion (G2P) to a list of new words
- Rule-based approach

Refresh the pronunciation dictionary

- Apply Grapheme-to-Phoneme conversion (G2P) to a list of new words
- Rule-based approach
- Statistical approach: Hidden Markov Model trained with Baum-Welch on the base dictionary
  - state : set of phonemes
  - observation : letters or groups of letters

Performance of statistical approach

- HMM-based models perform quite well.
- English, Dutch, German, and French which is the hardest?

Performance of statistical approach

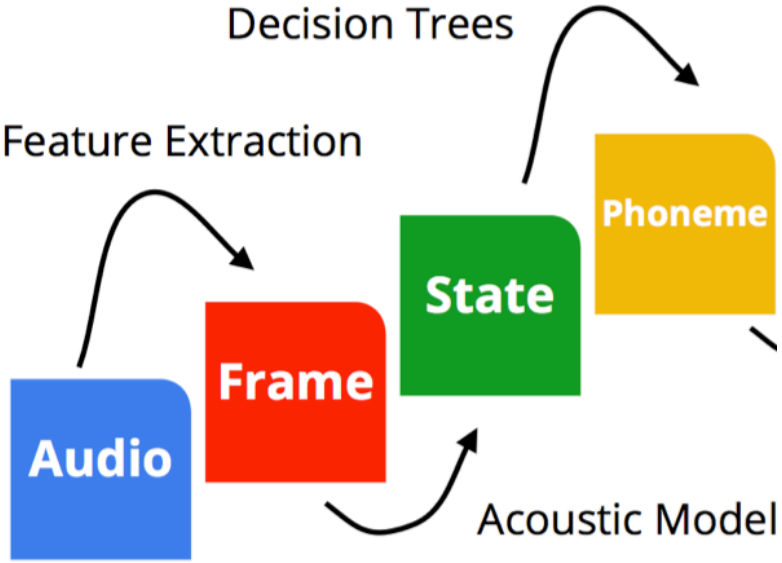
- HMM-based models perform quite well.
- English, Dutch, German, and French which is the hardest?

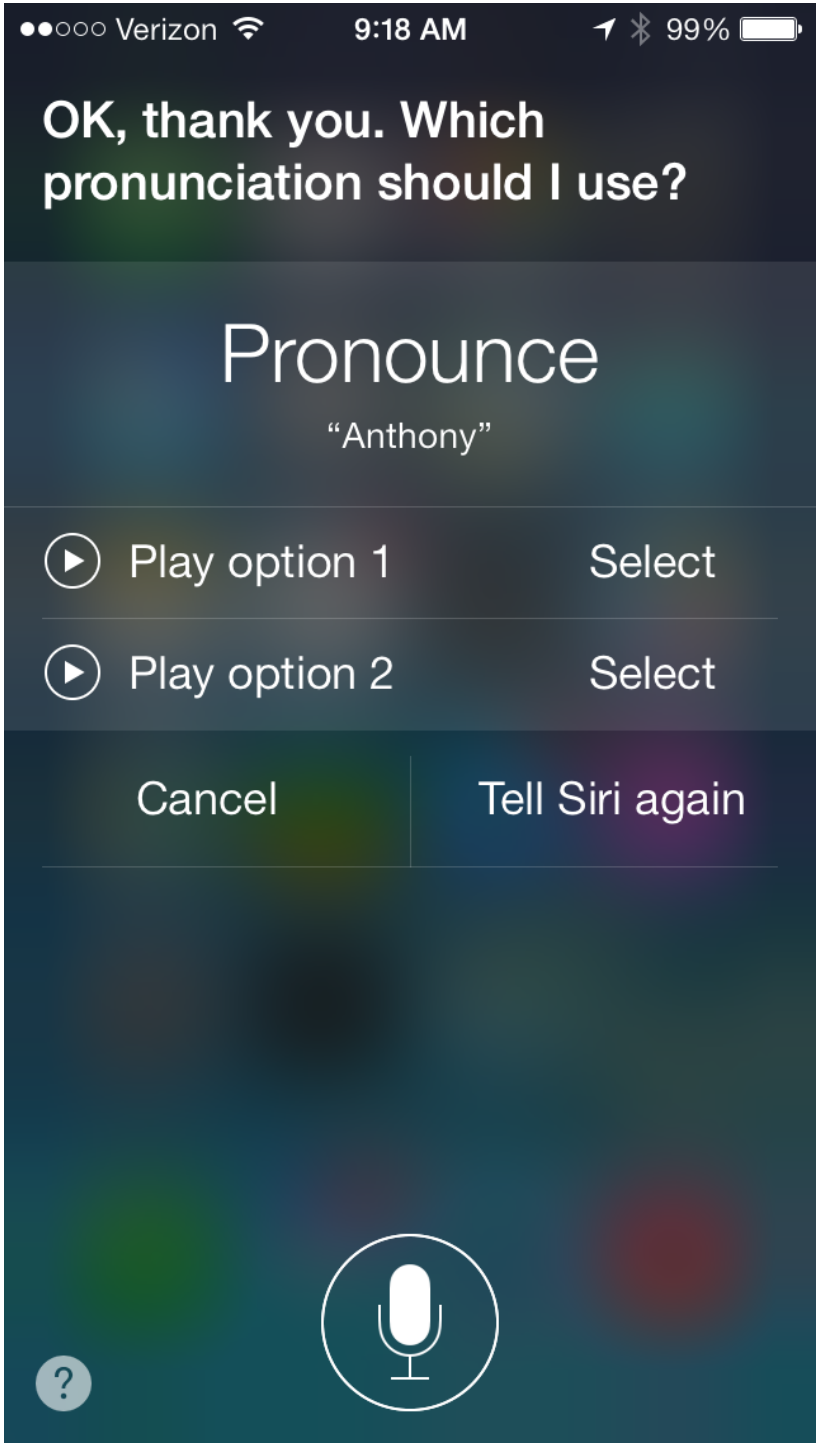
Language	Data set	M-M+HMM
English	CMUDict	65.6 ± 0.72
English	Celex	83.6 ± 0.63
Dutch	Celex	91.4 ± 0.24
German	Celex	89.8 ± 0.59
French	Brulex	90.9 ± 0.45

