Adam's Final Lecture

- Thanks for making this a fun experience!
  - Students!
  - TAs!!
  - Josh!!!
- I will probably teach EECS 255d (Audio Signal Processing in Humans and Machines) next fall.

Announcements

- Lecture on Robotics: Tue Nov 29, 9:30-11
- Homework 10: Due Wed Nov 30, 11:59pm
- Final Contest: Due Fri Dec 9, 11:59pm
  - Daily rankings starting Sat Dec 3
- Project 6: Due Mon Dec 5, 5:00pm
- Final Exam: Tue Dec 13, 3-6pm

Speech Recognition
A CS188 Perspective

Adam Janin


What we say to dogs
Okay, Ginger! I've had it! You stay out of the garbage! Understand, Ginger? Stay out of the garbage, or else!
What they hear
blah blah Siri blah blah blah blah blah blah Siri blah blah blah blah blah...

Why Is It Hard?

- Many different ways to say the same thing.
  - Volume
  - Accent
  - Gender
  - Speaking style
  - Speed
  - Etc.

Why Is It Hard?

- Word Boundaries?

Why Is It Hard?

- Homophony – The same sounds may mean different things.

- Mai numb burr is ate won ate too fore to for
  My number is 818-2424

- wreck a nice beach
  Recognize Speech

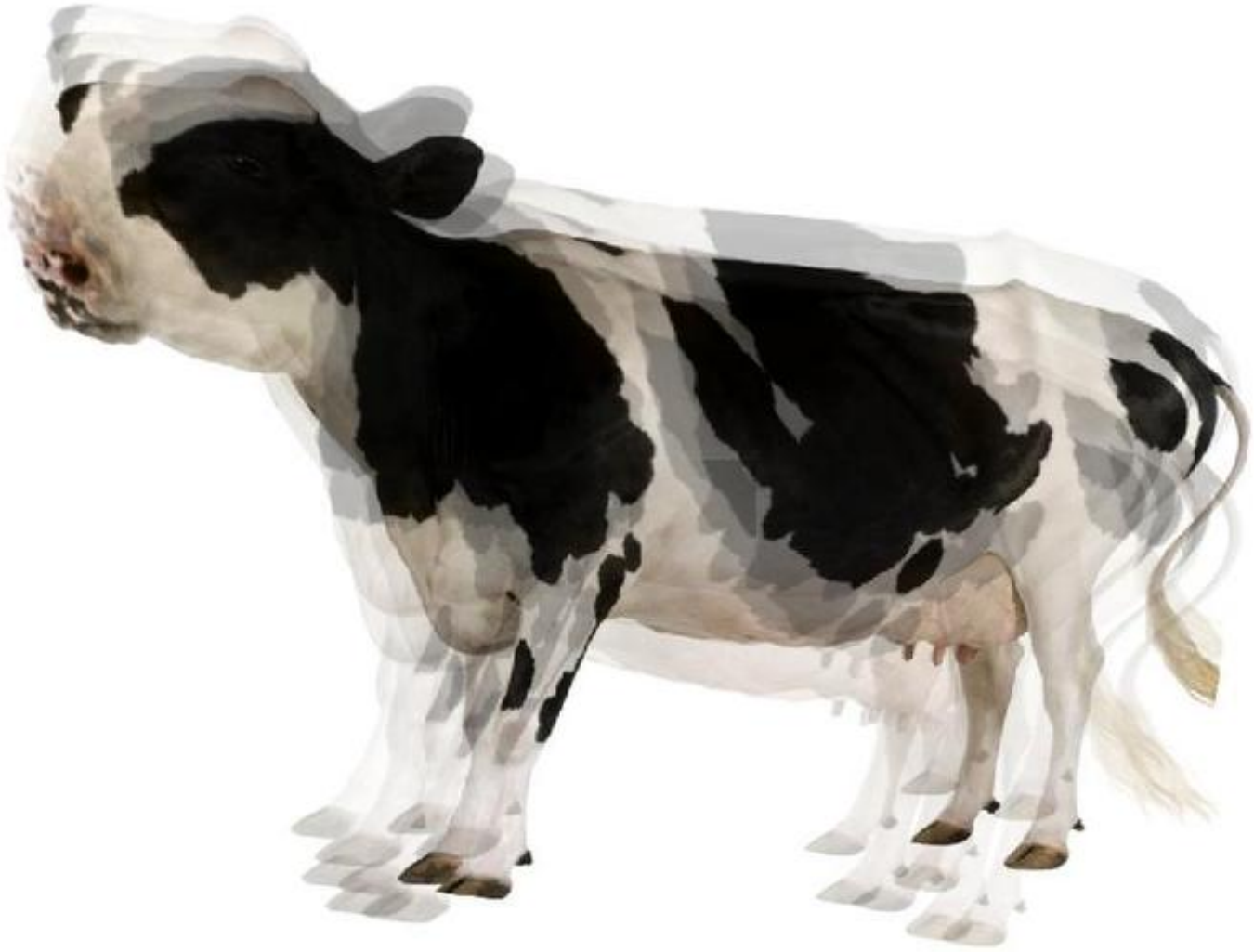
Why Is It Hard?

- Background noise
- Reverberation
- Channel effects

Clean Audio, One Speaker, One Word, …

Channel Effects

Reverberation / Echos



Background Noise

Many Background Noises!

A Little History

First Consumer Electronic “Speech Recognition”?

Radio Rex – 1922

Rex would leap out of his house when you called his name.

No computer!

Electromagnet tuned to 500 Hz, just around where the “eh” sound in “Rex” is strongest.

ASR Grew Up With AI

- Expert Systems
  - "Every time I fire a linguist, the performance of the speech recognizer goes up" – Fred Jelinek, IBM, early 80s.
- Nearest Neighbor
  - Template matching
  - DTW (Dynamic Time Warp)
- HMM / GMM
- Neural Networks

- Single Words
  - Small vocabulary
  - Highly controlled acoustics
- Short Phrases
  - Clean acoustics
- Continuous Speech
  - Increasing vocabulary size
  - Increasing grammar complexity
  - Increasingly difficult acoustics

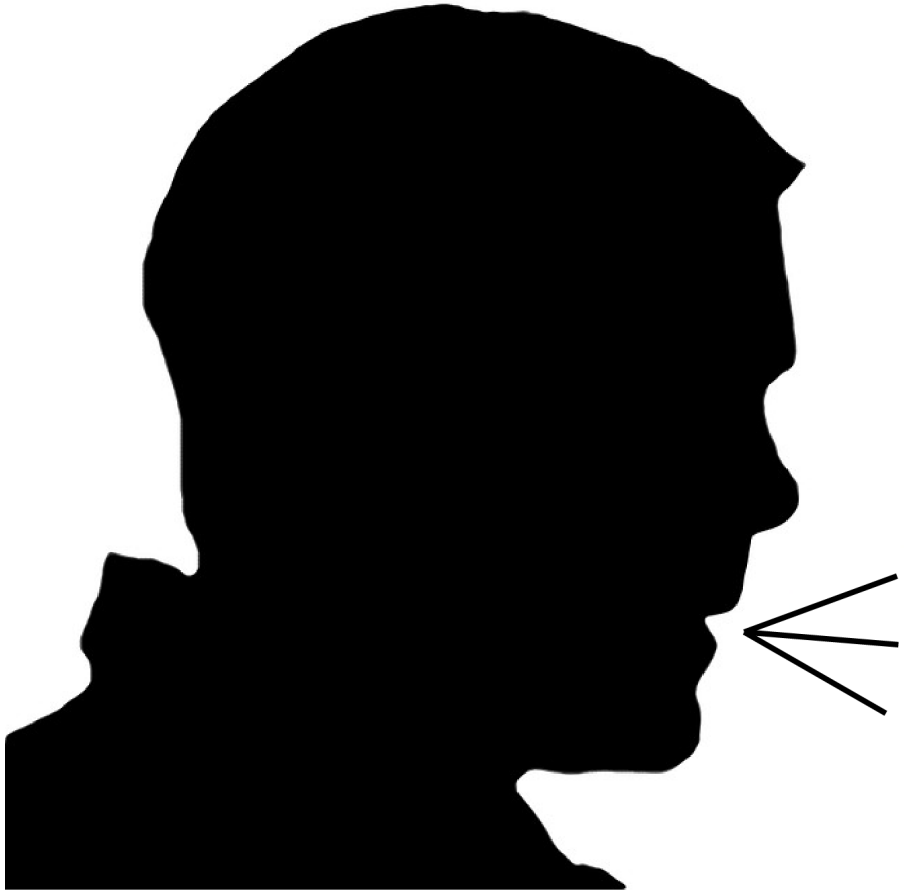
What I Wont Talk About (Much)

