Speech Processing

Karen Livescu

TOYOTA TECHNOLOGICAL INSTITUTE AT CHICAGO

Toyota Technological Institute at Chicago

- Independent academic graduate institute for computer science located on U. Chicago campus
- PhD program with a focus on ML, algorithms, AI
- Currently:
  - 12 tenure-track/tenured faculty
  - 11 research faculty (3-year endowed appointment)
  - 35 PhD students
- Visiting students, adjoint faculty, ...

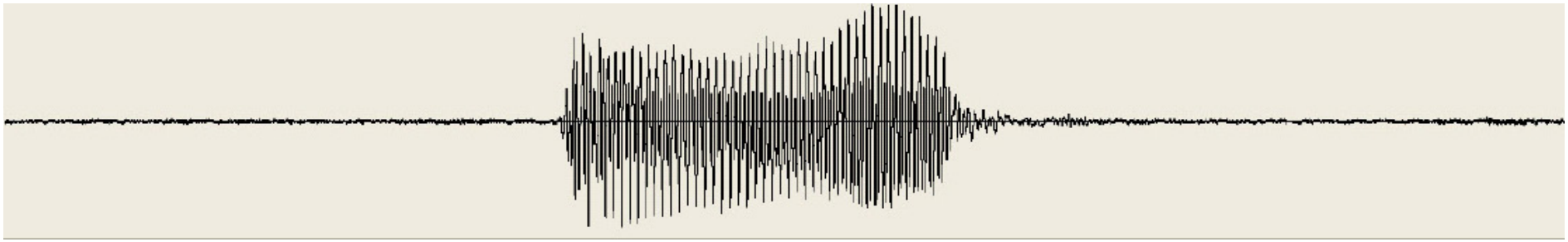


Plan for this tutorial

- Most of today: Intro to speech, historical tour of speech recognition research, speech recognition with hidden Markov models
- Most of tomorrow: Speech recognition with recurrent neural networks, representation learning for speech
- Lab exercise: Speech signals, speech recognition with HMMs and RNNs

Speech and text

This is a speech signal: ⟵ This is some text



Some differences between text and speech:

- Speech is continuous-valued, text is discrete
- Speech is continuous in time: Words/sounds are not separated

Speech and text processing also have much in common...

- Sequence prediction problems
- Many of the same algorithms
- Many researchers work on both
- Both involve mapping between sequence observations with great **variability** and underlying **meaning**

Variability in text

There are many ways of expressing the same semantic information via text

- I recommend this mattress. Very firm and supportive. Great night's sleep.
- Nice mattress. Firm.
- This is the best mattress I have ever had. It is a perfect combination of firmness and support. I have never slept better. ...

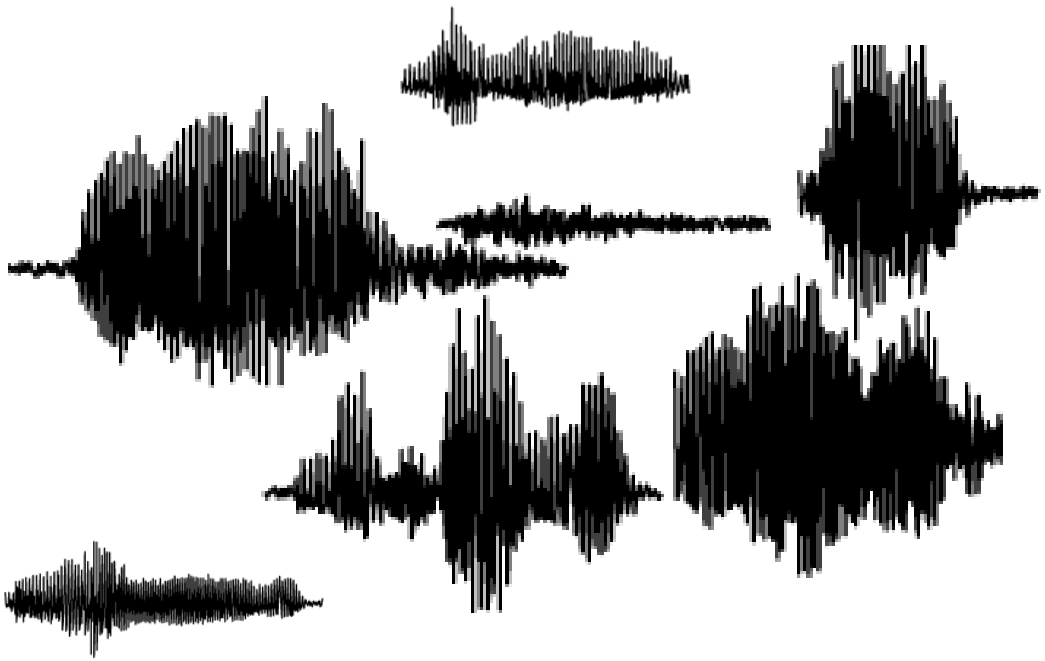
Variability in text

Text can be more or less formal. Informal text has many more variants.

- haha
- hahahahahahahaha
- haaaahaaaa
- lol
- rotflmao
- lol!!!!!!!!!!!!!!
- wow that is big
- that is biiiiiig
- that. is. big.
- waaaaaaay big

Variability in speech

Spoken words have even more variants: pronunciation, speaker, acoustic environment, mood, state of inebriation...



Variability makes it hard to determine the meaningful content

Speech technologies in the news

(This is a bit overly optimistic)

Commercial speech technologies

Some commercial success

- Personal assistants: Alexa, Siri, Google Home, Cortana, ...
- Skype Translator
- YouTube closed captioning
- Voicemail transcription

But

- Accuracies of these services are not great
- Performance varies for speakers of different ages, dialects, non-native accents, ...
- Poor coverage of lower-resource languages
- Lots of other tasks to solve

Examples of speech technologies

Automatic speech recognition (a.k.a. "speech-to-text")

Input: acoustic waveform $a(t)$

Output: word string w

$w =$ "one two three four five"

Speaker/language identification

Input: acoustic waveform $a(t)$

Output: speaker/language c

$c =$ "Karen"/"English"

Speech understanding: Output is a "meaning representation"

Speech technologies (cont'd)

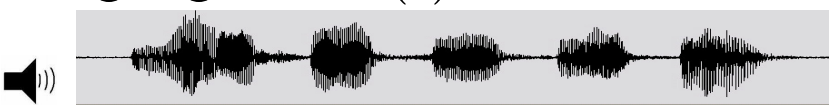
Speech synthesis:

Input: word string w
$w =$ "one two three four five"

Output: acoustic waveform $a(t)$

Speech translation:

Input: acoustic waveform in Language 1, $a_1(t)$



Output: acoustic waveform in Language 2, $a_2(t)$

Other: keyword search, speaker diarization, dialogue systems, summarization, language instruction/assessment, voice morphing, denoising, speaker separation, medical diagnosis ...

Rest of this lecture: Focus on speech recognition

Automatic speech recognition (a.k.a. "speech-to-text")

Input: acoustic waveform $a(t)$

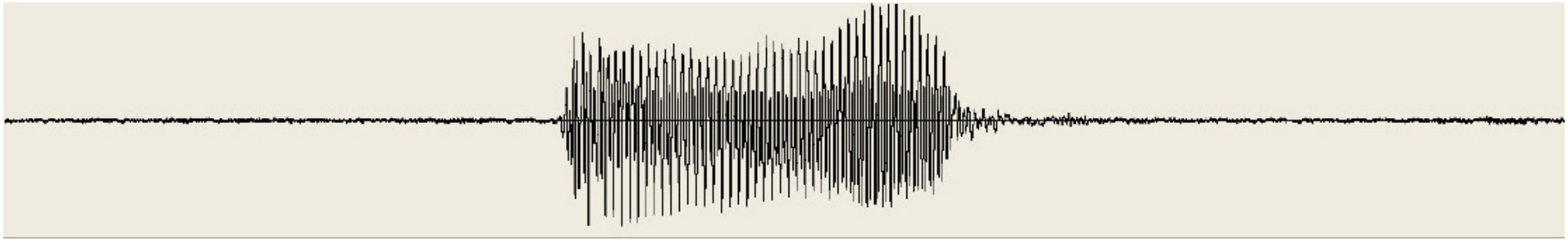
Output: word string w

$w =$ "one two three four five"

- By far the most well-studied speech task
- Involves many of the same techniques as other tasks

A "simple" speech task: Single-digit classification

This is a 1-second speech waveform. Which digit (0-9) was spoken?

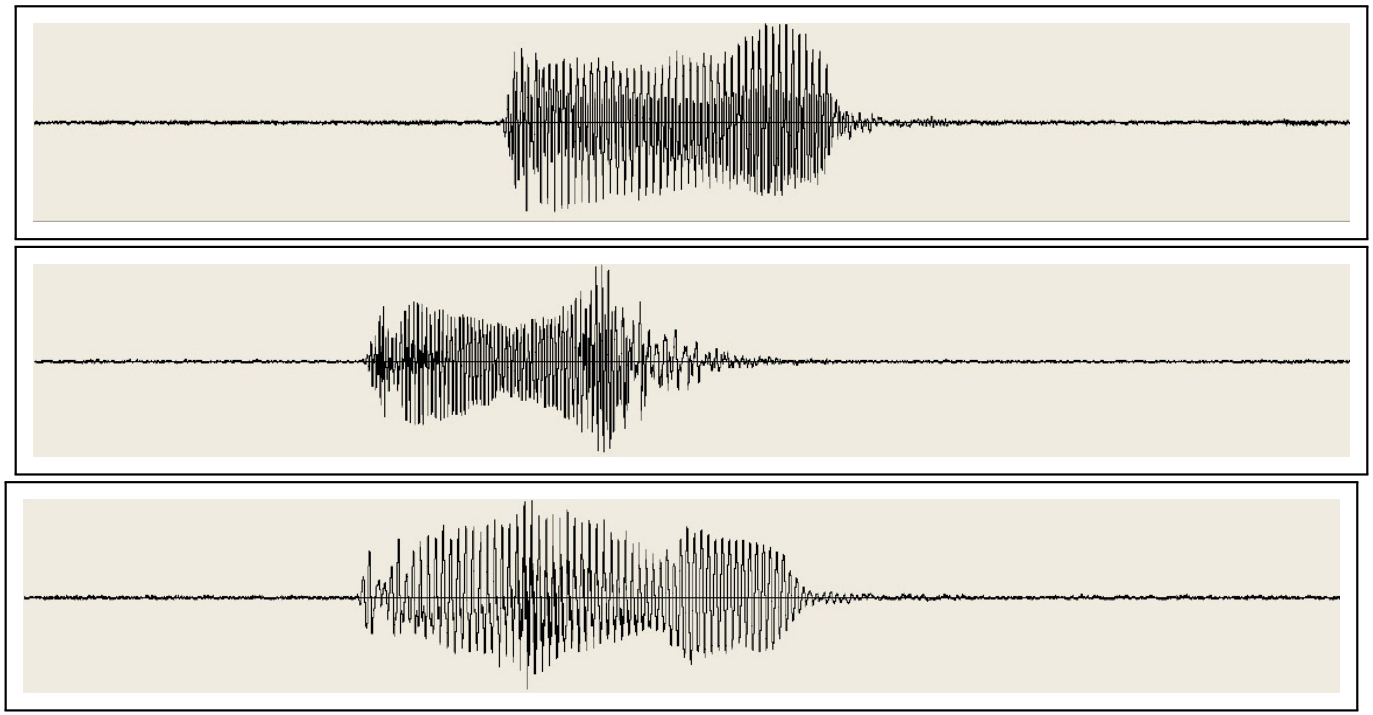


What are we looking at?

- Recording from a microphone: instantaneous air pressure vs. time
- Discretized in time (in this case, to 16,000 samples, i.e. sampling rate of 16kHz)
- Discretized in magnitude (in this case, to 16 bits per sample)
- Result: 16,000-dimensional vector,
  e.g. $a(t) = [3, 16, -1, 0, 427, 29, \ldots]$

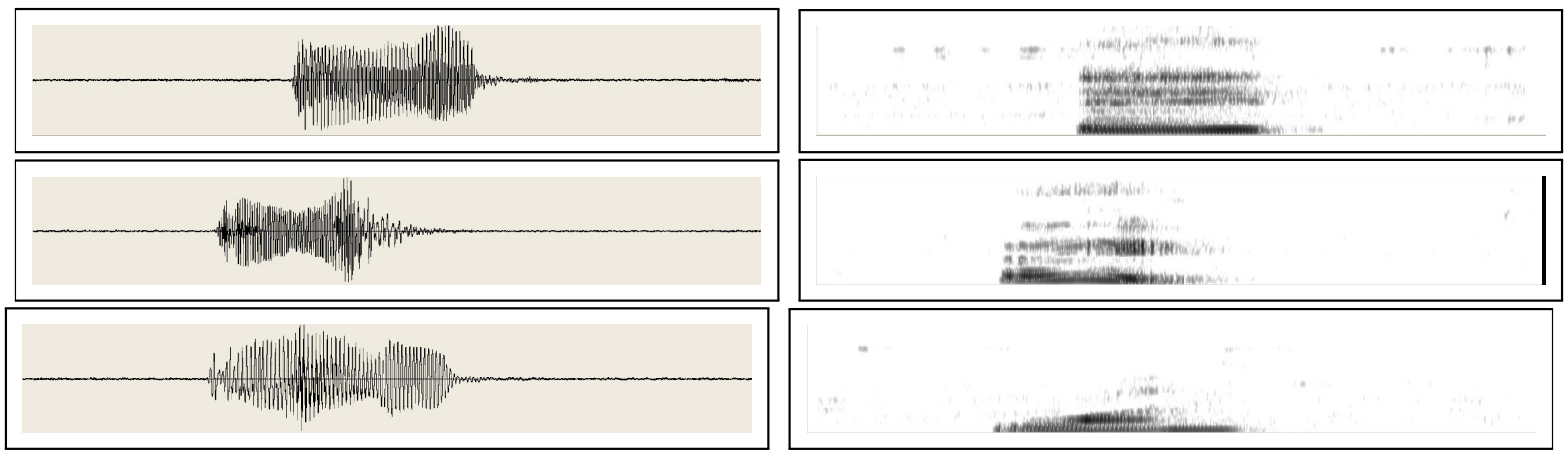
This is hard!

