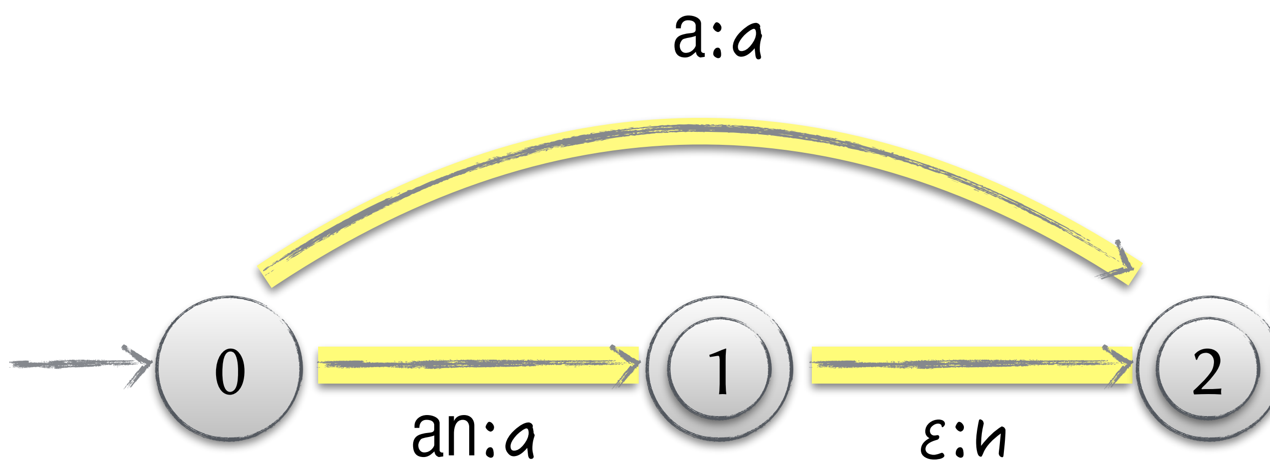
Automatic Speech Recognition (CS753)

Lecture 5: Hidden Markov Models (Part I)

Instructor: Preethi Jyothi
Lecture 5

OpenFst Cheat Sheet

Quick Intro to OpenFst (www.openfst.org)



"0" label is reserved for epsilon

0	1	an	a
1	2	<eps>	n
0	2	a	a
1			
2			

A.txt

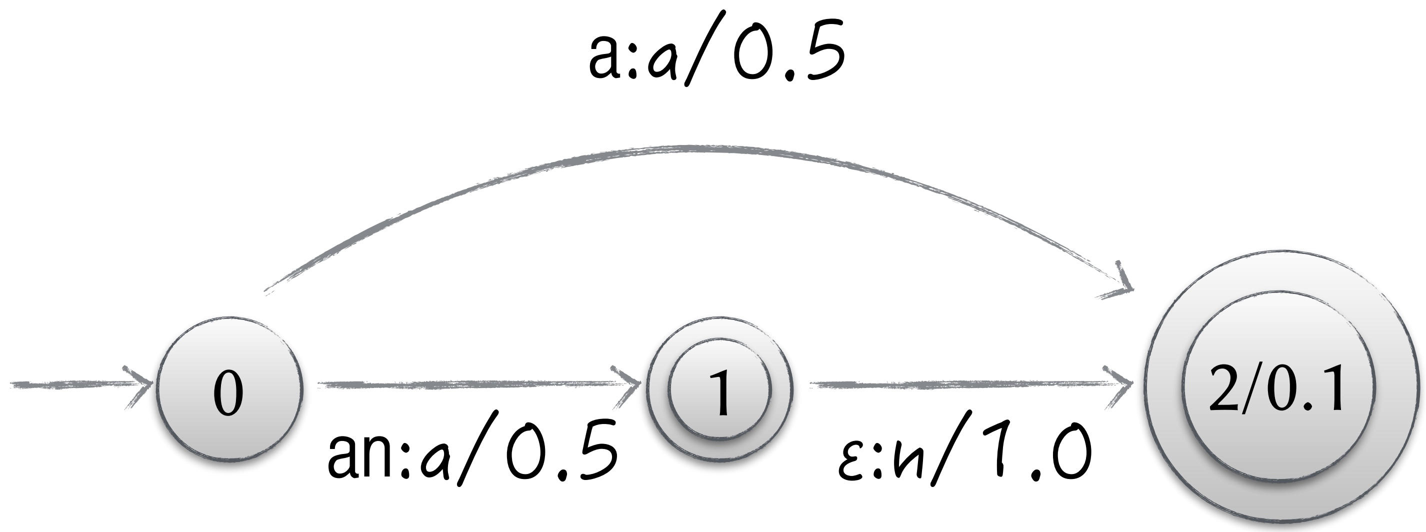
<eps>	0
an	1
a	2

Input alphabet (in.txt)

