

POS Tagging

Background

- **Part of speech:**
 - Noun, verb, pronoun, preposition, adverb, conjunction, particle, and article
- Recent lists of **POS** (also know as **word classes**, **morphological class**, or **lexical tags**) have much larger numbers of word classes.
 - 45 for Penn Treebank
 - 87 for the Brown corpus, and
 - 146 for the C7 tagset
- The significance of the POS for language processing is that it gives a significant amount of information about the word and its neighbors.
- POS can be used in stemming for IR, since
 - Knowing a word's POS can help tell us which morphological affixes it can take.
 - They can help an IR application by helping select out nouns or other important words from a document.

English Word Classes

- Give a more complete definition of the classes of POS.
 - Traditionally, the definition of POS has been based on morphological and syntactic function.
 - While, it has tendencies toward semantic coherence (e.g., nouns describe people, places, or things and adjectives describe properties), this is not necessarily the case.
- Two broad subcategories of POS:
 1. **Closed class**
 2. **Open class**

English Word Classes

1. Closed class

- Having relatively fixed membership, e.g., prepositions
- **Function words:**
 - Grammatical words like *of*, *and*, or *you*, which tend to be very short, occur frequently, and play an important role in grammar.

2. Open class

- Four major open classes occurring in the languages of the world: **nouns, verbs, adjectives, and adverbs.**
- Many languages have no adjectives, e.g., the native American language Lakota, and Chinese

Tagsets for English

- There are a small number of popular tagsets for English, many of which evolved from the 87-tag tagset used for the Brown corpus.
 - Three commonly used
 - **The small 45-tag Penn Treebank tagset**
 - The medium-sized 61 tag C5 tageset used by the Lancaster UCREL project's CLAWS tagger to tag the British National Corpus, and
 - The larger 146-tag C7 tagset

Tag	Description	Example	Tag	Description	Example
CC	Coordin. Conjunction	<i>and, but, or</i>	SYM	Symbol	+%, &
CD	Cardinal number	<i>one, two, three</i>	TO	"to"	<i>to</i>
DT	Determiner	<i>a, the</i>	UH	Interjection	<i>uh, oops</i>
EX	Existential 'there'	<i>there</i>	VB	Verb, base form	<i>eat</i>
FW	Foreign word	<i>meu culpa</i>	VBD	Verb, past tense	<i>ate</i>
IN	Preposition/sub-conj	<i>of, in, by</i>	VBG	Verb, gerund	<i>eating</i>
JJ	Adjective	<i>yellow</i>	VBN	Verb, past participle	<i>aten</i>
JJR	Adj., comparative	<i>bigger</i>	VBP	Verb, non-3sg pres	<i>eat</i>
JJS	Adj., superlative	<i>wildest</i>	VBZ	Verb, 3sg pres	<i>eats</i>
LS	List item marker	<i>I, 2, One</i>	WDT	Wh-determiner	<i>which, that</i>
MD	Modal	<i>can, should</i>	WP	Wh-pronoun	<i>what, who</i>
NN	Noun, sing. or mass	<i>llama</i>	WP\$	Possessive wh-	<i>whose</i>
NNPS	Noun, plural	<i>llamas</i>	WRB	Wh-adverb	<i>how, where</i>
NNP	Proper noun, singular	<i>IBM</i>	\$	Dollar sign	\$
NNPS	Proper noun, plural	<i>Carolinas</i>	#	Pound sign	#
PDT	Predeterminer	<i>all, both</i>	"	Left quote	(“ or “)
POS	Possessive ending	's	"	Right quote	(' or ')
PP	Personal pronoun	<i>I, you, he</i>	(Left parenthesis	([, (, {, <)
PP\$	Possessive pronoun	<i>your, one's</i>)	Right parenthesis	(],), }, >)
RB	Adverb	<i>quickly, never</i>	,	Comma	,
RBR	Adverb, comparative	<i>faster</i>	.	Sentence-final punc	(. ! ?)
RBS	Adverb, superlative	<i>fastest</i>	:	Mid-sentence punc	(: ; ... - -)
RP	Particle	<i>up, off</i>			

Penn Treebank POS tags

The grand jury commented on a number of other topics.

Tags for English

The grand jury commented on a number of other topics.

The/DT grand/JJ jury/NN commented/VBD on/IN a /DT number/NN of/IN other/JJ topics/NNS ./.

There are 70 children there

There/EX are/VBP 70/CD children/NNS there/RB

Although preliminary findings were reported more than a year ago, the latest results appear in today's New English Journal of Medicine

Although/IN preliminary/JJ findings/NNS were/VBD reported/VBN more/RBR than/IN a/DT year/NN ago/NN ./, the/DT latest/JJS results/NNS appear/VBP in/IN today/NN 's/POS New/NNP English/NNP Journal/NNP of/IN Medicine/NNP

Tag	Description	Example	Tag	Description	Example
CC	Coordin. Conjunction <i>and, but, or</i>		SYM	Symbol	+%, &
CD	Cardinal number	<i>one, two, three</i>	TO	"to"	<i>to</i>
DT	Determiner	<i>a, the</i>	UH	Interjection	<i>uh, oops</i>
EX	Existential 'there'	<i>there</i>	VB	Verb, base form	<i>eat</i>
FW	Foreign word	<i>meu culpa</i>	VBD	Verb, past tense	<i>ate</i>
IN	Preposition/sub-conj	<i>of, in, by</i>	VBG	Verb, gerund	<i>eating</i>
JJ	Adjective	<i>yellow</i>	VBN	Verb, past participle	<i>eaten</i>
JJR	Adj., comparative	<i>bigger</i>	VBP	Verb, non-3sg pres	<i>eat</i>
JJS	Adj., superlative	<i>wildest</i>	VBZ	Verb, 3sg pres	<i>eats</i>
LS	List item marker	<i>I, 2, One</i>	WDT	Wh-determiner	<i>which, that</i>
MD	Modal	<i>can, should</i>	WP	Wh-pronoun	<i>what, who</i>
NN	Noun, sing. or mass	<i>llama</i>	WP\$	Possessive wh-	<i>whose</i>
NNS	Noun, plural	<i>llamas</i>	WRB	Wh-adverb	<i>how, where</i>
NNP	Proper noun, singular	<i>IBM</i>	\$	Dollar sign	\$
NNPS	Proper noun, plural	<i>Carolinas</i>	#	Pound sign	#
PDT	Predeterminer	<i>all, both</i>	"	Left quote	(‘ or “)
POS	Possessive ending	<i>'s</i>	"	Right quote	(‘ or ”)
PP	Personal pronoun	<i>I, you, he</i>	(Left parenthesis	([, {, <)
PP\$	Possessive pronoun	<i>your, one's</i>)	Right parenthesis	(], }, >)
RB	Adverb	<i>quickly, never</i>	,	Comma	,
RBR	Adverb, comparative	<i>faster</i>	.	Sentence-final punc	(. ! ?)
RBS	Adverb, superlative	<i>fastest</i>	:	Mid-sentence punc	(: ; ... --)
RP	Particle	<i>up, off</i>			

