Automatic Speech Recognition (I)

borrowing from
Daniel Jurafsky and James Martin

Outline for ASR

- ASR Architecture
  - The Noisy Channel Model
- Five easy pieces of an ASR system
  1) Language Model
  2) Lexicon/Pronunciation Model (HMM)
  3) Feature Extraction
  4) Acoustic Model
  5) Decoder
- Training
- Evaluation

Speech Recognition

- Applications of Speech Recognition (ASR)
  - Dictation
  - Telephone-based Information (directions, air travel, banking, etc)
  - Hands-free (in car)
  - Speaker Identification
  - Language Identification
  - Second language ('L2') (accent reduction)
  - Audio archive searching

LVCSR

- Large Vocabulary Continuous Speech Recognition
- ~20,000-64,000 words
- Speaker independent (vs. speaker-dependent)
- Continuous speech (vs isolated-word)

Current error rates

Ballpark numbers; exact numbers depend very much on the specific corpus

Task	Vocabulary	Error Rate%
Digits	11	0.5
WSJ read speech	5K	3
WSJ read speech	20K	3
Broadcast news	64,000+	10
Conversational Telephone	64,000+	20

HSR versus ASR

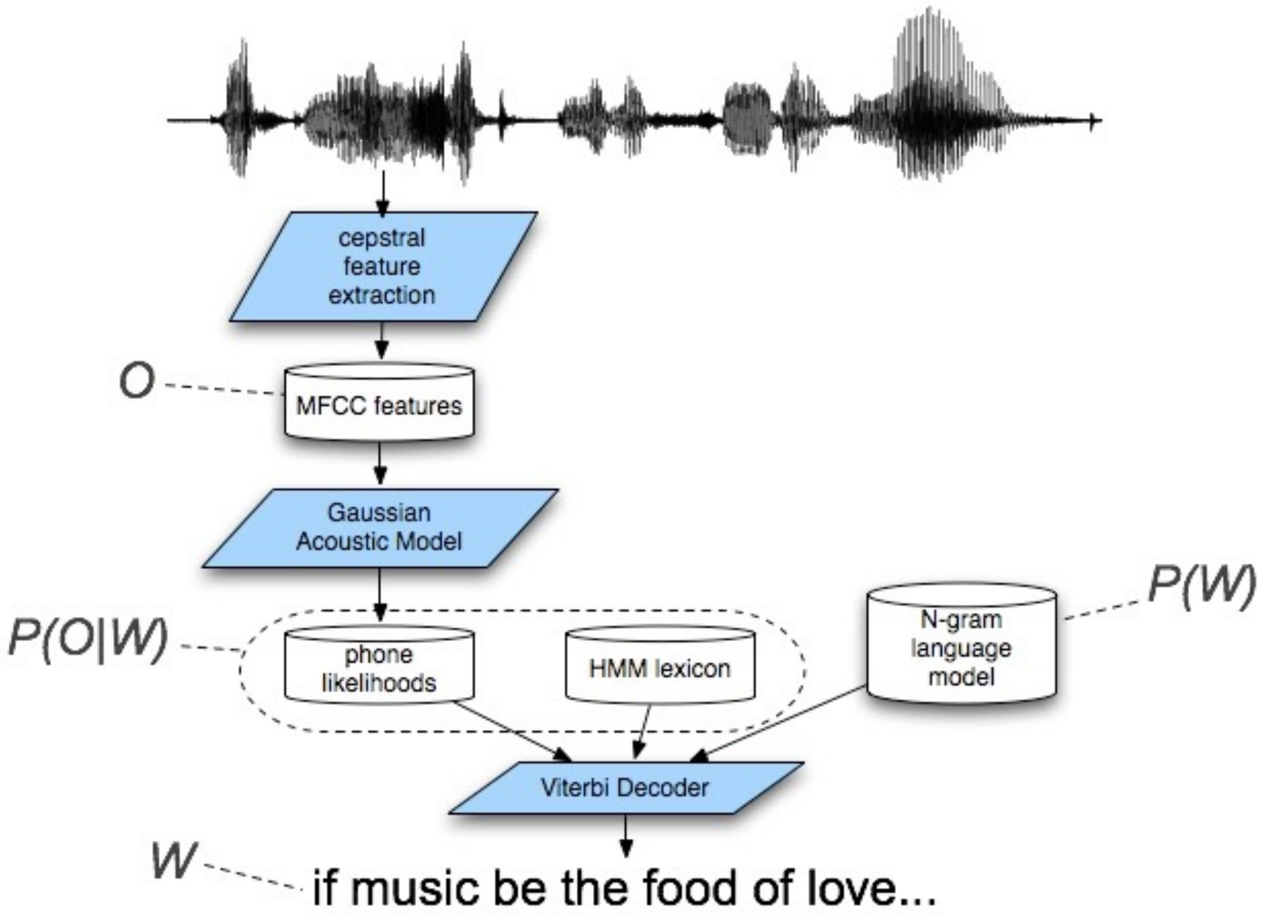
Task	Vocab	ASR	Hum SR
Continuous digits	11	.5	.009
WSJ 1995 clean	5K	3	0.9
WSJ 1995 w/noise	5K	9	1.1
SWBD 2004	65K	20	4

- Conclusions:
  - Machines about 5 times worse than humans
  - Gap increases with noisy speech
  - These numbers are rough, take with grain of salt

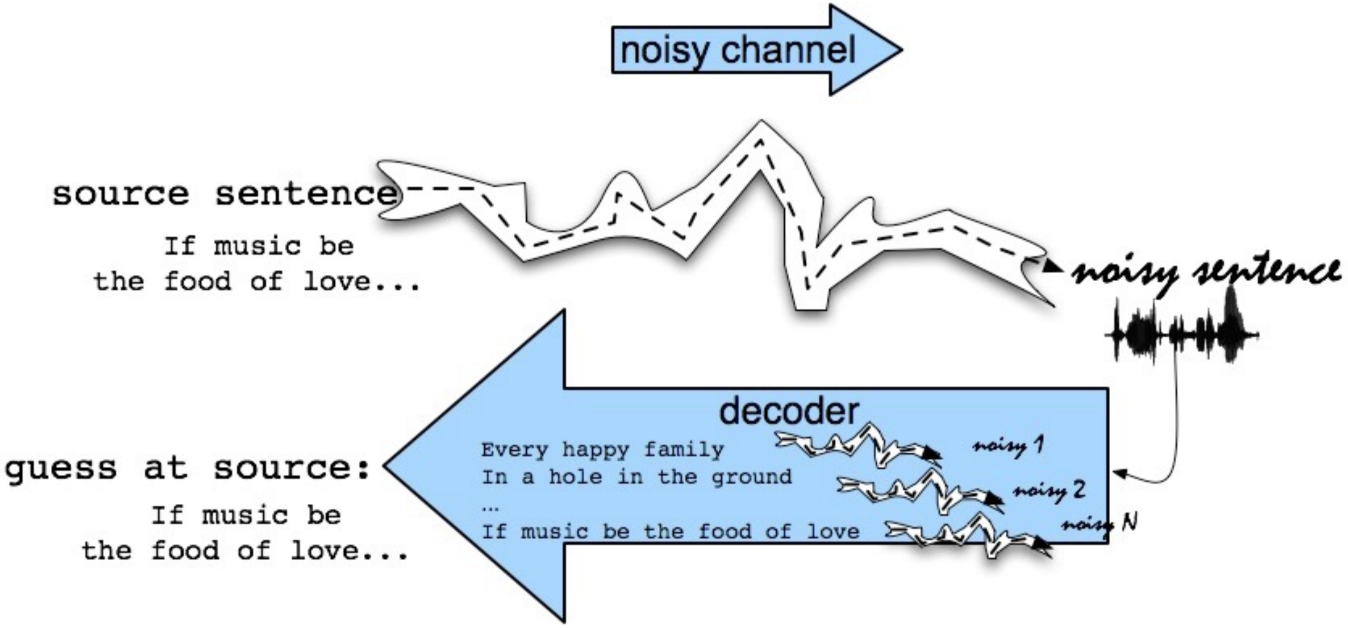
LVCSR Design Intuition

- Build a statistical model of the speech-to-words process
- Collect lots and lots of speech, and transcribe all the words.
- Train the model on the labeled speech
- Paradigm: Supervised Machine Learning + Search

Speech Recognition Architecture



The Noisy Channel Model



- Search through space of all possible sentences.
- Pick the one that is most probable given the waveform.