<eps>	0
a	1
n	2

Output alphabet (out.txt)

Quick Intro to OpenFst (www.openfst.org)



0	1	an	a	0.5
1	2	<eps>	n	1.0
0	2	a	a	0.5
1				
2	0.1			

Compiling & Printing FSTs

The text FSTs need to be "compiled" into binary objects before further use with OpenFst utilities

- Command used to compile:

```
fstcompile --isymbols=in.txt --osymbols=out.txt A.txt
A.fst
```

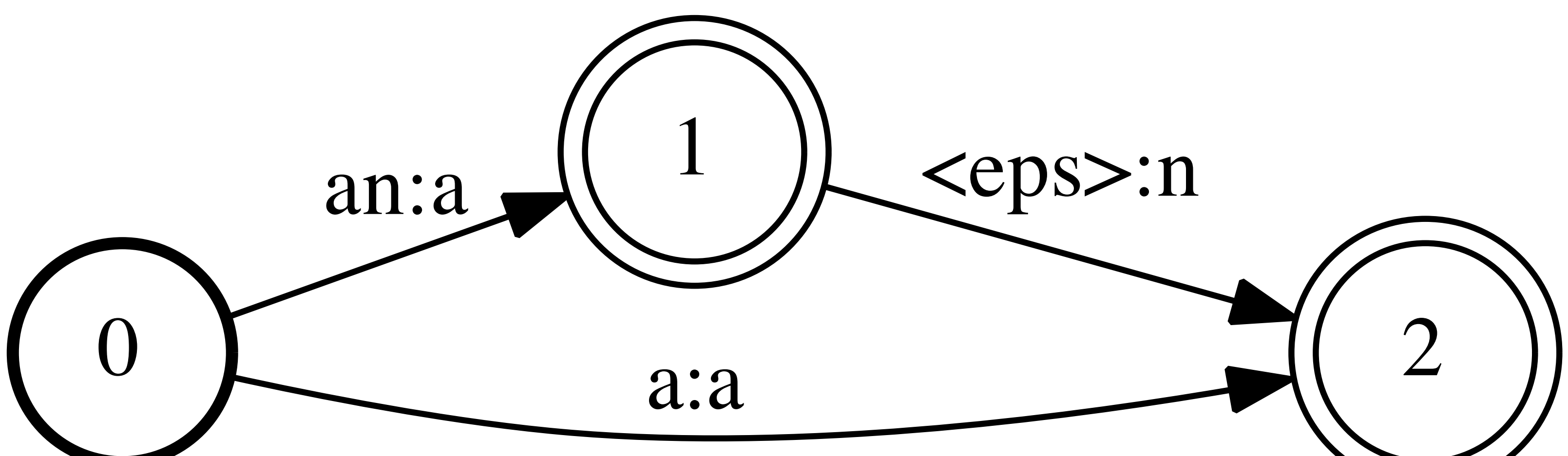
- Get back the text FST using a print command with the binary file:

```
fstprint --isymbols=in.txt --osymbols=out.txt A.fst A.txt
```