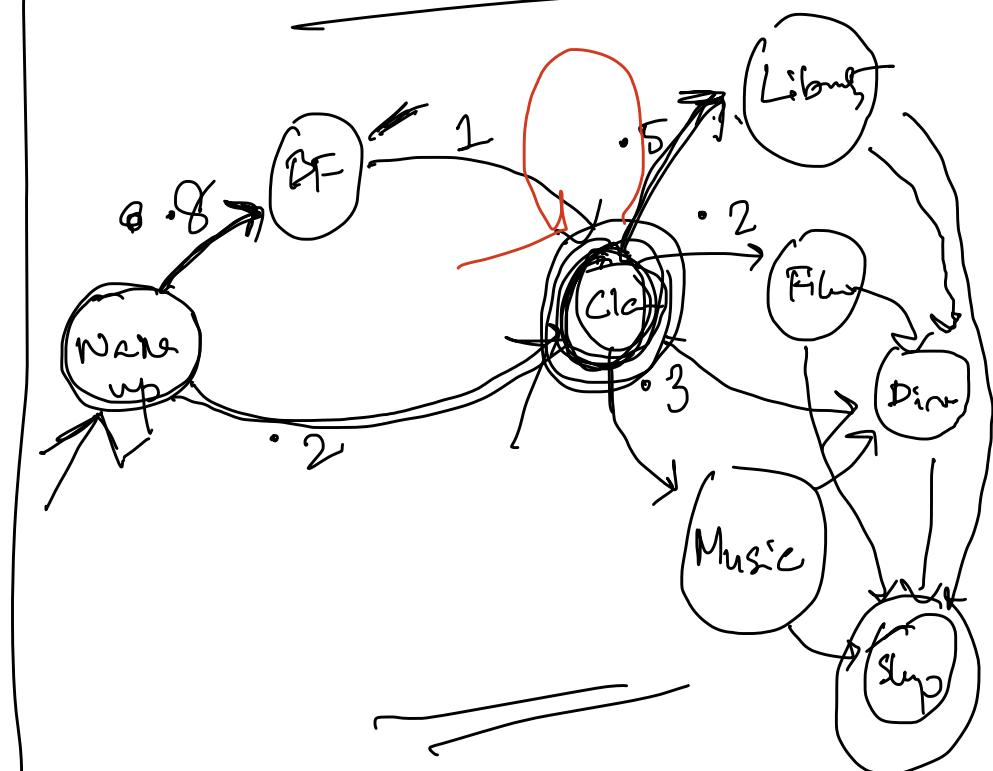


Hidden Markov Models

Outline

- **Markov Chains**
- **Hidden Markov Models**
- **Three Algorithms for HMMs**
 - The Forward Algorithm
 - The Viterbi Algorithm
- **Applications:**
 - The Ice Cream Task
 - Part of Speech Tagging

