Lecture 5 (part 1): Viterbi example

Cornell Bowers CIS
Computer Science

Claire Cardie, Tanya Goyal

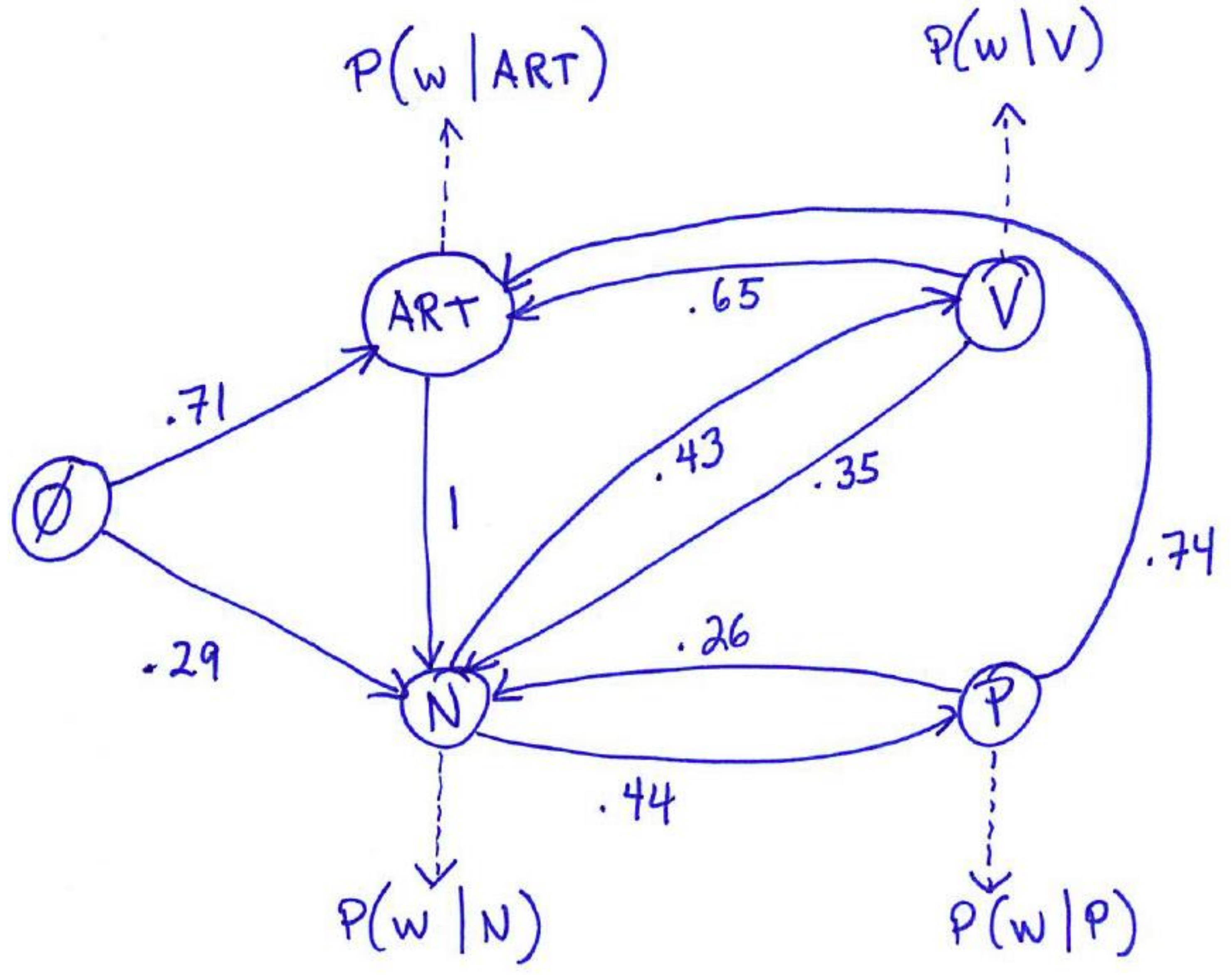
CS 4740 (and crosslists): Introduction to Natural Language Processing

Recall: HMMs for POS tagging

? ? ?
Cornell beat Harvard

$$\text{argmax}_{t_1..t_N} P(t_1 \ldots t_N \mid w_1 \ldots w_N) \cong \prod_{i=1,N} P(t_i \mid t_{i-1}) \cdot P(w_i \mid t_i)$$

- Equation is modeled by an HMM (probabilistic finite-state machine)
  - **States:** represent the possible POS
  - **Transition probabilities:** bigram probabilities for tags
  - **Emission (observation) probabilities:** indicate, for each word, how likely that word is to be selected if we randomly select a POS

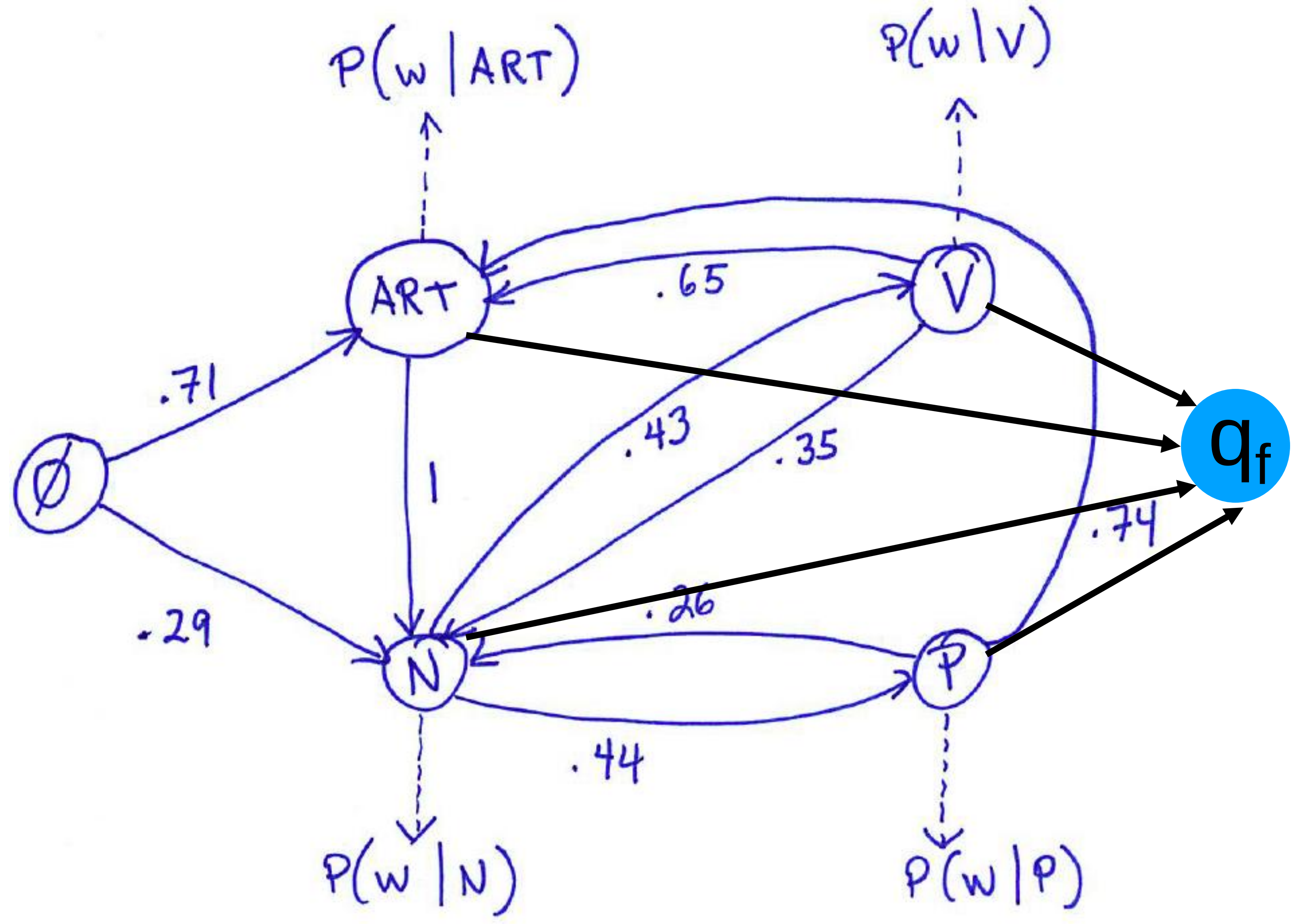


HMMs

? ? ?
Cornell beat Harvard

Given:

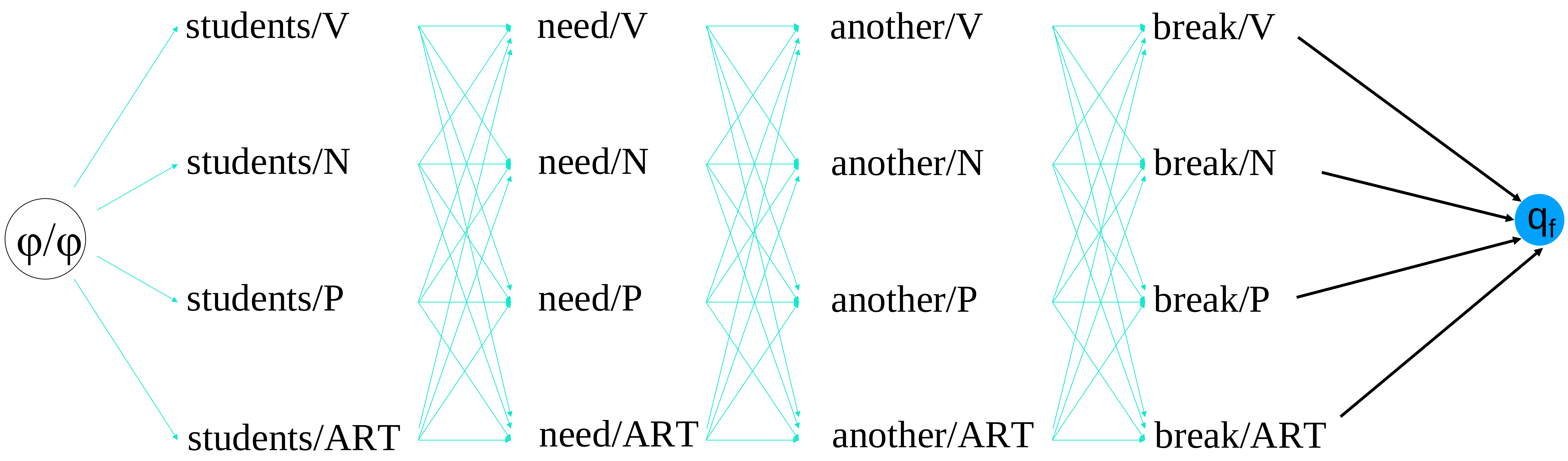
$Q = q_1 q_2 \dots q_N$	a set of N **states**
$A = a_{11} a_{12} \dots a_{n1} \dots a_{nn}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{n} a_{ij} = 1 \quad \forall i$
$O = o_1 o_2 \dots o_T$	a sequence of T **observations**, each one drawn from a vocabulary $V = v_1, v_2, \dots, v_V$
$B = b_i(o_t)$	a sequence of **observation likelihoods**, also called **emission probabilities**, each expressing the probability of an observation o_t being generated from a state i
q_0, q_F	a special **start state** and **end (final) state** that are not associated with observations, together with transition probabilities $a_{01} a_{02} \dots a_{0n}$ out of the start state and $a_{1F} a_{2F} \dots a_{nF}$ into the end state



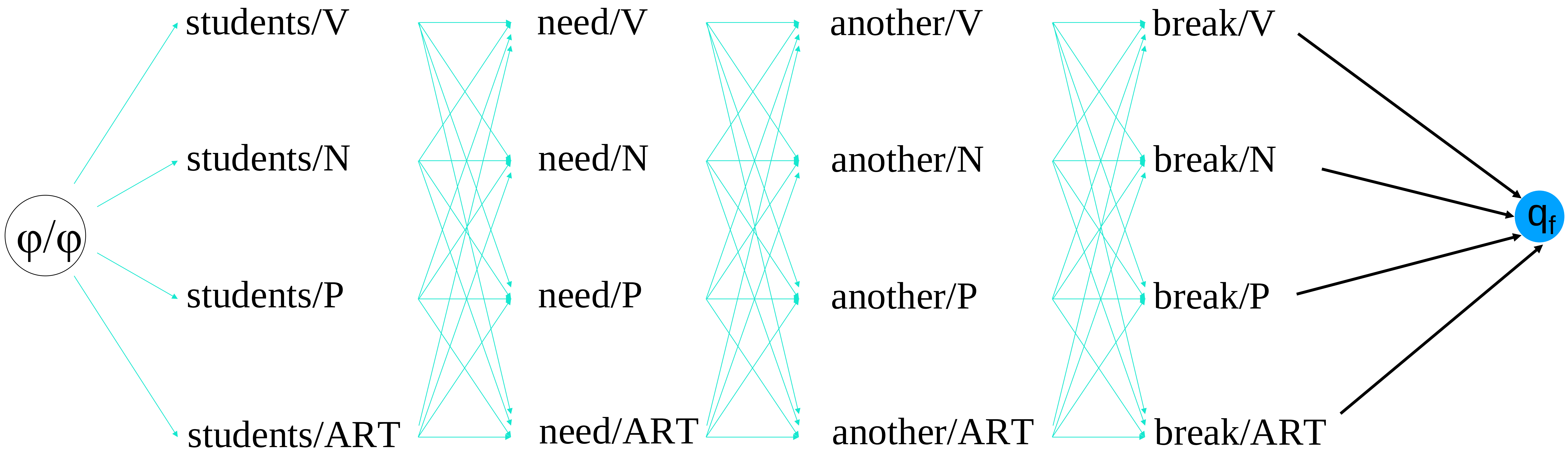
Viterbi Algorithm Allows Efficient Search for the Most Likely Sequence

- Key idea: Markov assumptions mean that we do not need to enumerate all possible sequences
- Viterbi algorithm
  - Sweep forward, one word at a time, finding the most likely (highest-scoring) tag sequence ending with each possible tag
  - With the right bookkeeping, we can then “read off” the most likely tag sequence once we reach the end of the sentence

