

CS 4705

Hidden Markov Models

10/2/19

(1)

Slides adapted from Dan Jurafsky, and James Martin

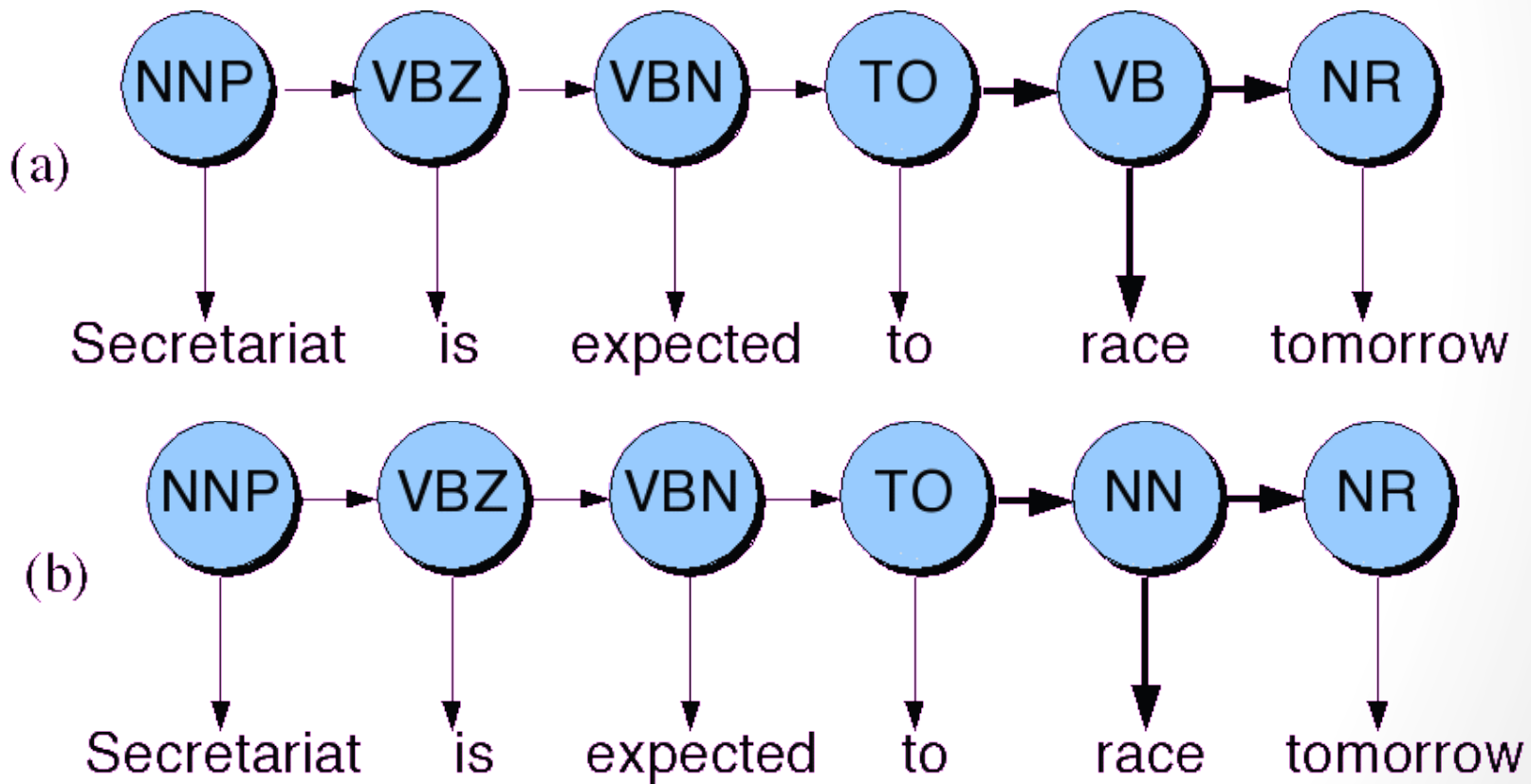
Responses on “well”

- There is an oil well a few miles away from my house. **Noun.**
- Well! I never thought I would see that! **Interjection**
- It is a well designed program. **Adv**
- Tears welled in her eyes. **Verb**
- The store sells fruit as well as vegetables. **Conjunction.**
- Are you well? **Adj.**
- He and his family were well off. **Adjectival Phrase (?)**

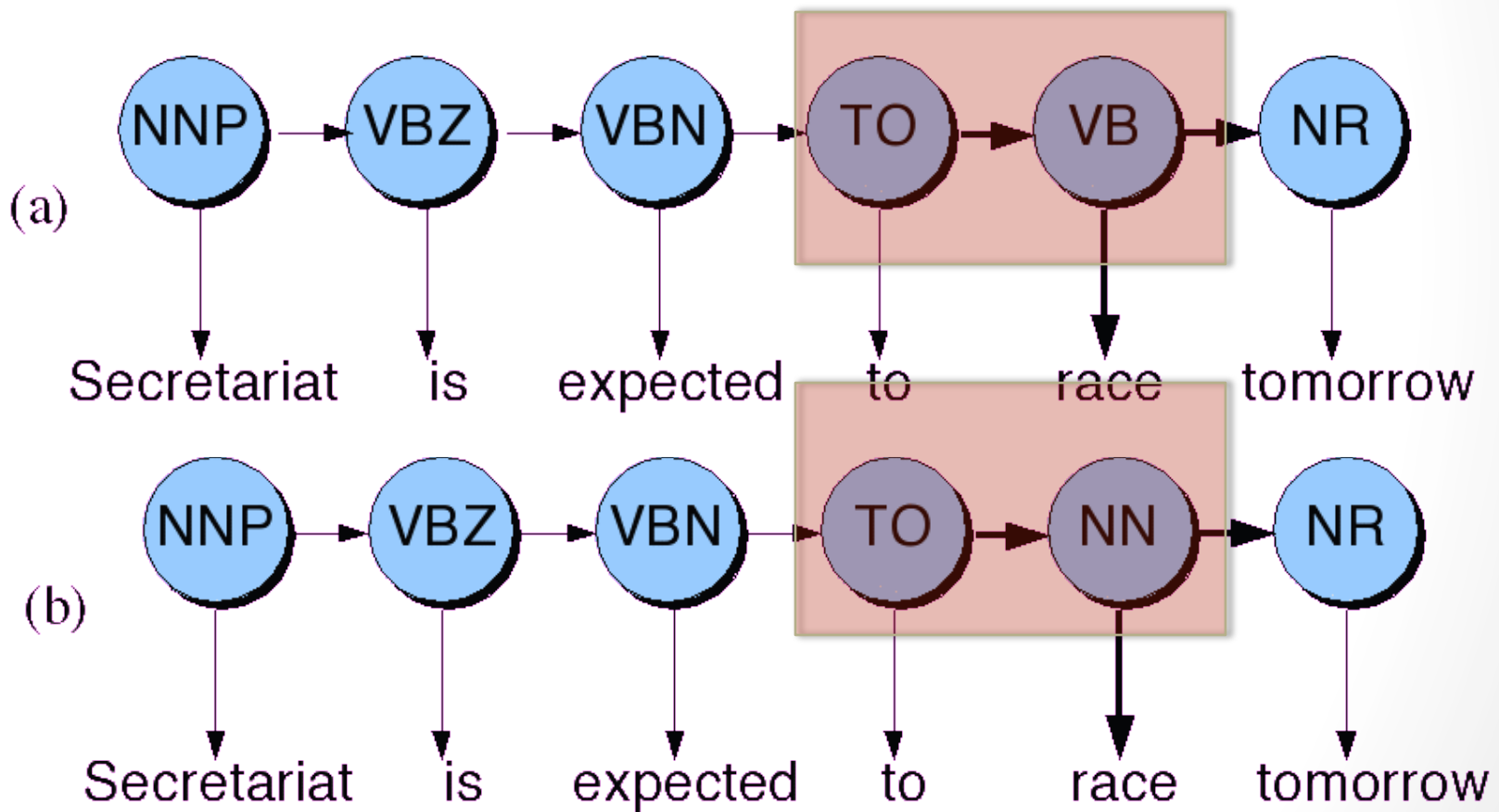
Announcements

- Reading for today: Chapter 7-7.5 (NLP), C 8.4 Speech and Language
- The Tas will be offering tutorials on the math of neural nets and in particular, back propagation

