

CS440/ECE448 Lecture 26: Speech

Mark Hasegawa-Johnson, 4/17/2019, [CC-By 3.0](#)

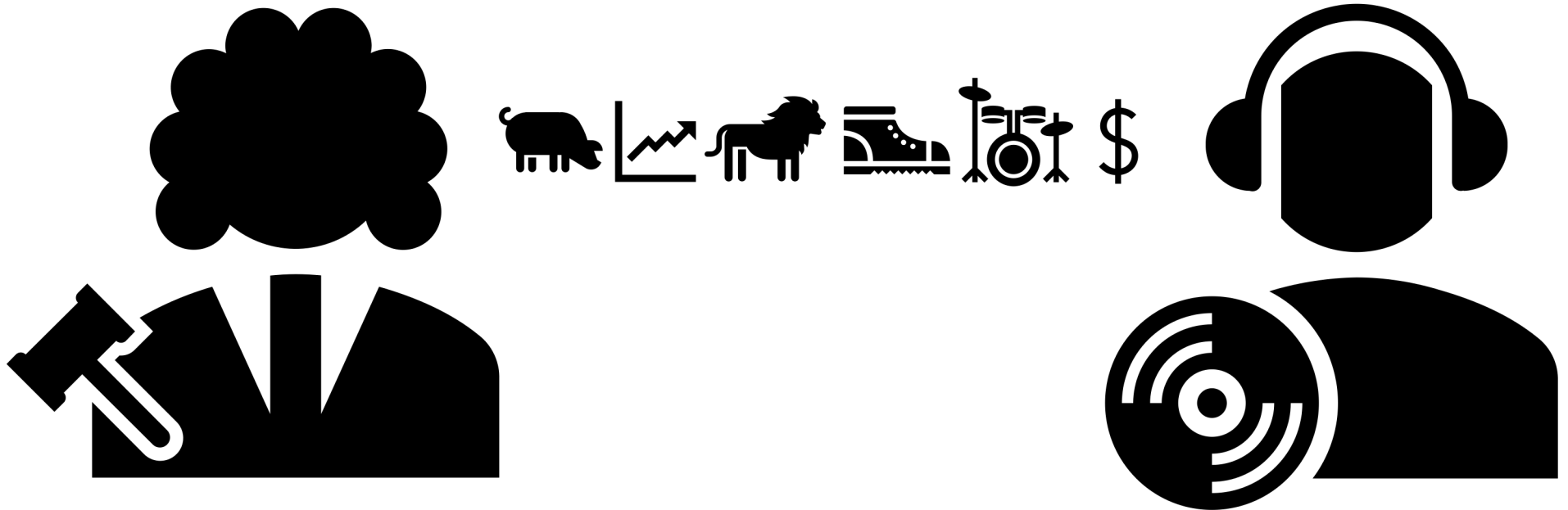
Outline

- Human speech processing
- Modeling the ear: Fourier transforms and filterbanks
- Speech-to-text-to-speech (S2T2S)
 - Hybrid DNN-HMM systems (*)
 - Recurrent neural nets (RNNs): Connectionist Temporal Classification, Trajectory nets
 - Sequence-to-sequence RNNs with attention
- Languages other than English
 - Training and testing on 300 languages
 - Transfer learning: from languages with data, to languages without
 - Dialog systems for unwritten languages
- Distorted speech
 - Motor disability
 - Second-language learners

(*): Underline shows which topic you need to understand for the exam.
Everything else in today's lecture is considered optional background knowledge.

Speech communication

Speech communication: a message, stored in Alice's brain, is converted to language, then converted to speech. Bob hears the speech, and decodes it to get the message.



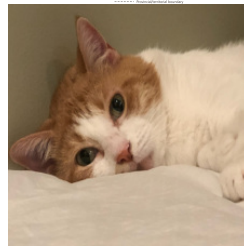
What sort of messages do humans send?

(Levelt, *Speaking*, 1989)

Experiments have shown at least three different ways that knowledge can be stored in long-term memory:

$\forall p \exists q: \text{Carries}(p, q)$

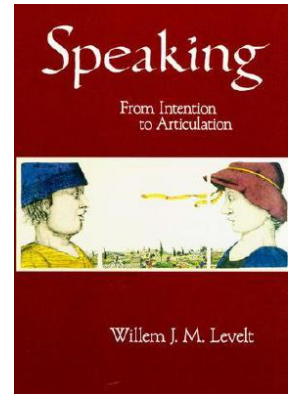
```
for x in range(0,10):  
    print('This is the %sth line'%(x))
```



visual/
spatial

logical/
propositional/
linguistic

procedural/
kinematic/
somatosensory

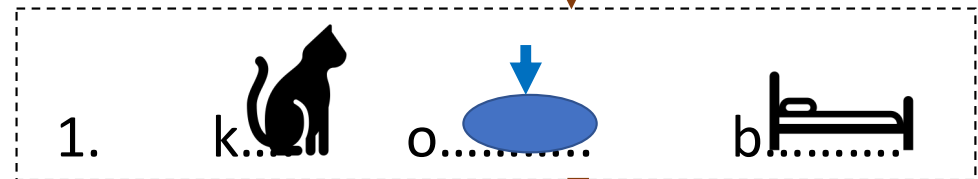
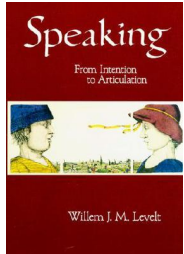
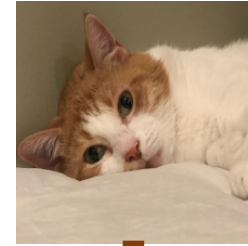


Speaking

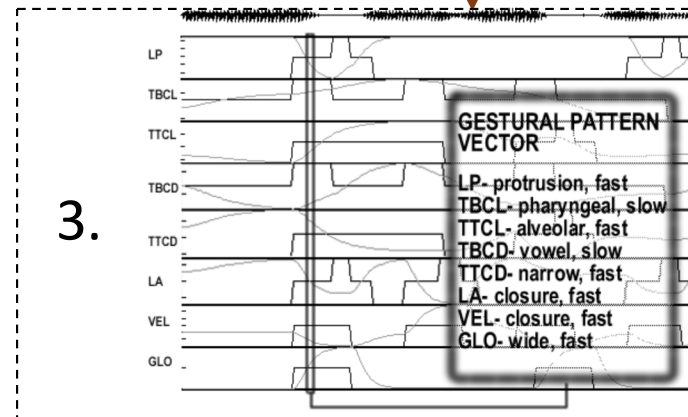
Experiments show that speaking consists of the following distinct mental activities, with no feedback from later to earlier activities:

- Step 1: convert to propositional form. Experiments show that the **starting sound** and **meaning** of each word are known before the order of the words is known.
- Step 2: fill in the pronunciation of each word
- Step 3: plan a smooth articulatory trajectory
- Step 4: speak

visual/spatial
source knowledge



2. the cat is on the bed



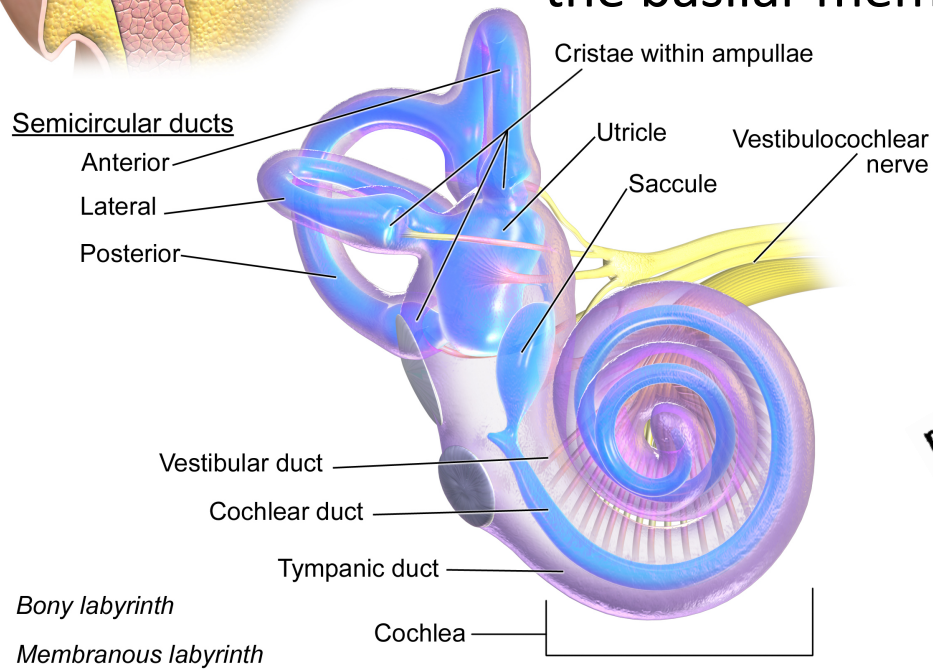
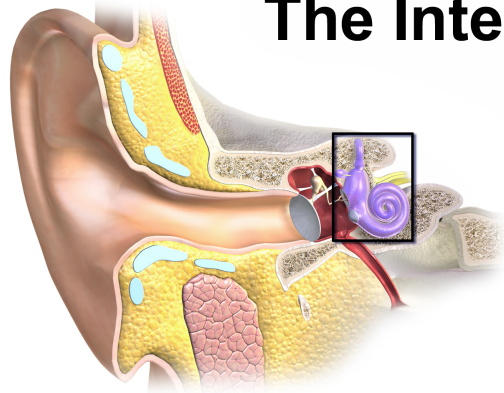
Speech perception

1. Most structures of the ear: protect the basilar membrane
2. Basilar membrane: a mechanical continuous bank of band-pass filters
3. Inner hair cell: mechanoelectric transduction; half-wave rectification
4. Auditory nerve: dynamic range compression
5. Brainstem: source localization, source separation, echo suppression
6. Auditory cortex: continuous-to-discrete conversion, from acoustic spectra to probabilities of speech sound categories
7. Posterior Middle Temporal Gyrus: word sequence candidates compete with each other to see which one can be the most probable

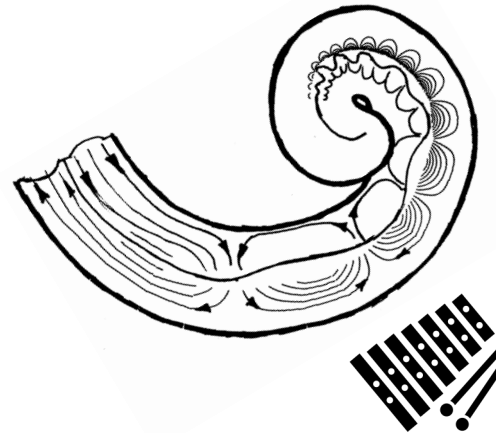
The Internal Ear

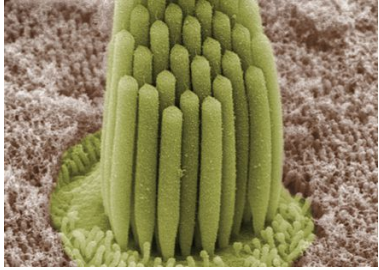
1. The main purpose of most of the structures of the ear is just to protect the basilar membrane.

2. The basilar membrane, down the center of the cochlea, is like a continuous xylophone: tuned to different frequencies at different locations



By Dicklyon (talk) (Uploads) - Own work, Public Domain, <https://en.wikipedia.org/w/index.php?curid=12469498>



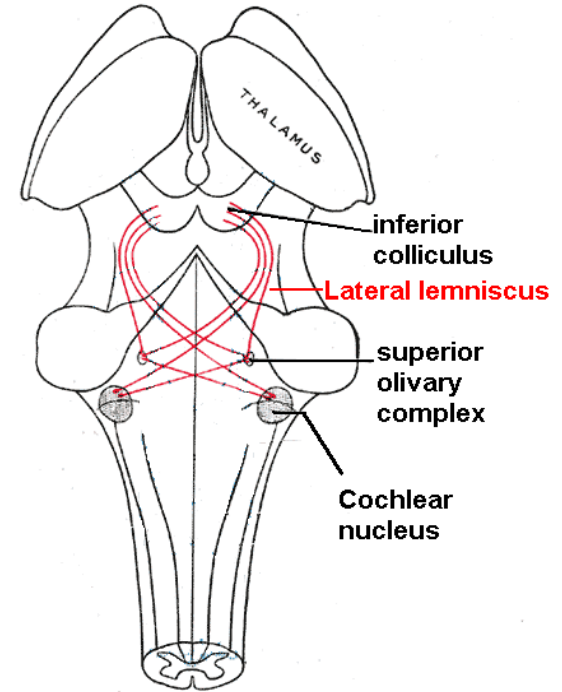


By Bechara Kachar - <http://irp.nih.gov/our-research/research-in-action/high-fidelity-stereocilia/slideshow>, Public Domain, <https://commons.wikimedia.org/w/index.php?curid=24468731>

4. Dynamic range compression: each hair cell connected to ~ 10 neurons, with thresholds distributed so that # cells that fire $\sim \log(\text{signal amplitude})$

3. The basilar membrane is covered with little hair bundles.

- When the membrane moves upward, pores in the tip of each hair cell open, depolarizing the cell.
- When the membrane moves downward, no response.
- So the hair cell is like a ReLU: $y = \max(0, x)$