Jiampojamarn et al, 2007

Semi-automatic approach

- Learn the pronunciation of a word from a native speaker but not a linguist.
- Why is this possible?




●●○○○ Verizon
9:18 AM
99%
OK, thank you. Which pronunciation should I use?
Pronounce
“Anthony”
Play option 1
Select
Play option 2
Select
Cancel
Tell Siri again

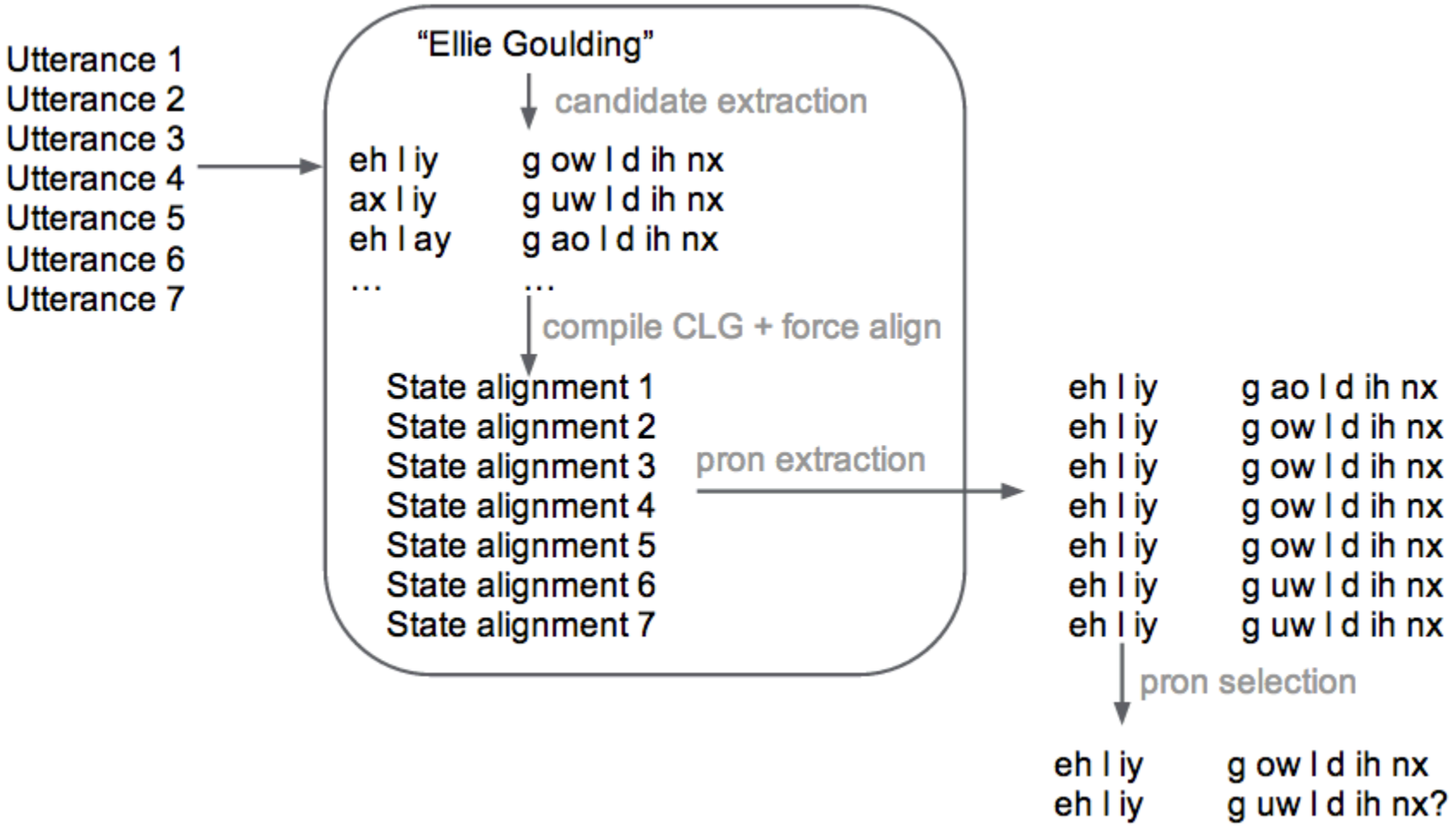
Pronunciation Learning from Crowd-sourcing