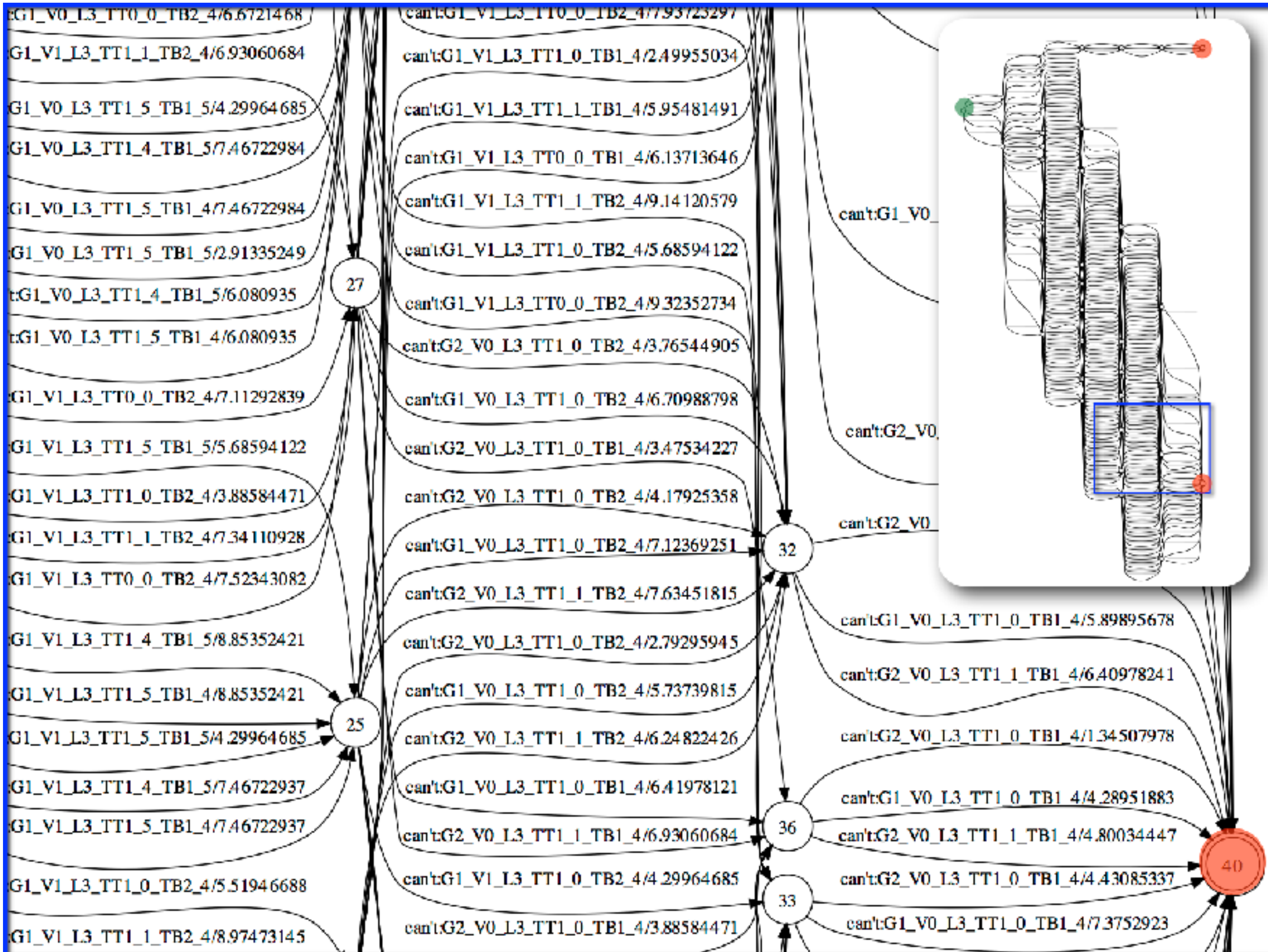
Drawing FSTs

Small FSTs can be visualized easily using the draw tool:

```
fstdraw --isymbols=in.txt --osymbols=out.txt A.fst
| dot -Tpdf > A.pdf
```



