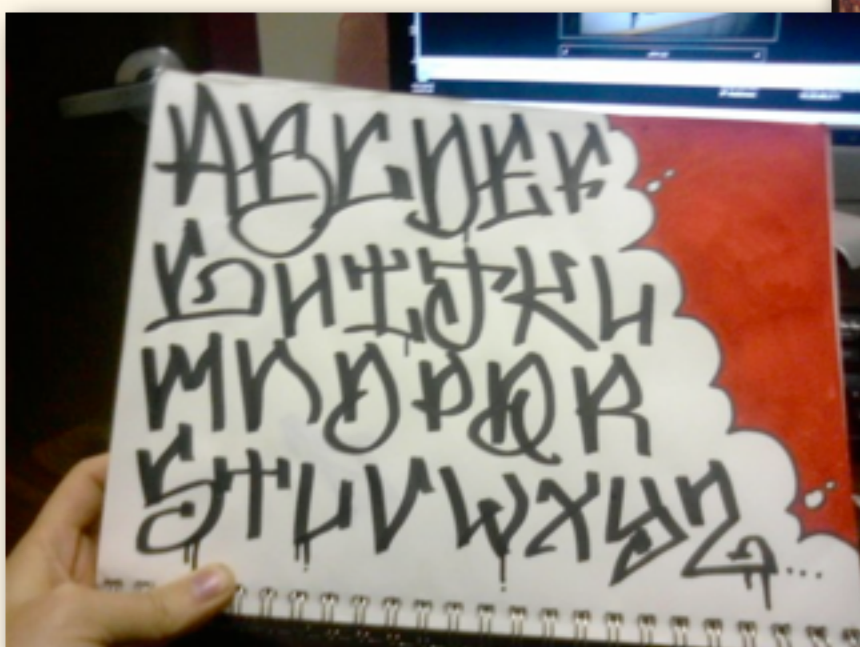
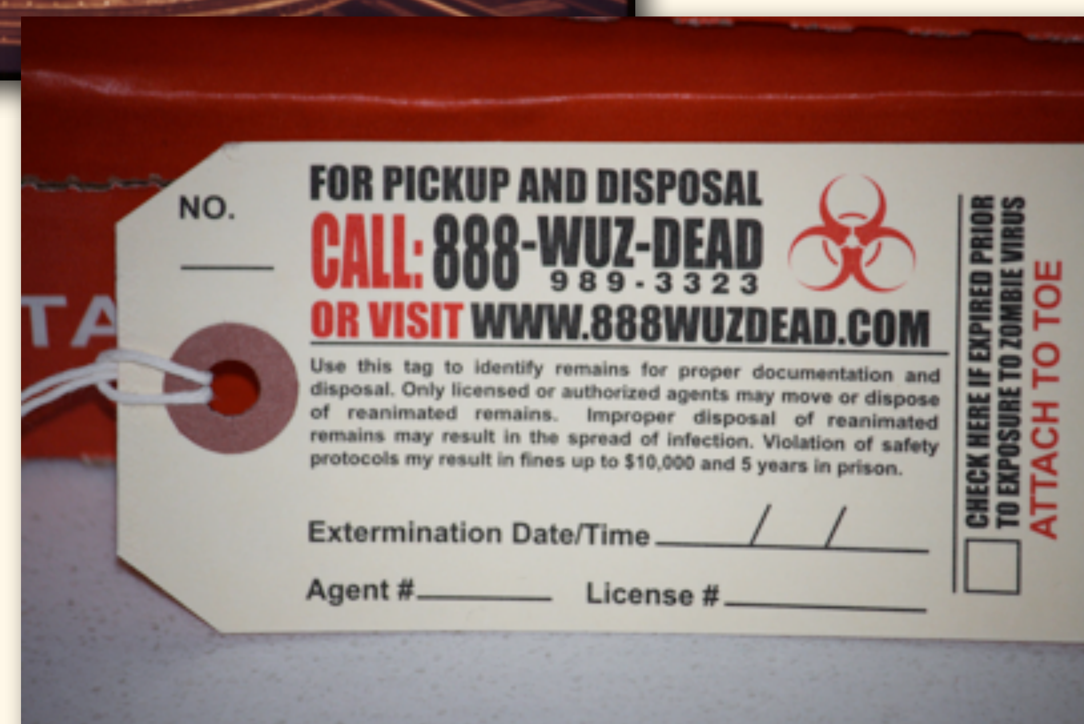
Hidden Markov Models, Part 1





Steven Bedrick
CS/EE 5/655, 10/22/14

Plan for the day:

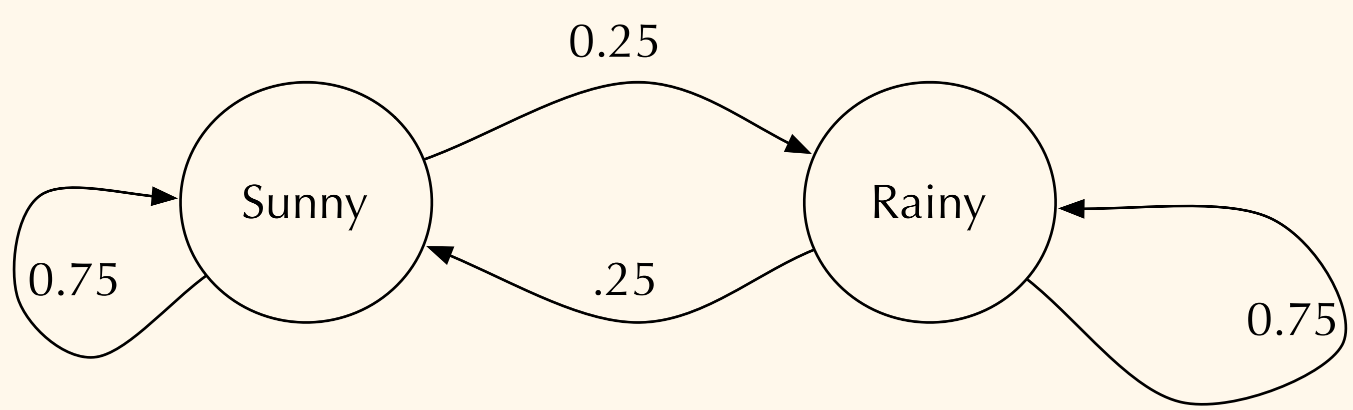
1. Quick Markov Chain review
2. Motivation: Part-of-Speech Tagging
3. Hidden Markov Models
4. Forward algorithm

Refresher: Markov Chain

A Markov Chain is a *memoryless* mathematical system, similar to a wFSA.

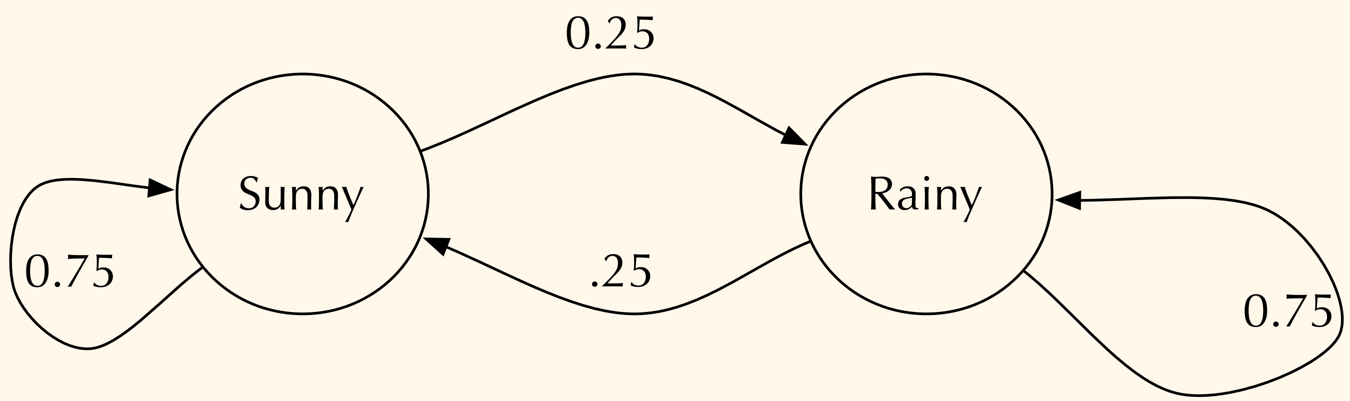
Consider the weather:

