Components: ASR

Ling575
Spoken Dialog Systems
April 17, 2013

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

Why is conversational speech harder?

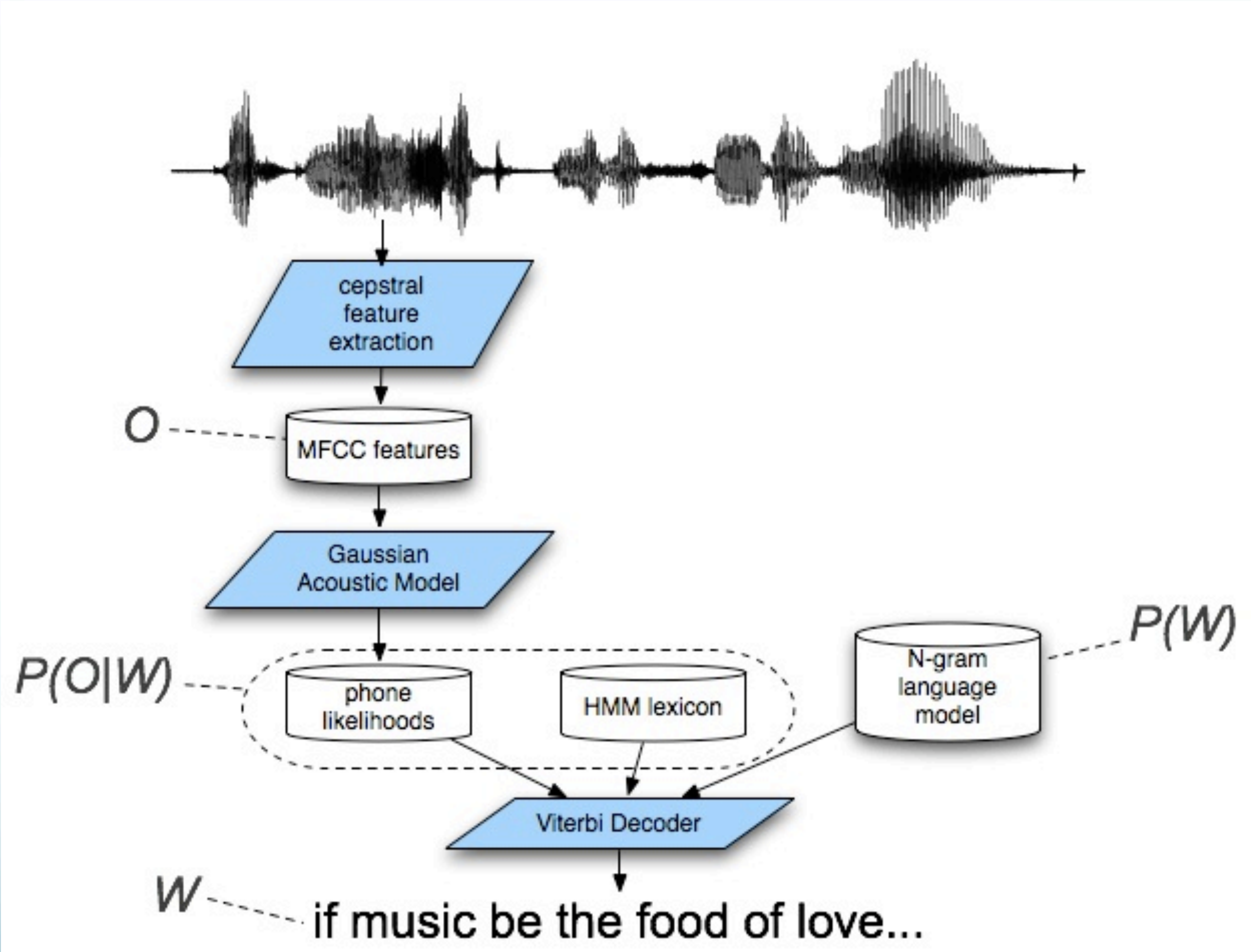
- A piece of an utterance without context