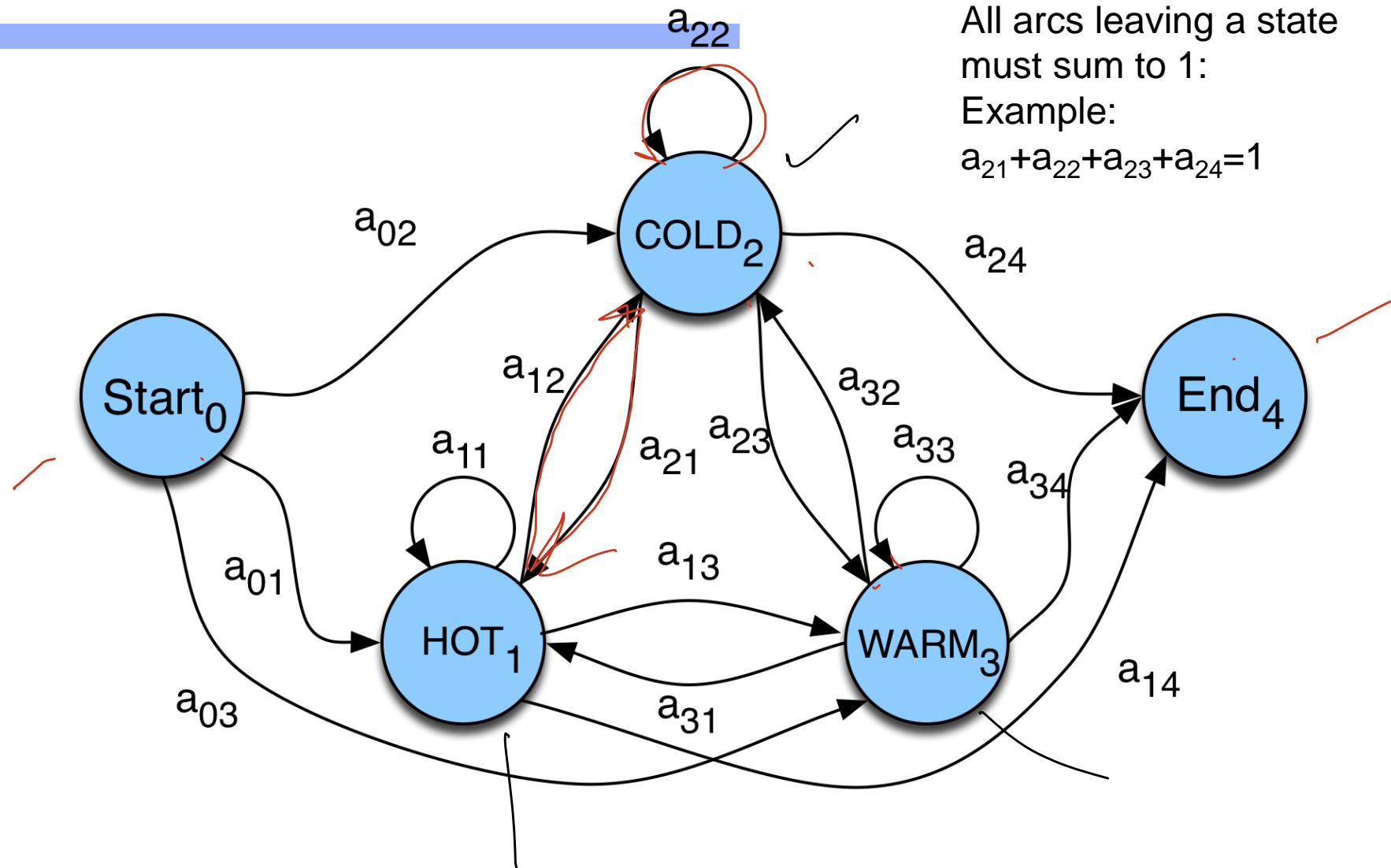
Finite State Mach



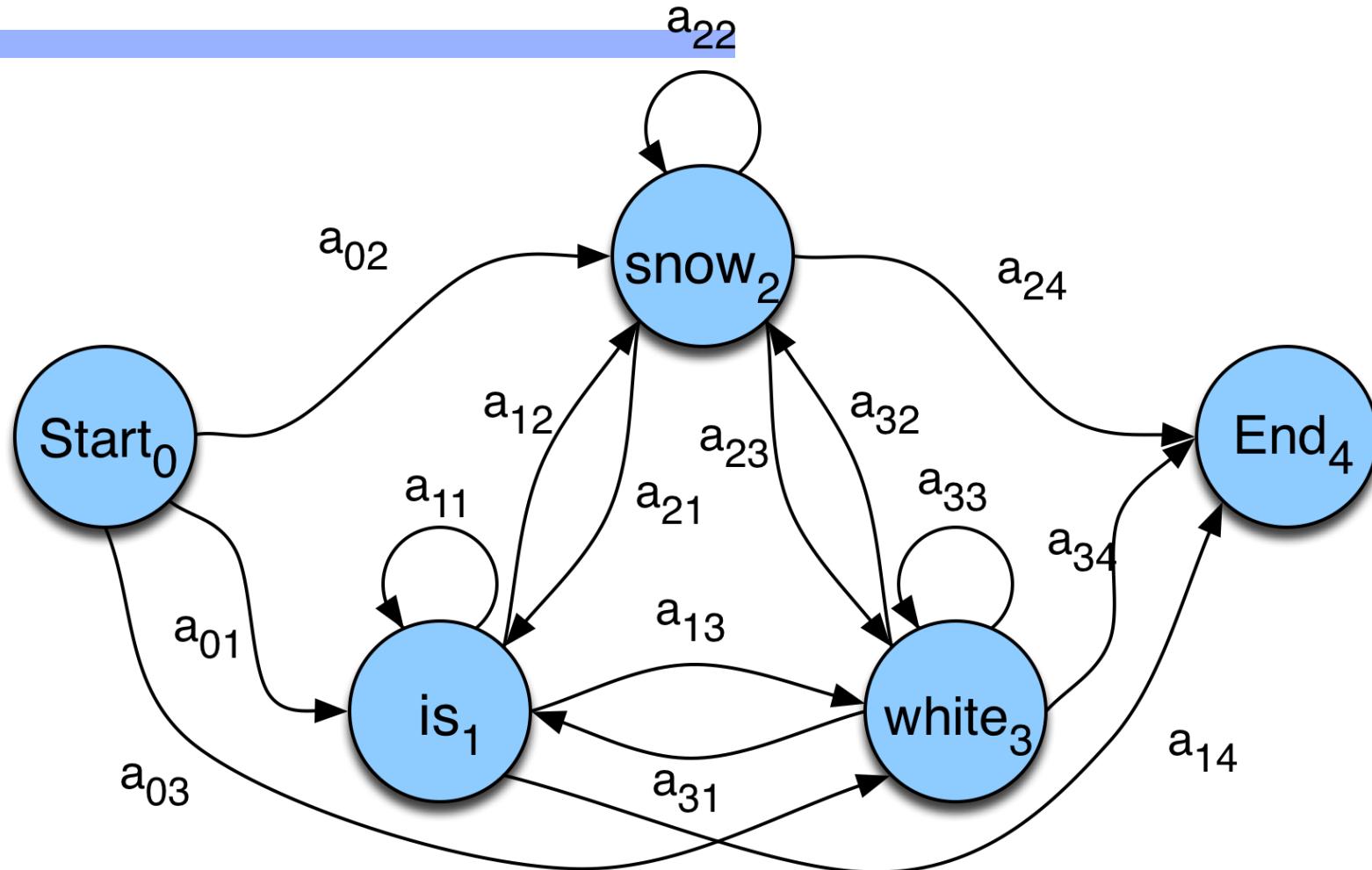
Definitions

- A **weighted finite-state automaton**
 - An FSA with probabilities on the arcs
 - The sum of the probabilities leaving any arc must sum to one
- A **Markov chain (or observable Markov Model)**
 - a special case of a WFST in which the input sequence uniquely determines which states the automaton will go through
- **Markov chains can't represent inherently ambiguous problems**
 - Useful for assigning probabilities to unambiguous sequences

Markov chain for weather



Markov chain for words



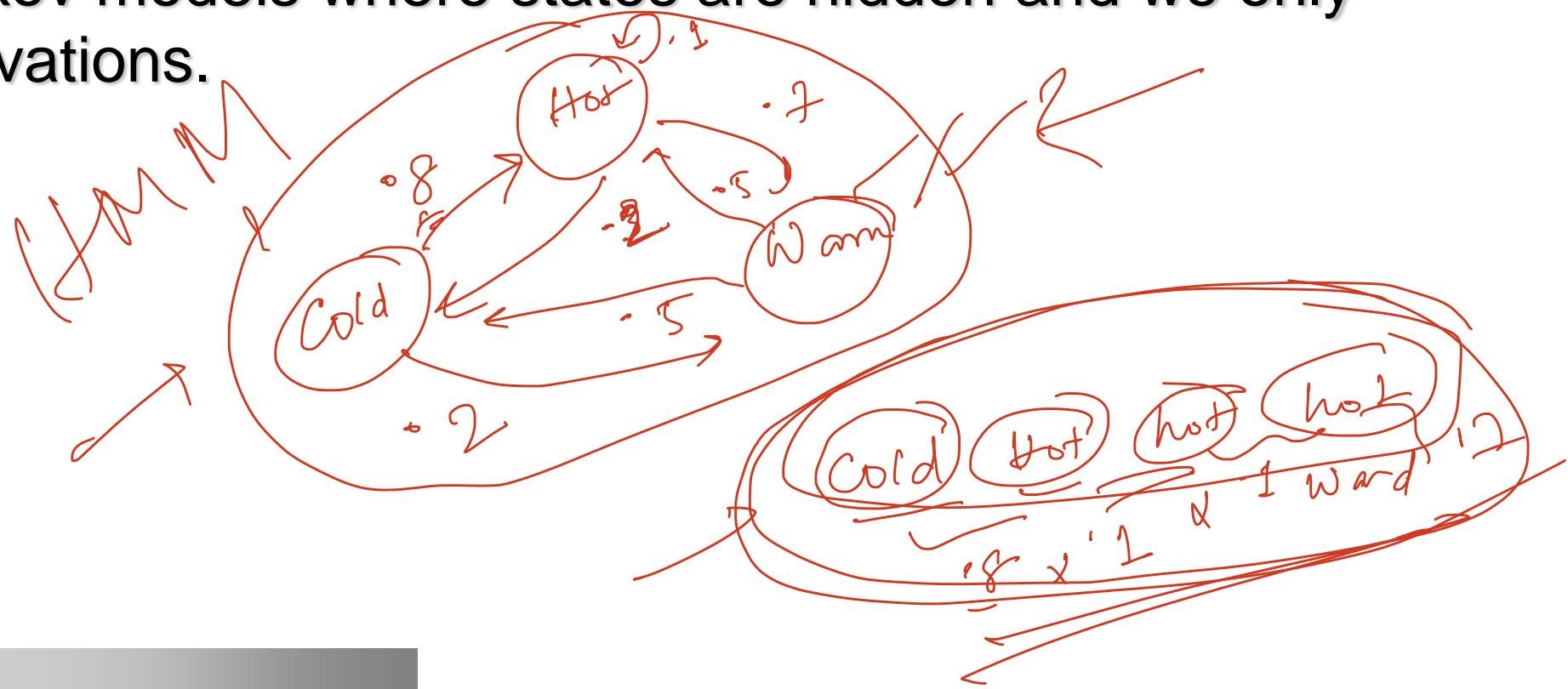
Markov chain = “First-order observable Markov Model”

- **Markov Assumption:**

Current state only depends on previous state

$$P(q_i | q_1 \dots q_{i-1}) = P(q_i | q_{i-1}) \quad - \text{First order MP}$$

- So far the states are visible
- To model more complex transitions we might need to use hidden markov models where states are hidden and we only make observations.