The Noisy Channel Model (II)

- What is the most likely sentence out of all sentences in the language L given some acoustic input O?
- Treat acoustic input O as sequence of individual observations
  - $O = o_1,o_2,o_3,\ldots,o_t$
- Define a sentence as a sequence of words:
  - $W = w_1,w_2,w_3,\ldots,w_n$

Noisy Channel Model (III)

- Probabilistic implication: Pick the highest prob S = W:

$$\hat{W} = \underset{W \in L}{\arg\max}\, P(W \mid O)$$

- We can use Bayes rule to rewrite this:

$$\hat{W} = \underset{W \in L}{\arg\max}\, \frac{P(O \mid W)P(W)}{P(O)}$$

- Since denominator is the same for each candidate sentence W, we can ignore it for the argmax:

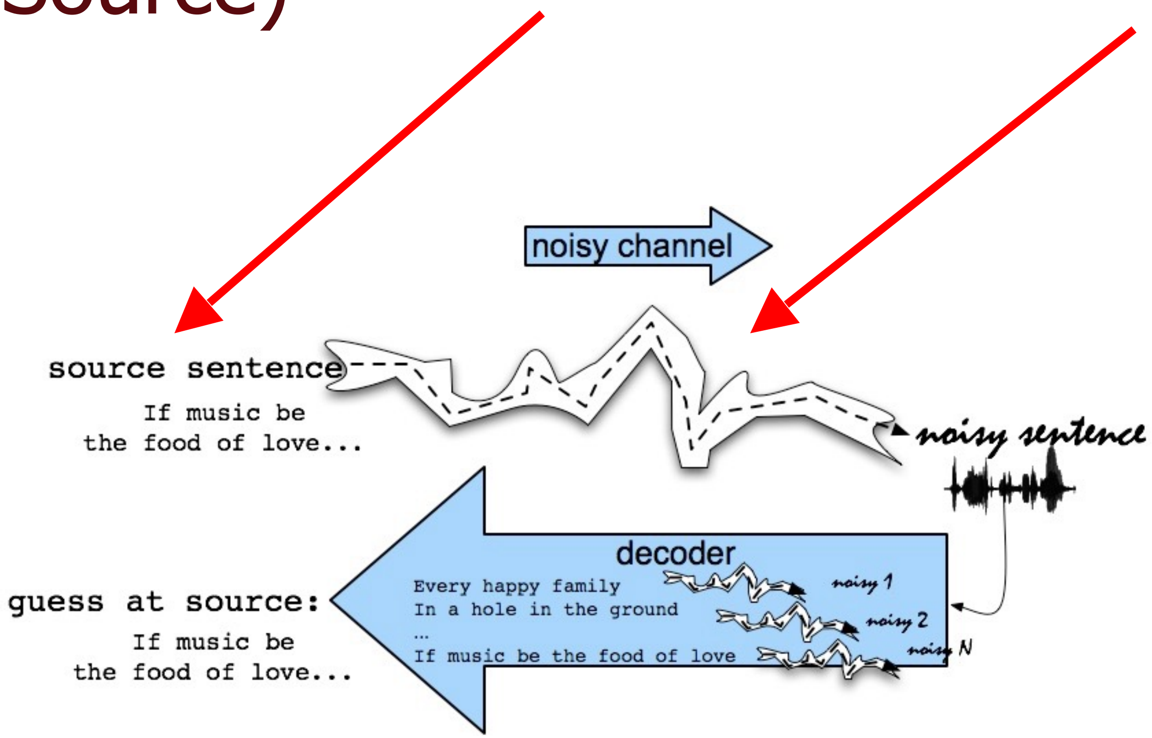
$$\hat{W} = \underset{W \in L}{\arg\max}\, P(O \mid W)P(W)$$

Noisy channel model

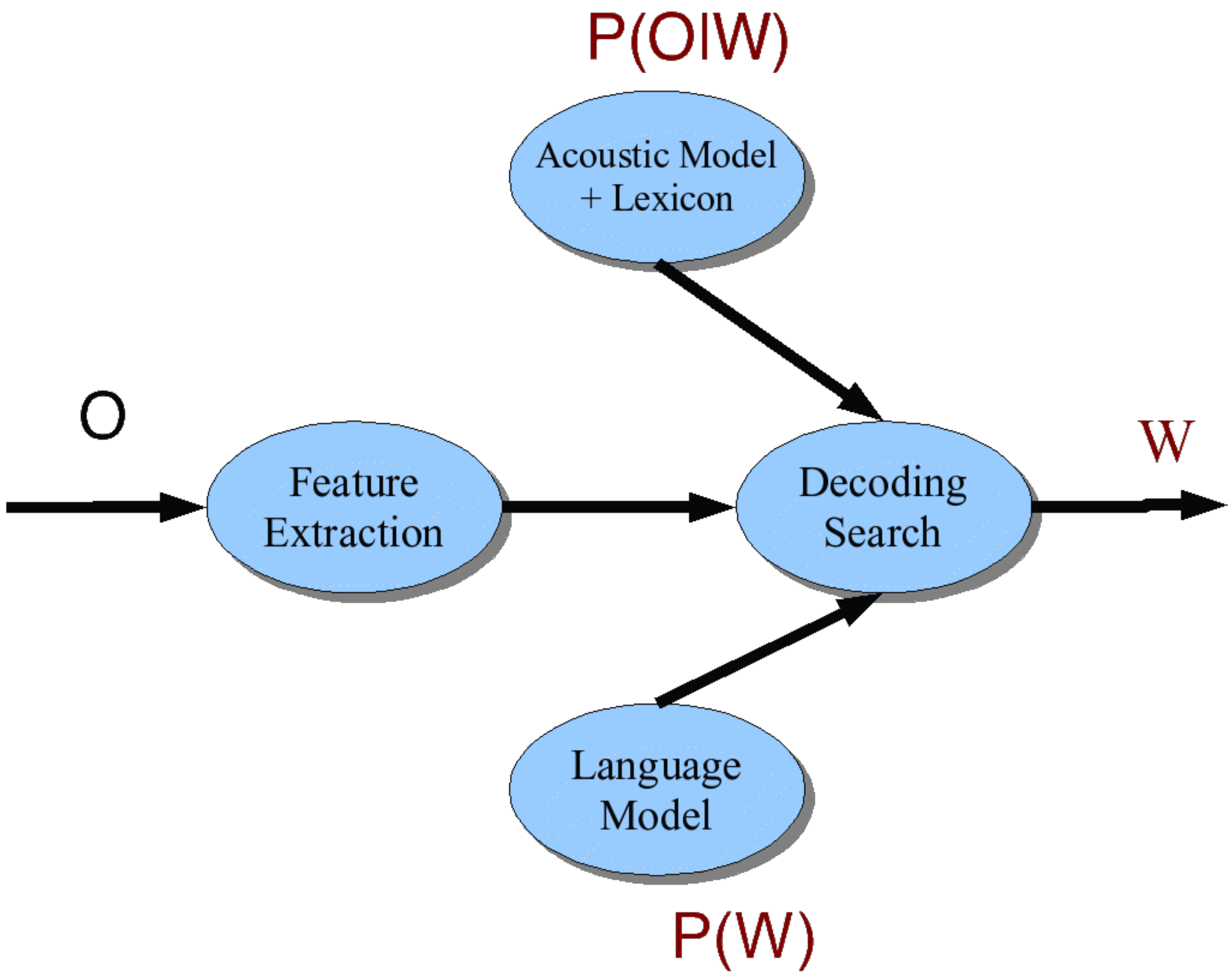
$$\hat{W} = \underset{W \in L}{\operatorname{argmax}} \underbrace{P(O \mid W)}_{\text{likelihood}} \underbrace{P(W)}_{\text{prior}}$$