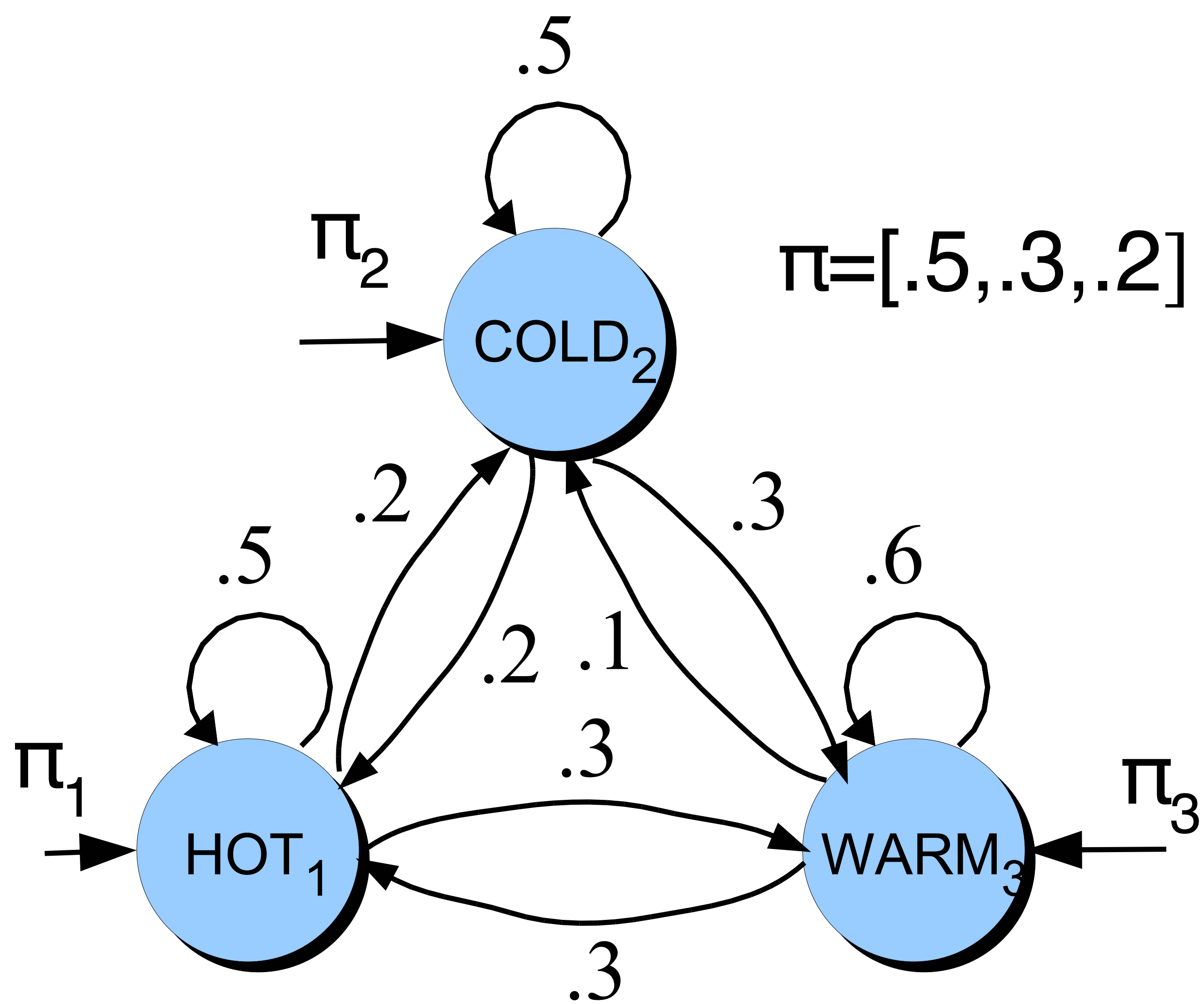
Fairly large FST!



Hidden Markov Models (HMMs)

Following slides contain figures/material from "Hidden Markov Models", Chapter 9, "Speech and Language Processing", D. Jurafsky and J. H. Martin, 2016. (https://web.stanford.edu/~jurafsky/slp3/9.pdf)