Avoid computing the probabilities for all possible paths

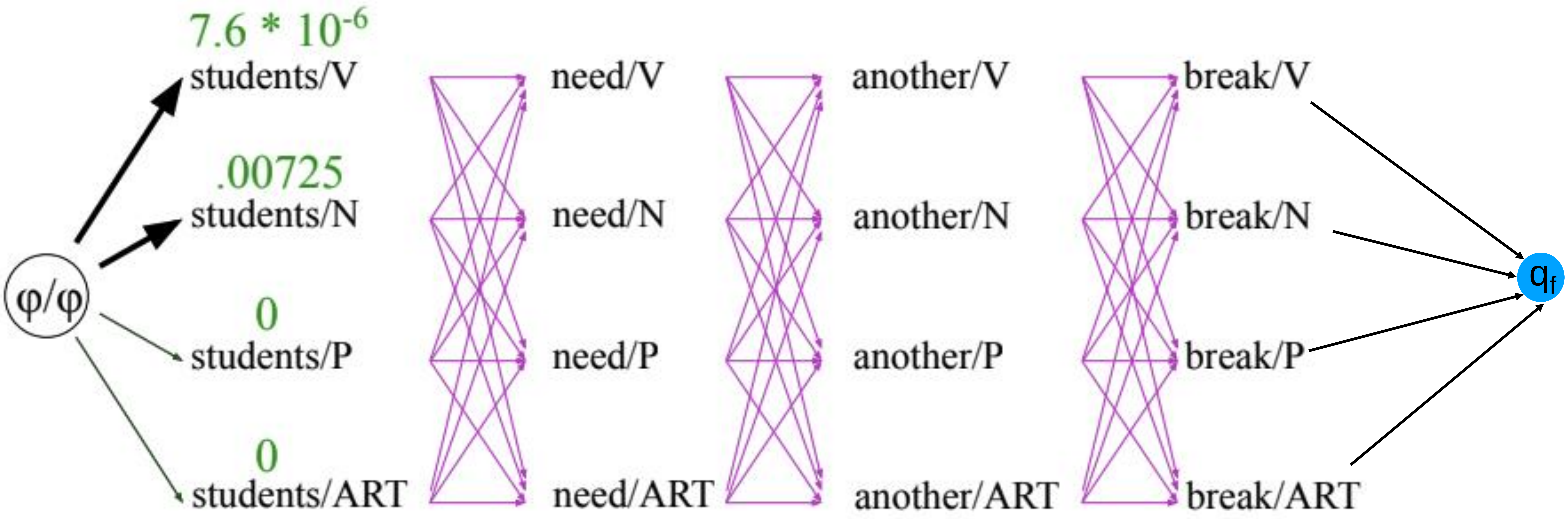


$$\prod_{i=1,n} P(t_i \mid t_{i-1}) \cdot P(w_i \mid t_i)$$



Initialization step

$$\prod_{i=1,n} P(t_i \mid t_{i-1}) \cdot P(w_i \mid t_i)$$

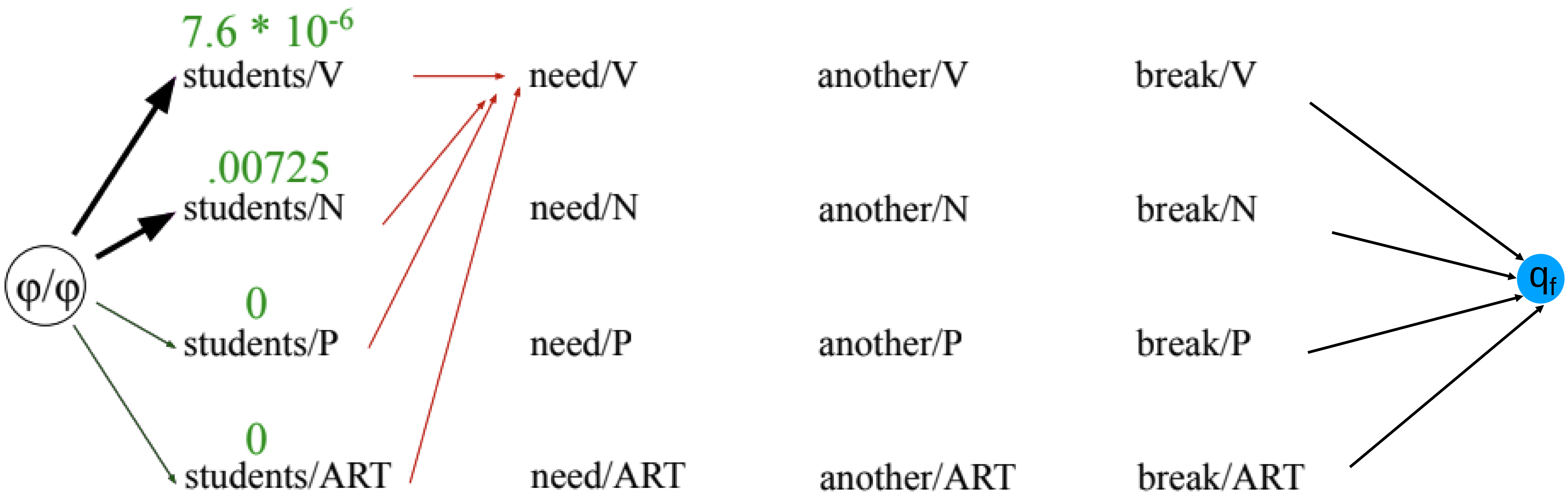


$v_0(j) = P(tag_0=j \mid φ) * P(students \mid tag_0=j)$

$vb_0(j) = φ$

Dynamic programming step(s)

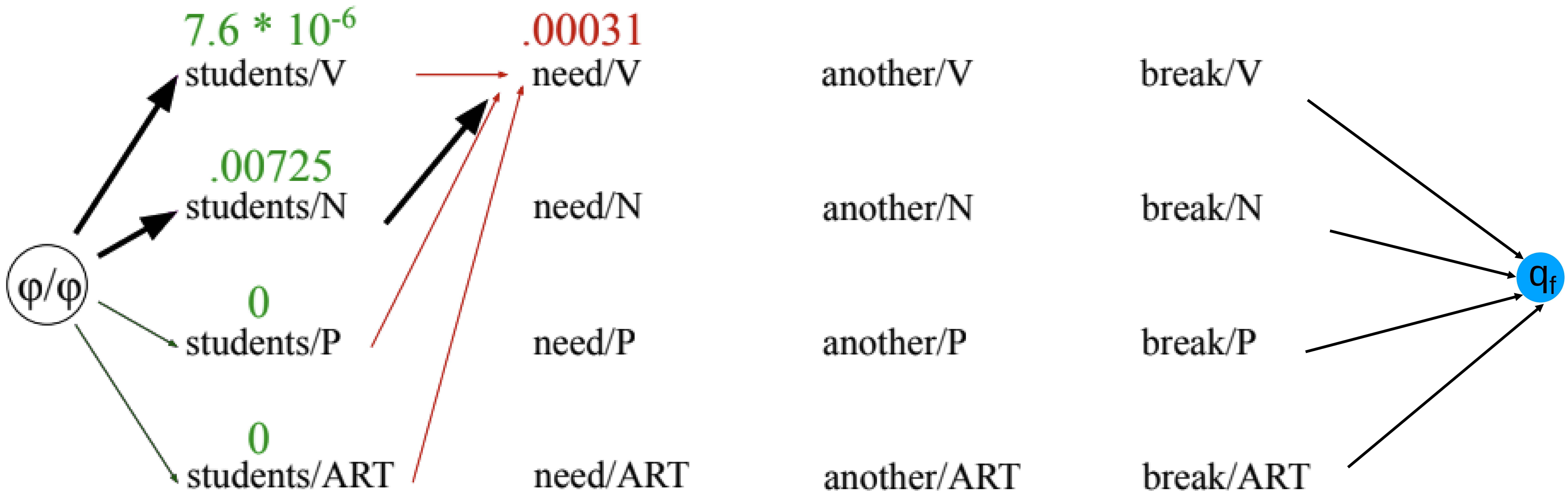
$$\prod_{i=1,n} P(t_i \mid t_{i-1}) \cdot P(w_i \mid t_i)$$



$v_1(j) = \max^J_{i=1} v_0(i) * P(tag_1=j \mid tag_0=i) * P(need \mid tag_1=j)$

$vb_t(j)$ = prev tag that maximizes $v_t(j)$

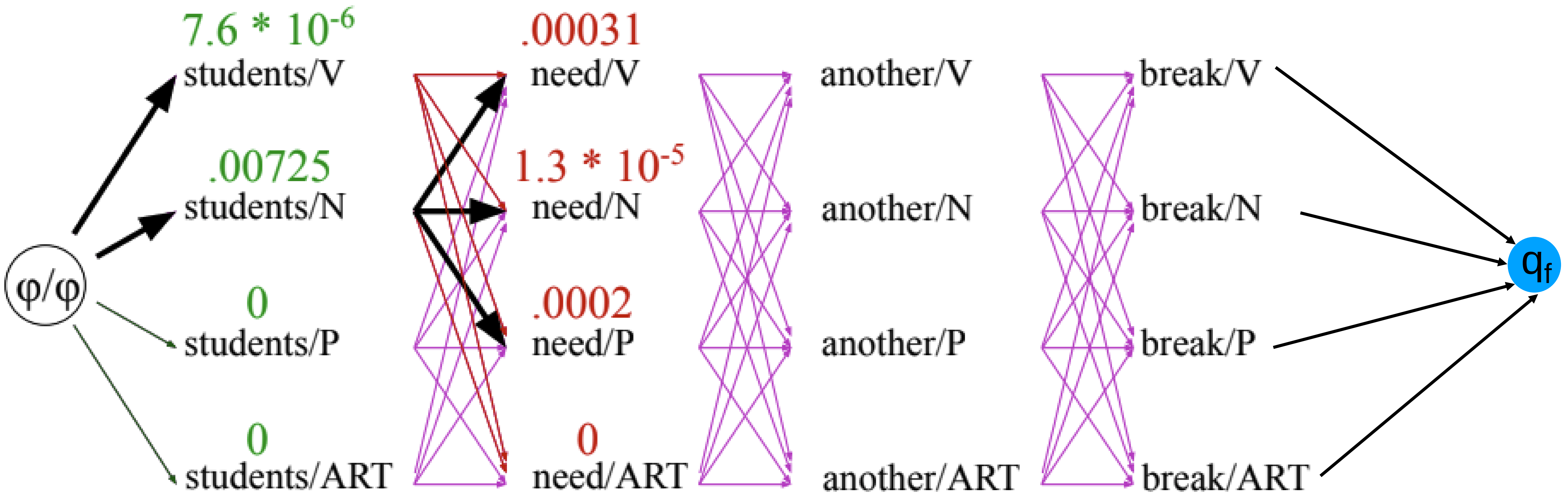
$$\prod_{i=1,n} P(t_i \mid t_{i-1}) \cdot P(w_i \mid t_i)$$



$v_1(j) = \max^{J}_{i=1} v_0(i) * P(tag_1{=}j \mid tag_0{=}i) * P(need \mid tag_1{=}j)$

$vb_t(j)$ = prev tag that maximizes $v_t(j)$

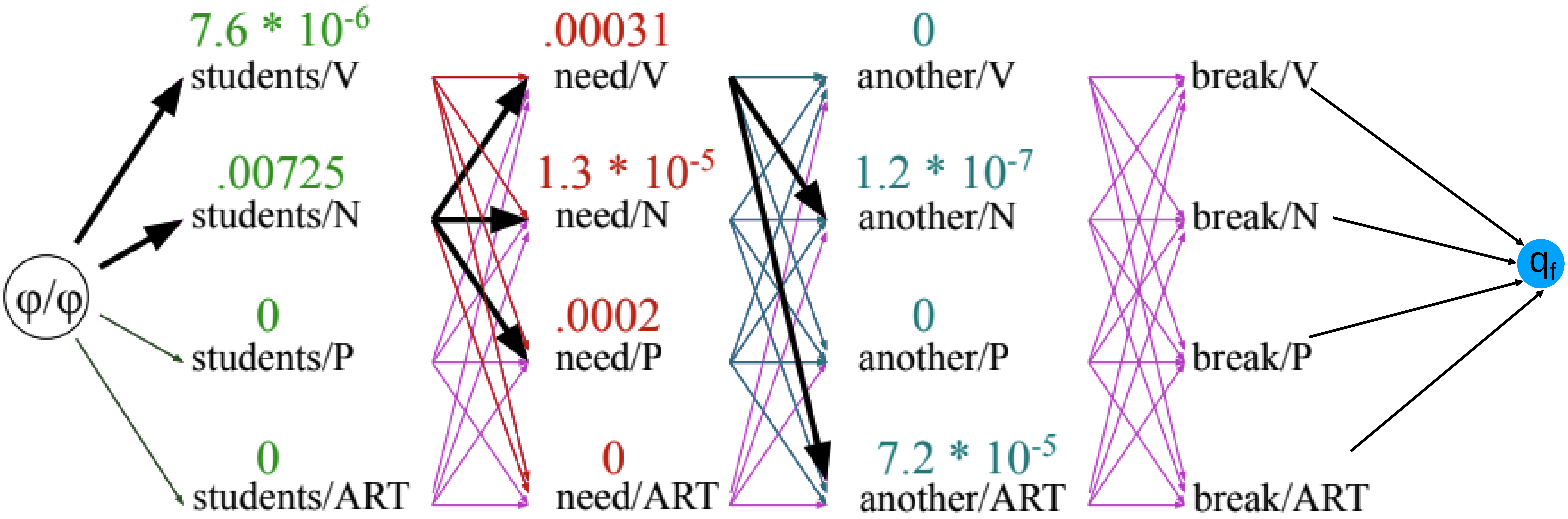
$$\prod_{i=1,n} P(t_i \mid t_{i-1}) \cdot P(w_i \mid t_i)$$