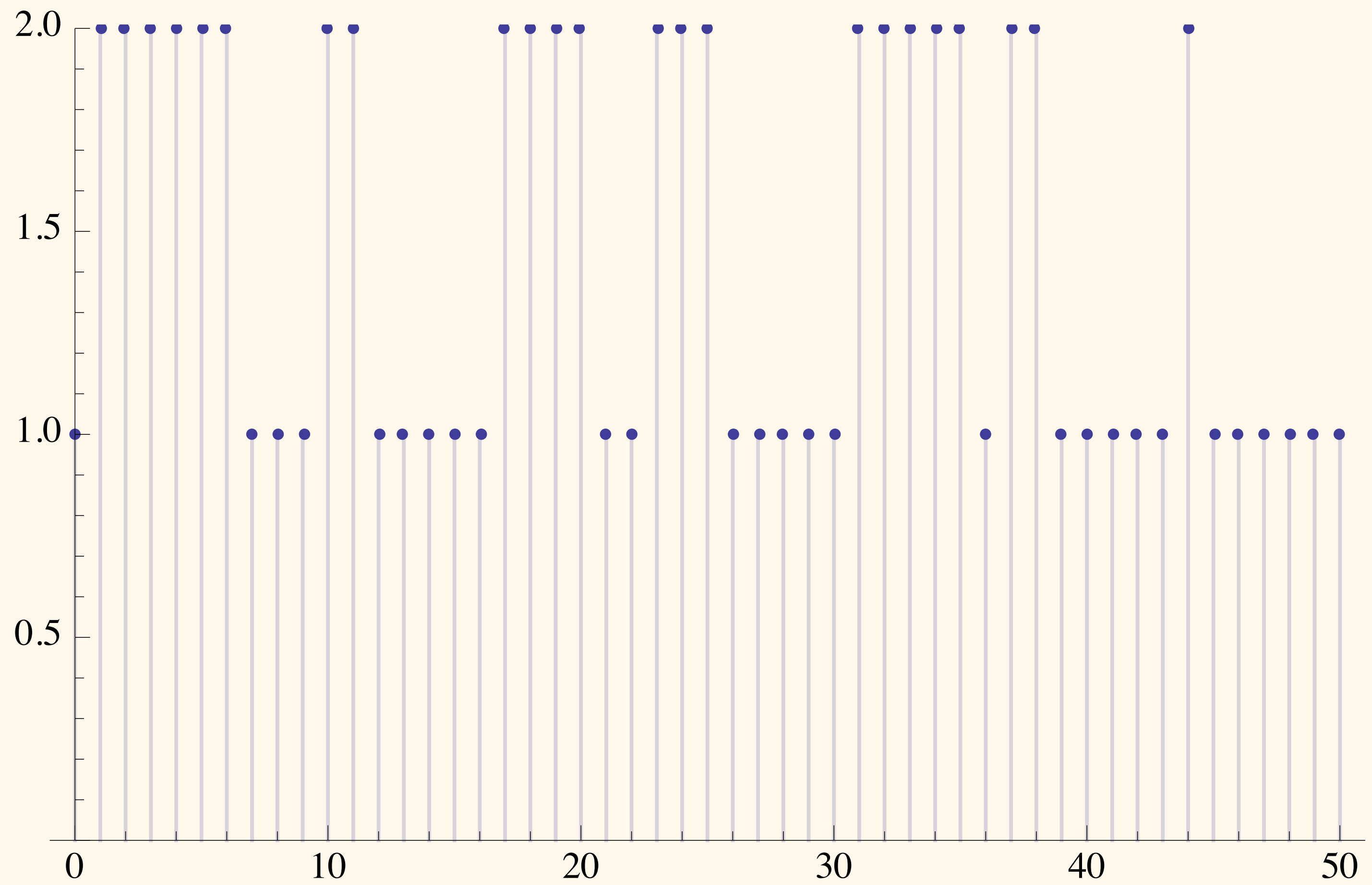
Today's weather is usually a good predictor of tomorrow's:



In Portland, if today was rainy, tomorrow has a 75% chance of the same.

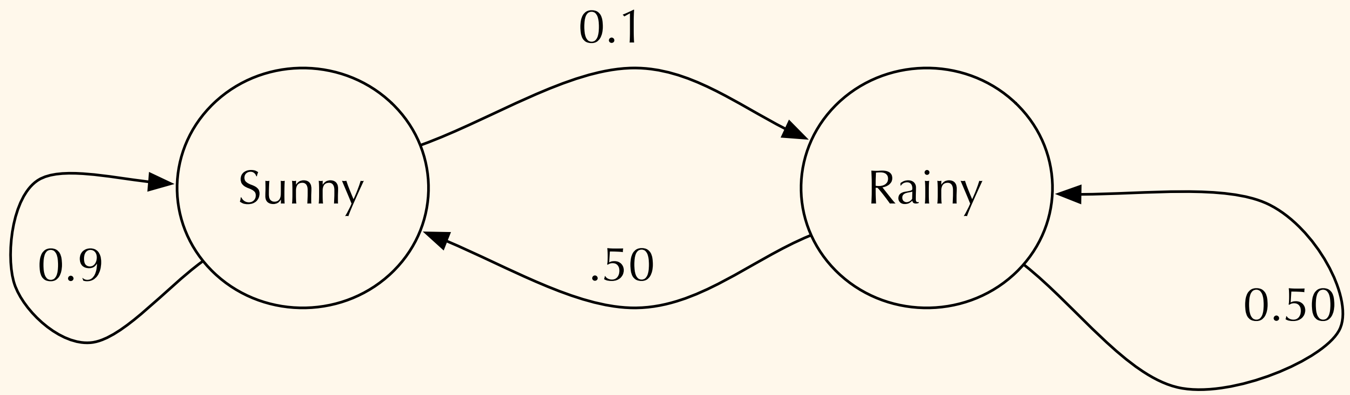
Refresher: Markov Chain



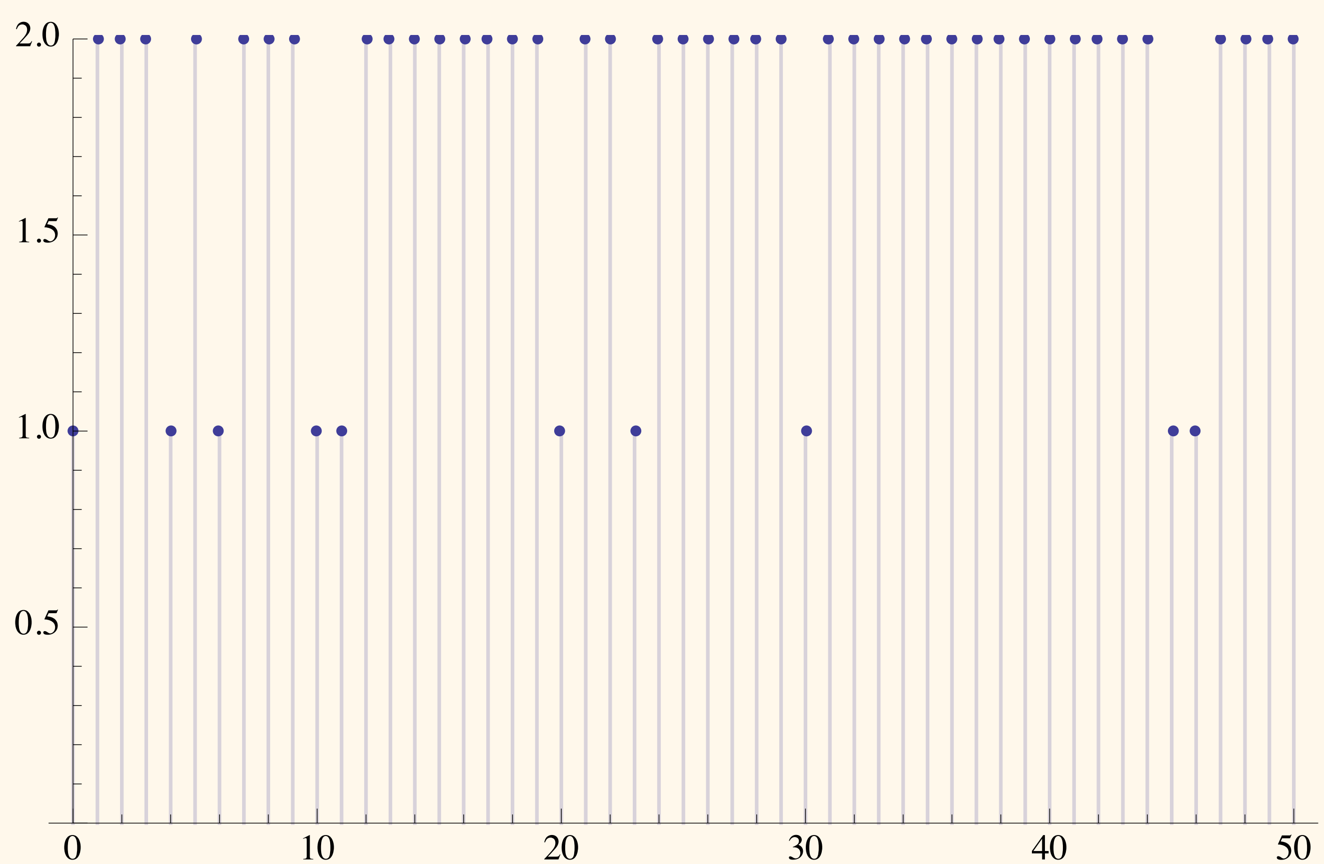


In Portland, if today was rainy, tomorrow has a 75% chance of the same.