- The same utterance with more context

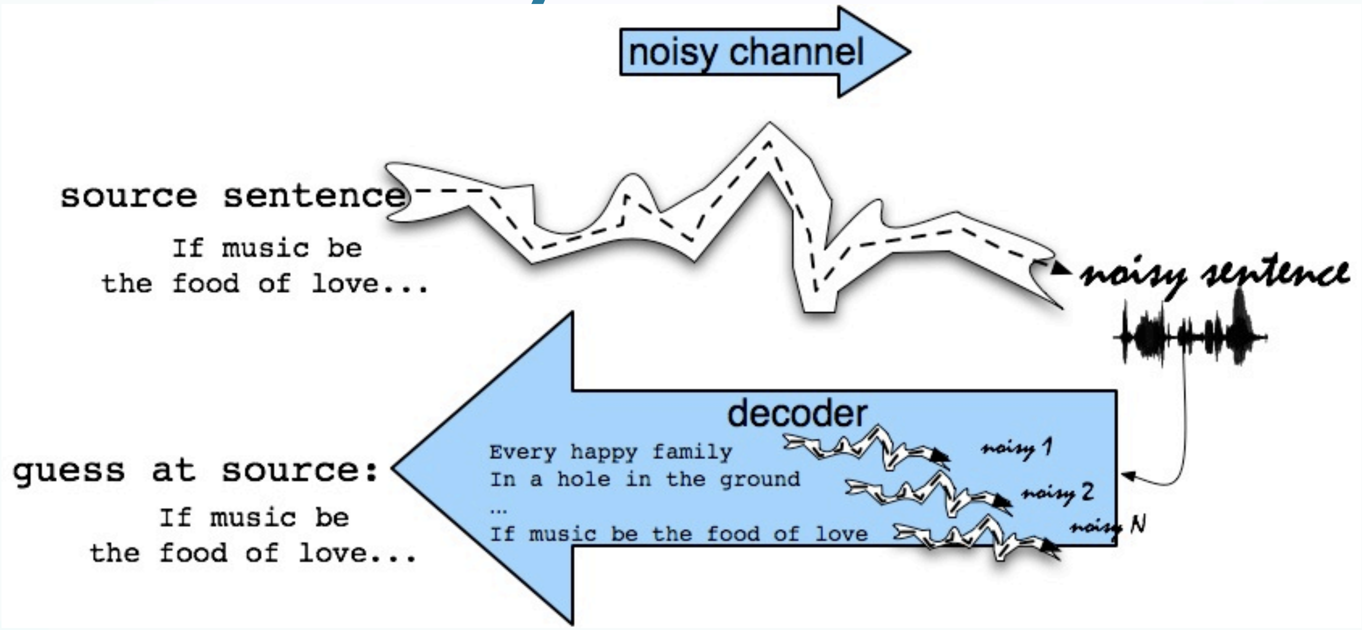
LVCSR Design Intuition

- 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



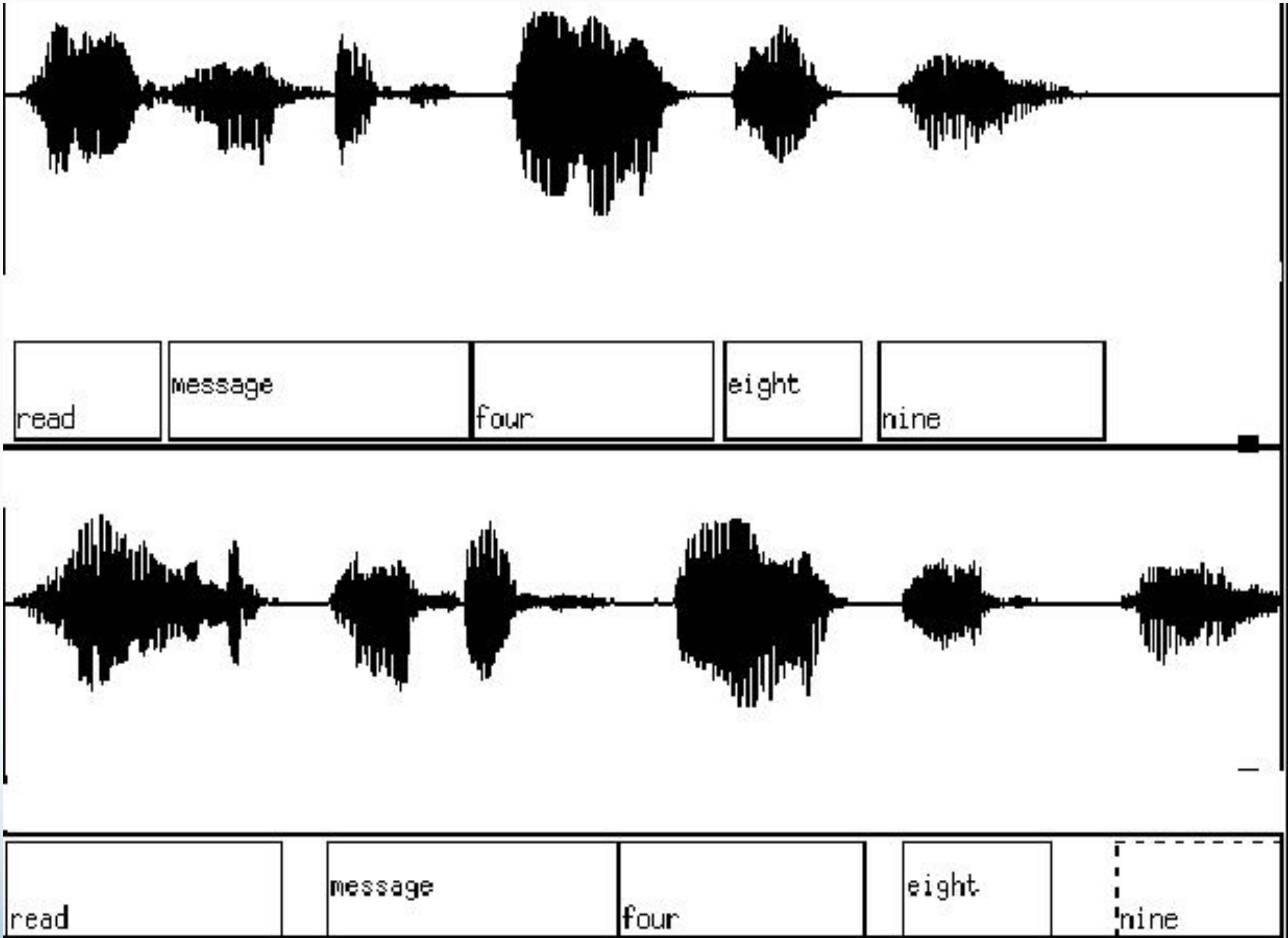
- Search through space of all possible sentences.
- Pick the one that is most probable given the waveform.

Decomposing Speech Recognition

- Q1: What speech sounds were uttered?
  - Human languages: 40-50 phones
    - Basic sound units: b, m, k, ax, ey, …(arpabet)
    - Distinctions categorical to speakers
      - Acoustically continuous
    - Part of knowledge of language
      - Build per-language inventory
      - Could we learn these?

Decomposing Speech Recognition

- Q2: What words produced these sounds?
  - Look up sound sequences in dictionary
  - Problem 1: Homophones
    - Two words, same sounds: too, two
  - Problem 2: Segmentation
    - No “space” between words in continuous speech
    - “I scream”/”ice cream”, “Wreck a nice beach”/”Recognize speech”
- Q3: What meaning produced these words?
  - NLP (But that’s not all!)


read
message
four
eight
nine
read
message
four
eight
nine

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:

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

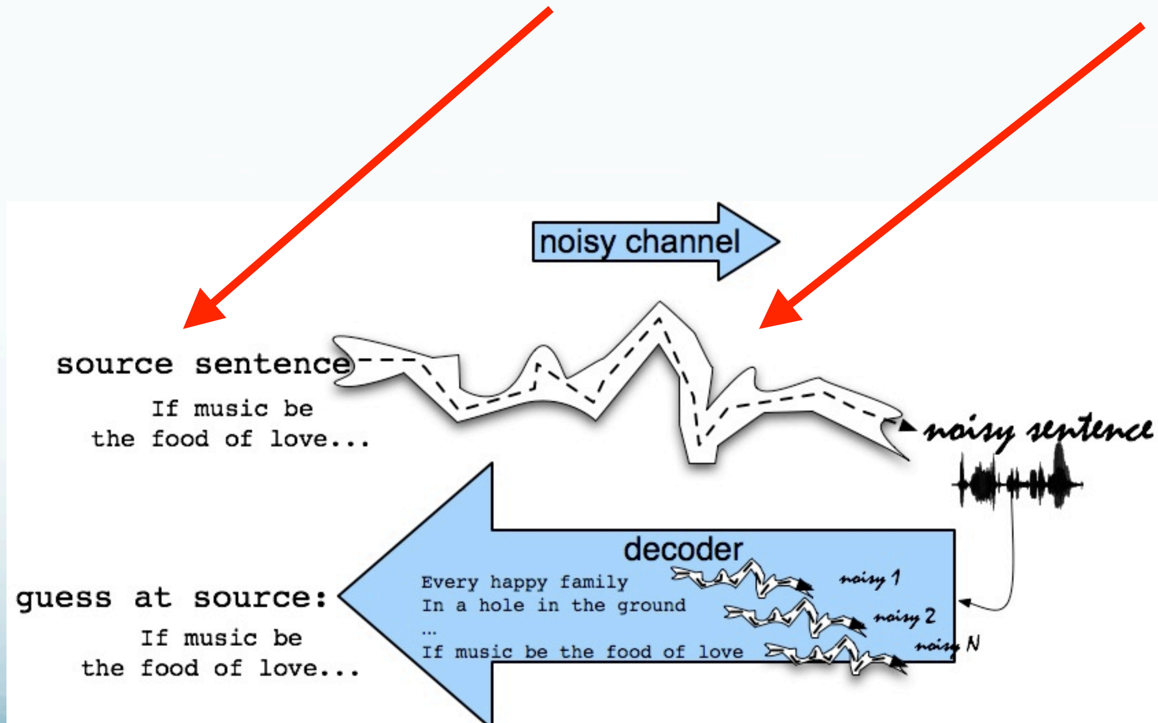
Noisy channel model

likelihood prior

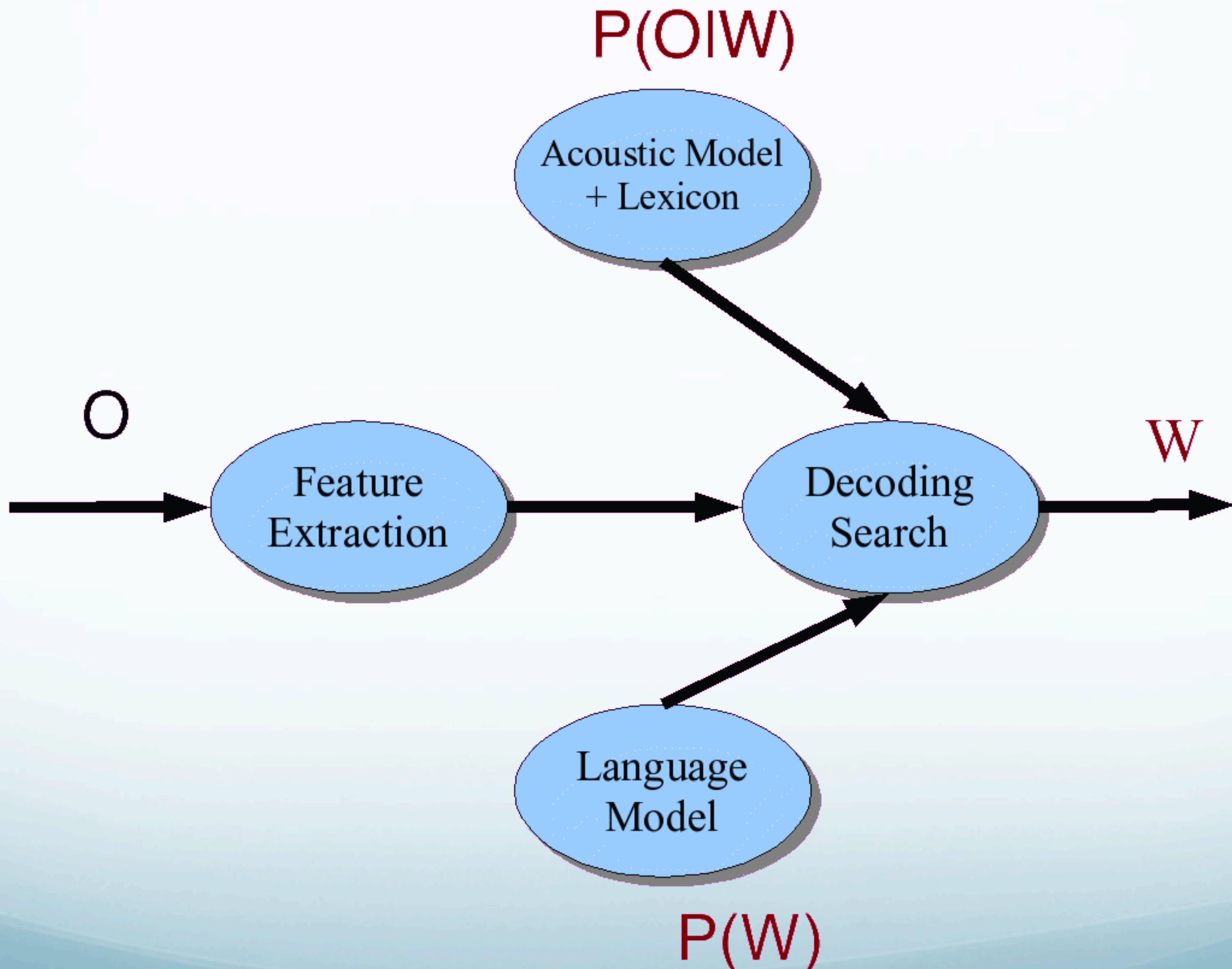
$$\hat{W} = \underset{W \in L}{\operatorname{argmax}}\, P(O \mid W)\, P(W)$$

The noisy channel model

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



Speech Architecture meets Noisy Channel



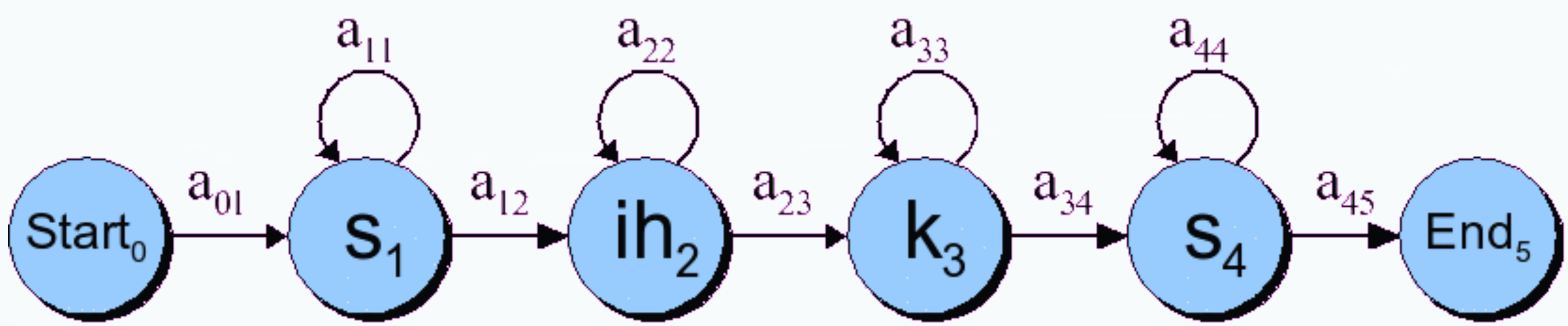
ASR Components

- Lexicons and Pronunciation:
  - Hidden Markov Models
- Feature extraction
- Acoustic Modeling
- Decoding
- Language Modeling:
  - Ngram Models