$v_1(j) = \max^{J}_{i=1} v_0(i) * P(tag_1=j \mid tag_0=i) * P(need \mid tag_1=j)$

$vb_t(j)$ = prev tag that maximizes $v_t(j)$

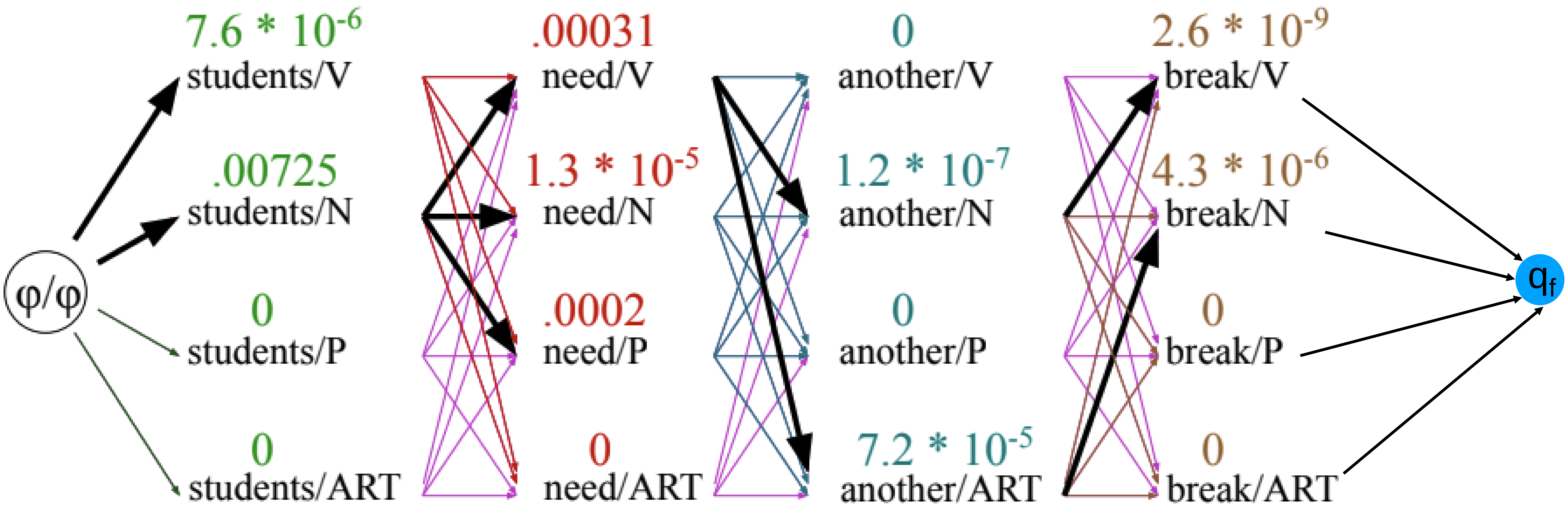
$$\prod_{i=1,n} P(t_i \mid t_{i-1}) \cdot P(w_i \mid t_i)$$



$v_2(j) = \max^{J}_{i=1} v_1(i) * P(tag_2=j \mid tag_1=i) * P(another \mid tag_2=j)$

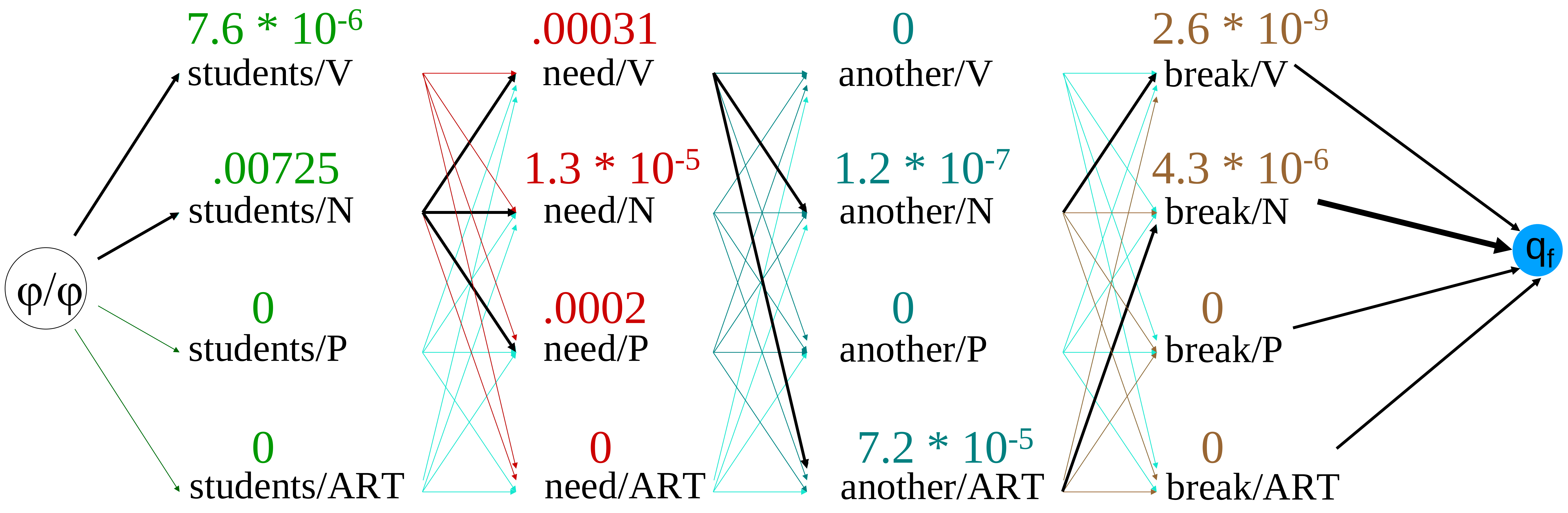
$vb_t(j)$ = prev tag that maximizes $v_t(j)$

$$\prod_{i=1,n} P(t_i \mid t_{i-1}) \cdot P(w_i \mid t_i)$$



$v_3(j) = \max^{J}_{i=1} v_2(i) * P(tag_3=j \mid tag_2=i) * P(break \mid tag_3=j)$

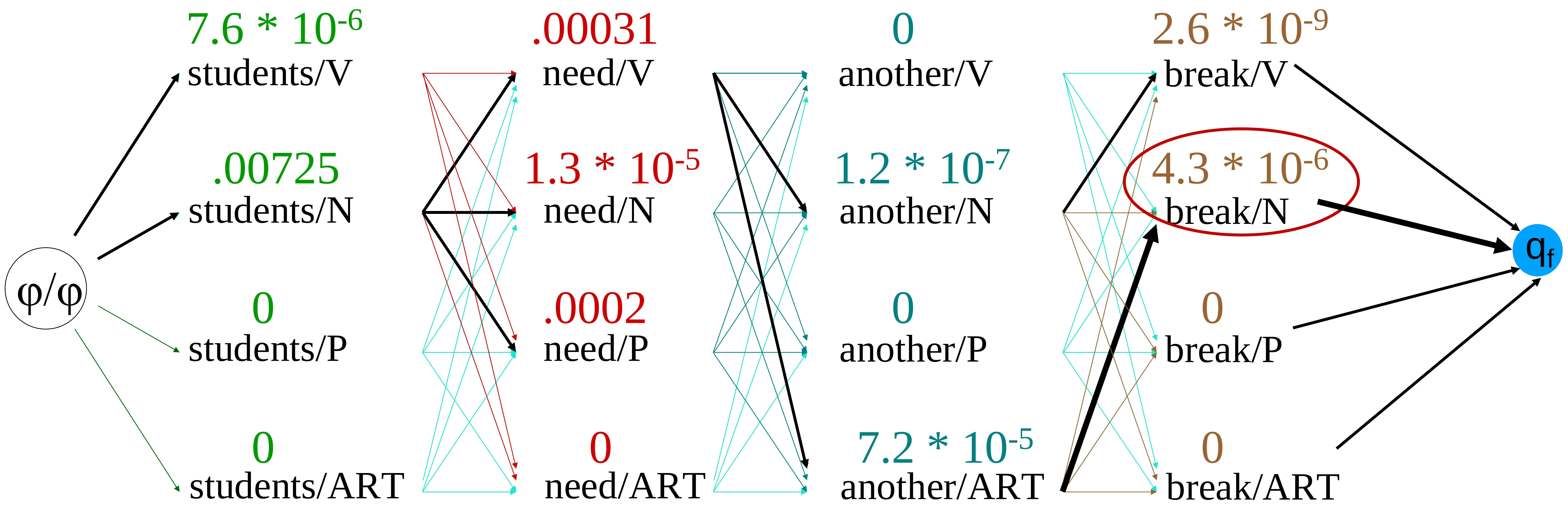
$vb_t(j)$ = prev tag that maximizes $v_t(j)$