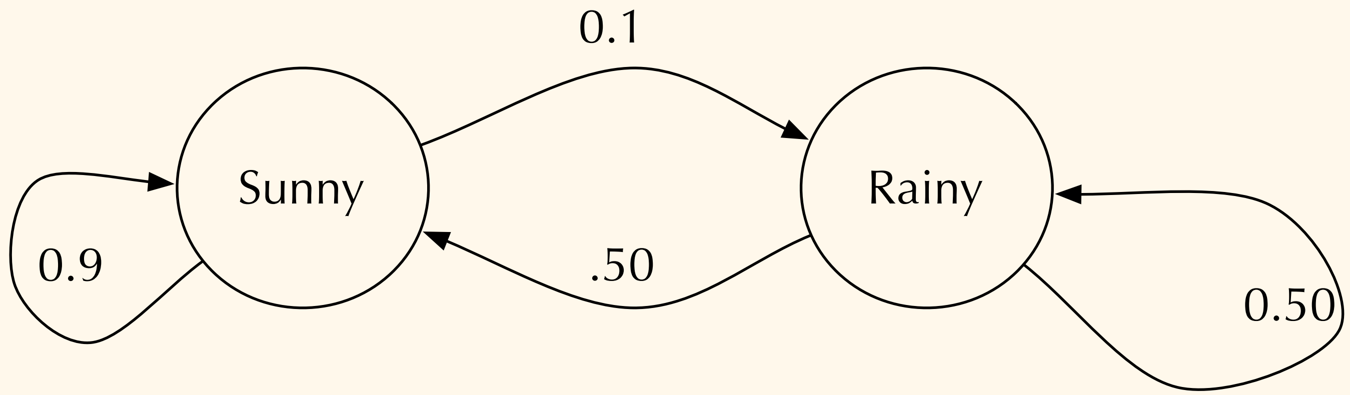
Refresher: Markov Chain



In Los Angeles, sun is far more common.



Refresher: Markov Chain



We can represent our Markov chain using a transition matrix:

	Sunny	Rainy
Sunny	0.9	0.1
Rainy	0.5	0.5

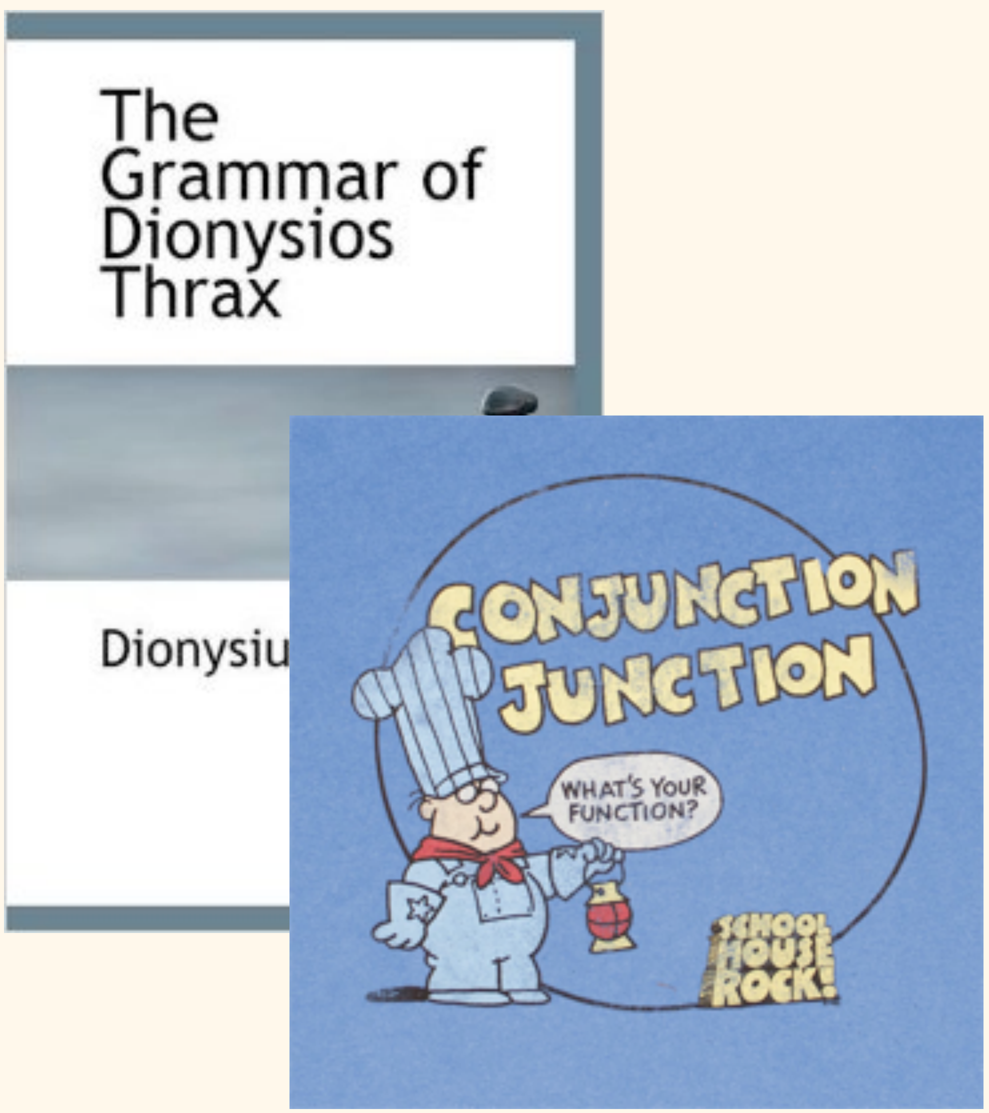
Plan for the day:

1. Quick Markov Chain review
2. Motivation: Part-of-Speech Tagging
3. Hidden Markov Models
4. Forward algorithm

What is a “part of speech”?

Quick definition:

Category of words (lexical items) grouped by syntactic function.



At least eight basic classes in English...

Noun	Monkeys
Pronoun	You
Adjective	Curious
Verb	Climb
Adverb	Quickly
Preposition	Until
Conjunction	And
Interjection	Egad!

... many classification schemes involve dozens.

Part-of-speech tagging is a fundamental building block of NLP. Why?

Thousands of commuters were trapped in cars overnight on highways in the greater Atlanta area, hundreds of students remained inside dozens of schools Wednesday morning and at least 50 children spent the night on school buses because of an ice storm that is still gripping the deepest parts of the South.

…

“This came very suddenly,” Craig Witherspoon, superintendent of Birmingham City Schools in Alabama, said Wednesday. An estimated 600 students in his district spent the night in schools, tended by about 100 staff members.

From “Ice Storm Strands Thousands in Ill-Equipped South”, NY Times 1/29/14

http://www.nytimes.com/2014/01/30/us/ice-storm-southern-united-states.html?hp

Part-of-speech tagging is a fundamental building block of NLP. Why?

Thousands of commuters were trapped in cars overnight on highways in the greater Atlanta area, hundreds of students remained inside dozens of schools Wednesday morning and at least 50 children spent the night on school buses because of an ice storm that is still gripping the deepest parts of the South.

…

"This came very suddenly," Craig Witherspoon, superintendent of Birmingham City Schools in Alabama, said Wednesday. An estimated 600 students in his district spent the night in schools, tended by about 100 staff members.

What is this text describing?

Verbs!

From "Ice Storm Strands Thousands in Ill-Equipped South", NY Times 1/29/14

http://www.nytimes.com/2014/01/30/us/ice-storm-southern-united-states.html?hp

Part-of-speech tagging is a fundamental building block of NLP. Why?

Thousands of commuters were trapped in cars overnight on highways in the greater Atlanta area, hundreds of students remained inside dozens of schools Wednesday morning and at least 50 children spent the night on school buses because of an ice storm that is still gripping the deepest parts of the South.

…

“This came very suddenly,” Craig Witherspoon, superintendent of Birmingham City Schools in Alabama, said Wednesday. An estimated 600 students in his district spent the night in schools, tended by about 100 staff members.

Where is it taking place?

Who said what?

Proper Nouns!

From “Ice Storm Strands Thousands in Ill-Equipped South”, NY Times 1/29/14

http://www.nytimes.com/2014/01/30/us/ice-storm-southern-united-states.html?hp

Part-of-speech tagging is a fundamental building block of NLP. Why?

PoS data is used for many NLP tasks:

Information extraction (as in NYT example)

Syntactic analysis (parsing)

Machine translation

Etc.

A key issue is that of choosing a classification scheme (tag set)...

One of the first large-scale tagged corpora was the Brown Corpus.