Slightly easier problem: Which two are the same digit?



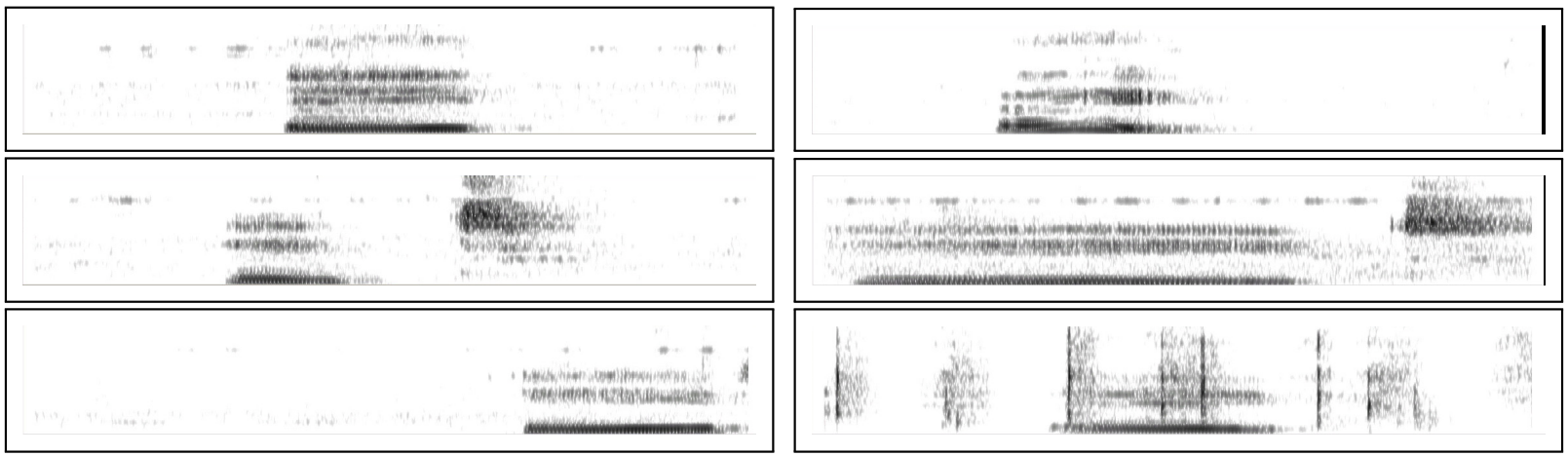
Idea: Use a frequency-domain representation

Spectrogram: Plot of energy at each frequency over time



This is still hard!

Several examples of the digit “eight”

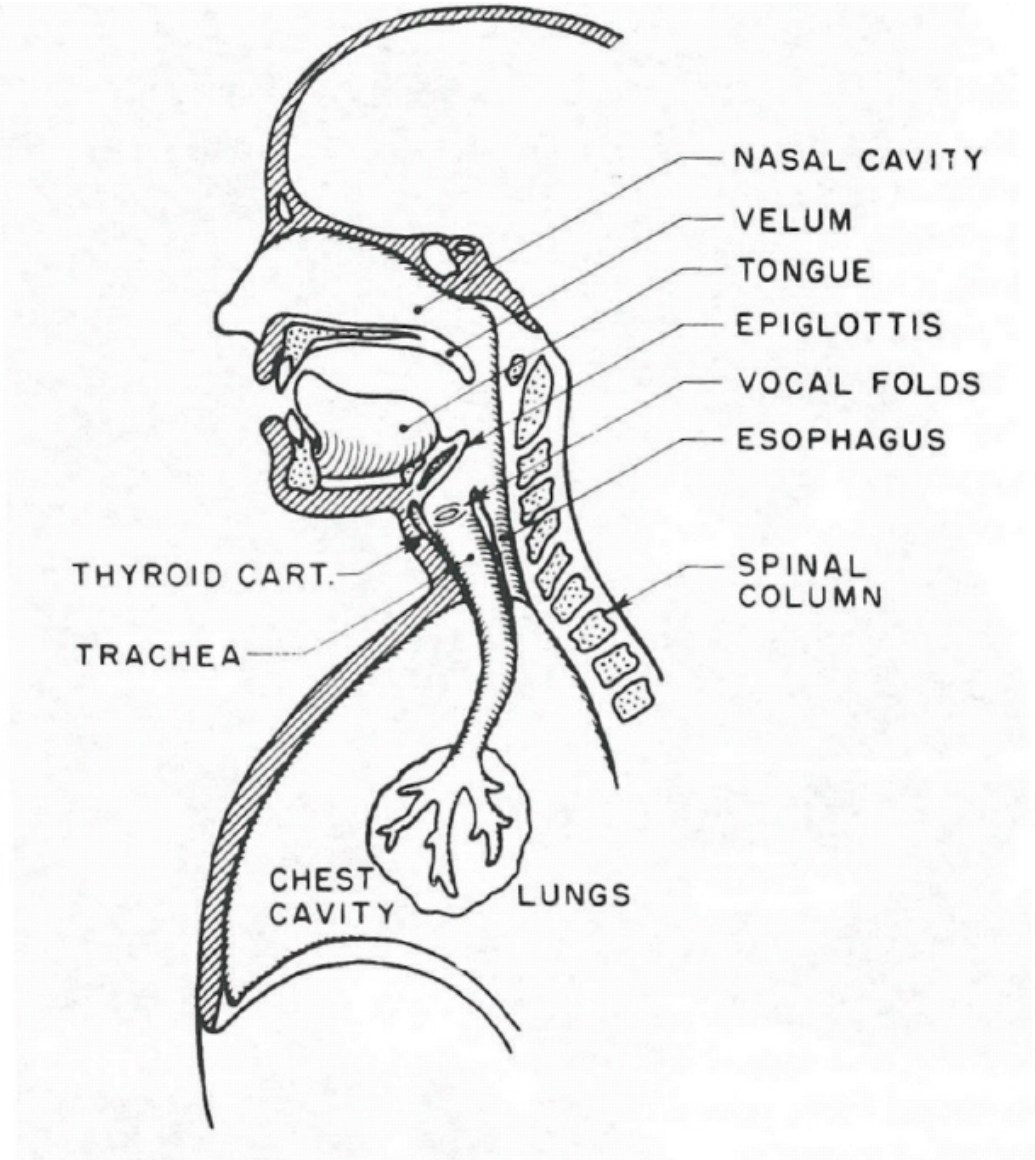


Is there any representation (acoustic features) of a speech signal that makes it easier to recognize the content?

Sources of variation in speech

acoustic:	channel, noise, vocal tract differences, pitch
phonetic:	*eight* $\longrightarrow$ [ey tcl] vs. [ey tcl t]
phonological:	*eight* before vowel $\longrightarrow$ [ey dx] *gas shortage*, *fish sandwich*
dialect:	*either* $\longrightarrow$ [iy dh er] vs. [ay dh er] *pin, pen*; *Mary, marry, merry*
coarticulation:	*she, shoe*
situational context:	*it is easy to recognize speech* vs. *it is easy to wreck a nice beach*

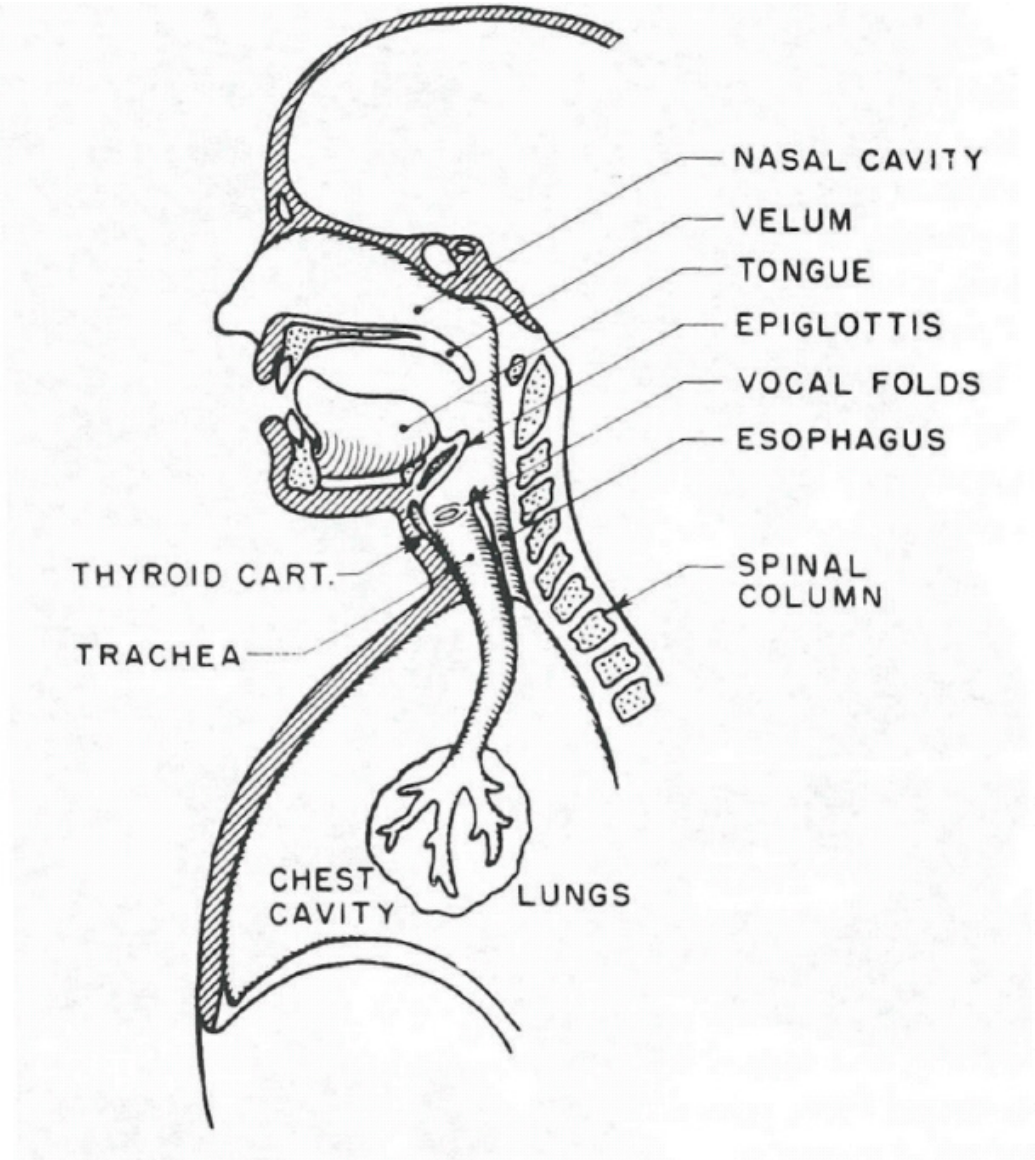
Speech production



(fig. from [Flanagan 1972])

- Air pushed up from lungs through vocal folds (glottis)
- Vocal folds: tensed for voiced sounds, spread for voiceless
- Tongue, lips, velum, nasal cavity form a "resonance chamber"

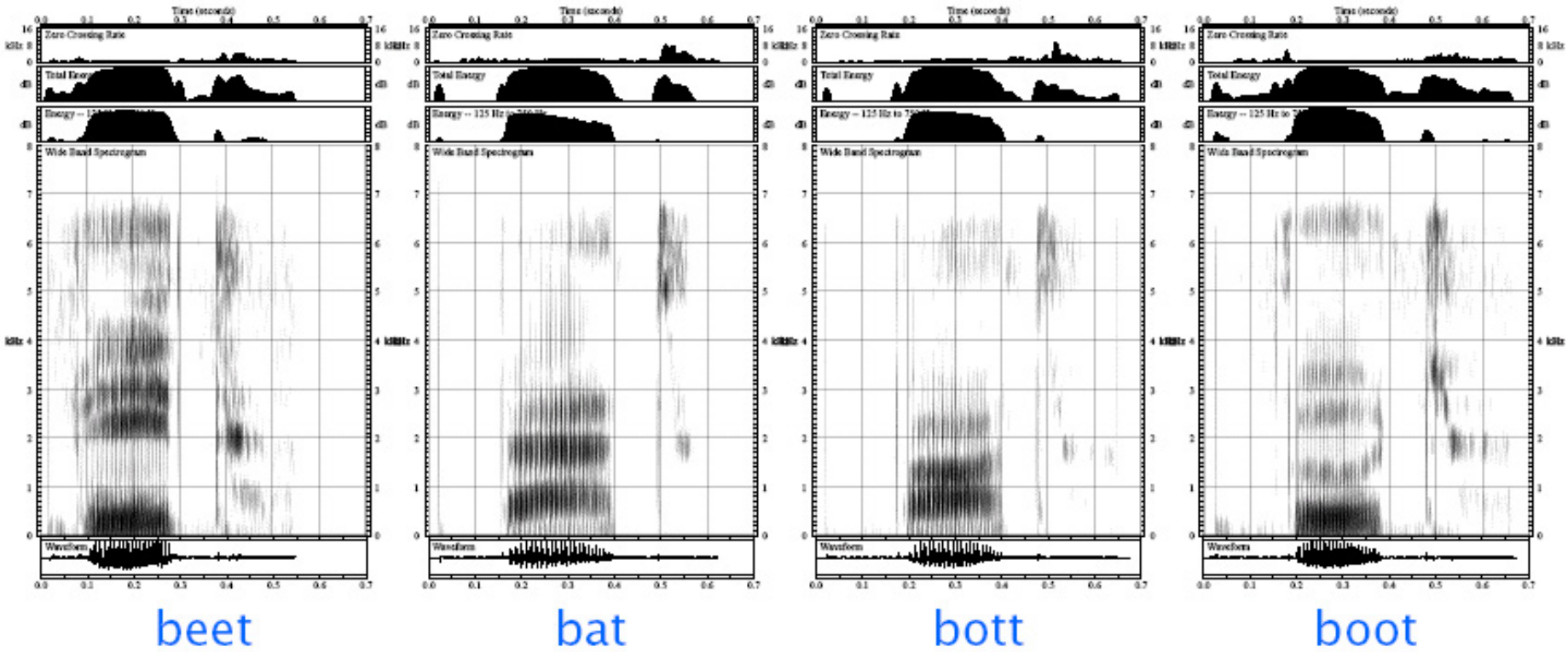
Speech production



(fig. from [Flanagan 1972])