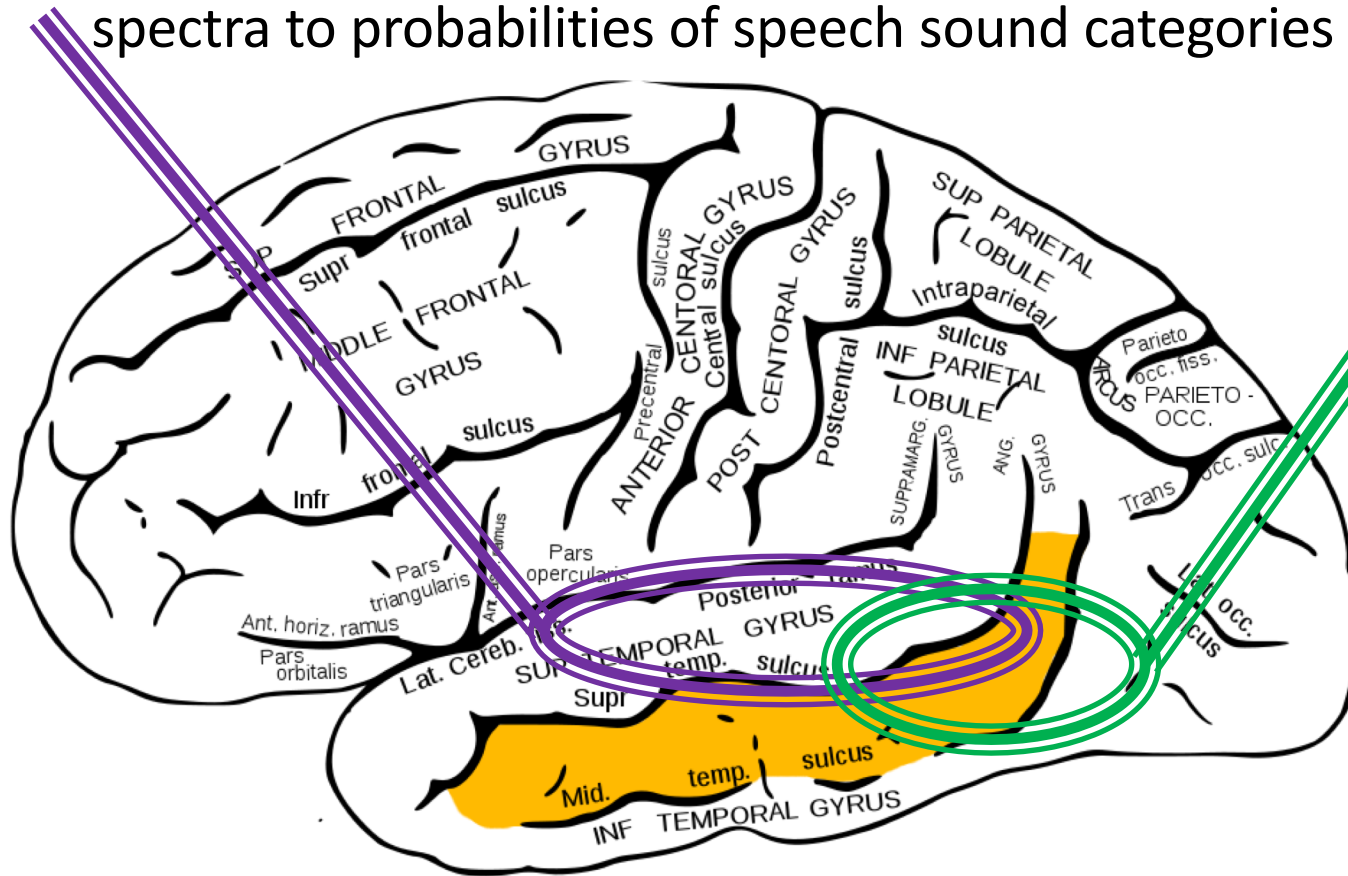
5. Neurons from the ear go to the brainstem, where:

- Cochlear nucleus does echo cancellation.
- Olivary complex, lateral lemniscus, & inferior colliculus do localization.

After the sound reaches the cortex: final processing

(6. Mesgarani & Chang, 2012; 7. Hickok & Poeppel, 2007;)

6. Auditory cortex: continuous-to-discrete conversion, from acoustic spectra to probabilities of speech sound categories



7. Posterior Middle Temporal Gyrus: word sequence candidates compete with each other to see which one can be the most probable

Outline

- Human speech processing
- Modeling the ear: Fourier transforms and filterbanks
- Speech-to-text-to-speech (S2T2S)
 - Hybrid DNN-HMM systems (*)
 - Recurrent neural nets (RNNs): Connectionist Temporal Classification, Trajectory nets
 - Sequence-to-sequence RNNs with attention
- Languages other than English
 - Training and testing on 300 languages
 - Transfer learning: from languages with data, to languages without
 - Dialog systems for unwritten languages
- Distorted speech
 - Motor disability
 - Second-language learners

(*): Underline shows which topic you need to understand for the exam.
Everything else in today's lecture is considered optional background knowledge.

Reminder: Image Features

You've seen this slide before, in lecture 24, on Deep Learning...

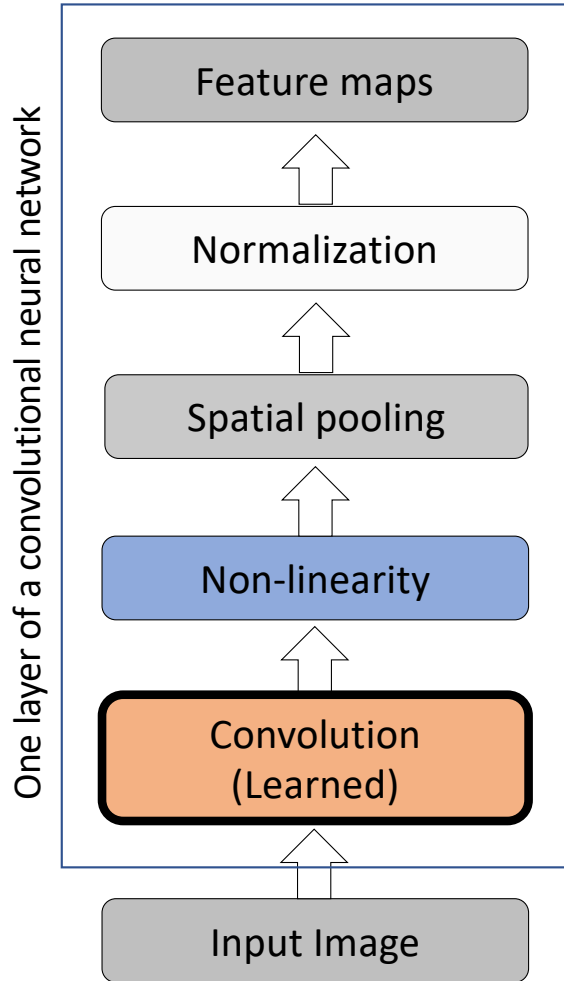
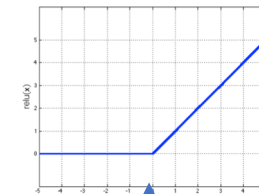
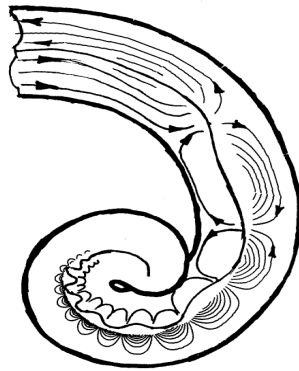


Image features are calculated by convolution, followed by ReLU.



Speech features as computed by the ear

- The basilar membrane is like convolution with a bank of bandpass filters



=

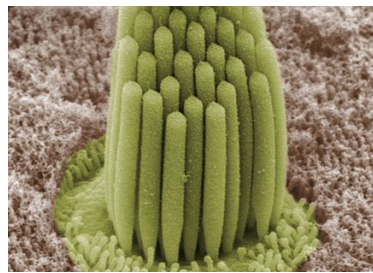


Input

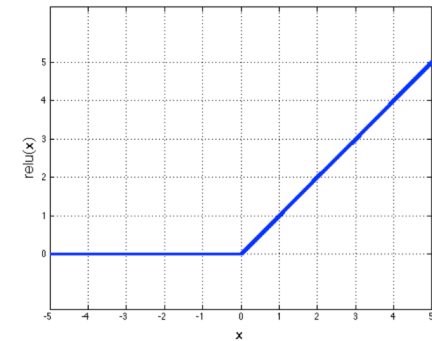


Feature Map

- The hair cell performs half-wave rectification (ReLU)



=



Artificial speech features

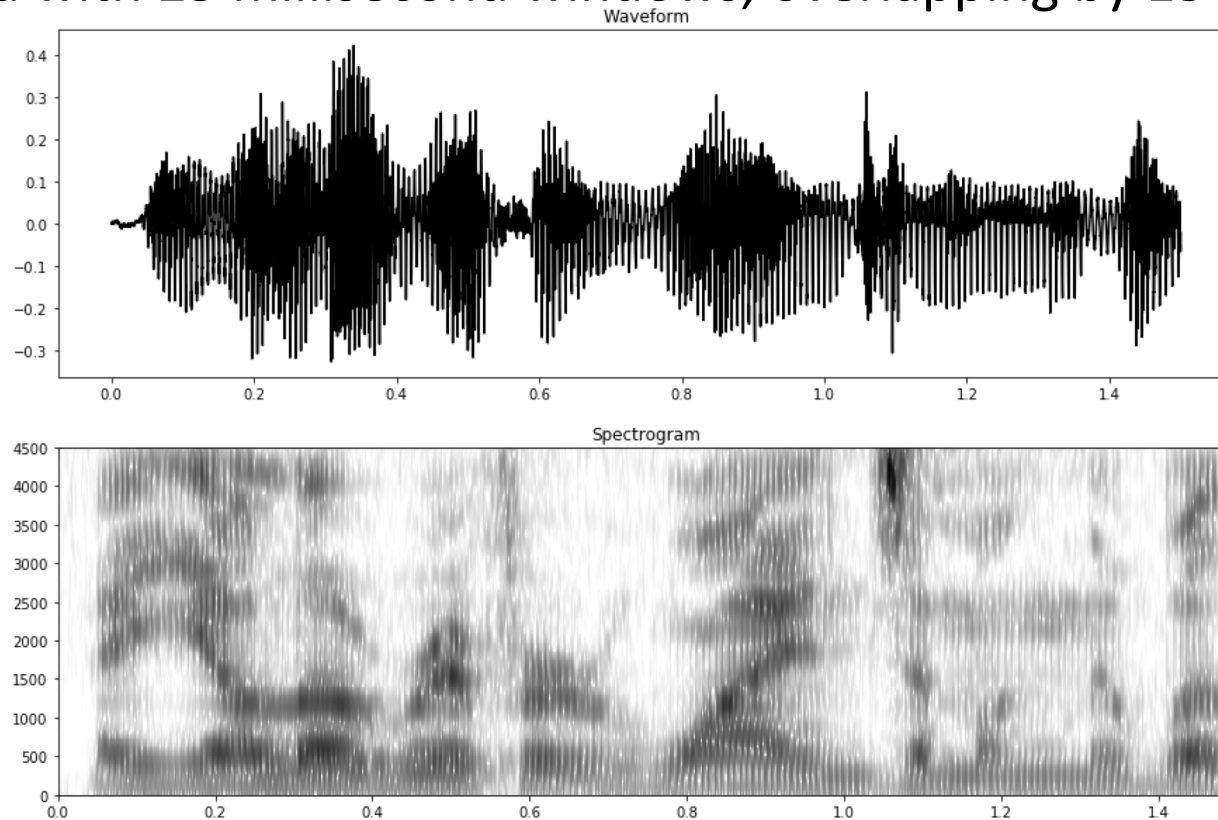
- Option 1: Exact matching of the filter/rectify operations of the basilar membrane
- Option 2: Learned filters, by applying a convolution directly to the input speech signal
- Option 3: Fast Fourier transform of the input speech signal, then compute the magnitude

Artificial speech features: Experimental results

- Option 1: Exact matching of the filter/rectify operations of the basilar membrane
 - Computationally expensive; results are sometimes better than options 2&3, but often not
- Option 2: Learned filters, by applying a convolution directly to the input speech signal
 - Computationally cheap during test time, but very expensive during training
 - Usually turns out to be exactly the same accuracy as option #3 --- in fact, the convolution kernels that are learned usually turn out to look like the kernels of a Fourier transform!
- Option 3: Fast Fourier transform of the input speech signal, then compute the magnitude

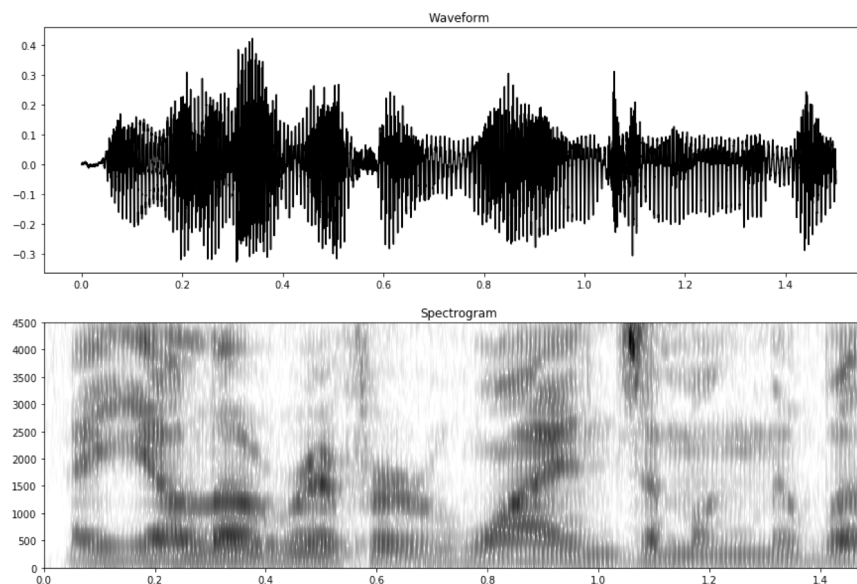
Artificial speech features: My recommendation