Markov Chains



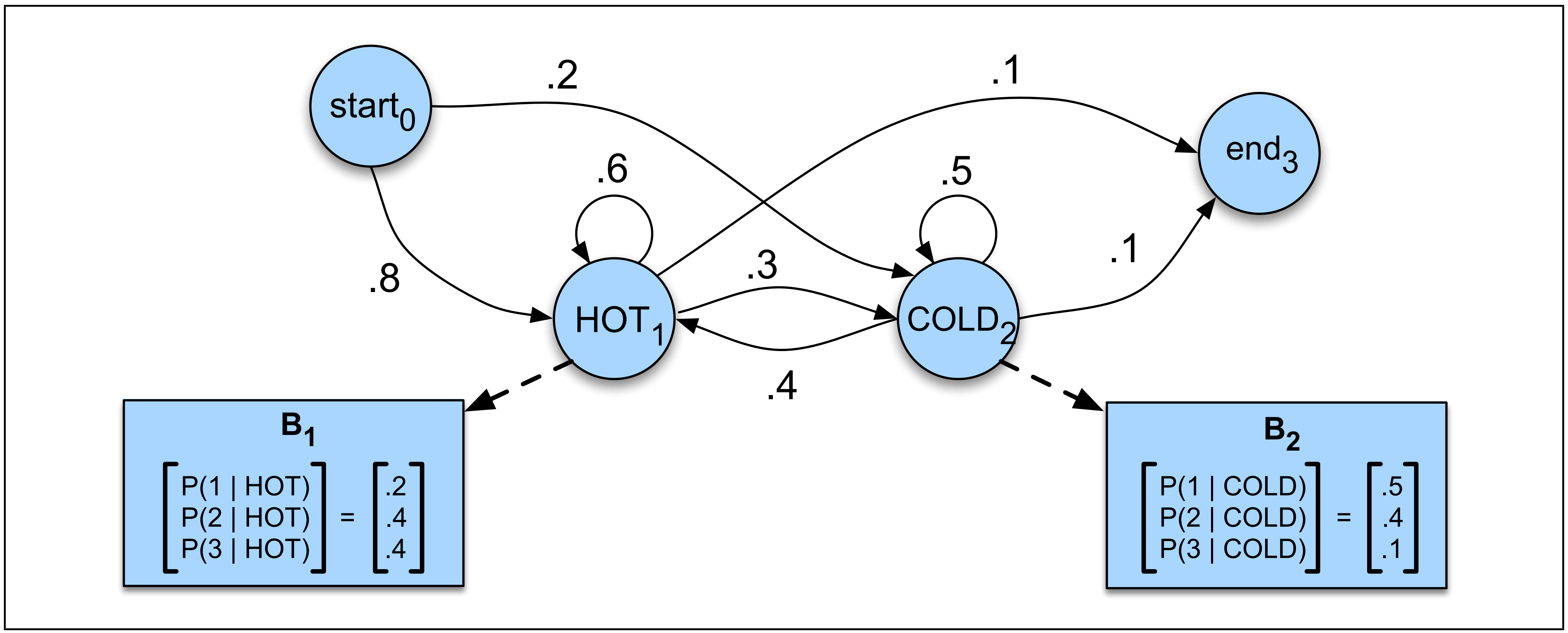
$Q = q_1 q_2 \ldots q_N$	a set of N **states**
$A = a_{01} a_{02} \ldots a_{n1} \ldots a_{nn}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{n} a_{ij} = 1 \quad \forall i$
q_0, q_F	a special **start state** and **end (final) state** that are not associated with observations

$\pi = \pi_1, \pi_2, ..., \pi_N$	an **initial probability distribution** over states. π_i is the probability that the Markov chain will start in state i. Some states j may have $\pi_j = 0$, meaning that they cannot be initial states. Also, $\sum_{i=1}^{n} \pi_i = 1$
$QA = \{q_x, q_y ...\}$	a set $QA \subset Q$ of legal **accepting states**

Hidden Markov Model

$Q = q_1 q_2 \ldots q_N$	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

HMM Assumptions



Markov Assumption: $P(q_i|q_1...q_{i-1}) = P(q_i|q_{i-1})$

Output Independence: $P(o_i|q_1 \ldots q_i, \ldots, q_T, o_1, \ldots, o_i, \ldots, o_T) = P(o_i|q_i)$