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 Delhi for summer of 2025
- But you find Rahul's diary
- Which lists how many ice-creams Rahul 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 named-entity or part-of-speech tagging (and speech recognition and other things)
 - The output symbols are **words**
 - But the hidden states are something else
 - **Part-of-speech tags**
 - **Named entity 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

$$Q = q_1 q_2 \dots q_N$$

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

Assumptions

- **Markov assumption:**

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

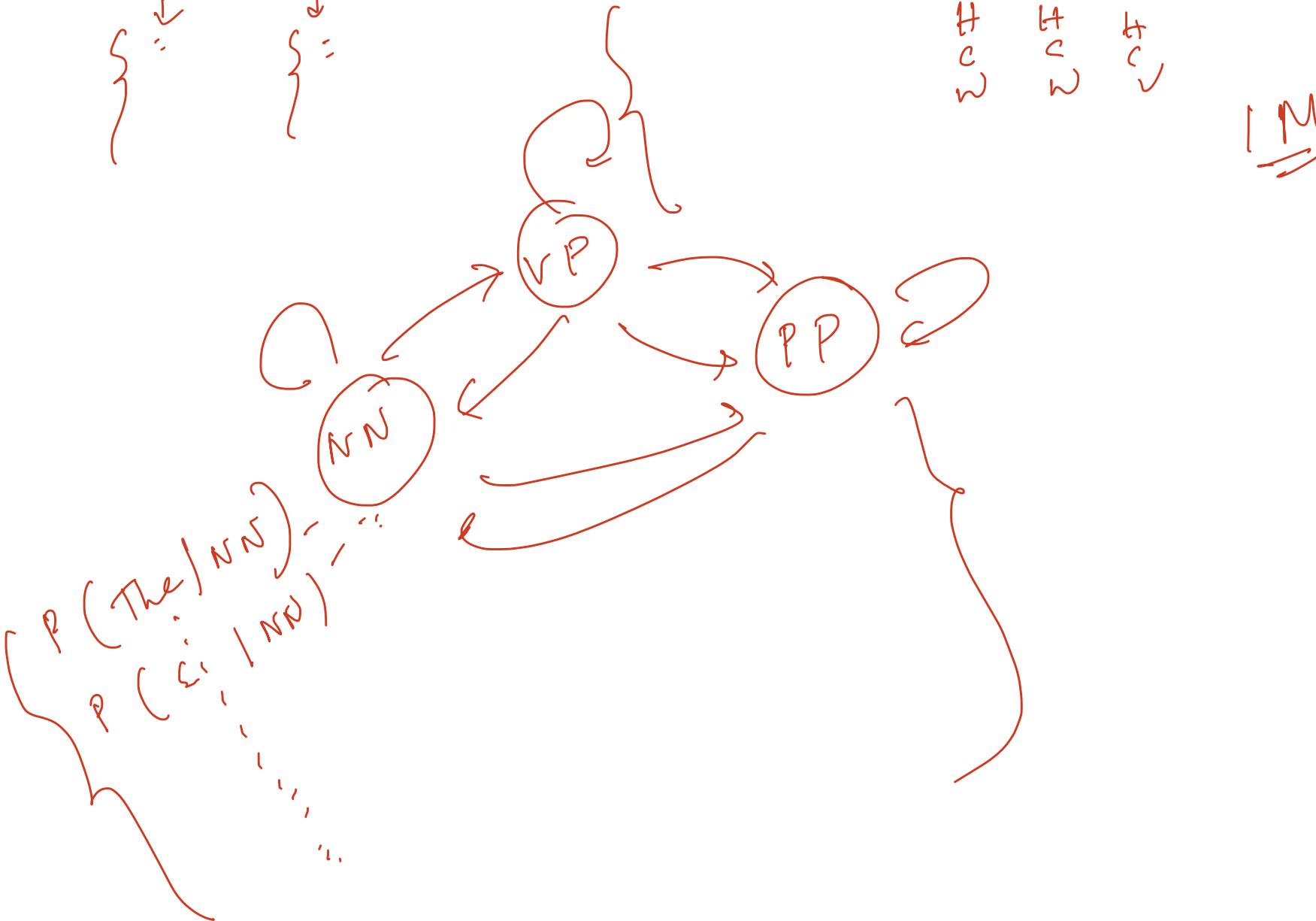
- the current state is dependent only on the previous state.
- this represents the memory of the model

- **Output-independence assumption**

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

- the output observation at time t is dependent only on the current state
- it is independent of previous observations and states

The sun is slow today.



The Ice Cream task (cont.)

- Given a sequence of observations O ,
 - each observation an integer = number of ice creams eaten
 - Figure out correct **hidden** sequence Q of weather states (H or C) which caused Rahul to eat the ice cream

In other words:

Given:

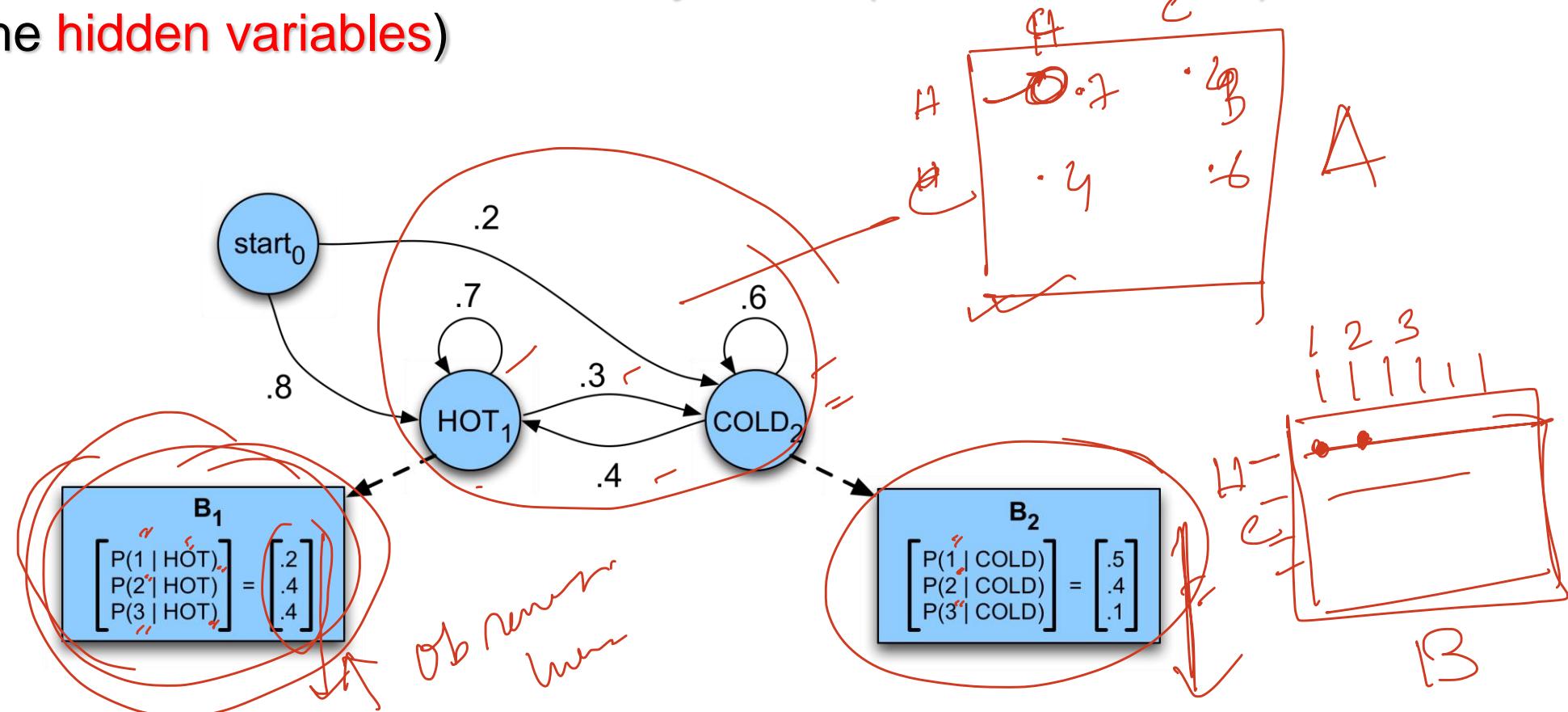
Ice Cream Observation Sequence: 1,2,3,2,2,2,3...