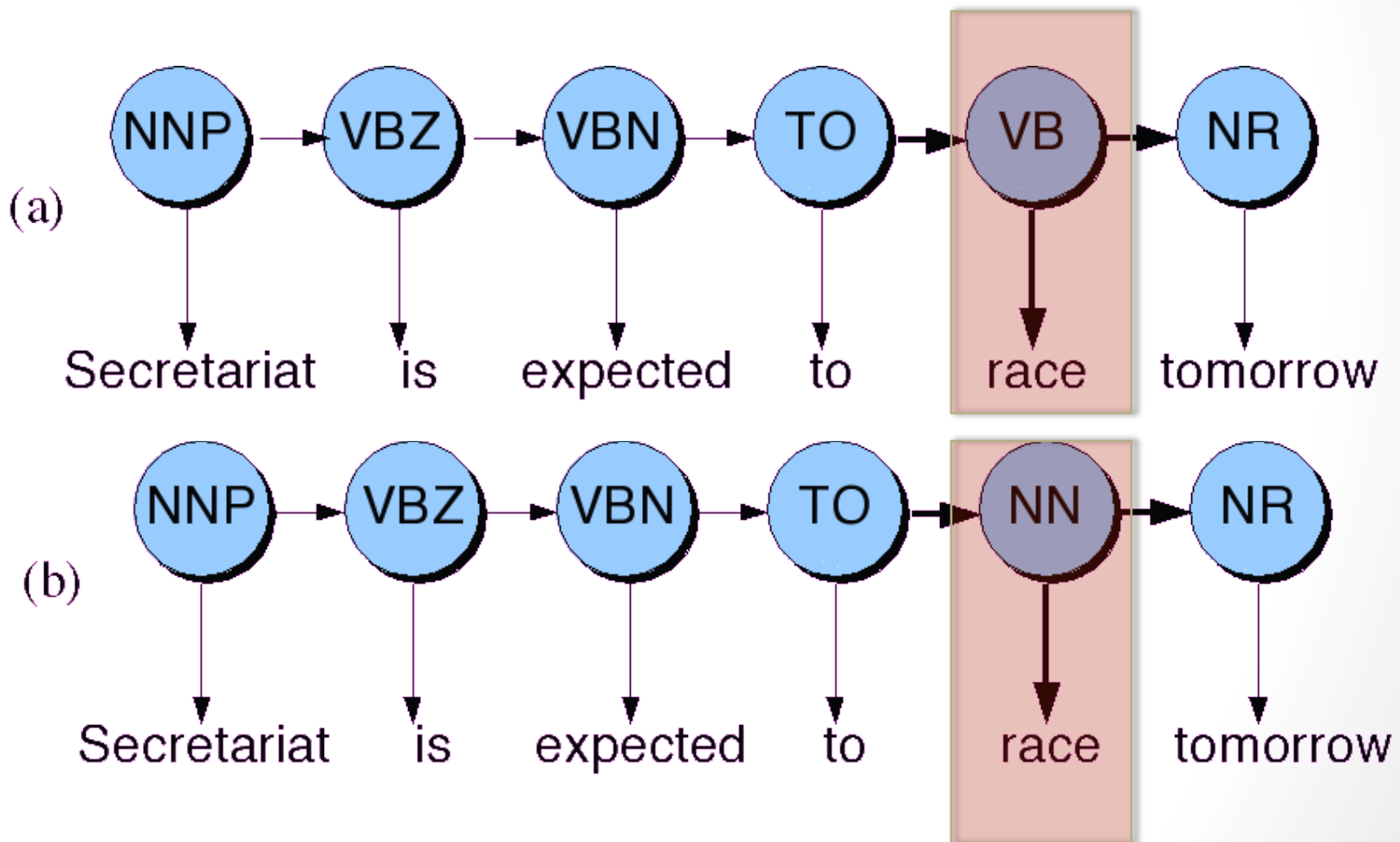
Disambiguating “race”



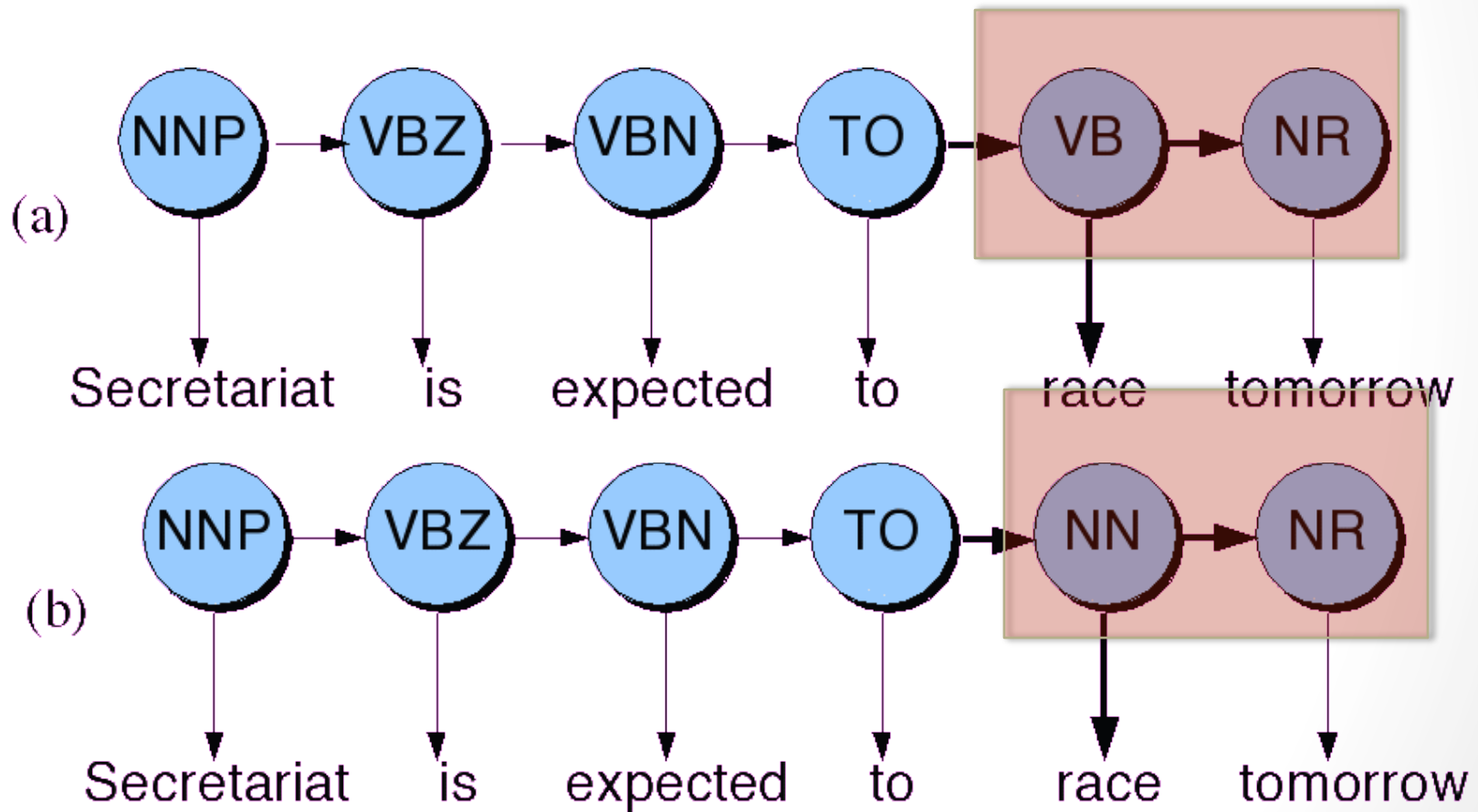
Disambiguating “race”



Disambiguating “race”



Disambiguating “race”



- $P(\text{NN} | \text{TO}) = .00047$

- $P(\text{VB} | \text{TO}) = .83$

- $P(\text{race} | \text{NN}) = .00057$

- $P(\text{race} | \text{VB}) = .00012$

- $P(\text{NR} | \text{VB}) = .0027$

- $P(\text{NR} | \text{NN}) = .0012$

- $P(\text{VB} | \text{TO})P(\text{NR} | \text{VB})P(\text{race} | \text{VB}) = .00000027$