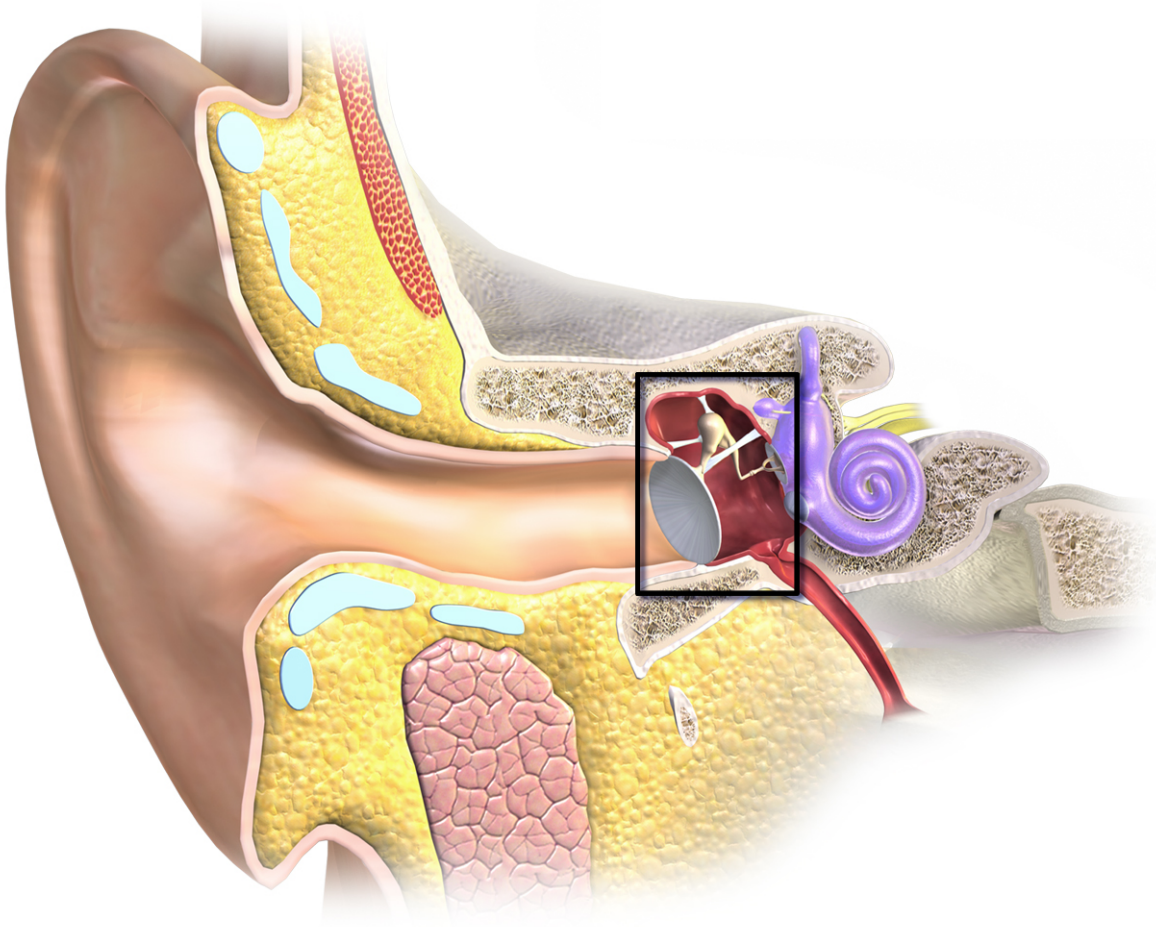
- Connectionist Temporal Classification (CTC)
  - One big recurrent neural net.
  - Computationally expensive; currently not as accurate.
- Language Modeling
- Uses of ASR
  - What do you optimize for?
- Details, Complications, Edge Cases

What I Will Talk About

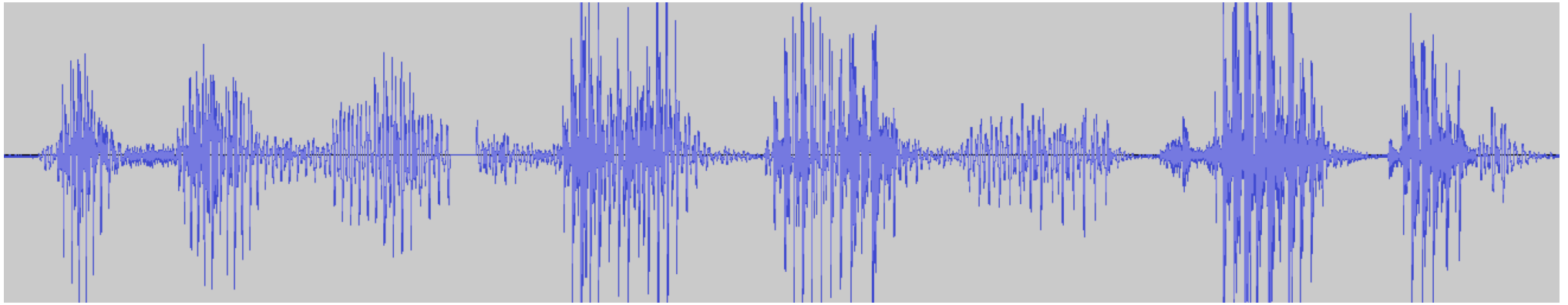
- Current Publicly Available State of the Art Systems
  - Kaldi, HTK
- Neural Networks for Acoustic Modeling
  - Kaldi nnet/nnet2/nnet3, Keras, Tensorflow, Theano, many others.
- Weighted Finite State Transducers (WFSTs) for Search
  - OpenFST

What Is Audio

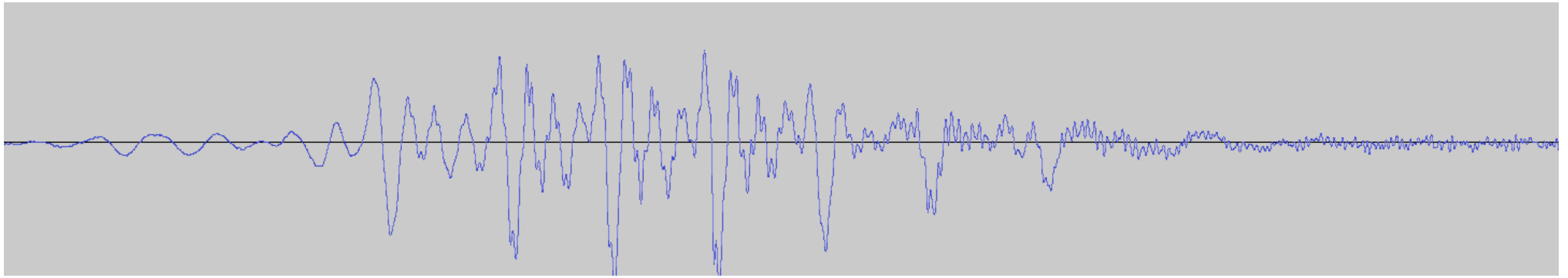




What Is Audio?