Produce:

Weather Sequence: H,C,H,H,H,C...

An HMM for this task

- Relating numbers of ice creams eaten by Rahul (the **observations**) to the weather (the **hidden variables**)



The Three Basic Problems for HMMs : more formally

- Problem 1 (**Evaluation**): Given the observation sequence $O = (o_1 o_2 \dots o_T)$, and an HMM model $\Phi = (A, B)$, how do we efficiently compute $P(O | \Phi)$, the probability of the observation sequence, given the model

$$P(O | \Phi)$$

$$A = \begin{bmatrix} & & \\ & \text{trans } p_{ij} & \\ & & \end{bmatrix}$$

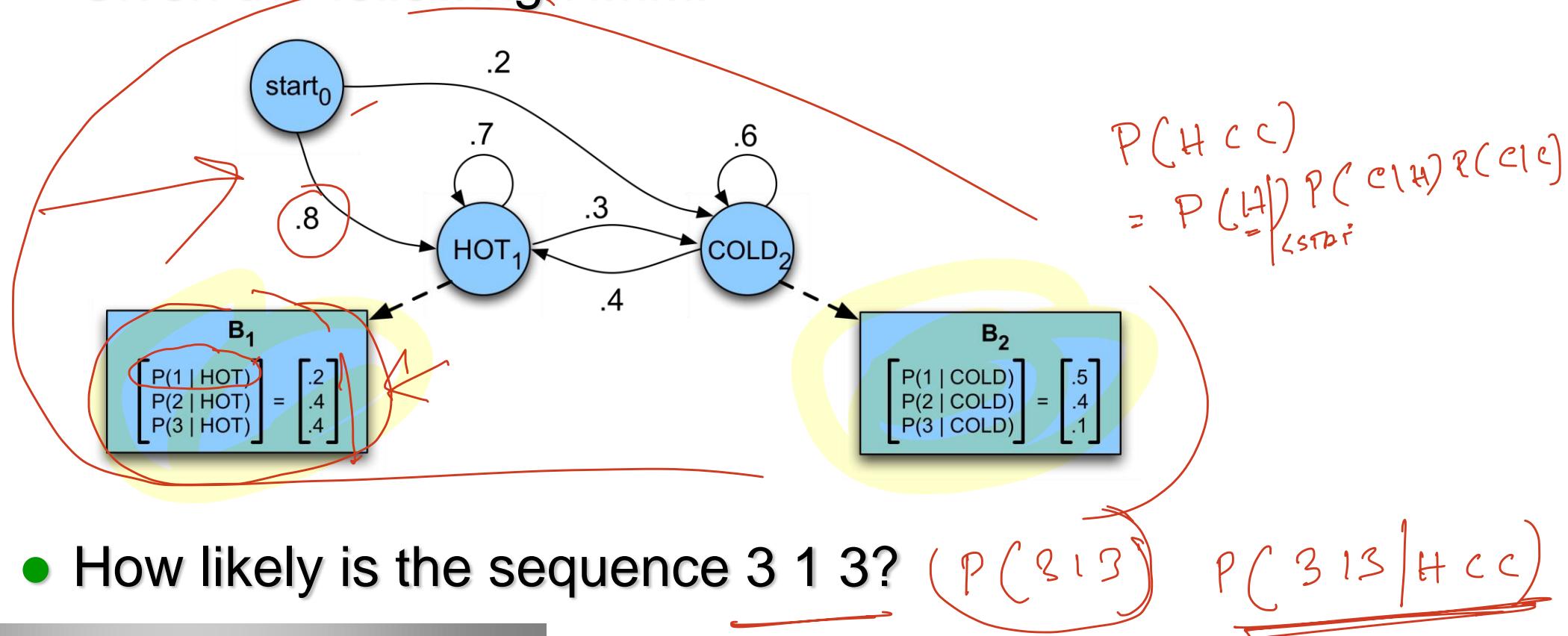
$$B = \begin{bmatrix} o_{11} & o_{12} & \dots \\ o_{21} & o_{22} & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

3x3

Problem 1: computing the observation likelihood

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

- Given the following HMM:



A simple line drawing of three red flip-flops arranged in a cluster. Each flip-flop has a small smiley face on its toe strap.

② 13
H H C

$$\begin{array}{c}
 \text{P} \left(\frac{1}{e} \right) + P \left(\frac{3}{e} \right) \\
 + P \left(\frac{4}{e} \text{ STAT} \right) + \dots \\
 \hline
 \end{array}$$

A hand-drawn diagram of a DNA double helix. The diagram shows two red lines forming the backbone of the helix, with a central ladder-like structure representing the nitrogenous bases. Several labels are written in red ink:

- At the top left, above the top strand, is $P(CCA)$.
- At the top right, above the bottom strand, is $C(C)P$.
- On the left side, below the top strand, is $+P$.
- On the right side, below the bottom strand, is $-CC- +H(D)$.
- In the center, between the strands, is $(3,3, HCC)$.
- On the left side, below the bottom strand, is $P(3,3, HCC)$.
- At the bottom left, below the bottom strand, is max .
- At the bottom right, below the bottom strand, is $P(CCA)$.

$$\begin{aligned}
 & P(3|3|HCC) + P(1+CC) \\
 & = P(3|H)P(1|H)P(3|e) + P(H|s) \\
 & \quad P(C1H|HCC)
 \end{aligned}$$

$$P(C(3^{15})) = \sum_{\text{any } k} P(C(3^{15}))$$

$$\begin{aligned} &= \sum_{\alpha \in \mathcal{A}} P(\mathcal{B}^{(1)} | \alpha) \\ &= P(\mathcal{B}^{(1)}, H(1)) + P(\mathcal{B}^{(1)}, \neg H(1)) \end{aligned}$$

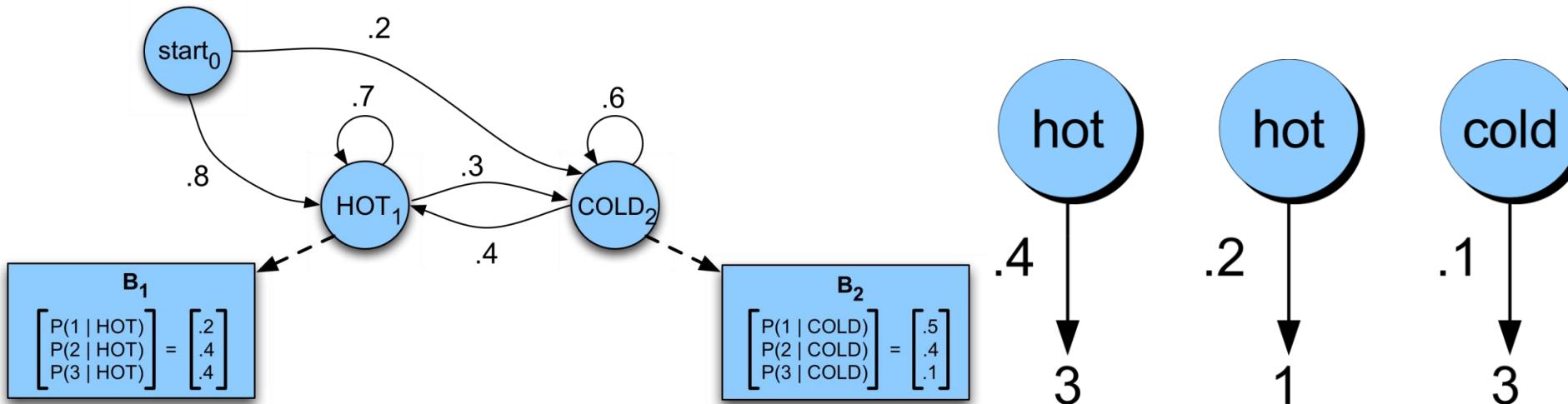
How to compute likelihood

- For a Markov chain, we just follow the states 3 1 3 and multiply the probabilities
- But for an HMM, we don't know what the states are!
- So let's start with a simpler situation.
- Computing the observation likelihood for a **given** hidden state sequence
 - Suppose we knew the weather and wanted to predict how much ice cream Rahul would eat.
 - i.e. $P(3 \ 1 \ 3 | H \ H \ C)$

