

# CS 224S / LINGUIST 236 Speech Recognition and Synthesis

Dan Jurafsky

## Lecture 7: Intro to ASR+HMMs: History+ Forward and Viterbi

IP Notice:

1/25/05

CS 224S Winter 2005

1

## Outline for Today

- **Speech Recognition Architectural Overview**
- **Hidden Markov Models in general**
  - Forward
  - Viterbi Decoding
- **HMMs for speech: structure**
- **How this fits into the ASR component of course**
  - 1/26: Baum-Welch (EM) training of HMMs
  - 2/1: Acoustic Model estimation; Gaussians, triphones, etc
  - 2/3: Advanced Issues in Acoustic Mod.: Guest Lecture
  - 2/8: Language Modeling: Lecture by Rion!
  - 2/10: Advanced Issues in Decoding Search

1/25/05

CS 224S Winter 2005

2

## LVCSR

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

1/25/05

CS 224S Winter 2005

3

## LVCSR Design Intuition

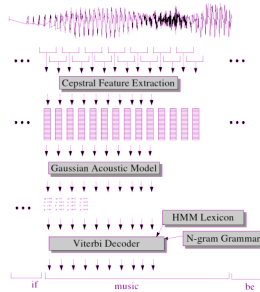
- **Build a statistical model of the speech-to-words process**
- **Collect lots and lots of speech, and transcribe all the words.**
- **Train the model on the labeled speech**
- **Paradigm: Supervised Machine Learning + Search**

1/25/05

CS 224S Winter 2005

4

## Speech Recognition Architecture



1/25/05

CS 224S Winter 2005

5

## The Noisy Channel Model



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

1/25/05

CS 224S Winter 2005

6

## 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
  - $O = o_1, o_2, o_3, \dots, o_t$
- Define a sentence as a sequence of words:
  - $W = w_1, w_2, w_3, \dots, w_n$

1/25/05

CS 224S Winter 2005

7

## Noisy Channel Model (III)

- Probabilistic implication: Pick the highest prob  $S$ :

$$\hat{W} = \operatorname{argmax}_{W \in L} P(W | O)$$

- We can use Bayes rule to rewrite this:

$$\hat{W} = \operatorname{argmax}_{W \in L} \frac{P(O|W)P(W)}{P(O)}$$

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

$$\hat{W} = \operatorname{argmax}_{W \in L} P(O|W)P(W)$$

1/25/05

CS 224S Winter 2005

8

## Noisy channel model

$$\hat{W} = \operatorname{argmax}_{W \in L} P(O|W)P(W)$$

likelihood      prior  
↓                      ↓

1/25/05

CS 224S Winter 2005

9

## The noisy channel model

- Ignoring the denominator leaves us with two factors:  $P(\text{Source})$  and  $P(\text{Signal}|\text{Source})$

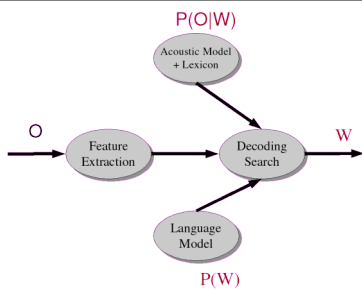


1/25/05

CS 224S Winter 2005

10

## Speech Architecture meets Noisy Channel



1/25/05

CS 224S Winter 2005

11

## Architecture: Five easy pieces

- Feature extraction
- Acoustic Modeling
- HMMs, Lexicons, and Pronunciation
- Decoding
- Language Modeling

1/25/05

CS 224S Winter 2005

12

## Feature Extraction

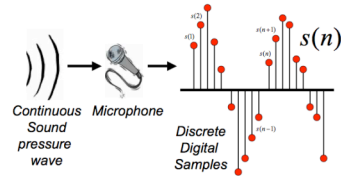
- Digitize Speech
- Extract Frames

1/25/05

CS 224S Winter 2005

13

## Digitizing Speech



Thanks to Bryan Pellom for this slide!