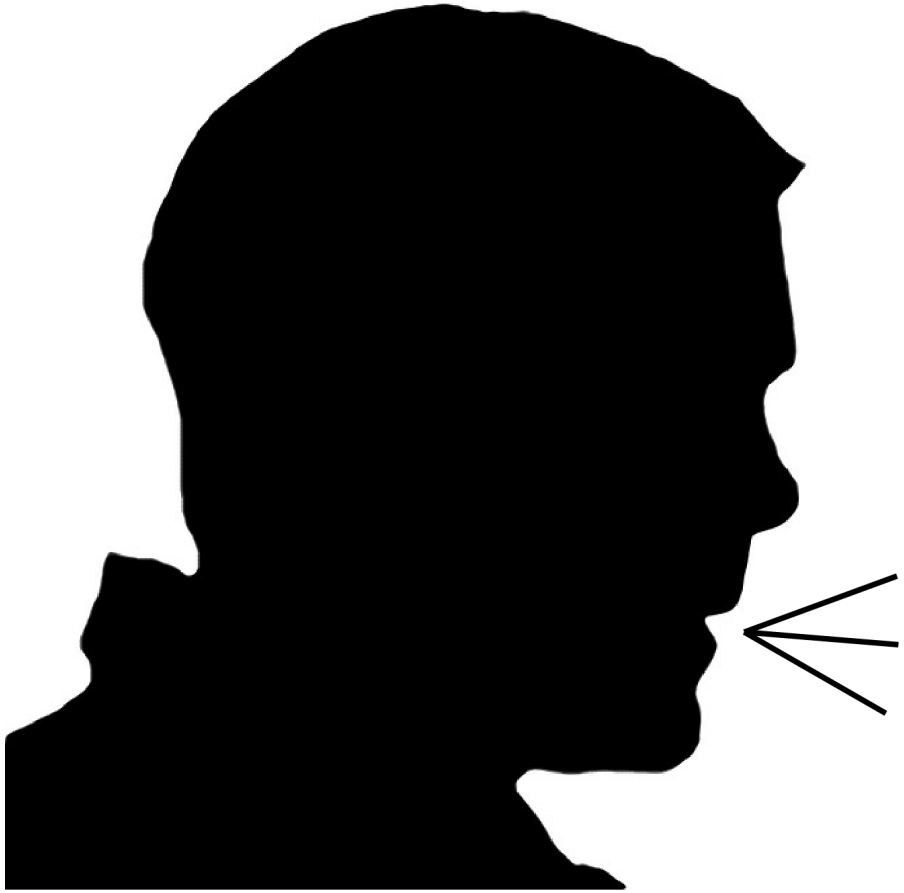


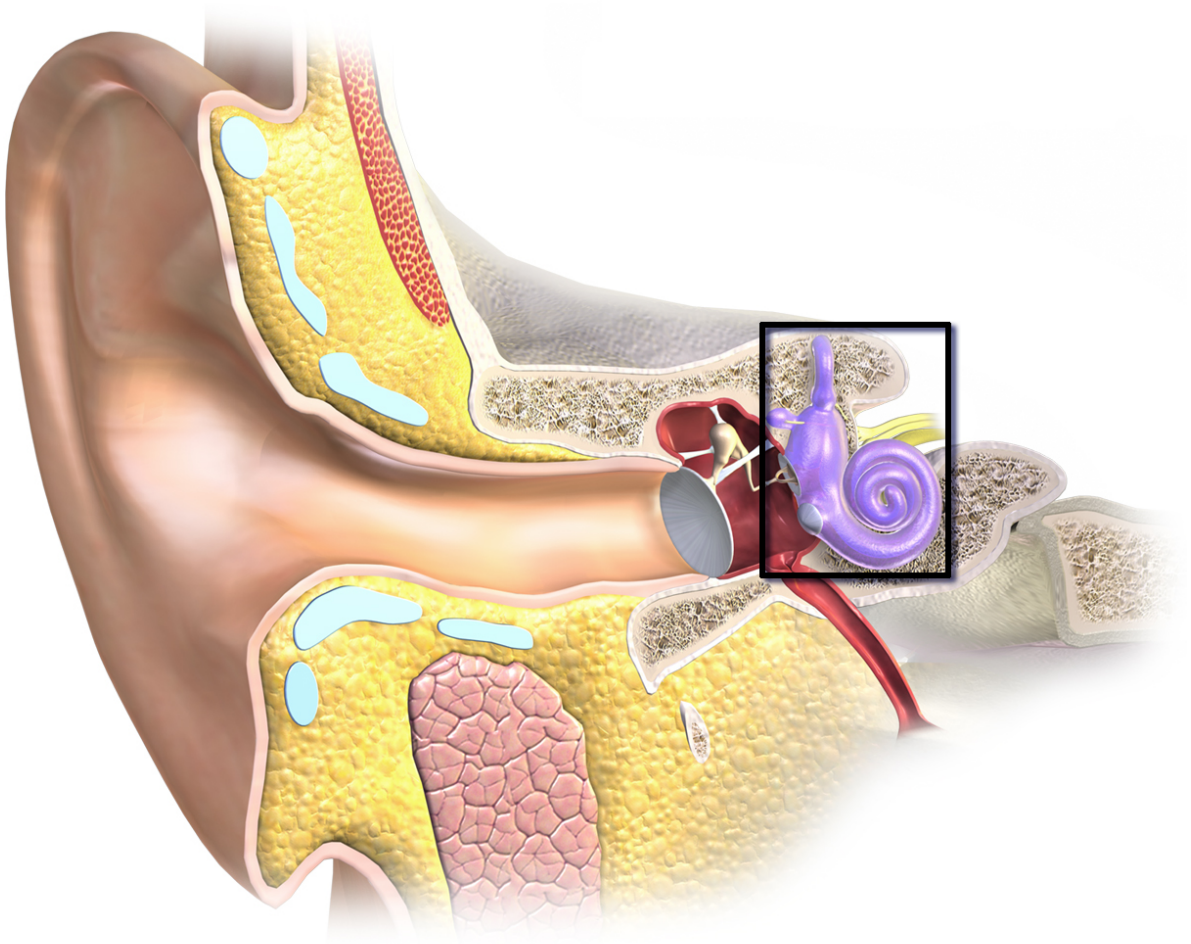
“This is an example of me talking”



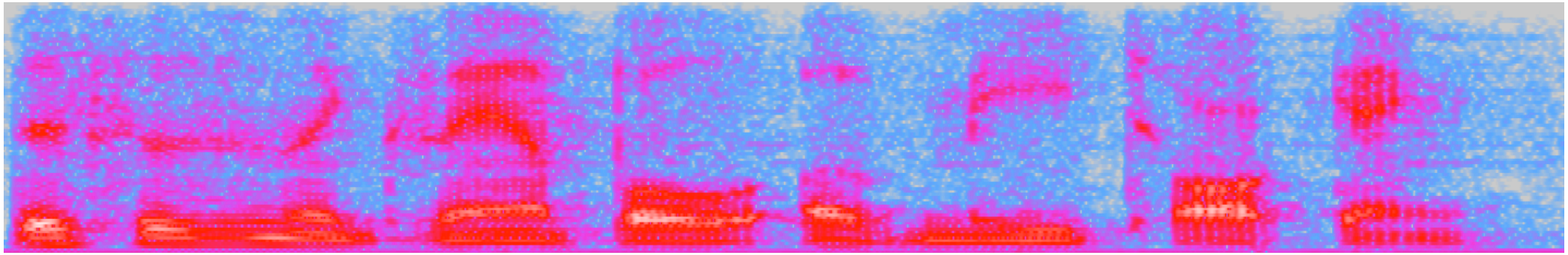
“This”