Spectrogram = $\log|X(f)|$ = log magnitude of the Fourier transform, computed with 25 millisecond windows, overlapping by 15 milliseconds



Speech synthesis from the spectrogram

- Exact reconstruction is possible from the complex FFT, but not from the FFT magnitude
 - If you have the **true FFT magnitude**, and your windows **overlap by at least 50%**, then exact reconstruction is possible
 - But if you have **synthetic FFT magnitude** (e.g., generated by an HMM or a neural net), then it **might not match any true speech signal**.
- If you have a synthetic FFT magnitude, you need to synthesize speech that is a “good match:”
 - Reconstruct a signal that matches the FFT with minimum squared error (**Griffin-Lim algorithm**)
 - Use a neural net to estimate the signal from the FTM (e.g., **wavenet**)



Griffin-Lim
or
wavenet

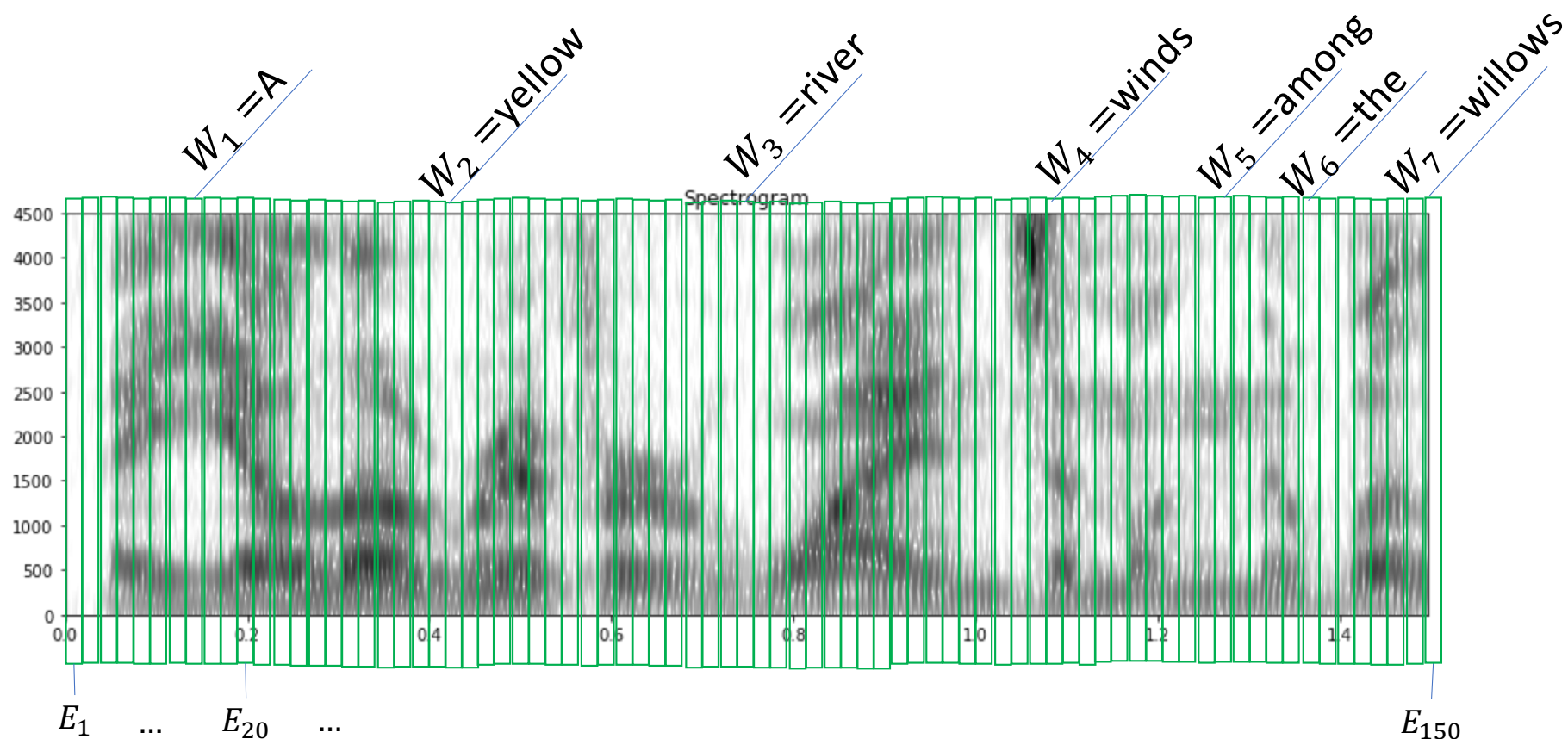
Outline

- Human speech processing
- Modeling the ear: Fourier transforms and filterbanks
- Speech-to-text-to-speech (S2T2S)
 - Hybrid DNN-HMM systems (*)
 - Recurrent neural nets (RNNs): Connectionist Temporal Classification, Trajectory nets
 - Sequence-to-sequence RNNs with attention
- Languages other than English
 - Training and testing on 300 languages
 - Transfer learning: from languages with data, to languages without
 - Dialog systems for unwritten languages
- Distorted speech
 - Motor disability
 - Second-language learners

(*): Underline shows which topic you need to understand for the exam.
Everything else in today's lecture is considered optional background knowledge.

The speech-to-text problem

From a spectrogram input (sequence of T vector observations, E_1 to E_T), compute a word sequence output (sequence of L label outputs, W_1 to W_L , $L < T$)



A Sequence Model you Know: HMM

You've seen this slide before, in lecture 20, on HMMs...

- **Markov assumption for state transitions**

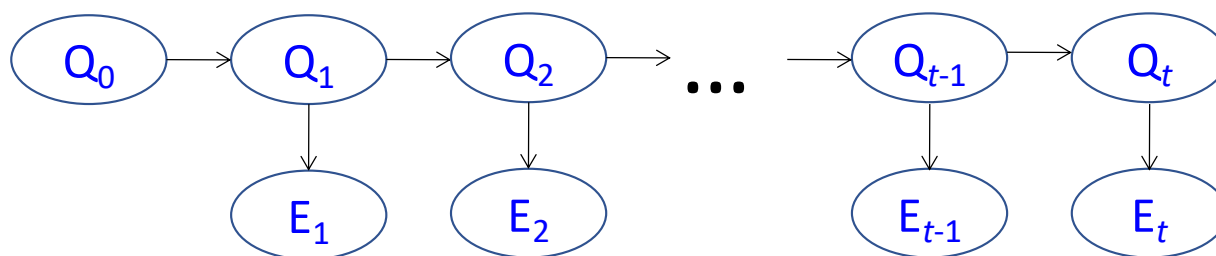
- The current state is conditionally independent of all the other states given the state in the previous time step

$$P(Q_t \mid \mathbf{Q}_{0:t-1}) = P(Q_t \mid Q_{t-1})$$

- **Markov assumption for observations**

- The evidence at time t depends only on the state at time t

$$P(E_t \mid \mathbf{Q}_{0:t}, \mathbf{E}_{1:t-1}) = P(E_t \mid Q_t)$$



HMMs for Speech Recognition

1. Decide, in advance, how many states each word will have. For example, choosing # states = three times # phonemes usually works well. (Get phonemes from an online dictionary, like ISLEdict)
 - “yellow” = j ε l oʊ, 4 phonemes = 12 states
 - “winds” = w aɪ n d z, 5 phonemes = 15 states



```
yelled(+yell+ed,vbu,vbu) # j 'ε . l u π #
yellen(nnp_surname_0.000) # j 'ε . l ŋ #
yeller(+yell+er,nn) # j 'ε . l ,ə #
yellin() # j 'ε . l ɪ n #
yelling(+yell+ing,vbg) # j 'ε . l ɪ ŋ #
yellow(jj,nn,nnp_surname_0.000,vb) # j 'ε . l oʊ #
yellow_alert() # j 'ε . l oʊ # ə . l 'ə ɪ t #
yellow_bell() # j 'ε . l oʊ # b ε l #
yellow_bile() # j 'ε . l oʊ # b aɪ l #
yellow_birch() # j 'ε . l oʊ # b ɜ ɪ tʃ #
```

Search: yellow | Highlight All | Match Case | Whole Words | 22 of 250 matches

HMMs for Speech Recognition

1. Decide, in advance, how many states each word will have. For example, choosing # states = three times # phonemes usually works well. (Get phonemes from an online dictionary, like ISLEdict)
 - “yellow” = j ε l oʊ, 4 phonemes = 12 states
 - “winds” = w aɪ n d z, 5 phonemes = 15 states
2. Pool together, across words, the HMM states that sound similar. For example, the 4th state in the word “winds” might be called “*the 1st state of the phoneme aɪ, in words where that phoneme follows w and precedes n*” (denoted $Q=w-aɪ+n_1$), and it would share parameters with all other words that have a similar-sounding aɪ.
3. Those pooled HMM-states are called **senones**. You’ll have more or less senones, depending on how you define „similar-sounding,” but most speech recognizers have about 3000-5000 of them.

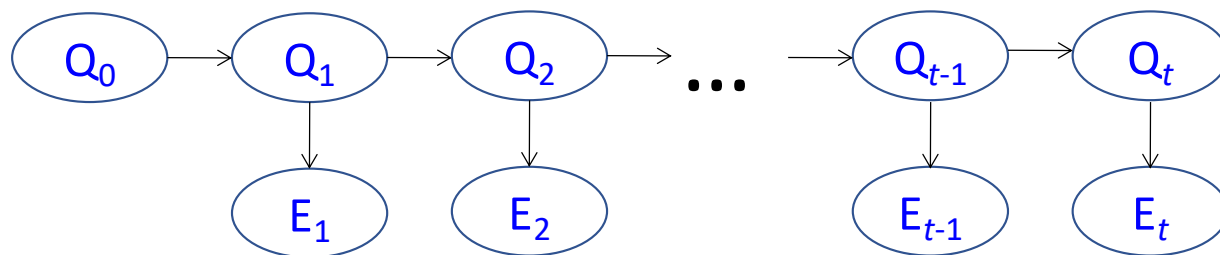
HMMs for Speech Recognition

- Now the HMM parameters depend on which word you're recognizing!

- For example, the transition probabilities for word W are now

$$P_W(Q_t | Q_{t-1})$$

- W = the word being spoken
- Q_t = the senone being spoken at time t



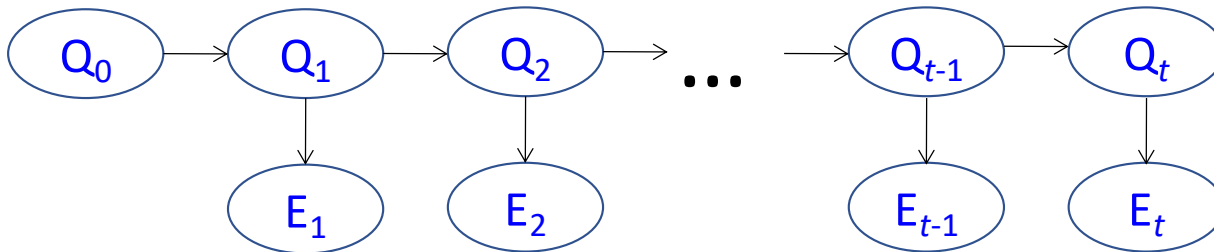
- Every word has the same **structure**, but

- Different words have different **parameters**, for example,

$$P_W(Q_t | Q_{t-1})$$

The Problem of Continuous Observations

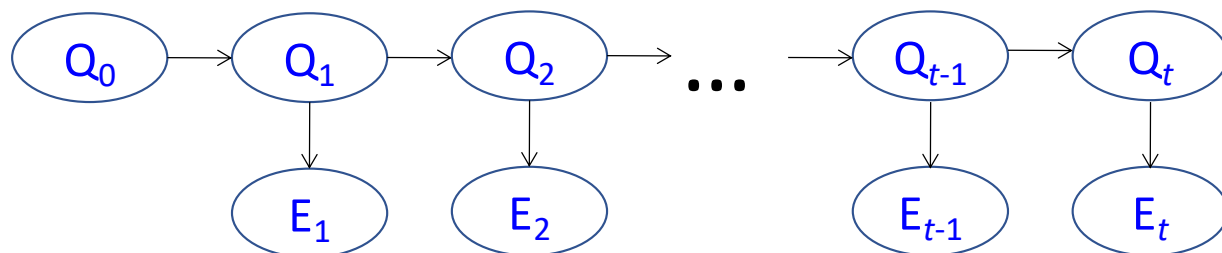
- But what about the likelihood? How can we model $P(E_t | Q_t)$?
- The big problem: E_t is continuous, not discrete, so we can't model $P(E_t | Q_t)$ using a lookup table!



Solutions to the Problem of Continuous Observations

Most systems model $P(E|Q)$ using one of these three standard methods:

1. Use a parameterized probability density, such as a Gaussian. In this case you learn senone-dependent parameters (μ_Q and σ_Q^2).
2. Quantize E (using vector quantization) to one of K different code vectors. Then you can learn the lookup table $P_W(E = k|Q)$ for $1 \leq k \leq K$.
3. Use a neural net with a softmax output to compute $P(Q|E)$, then use Bayes' rule to get $P(E|Q)$ from $P(Q|E)$.



Classifier output: Softmax

You've seen this slide before, in lecture 24, on Deep Learning....

- We want Q_t to be a senone, for example, $Q_t =$ “the j th type of phoneme α ”.
- In that case, we can force the neural net to learn what the neural net to compute a probability,

$$F_j = P(Q = j | E)$$

...if we just force F_j to meet the criteria for a probability, i.e., we need

$$F_j \geq 0, \quad \sum_j F_j = 1$$

- In order to do that, we use a special kind of nonlinearity in the last layer of the neural net, called a softmax:

$$F_j = \frac{e^{Z_j}}{\sum_k e^{Z_k}}$$

Hybrid DNN-HMM: the problem

- The softmax computes $P(Q|E)$
- The HMM needs to know $P(E|Q)$
- How can we get $P(E|Q)$ from $P(Q|E)$?
- Answer: Bayes' rule!