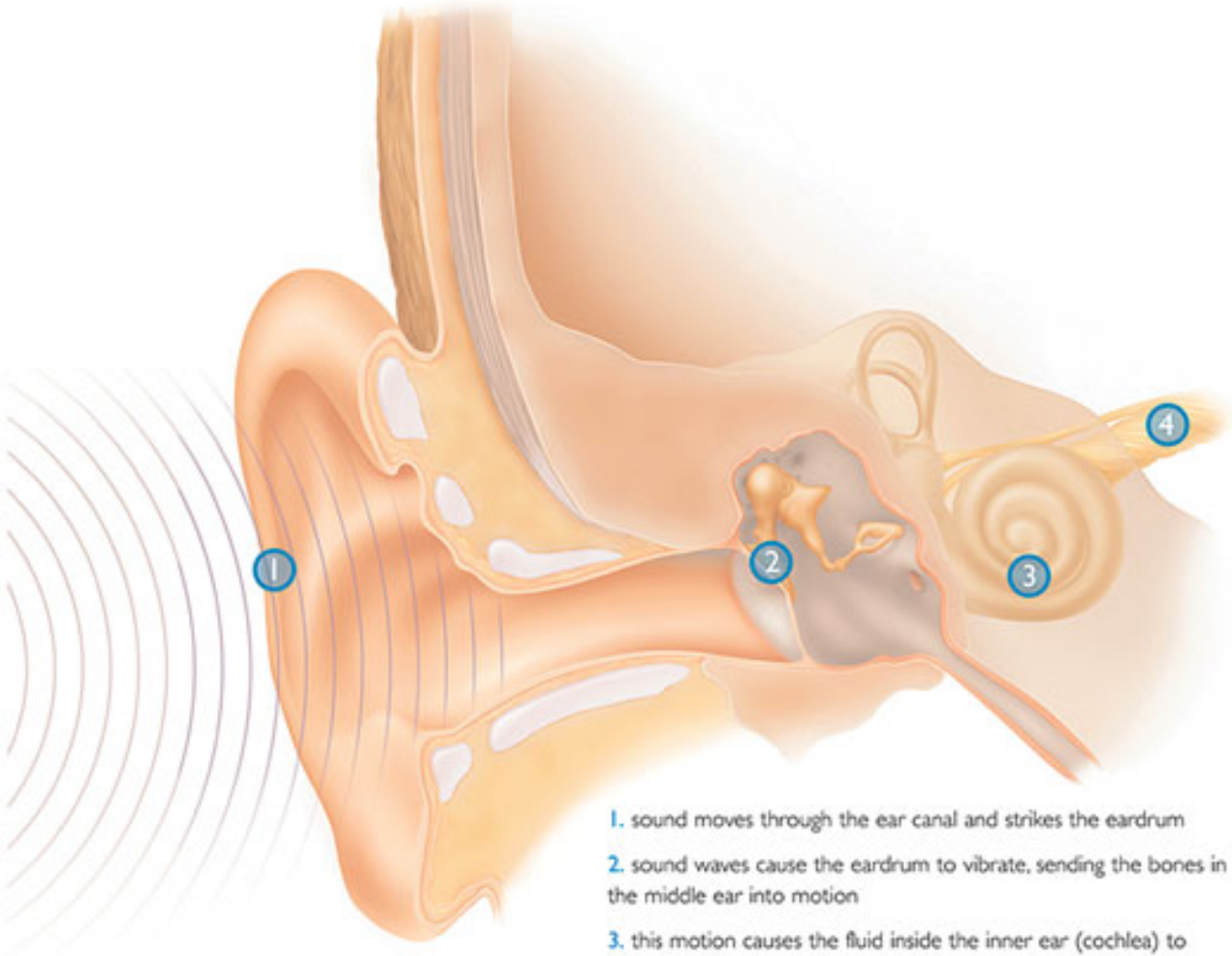
- *Source-filter model*: Vocal tract acts as a filter, modulating the spectrum of the source signal
- Source is air pressure waveform either from glottis (e.g. vowels) or from another constriction (most consonants)

Spectrograms of several vowels



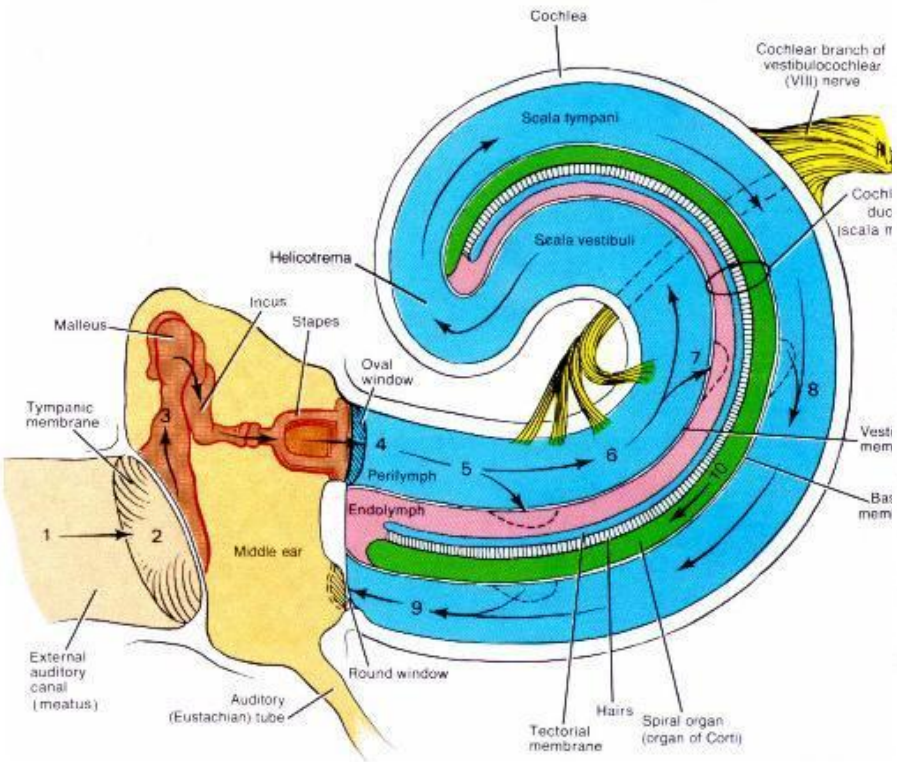
(figs. from MIT 6.345 Spring '03, OpenCourseWare `http://ocw.mit.edu`)

Physiology of hearing



1. sound moves through the ear canal and strikes the eardrum
2. sound waves cause the eardrum to vibrate, sending the bones in the middle ear into motion
3. this motion causes the fluid inside the inner ear (cochlea) to move the hair cells
4. hair cells change the movement into electric impulses, which are sent to the hearing nerve into the brain; you hear sound

Physiology of hearing (cont'd)



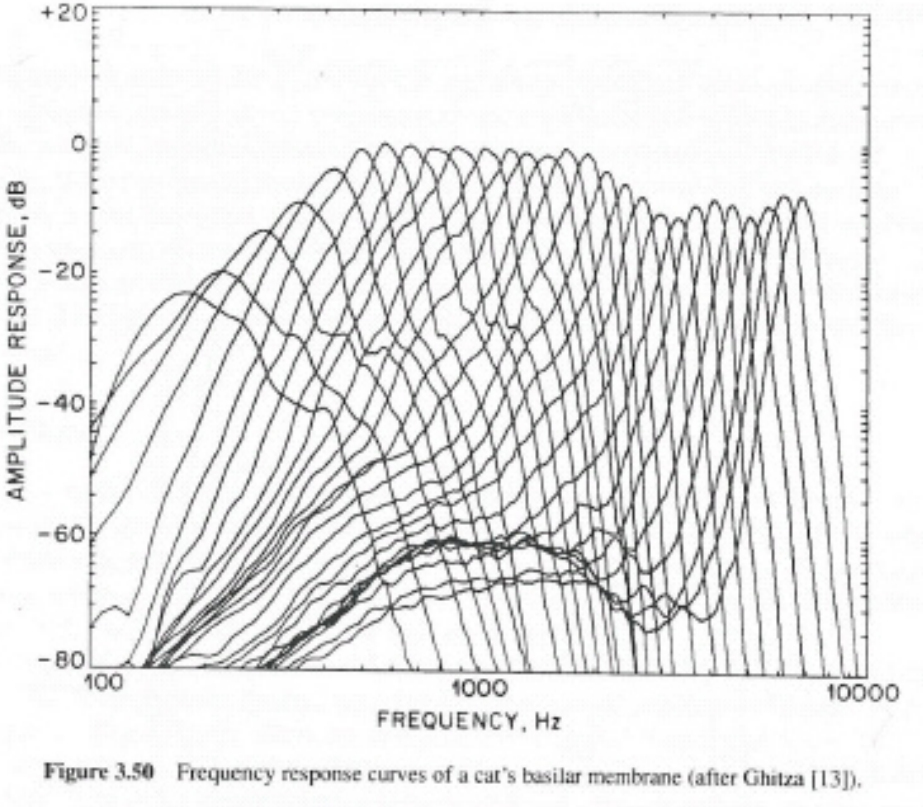


Figure 3.50 Frequency response curves of a cat's basilar membrane (after Ghitza [13]).

- Hairs on basilar membrane have different frequency responses

Perception

- Humans are less sensitive to differences in frequency at high frequencies than at low frequencies
- I.e., our internal "frequency axis" is not linear
- Stevens and Volkmann (1972) measured this warping with a set of perceptual experiments
- Result is the mel scale: $f_{\text{mel}} = 2595 \log_{10}(1 + f/700)$

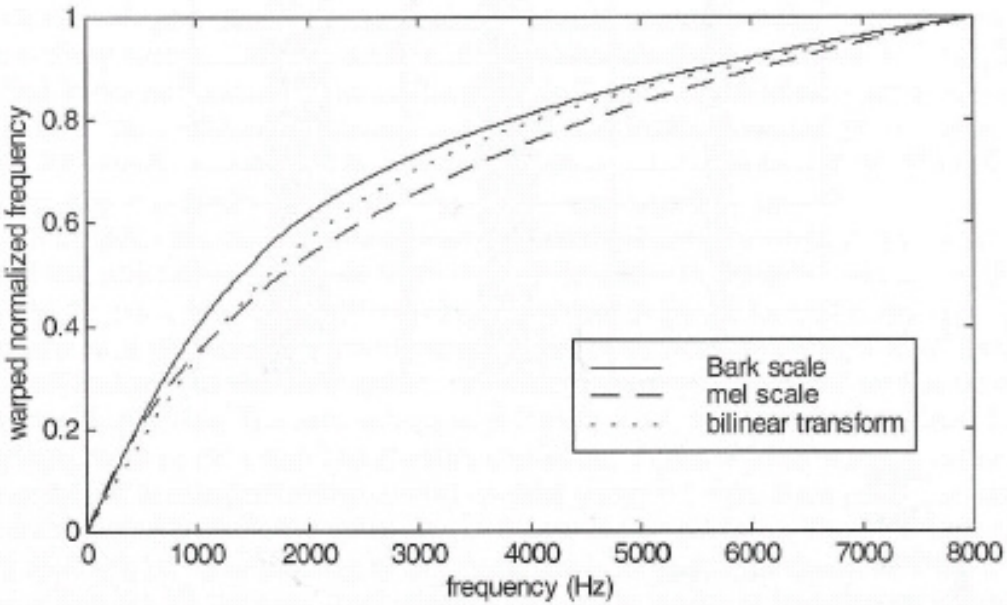
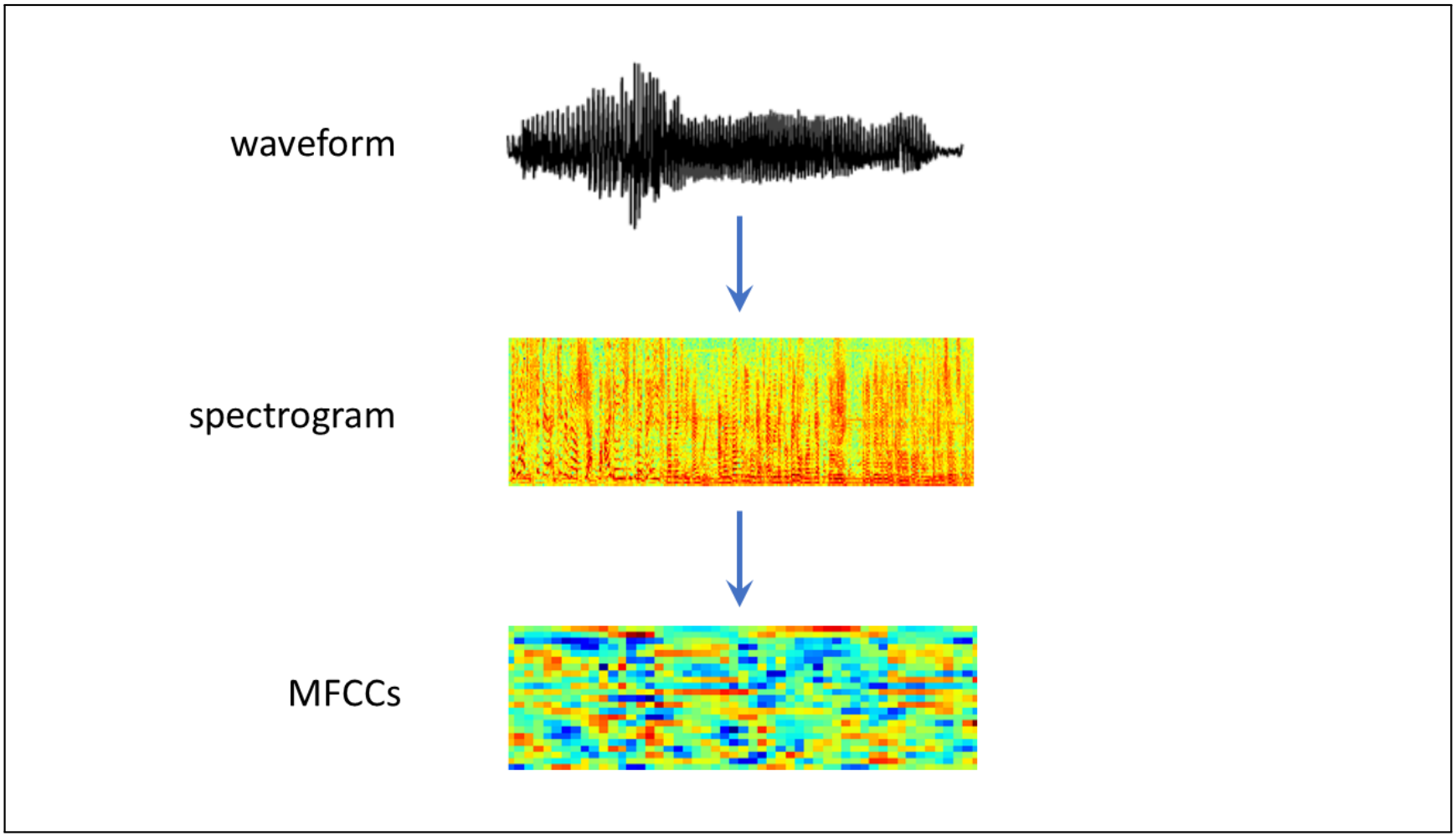