Computing likelihood of 3 1 3 given hidden state sequence

$$P(O|Q) = \prod_{i=1}^T P(o_i|q_i)$$

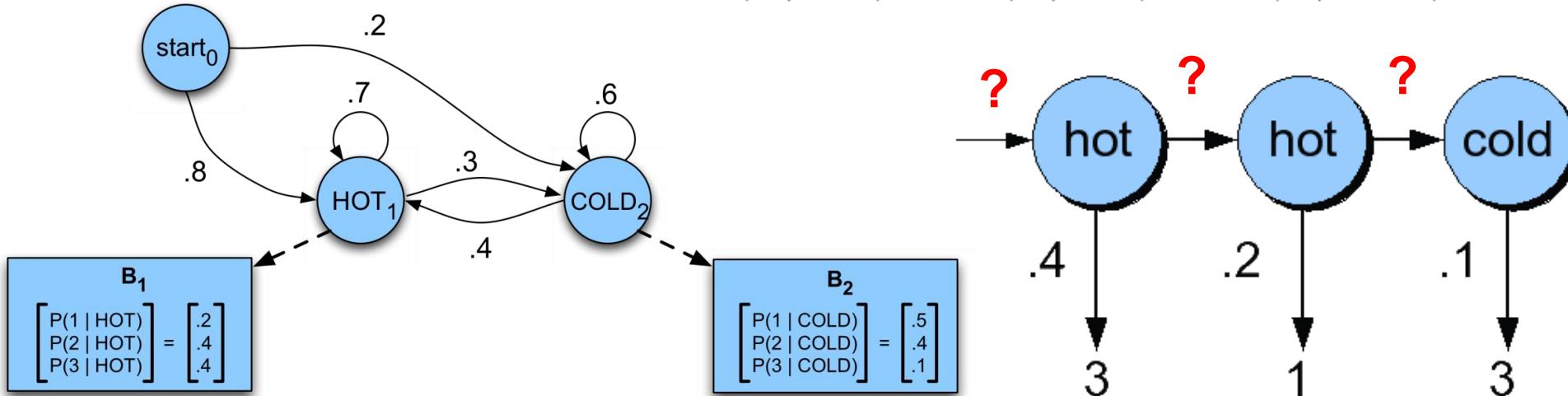
$$P(3 \ 1 \ 3 | \text{hot hot cold}) = P(3|\text{hot}) \times P(1|\text{hot}) \times P(3|\text{cold})$$



Computing joint probability of observation and state sequence

$$P(O, Q) = P(O|Q) \times P(Q) = \prod_{i=1}^n P(o_i|q_i) \times \prod_{i=1}^n P(q_i|q_{i-1})$$

$$\begin{aligned} P(3 \ 1 \ 3, \text{hot hot cold}) &= P(\text{hot|start}) \times P(\text{hot|hot}) \times P(\text{cold|hot}) \\ &\quad \times P(3|\text{hot}) \times P(1|\text{hot}) \times P(3|\text{cold}) \end{aligned}$$

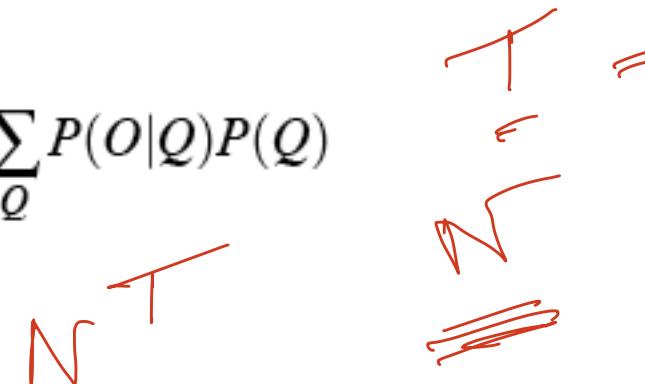


Computing total likelihood of 3 1 3

- We would need to sum over

- Hot hot cold
- Hot hot hot
- Hot cold hot
-

$$P(O) = \sum_Q P(O, Q) = \sum_Q P(O|Q)P(Q)$$



- How many possible hidden state sequences are there for this sequence?

$$P(3 1 3) = P(3 1 3, \text{cold cold cold}) + P(3 1 3, \text{cold cold hot}) + P(3 1 3, \text{hot hot cold}) + \dots$$

- How about in general for an HMM with N hidden states and a sequence of T observations?

- N^T

- So we can't just do separate computation for each hidden state sequence.

Instead: the Forward algorithm

- A kind of **dynamic programming** algorithm
 - Just like Minimum Edit Distance
 - Uses a table to store intermediate values
- Idea:
 - Compute the likelihood of the observation sequence
 - By summing over all possible hidden state sequences
 - But doing this efficiently
 - By folding all the sequences into a single **trellis**

The forward algorithm

- The goal of the forward algorithm is to compute

$$P(o_1, o_2 \dots o_T, q_T = q_F \mid \lambda)$$

- We'll do this by recursion

A =
B =

The forward algorithm

- Each cell of the forward algorithm trellis $\alpha_t(j)$
 - Represents the probability of being in state j
 - After seeing the first t observations
 - Given the automaton
- Each cell thus expresses the following probability