Rutherford et al., 2014

Learn automatically from human

- For each word or phrase (transcription),
  - get people to pronounce it
  - use the speech recognizer to extract the pronunciation
    - we already have the acoustic model + transcription!
- Fast : <1 day turn-around
- Cheap : ~5 cents/transcription
- Sounds great. But would it work?

Overview of the algorithm

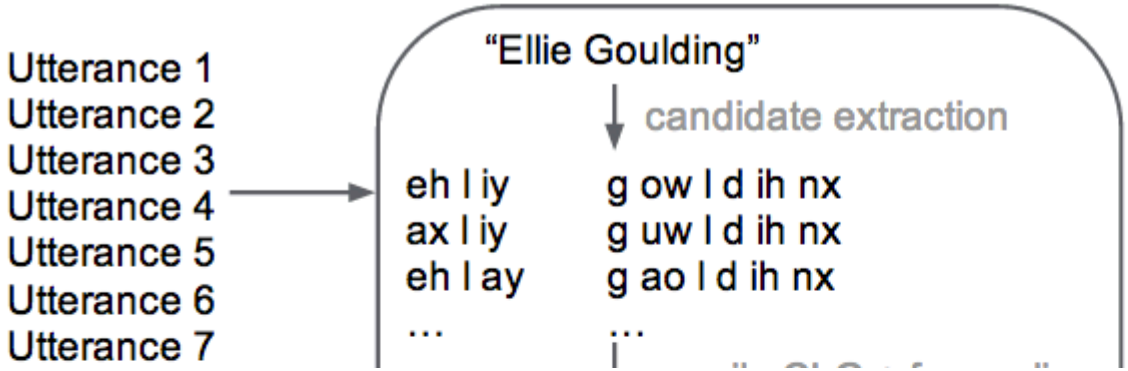


Data Acquisition

- Picked top 1,000 downloaded entries from Google Play Store for each of the four categories
  - Artist names
  - Song titles
  - TV show names
  - Movie titles
- Made 4 data sets
- Send them to Amazon Mechanical Turk
  - 10 different Turkers pronounce the same transcription
  - 7 utterances for pronunciation learning
  - 3 utterances for testing

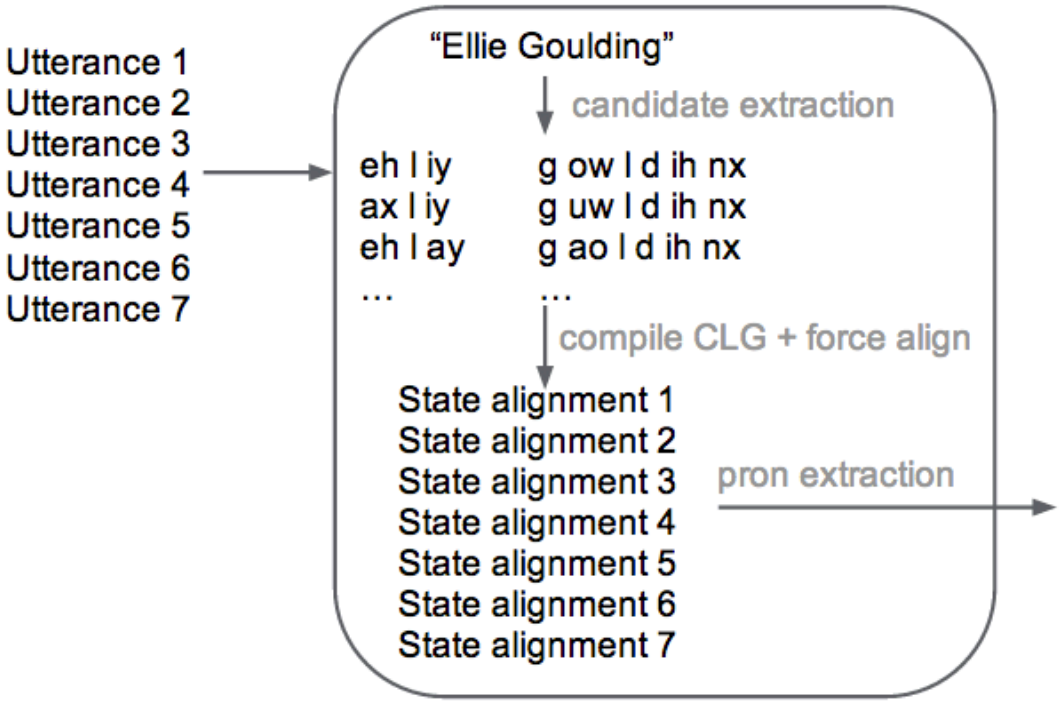
Pronunciation candidate generation

- Use Grapheme-to-Phoneme conversion to generate 20 pronunciations per word
- If a word is already in the dictionary, use that pronunciation too.