Figure 2.13 Frequency warping according to the Bark scale, ERB scale, mel-scale, and bilinear transform for $\alpha = 0.6$: linear frequency in the x-axis and normalized frequency in the y-axis.

Typical representations (features) for speech



Acoustic features in the neural network era

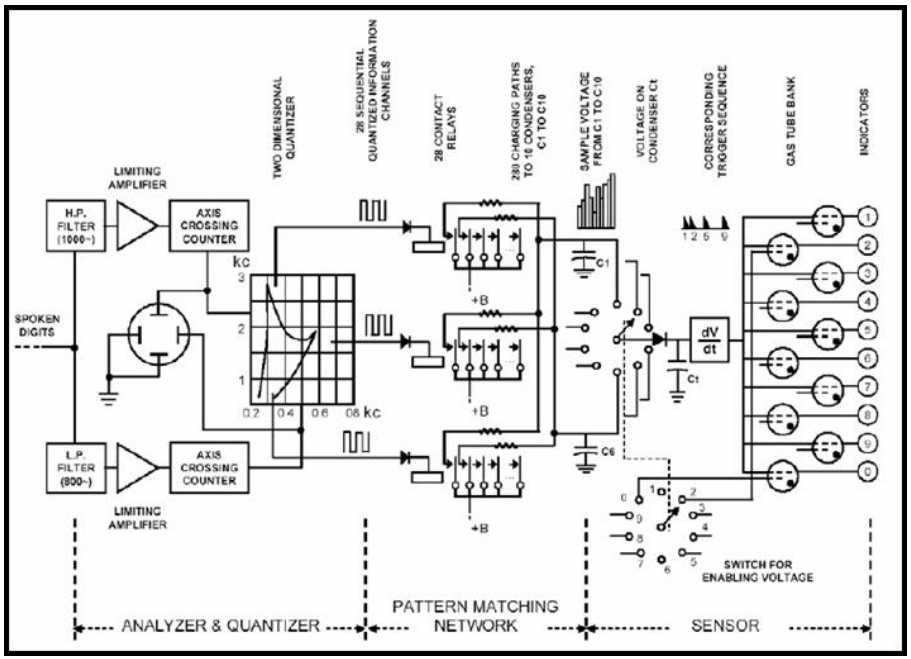
In recent neural network-based research, 3 types of acoustic features are common:

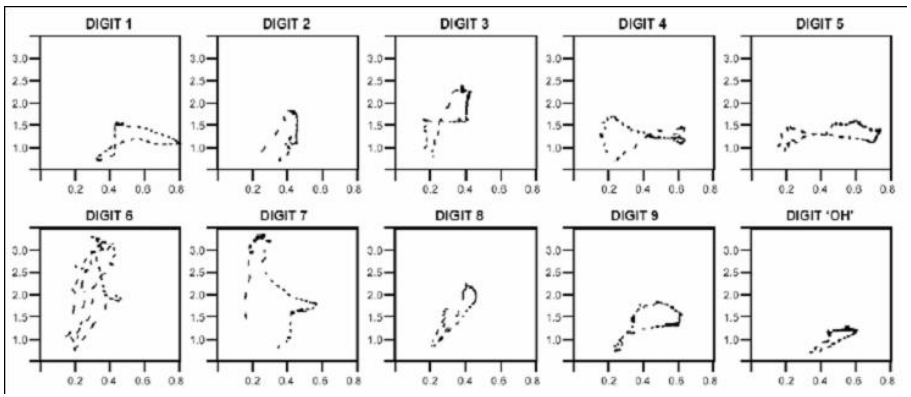
- No signal processing, just use the raw signal: Works best with extremely large amounts of data
- (Lightly post-processed) spectrogram: Works in many typical settings
- Traditional signal processing-based features (e.g. MFCCs): Work best in low-data settings

1922: The first ASR system?

Radio Rex (1922):
Toy dog responded to high energy signal around 500Hz (as in the vowel in “Rex”)

1950s

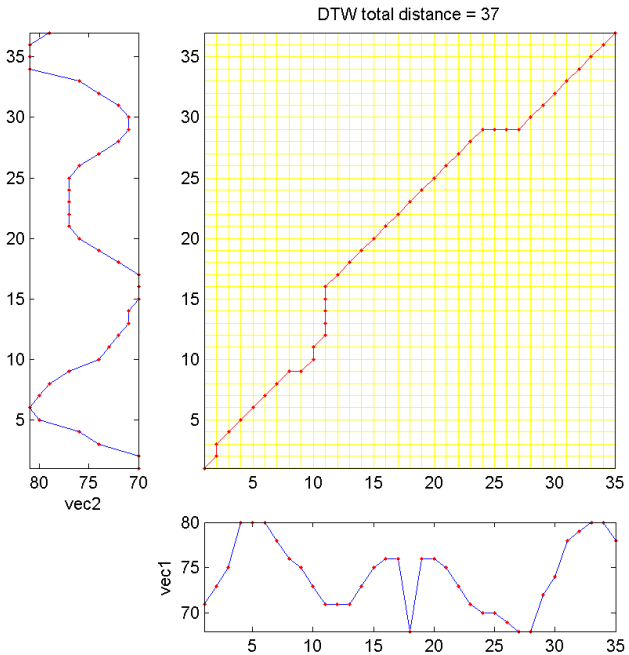




(fig. from [Davis+ 1952])

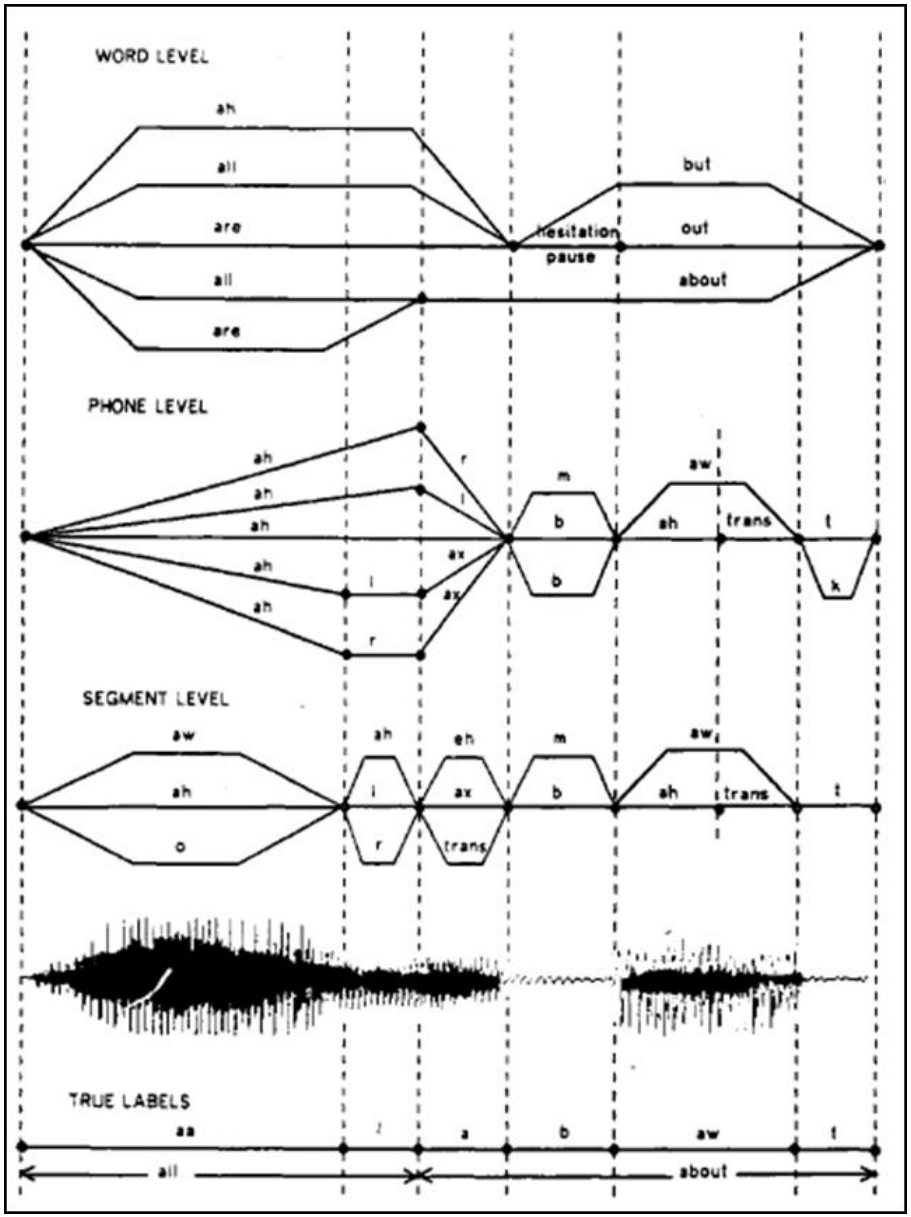
- Small-vocabulary isolated-word/ phoneme recognition
- Fixed-length feature vectors of spectral/ad hoc measurements
- Nearest-neighbor classification

1960s



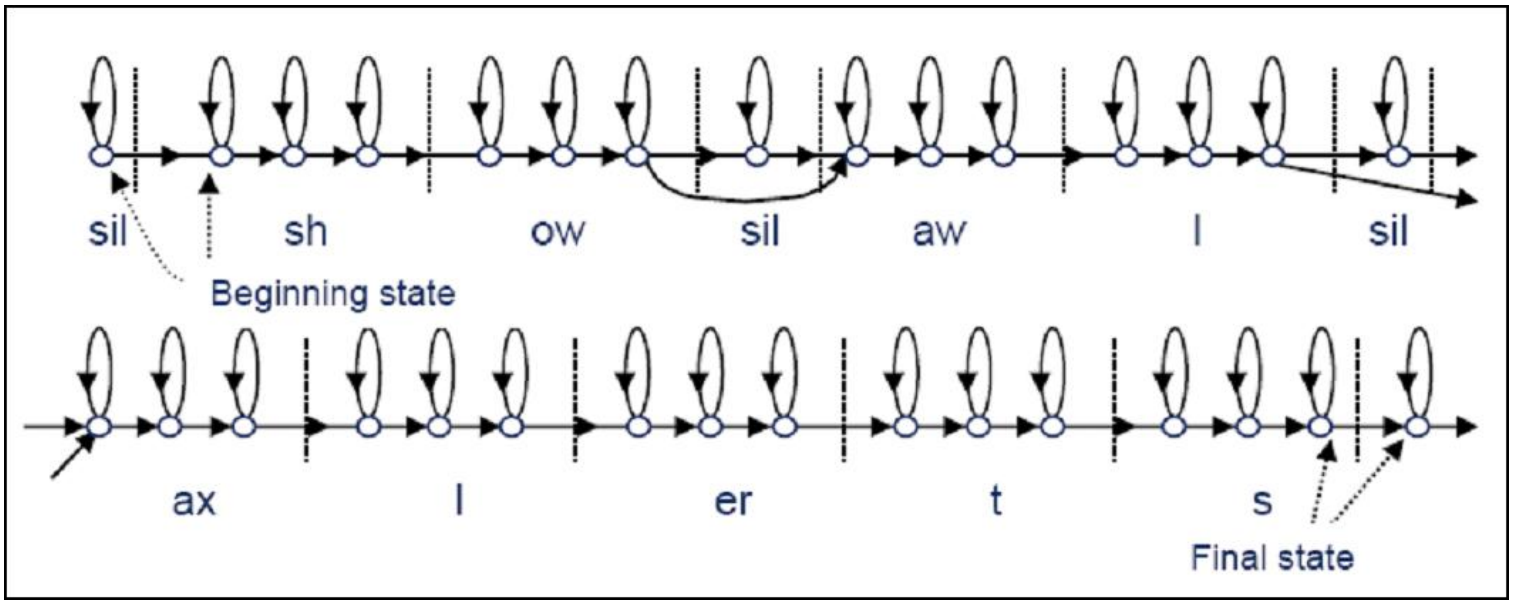
- Dynamic time warping is introduced, allowing for different-length test and reference sequences for isolated word/phoneme recognition
- [Vintsyuk 1968, Sakoe and Chiba 1978]