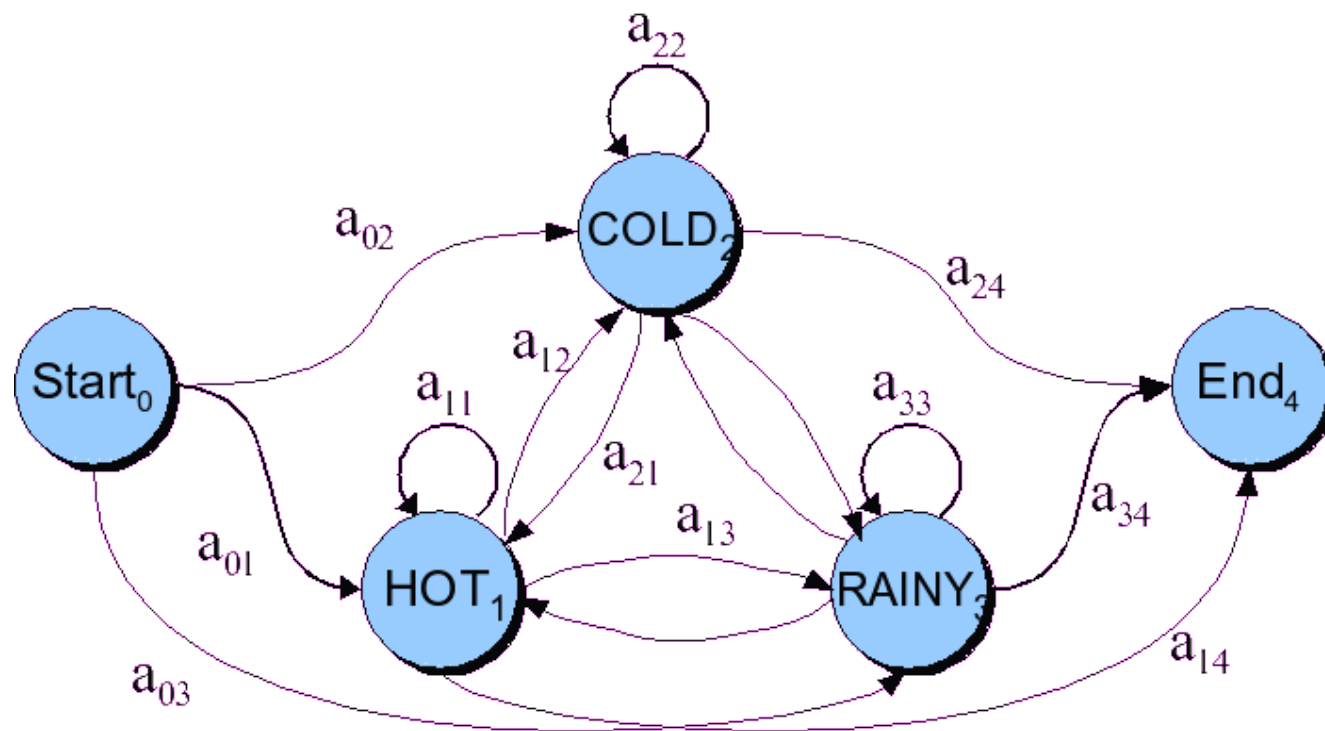
- $P(\text{NN} | \text{TO})P(\text{NR} | \text{NN})P(\text{race} | \text{NN}) = .00000000032$

- So we (correctly) choose the verb reading,

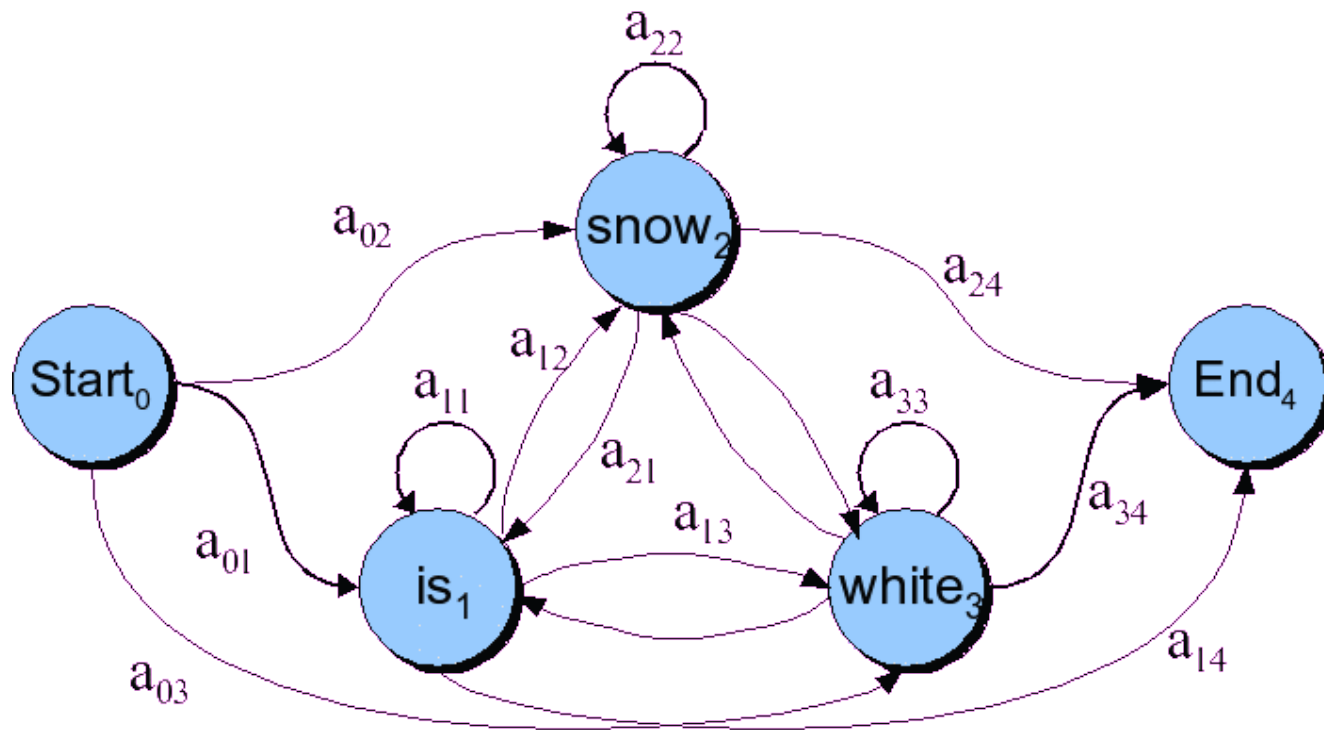
Definitions

- A **weighted finite-state automaton** adds probabilities to the arcs
 - The sum of the probabilities leaving any arc must sum to one
- A **Markov chain** is a special case of a WFST
 - the input sequence uniquely determines which states the automaton will go through
- Markov chains can't represent inherently ambiguous problems
 - Assigns probabilities to unambiguous sequences

Markov chain for weather



Markov chain for words



Markov chain = “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
- Transition probabilities:
 - a set of probabilities $A = a_{01} a_{02} \dots a_{n1} \dots a_{nn}$.
 - Each a_{ij} represents the probability of transitioning from state i to state j
 - The set of these is the transition probability matrix A

$$a_{ij} = P(q_t = j \mid q_{t-1} = i) \quad 1 \leq i, j \leq N$$
$$\sum_{j=1}^N a_{ij} = 1; \quad 1 \leq i \leq N$$

- Distinguished start and end states

Markov chain = “First-order observable Markov Model”

- Current state only depends on previous state

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

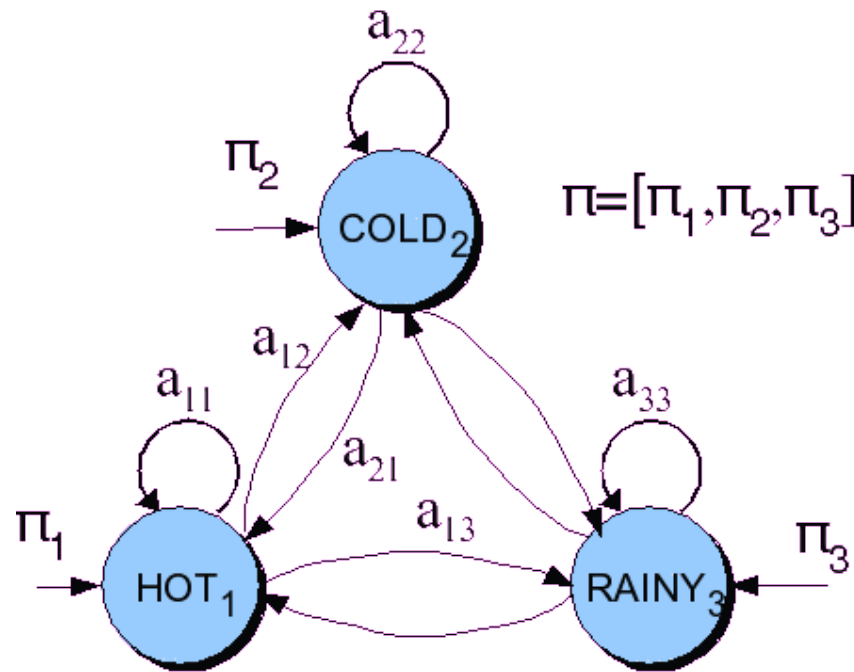
Another representation for start state

- Instead of start state
- Special initial probability vector π
$$\pi_i = P(q_1 = i) \quad 1 \leq i \leq N$$
 - An initial distribution over probability of start states