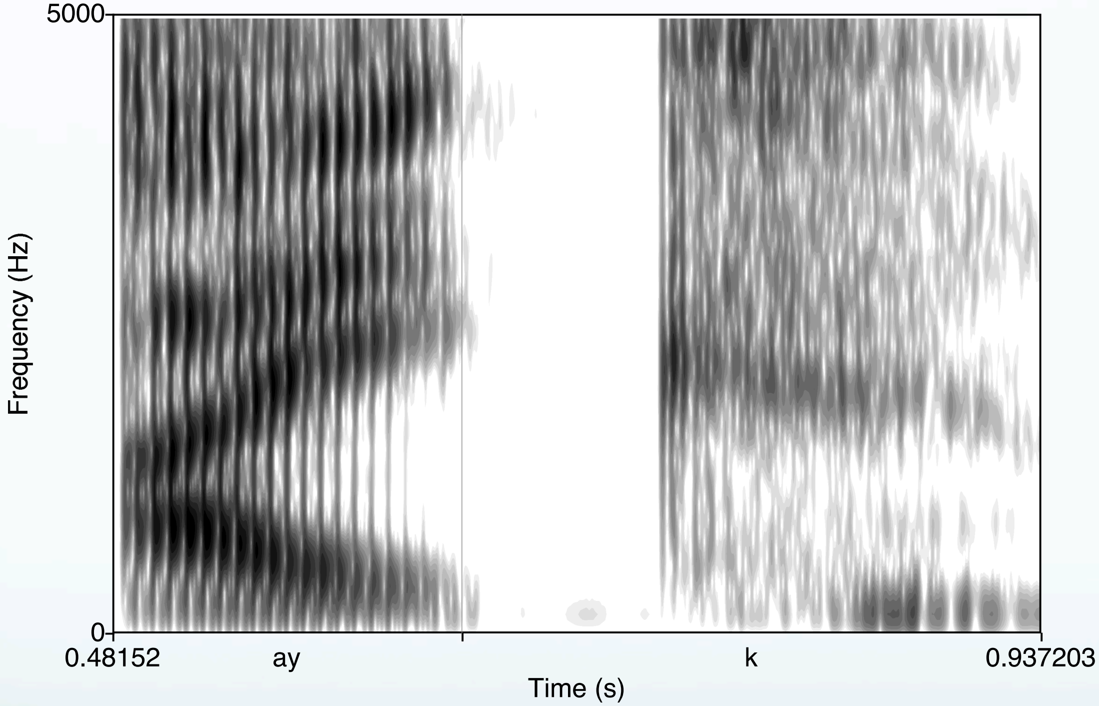
Lexicon

- A list of words
- Each one with a pronunciation in terms of phones
- We get these from 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

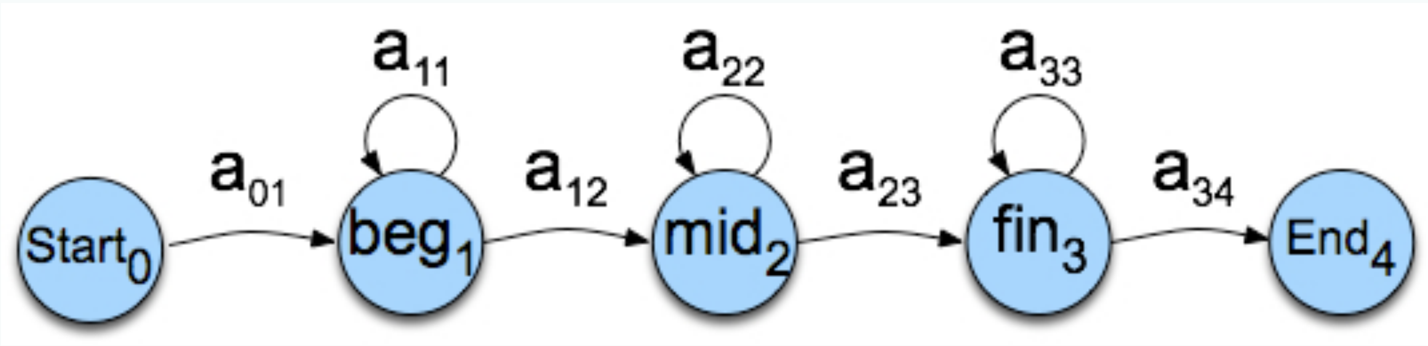
HMMs for speech: the word "six"



Phones are not homogeneous!

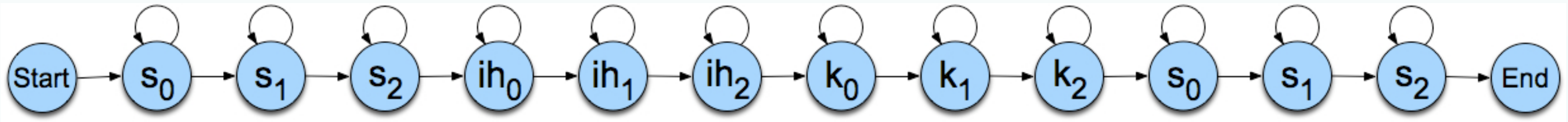


Each phone has 3 subphones



HMM word model for "six"