Spectrograms





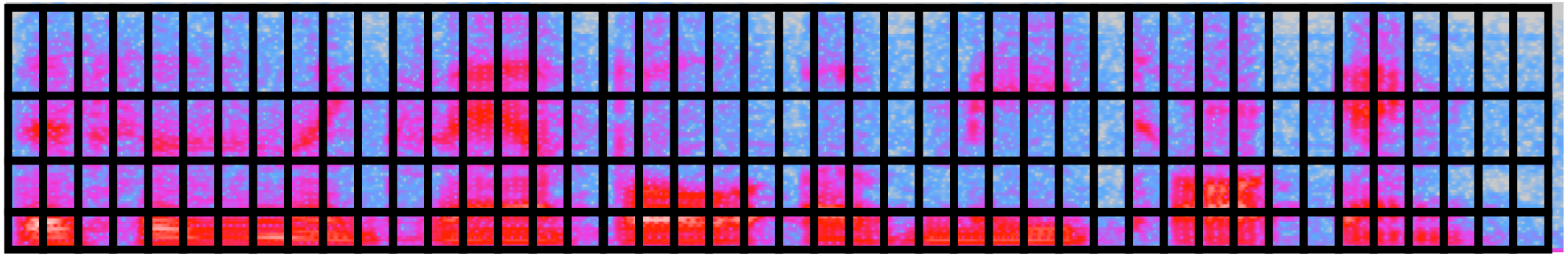
Audio Spectrogram



“This is an example of me talking”

- Time on horizontal axis.
- Frequency (tone) on vertical axis.
- Color is amount of energy in that frequency range.

Audio Features



“This is an example of me talking”

- A feature is the total energy in a region of the spectrogram.
- Humans are less sensitive to higher frequencies.
- Usually uniform in time (100/second = 10ms)
  - Known as a *frame*.
- More like what the ear sends to higher regions in the brain.

Start Simple: Single Word Recognition

Training data are audio files, each with a single word in it. Is the word "*zero*" or "*one*"?

- What algorithm that you already know could you use to solve this if you had a huge number of examples and lots of compute? Hint: It would be naïve to use this algorithm.
- Why would this be terrible if you did not have a huge number of examples?

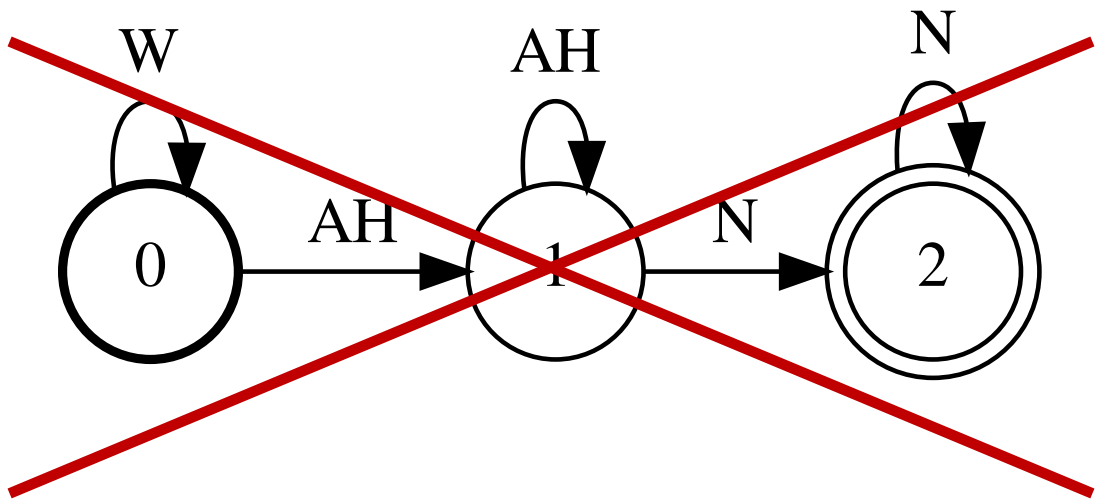
  Hint: How many features are there in each audio file?

Solution: Model Subparts of the Word

ZERO	Z	IH	R	OW	
ONE	W	AH	N		
TWO	T	UW			
THREE	TH	R	IY		
FOUR	F	AO	R		
FIVE	F	AY	V		
SIX	S	IH	K	S	
SEVEN	S	EH	V	AH	N
EIGHT	EY	T			
NINE	N	AY	N		

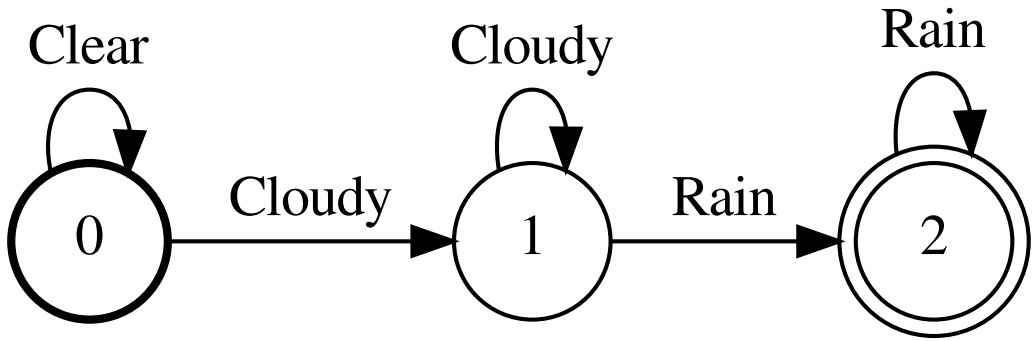
Pronunciation Dictionary (e.g. CMUDICT)

Solution: Model Subparts of the Word



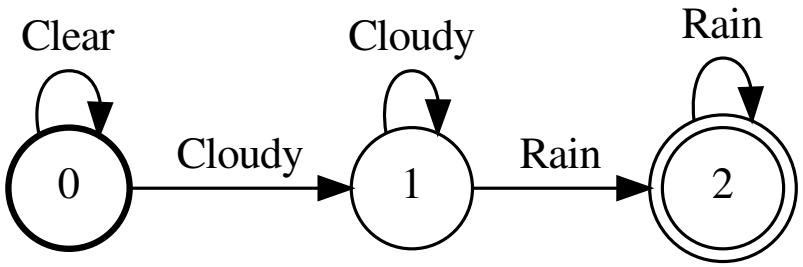
Too Scary!

Less Scary: Weather

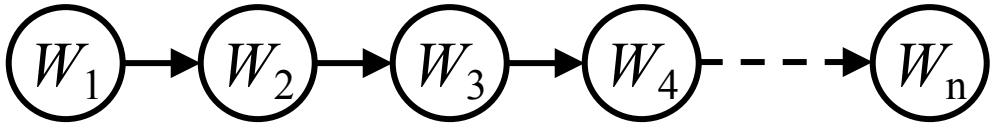


Clear for a few days, then cloudy for a few days, then rain for a few days.