1970s



- U.S. ARPA funds larger-vocabulary, continuous-speech recognition research
- Finite-state machines, graph search algorithms, statistical models trained on data emerge
- [Jelinek+ 1975, Reddy+ 1976, Juang+ 1985]

1980s



- Hidden Markov model (HMM)-based approaches
- Gaussian mixture observation densities
- Learning from data via expectation-maximization algorithm
- First neural network-based approaches (!)
- [Baum 1972, Levinson+ 1983, Juang & Rabiner 2005, Bourlard & Morgan 1994]

1990s-2000s

More of everything:

- More data, more compute power $\Longrightarrow$ models with more parameters
- Multi-pass systems
- Context-dependent models

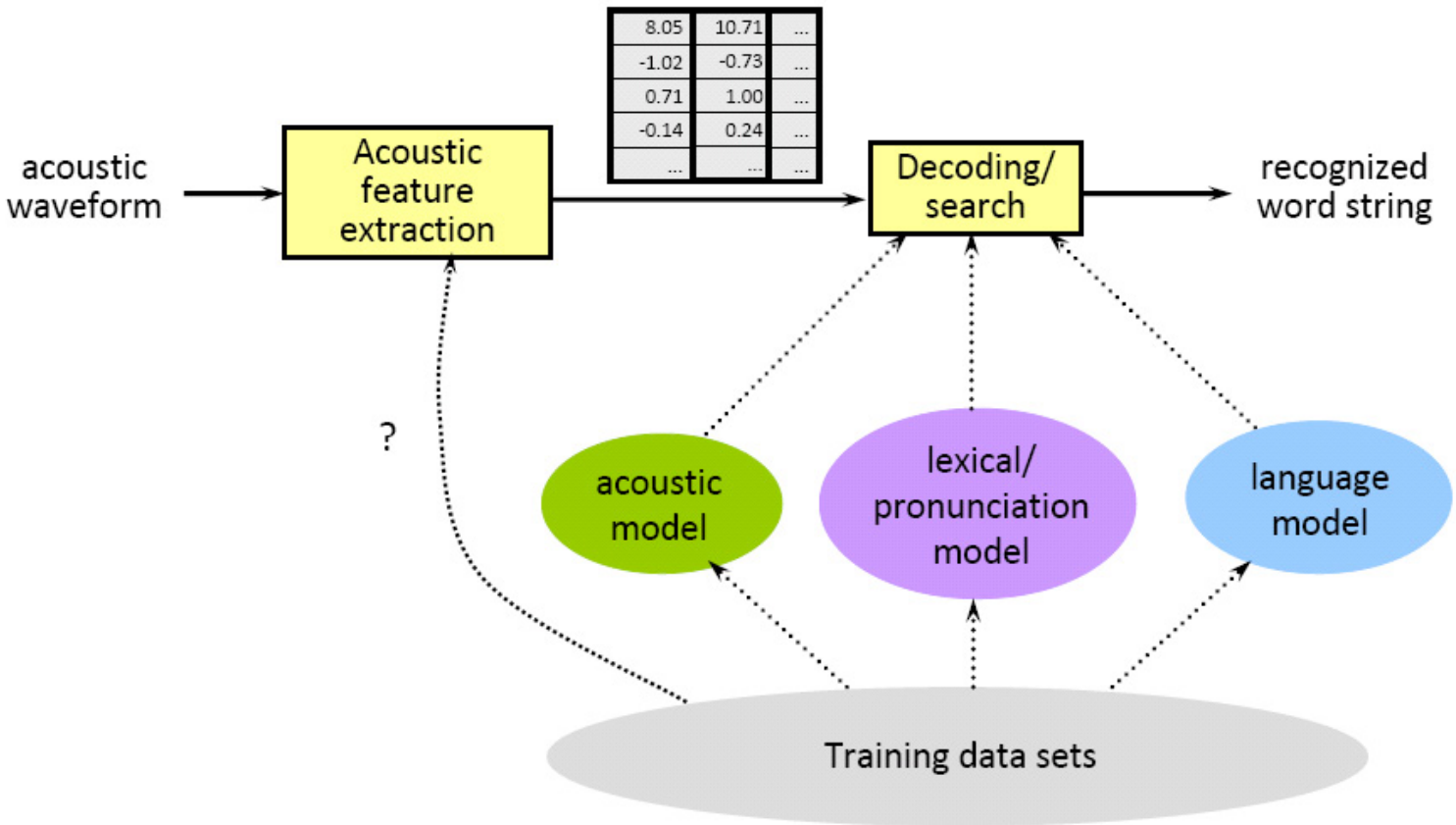
Robustness/adaptation to noise, speaker, style
Discriminative training
Alternatives to HMMs

2010s

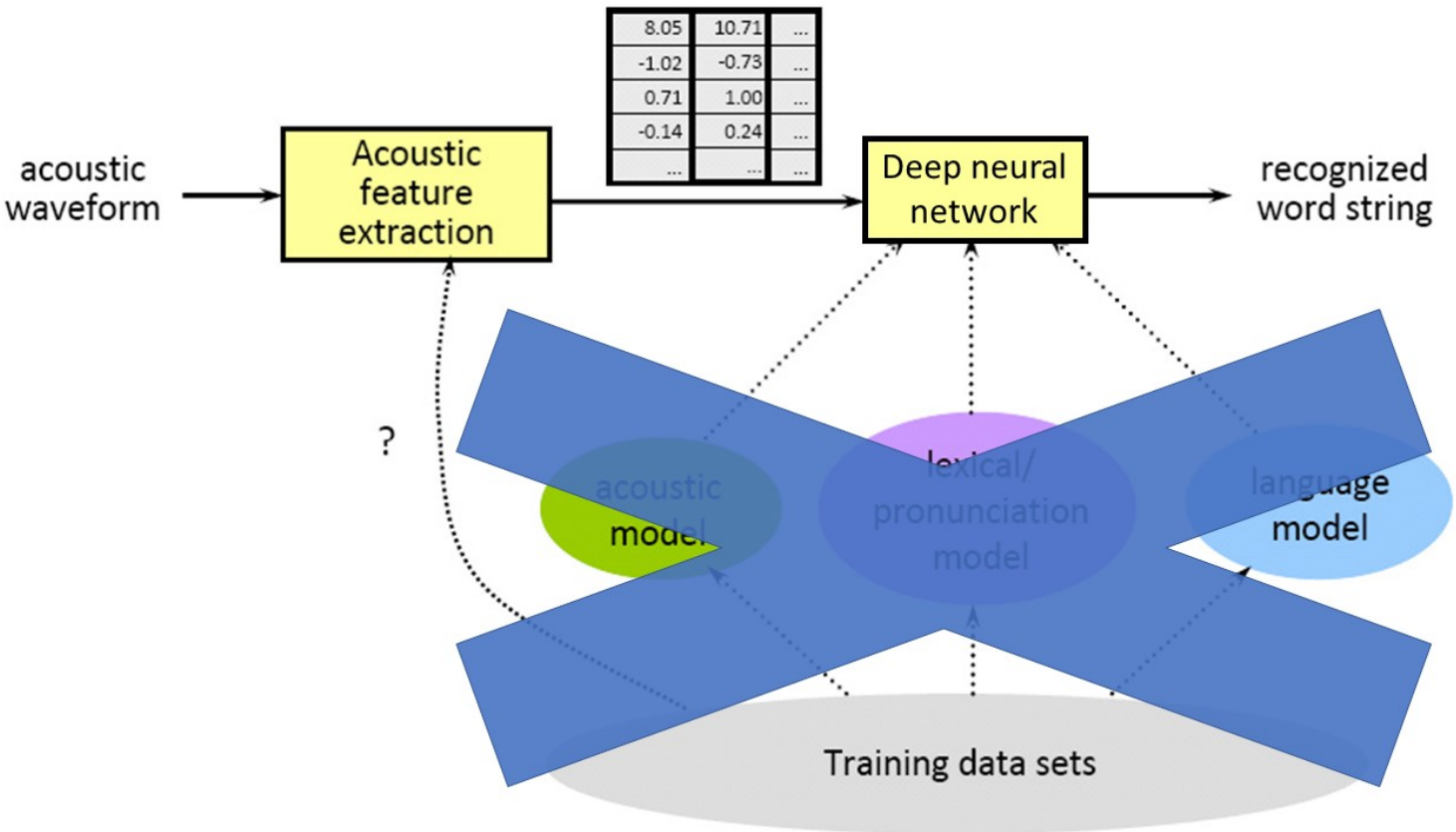
Closer ties with machine learning

- Renaissance of neural network-based methods
- Increased interest in speech tasks among machine learning community
- Hybrid HMM/NN approaches becomes the state of the art for most speech tasks
- End-to-end neural network methods gaining on HMMs

Architecture of a “traditional” speech recognizer

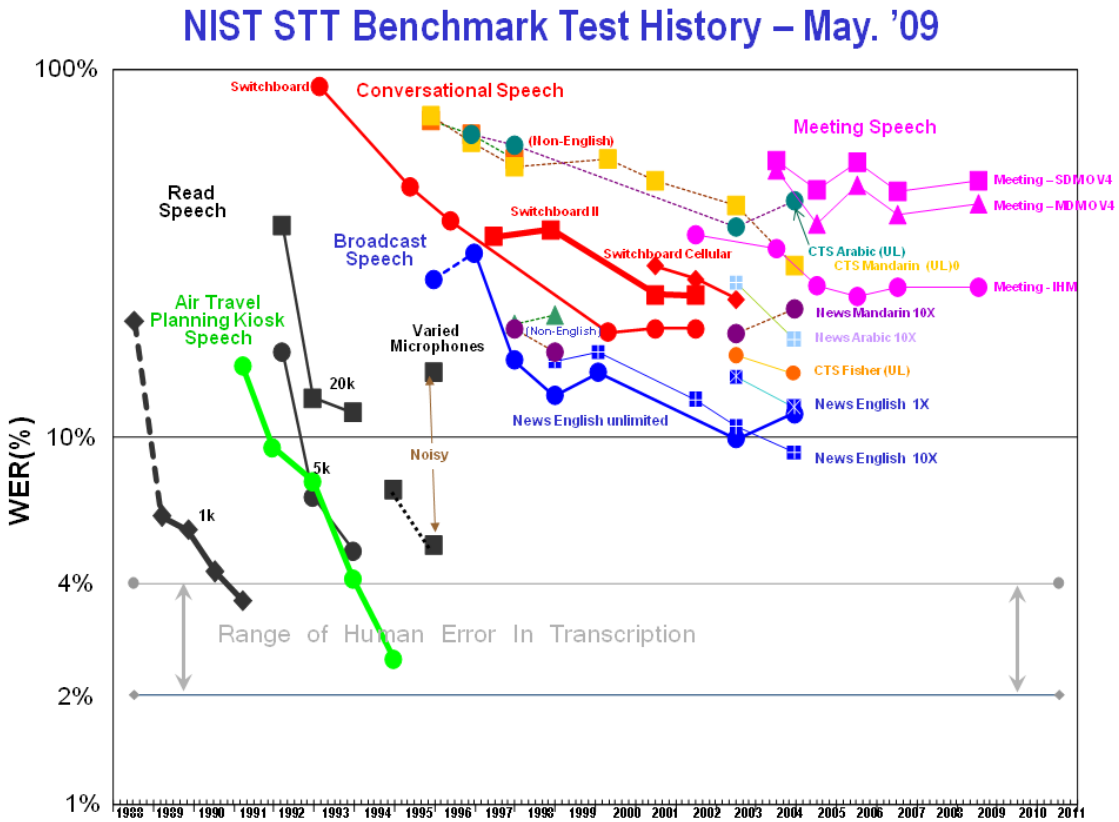


Architecture of an “end-to-end” speech recognizer



State of the art, 1988-2009

- NIST benchmark evaluation results
- WER = word error rate = $\frac{(\#\text{subs}+\#\text{ins}+\#\text{del})}{\#\text{ref}}$



meetings: ~25-40%

telephone conversations: ~20%

broadcast news: ~10%

WSJ dictation: ~5-10%

digits: $< 1\%$

(fig. from [Pallett+ 2009])