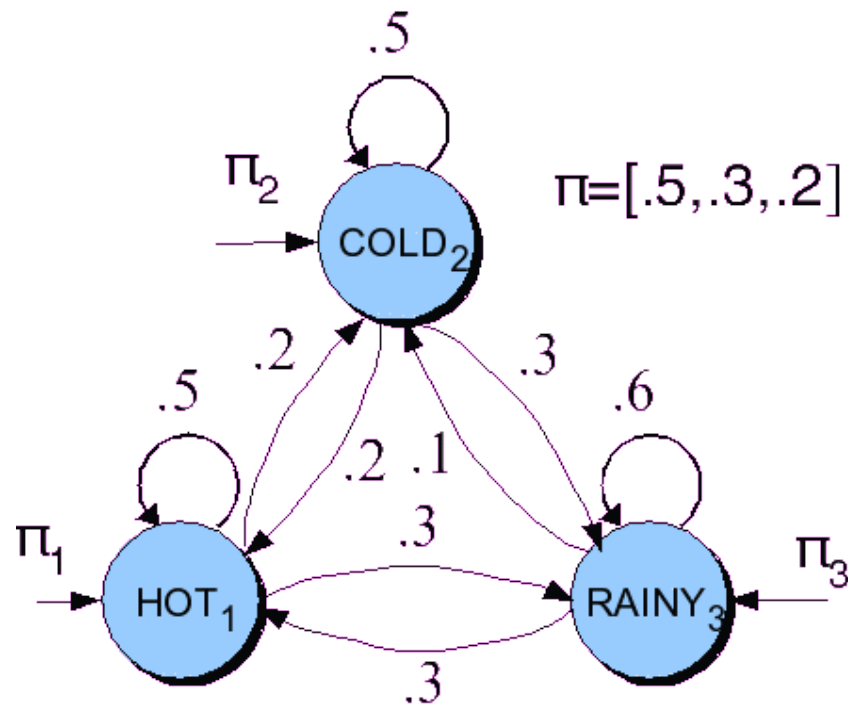
- Constraints:

$$\sum_{j=1}^N \pi_j = 1$$

The weather figure using pi



The weather figure: specific example



Markov chain for weather

- What is the probability of 4 consecutive rainy days?



What is the probability of 4 consecutive rainy days?

Hidden Markov Models

- We don't observe POS tags
 - We infer them from the words we see
- Observed events
- Hidden events

HMM for Ice Cream

- You are a climatologist in the year 2799
- Studying global warming
- You can't find any records of the weather in New York, NY for summer of 2007
- But you find Kathy McKeown's diary
- Which lists how many ice-creams Kathy ate every date that summer
- Our job: figure out how hot it was

Hidden Markov Model

- For Markov chains, the output symbols are the same as the states.
 - See **hot** weather: we're in state **hot**
- But in part-of-speech tagging (and other things)
 - The output symbols are **words**
 - The hidden states are **part-of-speech tags**
- So we need an extension!
- A **Hidden Markov Model** is an extension of a Markov chain in which the input symbols are not the same as the states.
- This means **we don't know which state we are in.**

Hidden Markov Models

- States $Q = q_1, q_2 \dots q_N$;
- Observations $O = o_1, o_2 \dots o_N$;
 - Each observation is a symbol from a vocabulary $V = \{v_1, v_2, \dots, v_V\}$
- Transition probabilities
 - Transition probability matrix $A = \{a_{ij}\}$

$$a_{ij} = P(q_t = j \mid q_{t-1} = i) \quad 1 \leq i, j \leq N$$

- Observation likelihoods
 - Output probability matrix $B = \{b_i(k)\}$

$$b_i(k) = P(X_t = o_k \mid q_t = i)$$

- Special initial probability vector π

$$\pi_i = P(q_1 = i) \quad 1 \leq i \leq N$$

Hidden Markov Models

- Some constraints

$$\sum_{j=1}^N a_{ij} = 1; \quad 1 \leq i \leq N$$

$$\pi_i = P(q_1 = i) \quad 1 \leq i \leq N$$

$$\sum_{k=1}^M b_i(k) = 1$$

$$\sum_{j=1}^N \pi_j = 1$$

Assumptions

- **Markov assumption:**

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

- **Output-independence assumption**

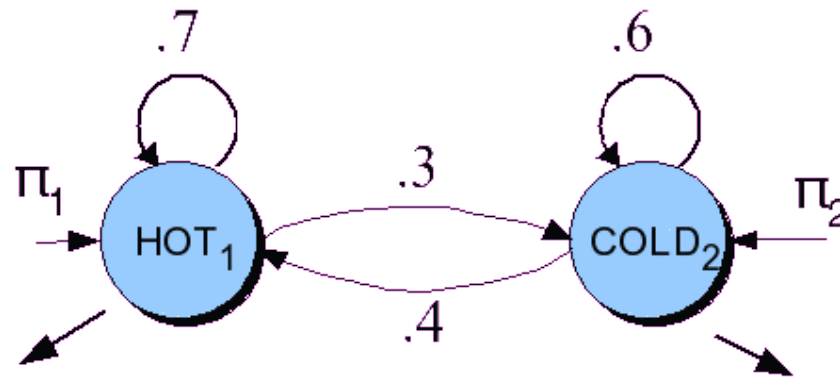
$$P(o_t | O_1^{t-1}, q_1^t) = P(o_t | q_t)$$

McKeown task

- Given
 - Ice Cream Observation Sequence: 2,1,3,2,2,2,3...
- Produce:
 - Weather Sequence: H,C,H,H,H,C...

HMM for ice cream

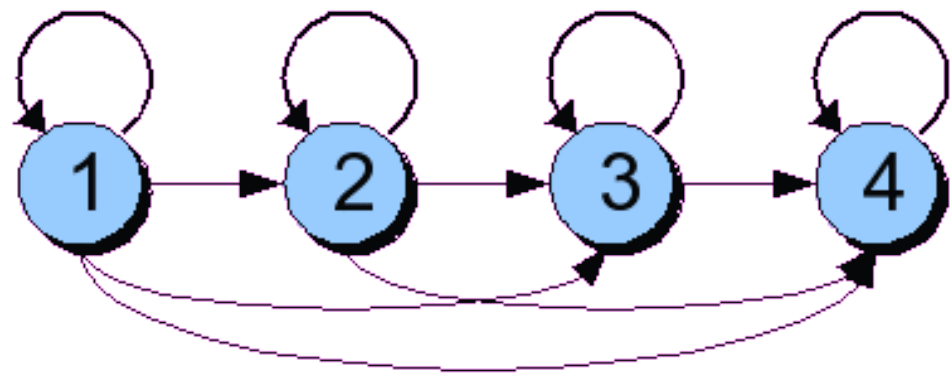
$$\pi = [.8, .2]$$



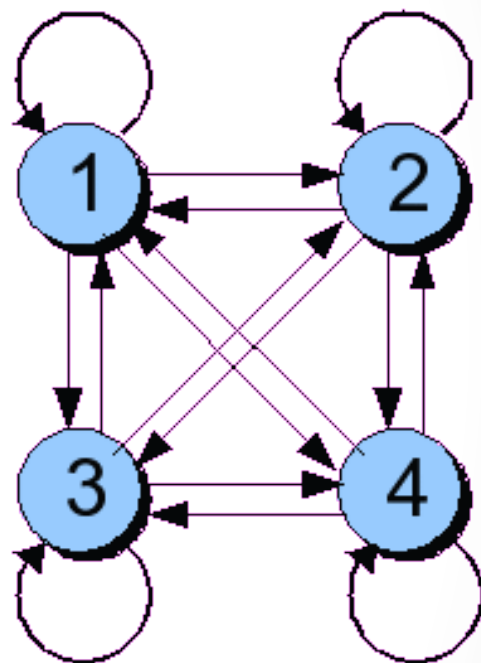
$$\mathbf{B}_1 = \begin{bmatrix} P(1 | HOT) \\ P(2 | HOT) \\ P(3 | HOT) \end{bmatrix} = \begin{bmatrix} .2 \\ .4 \\ .4 \end{bmatrix}$$

$$\mathbf{B}_2 = \begin{bmatrix} P(1 | COLD) \\ P(2 | COLD) \\ P(3 | COLD) \end{bmatrix} = \begin{bmatrix} .5 \\ .4 \\ .1 \end{bmatrix}$$