Estimating $p(E | Q)$ from $p(Q | E)$

Bayes rule:

$$P(E|Q) = \frac{P(Q|E)P(E)}{P(Q)}$$

... but notice, if our goal is to find the best possible state sequence Q_1, \dots, Q_T , then we don't care about the $P(E)$ factor:

$$\operatorname{argmax}_Q P(E|Q) = \operatorname{argmax}_Q \frac{P(Q|E)}{P(Q)}$$

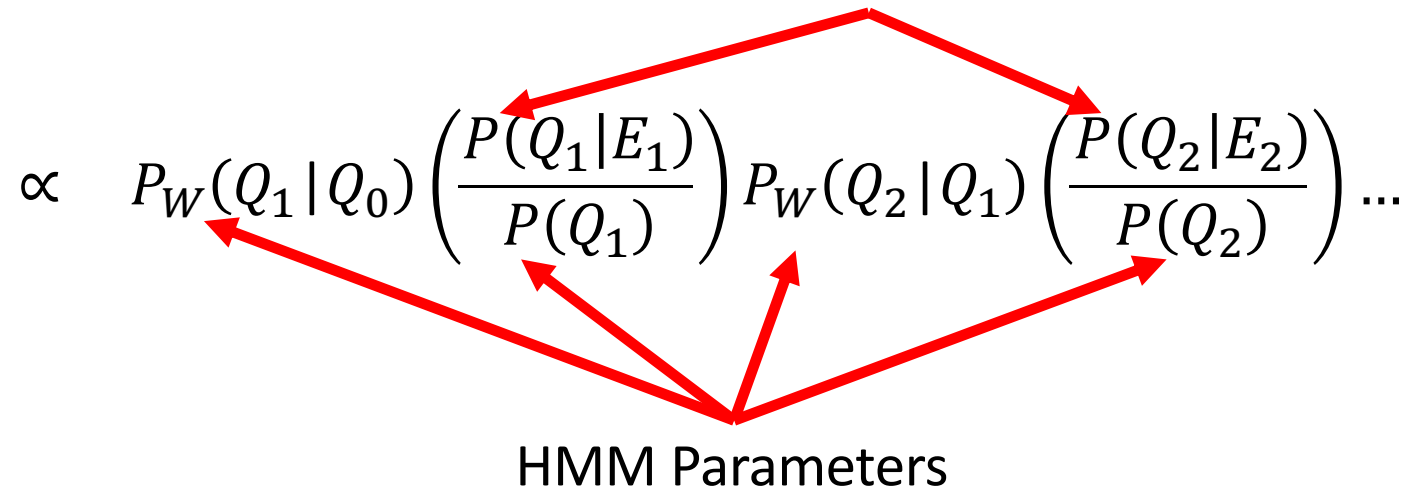
Hybrid DNN-HMM: the solution

$$P(E_1, E_2, Q_1, Q_2, \dots | W) = P_W(Q_1 | Q_0) P(E_1 | Q_1) P_W(Q_2 | Q_1) P(E_2 | Q_2) \dots$$

From the neural net

$$\propto P_W(Q_1 | Q_0) \left(\frac{P(Q_1 | E_1)}{P(Q_1)} \right) P_W(Q_2 | Q_1) \left(\frac{P(Q_2 | E_2)}{P(Q_2)} \right) \dots$$

HMM Parameters



Hybrid DNN-HMM: intuitive explanation

- Prior probability, $p(Q)$, tells how frequently HMM state Q is, in normal conversations, **if we don't hear the speech**
- DNN computes a posterior probability, $p(Q|E)$, saying how probable Q is **given the available evidence**
- If $p(Q|E) > p(Q)$, that means that **the evidence favors Q more than usual**, so we should consider the possibility that this rare word has been spoken.
- If $p(Q|E)$ is still a small number, that doesn't really matter; what really matters is whether $p(Q|E) > p(Q)$

Speech synthesis using an HMM

Given the word sequence, W :

- Use $P_W(Q_t | Q_{t-1})$, with a random number generator, to generate a random state sequence that matches the given word sequence
- Run the neural net backward to generate a spectrum:
 - Set $Z^{(L)}$ to a vector with all zeros, except some gain G in the Q 'th entry
 - Invert the matrix at each level to find $A^{(l-1)}$ from $Z^{(l)}$
 - The last level (going backward!), $A^{(0)}$, is the spectrum
- Use Griffin-Lim or wavenet to generate signal from spectrogram
- This method results in discontinuous jumps at HMM-state boundaries. Solution: recurrent neural net

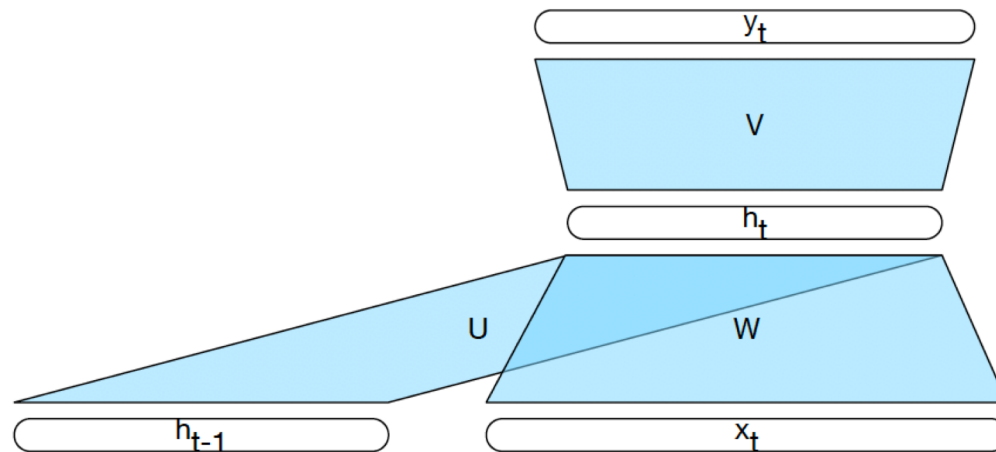
Outline

- Human speech processing
- Modeling the ear: Fourier transforms and filterbanks
- Speech-to-text-to-speech (S2T2S)
 - Hybrid DNN-HMM systems (*)
 - Recurrent neural nets (RNNs): Connectionist Temporal Classification, Trajectory nets
 - Sequence-to-sequence RNNs with attention
- Languages other than English
 - Training and testing on 300 languages
 - Transfer learning: from languages with data, to languages without
 - Dialog systems for unwritten languages
- Distorted speech
 - Motor disability
 - Second-language learners

(*): Underline shows which topic you need to understand for the exam.
Everything else in today's lecture is considered optional background knowledge.

Basic RNNs

Each time step corresponds to a feedforward net where the hidden layer gets its input not just from the layer below but also from the activations of the hidden layer at the previous time step



A recurrent net for speech synthesis

- Output #1 is the magnitude FFT
- Output #2 is the state vector, which is an input to the next time step
- Input is a list of all of the HMM states within a window of +/-D frames of the current frame, $X_t = [Q_{t-D}, \dots, Q_t, \dots, Q_{t+D}]$
- This is called a “trajectory mixture density network” (Korin Richmond, 2007)

A recurrent net for speech recognition

- Output #1 is a softmax over
 - HMM states, Q , for DNN-HMM hybrid speech recognition
 - Words, if the RNN is being used by itself for stand-alone speech recognition
- Output #2 is a state vector, which is fed back as input to the same neural net at time $t+1$

Connectionist temporal classification: Speech recognition using a stand-alone RNN

- The problem solved by CTC: T input frames, $K < T$ output words
- The solution:
 - Softmax outputs = {set of all known words, or “blank”}
 - State sequence is $Q = \{Q_1, \dots, Q_T\}$
 - Label sequence is the set of words $W = \{W_1, \dots, W_l\}$
 - The set of all state sequences that match W includes all state sequences that produce the words in W , with any combination of blanks in between them. This set is called $B(W)$
 - Neural net training criterion is $-\log P(B(W|E)) = -\log \sum_{Q \in B(W)} P(Q|E)$
- Training algorithm: Graves et al., 2006
- Speech recognition application: Miao and Metze, 2014

Outline

- Human speech processing
- Modeling the ear: Fourier transforms and filterbanks
- Speech-to-text-to-speech (S2T2S)
 - Hybrid DNN-HMM systems (*)
 - Recurrent neural nets (RNNs): Connectionist Temporal Classification, Trajectory nets
 - Sequence-to-sequence RNNs with attention
- Languages other than English
 - Training and testing on 300 languages
 - Transfer learning: from languages with data, to languages without
 - Dialog systems for unwritten languages
- Distorted speech
 - Motor disability
 - Second-language learners

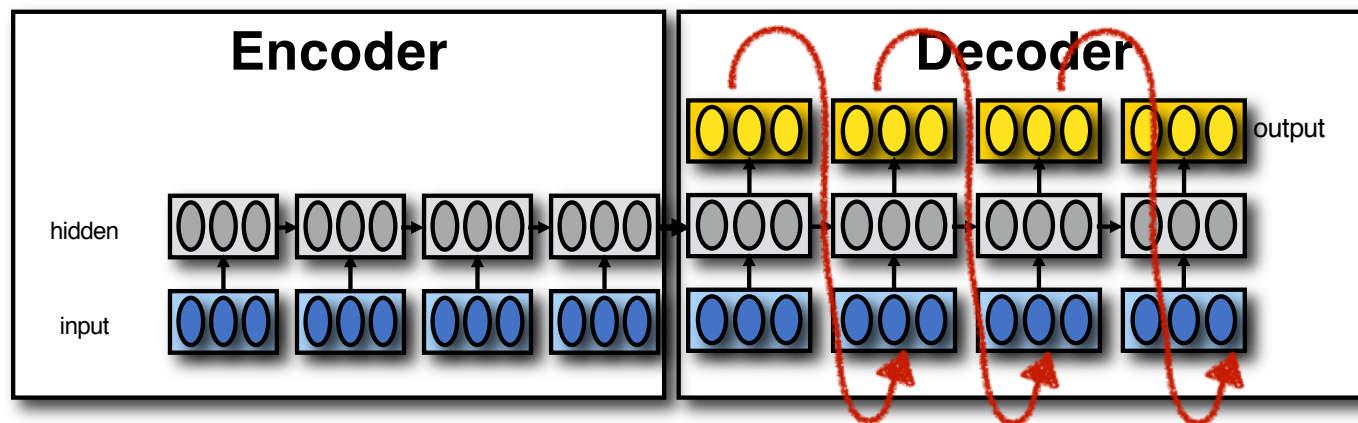
(*): Underline shows which topic you need to understand for the exam.
Everything else in today's lecture is considered optional background knowledge.

Sequence-to-sequence with attention

- Input encoder is an RNN
- Output decoder is an RNN
- Each cell of the output decoder takes, as input, a weighted summation of the input encoder hidden nodes vectors, concatenated to the previous output-time state vector, concatenated to a unit indicator showing which output was generated in the previous time
- Weights for the weighted summation change, from one output-time to the next. The weights themselves are computed by another neural net.

Encoder-Decoder (seq2seq) model

- Task: Read an input sequence and return an output sequence
 - Machine translation: translate source into target language
 - Dialog system/chatbot: generate a response
- Reading the input sequence: RNN Encoder
- Generating the output sequence: RNN Decoder



Outline

- Human speech processing
- Modeling the ear: Fourier transforms and filterbanks
- Speech-to-text-to-speech (S2T2S)
 - Hybrid DNN-HMM systems (*)
 - Recurrent neural nets (RNNs): Connectionist Temporal Classification, Trajectory nets
 - Sequence-to-sequence RNNs with attention
- Languages other than English
 - Training and testing on 300 languages
 - Transfer learning: from languages with data, to languages without
 - Dialog systems for unwritten languages
- Distorted speech
 - Motor disability
 - Second-language learners

(*): Underline shows which topic you need to understand for the exam.
Everything else in today's lecture is considered optional background knowledge.

Can speech recognition and speech synthesis be trained for any language?

- As far as we know, the algorithms work for any language, as long as you have labeled training data
- "Labeled data" = speech files, together with their text transcriptions
- Having audio is not enough, because usually there is no transcription. Having text is not enough, because usually you don't know how it sounds. You need matched text+audio.
- In how many languages do we have such corpora?

“Automatic Speech Recognition” corpora available from the Linguistic Data Consortium

- English: ~120 distinct corpora!!!
- ≥ 10 corpora: Arabic, Chinese, Hindi, Japanese, Korean, Spanish
- 2-10 corpora: Czech, French, German, Italian, Portuguese
- 1 corpus: 24 languages
- What about all of the other languages in the world?

Suggested solution: use transfer learning, from well-resourced languages, to learn speech recognition for under-resourced languages.

Every phoneme system in the world
differentiates these two categories of phonemes:



Phonemes made with the
mouth open,
e.g.,

Vowels,
Approximants,
(Fricatives)



Phonemes made with the
mouth closed,
e.g.,

Clicks,
Plosives,
(Nasals, Taps, Trills)

The acoustic consequences of mouth opening and closing

Acoustic Landmark = perceptually salient instantaneous marker of phoneme presence.