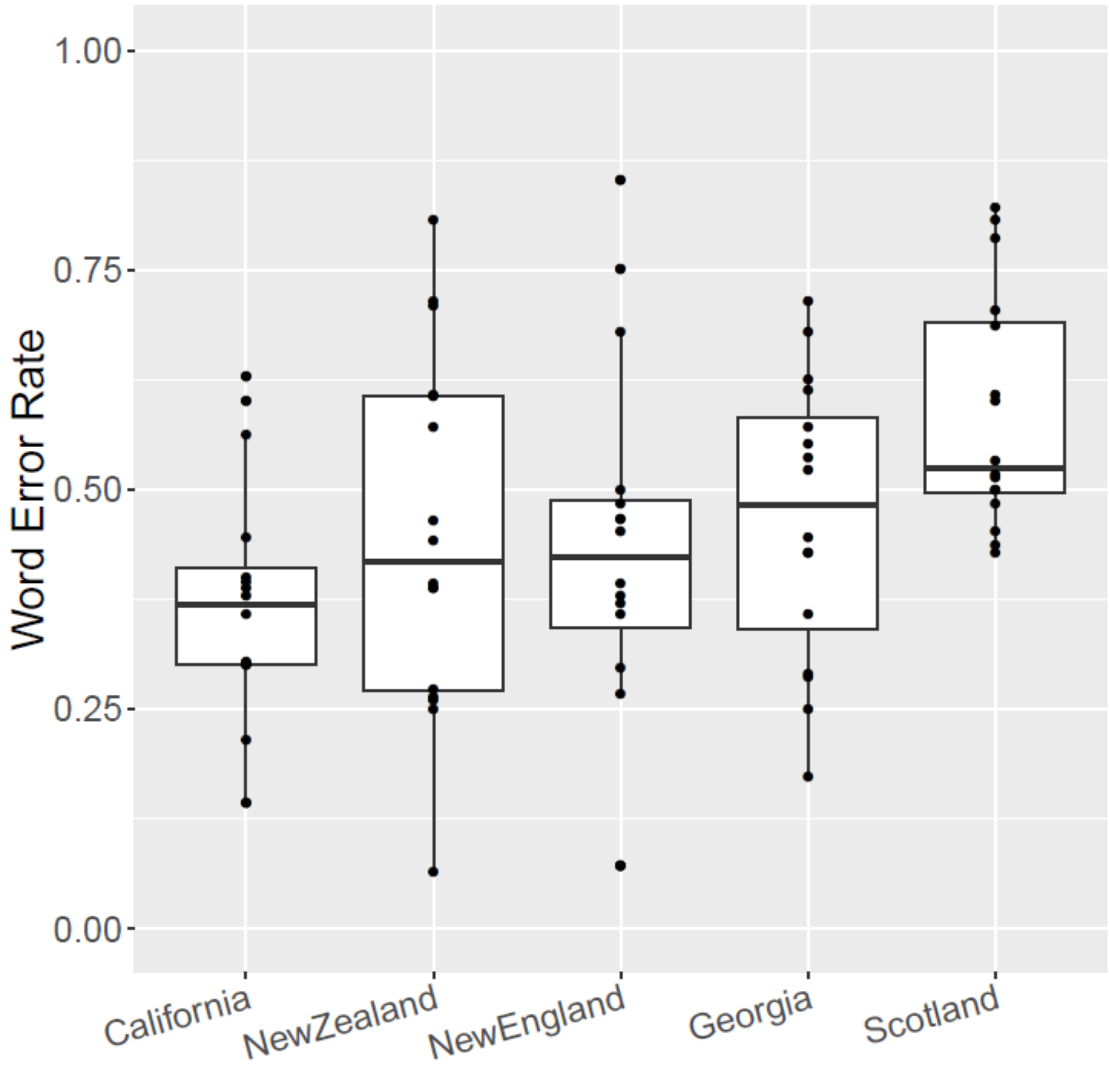
- Two trends: More informal speech, more acoustic variability

Some recent word error rates

- Switchboard (conversational telephone speech between strangers): 5.5% WER (Human: 5.1-6.9%)
- CallHome (conversational telephone speech between friends and family): 10.3% WER (Human: 6.8-8.7%)
- Wall Street Journal (read speech): $\sim 3\%$ (Human: $< 1\%$)

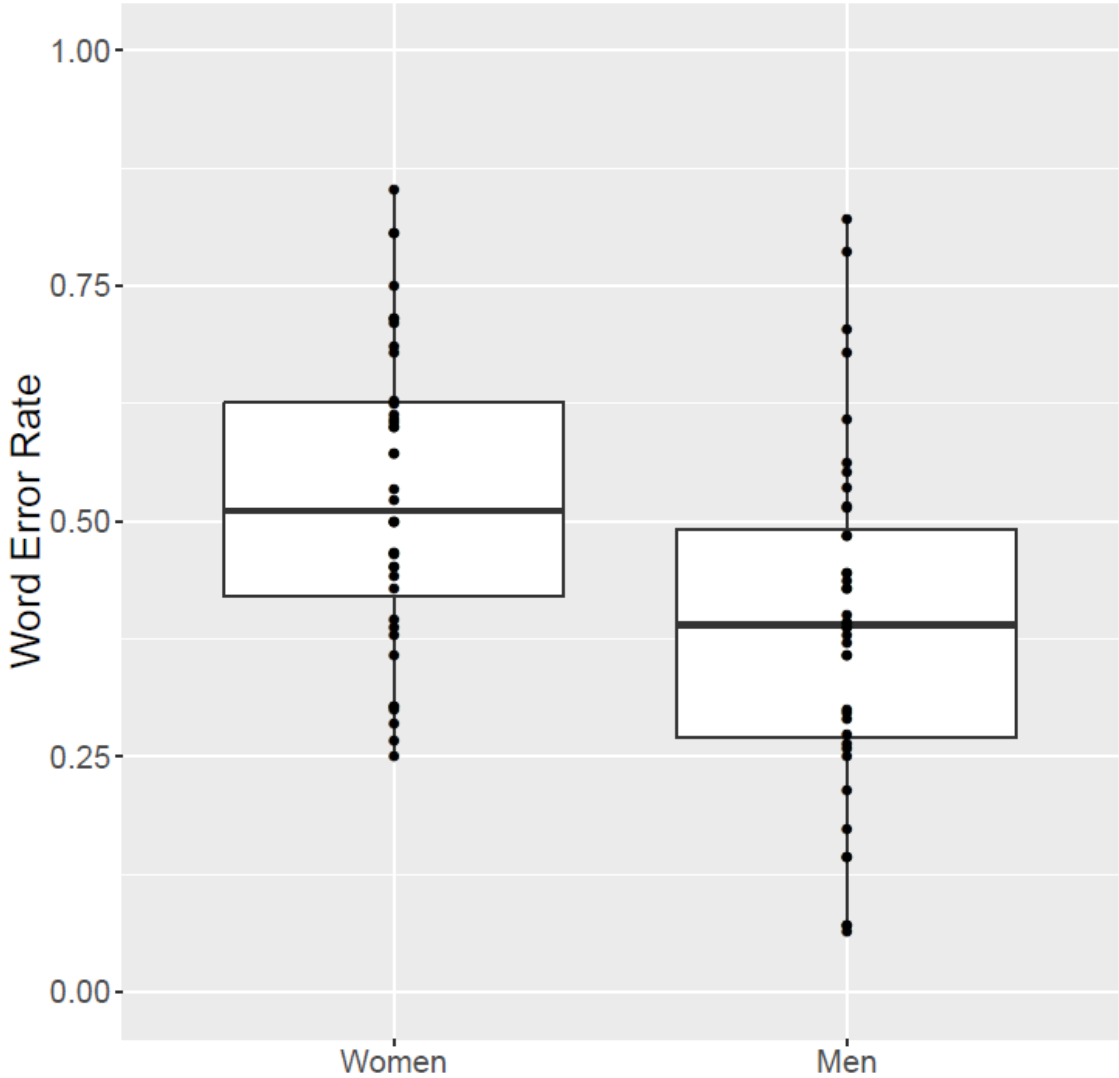
[Saon+ 2017, Hadian+ 2018, Lippmann 1997]

Dependence on dialect



(Fig. from [Tatman 2017])

Dependence on gender

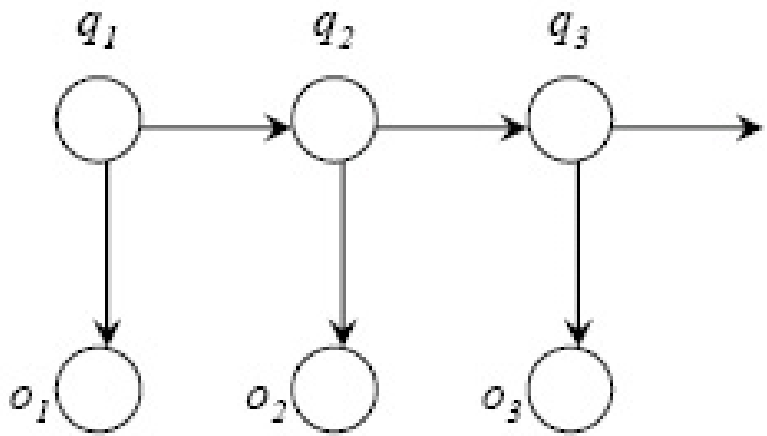


(Fig. from [Tatman 2017])

BREAK?

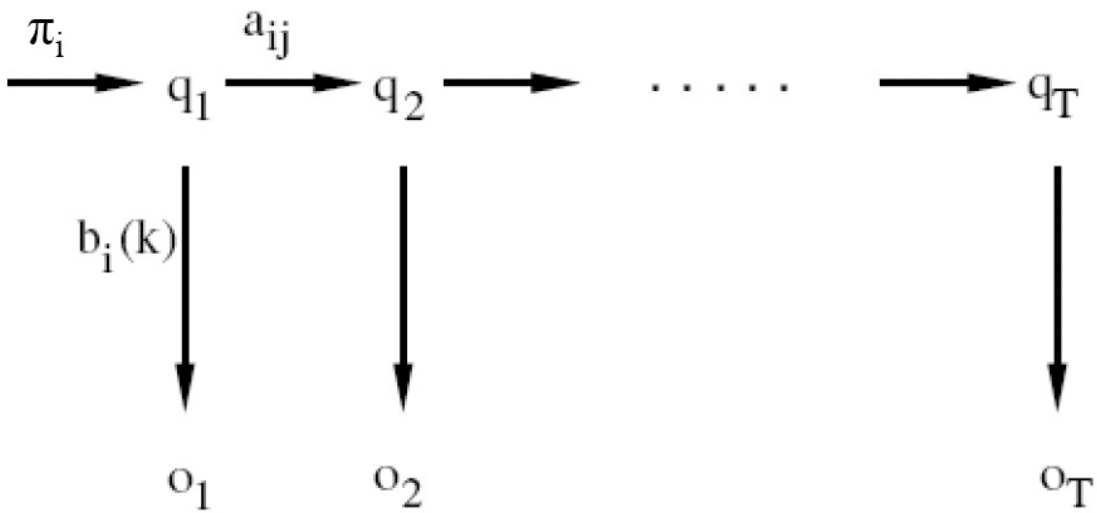
Hidden Markov models for speech recognition

- You've seen HMMs earlier in MLSS
- Reminder: Graphical model representation of HMMs



Review: Generation of observations from an HMM

1. Draw an initial state $q_1 = i$ from the initial distribution $\boldsymbol{\pi}$
2. For $t = 1$ to T:
   - Choose $o_t = k$ according to state i's observation distribution $b_i(k)$
   - Choose a new state $q_{t+1} = j$ according to state i's transition probabilities a_{ij}. Let $i \leftarrow j$



The three basic HMM problems

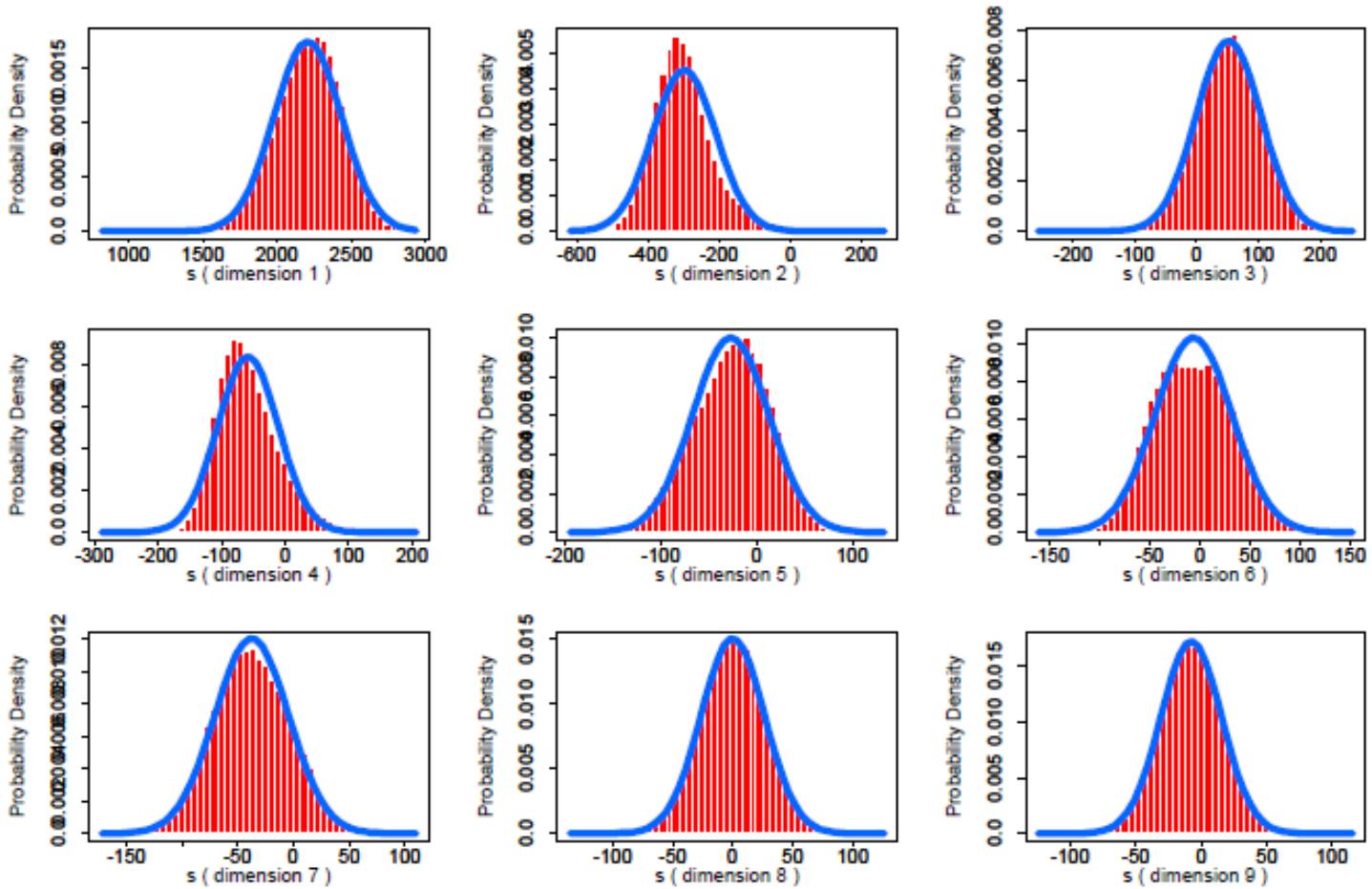
1. *Scoring*: Given an observation sequence $\mathbf{O} = \{\mathbf{o}_1, \ldots, \mathbf{o}_T\}$ and a model $\lambda = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$, compute $P(\mathbf{O}|\lambda)$, the probability of the observation sequence
   $\longrightarrow$ The forward & backward algorithms
2. *Decoding*: Given an observation sequence $\mathbf{O} = \{o_1, \ldots, o_T\}$, find the state sequence $\mathbf{q} = \{q_1, \ldots, q_T\}$ most likely to have generated the observations
   $\longrightarrow$ The Viterbi algorithm
3. *Training*: Given a training set of observations $\mathbf{O}$, set the model parameters $\lambda = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$ so as to maximize $P(\mathbf{O}|\lambda)$ (maximum-likelihood estimation)
   $\longrightarrow$ The Baum-Welch algorithm (EM applied to HMMs)