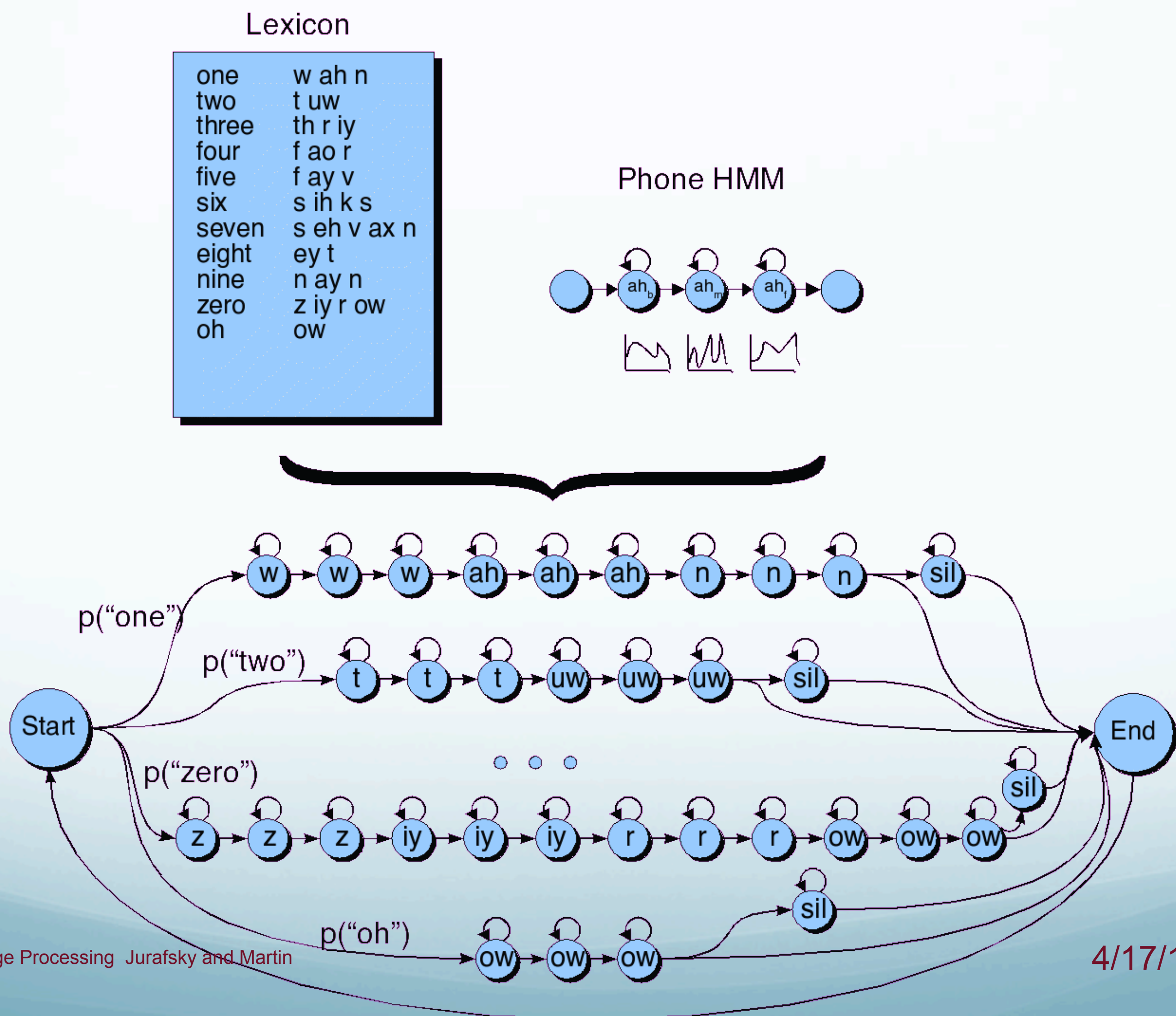
- Resulting model with subphones



HMMs for speech

$Q = q_1 q_2 \ldots q_N$	a set of **states** corresponding to **subphones**
$A = a_{01} a_{02} \ldots a_{n1} \ldots a_{nn}$	a **transition probability matrix** A, each a_{ij} representing the probability for each subphone of taking a **self-loop** or going to the next subphone. Together, Q and A implement a **pronunciation lexicon**, an HMM state graph structure for each word that the system is capable of recognizing.
$B = b_i(o_t)$	A set of **observation likelihoods:**, also called **emission probabilities**, each expressing the probability of a cepstral feature vector (observation o_t) being generated from subphone state i.

HMM for the digit recognition task



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
- Fits well with HMM modelling

Decoding

- In principle:

$$\hat{W} = \underset{W \in \mathcal{L}}{\operatorname{argmax}} \overbrace{P(O|W)}^{\text{likelihood}} \overbrace{P(W)}^{\text{prior}}$$

- In practice:

$$\hat{W} = \underset{W \in \mathcal{L}}{\operatorname{argmax}} P(O|W)P(W)^{LMSF}$$

$$\hat{W} = \underset{W \in \mathcal{L}}{\operatorname{argmax}} P(O|W)P(W)^{LMSF} WIP^{N}$$

$$\hat{W} = \underset{W \in \mathcal{L}}{\operatorname{argmax}} \log P(O|W) + LMSF \times \log P(W) + N \times \log WIP$$

Why is ASR decoding hard?

[ay d ih s hh er d s ah m th ih ng ax b aw m uh v ih ng r ih s en l ih]

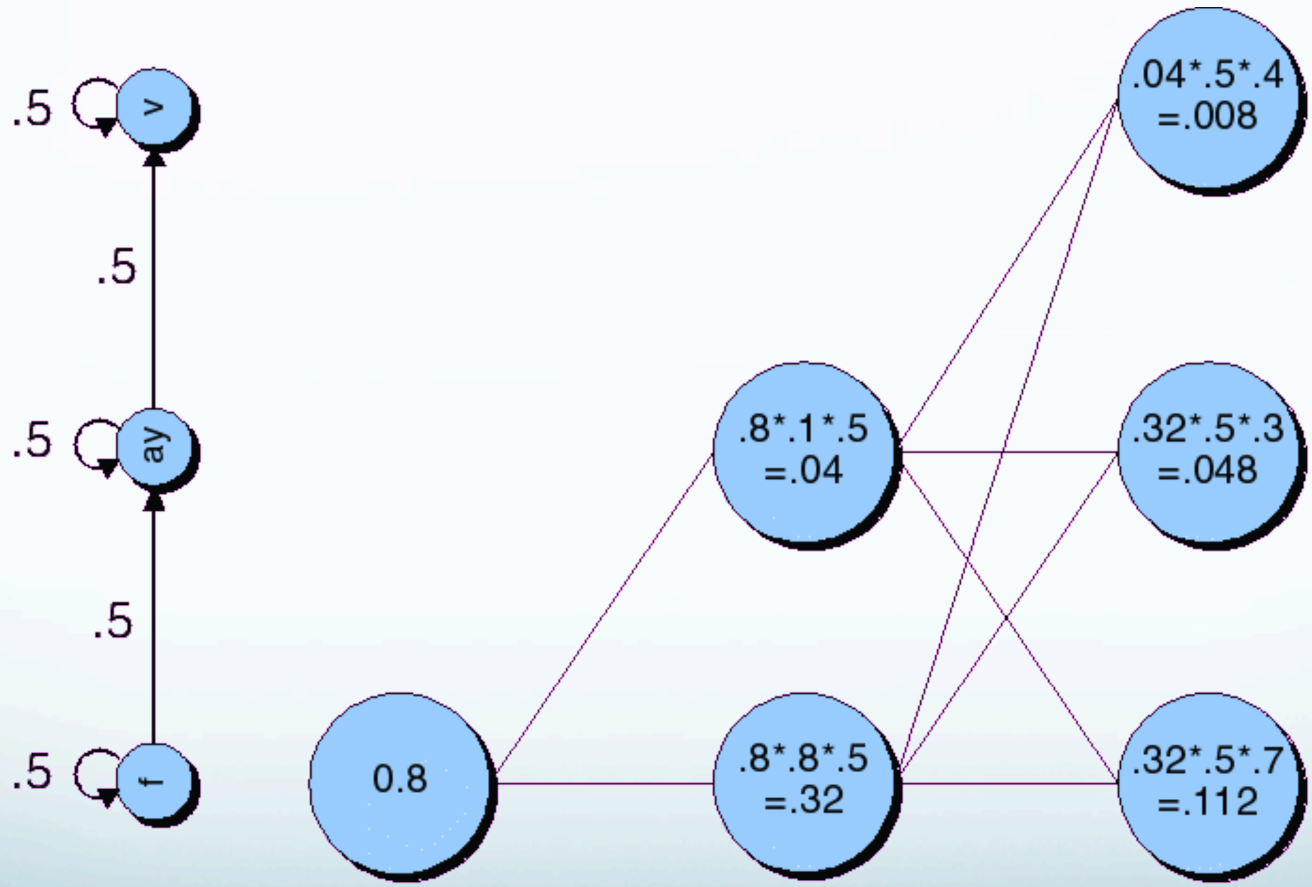
The Evaluation (forward) problem for speech