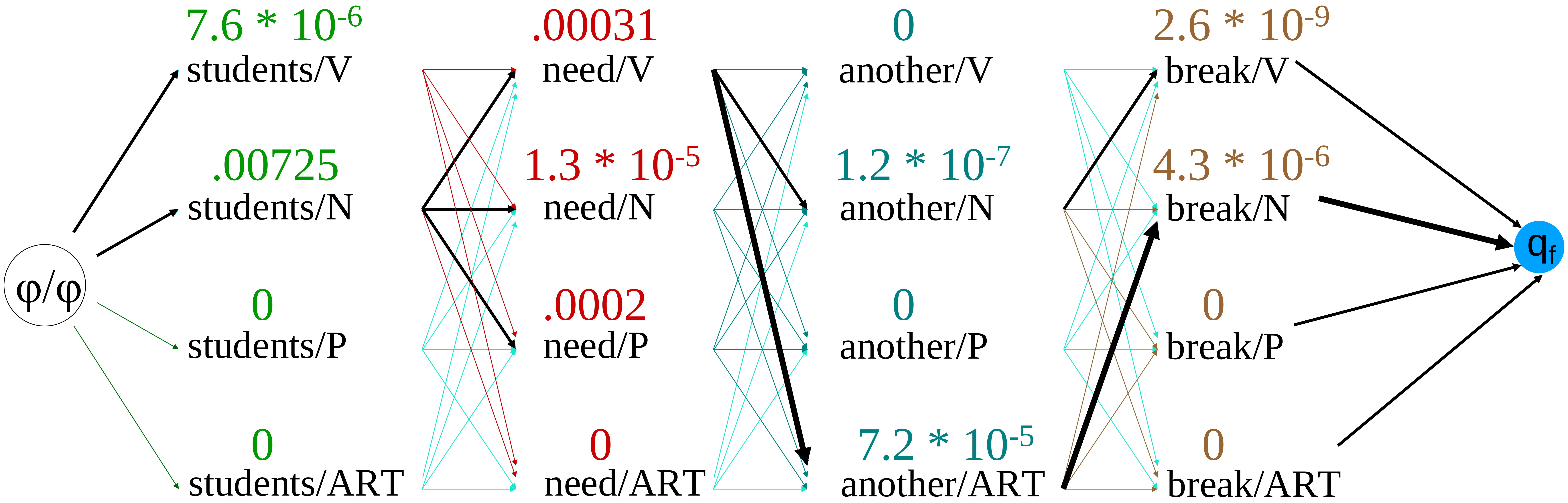
7.6 * 10-6
students/V
.00725
students/N
0
students/P
0
students/ART
.00031
need/V
1.3 * 10-5
need/N
.0002
need/P
0
need/ART
0
another/V
1.2 * 10-7
another/N
0
another/P
7.2 * 10-5
another/ART
2.6 * 10-9
break/V
4.3 * 10-6
break/N
0
break/P
0
break/ART
φ/φ
qf

$t_3 = N$

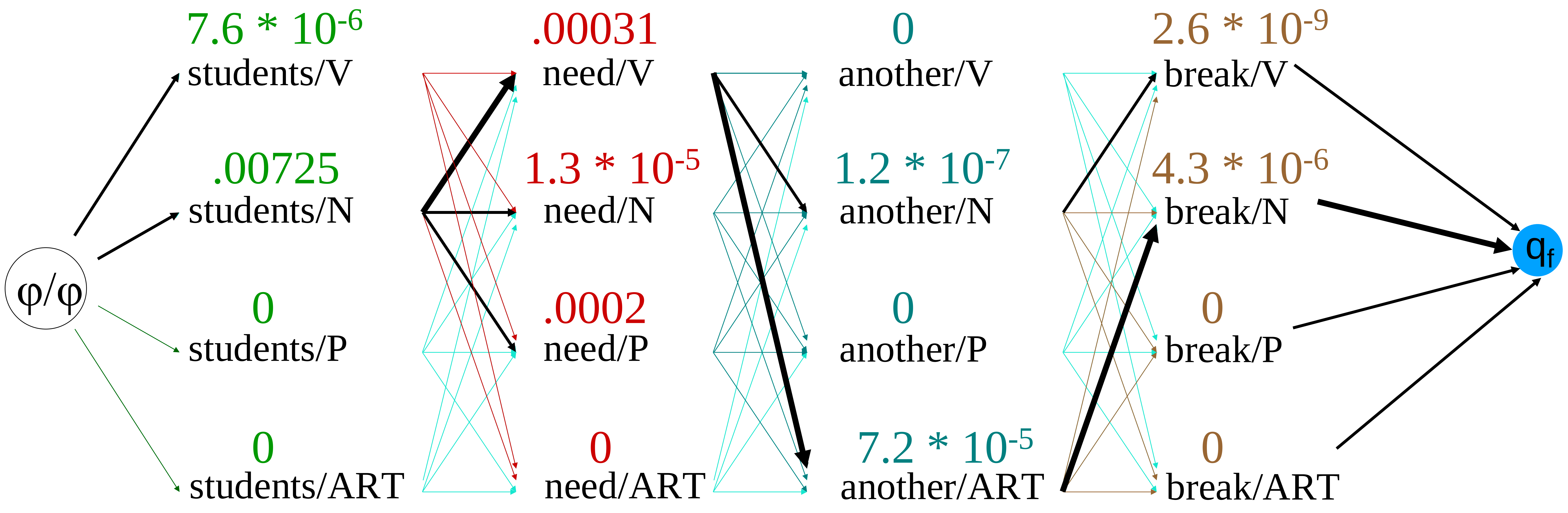
Termination: follow backpointers



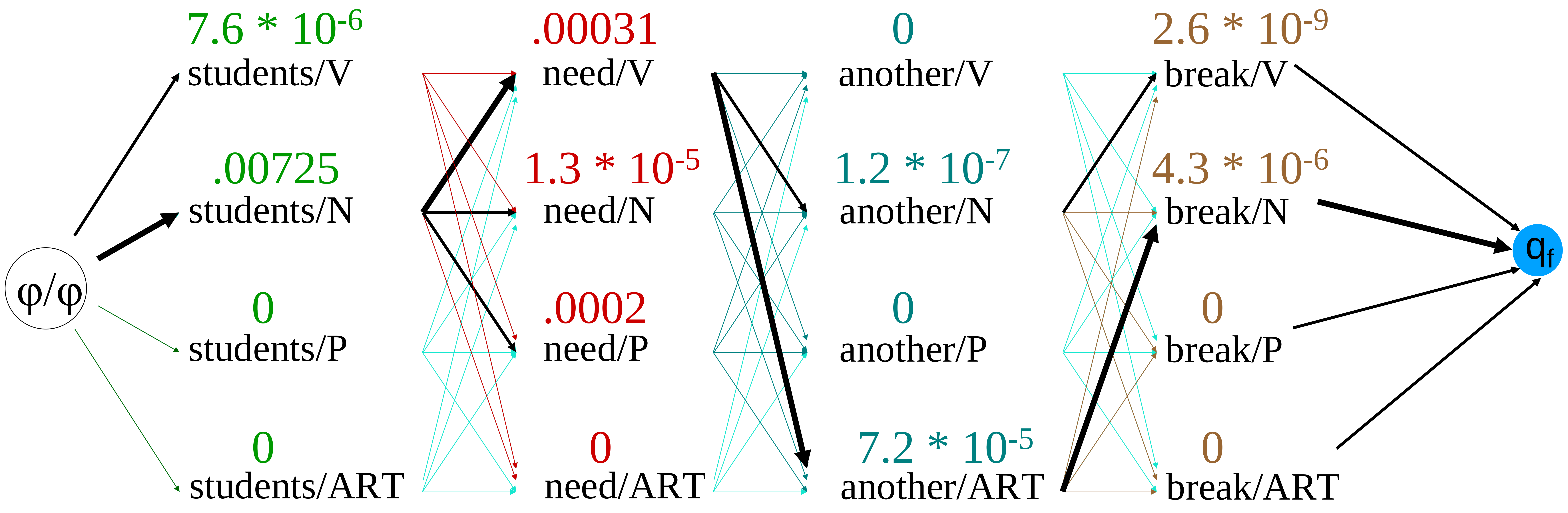
$t_3 = N$


7.6 * 10-6
students/V
.00725
students/N
0
students/P
0
students/ART
.00031
need/V
1.3 * 10-5
need/N
.0002
need/P
0
need/ART
0
another/V
1.2 * 10-7
another/N
0
another/P
7.2 * 10-5
another/ART
2.6 * 10-9
break/V
4.3 * 10-6
break/N
0
break/P
0
break/ART
φ/φ
qf