$$\alpha_t(j) = P(o_1, o_2 \dots o_t, q_t = j \mid \lambda)$$

The Forward Recursion

1. Initialization:

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

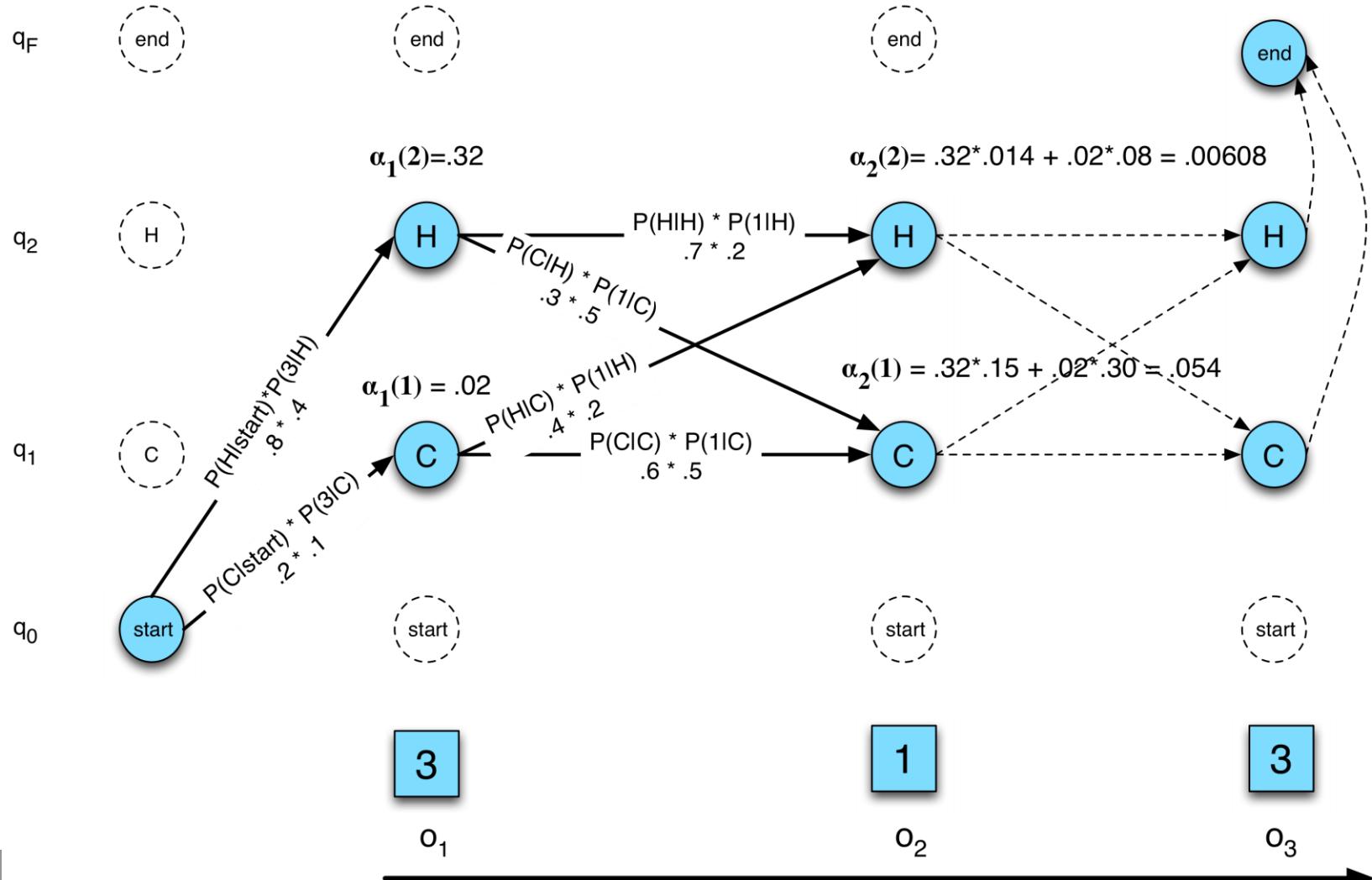
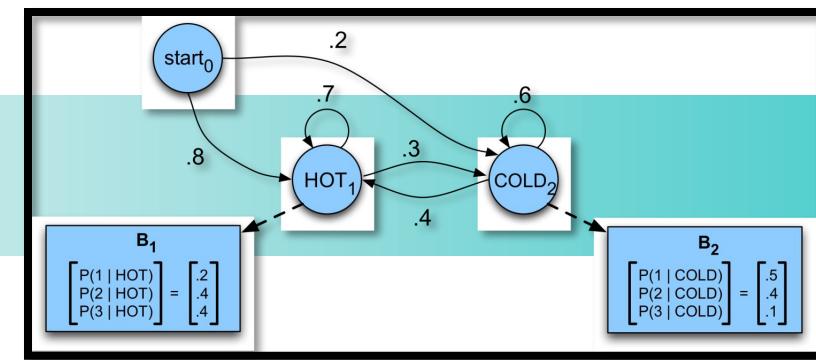
2. Recursion (since states 0 and F are non-emitting):

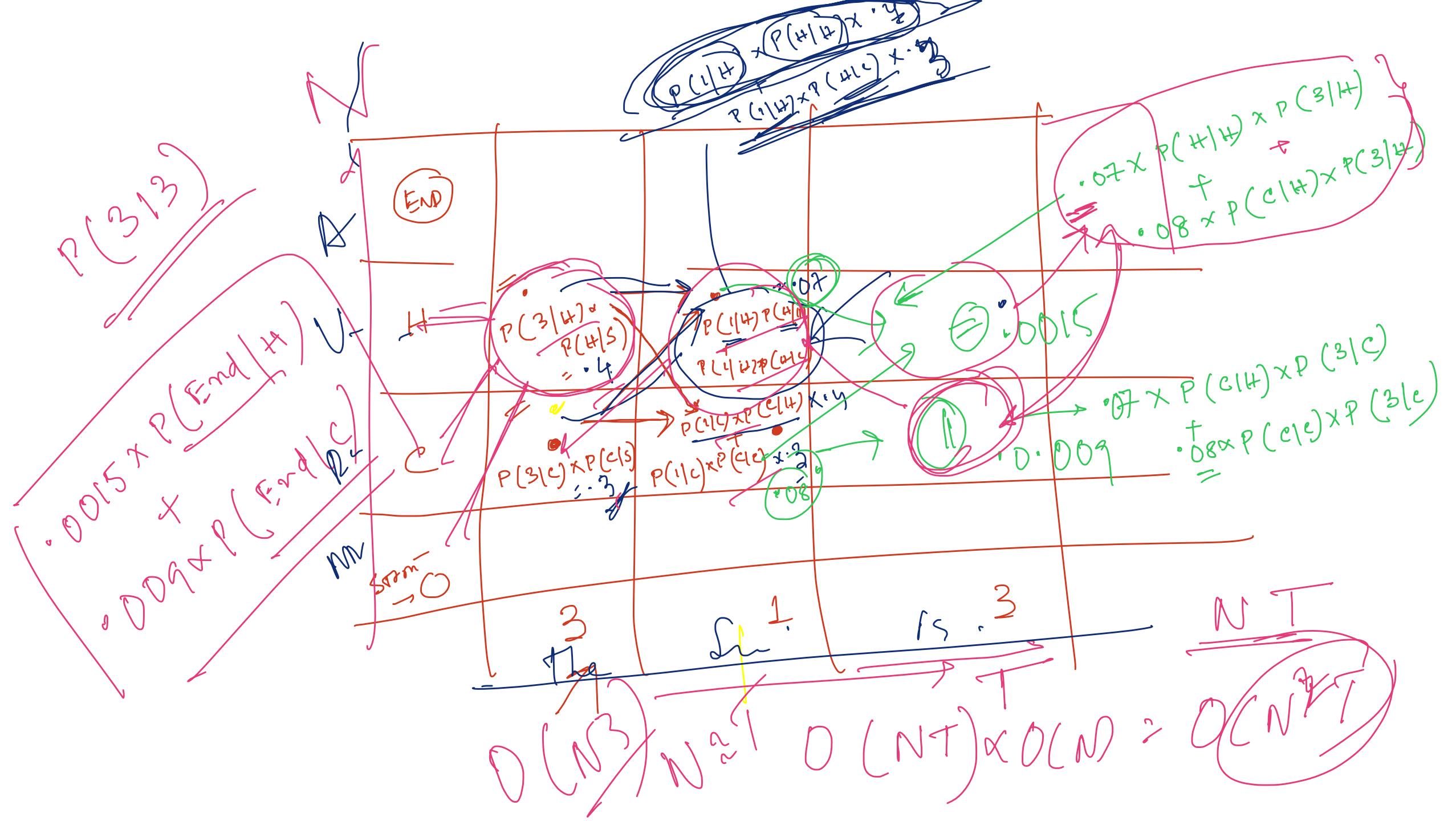
$$\alpha_t(j) = \sum_{i=1}^N \alpha_{t-1}(i) a_{ij} b_j(o_t); \quad 1 \leq j \leq N, 1 < t \leq T$$