The noisy channel model

- Ignoring the denominator leaves us with two factors: P(Source) and P(Signal| Source)



Speech Architecture meets Noisy Channel



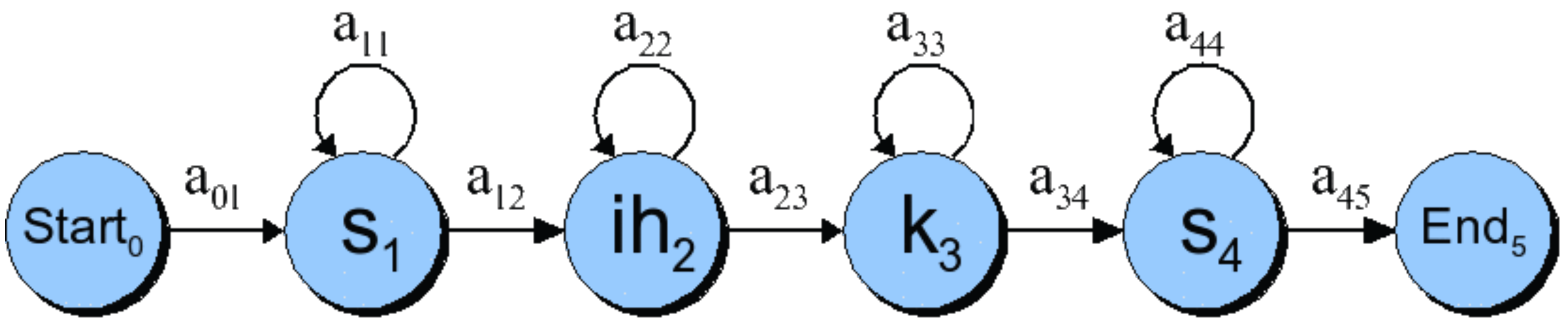
Architecture: Five easy pieces

- HMMs, Lexicons, and Pronunciation
- Feature extraction
- Acoustic Modeling
- Decoding
- Language Modeling (seen this already)

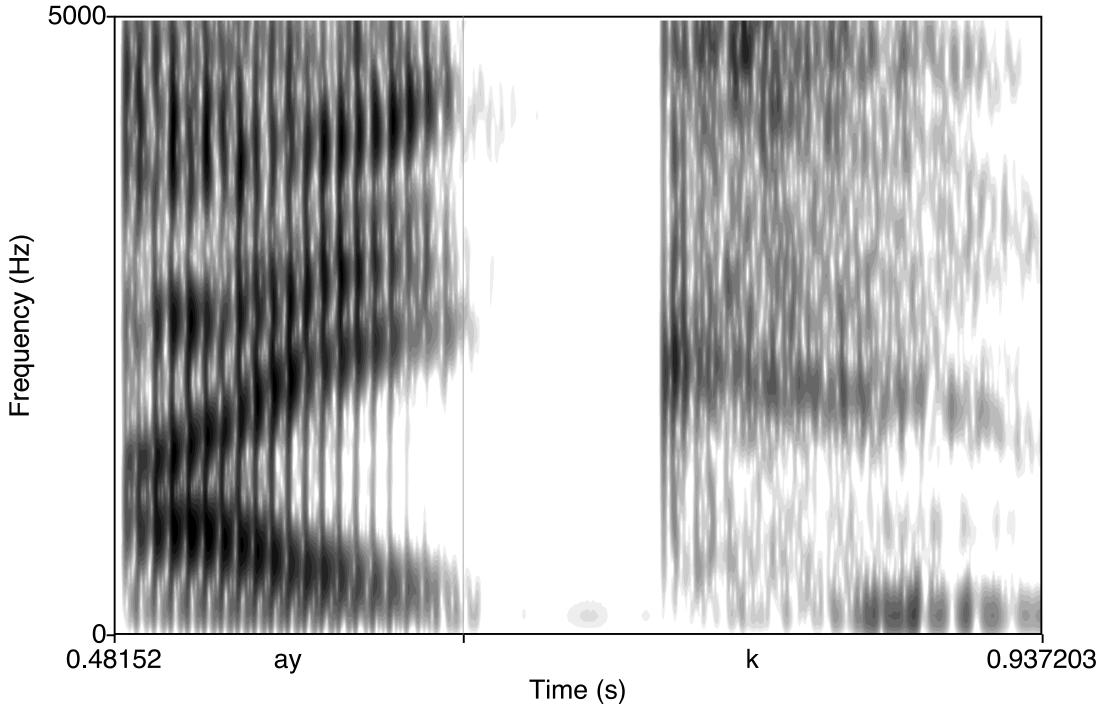
Lexicon

- A list of words
- Each one with a pronunciation in terms of phones
- We get these from an on-line pronunciation dictionary
- CMU dictionary: 127K words
  - http://www.speech.cs.cmu.edu/cgi-bin/cmudict
- We'll represent the lexicon as an HMM

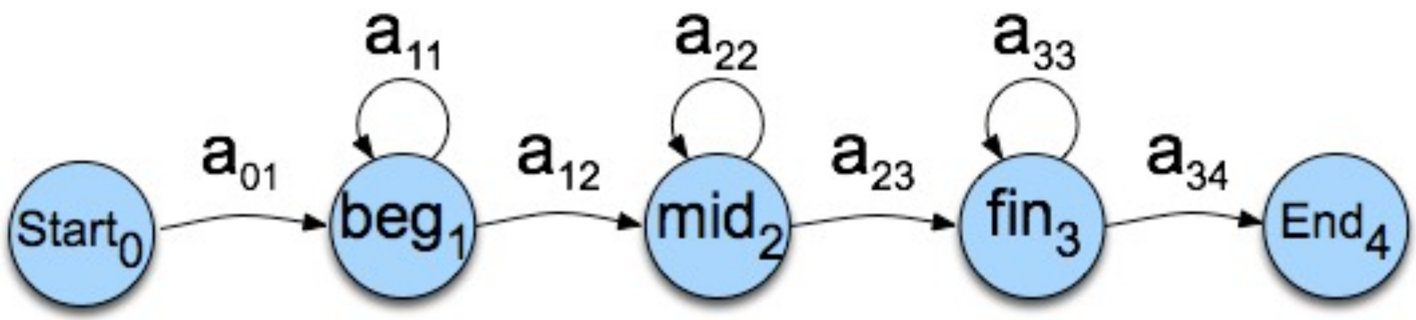
HMMs for speech: the word "six"



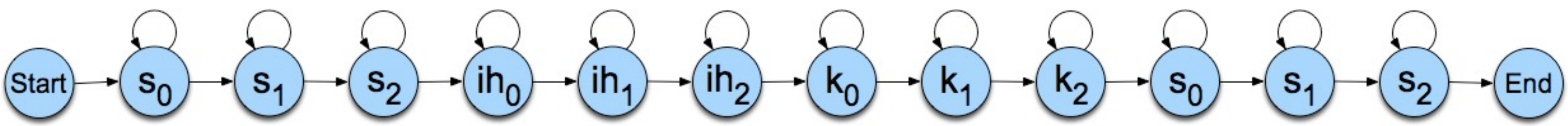
Phones are not homogeneous!