t_3= N, t_4= ART



t_3= N, t_2= ART, t_1= V



t_3= N, t_2= ART, t_1= V, t_0= N

An Example: weather/ice-cream HMM

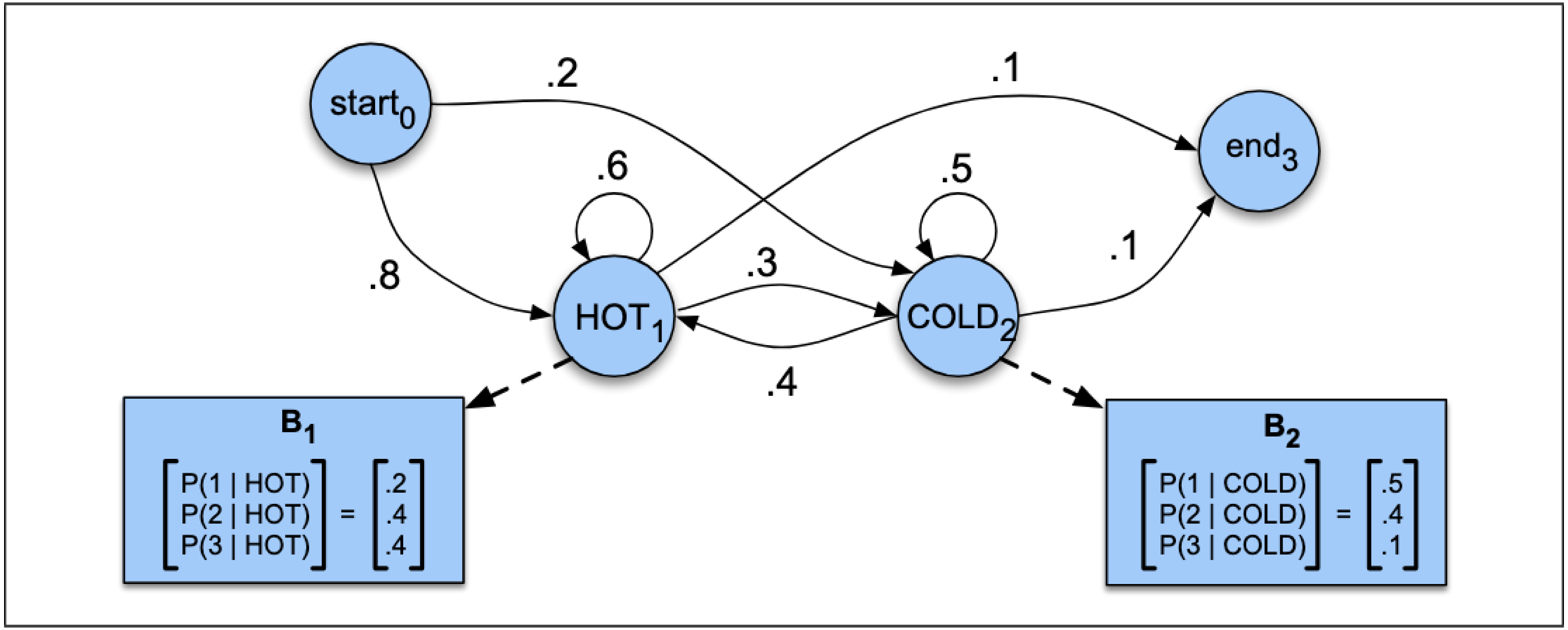


Figure 9.3 A hidden Markov model for relating numbers of ice creams eaten by Jason (the observations) to the weather (H or C, the hidden variables).

An Example: weather/ice-cream HMM

$Q = q_1 q_2 \ldots q_N$	a set of N **states**
$A = a_{11} a_{12} \ldots a_{n1} \ldots a_{nn}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{n} a_{ij} = 1 \quad \forall i$
$O = o_1 o_2 \ldots o_T$	a sequence of T **observations**, each one drawn from a vocabulary $V = v_1, v_2, ..., v_V$
$B = b_i(o_t)$	a sequence of **observation likelihoods**, also called **emission probabilities**, each expressing the probability of an observation o_t being generated from a state i
q_0, q_F	a special **start state** and **end (final) state** that are not associated with observations, together with transition probabilities $a_{01} a_{02} \ldots a_{0n}$ out of the start state and $a_{1F} a_{2F} \ldots a_{nF}$ into the end state

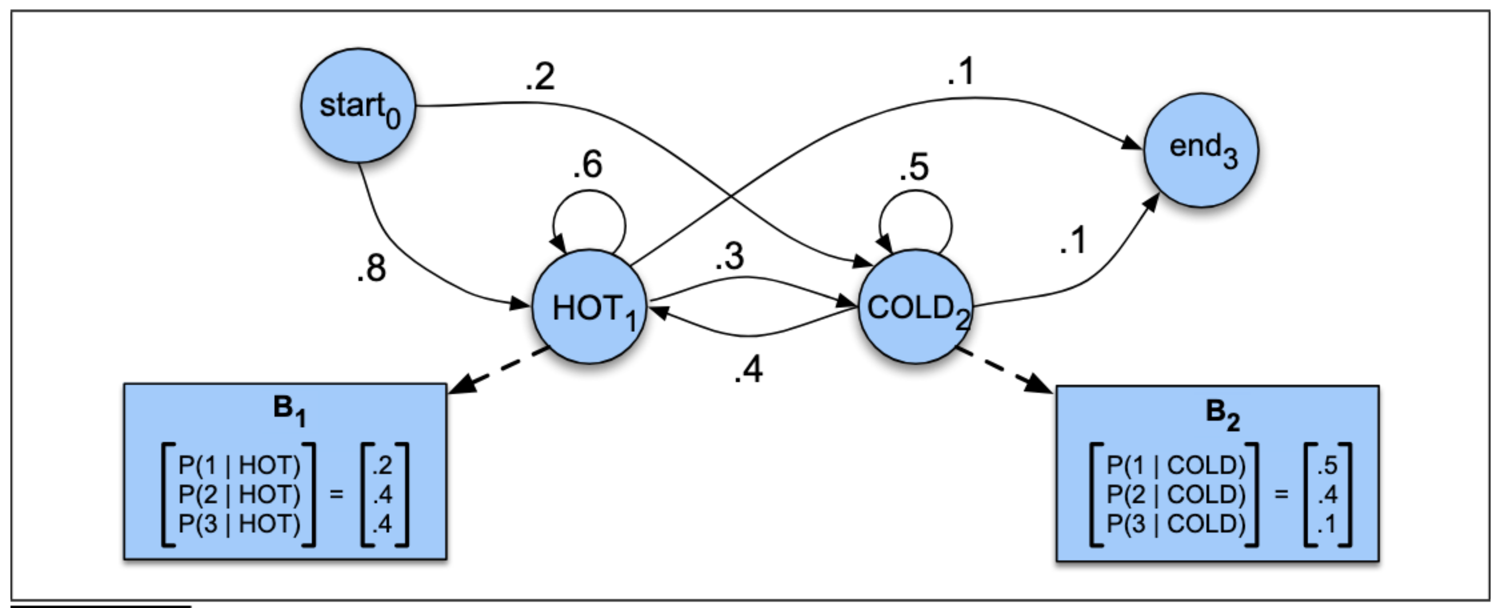


Figure 9.3 A hidden Markov model for relating numbers of ice creams eaten by Jason (the observations) to the weather (H or C, the hidden variables).

? ? ?

1(few) 3(lots) 2(mid)

Intuitions: weather/ice-cream HMM

States: q_o= START, q_1 = Cold day, q_2= Hot day, q_F = END

Vocabulary: "few" (ice creams eaten), "mid", "lot"

A, the transitions matrix

	H	C	q_F
H	0.7	0.2	0.1
C	0.4	0.5	0.1
q_0	0.8	0.2	

B, the emission "rows"

	few	mid	lot
H	0.1	0.3	0.6
C	0.5	0.4	0.1

- A priori, day 1 (after day 0) is more likely to be Hot (.8) than Cold (0.2).
- Hot days mostly likely result in a lot of eating (H row)
- Each day, tomorrow's weather will likely the same as today's. (A's diagonal)

Visual indexing convention: lower-left is (0,0).Row numbering increases upward.

We also omitted row 3 (nothing transitions from q_F = END) and column 0 (nothing transitions into q_0= START).

Example: weather/ice-cream HMM

@thissillygirlskitchen

States: q_o= START, q_1 = Cold day, q_2= Hot day, q_F = END

Vocabulary: "few" (ice creams eaten), "mid", "lot"

A, the transitions matrix

	H	C	q_F
H	0.7	0.2	0.1
C	0.4	0.5	0.1
q_0	0.8	0.2	

B, the emission "rows"

	few	mid	lot
H	0.1	0.3	0.6
C	0.5	0.4	0.1

Q1: if today is cold, what is the probability that a "lot" are eaten today?
Q2: … and what's the probability that tomorrow is cold?
Q3: If the eating records show "mid mid few", what was the weather then?

Visual indexing convention: lower-left is (0,0).Row numbering increases <u>upward.</u>
We also omitted row 3 (nothing transitions from q_F = END) and column 0 (nothing transitions into q_0= START).

Example: weather/ice-cream HMM

@thissillygirlskitchen

States: q_o= START, q_1 = Cold day, q_2= Hot day, q_F = END

Vocabulary: “few” (ice creams eaten), “mid”, “lot”

A, the transitions matrix

	H	C	q_F
H	0.7	0.2	0.1
C	0.4	0.5	0.1
q_0	0.8	0.2	

B, the emission “rows”

	few	mid	lot
H	0.1	0.3	0.6
C	0.5	0.4	0.1

Visual indexing convention: lower-left is (0,0).Row numbering increases upward.