- The observation sequence O is a series of MFCC vectors
- The hidden states W are the phones and words
- For a given phone/word string W, our job is to evaluate P(O|W)
- Intuition: how likely is the input to have been generated by just that word string W

Evaluation for speech: Summing over all different paths!

- f ay ay ay ay v v v v
- f f ay ay ay ay v v v
- f f f f ay ay ay ay v
- f f ay ay ay ay ay ay v
- f f ay ay ay ay ay ay ay ay v
- f f ay v v v v v v v

Viterbi trellis for "five"



Viterbi trellis for "five"

V	0	0	0.008	0.0072	0.00672	0.00403	0.00188	0.00161	0.000667	0.000493
AY	0	0.04	0.048	0.0448	0.0269	0.0125	0.00538	0.00167	0.000428	8.78e-05
F	0.8	0.32	0.112	0.0224	0.00448	0.000896	0.000179	4.48e-05	1.12e-05	2.8e-06
Time	1	2	3	4	5	6	7	8	9	10
B	*f* 0.8	*f* 0.8	*f* 0.7	*f* 0.4	*f* 0.4	*f* 0.4	*f* 0.4	*f* 0.5	*f* 0.5	*f* 0.5
	ay 0.1	*ay* 0.1	*ay* 0.3	*ay* 0.8	*ay* 0.8	*ay* 0.8	*ay* 0.8	*ay* 0.6	*ay* 0.5	*ay* 0.4
	v 0.6	*v* 0.6	*v* 0.4	*v* 0.3	*v* 0.3	*v* 0.3	*v* 0.3	*v* 0.6	*v* 0.8	*v* 0.9
	p 0.4	*p* 0.4	*p* 0.2	*p* 0.1	*p* 0.1	*p* 0.1	*p* 0.1	*p* 0.1	*p* 0.3	*p* 0.3
	iy 0.1	*iy* 0.1	*iy* 0.3	*iy* 0.6	*iy* 0.6	*iy* 0.6	*iy* 0.6	*iy* 0.5	*iy* 0.5	*iy* 0.4