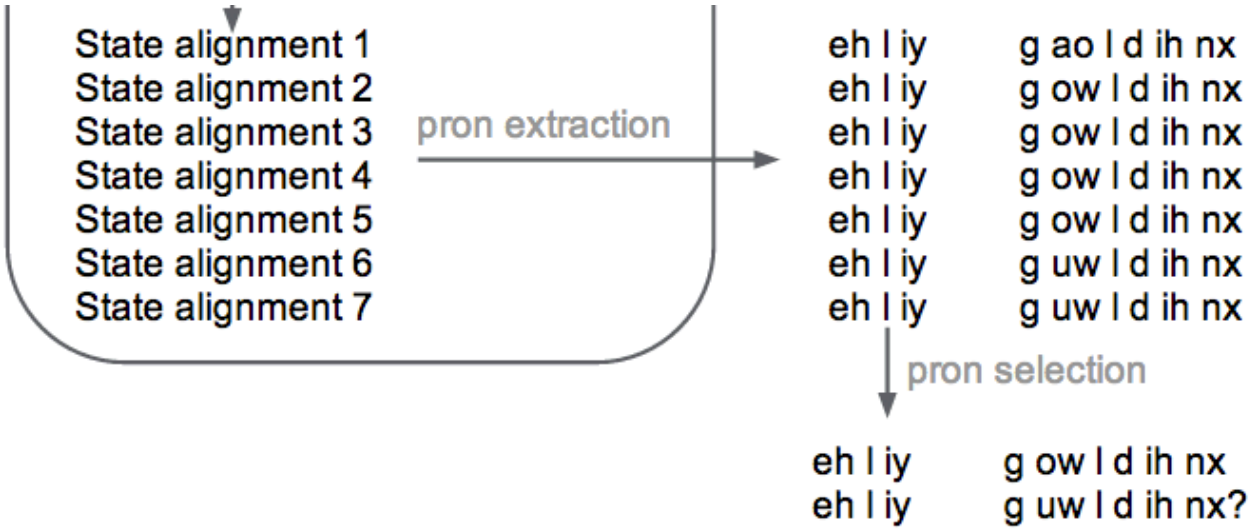
Extract pronunciation

- Force-aligned G2P: find the triphone state sequences that best align with the utterance