We also omitted row 3 (nothing transitions from q_F = END) and column 0 (nothing transitions into q_0= START).

Viterbi question: Given observation “<s> **mid mid few** </s>”, what state sequence assigns the highest likelihood?

The Viterbi chart v(state, observation):

This stores "the max prob of getting to us to this observation and this state".

A (Transitions)

	H	C	q_F
H	0.7	0.2	0.1
C	0.4	0.5	0.1
q_0	0.8	0.2	

B (Emissions)

	few	mid	lot
H	0.1	0.3	0.6
C	0.5	0.4	0.1

V Matrix Row 3 (Paths that go through State 3) → q_F

V Matrix Row 2 (Paths that go through State 2) → H

V Matrix Row 1 (Paths that go through State 1) → C

V Matrix Row 0 (Paths that go through State 0) → q_0

q_F				
H				
C				
q_0				
	mid	mid	few	<END>

The backpointer matrix

	H	C	q_F
H	0.7	0.2	0.1
C	0.4	0.5	0.1
q_0	0.8	0.2	

	few	mid	lot
H	0.1	0.3	0.6
C	0.5	0.4	0.1

v =

q_F				
H				
C				
q_0				
	mid	mid	few	<END>

vb =

				q_F
H				
C				

Preview. The goal is the backpointers.

1) Initialize

$v_1(j) = a_{0j} b_j(o_1)$

$vb_1(j) = 0 \qquad 1 \le j \le J;\ v=(J,T)$

2) Dynamic programming step

$v_t(j) = \max^J_{i=1} v_{t-1}(i) a_{ij} b_j(o_t) \qquad 1 \le j \le J;\ 1 < t \le T$

$vb_t(j) = \mathrm{argmax}_{i,t-1}[v_t(j)]$

3) Termination

$P(O|\Theta,Q) = v_T(F) = \max^J_{i=1} v_T(i) a_{iF}$

$\mathrm{argmax}_Q[P(O|\Theta,Q)] = vb_T(F)$

Read off the best tag sequence "backwards" from q_F in vb to find that it is H H C.
(Actually, q_0 H H C q_F.)

	H	C	q_F
H	0.7	0.2	0.1
C	0.4	0.5	0.1
q_0	0.8	0.2	

	few	mid	lot
H	0.1	0.3	0.6
C	0.5	0.4	0.1

vb =

				q_F
H	0= q_0	2 =H	2	1 = C
C	0	2	2 =H	

v =

q_F				0.000504
H	0.24	0.0504	0.003528	
C	0.08	0.0192	0.00504	
q_0				
	mid	mid	few	<END>

The Viterbi Algorithm (HMM)

	H	C	q_F
H	0.7	0.2	0.1
C	0.4	0.5	0.1
q_0	0.8	0.2	

	few	mid	lot
H	0.1	0.3	0.6
C	0.5	0.4	0.1

1) Initialize

$v_1(j) = a_{0j} b_j(o_1)$

$vb_1(j) = 0 \quad 1 \leq j \leq J;\ v=(J,T)$

2) Dynamic programming step

$v_t(j) = \max^J_{i=1} v_{t-1}(i) a_{ij} b_j(o_t) \quad 1 \leq j \leq J;\ 1 < t \leq T$

$vb_t(j) = \text{argmax}_{i,t-1}[v_t(j)]$

3) Termination

$P(O|\Theta,Q) = v_T(F) = \max^J_{i=1} v_T(i) a_{iF}$

$\text{argmax}_Q[P(O|\Theta,Q)] = vb_T(F)$

vb =

					q_F
H					
C					

v =

q_F				
H				
C				
q_0				
	mid	mid	few	<END>

Initializing v matrix

	H	C	q_F
H	0.7	0.2	0.1
C	0.4	0.5	0.1
q_0	0.8	0.2	

	few	mid	lot
H	0.1	0.3	0.6
C	0.5	0.4	0.1

1) Initialize

$v_1(j) = a_{0j}b_j(o_1)$

$vb_1(j) = 0 \quad 1 \le j \le J;\ v=(J,T)$

2) Dynamic programming step

$v_t(j) = \max^J_{i=1} v_{t-1}(i) a_{ij} b_j(o_t) \quad 1 \le j \le J;\ 1 < t \le T$

$vb_t(j) = \mathrm{argmax}_{i,t-1}[v_t(j)]$

3) Termination

$P(O|\Theta,Q) = v_T(F) = \max^J_{i=1} v_T(i) a_{iF}$

$\mathrm{argmax}_Q[P(O|\Theta,Q)] = vb_T(F)$

vb =

				q_F
H	0			
C	0 = q_0			

v =

q_F				
H	0.8*0.3			
C	0.2*0.4			
q_0				
	mid	mid	few	<END>

(do the multiply)

1) Initialize

$v_1(j) = a_{0j} b_j(o_1)$

$vb_1(j) = 0 \qquad 1 \leq j \leq J;\ v=(J,T)$

2) Dynamic programming step

$v_t(j) = \max^J_{i=1} v_{t-1}(i) a_{ij} b_j(o_t) \qquad 1 \leq j \leq J;\ 1 < t \leq T$

$vb_t(j) = \mathrm{argmax}_{i,t-1}[v_t(j)]$

3) Termination

$P(O|\Theta,Q) = v_T(F) = \max^J_{i=1} v_T(i) a_{iF}$

$\mathrm{argmax}_Q[P(O|\Theta,Q)] = vb_T(F)$

	H	C	q_F
H	0.7	0.2	0.1
C	0.4	0.5	0.1
q_0	0.8	0.2	

	few	mid	lot
H	0.1	0.3	0.6
C	0.5	0.4	0.1

vb =

				q_F
H	0			
C	0			

v =

q_F				
H	0.24			
C	0.08			
q_0				
	mid	mid	few	<END>

Next column!

	H	C	q_F
H	0.7	0.2	0.1
C	0.4	0.5	0.1
q_0	0.8	0.2	

	few	mid	lot
H	0.1	0.3	0.6
C	0.5	0.4	0.1

1) Initialize

$v_1(j) = a_{0j} b_j(o_1)$

$vb_1(j) = 0 \quad 1 \leq j \leq J; v=(J,T)$

2) Dynamic programming step

$v_t(j) = \max^J_{i=1} v_{t-1}(i) a_{ij} b_j(o_t) \quad 1 \leq j \leq J; 1< t \leq T$

$vb_t(j) = \text{argmax}_{i,t-1}[v_t(j)]$

3) Termination

$P(O|\Theta,Q) = v_T(F) = \max^J_{i=1} v_T(i) a_{iF}$

$\text{argmax}_Q[P(O|\Theta,Q)] = vb_T(F)$

vb =

				q_F
H	0			
C	0			

v =

q_F				
H	0.24	max(0.24*0.7*0.3,0.08*0.4*0.3)		
C	0.08	max(0.24*0.2*0.4, 0.08*0.5*0.4)		
q_0				
	mid	mid	few	<end>

(do the math on the product)

	H	C	q_F
H	0.7	0.2	0.1
C	0.4	0.5	0.1
q_0	0.8	0.2	

	few	mid	lot
H	0.1	0.3	0.6
C	0.5	0.4	0.1

1) Initialize

$v_1(j) = a_{0j} b_j(o_1)$

$vb_1(j) = 0 \qquad 1 \leq j \leq J;\ v=(J,T)$

2) Dynamic programming step

$v_t(j) = \max^J_{i=1} v_{t-1}(i) a_{ij} b_j(o_t) \qquad 1 \leq j \leq J;\ 1 < t \leq T$

$vb_t(j) = \text{argmax}_{i,t-1}[v_t(j)]$

3) Termination

$P(O|\Theta,Q) = v_T(F) = \max^J_{i=1} v_T(i) a_{iF}$

$\text{argmax}_Q[P(O|\Theta,Q)] = vb_T(F)$

vb =

				q_F
H	0			
C	0			

v =

q_F				
H	0.24	max(0.0504, 0.0096)		
C	0.08	max(0.0192, 0.016)		
q_0				
	mid	mid	few	

Get backpointers to best previous state

	H	C	q_F
H	0.7	0.2	0.1
C	0.4	0.5	0.1
q_0	0.8	0.2	

	few	mid	lot
H	0.1	0.3	0.6
C	0.5	0.4	0.1

1) Initialize
   $v_1(j) = a_{0j} b_j(o_1)$
   $vb_1(j) = 0 \quad 1 \leq j \leq J;\ v=(J,T)$

2) Dynamic programming step
   $v_t(j) = \max^J_{i=1} v_{t-1}(i) a_{ij} b_j(o_t) \quad 1 \leq j \leq J;\ 1 < t \leq T$
   $vb_t(j) = \mathrm{argmax}_{i,t-1}[v_t(j)]$