HMMs with continuous observation distributions

- In a *continuous-density* HMM, the discrete observation probabilities, $b_i(k)$, are replaced by continuous densities $b_i(\mathbf{o})$
- The observation distribution is typically a Gaussian or Gaussian mixture model (GMM): $b_i(\mathbf{o}) = \sum_{k=1}^{K} c_{ik}\mathcal{N}(\mathbf{o}|\boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik}), \quad 1 \leq i \leq N$
- $c_{ik}, \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik}$ are the parameters of Gaussian component k in state i
- Doesn't change much in the algorithms, but for GMMs we now have one more hidden variable (the choice of Gaussian component)

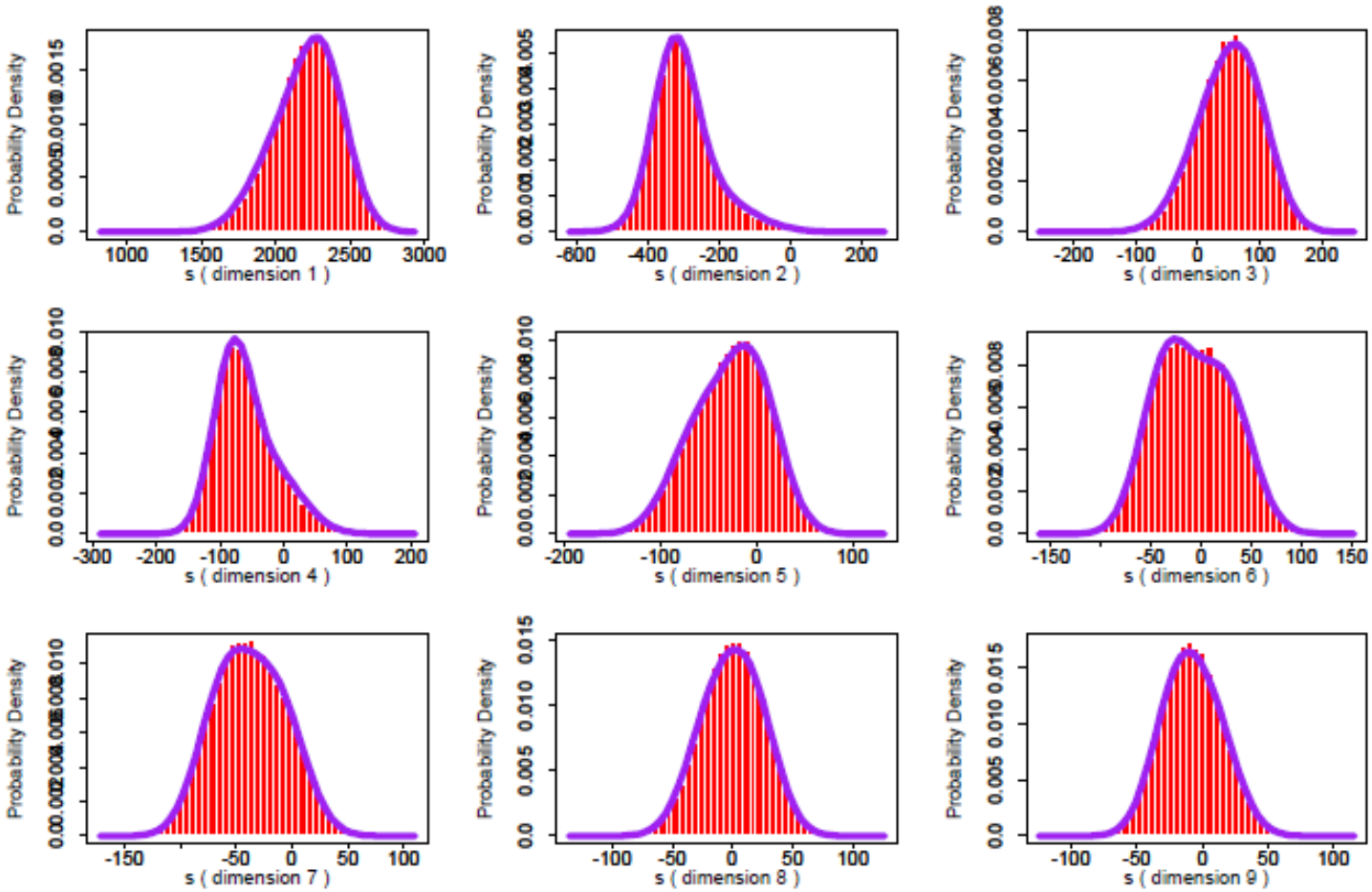
Example: Gaussian density for MFCCs

First 9 MFCC's from [s]: Gaussian PDF

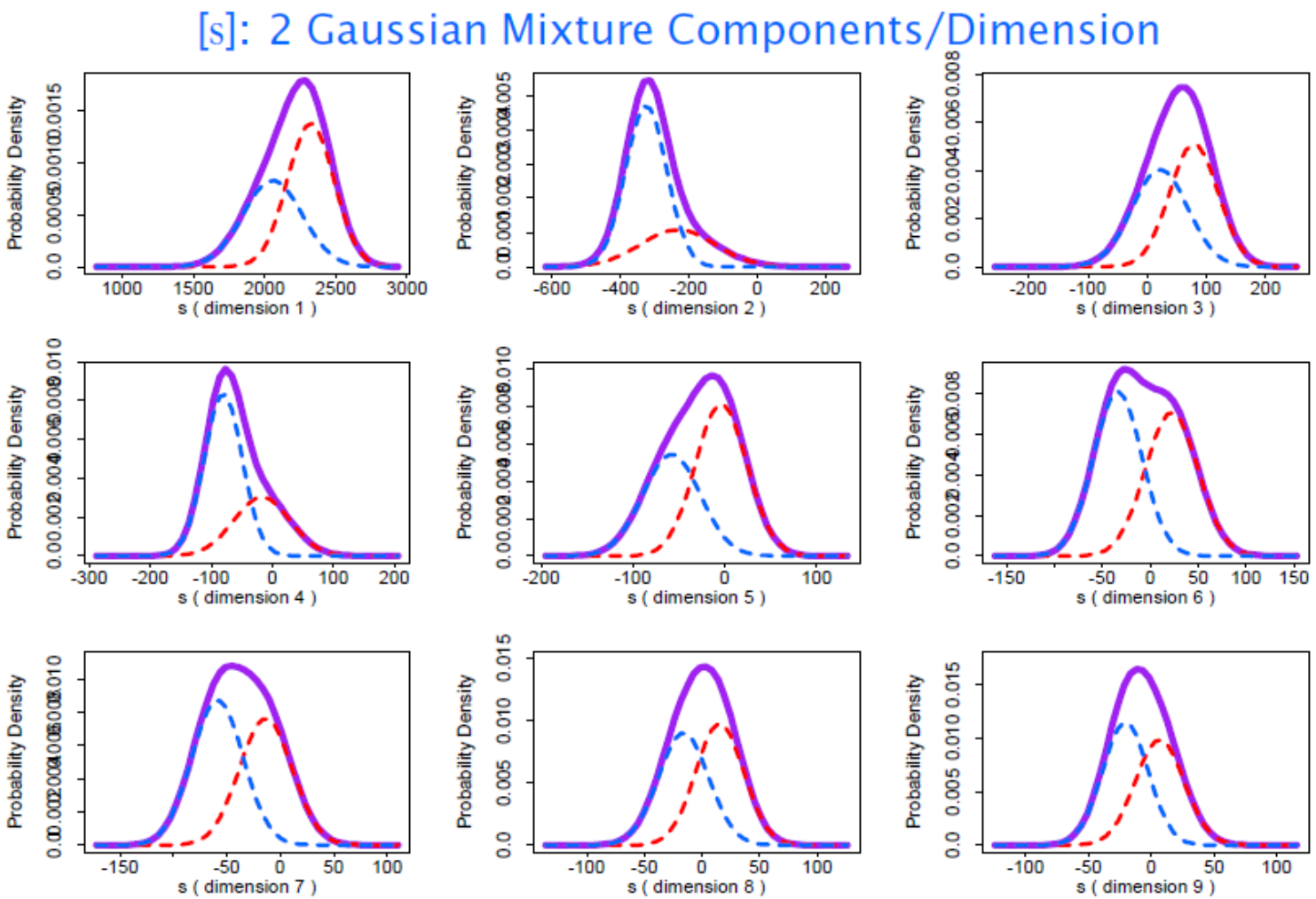


Example: Gaussian density for MFCCs

[s]: 2 Gaussian Mixture Components/Dimension



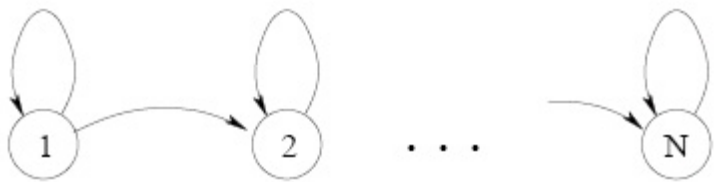
Example: Gaussian density for MFCCs



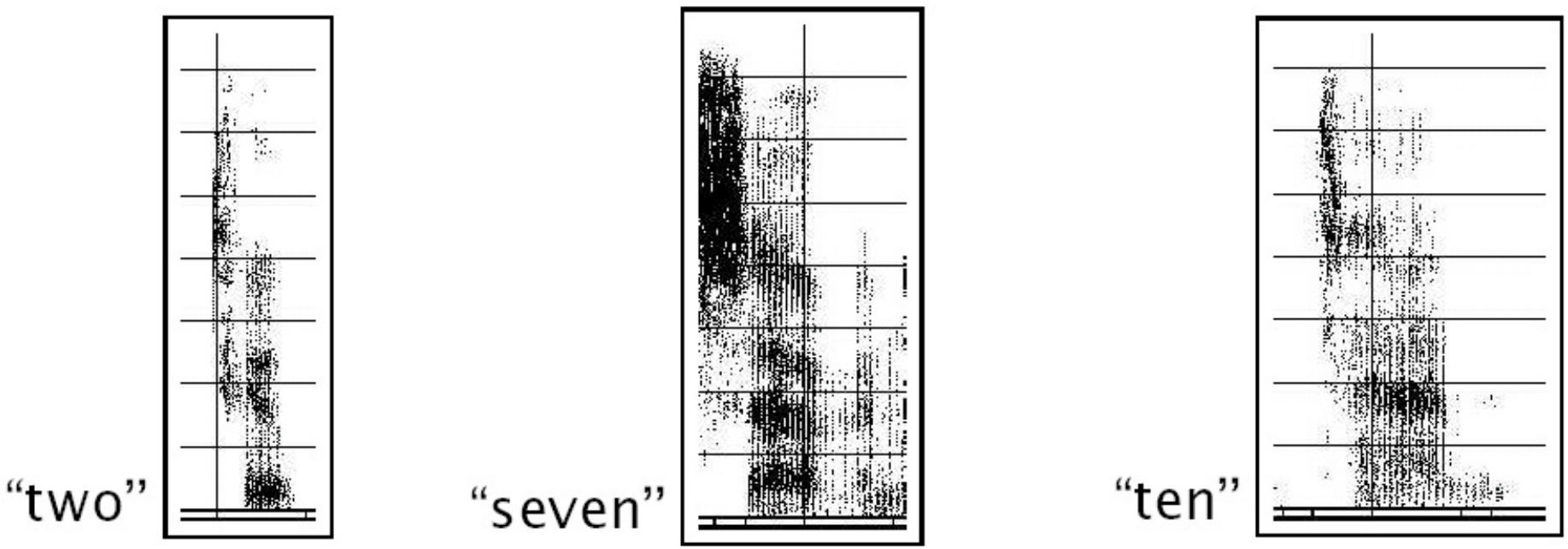
Typical HMM/GMM-based speech recognizers used 10s of Gaussians per state

Isolated-word recognition with whole-word HMMs

- Observations o_t: vector of acoustic features at time index ("frame") t
  - Typically, feature vectors are computed at 10ms intervals
- Each word is represented by one HMM; q_t represents a phonetic state within the word
- Assuming the words are equally likely, the hypothesized word w^* is the one with the highest $p(\mathbf{O}|\lambda_w)$
- Typical HMMs for ASR are "left-to-right"; example transition diagram for a word with N states:



How many states per word model? (What IS a state?)