Weather Markov Chain

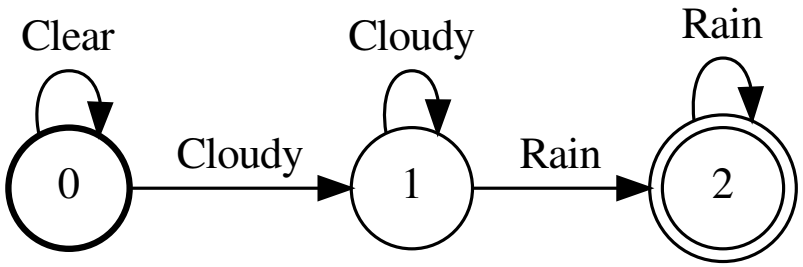


Transition Graph

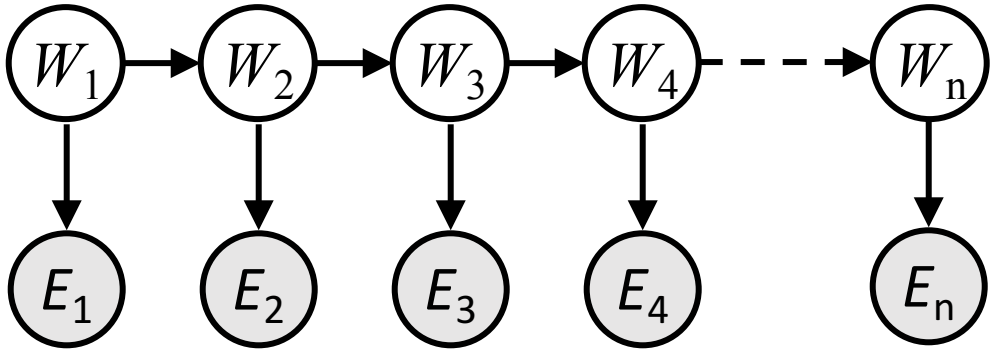


Bayes' Net (Markov Chain)
n Days of Weather

Weather Hidden Markov Model

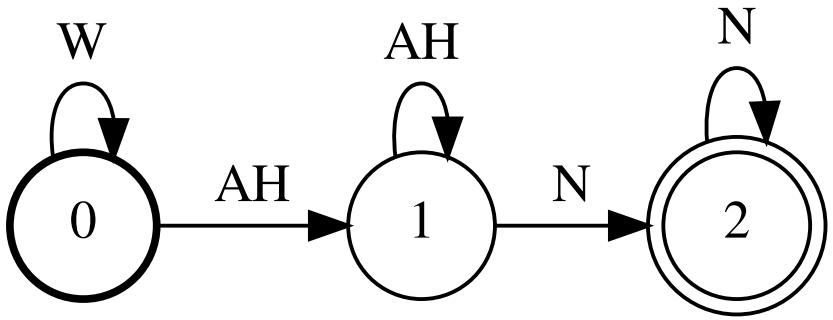


Transition Graph

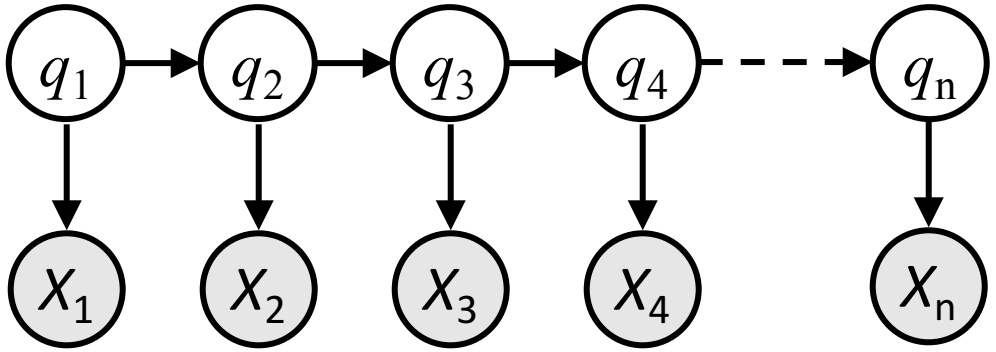


Bayes' Net (HMM)
n Days of Weather

“One” Hidden Markov Model

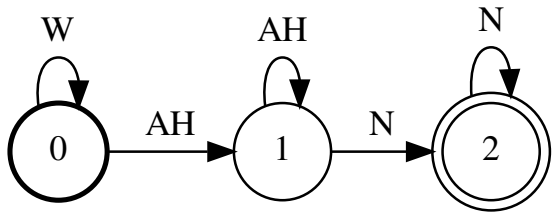


Transition Graph

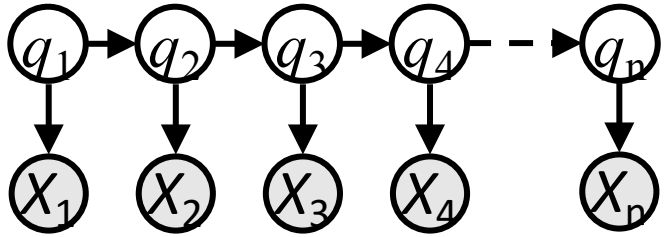


Bayes’ Net (HMM)
n Frames of Acoustics

If I gave you all the CPTs, what algorithm would you use to compute the probability of "*one*" given the acoustic features? E.g. P(q1, q2, … qn | X1, X2, ... Xn).

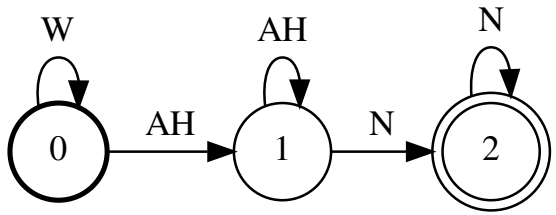


Transition Graph

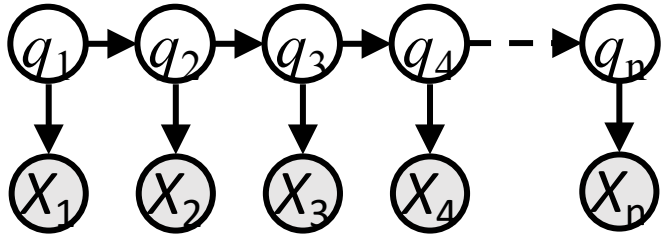


Bayes' Net (HMM)
n Frames of Acoustics

If I gave you all the CPTs, what algorithm would you use to compute how long the person said “ah” assuming they did say “one”?