1/25/05

CS 224S Winter 2005

14

## Digitizing Speech (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

1/25/05

CS 224S Winter 2005

15

## 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

1/25/05

CS 224S Winter 2005

16

## Frame Extraction

- A frame (25 ms wide) extracted every 10 ms

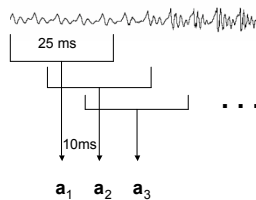


Figure from Simon Arnfield

1/25/05

CS 224S Winter 2005

17

## MFCC (Mel Frequency Cepstral Coefficients)

- Do FFT to get spectral information
  - Like the spectrogram/spectrum we saw earlier
- Apply Mel scaling
  - Linear below 1kHz, log above, equal samples above and below 1kHz
  - Models human ear; more sensitivity in lower freqs
- Plus Discrete Cosine Transformation

1/25/05

CS 224S Winter 2005

18

## Final Feature Vector

- 39 Features per 10 ms frame:
  - 12 MFCC features
  - 12 Delta MFCC features
  - 12 Delta-Delta MFCC features
  - 1 (log) frame energy
  - 1 Delta (log) frame energy
  - 1 Delta-Delta (log frame energy)
- So each frame represented by a 39D vector

1/25/05

CS 224S Winter 2005

19

## Where we are

- Given: a sequence of acoustic feature vectors, one every 10 ms
- Goal: output a string of words
- We'll spend 6 lectures on how to do this
- Rest of today:
  - Markov Models
  - Hidden Markov Models in the abstract
    - Forward Algorithm
    - Viterbi Algorithm
  - Start of HMMs for speech

1/25/05

CS 224S Winter 2005

20

## First-order observable Markov Model

- a set of states
  - $Q = q_1, q_2, \dots, q_N$ ; the state at time  $t$  is  $q_t$
- Current state only depends on previous state
 
$$P(q_i | q_1, \dots, q_{i-1}) = P(q_i | q_{i-1})$$
- Transition probability matrix  $A$ 

$$a_{ij} = P(q_t = j | q_{t-1} = i) \quad 1 \leq i, j \leq N$$
- Special initial probability vector  $\pi$ 

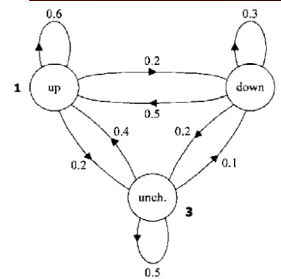
$$\pi_i = P(q_1 = i) \quad 1 \leq i \leq N$$
- Constraints:
 
$$\sum_{j=1}^N a_{ij} = 1; \quad 1 \leq i \leq N \quad \sum_{j=1}^N \pi_j = 1$$

1/25/05

CS 224S Winter 2005

21

## Markov model for Dow Jones



Initial state probability matrix

$$\pi = (\pi_i) = \begin{pmatrix} 0.5 \\ 0.2 \\ 0.3 \end{pmatrix}$$

State-transition probability matrix

$$A = \{a_{ij}\} = \begin{bmatrix} 0.6 & 0.2 & 0.2 \\ 0.5 & 0.3 & 0.2 \\ 0.4 & 0.1 & 0.5 \end{bmatrix}$$

Figure from Huang et al, via

## Markov Model for Dow Jones

- What is the probability of 5 consecutive up days?
- Sequence is up-up-up-up-up
- I.e., state sequence is 1-1-1-1-1
- $P(1,1,1,1,1) =$ 
  - $\pi_1 a_{11} a_{11} a_{11} a_{11} = 0.5 \times (0.6)^4 = 0.0648$

1/25/05

CS 224S Winter 2005

23

## Hidden Markov Models

- a set of states
  - $Q = q_1, q_2, \dots, q_N$ ; the state at time  $t$  is  $q_t$
- Transition probability matrix  $A = \{a_{ij}\}$ 

$$a_{ij} = P(q_t = j | q_{t-1} = i) \quad 1 \leq i, j \leq N$$
- Output probability matrix  $B = \{b_i(k)\}$ 

$$b_i(k) = P(X_t = o_k | q_t = i)$$
- Special initial probability vector  $\pi$ 

$$\pi_i = P(q_1 = i) \quad 1 \leq i \leq N$$
- Constraints:
 
$$\sum_{j=1}^N a_{ij} = 1; \quad 1 \leq i \leq N \quad \sum_{k=1}^M b_i(k) = 1 \quad \sum_{j=1}^N \pi_j = 1$$