- Henry Kučera and Nelson Francis, Brown University, published 1967 in book form
- 1 million words from a diverse sample of 500 publications
- Houghton-Mifflin used the the corpus for the 1969 edition of the American Heritage Dictionary. The corpus included a chapter from Robert Heinlein's 1964 sci-fi novel Stranger in a Strange Land, and this is why grok 'empathetically understand' is in most dictionaries
- 85 tags; some infamous decisions include:
    - a tag for not and n't
    - tags specific to each form of various light verbs (forms of be, do, have, etc.)
    - the FW foreign word tag
- Tags were generated by a program enumerated possible tag sequences, from which human annotators selected the best
- This was used to develop fully automated tagging systems:
    - The CLAWS tagger enumerated all possible tag sequences (which may be an enormous set) then selected the one which maximized the HMM probabilities estimated from this corpus
    - Steven DeRose and Ken Church independently discovered (in 1988) dynamic programming methods (akin to the Viterbi algorithm) to achieve the same objective without this expensive enumeration

Tagset-related background & commentary thanks to Kyle Gorman

Today, the Penn Treebank is the most commonly-used tagged corpus.

- Tagset designed by (linguist) Beatrice Santorini (though many of her proposed distinctions were vetoed by engineers on the project); 45 tags in all
- Whereas the Brown corpus seems to attempt to minimize token-given-tag entropy (many tags have only one token), the Treebank tag set minimizes tag-given-word entropy (i.e., the kind of entropy that makes automated tagging difficult)
- Occasionally permits ambiguous tags (e.g., `JJ|NN`)

Tagset-related background & commentary thanks to Kyle Gorman

CC	Coordinating conjunction	and
CD	Cardinal Number	12, 1,000,000
DT	Determiner	the
EX	Existential there	there
FW	Foreign Word	persona non grata
IN	Preposition or subordinating conjunction	under, that
JJ	Positive adjective	big
JJR	Comparative adjective	bigger
JJS	Superlative	biggest
LS	Marker for list items	A.
MD	Modal	may
NN	Singular (or mass) common noun	dog, grass
NNS	Plural common noun	dogs
NNP	Singular proper noun	Vincent
NNPS	Plural proper noun	Beatles
PDT	Predeterminer	quite
POS	Possessive clitic	s
PRP	Personal pronoun	she, myself
PRP$	Possessive pronoun	yours
RB	Positive adverb	RB
RBR	Comparative adjective	late
RBS	Superlative adjective	later
RP	Particle	latest
SYM	Symbol	&
TO	to	to
UH	Interjection	uh, yes
VB	Uninflected verb	strive (in _to strive_)
VBD	Simple past tense verb	strove
VBG	Present participle or gerund	striving
VBN	Past participle	striven
VBP	Non-3rd person present verb	strive (in _We strive to..._)
VBZ	3rd person singular present verb	strives
WDT	Wh-determiner	which
WP	Wh-pronoun	whom
WP$	Possessive wh-pronoun	whose
WRB	Wh-adverb	how

Treebank tagset (2/2)

- Punctuation tags: `#` `$` ` `` ` `''` `(` `)` `,` `.` `:`
- Major critiques:
  - `EX`: why distinguish between existential *there* and existential *it* (*It is known*…) or "weather *it*" (*It rains a lot in Portland*)?
  - `TO`: *to* can be many parts of speech (infinitive marker, preposition, etc.), why punt?
- Distinctions not made, but recoverable:
  - `IN`: subordinating conjunction (heading a clause) vs. preposition (heading a prepositional phrase)
  - `UH`: actual interjections (*yes*) vs. filled pause (*uh, um*)
  - `DT`: articles (*a*, *an*, *the*) vs. demonstratives (*those*)
  - `PRP`: actual personal pronoun (*I*, *her*) vs. reflexive pronouns (*myself*)

Tagset-related background & commentary thanks to Kyle Gorman

Once you’ve got a tag set, the next question becomes: how to assign tags to words?

There are three main families of approaches:

1. Rule-based
2. Stochastic
3. Transformation-based learning

Why not rely strictly on a dictionary?

Many words serve different functions in different situations:

> "He got a good deal on his car."
>
> "He will deal well with car troubles."
>
> "Secretariat is expected to race tomorrow."
>
> "Secretariat won the race last week."

Rule-based taggers generally begin with a dictionary of possible word-tag pairs...

... then use a set of (many!) hand-written rules to handle ambiguous situations.

Rules can be based on context:

"He got a good/JJ deal/NN on/IN his car."

```
If JJ(prev-word), NN; else VB
```

Or on morphology:

"We are going glorping/?? today.

```
If /ing$/, VBG
```

Modern rule-based taggers use many kinds of syntactic and morphological information...

... and often include some information about probabilities, as well.

Stochastic PoS techniques rely entirely on probability.

Notation:

w_1^n	Word sequence of length *n*
t_1^n	Tag sequence of length *n*

The goal of a stochastic PoS tagger is to find:

$$\hat{t}_1^n = \arg\max_{t_1^n} P(t_1^n | w_1^n)$$

$$\hat{t}_1^n = \arg\max_{t_1^n} P(t_1^n|w_1^n)$$

This is all well and good, but the whole point is that we *don't* know $P(t_1^n|w_1^n)$.

Maybe Bayes' Rule can help?

$$\hat{t}_1^n = \arg\max_{t_1^n} P(w_1^n|t_1^n)P(t_1^n)$$

Likelihood

Prior

$$\hat{t}_1^n = \arg\max_{t_1^n} P(w_1^n|t_1^n)P(t_1^n)$$

This is better, but still too hard to actually calculate. Let's make two assumptions:

Words only depend on their part of speech tag (not on their neighbors'):

$$P(w_1^n|t_1^n) \approx \prod_{i=1}^{n} P(w_i|t_i)$$

The probability of any given tag only depends on that of the previous tag, not on the whole sequence.

$$P(t_1^n) \approx \prod_{i=1}^{n} P(t_i|t_{i-1})$$

Putting it all together, we get:

$$\hat{t}_1^n = \underset{t_1^n}{\arg\max}\, P(t_1^n | w_1^n) \approx \underset{t_1^n}{\arg\max} \prod_{i=1}^{n} P(w_i | t_i) P(t_i | t_{i-1})$$

Probability of word given tag

Probability of tag given previous tag

Does this look familiar?

One way to compute this: Hidden Markov Model

Andrei Andreievich Markov
1856–1922

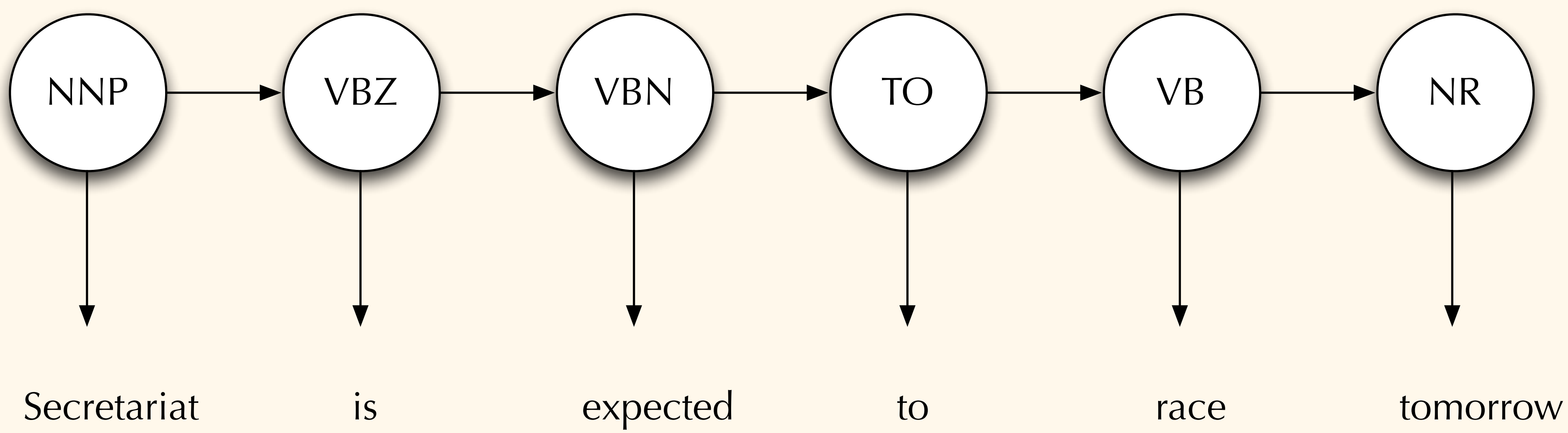
HMMs are a type of stochastic model used to examine sequential data.

The basic idea: there are two parameters changing over time, but we can only directly observe one of them. We want to know about the other.

For example:

Tags can be thought of as hidden states...

Observed words can be thought of as emissions...