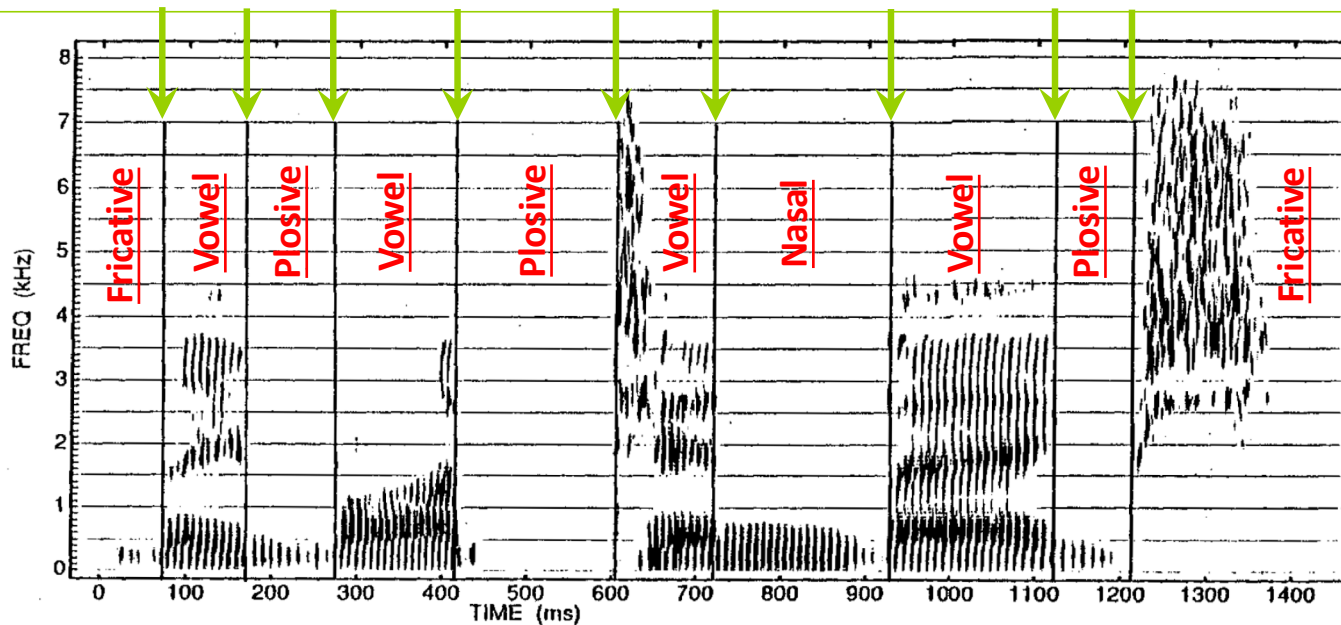


Fig. 1 Spectrogram of the utterance "They bought ten bags." The vertical lines indicate times of implosion or release of articulators when consonants are produced.

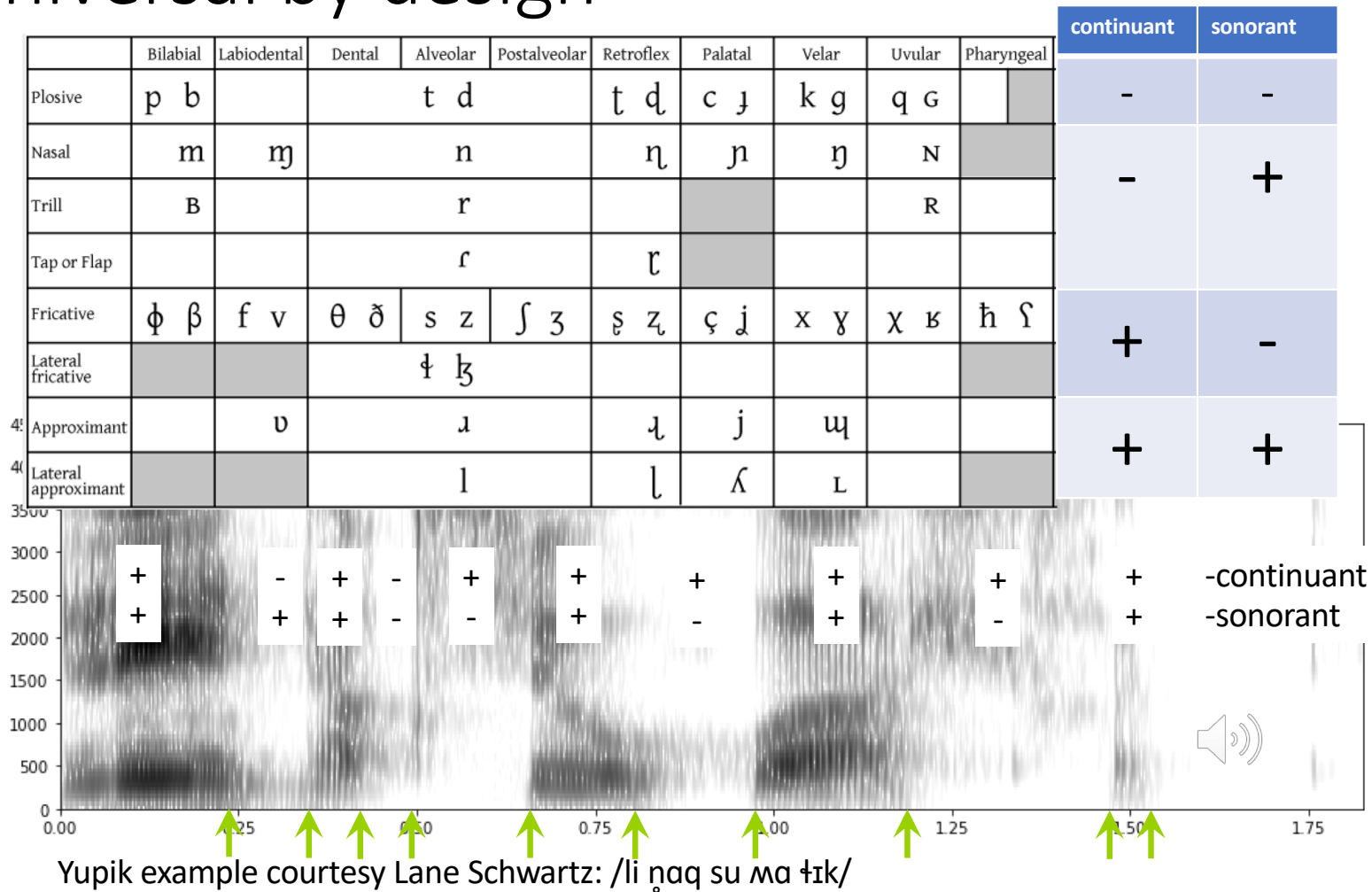
(Stevens, Manuel, Shattuck-Hufnagel & Liu, ICSLP1992).

Articulatory Features as Linguistic Universals

- Articulatory features are designed to be a superset of the phoneme distinctions in every language (universal by design).
- The universal features “mouth open” and “mouth closed” can be summarized by just two features: [sonorant] and [continuant]
 - [+continuant]: mouth is unobstructed along midline of the vocal tract
 - [+sonorant]: mouth is open in the sense that there is a low-acoustic-impedance shunt from vocal folds to air (though the shunt might go through the nose, or around the tongue tip)

	[-sonorant]	[+sonorant]
[-continuant]	Plosives	Nasals, Flaps, Trills
[+continuant]	Fricatives	Vowels, Approximants

“Universal by design”



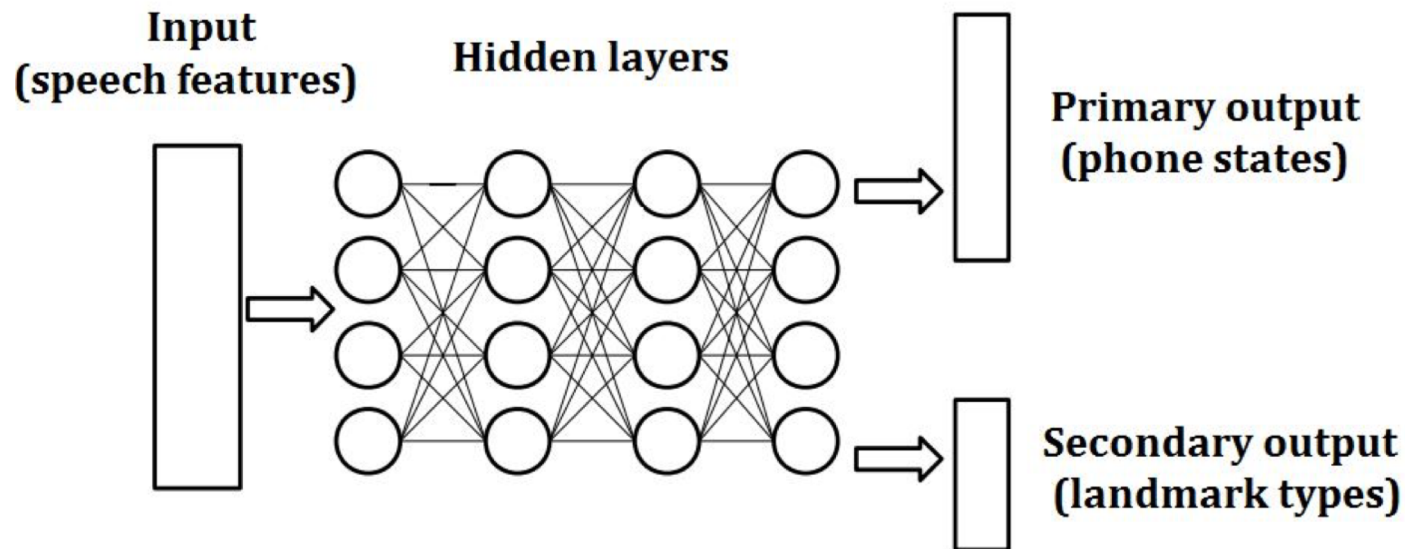
Landmarks as regularizers for training a DNN-HMM phone model

- Train on English (TIMIT, 14 hours):
 - Train two DNNs, one for phones, one for landmarks. Result: phone models get better time alignment during training.
 - Test: phone error rate.
- Adapt to Iban (Juan, Besacier & Rossato, 8 hours):
 - Automatic landmark detection, phones force-aligned, then adapt both DNNs from English to Iban.
 - Test: word error rate.

Multi-task learning (MTL)

MTL = Train one neural network on multiple tasks (1 set of inputs vs. multiple sets of labels). Can reduce overfitting, generalize better to testing data if

- a) training data limited (under-resourced), or
- b) the tasks compliment each other (e.g., landmark detection & phone recognition)



Experiments: 2 types of landmark definitions, 2 types of ASR

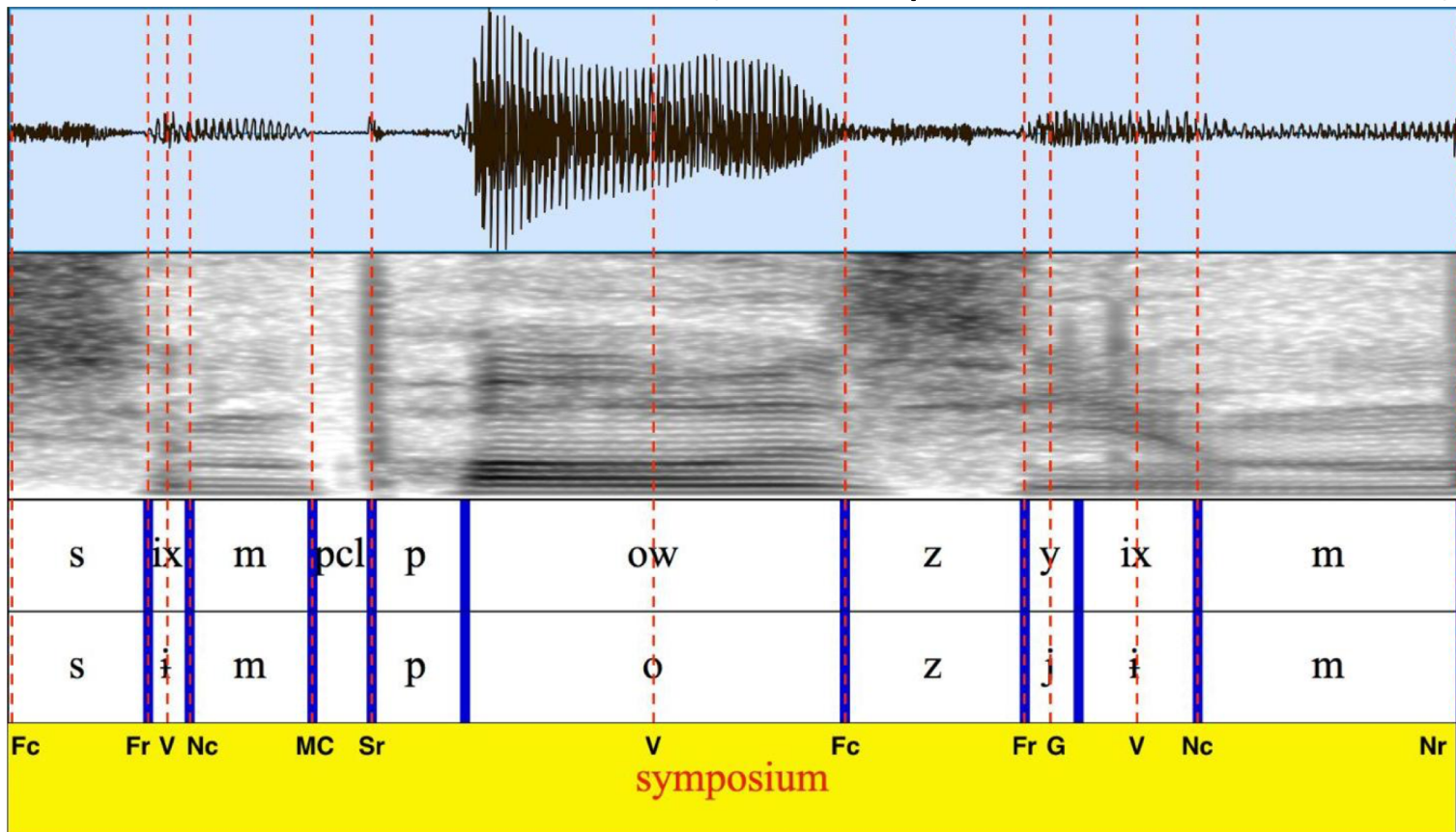
- Two types of landmark definitions
 - Experiment 1: release/closure/middle notation
 - Experiment 2: change in value of the features [continuant] or [sonorant]
- Two types of ASR
 - Experiment 1: TDNN-HMM hybrid
 - Experiment 2: CTC