Three problems for HMMs

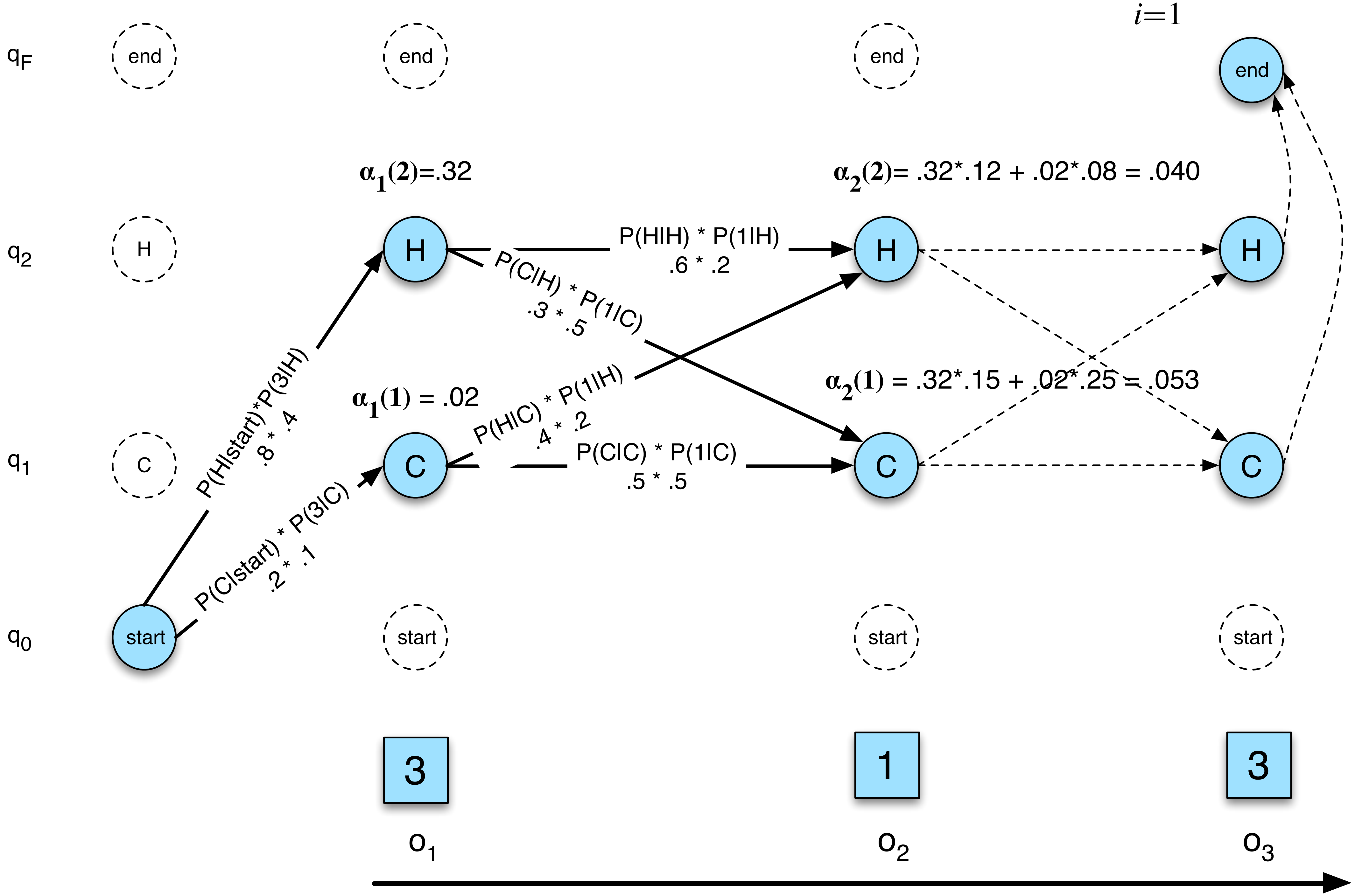
Problem 1 (Likelihood):	Given an HMM $\lambda = (A,B)$ and an observation sequence O, determine the likelihood $P(O\|\lambda)$.
Problem 2 (Decoding):	Given an observation sequence O and an HMM $\lambda = (A,B)$, discover the best hidden state sequence Q.
Problem 3 (Learning):	Given an observation sequence O and the set of states in the HMM, learn the HMM parameters A and B.

Computing Likelihood: Given an HMM $\lambda = (A,B)$ and an observation sequence O, determine the likelihood $P(O|\lambda)$.