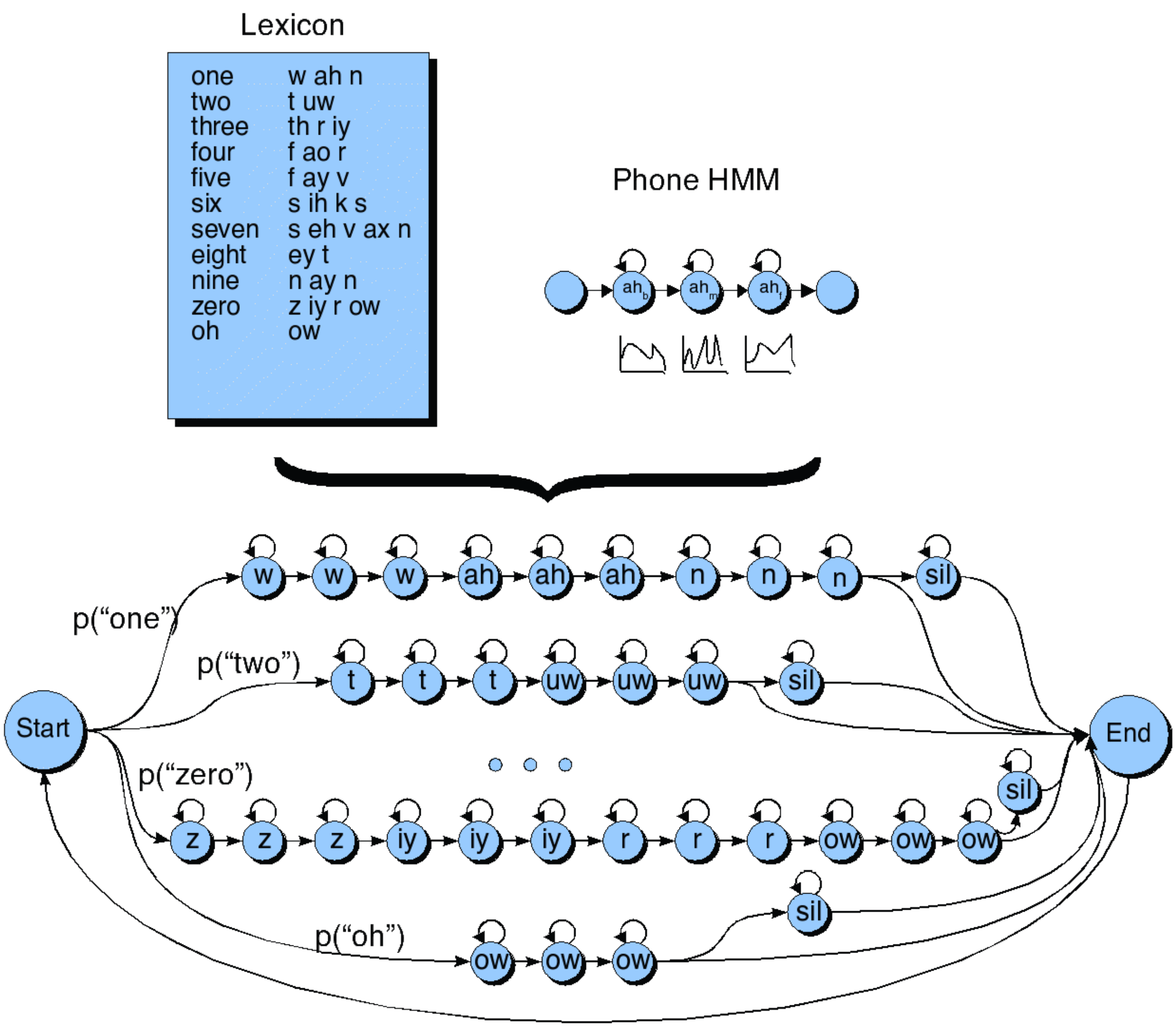
Each phone has 3 subphones



Resulting HMM word model for "six" with their subphones



HMM for the digit recognition task

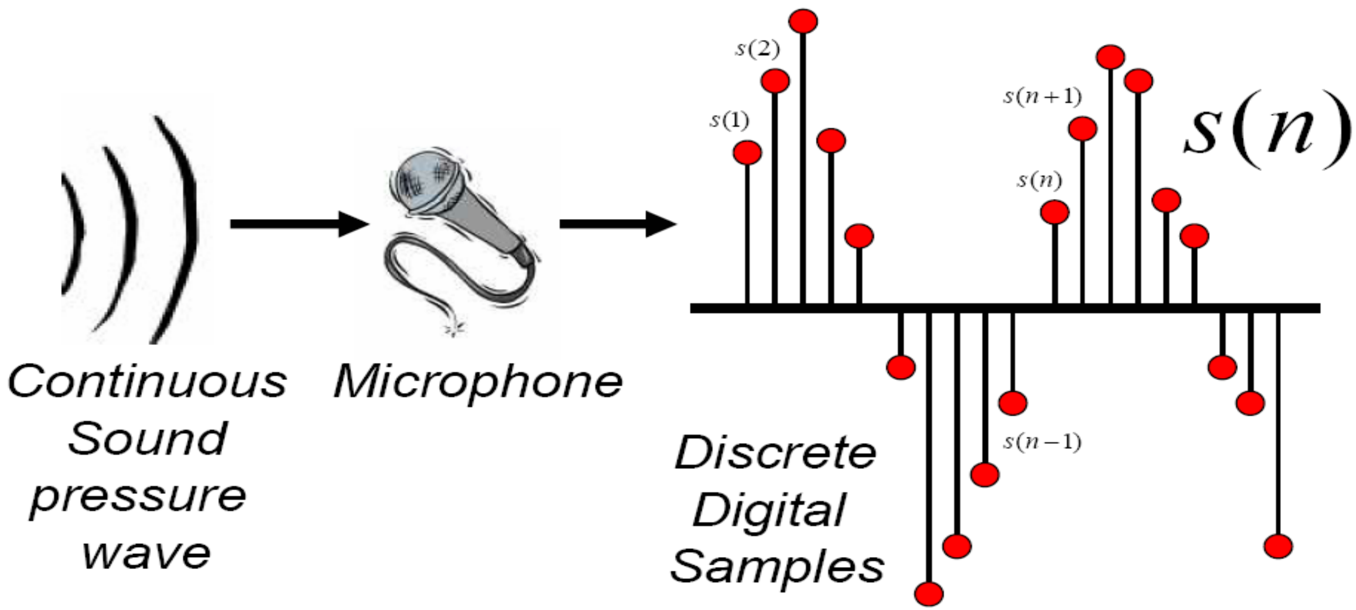


Detecting Phones

- Two stages
  - **Feature extraction**
    - Transforming raw acoustics into features
  - **Computing phone likelihoods**
    - Using GMM classifier

Discrete Representation of Signal

- Represent continuous signal into discrete form.

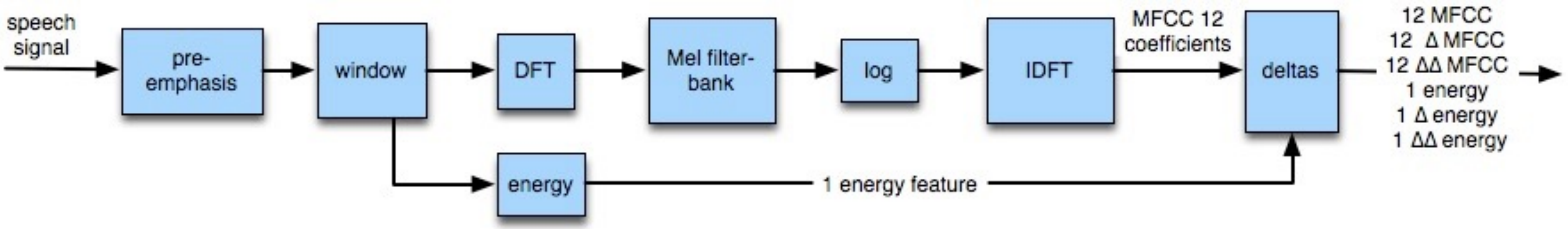


Thanks to Bryan Pellom for this slide

Digitizing the signal (A-D)

- **Sampling**:
  - measuring amplitude of signal at time t
  - 16,000 Hz (samples/sec) Microphone ("Wideband"):
  - 8,000 Hz (samples/sec) Telephone
  - Why?
    - Need at least 2 samples per cycle
    - max measurable frequency is half sampling rate
    - Human speech < 10,000 Hz, so need max 20K
    - Telephone filtered at 4K, so 8K is enough