Different types of HMM structure

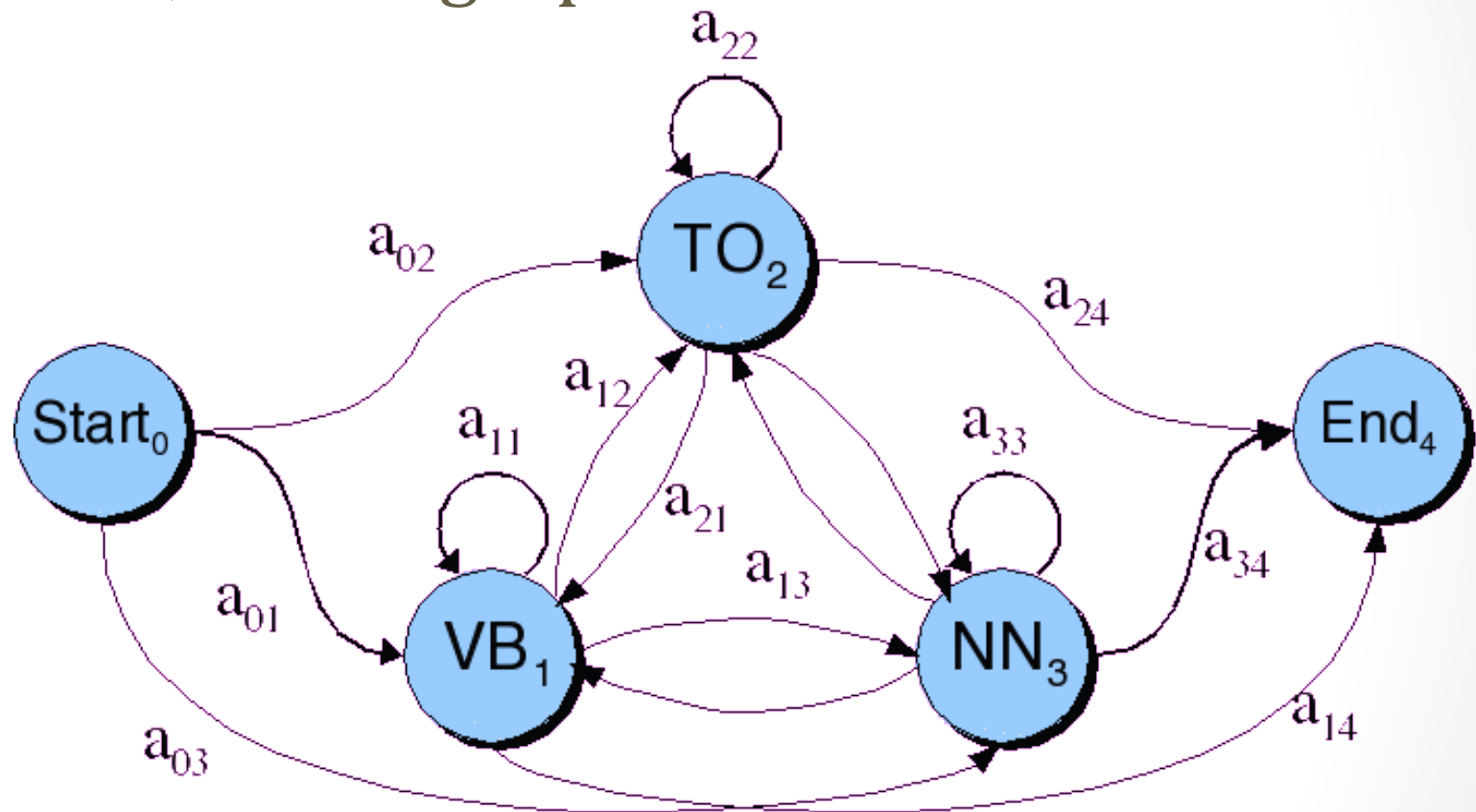


Bakis = left-to-right

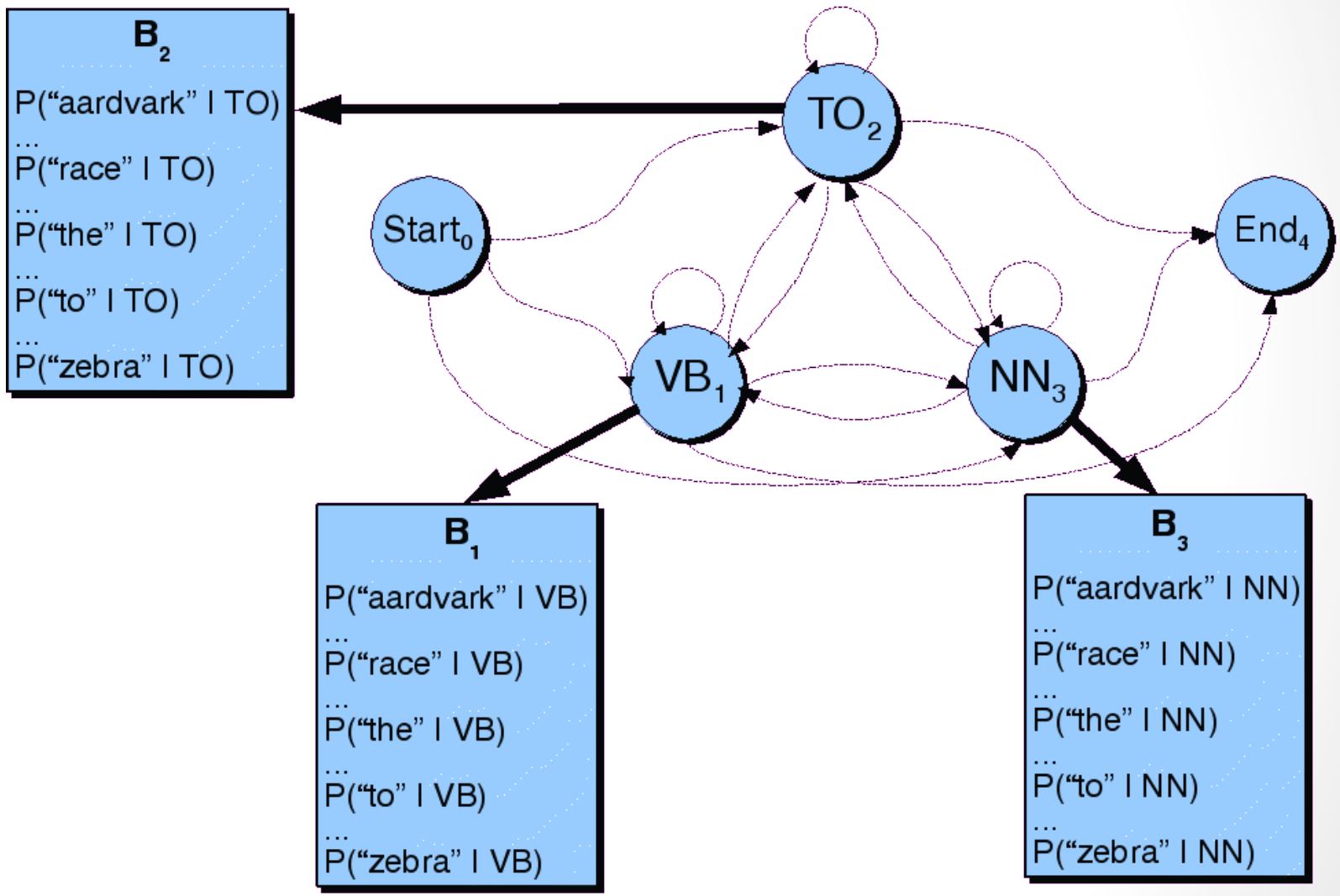


Ergodic =
fully-connected

Transitions between the hidden states of HMM, showing A probs



B observation likelihoods for POS HMM



Three fundamental Problems for HMMs

- **Likelihood:** Given an HMM $\lambda = (A, B)$ and an observation sequence O , determine the likelihood $P(O, \lambda)$.
- **Decoding:** Given an observation sequence O and an HMM $\lambda = (A, B)$, discover the best hidden state sequence Q .
- **Learning:** Given an observation sequence O and the set of states in the HMM, learn the HMM parameters A and B .

What kind of data would we need to learn the HMM parameters?