Experiment 1: closure/release/middle landmark notation (from phoneme labels)

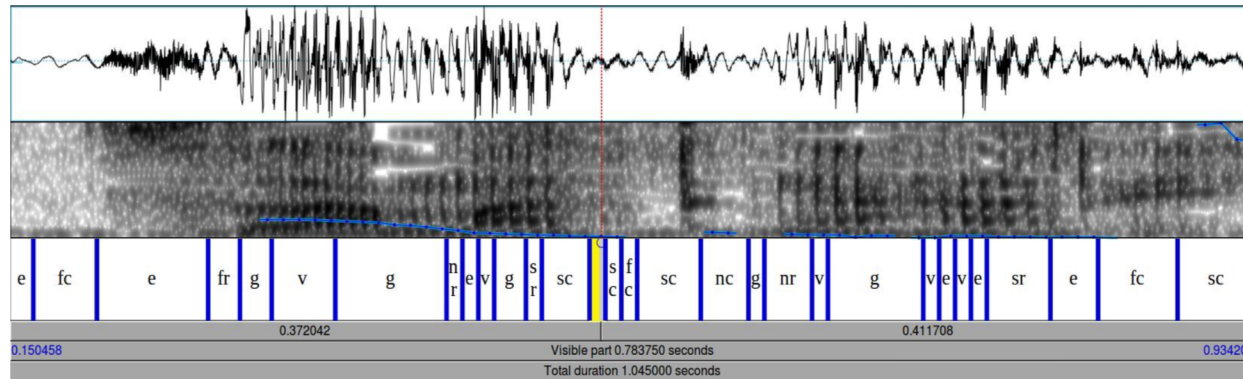
Landmark Type	Temporal midpoint of a...
V	Vowel
G	Glide

Landmark Type	At the end of a...	...and beginning of a...
Sc (Stop Closure)	Vowel or Glide	Stop Closure
Sr (Stop Release)	Stop Closure	Stop Release, Vowel or Glide
Fc (Fricative Closure)	Vowel or Glide	Fricative
Fr (Fricative Release)	Fricative or Affricate	Vowel or Glide
Nc (Nasal Closure)	Vowel or Glide	Nasal
Nr (Nasal Release)	Nasal	Vowel or Glide
MC (Manner Change)	Non-vowel, non-glide	Different type of non-vowel

Experiment 1: closure/release/middle landmark notation (from phoneme labels)



Migrate to the Iban language, step #1: Automatically generate landmark labels in Iban using TIMIT-trained landmark detectors



Migrate to Iban, step #2: Multi-task learning (MTL). Task 1 = phone labels, Task 2 = landmarks.

$$\mathcal{L}_x = (1 - \alpha c_x) \sum_{i=1}^{C^{ph}} (l_i^{ph} \log(P_i^{ph}(x)))$$

$$+ \alpha c_x \sum_{j=1}^{C^{la}} (l_j^{la} \log(P_j^{la}(x)))$$

$$c_x = P_m^{la_de}(x) - \frac{1}{C^{la} - 1} \sum_{k=1, k \neq m}^{C^{la}} (P_k^{la_de}(x))$$

l_i^{ph} : 1-hot label for phone recognition of phone state i (forced alignment)
 P_j^{la} : posterior probability for Landmark detection of Landmark j (automatic)
 α : a weighting factor between the 2 tasks
 c_x : confidence weighting for Landmark detect result on frame x

Experiment 1 Results

Average ASR Error rate:

- PER for TIMIT, WER for Iban
- Some Iban training data was randomly left out to simulate a **very-low-resource** scenario

Corpus	AM	Baseline	MTL	MTL w/ Confid
TIMIT (PER)	Mono	24.6	24.2	NA
	Tri	20.6	20.0	NA
Iban-full (WER)	Mono	24.62	24.22	24.18
	Tri	18.40	18.03	17.93
Iban-25% (WER)	Mono	28.87	27.97	27.64
	Tri	21.31	20.70	20.63
Iban-10% (WER)	Mono	31.16	28.49	28.48
	Tri	25.12	23.64	23.57

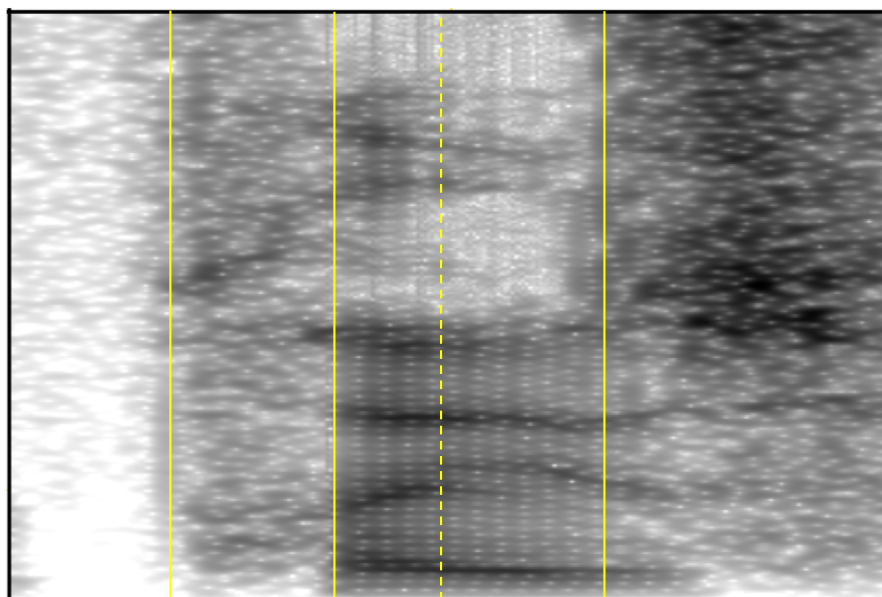
Assume TIMIT boundaries are 100% correct, no need for confidence weighting

As Training data reduce; error rate reduction increases for MTL

MTL error rate is **always lower** than baseline; MTL with confidence **always** returns **slightly lower** error rate

Experiment 2 notation: change in value of [continuant] or [sonorant]

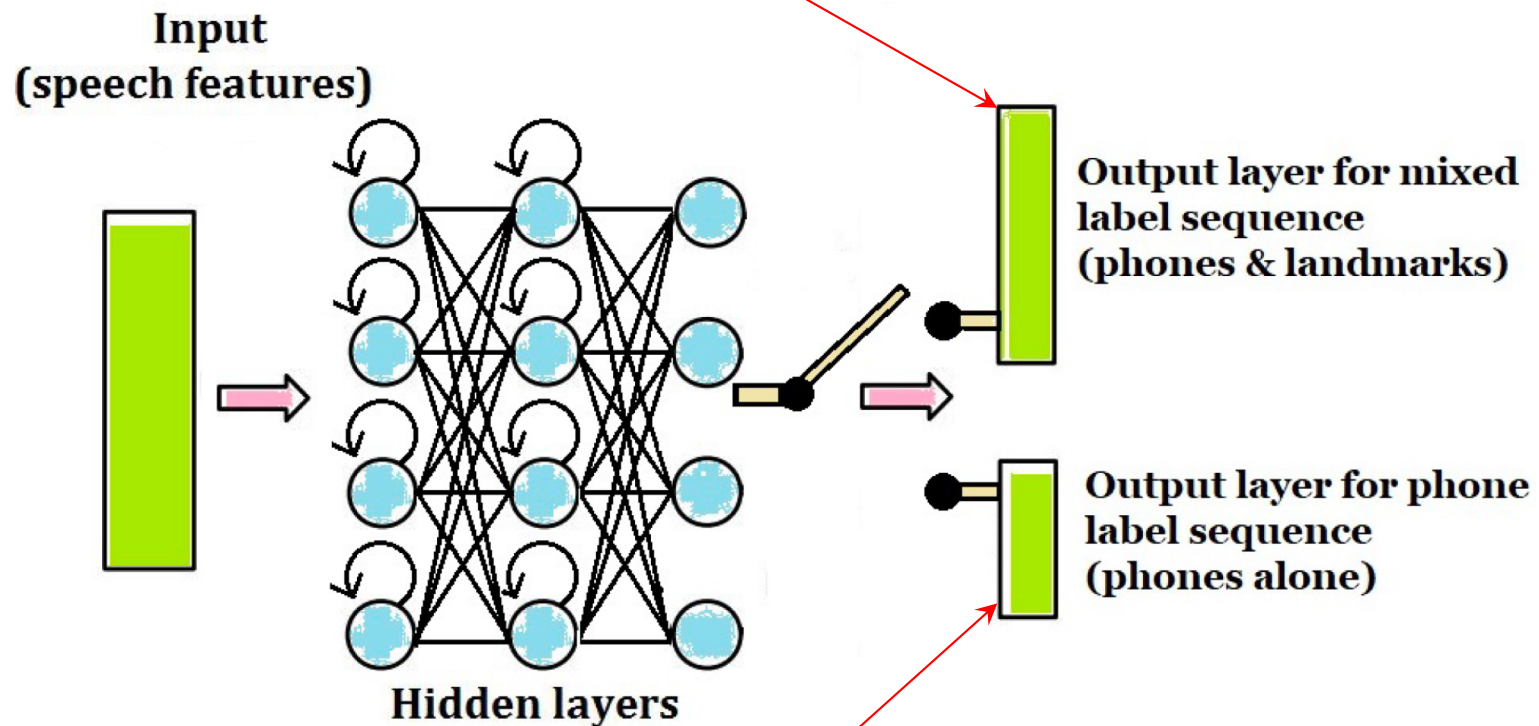
- **Phone Label:** CTC is trained to generate only the phone sequence, as given in TIMIT.
- **Mixed Label 1:** CTC also generates a landmark label every time [continuant] or [sonorant] changes value.
- **Mixed Label 2:** CTC generates a landmark label at every phone boundary, even if the values of [continuant] and [sonorant] don't change.



phone label	pcl	p	l	ey	s			
mixed label 1	pcl	--cont --sono	++cont --sono	l	ey	++cont --sono	s	
mixed label 2	pcl	--cont --sono	++cont --sono	l	++cont ++sono	ey	++cont --sono	s