MFCC: Mel-Frequency Cepstral Coefficients

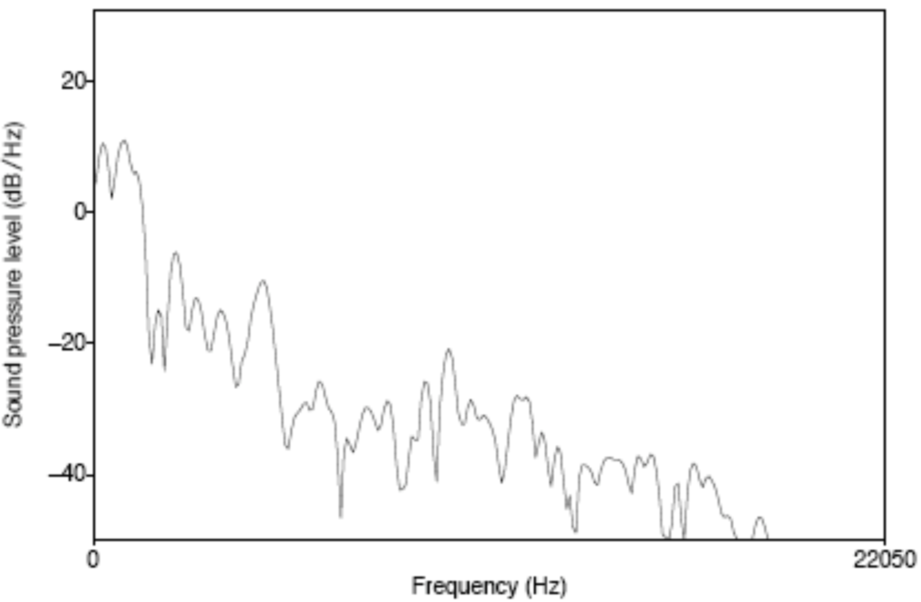


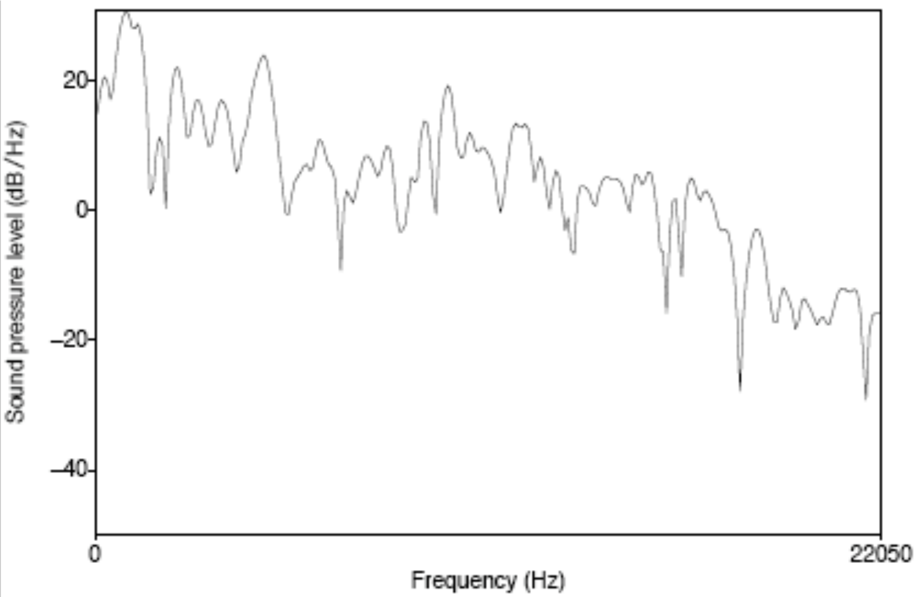
Pre-Emphasis

- **Pre-emphasis: boosting the energy in the high frequencies**
- Q: Why do this?
- A: The spectrum for voiced segments has more energy at lower frequencies than higher frequencies.
  - This is called **spectral tilt**
  - Spectral tilt is caused by the nature of the glottal pulse
- Boosting high-frequency energy gives more info to Acoustic Model
  - Improves phone recognition performance

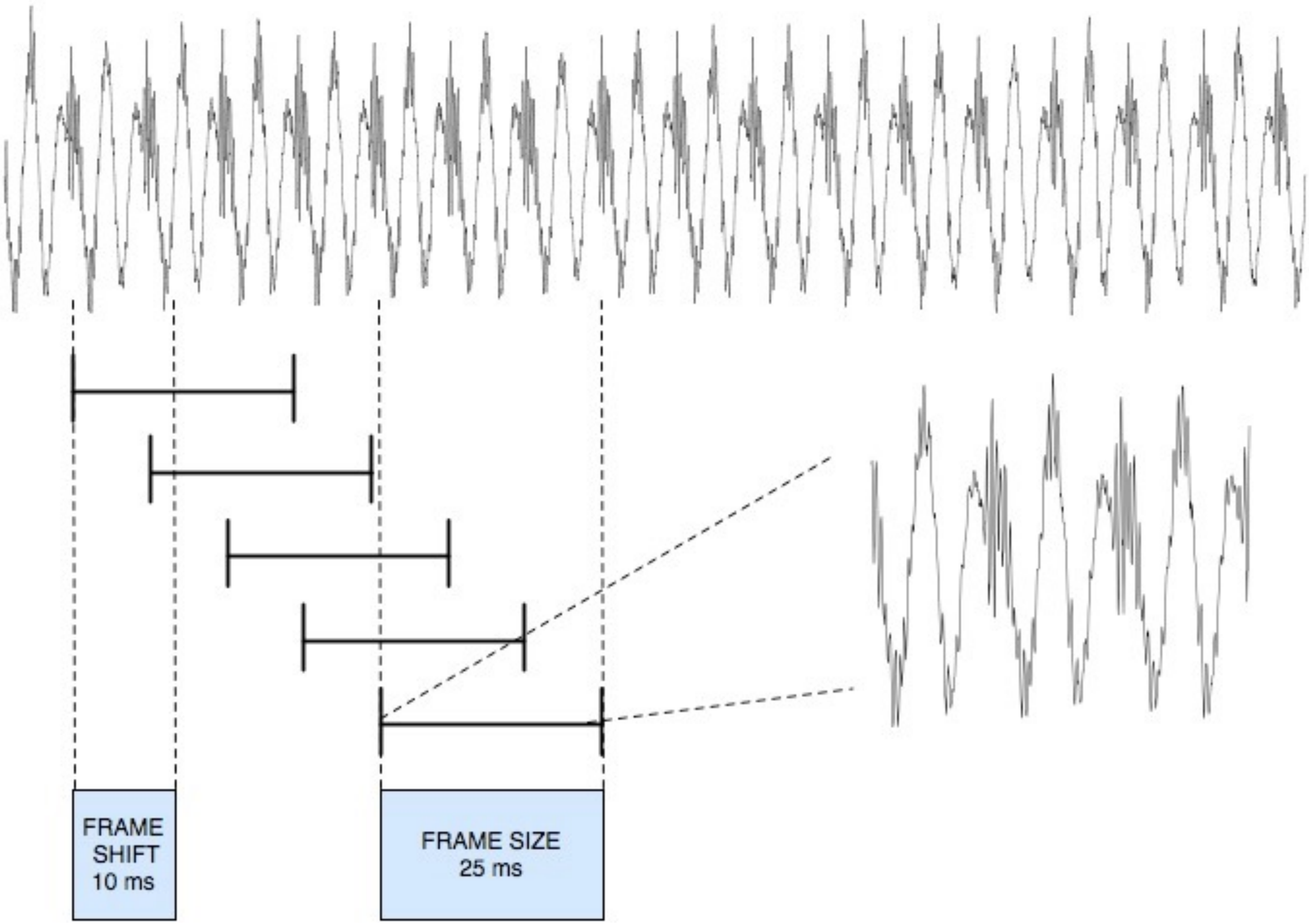
Example of pre-emphasis

- Before and after pre-emphasis
  - Spectral slice from the vowel [aa]