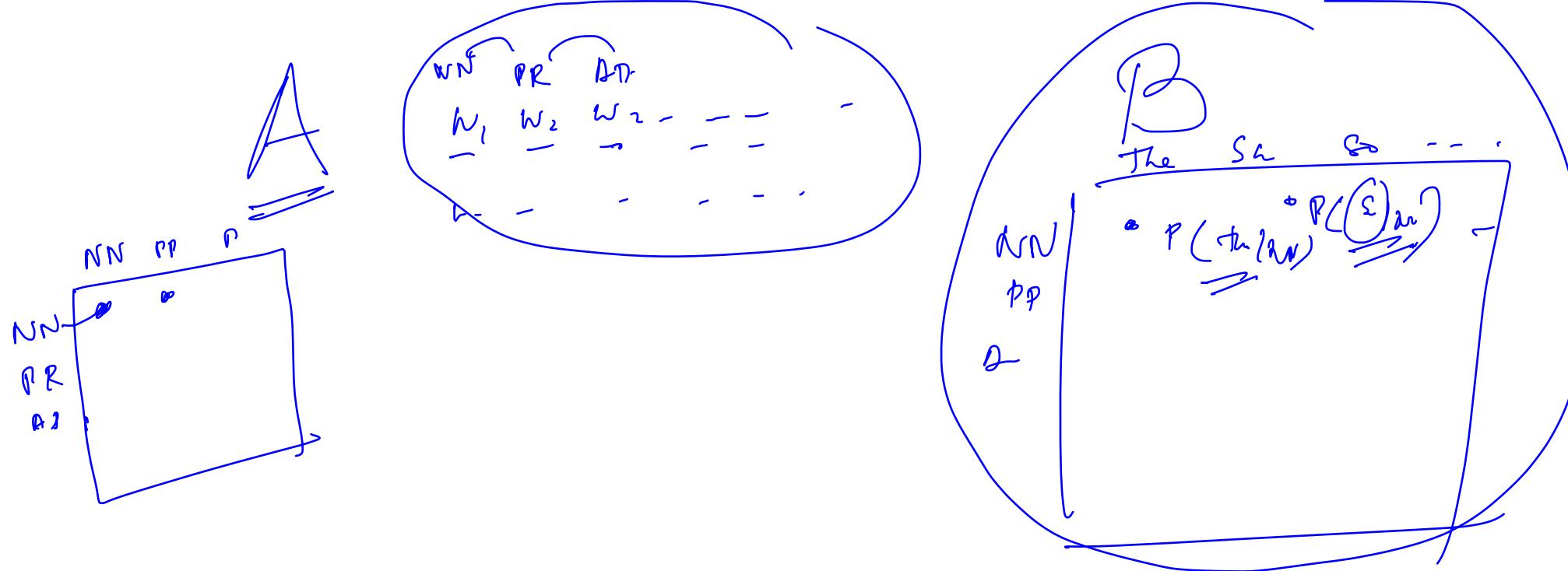
3. Termination:

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

The Forward Trellis





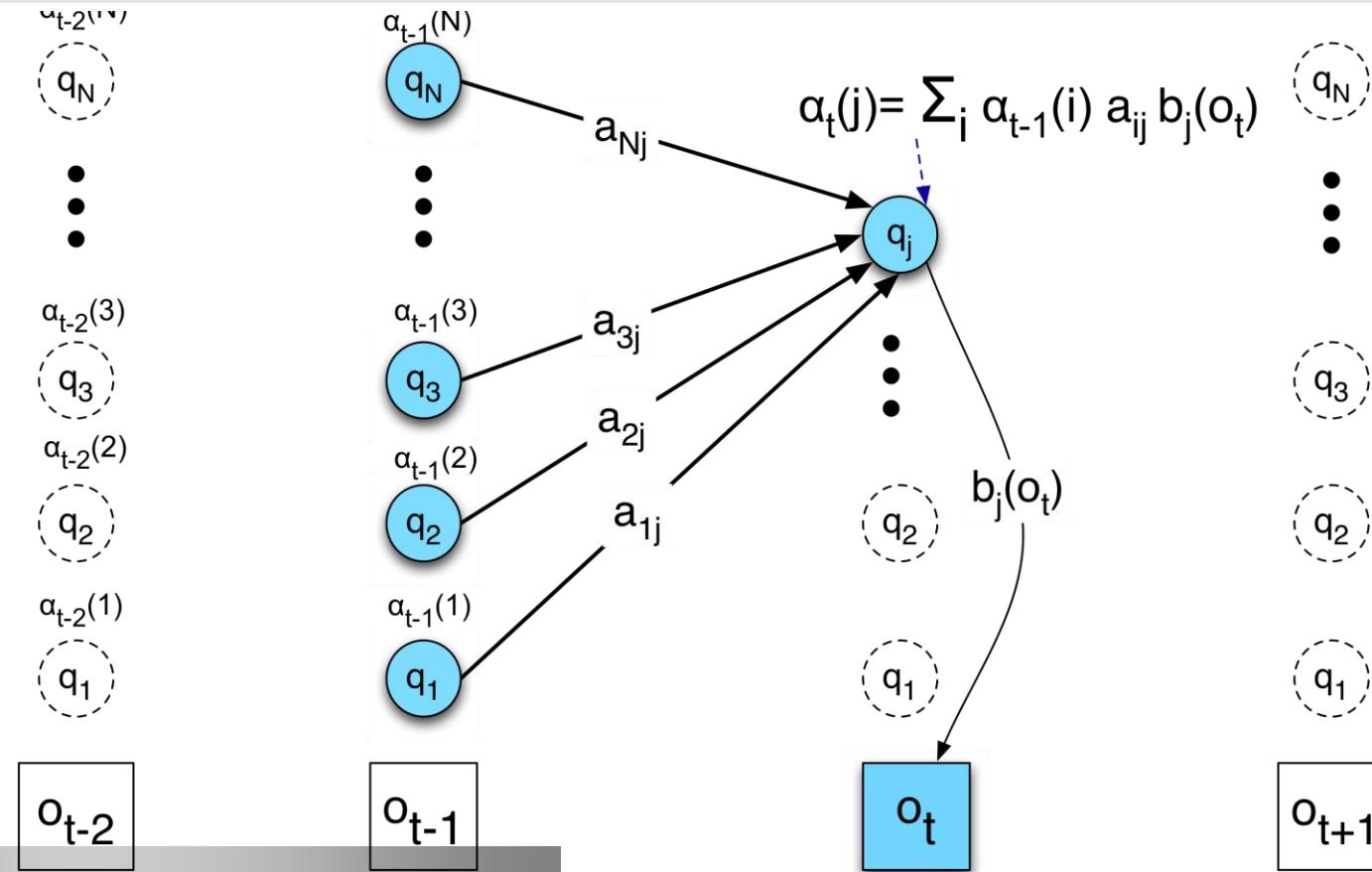


We update each cell

$\alpha_{t-1}(i)$ the **previous forward 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 Forward Algorithm

```
function FORWARD(observations of len  $T$ , state-graph of len  $N$ ) returns forward-prob
    create a probability matrix forward[ $N+2, T$ ]
    for each state  $s$  from 1 to  $N$  do ; initialization step
         $\text{forward}[s, 1] \leftarrow a_{0,s} * b_s(o_1)$ 
    for each time step  $t$  from 2 to  $T$  do ; recursion step
        for each state  $s$  from 1 to  $N$  do
             $\text{forward}[s, t] \leftarrow \sum_{s'=1}^N \text{forward}[s', t-1] * a_{s', s} * b_s(o_t)$ 
     $\text{forward}[q_F, T] \leftarrow \sum_{s=1}^N \text{forward}[s, T] * a_{s, q_F}$  ; termination step
    return forward[ $q_F, T$ ]
```