1/25/05

CS 224S Winter 2005

24

## Assumptions

- **Markov assumption:**  

$$P(q_i | q_1 \dots q_{i-1}) = P(q_i | q_{i-1})$$
- **Output-independence assumption**  

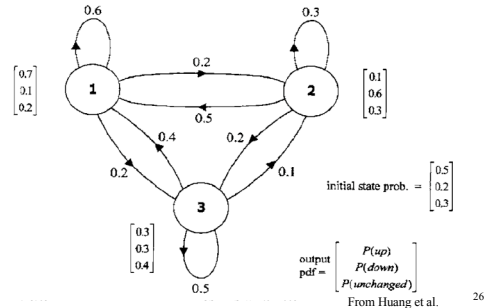
$$P(o_i | O_1^{i-1}, q_i) = P(o_i | q_i)$$

1/25/05

CS 224S Winter 2005

25

## HMM for Dow Jones



## HMMs for weather and ice-cream

- Jason Eisner's cute HMM in Excel, showing Viterbi and EM:  
[http://www.cs.jhu.edu/~jason/papers/-\\_tnlp02](http://www.cs.jhu.edu/~jason/papers/-_tnlp02)
- **Idea:**
  - You are climatologists in 3004
  - Want to know about Baltimore weather in 2004
  - Only data you have is Jason Eisner's diary
  - Which records how much ice cream he ate each day
- **Observation:**
  - Number of ice creams
- **Hidden State: Simplify to only 2 states**  
 Weather is Hot or Cold that day.

1/25/05

27

## The Three Basic Problems for HMMs

- (From the classic formulation by Larry Rabiner after Jack Ferguson)
- L. R. Rabiner. 1989. A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proc IEEE 77(2), 257-286. Also in Waibel and Lee volume.

1/25/05

CS 224S Winter 2005

28

## The Three Basic Problems for HMMs

- **Problem 1 (Evaluation):** Given the observation sequence  $O=(o_1o_2\dots o_T)$ , and an HMM model  $\Phi=(A,B,\pi)$ , how do we efficiently compute  $P(O|\Phi)$ , the probability of the observation sequence, given the model
- **Problem 2 (Decoding):** Given the observation sequence  $O=(o_1o_2\dots o_T)$ , and an HMM model  $\Phi=(A,B,\pi)$ , how do we choose a corresponding state sequence  $Q=(q_1q_2\dots q_T)$  that is optimal in some sense (i.e., best explains the observations)
- **Problem 3 (Learning):** How do we adjust the model parameters  $\Phi=(A,B,\pi)$  to maximize  $P(O|\Phi)$ ?

1/25/05

CS 224S Winter 2005

From Rabiner

29

## The Evaluation Problem

- Given observation sequence  $O$  and HMM  $\Phi$ , compute  $P(O|\Phi)$
- Why is this hard? Sum over all possible sequences of states!

$$P(O|\Phi) = \sum_{all\ S} P(S|\Phi)P(O|S,\Phi)$$

$$= \sum_{all\ S} a_{o_1} b_{o_1}(s_1) a_{s_1 s_2} b_{o_2}(s_2) \dots a_{s_{T-1} s_T} b_{o_T}(s_T)$$

$$P(o_1o_2o_3|q_0q_0q_0) + P(o_1o_2o_3|q_0q_0q_1) + P(o_1o_2o_3|q_0q_1q_2) + P(o_1o_2o_3|q_0q_1q_0)$$

1/25/05

CS 224S Winter 2005

30

## Computing observation likelihood $P(O|\Phi)$

- Why can't we do an explicit sum over all paths?
- Because it's intractable.  $O(N^T)$
- What we do instead:
- **The Forward Algorithm.**  $O(N^2T)$

1/25/05

CS 224S Winter 2005

31

## The Forward Algorithm

- The Idea: Fold these exponential paths into a simple trellis, so that all possible paths will remerge into N states at every time slice.
- We define the *forward probability* as follows:  $\alpha_t(i) = P(o_0o_1 \dots o_t, q_t = i | \Phi)$
- this is the probability that the HMM  $\Phi$  is in state  $i$  at time  $t$  having generated partial observation  $O_t$ .
- We compute it by induction:
  - Initialization:  $\alpha_1(i) = \pi_i P(o_1 | q_i), 1 \leq i \leq N$
  - (equivalently):  $\alpha_1(i) = \pi_i b_i(o_1), 1 \leq i \leq N$
  - Induction:

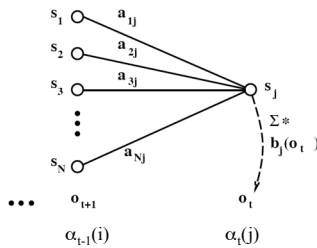
$$\alpha_t(j) = \sum_{i=1}^N \alpha_{t-1}(i) a_{ij} b_j(o_t), \quad 2 \leq t \leq T, 1 \leq j \leq N$$

(3)

- Termination:  $P(O|\Phi) = \sum_{i=1}^N \alpha_T(i)$

## The inductive step, from Rabiner and Juang

- Computation of  $\alpha_t(j)$  by summing all previous values  $\alpha_{t-1}(i)$  for all  $i$



1/25/05

CS 224S Winter 2005

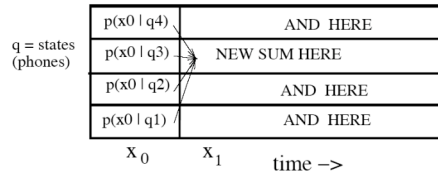
33

## The Forward trellis computation, another view

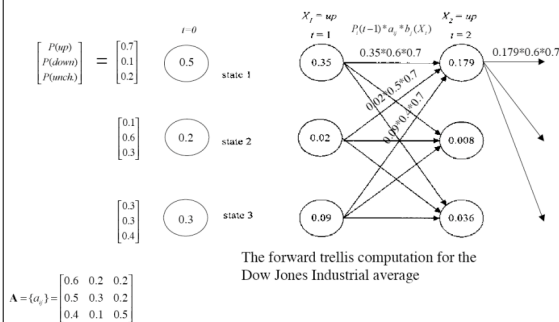
For frame  $x_1$ , compute:

$$\text{Sum} \left( \begin{array}{l} P(x_1 | q_3) P(x_0 | q_1) P(q_3 | q_1) \\ P(x_1 | q_3) P(x_0 | q_2) P(q_3 | q_2) \\ P(x_1 | q_3) P(x_0 | q_3) P(q_3 | q_3) \\ P(x_1 | q_3) P(x_0 | q_4) P(q_3 | q_4) \end{array} \right)$$

local distance



## Forward trellis for Dow Jones



## The Decoding Problem

- Given observations  $O=(o_1o_2\dots o_T)$ , and HMM  $\Phi=(A, B, \pi)$ , how do we choose best state sequence  $Q=(q_1, q_2, \dots, q_T)$ ?
- The forward algorithm computes  $P(O|W)$
- Could find best  $W$  by running forward algorithm for each  $W$  in  $L$ , picking  $W$  maximizing  $P(O|W)$
- But we can't do this, since number of sentences is  $O(W^T)$ . Instead:
  - Viterbi Decoding: dynamic programming, slight modification of the forward algorithm
  - A\* Decoding: search the space of all possible sentences using the forward algorithm as a subroutine.

1/25/05

CS 224S Winter 2005

36

## The Viterbi Algorithm

- The Idea: Just like Forward, fold exponential paths into a simple trellis, so that all possible paths will remerge into N states at every time slice.

- We define the *viterbi probability* as follows:

$$v_t(i) = P(o_0 o_1 \dots o_t, Q_1^{t-1}, q_t = i | \Phi)$$

- $v_t(i)$  is the probability that the HMM  $\Phi$  is in state  $i$  at time  $t$  having generated partial observation  $O_t^i$  by passing through the most likely state sequence  $Q_1^{t-1}$ .