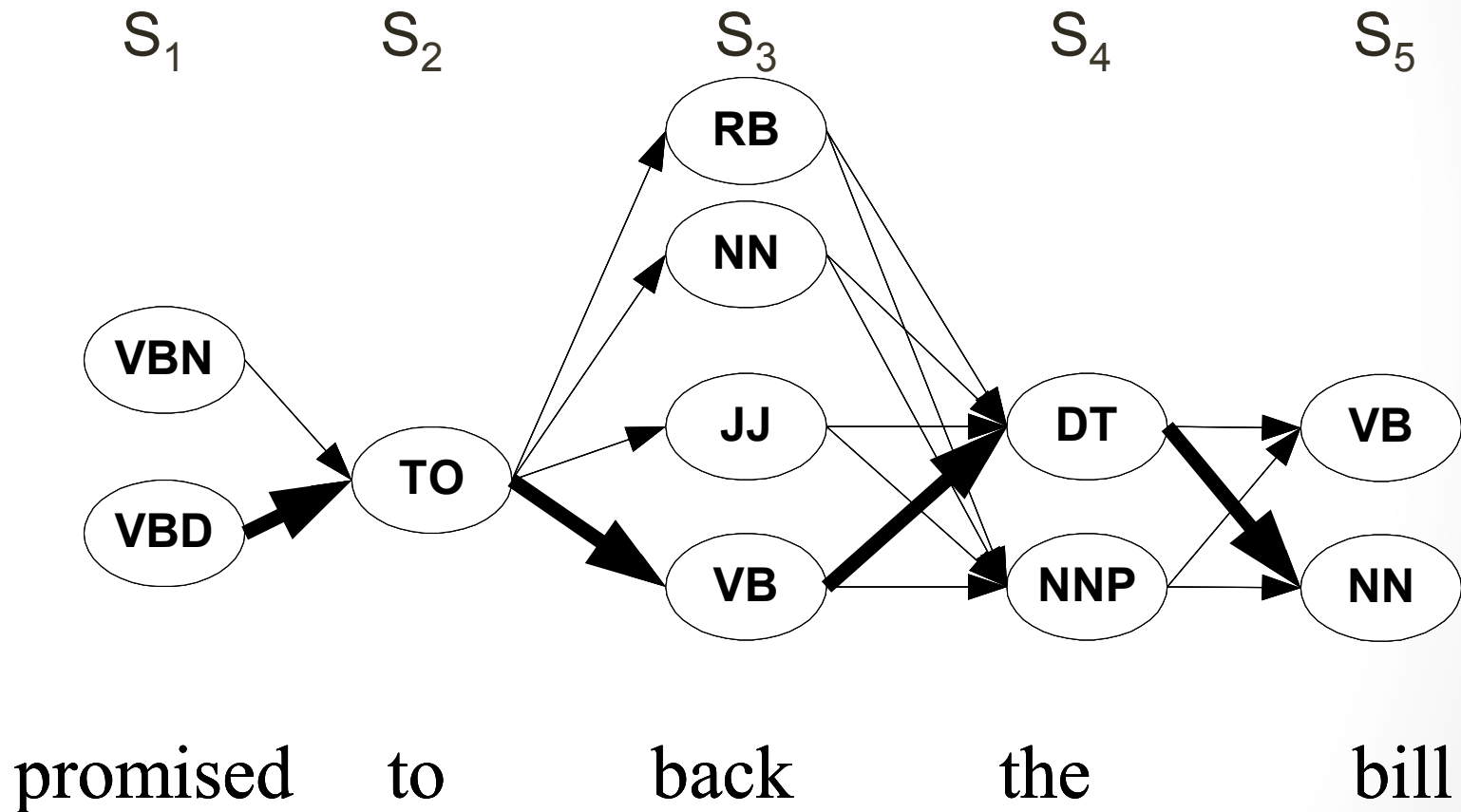


What kind of data would we need to learn the HMM parameters?

Decoding

- The best hidden sequence
 - Weather sequence in the ice cream task
 - POS sequence given an input sentence
- We could use argmax over the probability of each possible hidden state sequence
 - *Why not?*
- Viterbi algorithm
 - Dynamic programming algorithm
 - Uses a dynamic programming trellis
 - Each trellis cell represents, $v_t(j)$, represents the probability that the HMM is in state j after seeing the first t observations and passing through the most likely state sequence

Viterbi intuition: we are looking for the best 'path'



Intuition

- The value in each cell is computed by taking the MAX over all paths that lead to this cell.
- An extension of a path from state i at time $t-1$ is computed by multiplying:

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

$v_{t-1}(i)$ the **previous Viterbi path probability** from the previous time step

a_{ij} the **transition probability** from previous state q_i to current state q_j

$b_j(o_t)$ the **state observation likelihood** of the observation symbol o_t given the current state j

The Viterbi Algorithm

function VITERBI(*observations* of len T , *state-graph*) **returns** *best-path*

$num\text{-}states \leftarrow \text{NUM-OF-STATES}(state\text{-}graph)$

Create a path probability matrix $viterbi[num\text{-}states+2, T+2]$

$viterbi[0,0] \leftarrow 1.0$

for each time step t **from** 1 **to** T **do**

for each state s **from** 1 **to** $num\text{-}states$ **do**

$viterbi[s,t] \leftarrow \max_{1 \leq s' \leq num\text{-}states} viterbi[s',t-1] * a_{s',s} * b_s(o_t)$

$backpointer[s,t] \leftarrow \operatorname{argmax}_{1 \leq s' \leq num\text{-}states} viterbi[s',t-1] * a_{s',s}$

Backtrace from highest probability state in final column of $viterbi[]$ and return path

The A matrix for the POS HMM

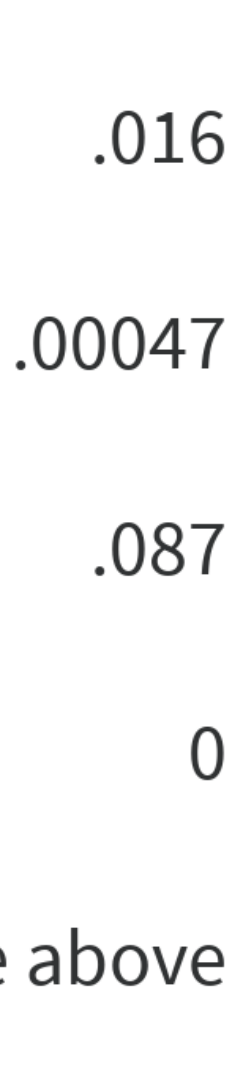
	VB	TO	NN	PPSS
<s>	.019	.0043	.041	.067
VB	.0038	.035	.047	.0070
TO	.83	0	.00047	0
NN	.0040	.016	.087	.0045
PPSS	.23	.00079	.0012	.00014

Figure 4.15 Tag transition probabilities (the a array, $p(t_i|t_{i-1})$) computed from the 87-tag Brown corpus without smoothing. The rows are labeled with the conditioning event; thus $P(PPSS|VB)$ is .0070. The symbol $\langle s \rangle$ is the start-of-sentence symbol.

What is $P(VB|TO)$? What is $P(NN|TO)$? Why does this make sense?

What is $P(TO|VB)$? What is $P(TO|NN)$? Why does this make sense?

What is P(NN|TO)?



What is $P(VB|TO)$?

.035

.0038

.047

.83

None of the above

Why does this make sense?

The B matrix for the POS HMM

	I	want	to	race
VB	0	.0093	0	.00012
TO	0	0	.99	0
NN	0	.000054	0	.00057
PPSS	.37	0	0	0

Figure 4.16 Observation likelihoods (the b array) computed from the 87-tag Brown corpus without smoothing.

Look at $P(\text{want}|\text{VB})$ and $P(\text{want}|\text{NN})$. Give an explanation for the difference in the probabilities.