Forward Trellis

$$\alpha_t(j) = P(o_1, o_2 \ldots o_t, q_t = j|\lambda)$$

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



Forward Algorithm

1. Initialization:

$$\alpha_1(j) \;=\; a_{0j}b_j(o_1) \;\; 1 \leq j \leq N$$

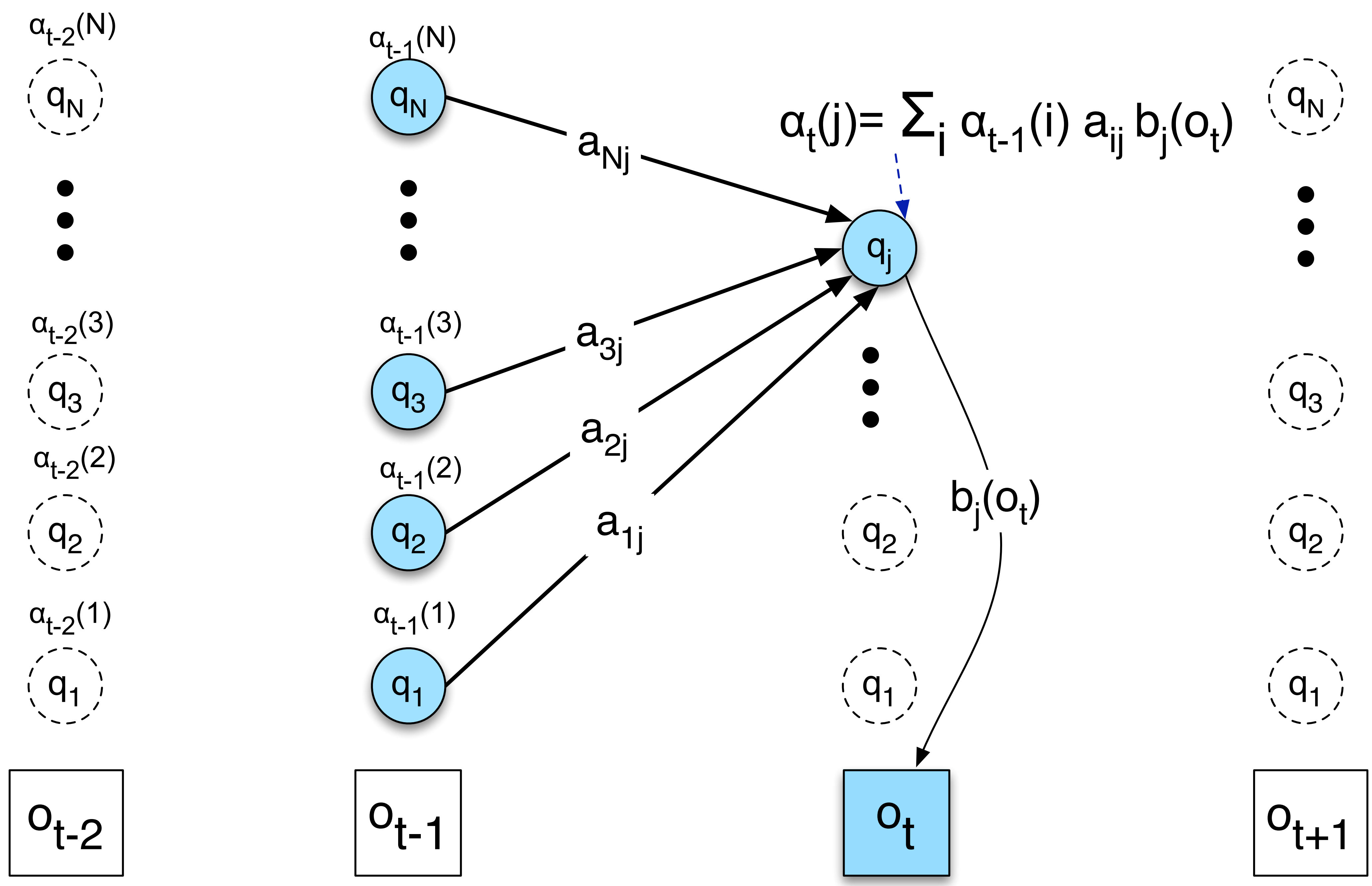
2. Recursion (since states 0 and F are non-emitting):