Mixed Label Training + Phone Finetuning

1. First, Train (until convergence) to reproduce the mixed label set.

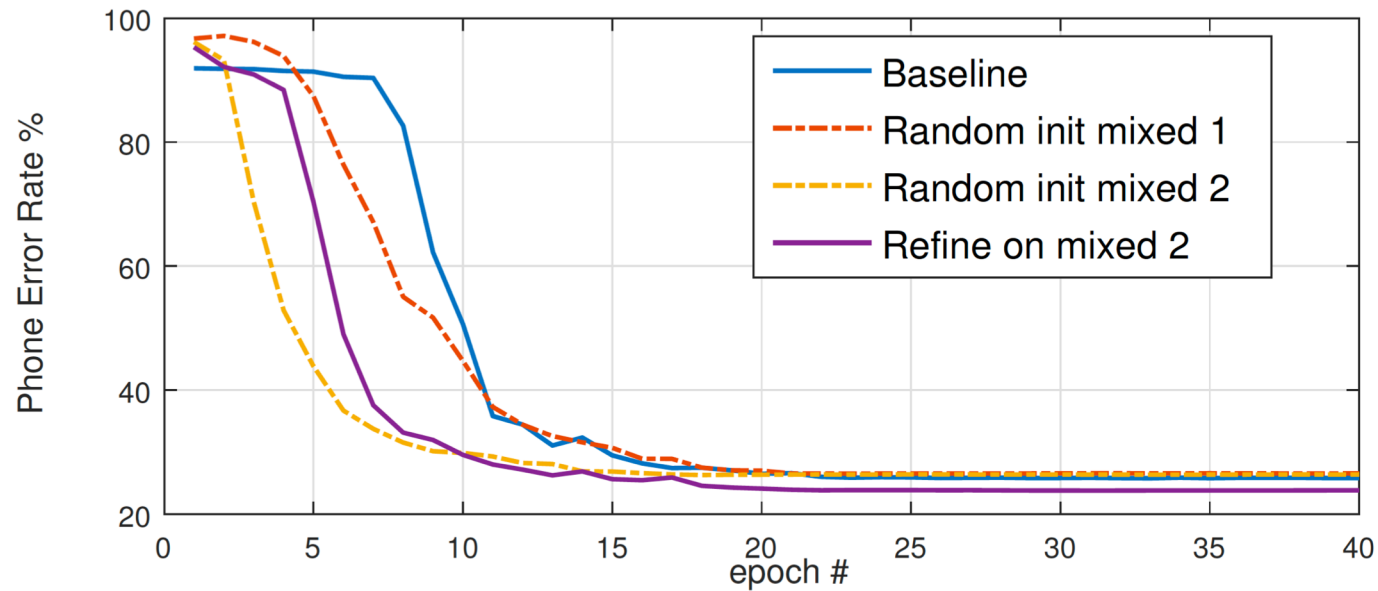


2. Then "Finetune:" continue to train, using phone labels only.

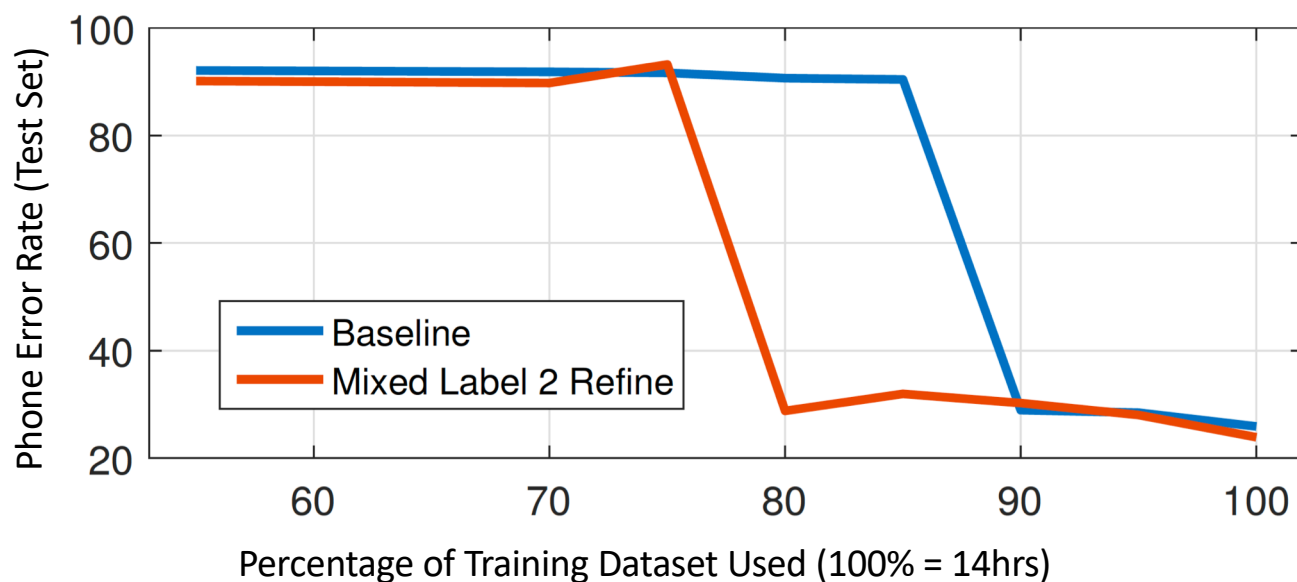
Experiment 2 results: PER and WER are reduced on both TIMIT and WSJ

Training labels	PER (TIMIT)	PER (WSJ eval92)	PER (WSJ dev93)	WER (WSJ eval92)	WER (WSJ dev93)
Phones	30.36	8.7	12.38	8.75	13.15
Phones + finetuning	30.36	Train to convergence using <u>phone labels</u> , then Finetune until convergence using <u>the same phone labels</u> : PER doesn't change (confirmed experimentally).			
Mixed 1	30.98	These numbers not calculated because Mixed 2 + finetuning was best on TIMIT.			
Mixed 1 + finetuning	28.96				
Mixed 2	29.10				
Mixed 2 + finetuning	27.72 (↓9% rel)	8.12	11.49	8.35	12.86

Experiment 2 results: CTC with mixed labels converges faster



Experiment 2 results: with mixed labels, CTC can be trained using a smaller training corpus.



Ongoing project: landmark-based ASR for 300 languages simultaneously

- The CMU-Wilderness corpus: Bibles, read in about 300 languages, transcribed by Alan Black at CMU
- General structure of the ongoing project:
 - Each student develops neural nets for a different type of landmark
 - We combine them all in a massively multilingual CTC-based system
 - Planned start: fall 2019, if students are interested

Outline

- Human speech processing
- Modeling the ear: Fourier transforms and filterbanks
- Speech-to-text-to-speech (S2T2S)
 - Hybrid DNN-HMM systems (*)
 - Recurrent neural nets (RNNs): Connectionist Temporal Classification, Trajectory nets
 - Sequence-to-sequence RNNs with attention
- Languages other than English
 - Training and testing on 300 languages
 - Transfer learning: from languages with data, to languages without
 - **Dialog systems for unwritten languages**
- **Distorted speech**
 - Motor disability
 - Second-language learners

How many languages are there in the world?

The screenshot shows a web browser window with the URL <https://www.ethnologue.com/guides/how-many-languages>. The page header includes the Ethnologue logo, a user greeting for the University of Illinois at Urbana-Champaign, and a search bar. A navigation menu contains 'Languages', 'Countries', and 'About'. A green notification box states: 'Welcome, University of Illinois at Urbana-Champaign. You have been automatically logged into Ethnologue.' The main article title is 'How many languages are there in the world?' followed by the text: '7,111 languages are spoken today. 🐦'. Below this, a paragraph explains that the number is constantly in flux and that roughly a third of languages are now endangered, with just 23 languages accounting for more than half the world's population. At the bottom, a world map titled 'Living Languages, 2019' shows language distribution by region: ASIA (purple), AFRICA (blue), PACIFIC (orange), AMERICAS (green), and EUROPE (yellow). A tooltip for 'Western Tawbuid' is visible over the Pacific region.

Our goal: Speech technology for unwritten languages

- Jelinek, 1976: speech recognition is defined as a transformation from speech to text.
- The problem: most languages don't have text.
 - About 50% of the world's languages: no orthography has ever been defined.
 - About 40% of the world's languages: orthography exists with multiple conflicting standards; native speakers type using a "chat alphabet" that has variable spelling.

Datasets

- Flickr8k: images downloaded by Hodosh, Hockenmaier & Young at UIUC in 2009, Turkers wrote captions for them
- Flickr-Speech: Text transcripts read out loud by Turkers
 - <https://groups.csail.mit.edu/sls/downloads/>
 - D. Harwath and J. Glass, “*Deep multimodal semantic embeddings for speech and images*” in IEEE ASRU, Scottsdale, Arizona, USA, December 2015
- Hasegawa-Johnson et al., 2017: throw away the text, keep only the audio.



- A brown and white dog is running through the snow
- A dog is running in the snow
- A dog running through snow
- A white and brown dog is running through a snow covered field
- The white and brown dog is running over the surface of the snow

image2speech system components

- image representation: very large scale convolutional neural net, trained on imagenet classification
- image-to-phone: neural machine translation! Sequence-to-sequence with attention
- phones-to-speech: ClusterGen speech synthesis, audio frames selected using decision trees

Image representation: CNNFEAT \vec{s}_{mn}

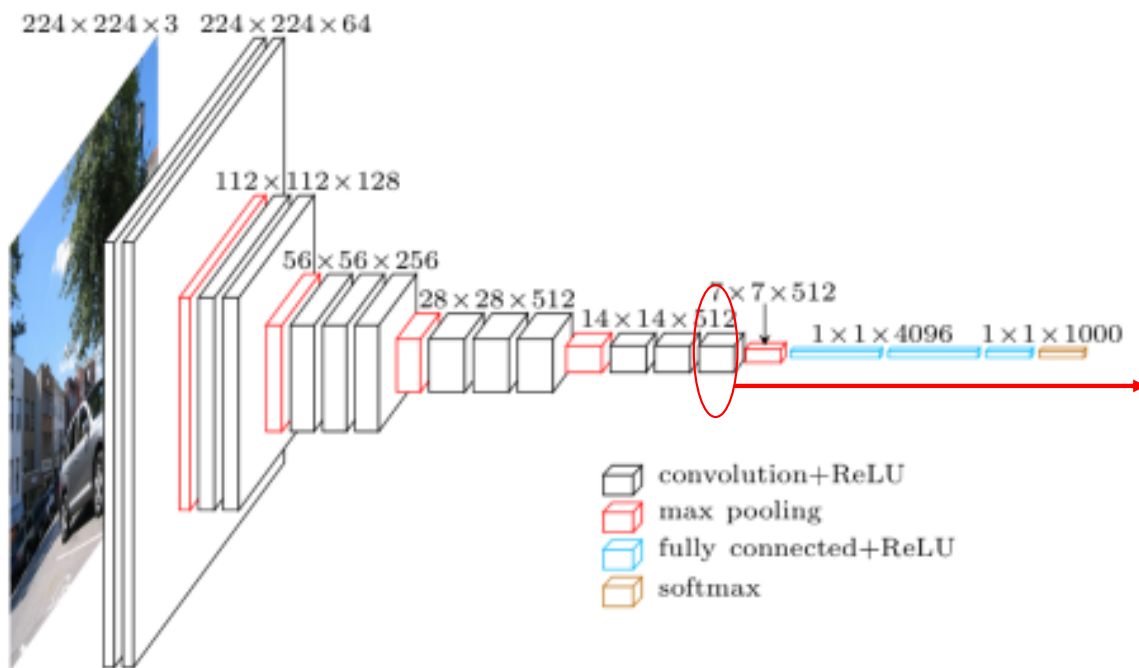
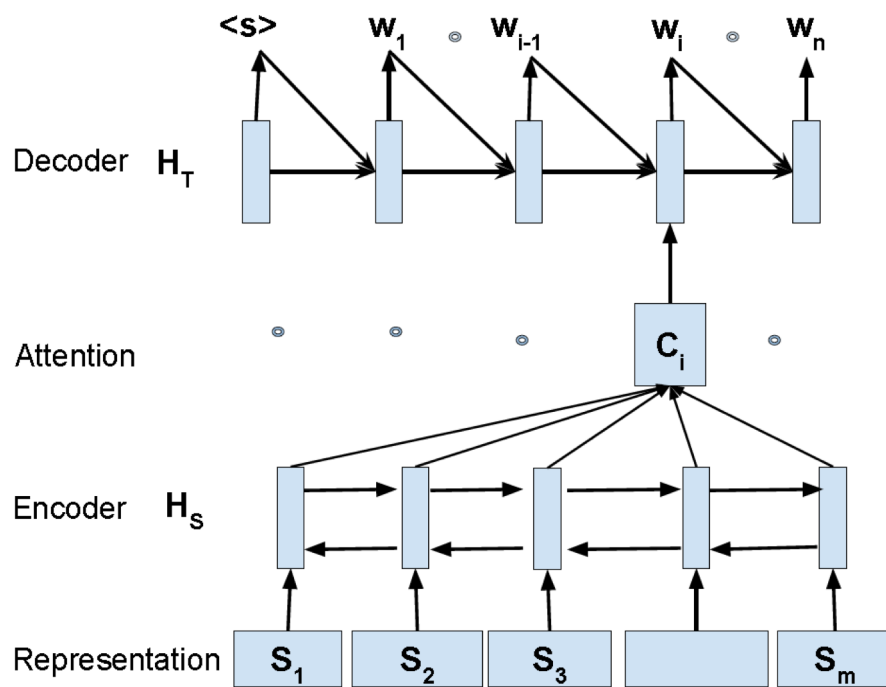


Figure copied from Simonyan & Zisserman, 2014.

- [ImageNet](#) = >500 images/noun of each of the nouns in WordNet.
- [VGG](#) = 13-layer CNN + 2-layer FCN, trained on 14m images, covering the 1000 most numerous nouns, 92.7% top-5 test accuracy.
- **CNNFEAT: 196 feature vectors/image, 512d/vector, from the last CNN layer. Each receptive field covers about 40x40 pixels in the original 224x224 image.**
- VGGFEAT (used later in today's talk, not right now): 1 vector/image, 4096d/vector, from penultimate FCN layer

Images to phonemes = machine translation



- “Representation:” 196 vectors/image
- “Encoder:” PyramidalLSTM with one 128d state vector. Sequence is row-wise raster scan of the image.
- “Attention:” StandardAttender, 128d input, 128d state vector, N hidden nodes
- “Decoder:” MlpSoftmaxDecoder, 3 layers, 1024d hidden vectors
- Output vocabulary: synthetic phones (MSCOCO), force-aligned phones (flickr8k), or acoustic unit discoveries (both)

Figure copied without permission from Duong, Anastasopoulos, Chiang, Bird & Cohn, NAACL-HLT 2016.

Phones to speech = "TTS without the T"

ZeroSpeech 2019

News
Tasks and intended goal
Getting started
Results

ZeroSpeech 2019: TTS without T

Task and intended goal

Young children learn to talk long before they learn to read and write. They can conduct a dialogue and produce novel sentences, without being trained on an annotated corpus of speech and text or aligned phonetic symbols. Presumably, they achieve this by recoding the input speech in their own internal phonetic representations (proto-phonemes or proto-text), which encode linguistic units in a speaker-invariant representation, and use this representation to generate output speech.

Duplicating the child's ability would be useful for the thousands of so-called low-resource languages, which lack the textual resources and/or linguistic expertise required to build a traditional speech synthesis system. The ZeroSpeech 2019 challenge addresses this problem by proposing to **build a speech synthesizer without any text** or phonetic labels, hence *TTS without T* (text-to-speech without text). We provide **raw audio** for the target voice(s) in an unknown language (the *Voice Dataset*), but **no alignment, text or labels**. Participants must discover **subword units** in an unsupervised way (using the *Unit Discovery Dataset*) and align them to the voice recording in a way that works best for the purpose of **synthesizing** novel utterances from novel speakers (see [Figure 1](#)).

The ZeroSpeech 2019 is a continuation and a logical extension of [the sub-word unit discovery track of ZeroSpeech 2017](#) and [ZeroSpeech 2015](#), as it demands of participants to discover such units, and then evaluate them by assessing their performance on a novel speech synthesis task. We provide a baseline system which performs the task using two off-the-shelf components: (1) a system which discovers discrete acoustic units automatically in the spirit of Track 1 of the Zero Resource Challenges 2015 [1] and 2017 [2], and (2) a standard TTS system.