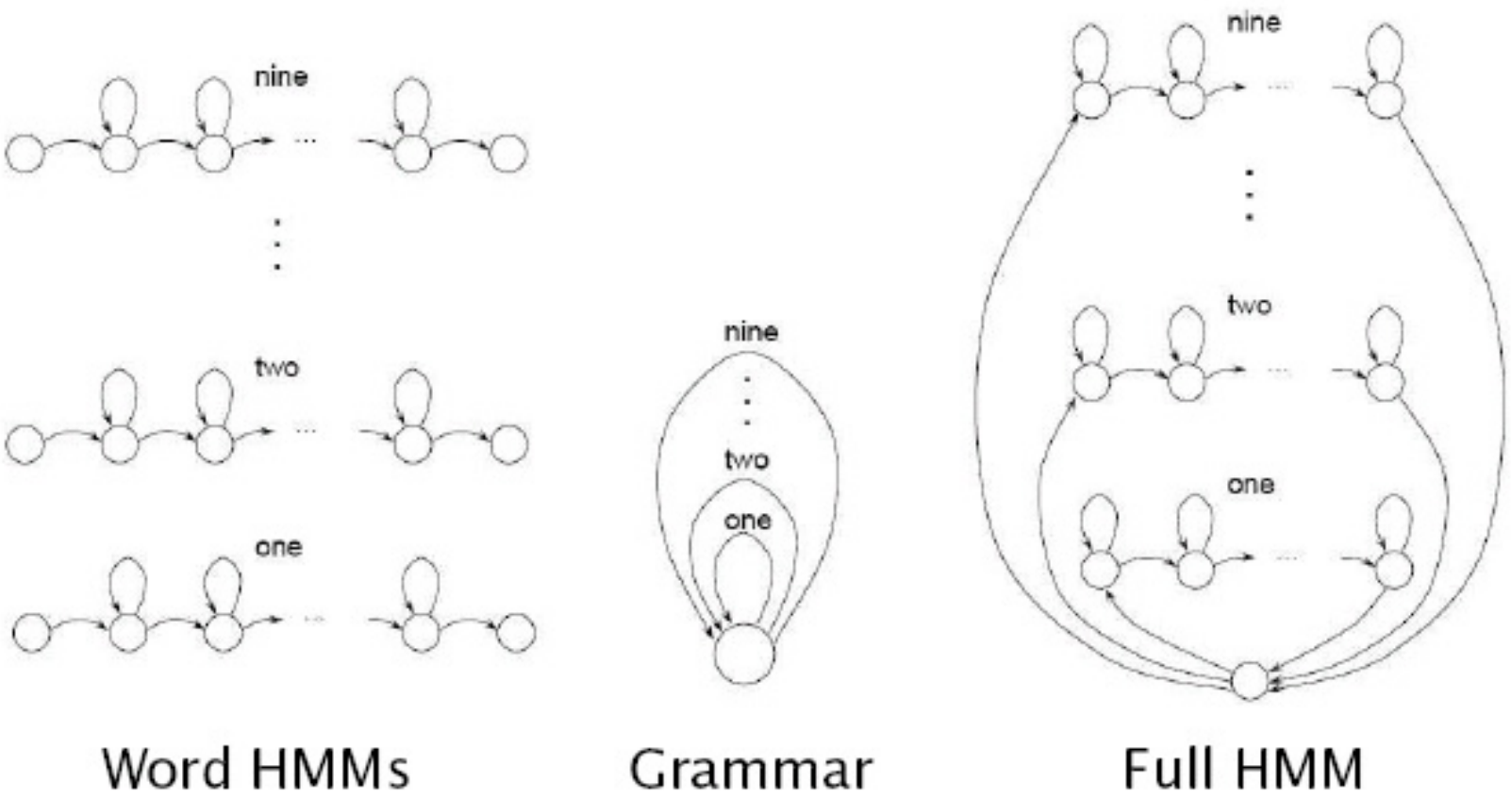
Typical choices: 8-16 states per word, or 3 $\times$ #phones

Continuous-speech (multi-word sequence) recognition with HMMs

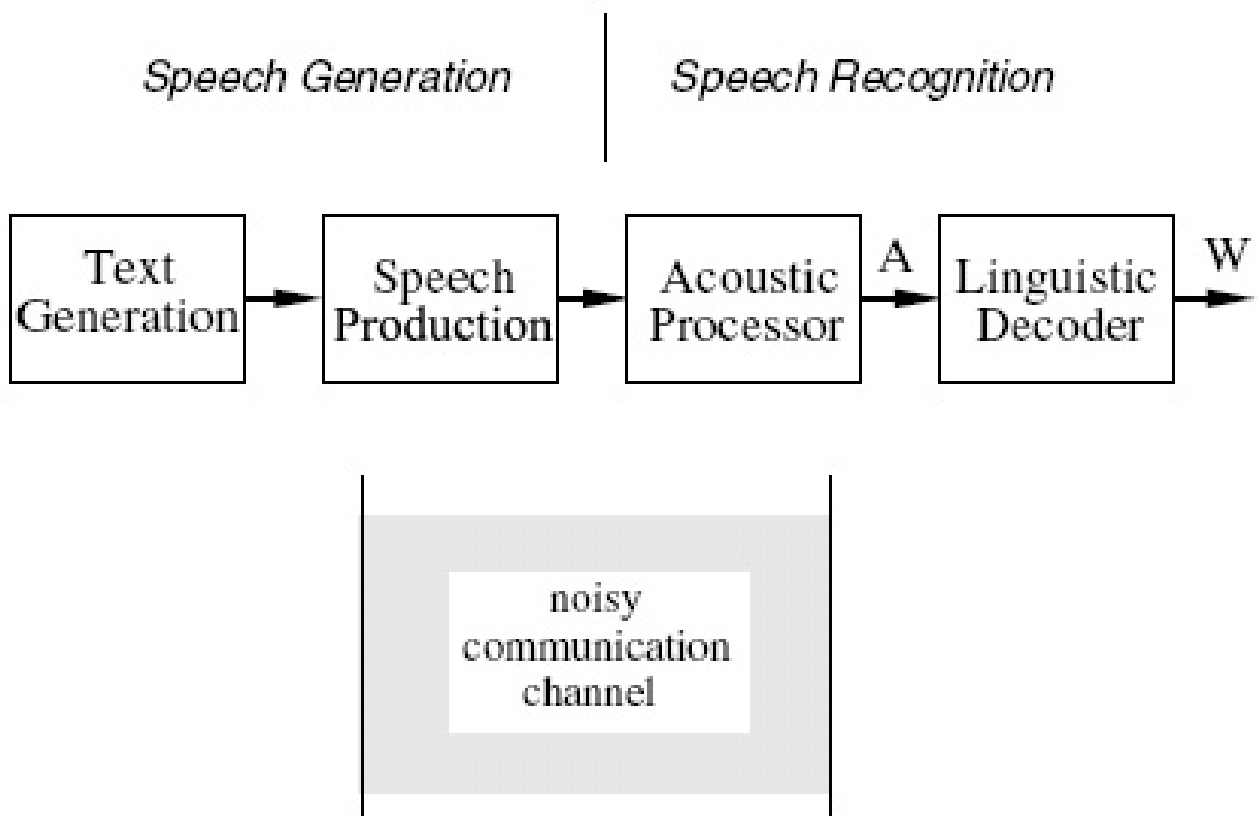
- For small vocabularies, string together whole-word HMMs to make whole-sentence HMMs
  - Works well if we have enough training data for each word
- For large-vocabulary tasks, string together sub-word HMMs (e.g. phone HMMs) instead – maybe more on this later...
- But, $w^* = \underset{w}{\operatorname{argmax}}\, p(\mathbf{O}|\lambda_w)$ is infeasible when w ranges over all possible sentences
  - Typical approach: Find the most likely *state* sequence and look up the corresponding word string
  - (Suboptimal: There are many possible state sequences for the same word string)

Continuous speech recognition with HMMs

Most basic model: String together word HMMs to make sentence HMM, using a grammar (Note: Start and end states of the word HMMs are non-emitting)



Continuous ASR: The noisy channel model



Recognition = finding the most probable word string $\mathbf{w}^*$ given the acoustic observations $\mathbf{O}$: $\mathbf{w}^* = \text{argmax}_{\mathbf{w}} p(\mathbf{w}|\mathbf{O})$

Continuous ASR: The noisy channel model (2)

From Bayes' rule, $p(\mathbf{w}|\mathbf{O}) = \frac{p(\mathbf{O}|\mathbf{w})p(\mathbf{w})}{p(\mathbf{O})}$. Therefore,

$$\begin{aligned} \mathbf{w}^* &= \operatorname*{argmax}_{\mathbf{w}} p(\mathbf{w}|\mathbf{O}) \\ &= \operatorname*{argmax}_{\mathbf{w}} p(\mathbf{O}|\mathbf{w})p(\mathbf{w}) \end{aligned}$$

- This is the "fundamental equation" of speech recognition
- The "whole-sentence" HMM gives $p(\mathbf{O}|\mathbf{w})$, the acoustic model
- $p(\mathbf{w})$ is the language model

Continuous ASR: The noisy channel model (3)

Summing over all possible state sequences $\mathbf{q}$ for the word string $\mathbf{w}$:

$$\begin{aligned}\mathbf{w}^* &= \operatorname*{argmax}_{\mathbf{w}} p(\mathbf{O}|\mathbf{w})p(\mathbf{w}) \\ &= \operatorname*{argmax}_{\mathbf{w}} \sum_{\mathbf{q}} p(\mathbf{O}|\mathbf{q},\mathbf{w})p(\mathbf{q}|\mathbf{w})p(\mathbf{w})\end{aligned}$$

- $p(\mathbf{O}|\mathbf{q},\mathbf{w})$ is given by the observation (emission) distribution
- $p(\mathbf{q}|\mathbf{w})$ is given by the state transition probabilities
- *Viterbi approximation*: Assume there is a single most probable state sequence $\mathbf{q}^*$ such that all other $\mathbf{q}$ contribute a negligible amount to the sum. Then
  $\mathbf{w}^* = \operatorname*{argmax}_{\mathbf{w}} p(\mathbf{O}|\mathbf{q}^*,\mathbf{w})p(\mathbf{q}^*|\mathbf{w})p(\mathbf{w})$
- So we can maximize jointly over $\mathbf{w}$ and $\mathbf{q}$:
  $\mathbf{w}^*, \mathbf{q}^* = \operatorname*{argmax}_{\mathbf{w},\mathbf{q}} p(\mathbf{O}|\mathbf{q},\mathbf{w})p(\mathbf{q}|\mathbf{w})p(\mathbf{w})$

Summary: Continuous ASR with HMMs under the noisy channel model

- To do continuous speech recognition, we will string together word HMMs according to a *grammar* or *language model*
- For now, assume grammar allows arbitrary word sequences with equal probabilities
- (We might get back to language modeling later)
- We will find the best state sequence using the Viterbi algorithm, and output the corresponding word string

EM for speech recognition

Training a speech recognizer is more than just training a single HMM...

- We have multiple HMMs, one per word or sub-word unit
- We often don't know the start and end time of each word/sub-word
- Then these are part of the "state sequence latent variable" in the EM algorithm

Meta-algorithm 1: Training a (whole-word) HMM/GMM-based speech recognizer

(1) Given:

- Training set of L utterances (acoustic features + corresponding word transcriptions)
- Hyperparameters: # states per word, # Gaussians per state, HMM "topology" (which transition probabilities are 0)
- Initial parameter values (guess)

(2) Repeat until convergence:

- E step: For each training utterance l, run forward and backward algorithms and compute the state occupation probabilities
- M step: Update parameters according to the Baum-Welch equations
- Check convergence (e.g., likelihood not higher than previous iteration by some amount δ)

The need for subword units

Whole-word HMMs have some problems

- Cannot model unseen words, or even words seen too few times in training data
- Number of parameters is proportional to the vocabulary size

$\Rightarrow$ Whole-word models mainly restricted to small-vocabulary tasks

Subword units

- In the same way that sentences can be composed of whole-word HMMs, words can be composed of sub-word HMMs
- Units are then shared among words

Type of units	Approximate # (in English)
words	>100,000
phones	50
diphones	2,000
triphones	10,000
syllables	5,000

- Number of parameters now proportional to the number of sub-word units, not number of words

Baseforms

- Each word can be represented as a sequence of phonemes, the word's *baseform*
  `dogs` $\longrightarrow$ d ao g z
- Some words may need more than one baseform
  `the` $\longrightarrow$ dh ah, dh iy
  `either` $\longrightarrow$ iy dh er, ay dh er
- Each word's baseform(s) can be looked up in a dictionary
- All words in training and test data must be in the dictionary, or else we get them wrong
- Typically, baseform dictionaries are written by hand $\longrightarrow$ prone to errors, inconsistencies, disagreements among linguists, ...