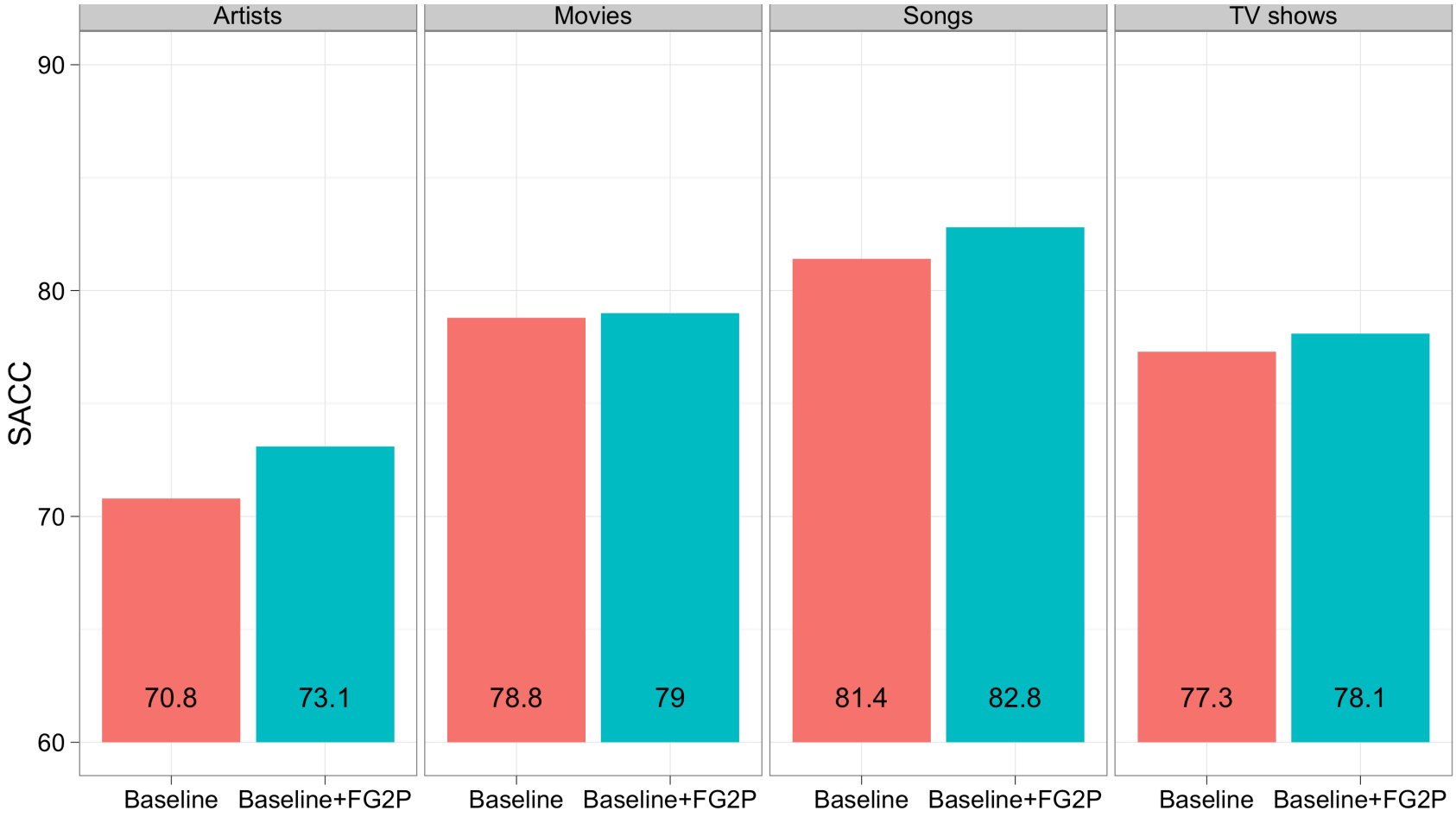


Pronunciation selection

- Evaluated 3 possibilities
  - Use all of the learned pronunciations
  - Use the pronunciations that occur more than once
  - Majority vote