- We again compute it by induction:

- Initialization:

$$v_1(i) = \pi_i b_i(o_1), 1 \leq i \leq N \quad (4)$$

$$b_1(i) = 0 \quad (5)$$

- Induction:

$$v_t(j) = \max_{1 \leq i \leq N} v_{t-1}(i) a_{ij} b_j(o_t)$$

## The Viterbi Algorithm

$$2 \leq t \leq T, 1 \leq j \leq N \quad (6)$$

$$b_t(j) = [\operatorname{argmax}_{1 \leq i \leq N} v_{t-1}(i) a_{ij}]$$

$$2 \leq t \leq T, 1 \leq j \leq N \quad (7)$$

- Termination: The best score is  $\max_{1 \leq i \leq N} v_T(i)$

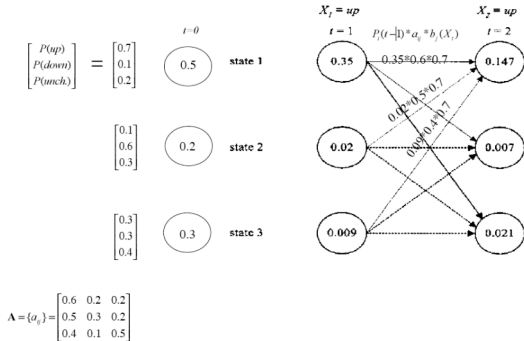
$$q_T^* = \operatorname{argmax}_{1 \leq i \leq N} b_T(i) \quad (8)$$

- Backtracking

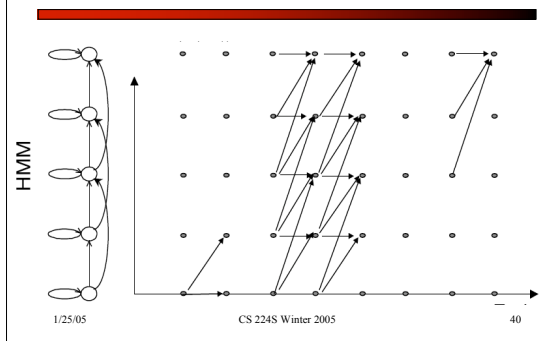
$$q_t^* = b_{t+1}(q_{t+1}^*); t = T-1, T-2, \dots, 1 \quad (9)$$

$$Q^* = (q_1^*, q_2^*, \dots, q_T^*) \text{ is the best state sequence} \quad (10)$$

## Viterbi for Dow Jones



## The Viterbi Trellis



## Why "Dynamic Programming"

"I spent the Fall quarter (of 1950) at RAND. My first task was to find a name for multistage decision processes. An interesting question is, Where did the name, dynamic programming, come from? The 1950s were not good years for mathematical research. We had a very interesting gentleman in Washington named Wilson. He was Secretary of Defense, and he actually had a pathological fear and hatred of the word, research. I'm not using the term lightly; I'm using it precisely. His face would suffuse, he would turn red, and he would get violent if people used the term, research, in his presence. You can imagine how he felt, then, about the term, mathematical. The RAND Corporation was employed by the Air Force, and the Air Force had Wilson as its boss, essentially. Hence, I felt I had to do something to shield Wilson and the Air Force from the fact that I was really doing mathematics inside the RAND Corporation. What title, what name, could I choose? In the first place I was interested in planning, in decision making, in thinking. But planning, is not a good word for various reasons. I decided therefore to use the word, "programming" I wanted to get across the idea that this was dynamic, this was multistage, this was time-varying I thought, lets kill two birds with one stone. Lets take a word that has an absolutely precise meaning, namely dynamic, in the classical physical sense. It also has a very interesting property as an adjective, and that is its impossible to use the word, dynamic, in a pejorative sense. Try thinking of some combination that will possibly give it a pejorative meaning. Its impossible. Thus, I thought dynamic programming was a good name. It was something not even a Congressman could object to. So I used it as an umbrella for my activities." Richard Bellman, "Eye of the Hurricane: an autobiography" 1984.

1/25/05

CS 224S Winter 2005

Thanks to Chen, Picheny, Elde, Nock

## HMMs for Speech

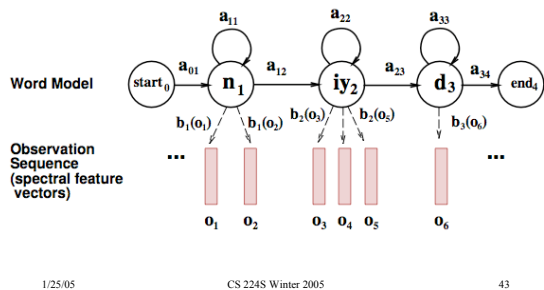
- We haven't yet shown how to learn the A and B matrices for HMMs; we'll do that on Thursday
- But let's return to think about speech