end

end

end

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

$$v_1(4) = .041 \times 0 = 0$$

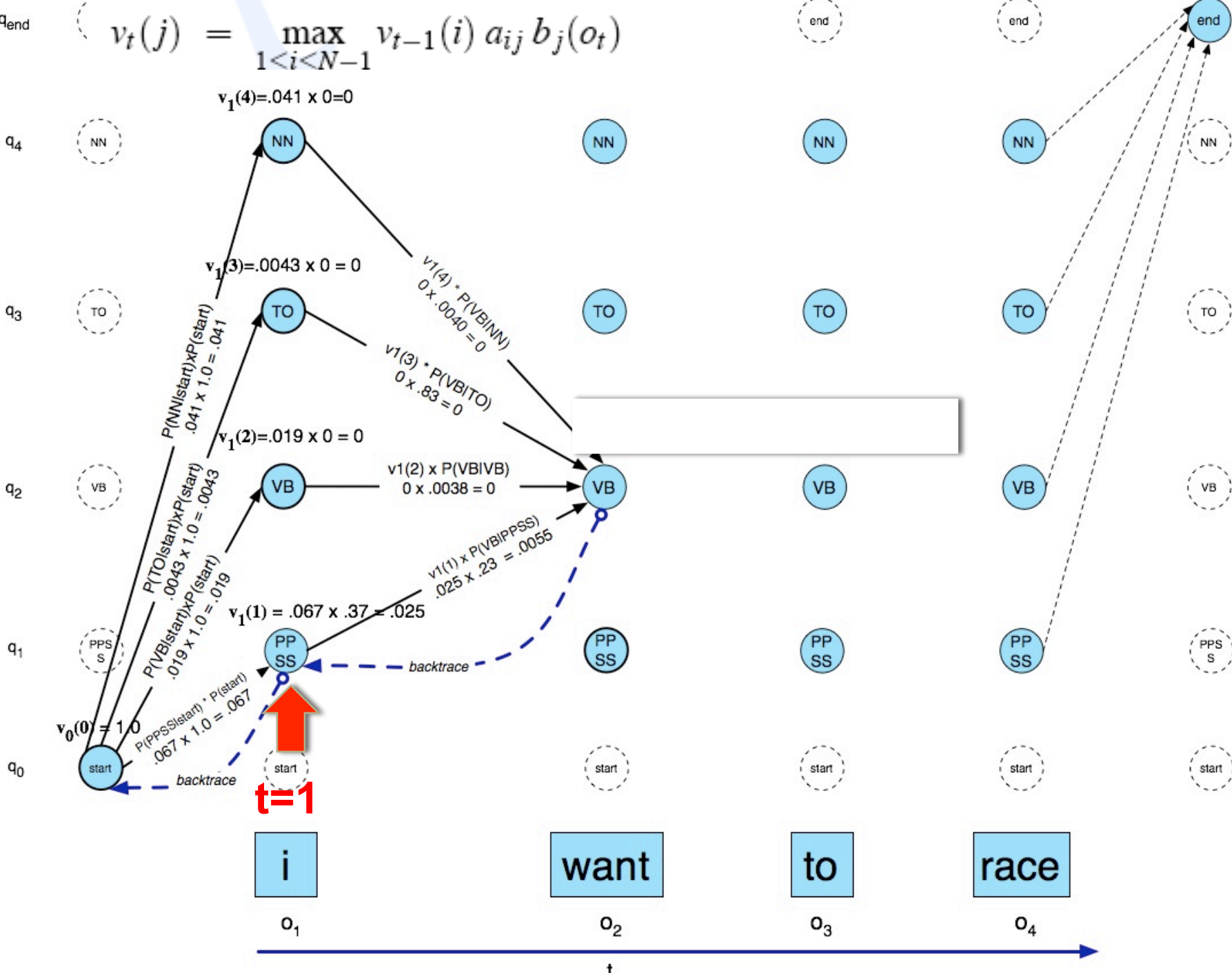
$$v_1(3) = .0043 \times 0 = 0$$

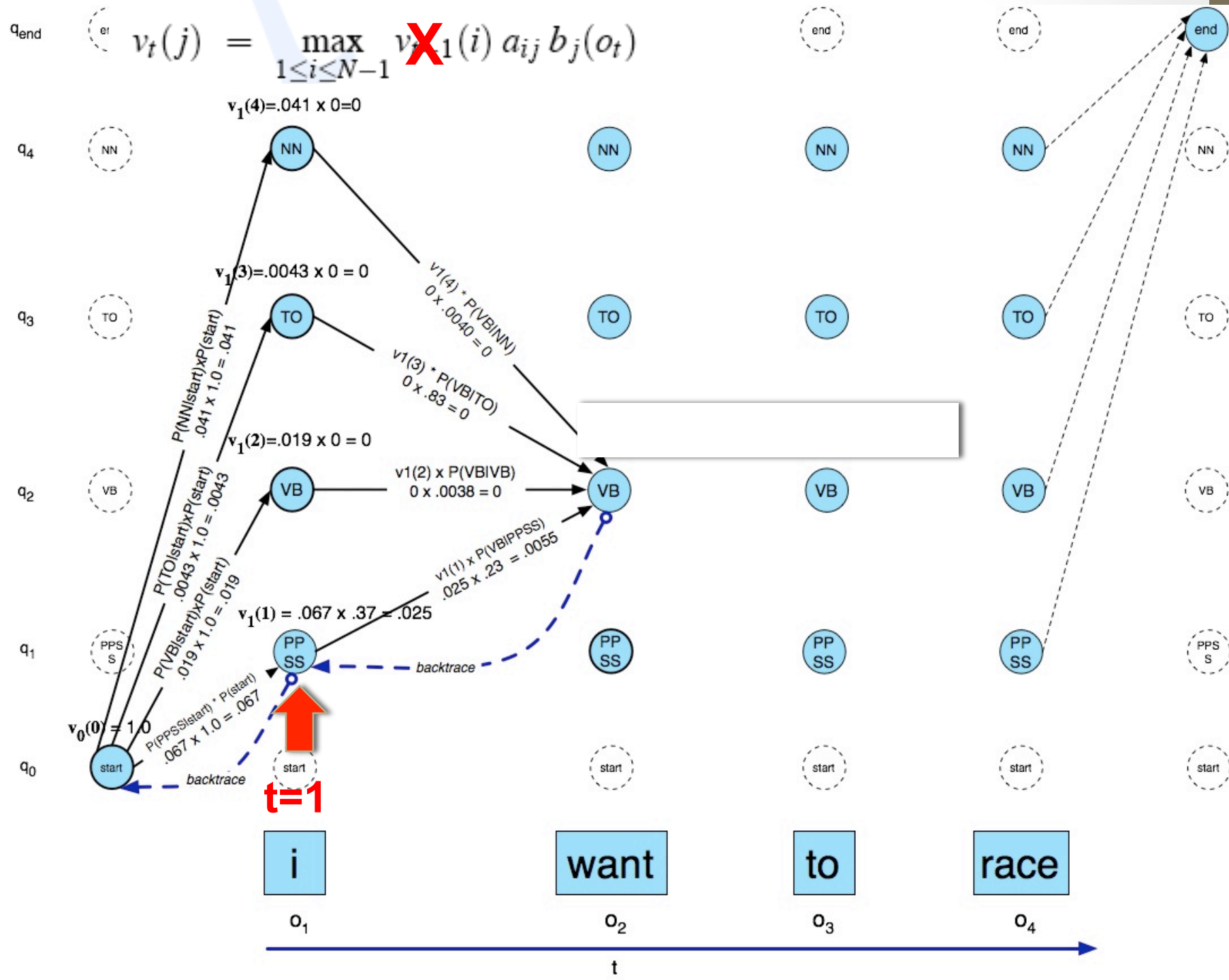
$$v_1(2) = .019 \times 0 = 0$$

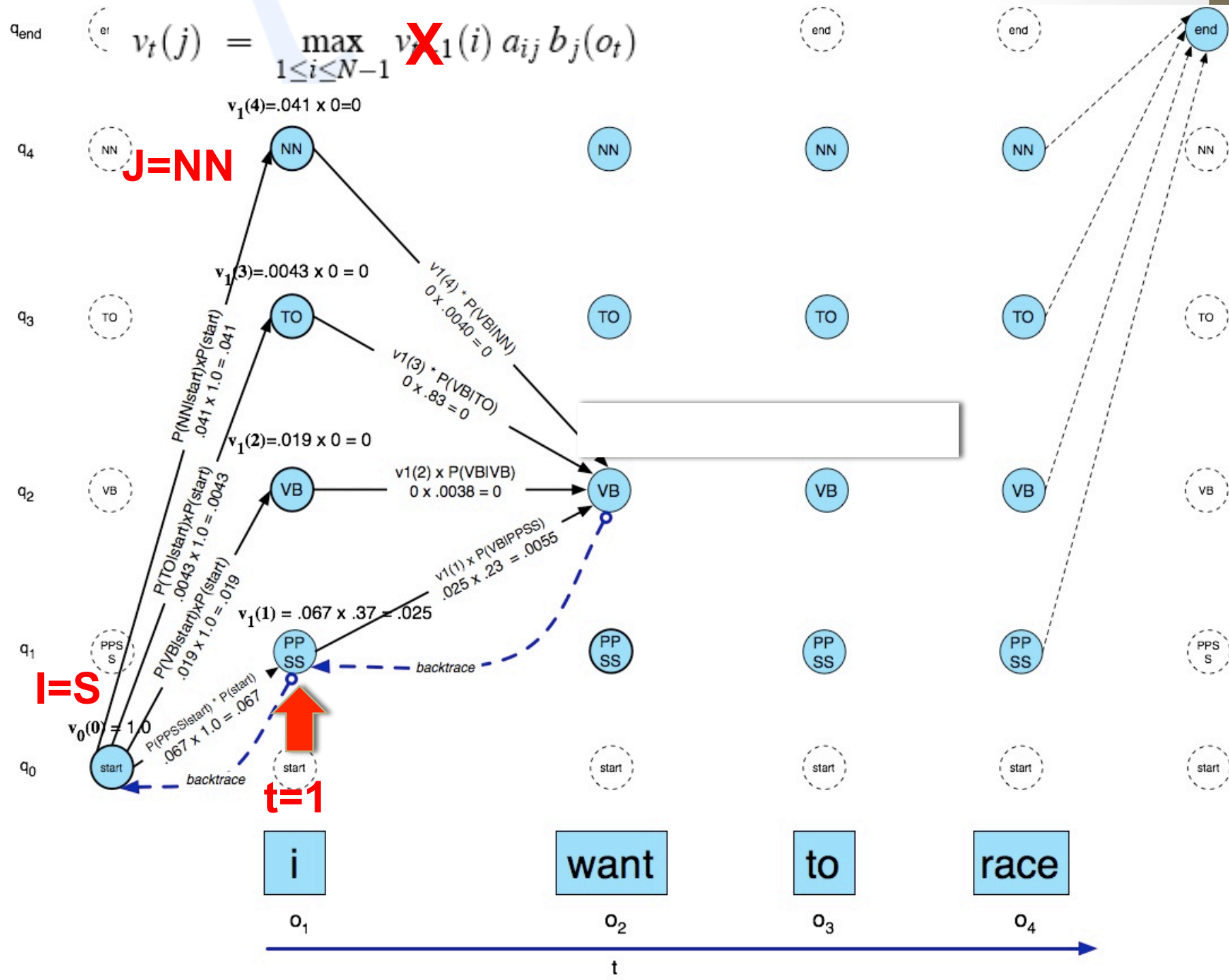
$$v_1(1) = .067 \times .37 = .025$$

$P(NN|start) \times P(start)$
 $.041 \times 1.0 = .041$
 $P(TO|start) \times P(start)$
 $.0043 \times 1.0 = .0043$
 $P(VB|start) \times P(start)$
 $.019 \times 1.0 = .019$
 $P(PPSS|start) \times P(start)$
 $.067 \times 1.0 = .067$

$v_1(4) \times P(VB|NN)$
 $0 \times .0040 = 0$
 $v_1(3) \times P(VB|TO)$
 $0 \times .83 = 0$
 $v_1(2) \times P(VB|VB)$
 $0 \times .0038 = 0$
 $v_1(1) \times P(VB|PPSS)$
 $.025 \times .23 = .0055$