Formally, an HMM is fully described as:

$Q = q_1, q_2, q_3 ... q_n$	A set of *N* hidden states
$A = a_{11} a_{12} ... a_{n1} ... a_{nn}$	A transition probability matrix giving the probabilities of going from state *i* to *j*
$O = o_1 o_2 ... o_T$	A sequence of *T* observations
$B = b_i(o_t)$	A set of observation likelihoods (aka *emission probabilities*) of observation o_t being generated from state b_i.
q_0, q_F	Special start and stop states, together with transition probabilities $a_{01} \ldots$

We will steal an example from Jason Eisner.

Jason Eisner
??? – Present

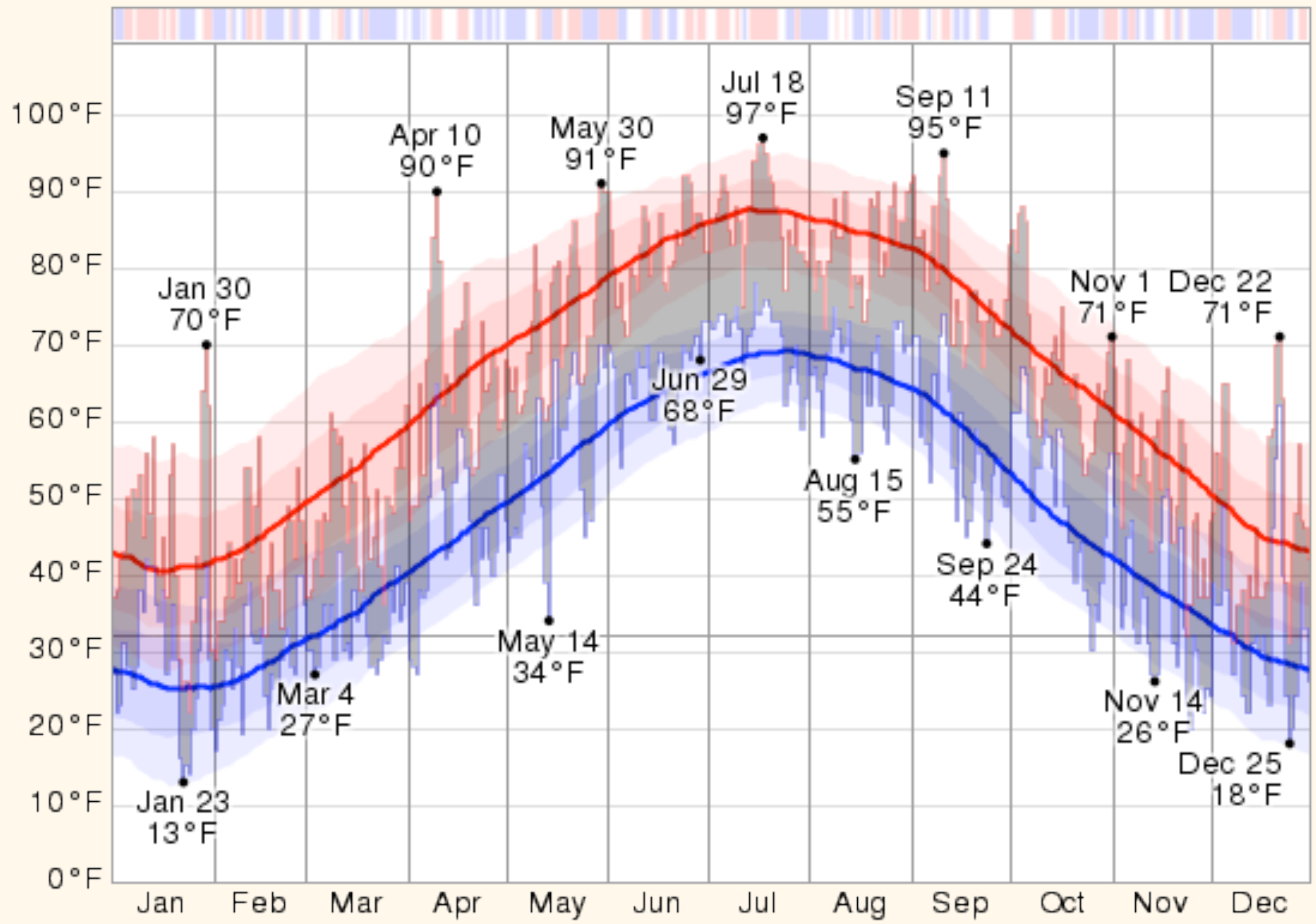
It is 2799; you are a climatologist studying the history of global warming.

Following the Zombie Apocalypse of 2325, all records of 20th-century weather were destroyed...

Eisner J. *An Interactive Spreadsheet for Teaching the Forward-Backward Algorithm*. In: Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching NLP and CL Radev D, Brew C, editors. Philadelphia; 2002. pp. 10–8.

... however, archaeologists excavating the ruins of Baltimore recently discovered Jason's diary...

... in which he obsessively recorded how often he ate ice cream over the summer of 2013.



We can infer that the weather influenced how much ice cream Jason ate on any given day.

We can further infer that today's weather is at least somehow related to yesterday's weather.

An HMM will let us model the situation:

Observed variable: Ice cream consumption

Hidden variable: Weather

Let's simplify things and say that there are two kinds of weather ("hot" and "cold"), and that he either ate 1, 2, or 3 units of ice cream per day.

Our transition matrices:

A:

	Hot	Cold
Hot	0.7	0.3
Cold	0.6	0.4

B:

	1	2	3
Hot	0.2	0.4	0.4
Cold	0.5	0.4	0.1

$a_{0,\text{Hot/Cold}}$:

	Start
Hot	0.8
Cold	0.2

Our transition matrices:

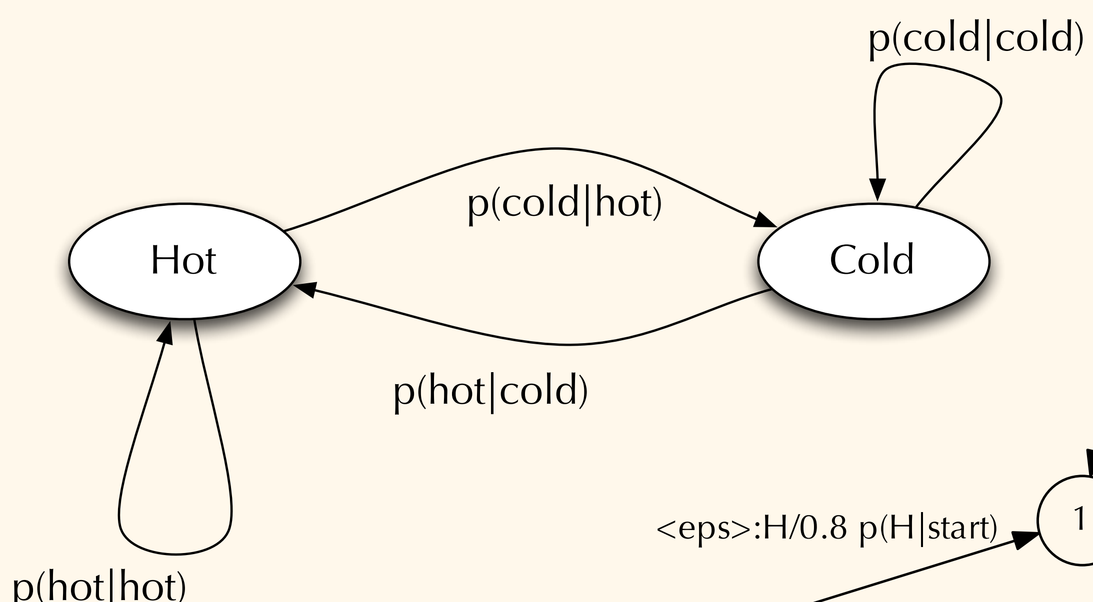
A:

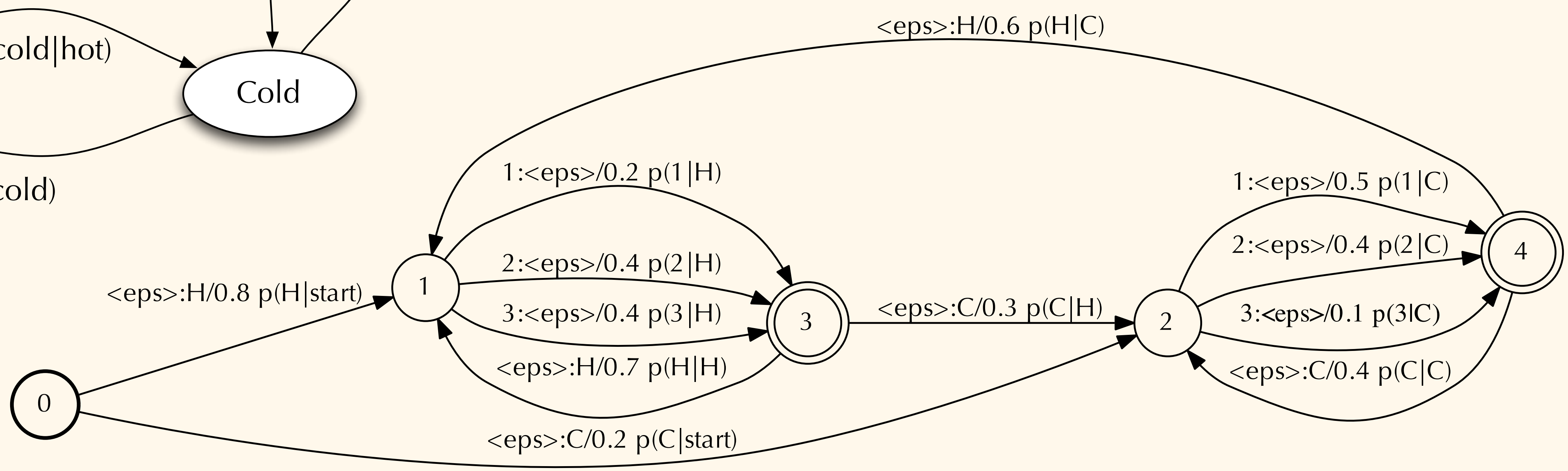
	Hot	Cold
Hot	0.7	0.3
Cold	0.6	0.4

B:

	1	2	3
Hot	0.2	0.4	0.4
Cold	0.5	0.4	0.1

We can represent parts of HMMs using wFS{A,T}s!





A note about starting and stopping conditions:

$a_{0,\text{Hot/Cold}}$:

	Start
Hot	0.8
Cold	0.2

In this example, we know *a priori* that the journal is from the summer months, so *P(Hot)* is higher than *P(Cold)*.

We don't have any reason to believe that the weather affected when Jason stopped his diary, so the stop probabilities are identical.

Can you think of an HMM problem where they might not be?
(Hint: think POS tagging)

There are three fundamental kinds of questions that we can ask with an HMM:

1. *Likelihood:* Given a sequence of states, what is the most likely observed sequence? *or* How likely is a given observation sequence?

2. *Decoding:* Given an observation sequence and a fully-specified HMM, what is the most likely sequence of states to have produced that observation?

3. *Learning:* Given an observation sequence and a set of states, what are the likely transition and emission probabilities (A and B)?

There are three fundamental kinds of questions that we can ask with an HMM:

1. ***Likelihood:* Given a sequence of states, what is the most likely observed sequence? *or* How likely is a given observation sequence?**

2. *Decoding:* Given an observation sequence and a fully-specified HMM, what is the most likely sequence of states to have produced that observation?

3. *Learning:* Given an observation sequence and a set of states, what are the likely transition and emission probabilities (A and B)?

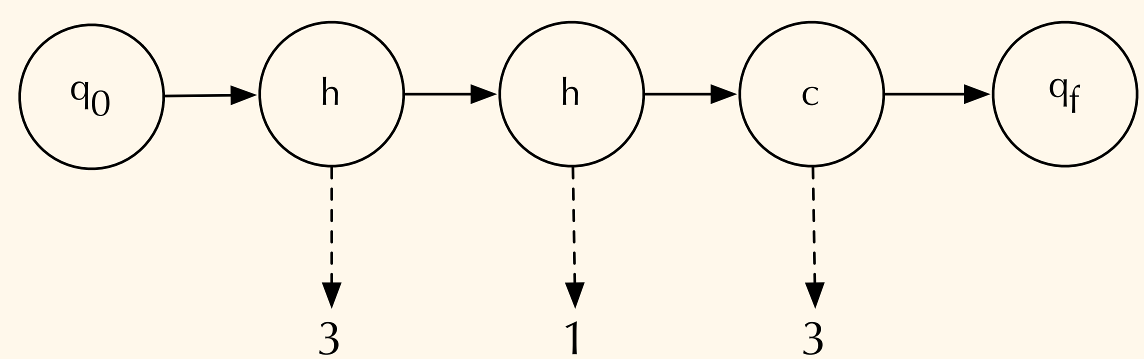
Let's say we have a sequence of diary entries:

$$O = 3, 1, 3$$

How likely is this sequence given the model described earlier? $P(O|\lambda)$

We start with a simpler problem: calculating the probability of a specific observation/state pair.

$$Q = hot, hot, cold$$



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

$$P(3,1,3|h,h,c) = P(3|h) \times P(1|h) \times P(3|c)$$

$$O = 3, 1, 3$$
$$Q = hot, hot, cold$$
$$P(O|Q) = \prod_{i=1}^{T} P(o_i|q_i)$$
$$P(3,1,3|h,h,c) = P(3|h) \times P(1|h) \times P(3|c)$$

But that's not the full story, since Q itself is only one of many sequences our machine can generate. So:

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

$$\begin{aligned} P([3,1,3],[h,h,c]) = {} & P(h|start) \times P(h|h) \times P(c|h) \\ & \times P(3|h) \times P(1|h) \times P(3|c) \end{aligned}$$

Now that we can find out the joint probability of an observation and a given state sequence...

... we know how to find the probability of the observation itself:

$$P(O) = \sum_{Q} P(O, Q) = \sum_{Q} P(O|Q)P(Q)$$

Intuition: the probability of an observation is the sum of the probabilities of all the different ways for the model to generate that observation.

$$\begin{aligned} P(3,1,3) = P([3,1,3],[h,h,h]) + P([3,1,3],[h,h,c]) + \\ P([3,1,3],[h,c,h]) + P([3,1,3],[c,h,h]) \ldots \end{aligned}$$

Problem: for N states and T observations, calculating $P(O)$ in this way is $O(N^T)$.

$$P(O) = \sum_{Q} P(O, Q) = \sum_{Q} P(O|Q)P(Q)$$

Often, N and T are large!*

Instead, we can use the $O(N^2T)$ *Forward Algorithm* to compute $P(O)$.

This is a simple instance of dynamic programming!

**Not that they have to be very large in order to cause problems! 20 states, 10 observations = tens of trillions of calculations.*

The key insight: build a *trellis* that keeps track of the probabilities of different paths through the machine.

This is represented by a T (# obs.) by N (# states) matrix α;

Each $\alpha_t(j)$ represents the probability of the machine being in state j given the first t observations ("forward probability").

Formally: $\alpha_t(j) = P(o_1, o_2 ... o_t, q_t = j | \lambda)$

$q_t = j$: "the t^{th} state in the sequence is state j"

Calculating $\alpha_t(j) = P(o_1, o_2 ... o_t, q_t = j|\lambda)$ is fairly straightforward:

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

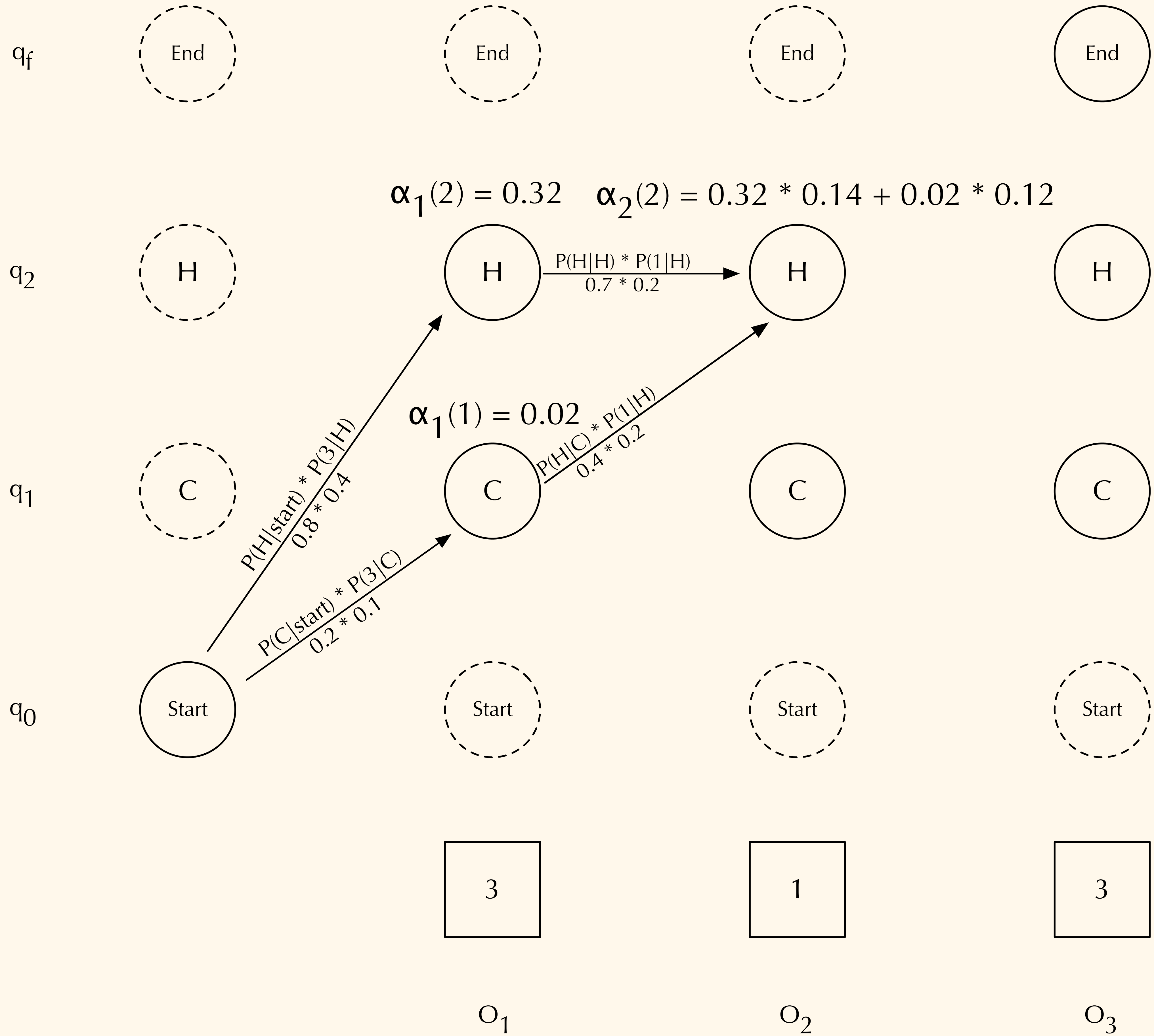
Calculating $\alpha_t(j) = P(o_1, o_2 ... o_t, q_t = j | \lambda)$ is fairly straightforward:

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

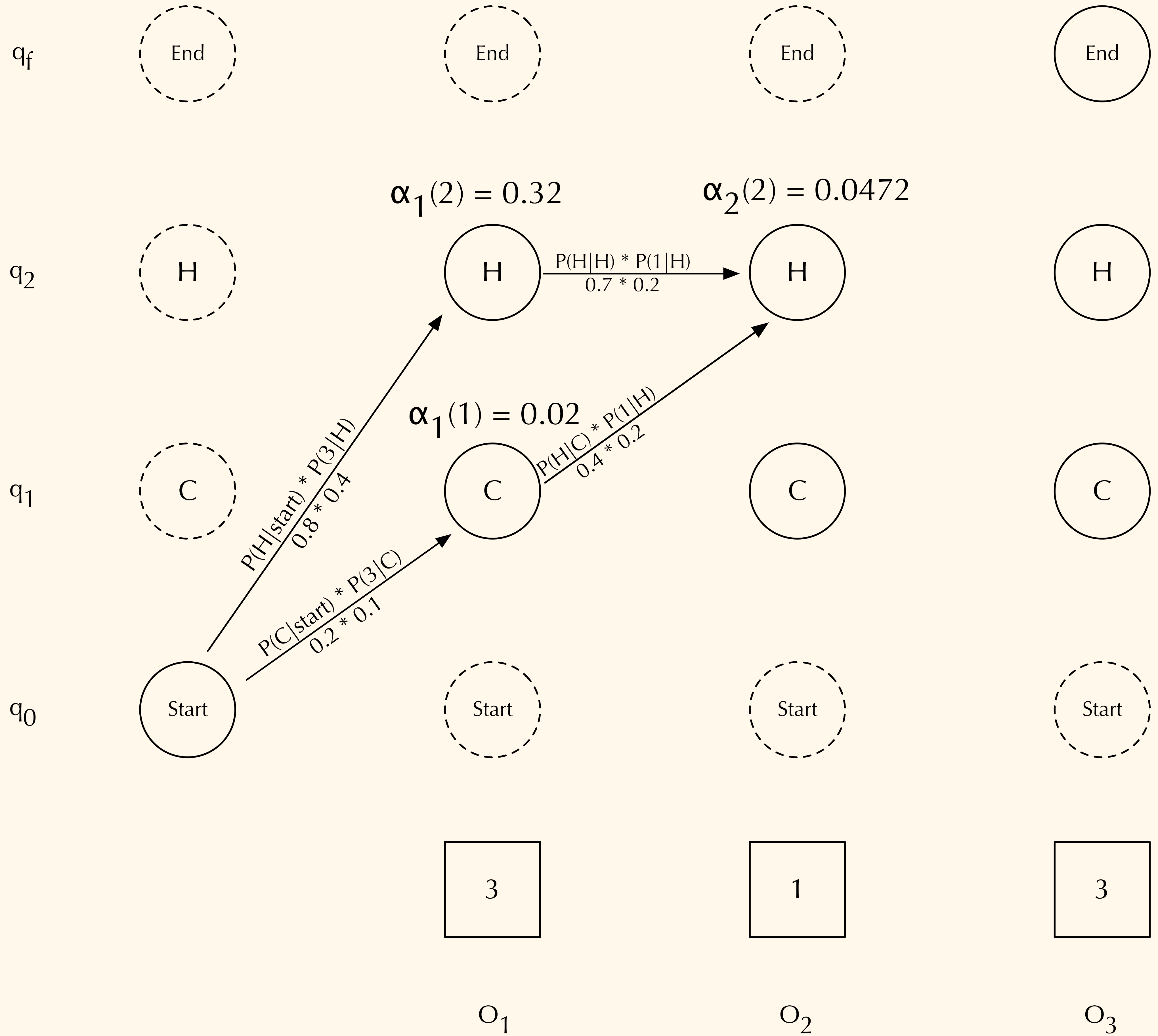
Previous time step's forward probability for state *i*

Transition prob. from previous state *i* to current state *j*

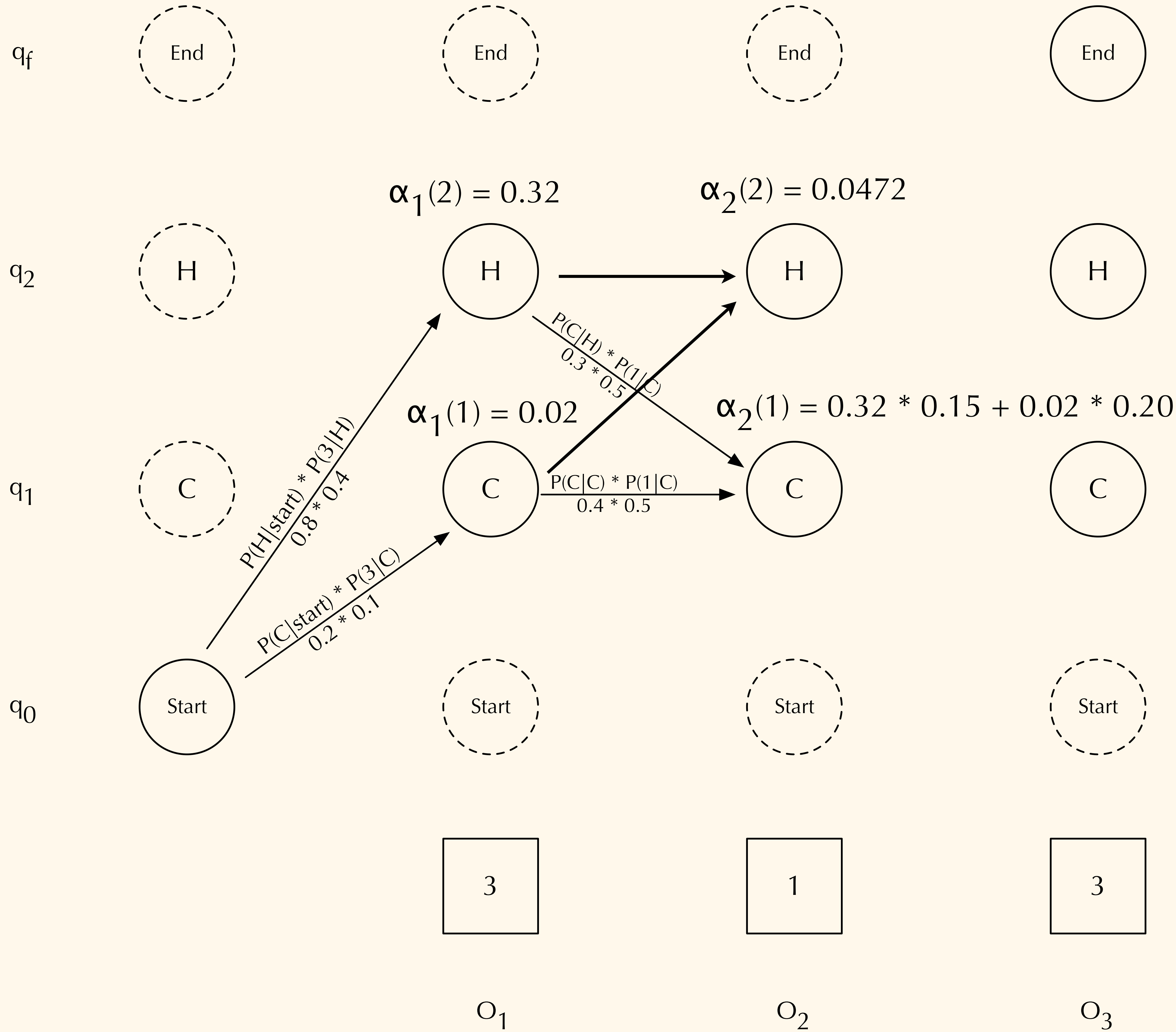
Emission likelihood for symbol o_t given current state *j*