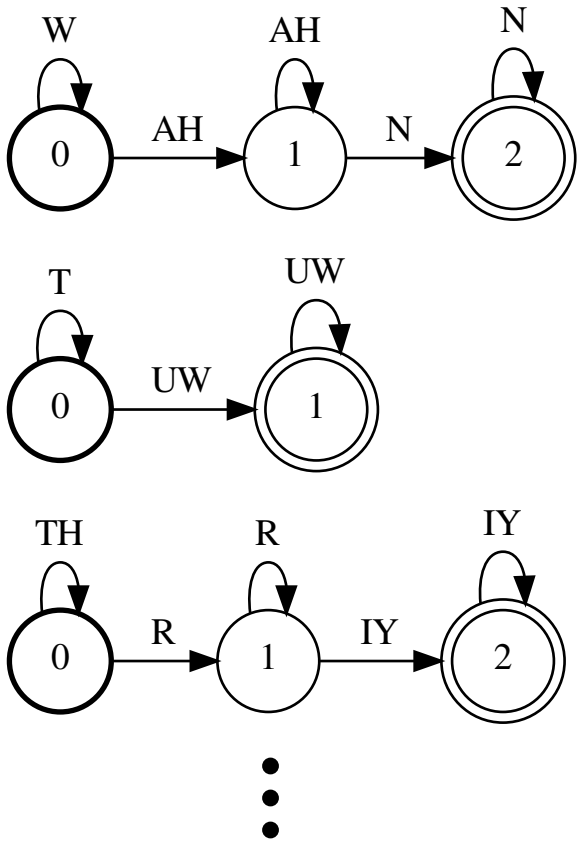


Transition Graph



Bayes’ Net (HMM)
n Frames of Acoustics

Isolated Word Recognizer

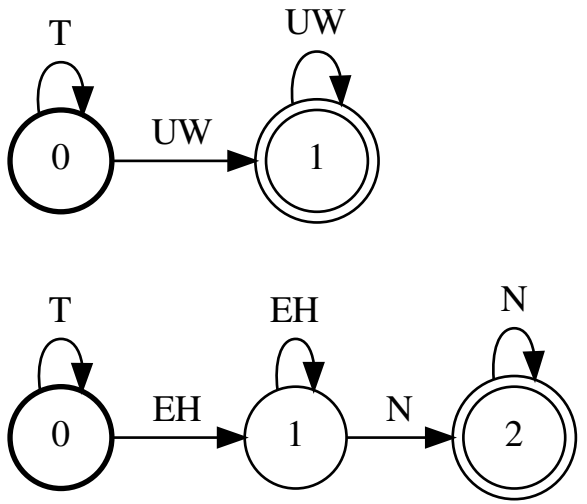


- Run an HMM for each word.
- Pick the one with the highest probability!

- Problems:
  - Scalability – 100,000 word vocabulary
  - Repeated work – “t” in ”two” vs. “t” in “ten”.
  - Sentences have more than one word!
  - Haven’t told you how to compute CPTs.

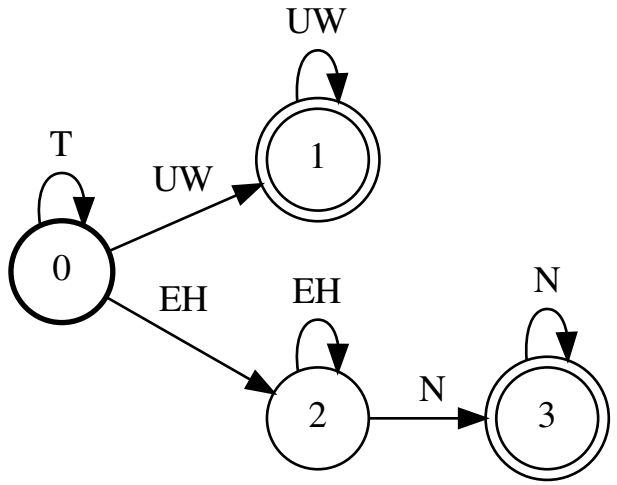
Avoiding Repeated Work

- Build a graph for each word.
- Combine where possible.
- Lots of theory behind this.
  - Weighted Finite State Transducers
- The toolkit OpenFST will do it all for you.



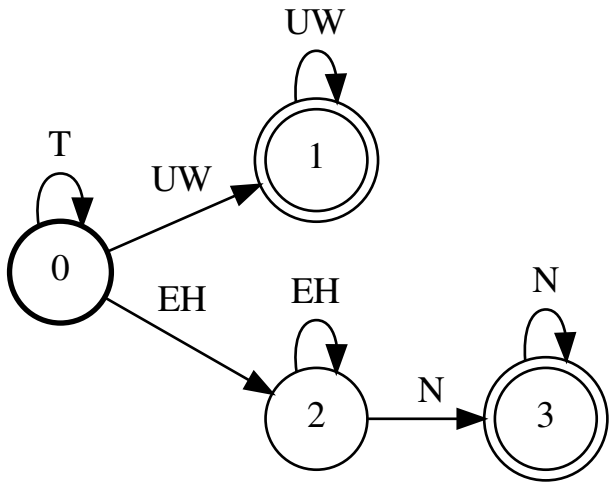
Avoiding Repeated Work

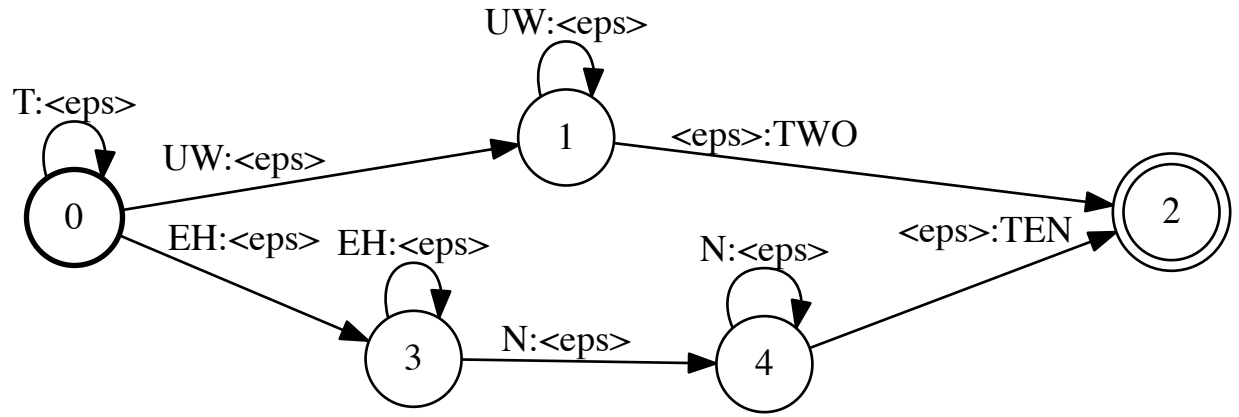
- Build a graph for each word.
- Combine where possible.
- Lots of theory behind this.
  - Weighted Finite State Transducers
- The toolkit OpenFST will do it all for you.



Avoiding Repeated Work

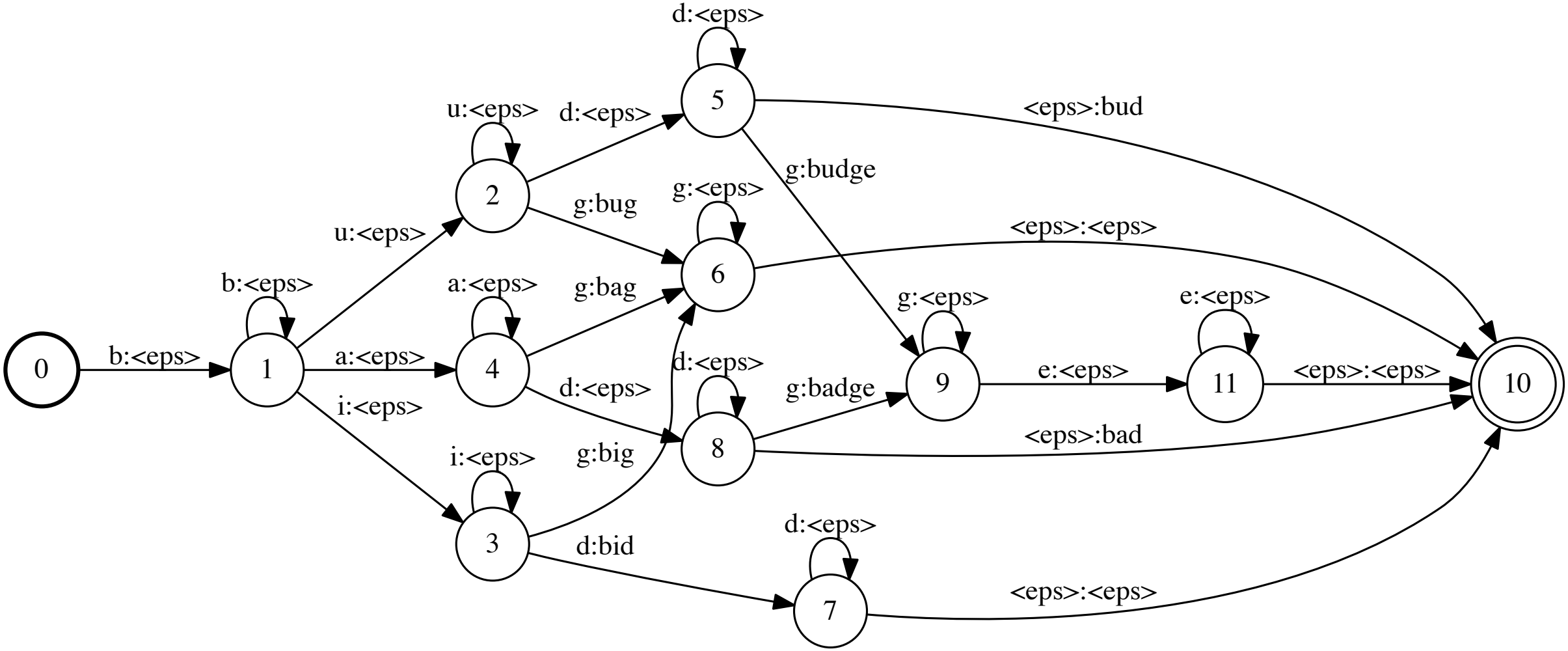
- New Notation on Arcs
  - x:y – When you traverse the arc, consume “x” and emit “y”.
  - <eps> - Epsilon. On input, do not consume any input. On output, do not emit any output.
  - For any word/pronunciation, notice that all input is consumed and one word is output.