A submission to the challenge will replace at least one of these systems. Participants can construct their own **end-to-end system** with the joint objective of discovering sub-word units and producing a waveform from them. Participants can, alternatively, make use of one of the two baseline systems, and improve the other. The challenge is therefore open to **ASR-only systems** which make a contribution primarily to unit discovery, focusing on improving the embedding evaluation scores (see [Evaluation metrics](#)). Vice versa, the challenge is open to **TTS-only systems** which concentrate primarily on improving the quality of the synthesis on the baseline sub-word units. (All submissions must be complete, however, including both resynthesized audio and the embeddings used to generate them.)

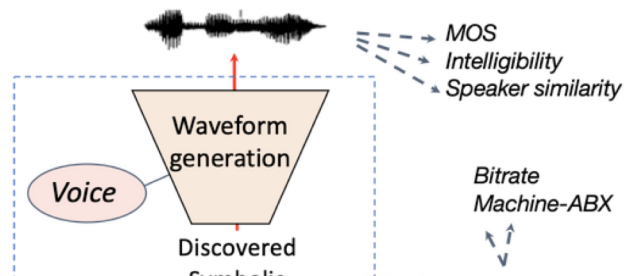
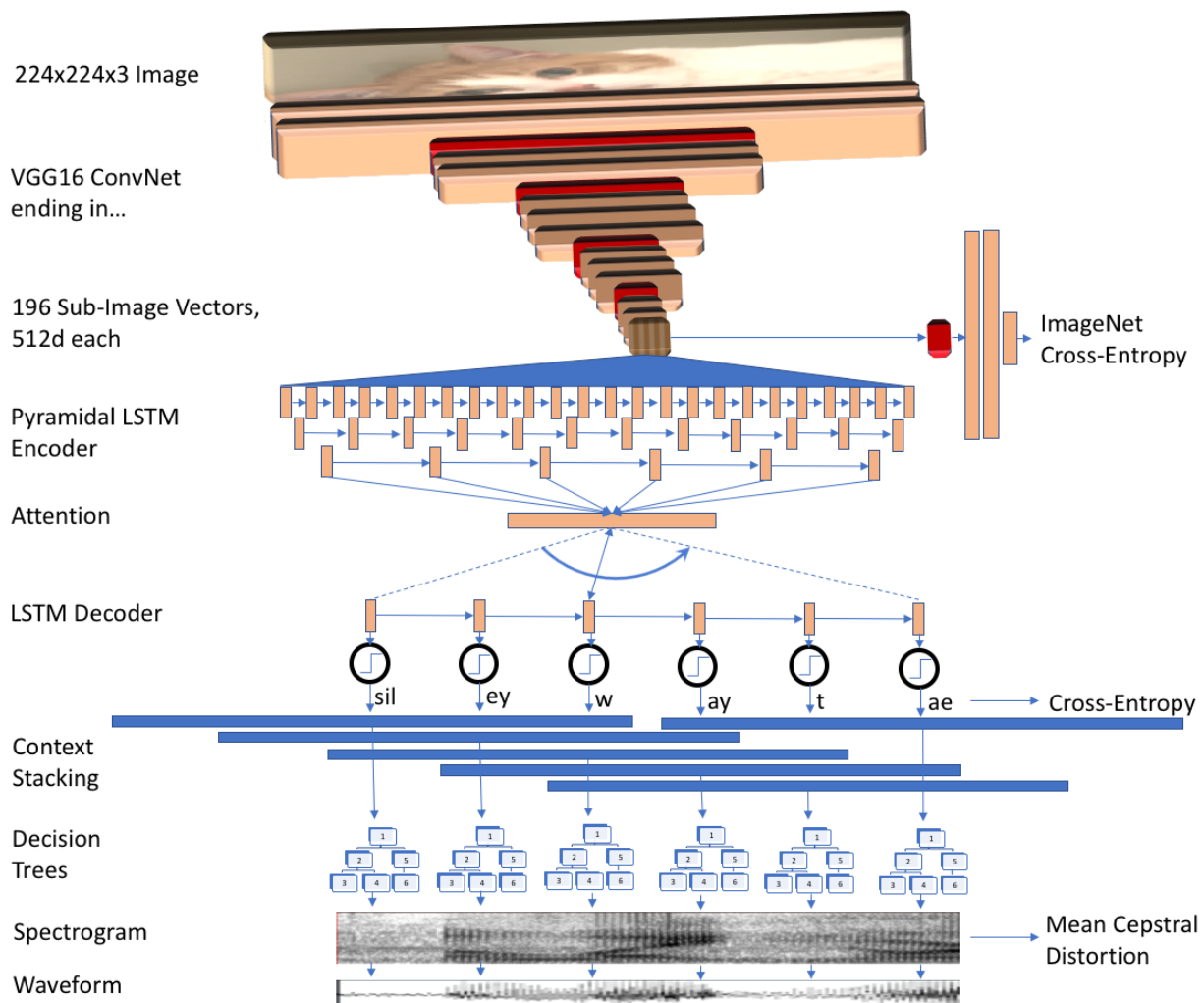
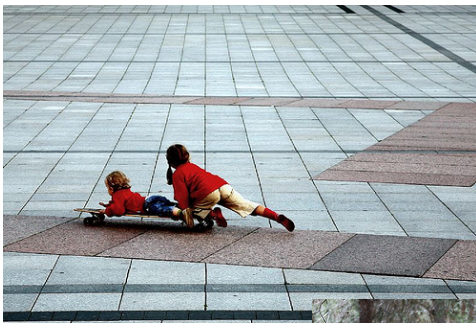


image2speech system overview



flickr8K: American phones



● Reference 1: “The boy +um+ laying face down on a skateboard is being pushed along the ground by +laugh+ another boy.”



● Reference 2: “Two girls +um+ play on a skateboard +breath+ in a court +laugh+ yard.”

● Hypothesis (128d attender): SIL +BREATH+ SIL T UW M EH N AA R R AY D IX NG AX R EH D AE N W AY T SIL R EY S SIL



● Hypothesis (64d attender): SIL +BREATH+ SIL T UW W IH M AX N W AO K IX NG AA N AX S T R IY T SIL



● Reference 1: “A boy +laugh+ in a blue top +laugh+ is jumping off some rocks in the woods.”

● Reference 2: “A boy +um+ jumps off a tan rock.”



● Hypothesis (128d attender): SIL +BREATH+ SIL EY M AE N IH Z JH AH M P IX NG IH N DH AX F AO R EH S T SIL



● Hypothesis (64d attender): SIL +BREATH+ SIL EY Y AH NG B OY W EY R IX NG AX B L UW SH ER T SIL IH Z R AY D IX NG AX HH IH L SIL



Images and Reference Texts: Hodosh, Young & Hockenmaier, 2013.
Waveforms: Harwath and Glass, 2015

Outline

- Human speech processing
- Modeling the ear: Fourier transforms and filterbanks
- Speech-to-text-to-speech (S2T2S)
 - Hybrid DNN-HMM systems (*)
 - Recurrent neural nets (RNNs): Connectionist Temporal Classification, Trajectory nets
 - Sequence-to-sequence RNNs with attention
- Languages other than English
 - Training and testing on 300 languages
 - Transfer learning: from languages with data, to languages without
 - Dialog systems for unwritten languages
- **Distorted speech**
 - **Motor disability**
 - **Second-language learners**

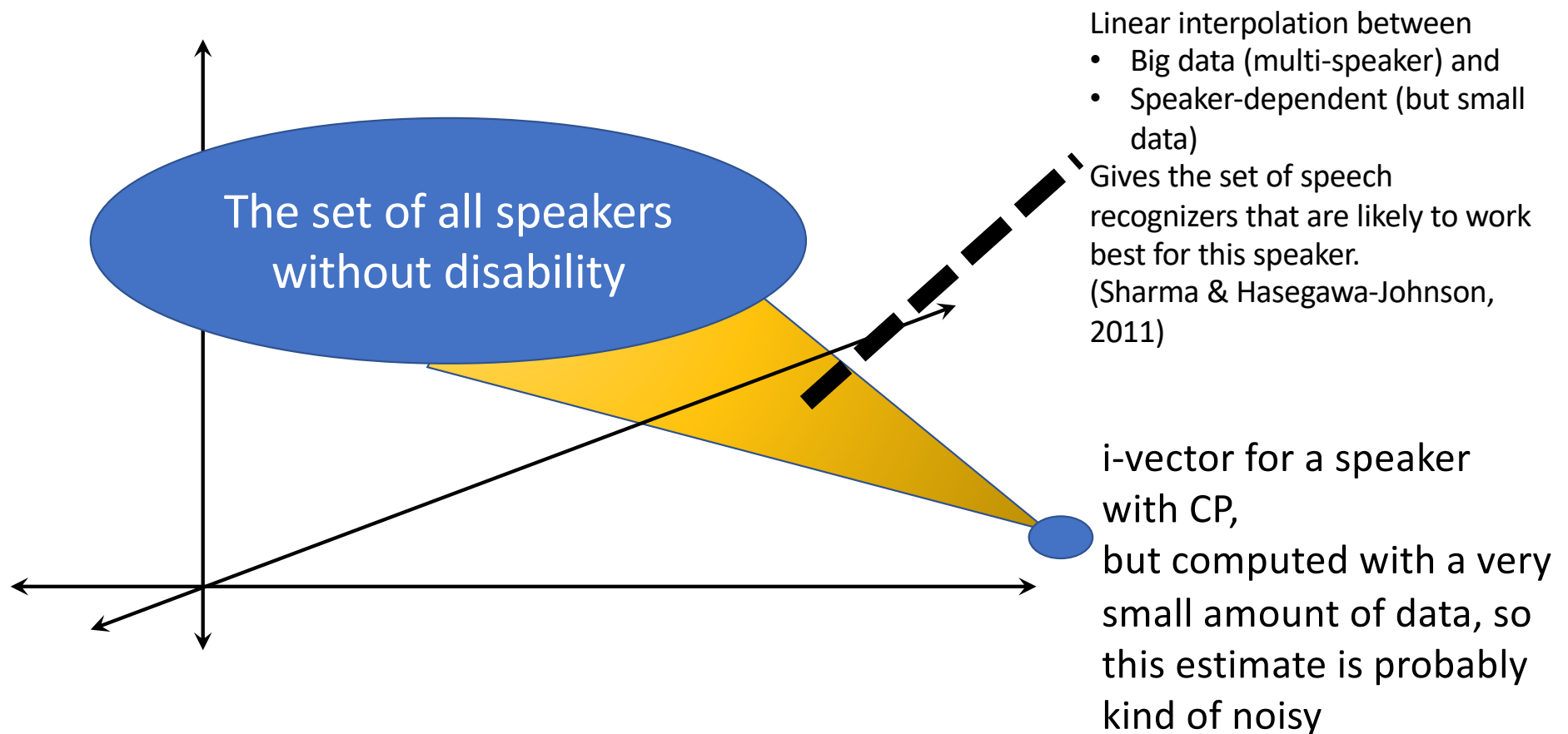
Types of motor disability

- Diseases you're born with, e.g., Cerebral Palsy
 - It's hard to walk, to hold a pencil, or to type on a computer
 - Speech recognition is often faster than typing, but...
 - Speech is also distorted, because it's hard to speak clearly
- Diseases that get worse over time, e.g., Parkinson's Disease
 - In the early and mid stages of the disease, speech is still perfectly intelligible, but...
 - Speech recognition stops working. For example, you've been using speech input on your cell phone all your life, but lately it's been working less and less well.

Speaker adaptation of neural nets

- Stack up all of the network weights in a vector w , then set $w = Tv + w_0$
- Actually nobody does that, because the weight vector is too huge, e.g., 5 million trained weights. Methods that have been used instead:
 - Gaussian i-vector: 40k-dimensional Gaussian supervector, reduced to 300-d i-vector, used as input to a neural net
 - Neural i-vector: 5000-d hidden node activation vector, averaged over all speech of that talker, then reduced to 300-d i-vector
 - Auxiliary network: trained to estimate the way a new speaker's voice has shifted the hidden node activations
- 5-million-d weight vector might be useful with a large enough training database. Then the i-vector, v , might be 300-d

Adaptation to speakers with disability



Outline

- Human speech processing
- Modeling the ear: Fourier transforms and filterbanks
- Speech-to-text-to-speech (S2T2S)
 - Hybrid DNN-HMM systems (*)
 - Recurrent neural nets (RNNs): Connectionist Temporal Classification, Trajectory nets
 - Sequence-to-sequence RNNs with attention
- Languages other than English
 - Training and testing on 300 languages
 - Transfer learning: from languages with data, to languages without
 - Dialog systems for unwritten languages
- Distorted speech
 - Motor disability
 - **Second-language learners**

Automatic pronunciation scoring

- $P(\text{audio} | \text{native speaker})$ vs. $P(\text{audio} | \text{non-native})$
 - Might work well if we had lots of data from non-native speakers
- Goodness of pronunciation (GoP):
 - $P(\text{audio} | \text{correct transcription}) / P(\text{audio} | \text{arbitrary phone sequence})$

Landmark-based pronunciation scoring

(Yoon, Sproat & H-J, 2010)

- Identify the landmarks, first
- Score whether each landmark was correctly vs. incorrectly pronounced
- Use information about the speaker's native language

Conclusions: research opportunities

- Take advanced courses
 - Audio enhancement: CS 598PS
 - Speech and video recognition & synthesis: ECE 417
- Download a software development recipe
 - Hybrid DNN-HMM: Kaldi (kaldi-asr.org)
 - Sequence-to-sequence with attention: XNMT (<https://github.com/neulab/xnmt>)
- Create your own startup company!!
 - Dialog systems, e.g., for smart glasses, dishwashers, and everything else
 - Audio search for domains that don't work yet (rap music?)
- Join a research group
 - Landmark-based ASR in 300 languages; dialog system for unwritten languages
 - Speech interface for disability, or for second-language learners