1/25/05

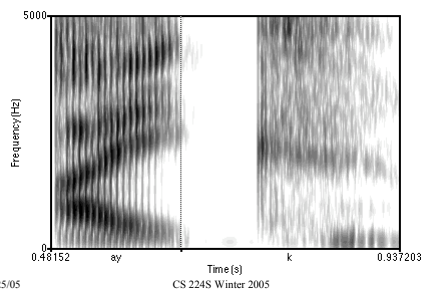
CS 224S Winter 2005

42

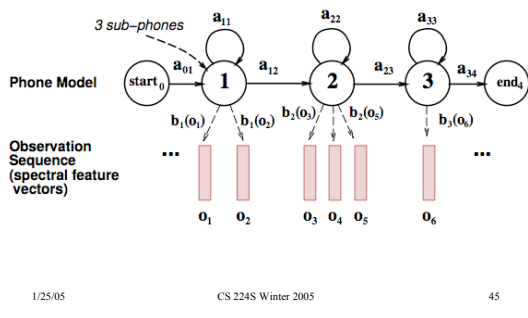
## HMMs for speech



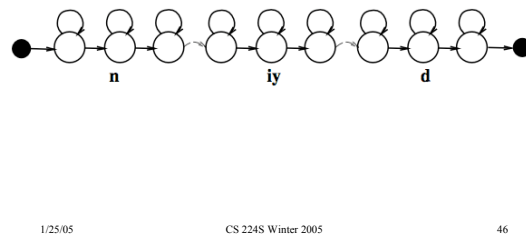
## But phones aren't homogeneous



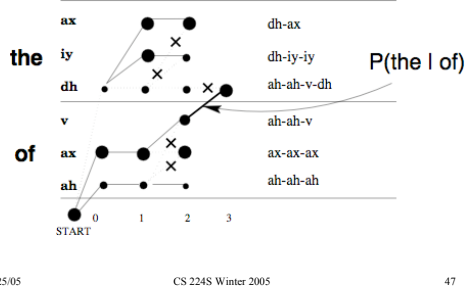
## So we'll need to break phones into subphones



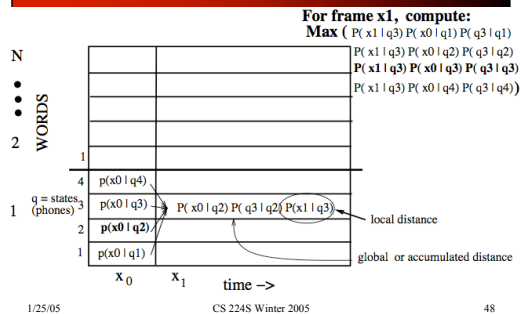
## Now a word looks like this:



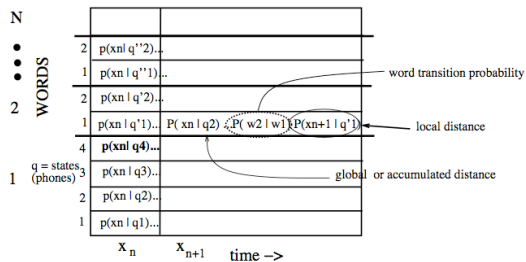
## Back to Viterbi with speech, but w/out subphones for a sec



## Viterbi: Word Internal



## Viterbi: Between words

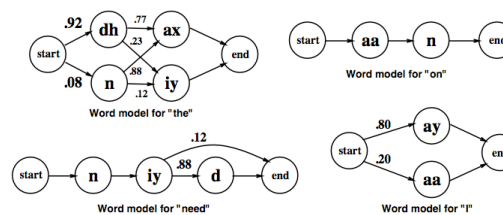


1/25/05

CS 224S Winter 2005

49

## ASR Lexicon: Markov Models for pronunciation



1/25/05

CS 224S Winter 2005

50

## Summary

- **Speech Recognition Architectural Overview**
- **Hidden Markov Models in general**
  - Forward
  - Viterbi Decoding
- **Hidden Markov models for Speech**
- **Next time: Learning and EM**

1/25/05

CS 224S Winter 2005

51