Avoiding Repeated Work

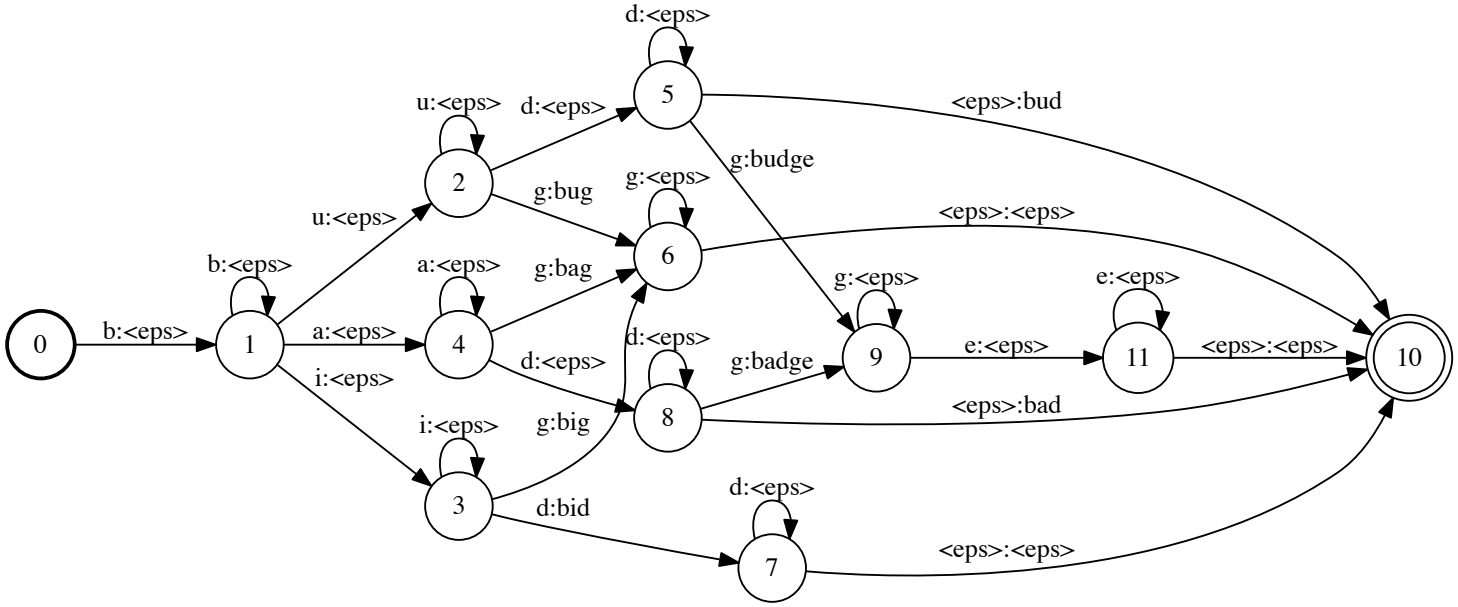
- Next slide has a big example:
  - **bad, badge, bag, bid, big, bud, budge, bug**
- The example uses letters rather than phonemes to make it easier to read.
- The process to generate the graph is fully automated once you have pronunciations.
- The data structure is known as a “decoding graph”.



bad, badge, bag, bid, big, bud, budge, bug

Inference on Decoding Graph

- Summing over all paths in the search tree would yield the probability that the file contained any of *bad*, *badge*, etc.
- How do we tell which word was actually there?
- Cast your mind back to the very beginning of the term…