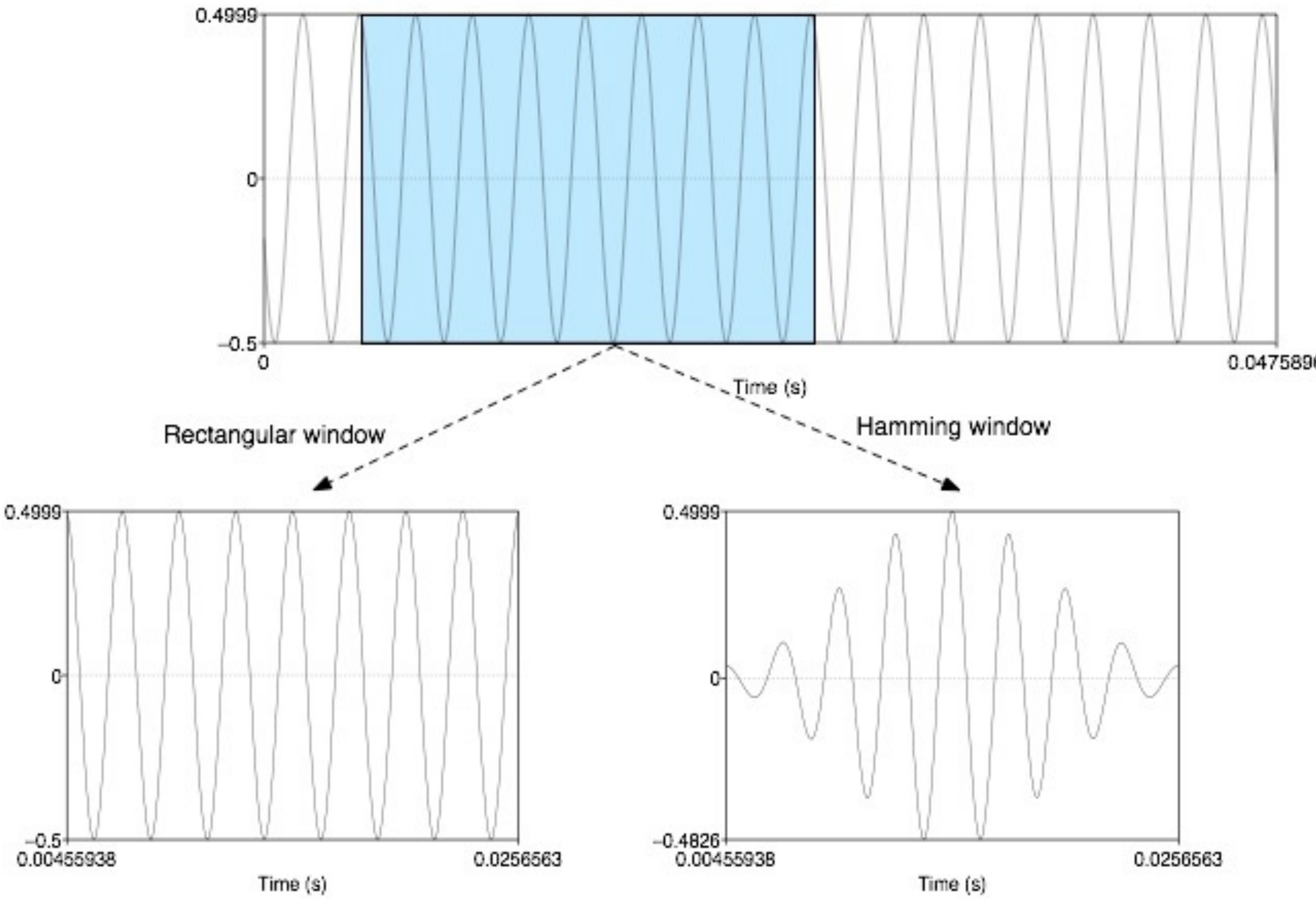
MFCC process: windowing



Windowing

- Why divide speech signal into successive overlapping frames?
  - Speech is not a stationary signal; we want information about a small enough region that the spectral information is a useful cue.
- Frames
  - Frame size: typically, 10-25ms
  - Frame shift: the length of time between successive frames, typically, 5-10ms

MFCC process: windowing



Common window shapes

- Rectangular window:

$$w[n] = \begin{cases} 1 & 0 \leq n \leq L-1 \\ 0 & \text{otherwise} \end{cases}$$

- Hamming window

$$w[n] = \begin{cases} 0.54 - 0.46\cos\left(\frac{2\pi n}{L-1}\right) & 0 \leq n \leq L-1 \\ 0 & \text{otherwise} \end{cases}$$

Discrete Fourier Transform

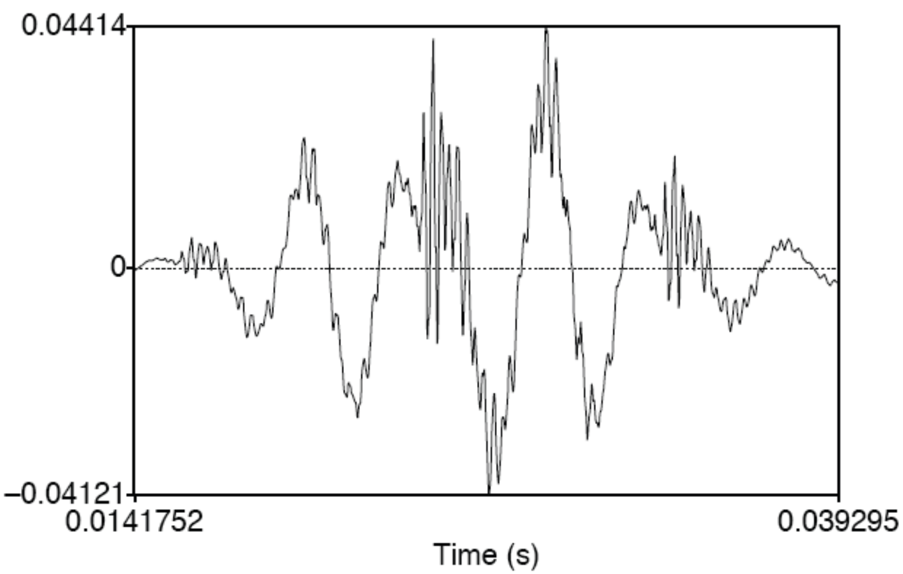
- Input:
  - Windowed signal x[n]…x[m]
- Output:
  - For each of N discrete frequency bands
  - A complex number X[k] representing magnitude and phase of that frequency component in the original signal
- Discrete Fourier Transform (DFT)

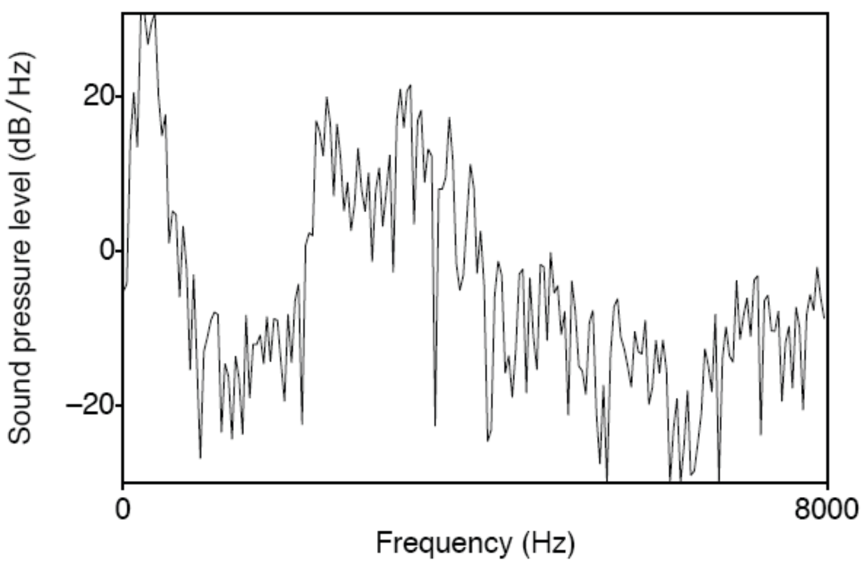
$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\frac{\pi}{N}kn}$$

- Standard algorithm for computing DFT:
  - Fast Fourier Transform (FFT) with complexity N*log(N)
  - In general, choose N=512 or 1024