The A matrix for the POS HMM

	VB	TO	NN	PPSS
<s>	.019	.0043	.041	.067
VB	.0038	.035	.047	.0070
TO	.83	0	.00047	0
NN	.0040	.016	.087	.0045
PPSS	.23	.00079	.0012	.00014

Figure 4.15 Tag transition probabilities (the a array, $p(t_i|t_{i-1})$) computed from the 87-tag Brown corpus without smoothing. The rows are labeled with the conditioning event; thus $P(PPSS|VB)$ is .0070. The symbol $\langle s \rangle$ is the start-of-sentence symbol.

end

end

end

NN

NN

NN

TO

TO

TO

VB

VB

VB

PP
SS

PP
SS

PP
SS

start

start

start

q_{end}

q_4

q_3

q_2

q_1

q_0

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

$v_1(4) = .041 \times 0 = 0$ **.041X**

J=NN

$$v_1(3) = .0043 \times 0 = 0$$

$$v_1(4) * P(VB|NN) = 0 \times .0040 = 0$$

$$P(NN|start) \times P(start) = .041 \times 1.0 = .041$$

$$v_1(3) * P(VB|TO) = 0 \times .83 = 0$$

$$v_1(2) = .019 \times 0 = 0$$

$$v_1(2) \times P(VB|VB) = 0 \times .0038 = 0$$

$$P(NN|start) \times P(start) = .041 \times 1.0 = .041$$

$$P(TO|start) \times P(start) = .0043 \times 1.0 = .0043$$

$$P(VB|start) \times P(start) = .019 \times 1.0 = .019$$

$$v_1(1) = .067 \times .37 = .025$$

$$v_1(1) \times P(VB|PPSS) = .025 \times .23 = .0055$$

$$v_0(0) = 1.0$$

$$P(PPSS|start) \times P(start) = .067 \times 1.0 = .067$$

I=S

t=1

i

want

to

race

o_1

o_2

o_3

o_4

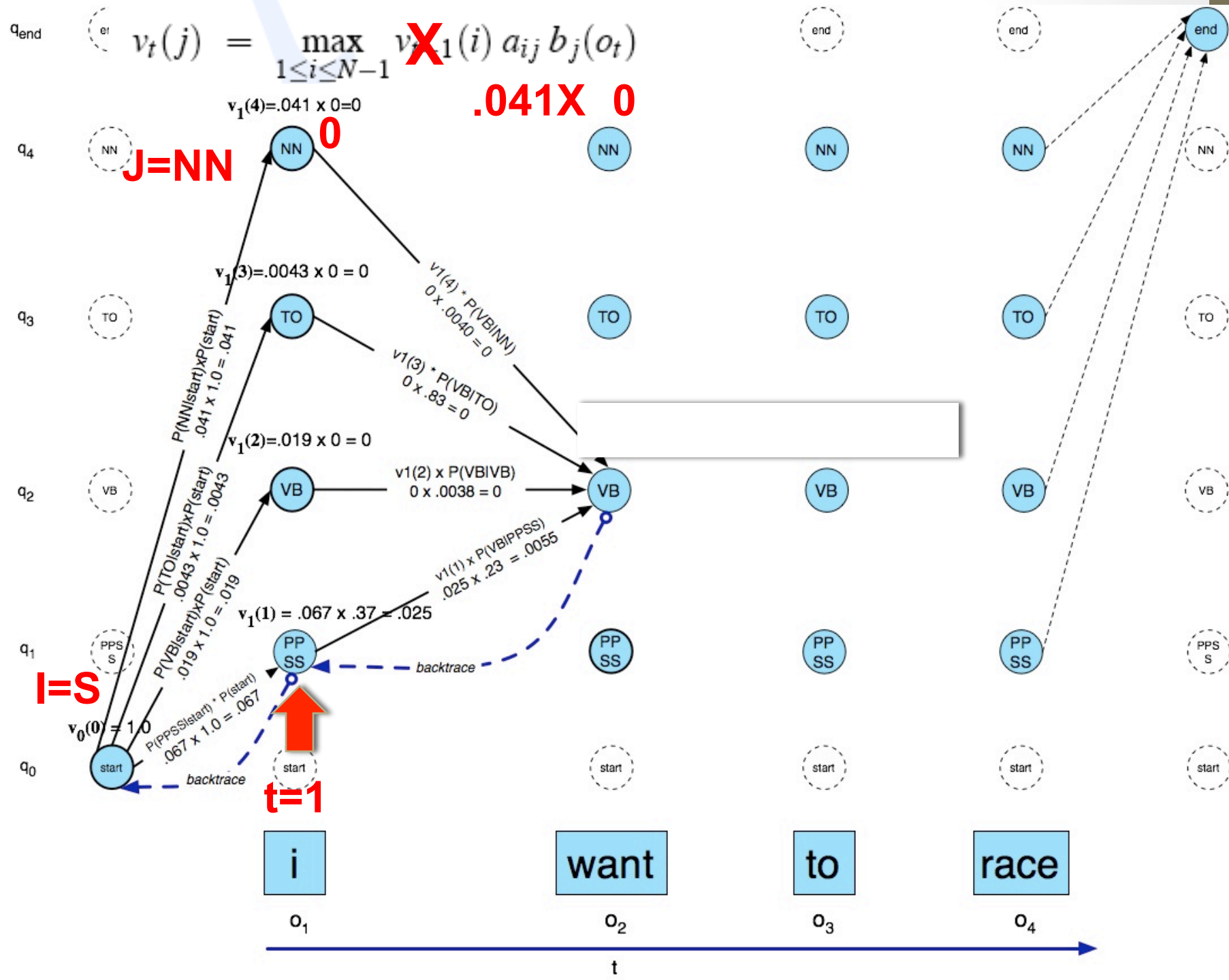
t

The B matrix for the POS HMM

	I	want	to	race
VB	0	.0093	0	.00012
TO	0	0	.99	0
NN	0	.000054	0	.00057
PPSS	.37	0	0	0

Figure 4.16 Observation likelihoods (the b array) computed from the 87-tag Brown corpus without smoothing.

Look at $P(\text{want}|\text{VB})$ and $P(\text{want}|\text{NN})$. Give an explanation for the difference in the probabilities.



q_{end}

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

end

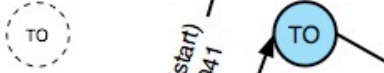
end

end

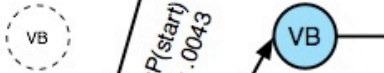
q_4



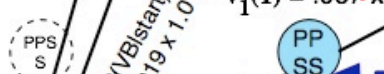
q_3



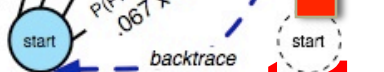
q_2



q_1



q_0



$I=S$

$J=NN$

$v_1(4) = .041 \times 0 = 0$

$.041 \times 0$

$v_1(3) = .004 \times 0 = 0$

$0 \times .0040 = 0$

$v_1(2) = .019 \times 0 = 0$

$v_1(2) \times P(VB|VB) = 0 \times .0038 = 0$

$v_1(1) = .067 \times .37 = .025$

$v_1(1) \times P(VB|PPSS) = .025 \times .23 = .0055$

$v_0(0) = 1.0$

$P(PPSS|start) \times P(start) = .067 \times 1.0 = .067$

$P(VB|start) \times P(start) = .019 \times 1.0 = .019$

$P(TO|start) \times P(start) = .0043 \times 1.0 = .0043$

$P(NN|start) \times P(start) = .041 \times 1.0 = .041$

backtrace

backtrace

$t=1$

i

o_1

want

o_2

to

o_3

race

o_4

t

10/2/19

end

end

end

NN

NN

NN

TO

TO

TO

VB

VB

VB

PP SS

PP SS

PP SS

start

start

start

q_{end}
q₄
q₃
q₂
q₁
q₀

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

$$v_1(4) = .041 \times 0 = 0$$

$$v_1(3) = .0041 \times 0 = 0$$

$$v_1(2) = .019 \times 0 = 0$$

$$v_1(1) = .067 \times .37 = .025$$

$$v_0(0) = 1.0$$

J=NN

I=S

t=1

Show the 4 formulas you would use to compute the value at this node and the max.



i want to race

o₁

o₂

o₃

o₄

t

Computing the Likelihood of an observation

- Forward algorithm
- Exactly like the viterbi algorithm, except
 - To compute the probability of a state, sum the probabilities from each path

Error Analysis: ESSENTIAL!!!

- Look at a confusion matrix

	IN	JJ	NN	NNP	RB	VBD	VBN
IN	-	.2			.7		
JJ	.2	-	3.3	2.1	1.7	.2	2.7
NN		8.7	-				.2
NNP	.2	3.3	4.1	-	.2		
RB	2.2	2.0	.5		-		
VBD		.3	.5			-	4.4
VBN		2.8				2.6	-

- See what errors are causing problems
 - Noun (NN) vs ProperNoun (NN) vs Adj (JJ)
 - Adverb (RB) vs Prep (IN) vs Noun (NN)
 - Preterite (VBD) vs Participle (VBN) vs Adjective (JJ)