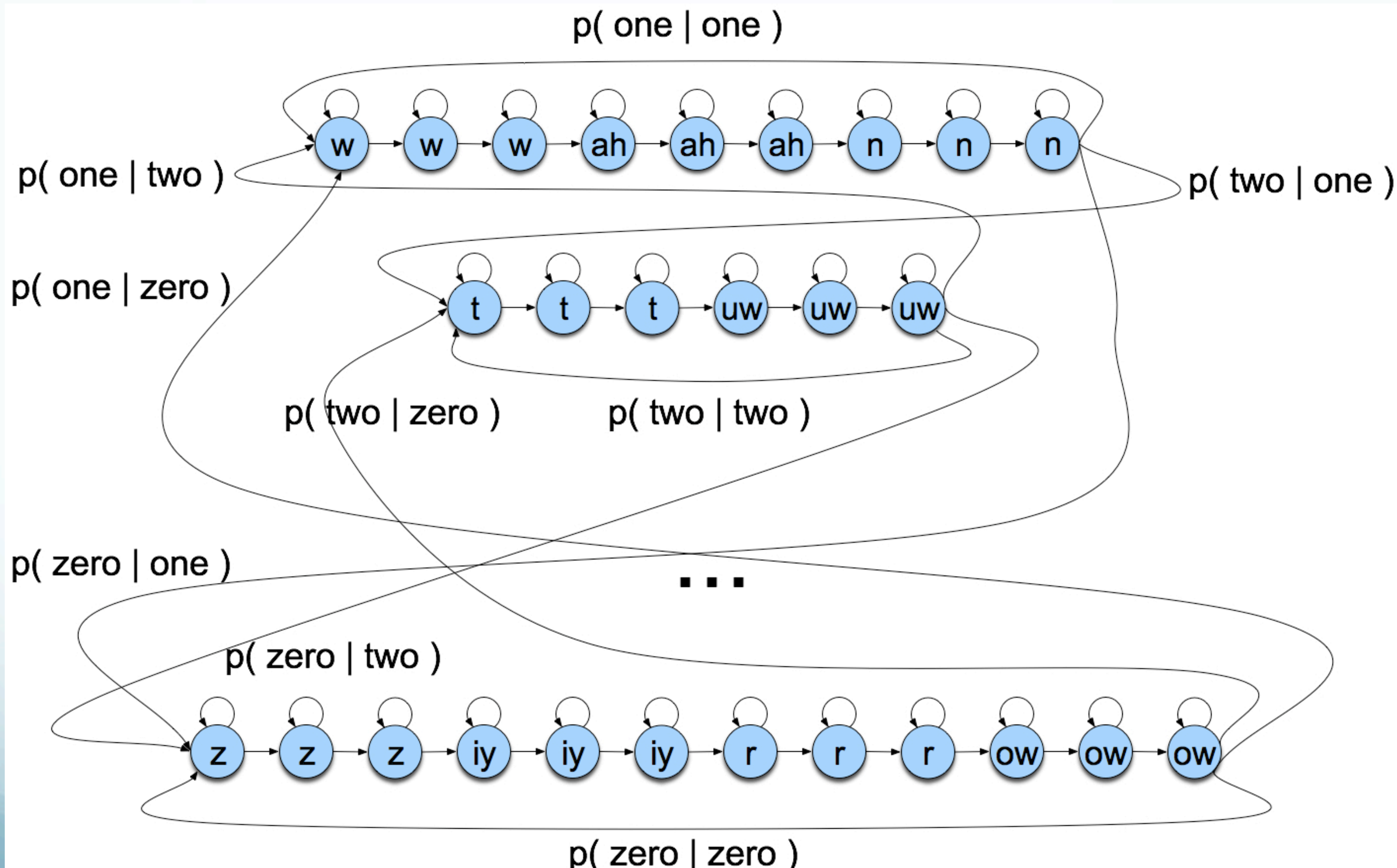
Language Model

- Idea: some utterances more probable

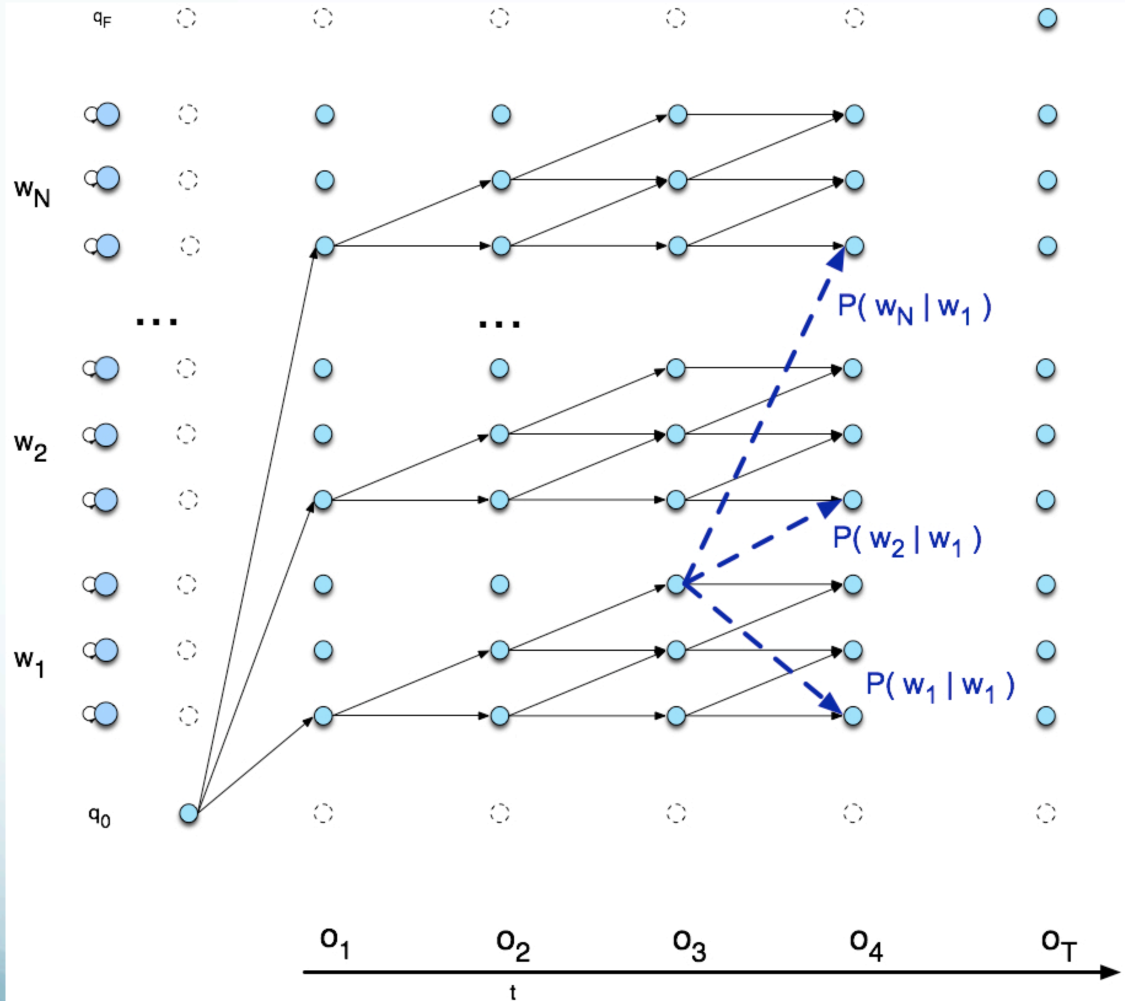
- Standard solution: "n-gram" model
  - Typically tri-gram: $P(w_i|w_{i-1},w_{i-2})$
    - Collect training data from large side corpus
      - Smooth with bi- & uni-grams to handle sparseness
  - Product over words in utterance:

$$P(w_1^n) \approx \prod_{k=1}^{n} P(w_k \mid w_{k-1}, w_{k-2})$$

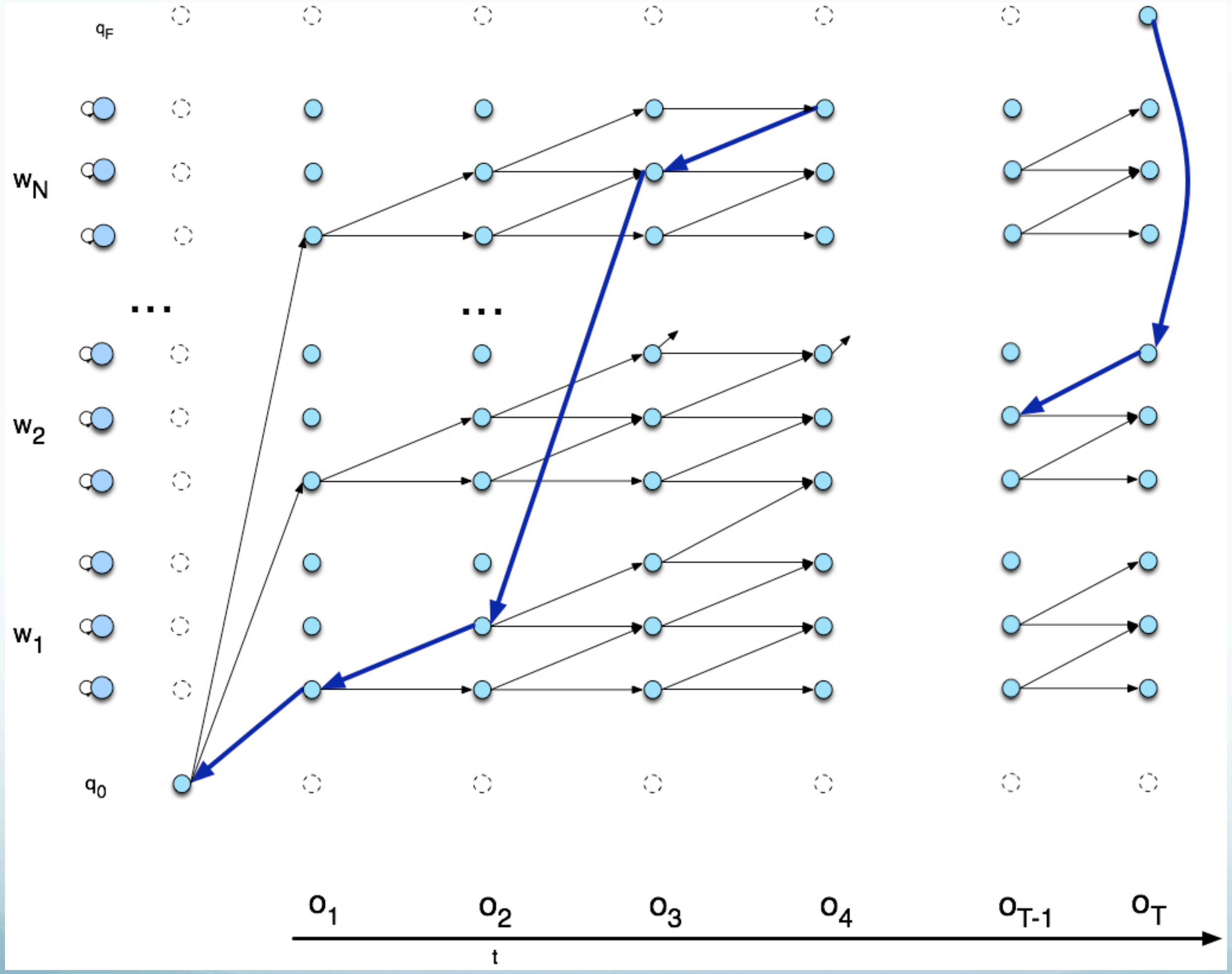
Search space with bigrams



Viterbi trellis



Viterbi backtrace

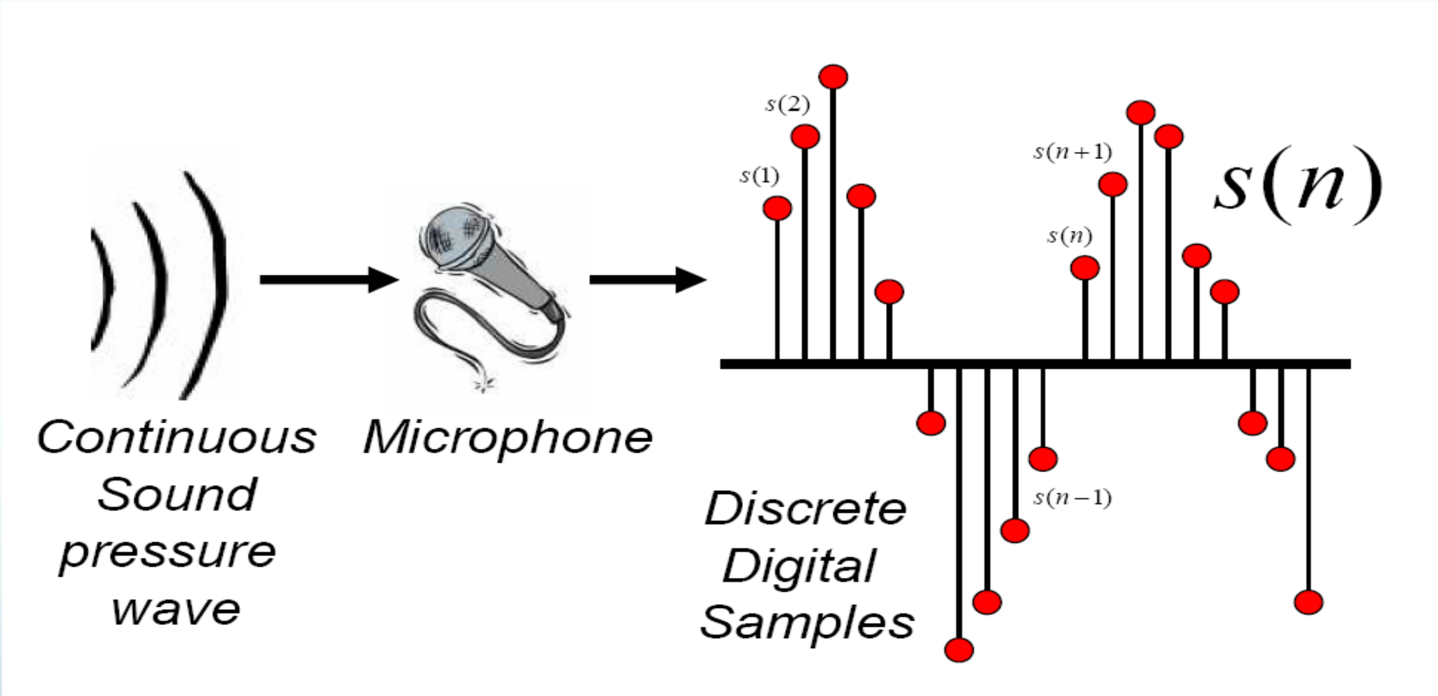


Detecting Phones

- Two stages
  - **Feature extraction**
    - Basically a slice of a spectrogram
  - **Phone classification**
    - Using GMM classifier

Discrete Representation of Signal

- Represent continuous signal into discrete form.



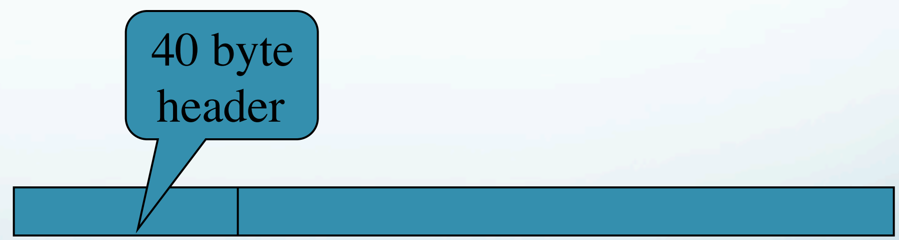
Thanks to Bryan Pellom for this slide

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

Digitizing Speech (II)

- **Quantization**
  - Representing real value of each amplitude as integer
  - 8-bit (-128 to 127) or 16-bit (-32768 to 32767)
- **Formats**:
  - 16 bit PCM
  - 8 bit mu-law; log compression
- LSB (Intel) vs. MSB (Sun, Apple)
- **Headers**:
  - Raw (no header)
  - Microsoft wav →
  - Sun .au



MFCC: Mel-Frequency Cepstral Coefficients