Discrete Fourier Transform computing a spectrum

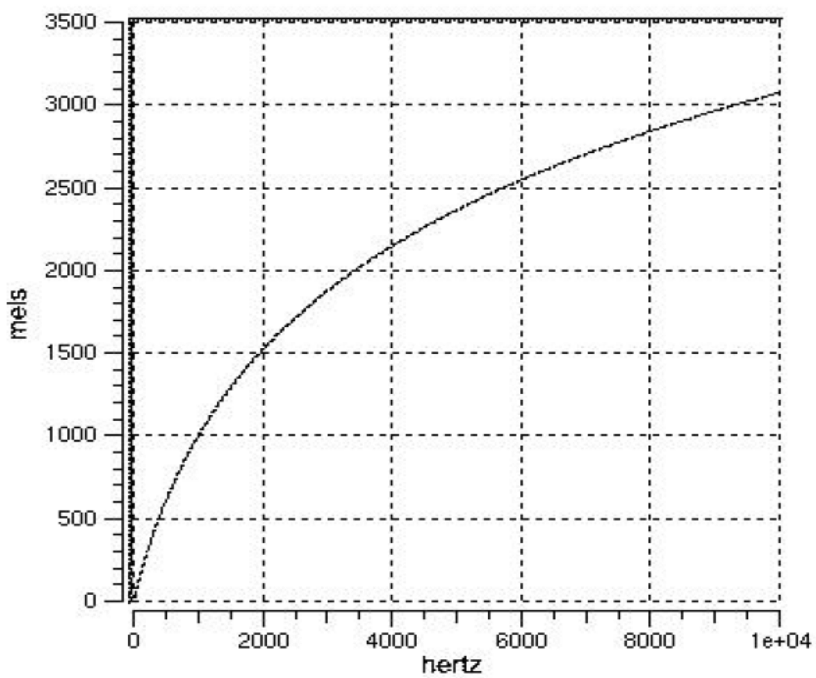
- A 25 ms Hamming-windowed signal from [iy]
  - And its spectrum as computed by DFT (plus other smoothing)





Mel-scale

- Human hearing is not equally sensitive to all frequency bands
- Less sensitive at higher frequencies, roughly > 1000 Hz
- I.e. human perception of frequency is non-linear:



Mel-scale

- A **mel** is a unit of pitch

  Pairs of sounds perceptually equidistant in pitch
  Are separated by an equal number of mels

- Mel-scale is approximately linear below 1 kHz and logarithmic above 1 kHz

- Definition:

$$Mel(f) = 2595 \log_{10} \left(1 + \frac{f}{700}\right)$$

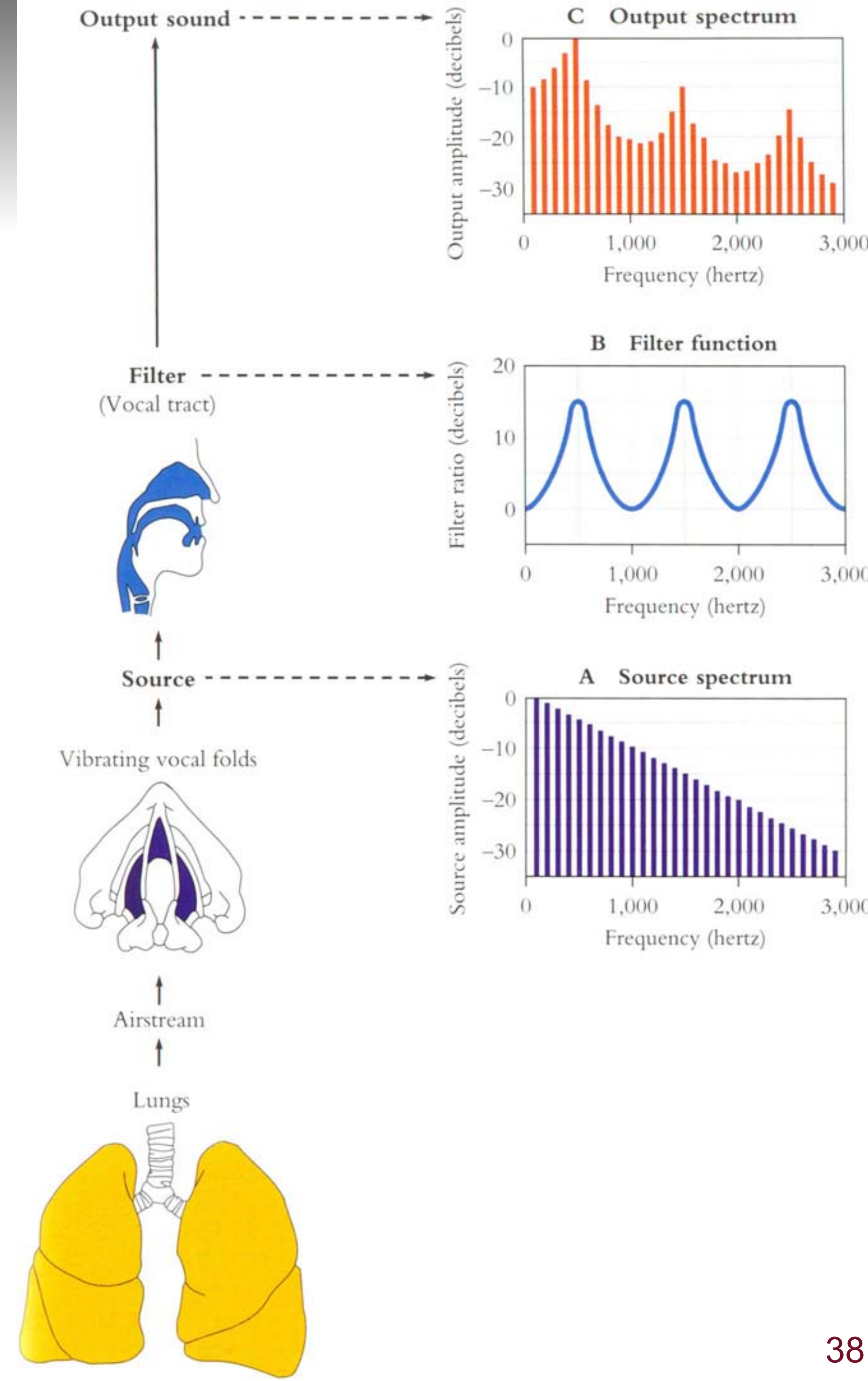
The Cepstrum

- One way to think about this
  - Separating the **source** and **filter**
  - Speech waveform is created by
    - A glottal source waveform
    - Passes through a vocal tract which because of its shape has a particular filtering characteristic
- Articulatory facts:
  - The vocal cord vibrations create harmonics
  - The mouth is an amplifier
  - Depending on shape of oral cavity, some harmonics are amplified more than others

Vocal Fold Vibration

UCLA Phonetics Lab Demo

George Miller figure

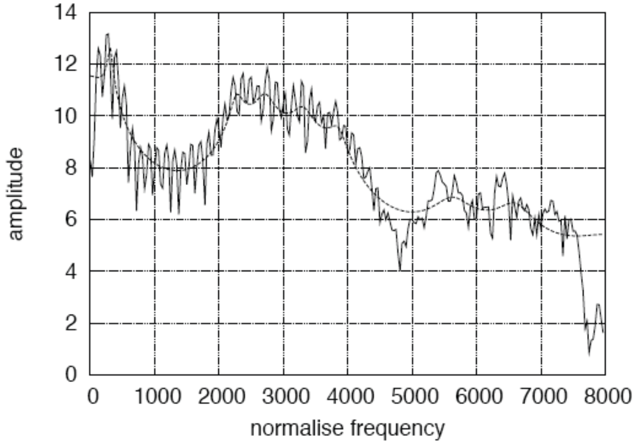


We care about the filter not the source

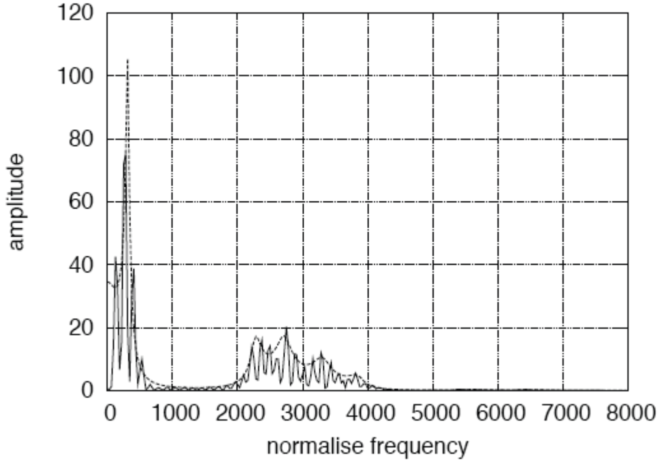
- Most characteristics of the source
  - F0
  - Details of glottal pulse
- Don’t matter for phone detection
- What we care about is the **filter**
  - The exact position of the articulators in the oral tract
- So we want a way to separate these
  - And use only the filter function