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

3. Termination:

$$P(O|\lambda) = \alpha_T(q_F) = \sum_{i=1}^{N} \alpha_T(i)\, a_{iF}$$

Visualizing the forward recursion



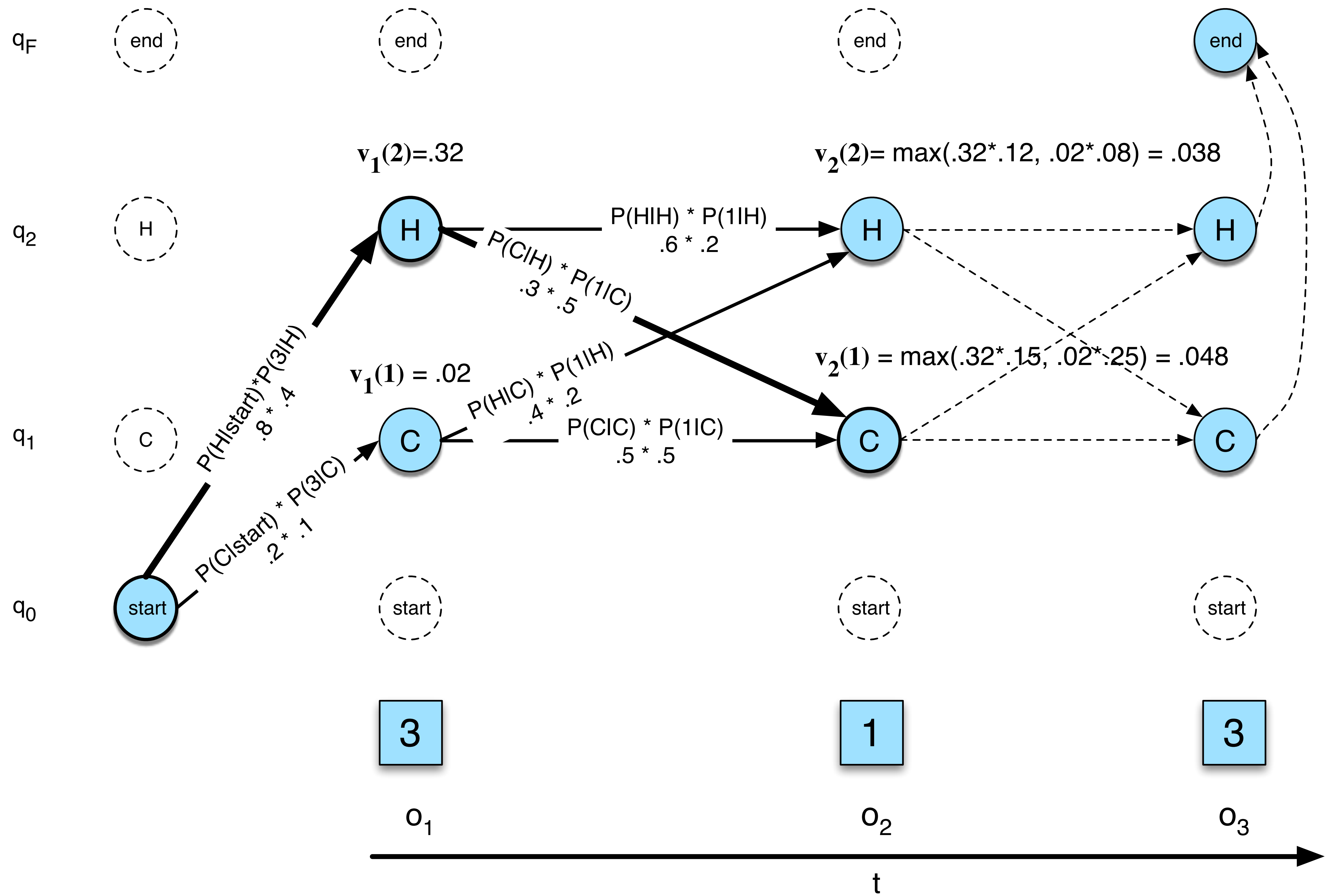
Three problems for HMMs

Problem 1 (Likelihood):	Given an HMM $\lambda = (A, B)$ and an observation sequence O, determine the likelihood $P(O\|\lambda)$.
Problem 2 (Decoding):	Given an observation sequence O and an HMM $\lambda = (A, B)$, discover the best hidden state sequence Q.
Problem 3 (Learning):	Given an observation sequence O and the set of states in the HMM, learn the HMM parameters A and B.

Decoding: Given as input an HMM $\lambda = (A, B)$ and a sequence of observations $O = o_1, o_2, ..., o_T$, find the most probable sequence of states $Q = q_1 q_2 q_3 \ldots q_T$.

Viterbi Trellis

$$v_t(j) = \max_{q_0,q_1,\ldots,q_{t-1}} P(q_0,q_1\ldots q_{t-1},o_1,o_2\ldots o_t,q_t=j|\lambda) \qquad v_t(j) = \max_{i=1}^{N} v_{t-1}(i)\, a_{ij}\, b_j(o_t)$$



Viterbi recursion

1. **Initialization:**

$$\begin{aligned} v_1(j) &= a_{0j} b_j(o_1) \;\; 1 \leq j \leq N \\ bt_1(j) &= 0 \end{aligned}$$

2. **Recursion** (recall that states 0 and q_F are non-emitting):

$$\begin{aligned} v_t(j) &= \max_{i=1}^{N} v_{t-1}(i)\, a_{ij}\, b_j(o_t); \quad 1 \leq j \leq N, 1 < t \leq T \\ bt_t(j) &= \operatorname*{argmax}_{i=1}^{N} v_{t-1}(i)\, a_{ij}\, b_j(o_t); \quad 1 \leq j \leq N, 1 < t \leq T \end{aligned}$$

3. **Termination:**

$$\begin{aligned} \text{The best score:} \quad P* = v_T(q_F) &= \max_{i=1}^{N} \; v_T(i) * a_{iF} \\ \text{The start of backtrace:} \quad q_T* = bt_T(q_F) &= \operatorname*{argmax}_{i=1}^{N} \; v_T(i) * a_{iF} \end{aligned}$$

Viterbi backtrace