3) Termination
   $P(O|\Theta,Q) = v_T(F) = \max^J_{i=1} v_T(i) a_{iF}$
   $\mathrm{argmax}_Q[P(O|\Theta,Q)] = vb_T(F)$

vb =

				q_F
H	0	2		
C	0	2 =H		

v =

q_F				
H	0.24	0.0504		
C	0.08	0.0192		
q_0				
	mid	mid	few	

Done w/ 2nd "mid" column

	H	C	q_F
H	0.7	0.2	0.1
C	0.4	0.5	0.1
q_0	0.8	0.2	

	few	mid	lot
H	0.1	0.3	0.6
C	0.5	0.4	0.1

1) Initialize

$v_1(j) = a_{0j} b_j(o_1)$

$vb_1(j) = 0 \quad 1 \le j \le J;\ v=(J,T)$

2) Dynamic programming step

$v_t(j) = \max^J_{i=1} v_{t-1}(i) a_{ij} b_j(o_t) \quad 1 \le j \le J;\ 1 < t \le T$

$vb_t(j) = \text{argmax}_{i,t-1}[v_t(j)]$

3) Termination

$P(O|\Theta,Q) = v_T(F) = \max^J_{i=1} v_T(i) a_{iF}$

$\text{argmax}_Q[P(O|\Theta,Q)] = vb_T(F)$

vb =

				q_F
H	0	2		
C	0	2		

v =

q_F				
H	0.24	0.0504		
C	0.08	0.0192		
q_0				
	mid	mid	few	

Next column (for observation "few")

1) Initialize

$v_1(j) = a_{0j} b_j(o_1)$

$vb_1(j) = 0 \quad 1 \le j \le J; v=(J,T)$

2) Dynamic programming step

$v_t(j) = \max^J_{i=1} v_{t-1}(i) a_{ij} b_j(o_t) \quad 1 \le j \le J; 1 < t \le T$

$vb_t(j) = \text{argmax}_{i,t-1}[v_t(j)]$

3) Termination

$P(O|\Theta,Q) = v_T(F) = \max^J_{i=1} v_T(i) a_{iF}$

$\text{argmax}_Q[P(O|\Theta,Q)] = vb_T(F)$

	H	C	q_F
H	0.7	0.2	0.1
C	0.4	0.5	0.1
q_0	0.8	0.2	

	few	mid	lot
H	0.1	0.3	0.6
C	0.5	0.4	0.1

vb =

				q_F
H	0	2		
C	0	2		

v =

q_F				
H	0.24	0.0504	max(0.0504*0.7*0.1, 0.0192*0.4*0.1)	
C	0.08	0.0192	max(0.0504*0.2*0.5, 0.0192*0.5*0.5)	
q_0				
	mid	mid	few	<END>

(compute the products)

1) Initialize

$v_1(j) = a_{0j} b_j(o_1)$

$vb_1(j) = 0 \qquad 1 \leq j \leq J;\ v=(J,T)$

2) Dynamic programming step

$v_t(j) = \max^J_{i=1} v_{t-1}(i) a_{ij} b_j(o_t) \qquad 1 \leq j \leq J;\ 1 < t \leq T$

$vb_t(j) = \mathrm{argmax}_{i,t-1}[v_t(j)]$

3) Termination

$P(O|\Theta,Q) = v_T(F) = \max^J_{i=1} v_T(i) a_{iF}$

$\mathrm{argmax}_Q[P(O|\Theta,Q)] = vb_T(F)$

	H	C	q_F
H	0.7	0.2	0.1
C	0.4	0.5	0.1
q_0	0.8	0.2	

	few	mid	lot
H	0.1	0.3	0.6
C	0.5	0.4	0.1

vb =

				q_F
H	0	2		
C	0	2		

v =

q_F				
H	0.24	0.0504	max(0.003528, 0.000768)	
C	0.08	0.0192	max(0.00504, 0.0048)	
q_0				
	mid	mid	few	<END>

Put the argmax into vb.

1) Initialize

$v_1(j) = a_{0j} b_j(o_1)$

$vb_1(j) = 0 \quad 1 \leq j \leq J;\ v=(J,T)$

2) Dynamic programming step

$v_t(j) = \max^J_{i=1} v_{t-1}(i) a_{ij} b_j(o_t) \quad 1 \leq j \leq J;\ 1 < t \leq T$

$vb_t(j) = \text{argmax}_{i,t-1}[v_t(j)]$

3) Termination

$P(O|\Theta,Q) = v_T(F) = \max^J_{i=1} v_T(i) a_{iF}$

$\text{argmax}_Q[P(O|\Theta,Q)] = vb_T(F)$

	H	C	q_F
H	0.7	0.2	0.1
C	0.4	0.5	0.1
q_0	0.8	0.2	

	few	mid	lot
H	0.1	0.3	0.6
C	0.5	0.4	0.1

vb =

				q_F
H	0	2	2	
C	0	2	2 =H	

v =

q_F				
H	0.24	0.0504	0.003528	
C	0.08	0.0192	0.00504	
q_0				
	mid	mid	few	<END>

Termination time

	H	C	q_F
H	0.7	0.2	0.1
C	0.4	0.5	0.1
q_0	0.8	0.2	

	few	mid	lot
H	0.1	0.3	0.6
C	0.5	0.4	0.1

1) Initialize

$v_1(j) = a_{0j} b_j(o_1)$

$vb_1(j) = 0 \qquad 1 \leq j \leq J;\ v=(J,T)$

2) Dynamic programming step

$v_t(j) = \max^{J}_{i=1} v_{t-1}(i) a_{ij} b_j(o_t) \qquad 1 \leq j \leq J;\ 1 < t \leq T$

$vb_t(j) = \text{argmax}_{i,t-1}[v_t(j)]$

3) Termination

$P(O|\Theta,Q) = v_T(F) = \max^{J}_{i=1} v_T(i) a_{iF}$

$\text{argmax}_Q[P(O|\Theta,Q)] = vb_T(F)$

vb =

				q_F
H	0	2	2	
C	0	2	2	

v =

q_F				max(0.1*0.003528, 0.1*0.00504)
H	0.24	0.0504	0.003528	
C	0.08	0.0192	0.00504	
q_0				
	mid	mid	few	<END>

(compute the product)

1) Initialize

$v_1(j) = a_{0j} b_j(o_1)$

$vb_1(j) = 0 \qquad 1 \le j \le J;\ v=(J,T)$

2) Dynamic programming step

$v_t(j) = \max^{J}_{i=1} v_{t-1}(i) a_{ij} b_j(o_t) \qquad 1 \le j \le J;\ 1 < t \le T$

$vb_t(j) = \text{argmax}_{i,t-1}[v_t(j)]$

3) Termination

$P(O|\Theta,Q) = v_T(F) = \max^{J}_{i=1} v_T(i) a_{iF}$

$\text{argmax}_Q[P(O|\Theta,Q)] = vb_T(F)$

	H	C	q_F
H	0.7	0.2	0.1
C	0.4	0.5	0.1
q_0	0.8	0.2	

	few	mid	lot
H	0.1	0.3	0.6
C	0.5	0.4	0.1

vb =

				q_F
H	0	2	2	
C	0	2	2	

v =

q_F				max(0.0003528, 0.000504)
H	0.24	0.0504	0.003528	
C	0.08	0.0192	0.00504	
q_0				
	mid	mid	few	<END>

Final backpointer!

(the red-bkgd square, in wrong row b/c of slide space constraints)

	H	C	q_F
H	0.7	0.2	0.1
C	0.4	0.5	0.1
q_0	0.8	0.2	

	few	mid	lot
H	0.1	0.3	0.6
C	0.5	0.4	0.1

1) Initialize

$v_1(j) = a_{0j} b_j(o_1)$

$vb_1(j) = 0 \qquad 1 \le j \le J;\ v=(J,T)$

2) Recursion

$v_t(j) = \max^J_{i=1} v_{t-1}(i) a_{ij} b_j(o_t) \qquad 1 \le j \le J;\ 1 < t \le T$

$vb_t(j) = \text{argmax}_{i,t-1}[v_t(j)]$

3) Termination

$P(O|\Theta,Q) = v_T(F) = \max^J_{i=1} v_T(i) a_{iF}$

$\text{argmax}_Q[P(O|\Theta,Q)] = vb_T(F)$

vb =

				q_F
H	0	2	2	1 = C
C	0	2	2	

v =

q_F				0.000504
H	0.24	0.0504	0.003528	
C	0.08	0.0192	0.00504	
q_0				
	mid	mid	few	<END>

Reconstruct the best tag sequence

	H	C	q_F
H	0.7	0.2	0.1
C	0.4	0.5	0.1
q_0	0.8	0.2	

	few	mid	lot
H	0.1	0.3	0.6
C	0.5	0.4	0.1

1) Initialize

$v_1(j) = a_{0j} b_j(o_1)$

$vb_1(j) = 0 \qquad 1 \leq j \leq J;\ v=(J,T)$

2) Dynamic programming step

$v_t(j) = \max^J_{i=1} v_{t-1}(i) a_{ij} b_j(o_t) \qquad 1 \leq j \leq J;\ 1 < t \leq T$

$vb_t(j) = \operatorname{argmax}_{i,t-1}[v_t(j)]$

3) Termination

$P(O|\Theta,Q) = v_T(F) = \max^J_{i=1} v_T(i) a_{iF}$

$\operatorname{argmax}_Q[P(O|\Theta,Q)] = vb_T(F)$

vb =

				q_F
H	0 = q_0	2 =H	2	1 = C
C	0	2	2 =H	

v =

q_F				0.000504
H	0.24	0.0504	0.003528	
C	0.08	0.0192	0.00504	
q_0				
	mid	mid	few	<END>

The best tag sequence: H H C