The Cepstrum

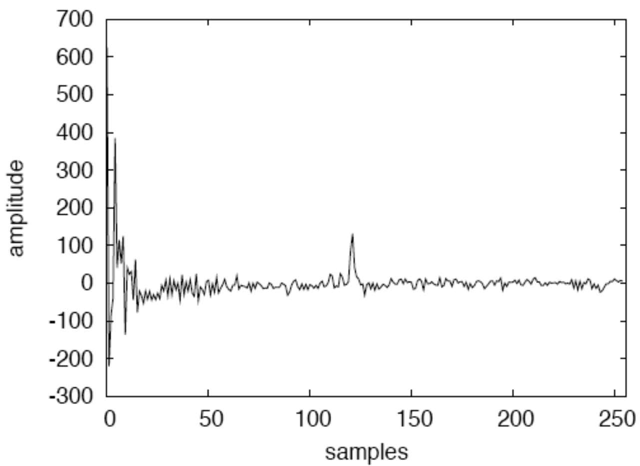
- The spectrum of the log of the spectrum



Spectrum

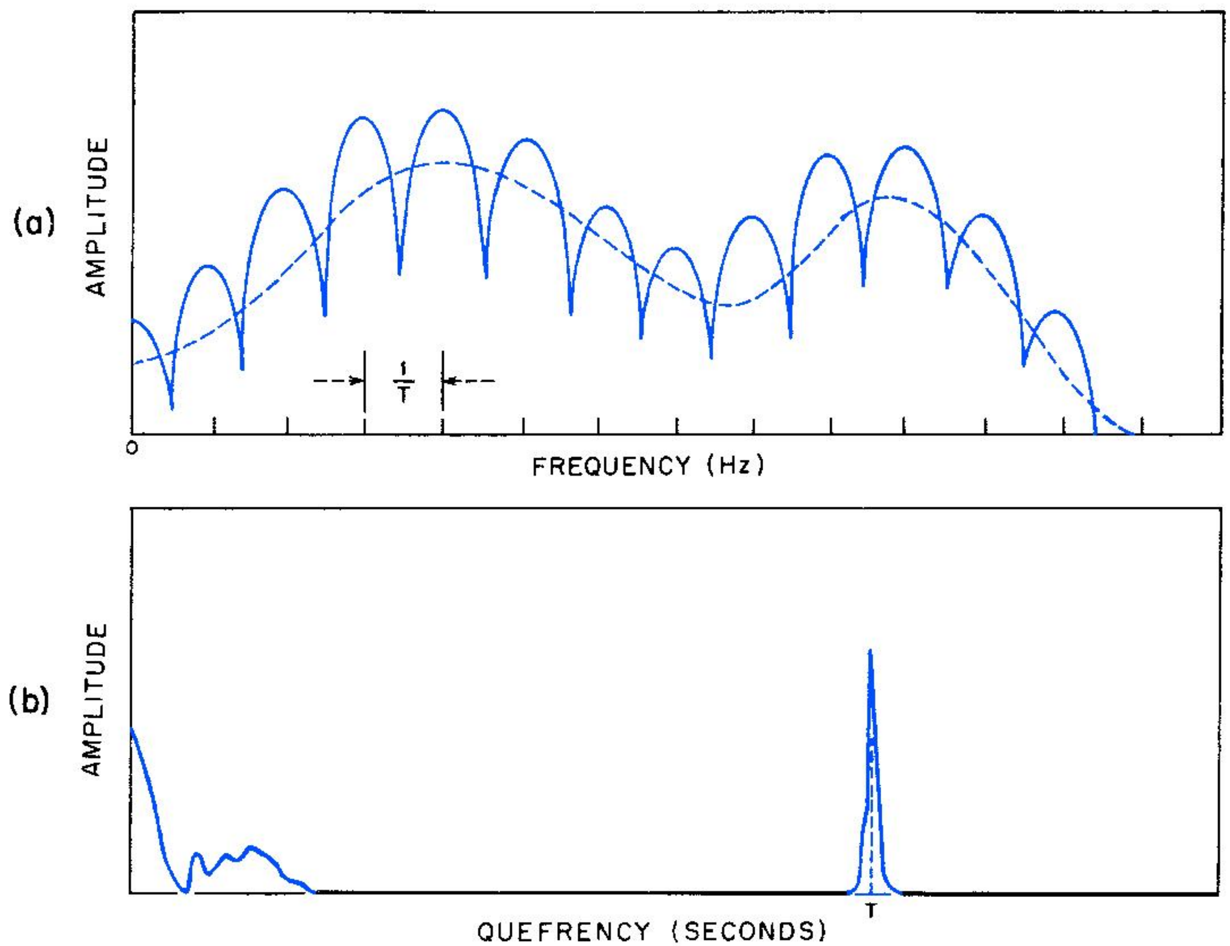


Log spectrum



Spectrum of log spectrum

Thinking about the Cepstrum



Pictures from John Coleman (2005)

Mel Frequency cepstrum

- The cepstrum requires Fourier analysis
- But we're going from frequency space back to time
- So we actually apply inverse DFT

$$y_t[k] = \sum_{m=1}^{M} \log(|Y_t(m)|) \cos(k(m-0.5)\frac{\pi}{M}), \text{ k=0,...,J}$$

- Details for signal processing gurus: Since the log power spectrum is real and symmetric, inverse DFT reduces to a Discrete Cosine Transform (DCT)

Another advantage of the Cepstrum

- DCT produces highly **uncorrelated** features
- We'll see when we get to acoustic modeling that these will be much easier to model than the spectrum
  - Simply modelled by linear combinations of Gaussian density functions with diagonal covariance matrices
- In general we'll just use the first 12 cepstral coefficients (we don't want the later ones which have e.g. the F0 spike)

Dynamic Cepstral Coefficient

- The cepstral coefficients do not capture energy
- So we add an energy feature $$Energy = \sum_{t=t_1}^{t_2} x^2[t]$$
- Also, we know that speech signal is not constant (slope of formants, change from stop burst to release).
- So we want to add the changes in features (the slopes).
- We call these **delta** features
- We also add **double-delta** acceleration features

Typical MFCC features

- Window size: 25ms
- Window shift: 10ms
- Pre-emphasis coefficient: 0.97
- MFCC:
  - 12 MFCC (mel frequency cepstral coefficients)
  - 1 energy feature
  - 12 delta MFCC features
  - 12 double-delta MFCC features
  - 1 delta energy feature
  - 1 double-delta energy feature
- Total 39-dimensional features

Why is MFCC so popular?

- Efficient to compute
- Incorporates a perceptual Mel frequency scale
- Separates the source and filter
- IDFT(DCT) decorrelates the features
  - Improves diagonal assumption in HMM modeling

Coming up: Acoustic Modeling (= Phone detection)

- Given a 39-dimensional vector corresponding to the observation of one frame o_i
- And given a phone q we want to detect
- Compute $p(o_i|q)$
- Most popular method:
  - GMM (Gaussian mixture models)
- Other methods
  - Neural nets, CRFs, SVM, etc

Summary

- ASR Architecture
  - The Noisy Channel Model
- Five easy pieces of an ASR system
  1) Language Model
  2) Lexicon/Pronunciation Model (HMM)
  3) Feature Extraction
  4) Acoustic Model
  5) Decoder
- Training
- Evaluation