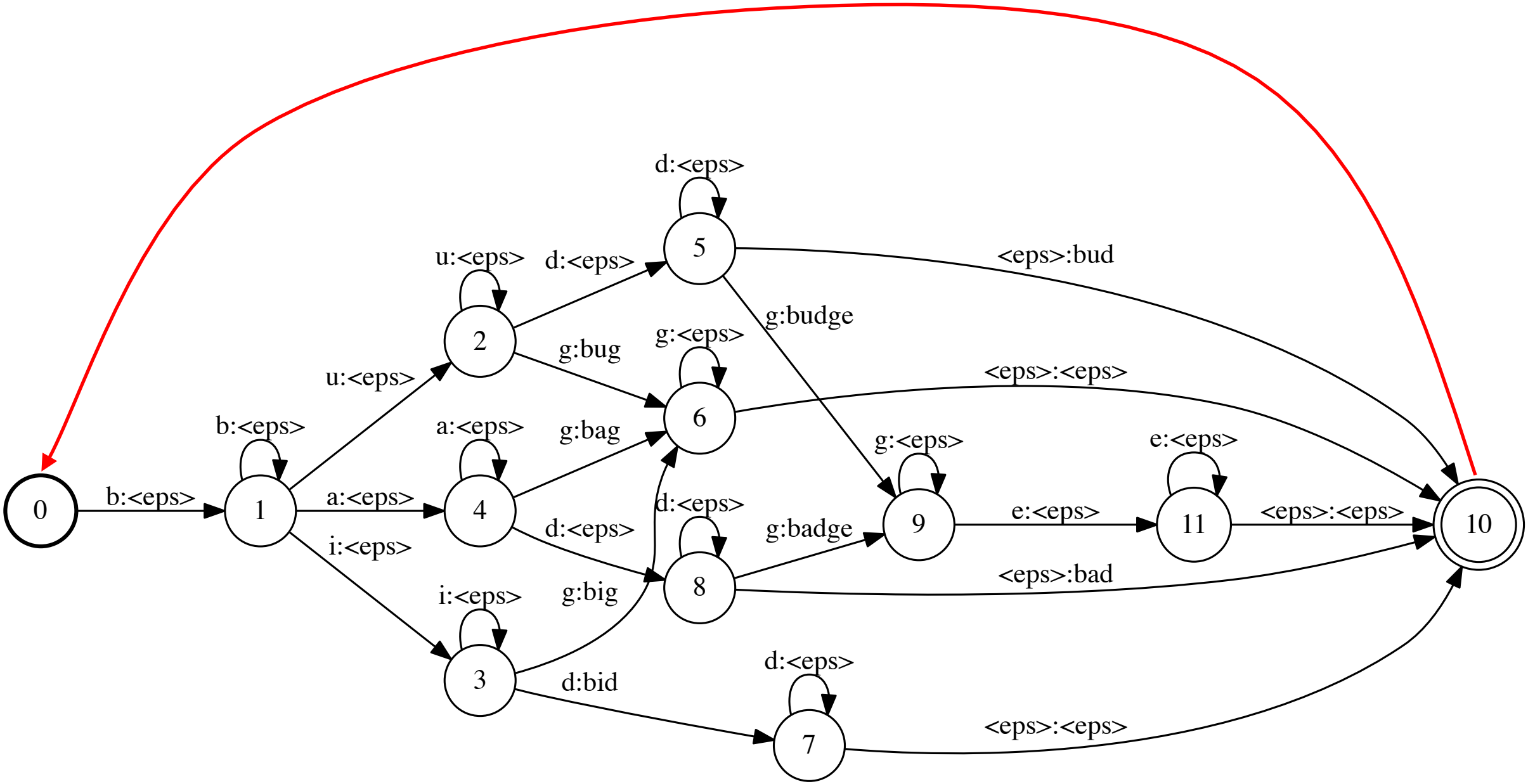


Inference on Decoding Graph

- Typically we do not use A* because:
  - Hard to find a heuristic.
  - We want not just the best path, but the scores of the best few words.
- Believe it or not, we use the much-maligned BFS!
- BFS made tractable using “pruning”.
  - Throw out members of the fringe that are not close to the best score on the fridge. Do this at each step.
  - Not optimal or complete, but works well in practice.

Continuous Speech Recognition

- What if the file contains more than one word?
- Example: “big bad badge bug bud”
- Very complex, so pay close attention!


0
1
2
3
4
5
6
7
8
9
10
11
b:<eps>
b:<eps>
u:<eps>
u:<eps>
d:<eps>
d:<eps>
a:<eps>
a:<eps>
g:bug
g:bag
g:budge
g:<eps>
g:<eps>
<eps>:bud
<eps>:<eps>
i:<eps>
i:<eps>
d:<eps>
d:<eps>
g:big
g:badge
e:<eps>
e:<eps>
<eps>:<eps>
<eps>:bad
d:bid
d:<eps>
<eps>:<eps>

Continuous Speech Recognition

- Okay, so that was a bit glib…
- In the real case, not all sentences have equal probability.
- This is encoded in the *language model*.
- See Midterm 2 question 2 (Pacmanian Language Modeling) for how we compute the language model.
- Can be encoded in the decoding graph in a mechanical way.
- Lots of interesting theory and practice, but not enough time to cover.

Where Do We Get the CPTs?

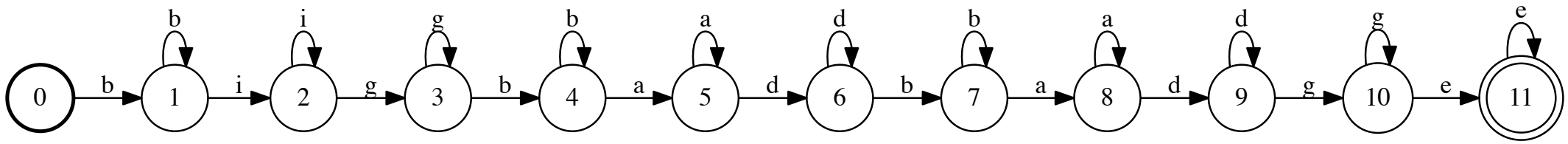
- The CPTs encode $P(q_t|X_t)$ the probability of each phoneme at time t given the acoustic features at time t.
  - e.g. At time t=1, $P(q_1 = ah|X_1) = 0.31, P(q_1 = r|X_1) = 0.01$, etc.
- Any machine learning algorithm can be used.
  - Historically, GMMs were used.
  - Deep neural nets are currently favored.
- Training data
  - Acoustic features every 10 ms.
  - Which phoneme it was every 10ms.

Training

- Computing features is usually easy.
- Where do labels come from?
  - Some corpora label every 10ms chunk of time with a phoneme. *Extremely* labor intensive, and only a tiny amount of data available.
  - Most corpora label with the sequence of words in each audio file.
- Going from the sequence of words to the sequence of phonemes is known as *forced alignment*.

Forced Alignment

- Construct a decoding graph with just the sequence of words that was actually spoken.
- This results in a *much* smaller search graph.
- Run the normal algorithm.
- Example: “big bad badge”



Bootstrapping

- How can we “run the normal algorithm” if we haven’t already trained it?!?

- Bootstrapping.
  - Start with another system.
  - “Flat Start”: Train a system where you assume each phoneme is of equal length. This is a terrible assumption, but you can then rerun a few times and it tends to converge.

Summary

- Acoustic features are binned energies from spectrograms (usually with additional processing).
- Words are decomposed into phonemes using a pronunciation dictionary.
- An HMM is formed from the phonemes with features as evidence.
- The CPTs come from deep neural networks.
- A language model weights likely sentences higher than unlikely ones.
- A decode graph represents all possible sequences of all possible pronunciations of all possible words.
- A search through the decode graph gives you the most likely sequence of words.

Further Reading

- Nuts and Bolts
  - Speech and Language Processing. Jurafsky and Martin.
- Broad Topics on Audio
  - Speech and Audio Signal Processing: Processing and Perception of Speech and Music. Gold, Morgan, and Ellis.
- History
  - The Voice in the Machine: Building Computers That Understand Speech. Roberto Pieraccini

Complications

- I glossed over how language models are integrated into the decoding graph. There is lots of interesting theory and practice on it.
- Language models always use *backoffs* – if there aren't enough examples in the corpus of three words in a row, you combine the two sequences of two words in a row to get a smooth estimate of the trigram.
- Smoothing of language models is very important in accounting for words you haven't seen.

Complications

- The way one pronounces e.g. “eh” depends on the phonemes around it. We model this with *triphones*. For example, the pronunciation of “seven” is:

  $ s eh s eh v eh v ah v ah n ah n $

- There are many methods to normalize features so that they are more similar from speaker to speaker. Just making the features have mean=0.0 and variance=1.0 helps. Other methods: VTLN, MLLR, fMLRT, etc.

Complications

- There are many methods of removing noise from audio before computing features. Examples: Spectral mean subtraction, Wiener Filtering, Beam Forming, CASA, etc.

- There are many many features one can use, each with many variants and many parameters. You often need to tune the parameters on the held-out set.

Complications

- The neural networks used to compute the CPTs can be very complex. We've only touched the surface on methods used.
- Similarly, the language models can be very complex. For example, we often train on multiple language models and combine them.
- There are many techniques for handling words where you don't have the pronunciation. These are known as "out of vocabulary" or OOV words.

Complications

- State of the art systems are usually system combinations of many systems. They might have 10 different neural nets for computing CPTs, 5 language models, multiple dictionaries, etc.

- Languages other than English can have issues. For example, in Nunavut Inuktitut (= Eskimo) the word *tusaatsiarunnanngittualuujunga* means “I can't hear very well”. In languages like this, the vocabulary is too big to fit in memory. You have to break words into pieces.

Complications

- Although there aren't generally pauses between words, there *can* be. Pause handling can be tricky in the decoding graph.
- Generating more than just the best hypothesis can be difficult. There is a compact data structure known as a *lattice* that can encode the list of hypotheses.
- Often, you will generate lattices using a simpler system, then *rescore* the lattice using a more complex system.