Experiment results:
Use all learned pronunciations



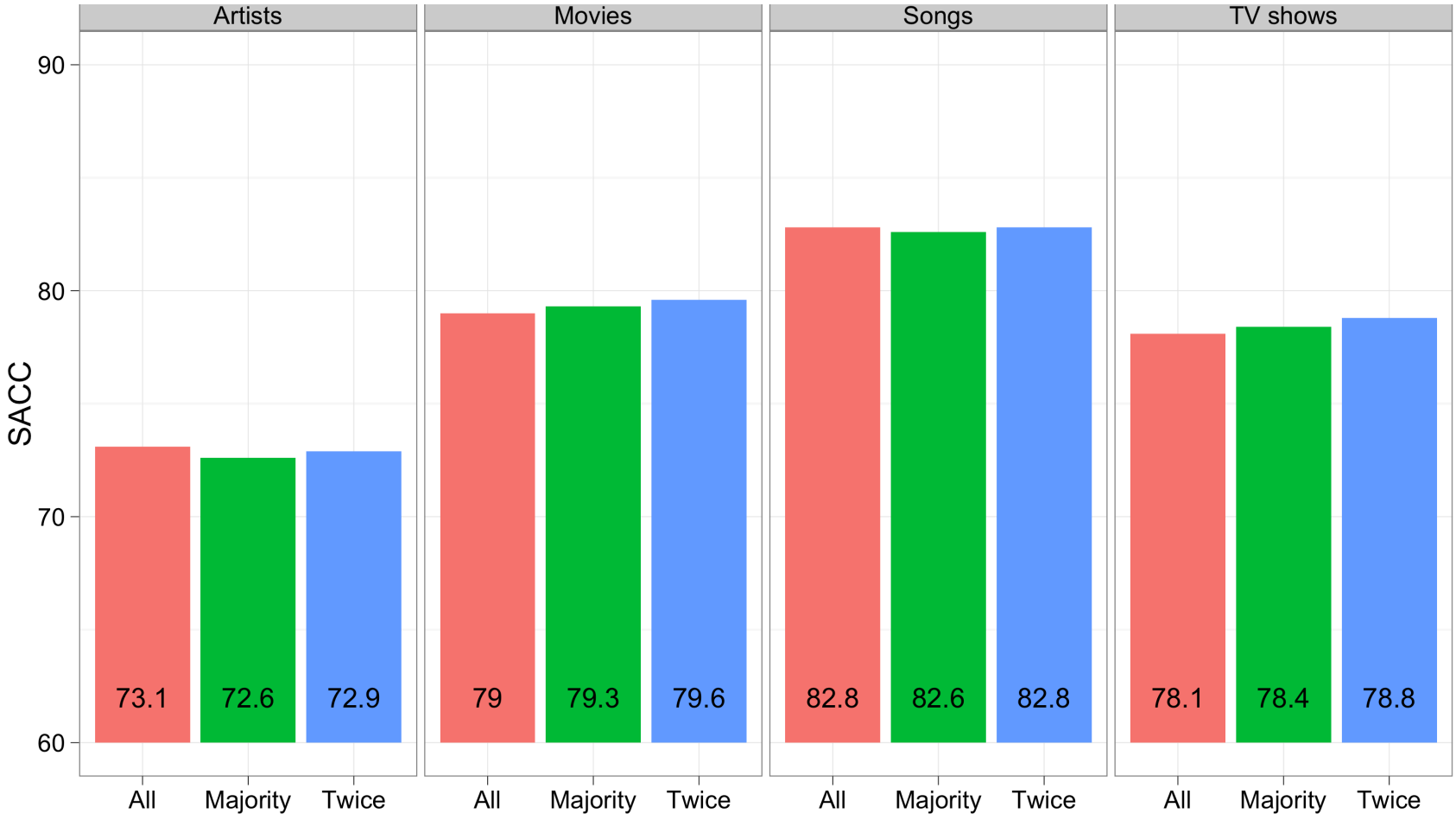
Conclusion so far

- Improved SACC across all datasets
- Highest impact on artist names

Question

- How should we select pronunciations?
- How many should we keep?

How many pronunciations should we use?



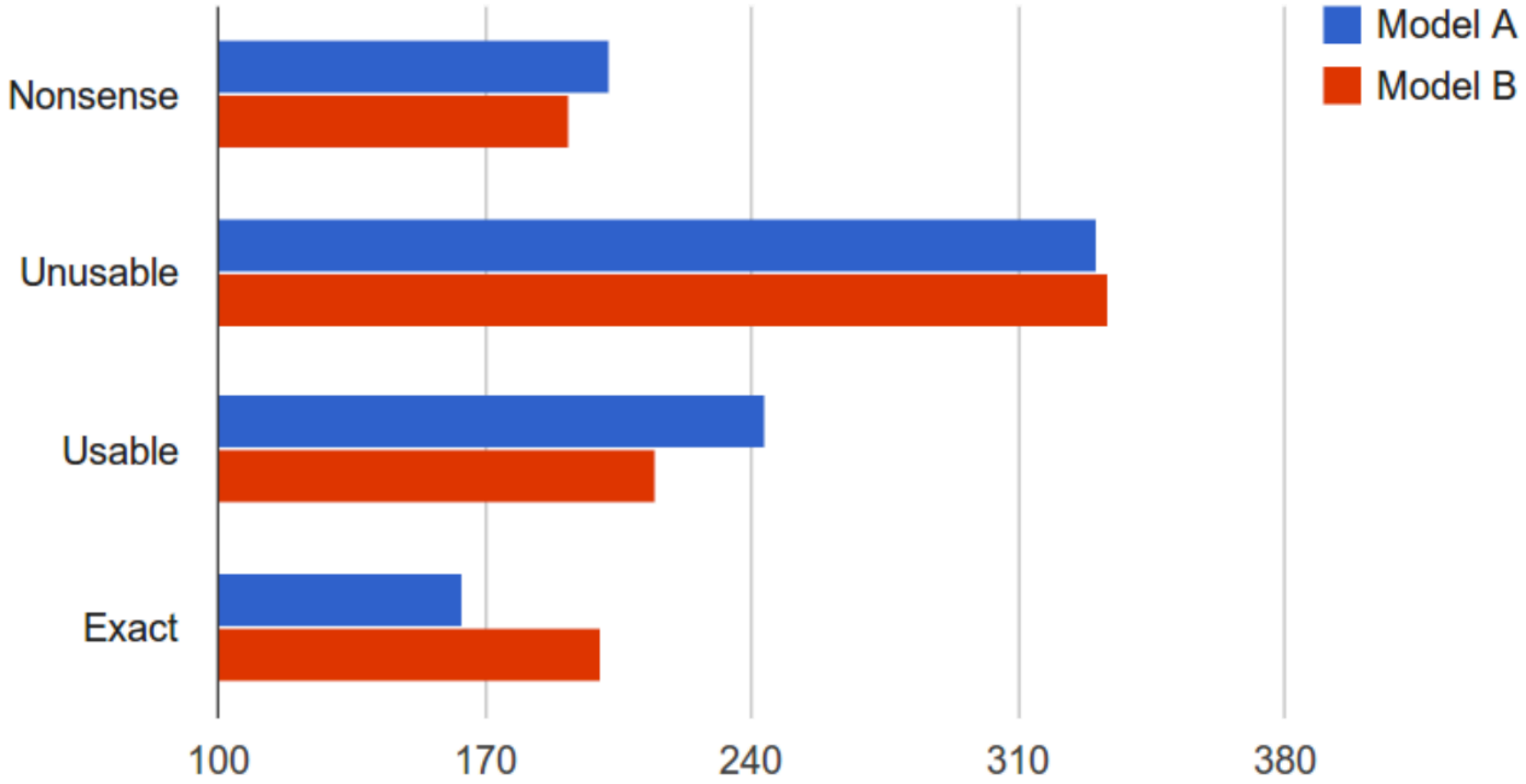
Question

- So far we tested on matching data only
  - Why can't we test it on the standard test set?
- Would it help to use the new pronunciations in the real setting?

Beyond WER

- Use majority vote to select pronunciations to add the base pronunciation dictionary
- Test on the voice search task
  - MTurkers rate the quality of the voice search result.

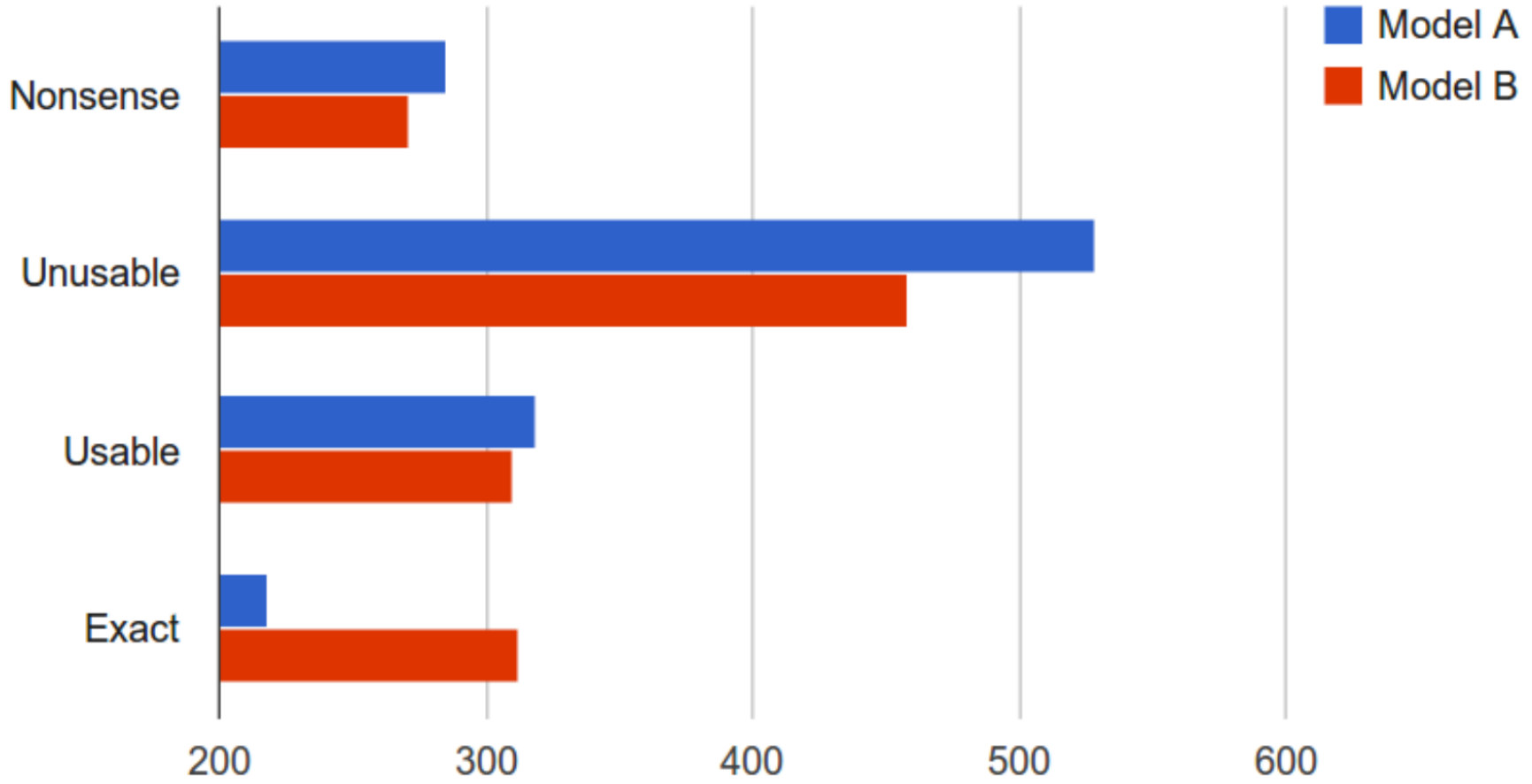
Results – Voice Search



Testing out the pipeline

- Extract 14,000 Google Map typed search queries that occur rarely in the log.
- Pick 5,000 words that are not in the lexicon.
- Can we do better than fully automatic G2P?

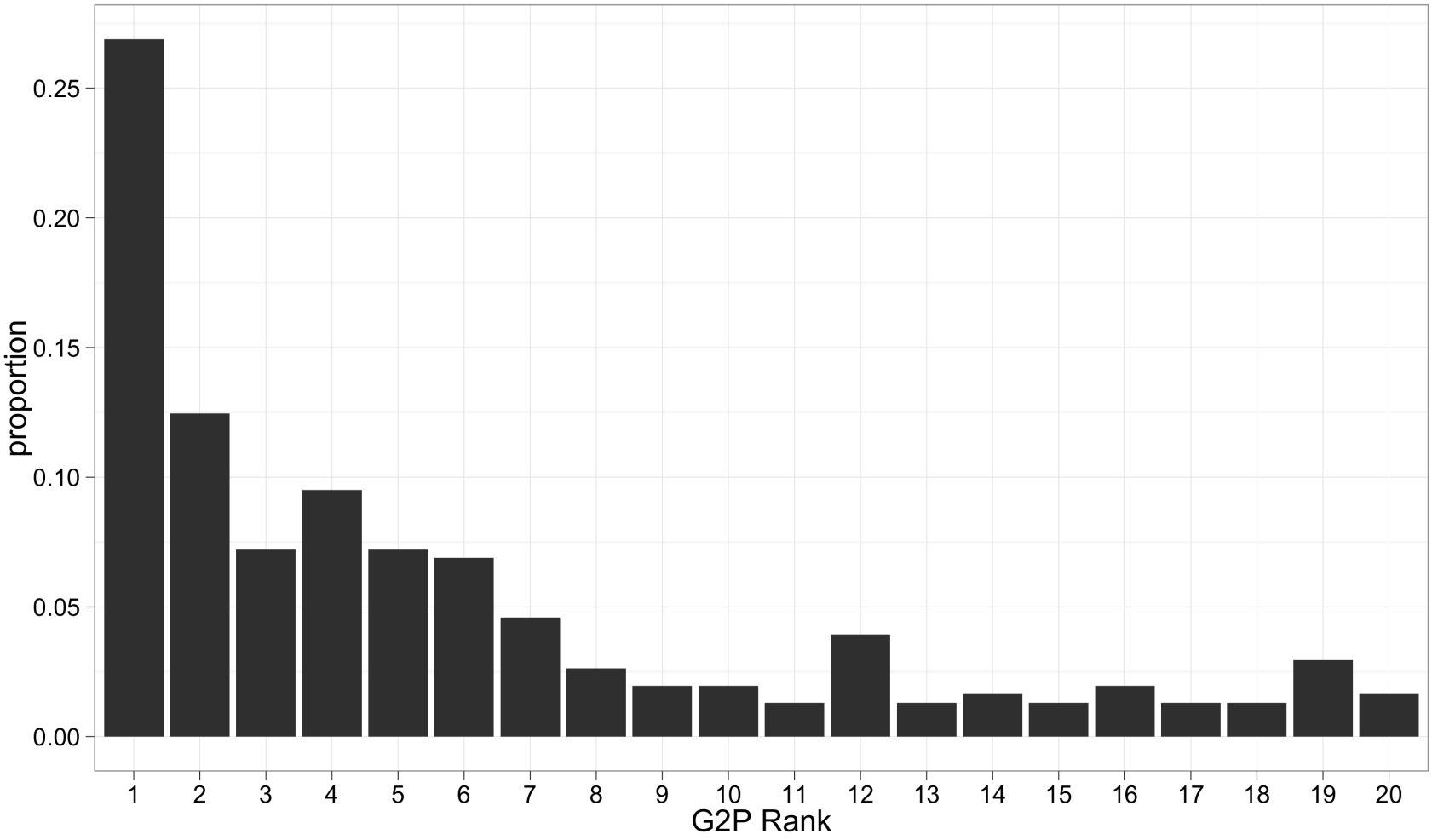
Results – Voice Google Map Search



Question

- Where do the learned pronunciations come from with respect to G2P rankings?
  - Our baseline system uses the top G2P candidate.

Distribution of G2P Ranks



Learned pronunciations from artist data

Artist name	Learned pronunciations	G2P Rank
Dan Omelio	AX M EH L IY OW	20
Tyga	T AY G ER	20
Mat Kearney	K EH EH R N IY	20
Jadakiss	JH EY D AX K IH S	20
Nasri Atweh	N AA Z R IY	19
Sonny Uwaezuoke	Y UW W EY Z UW OW K	15
Flo Rida	R AY D AX	2
Amadeus Mozart	AX M AX D EY AX S	4
David Guetta	G W EH T AX	6

Limitations

- Word boundary
  - Call Me Fitz → F IH T
  - Gwen Stefani → G W EH N Z + S T EY F AX N IY
  - Need to enforce boundary constraints

Conclusion

- Crowdsourcing pronunciations proved a viable quick path to refresh the pronunciation dictionary.
- The forcealigned-G2P algorithm can be used to learn pronunciation from audio data.
- Pronunciation in English is hard because it's such an international language.