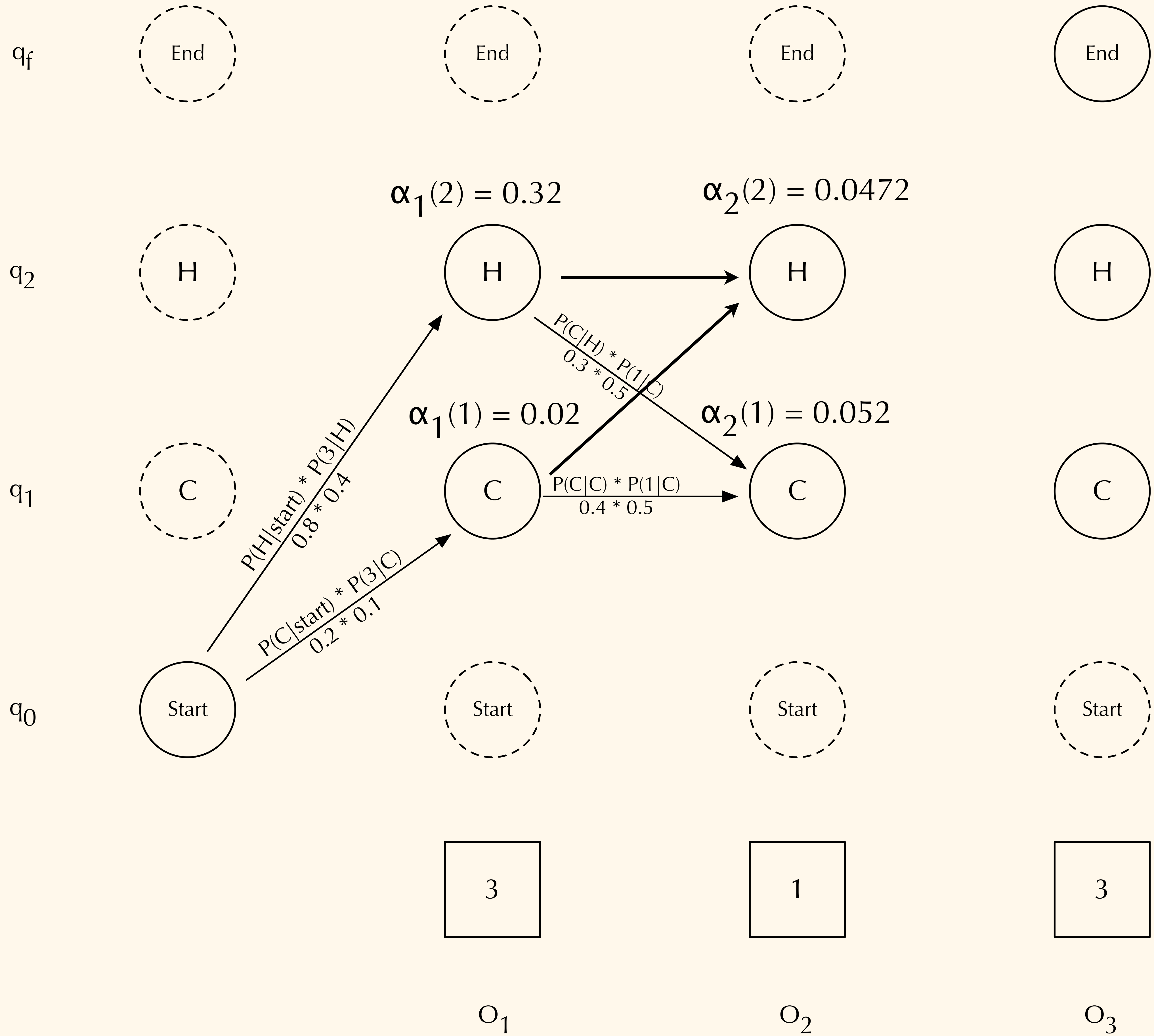


q_f
q_2
q_1
q_0
End
End
End
End
α_1(2) = 0.32
α_2(2) = 0.32 * 0.14 + 0.02 * 0.12
H
H
H
H
P(H|H) * P(1|H)
0.7 * 0.2
α_1(1) = 0.02
C
C
C
C
P(H|start) * P(3|H)
0.8 * 0.4
P(H|C) * P(1|H)
0.4 * 0.2
P(C|start) * P(3|C)
0.2 * 0.1
Start
Start
Start
Start
3
1
3
O_1
O_2
O_3

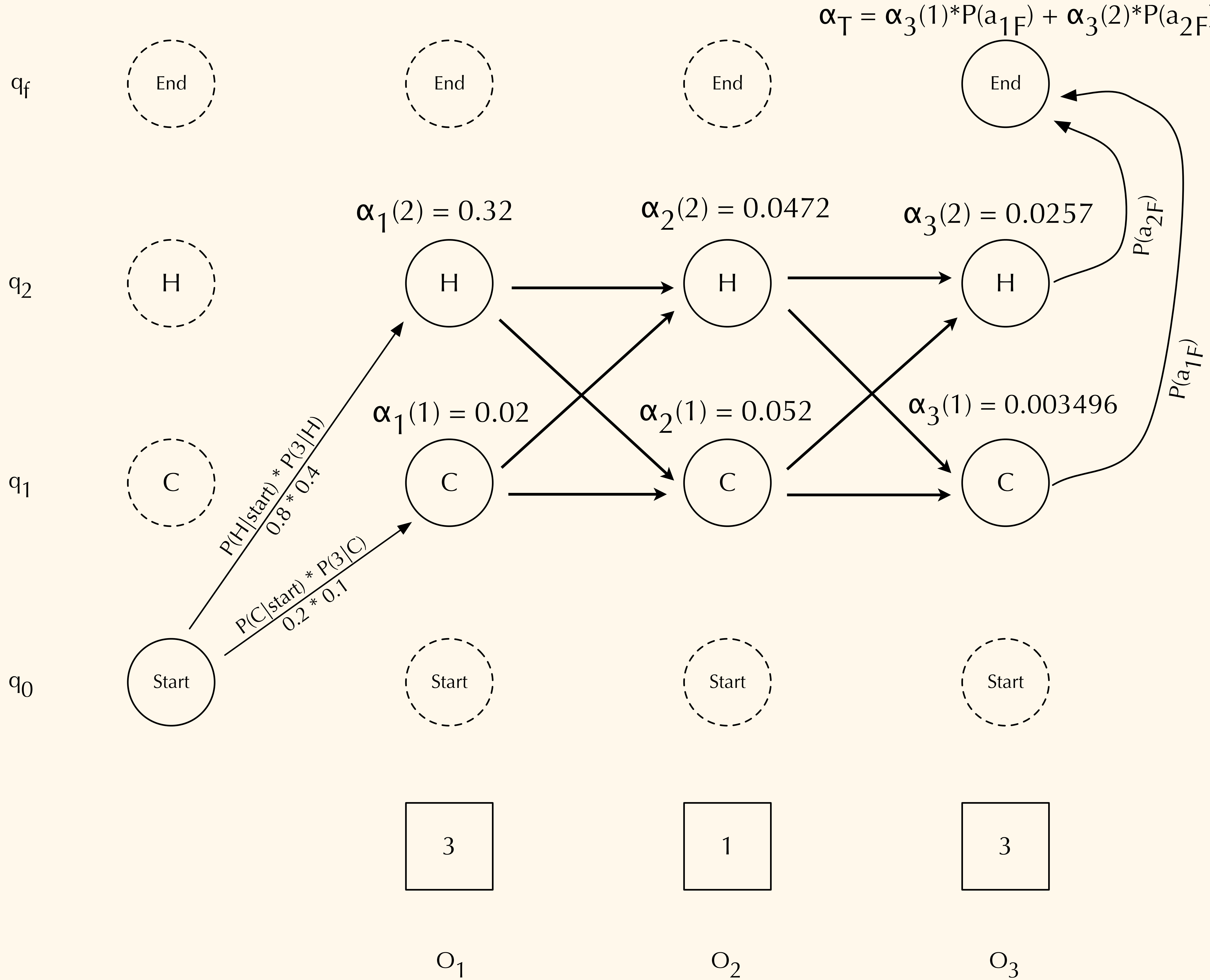


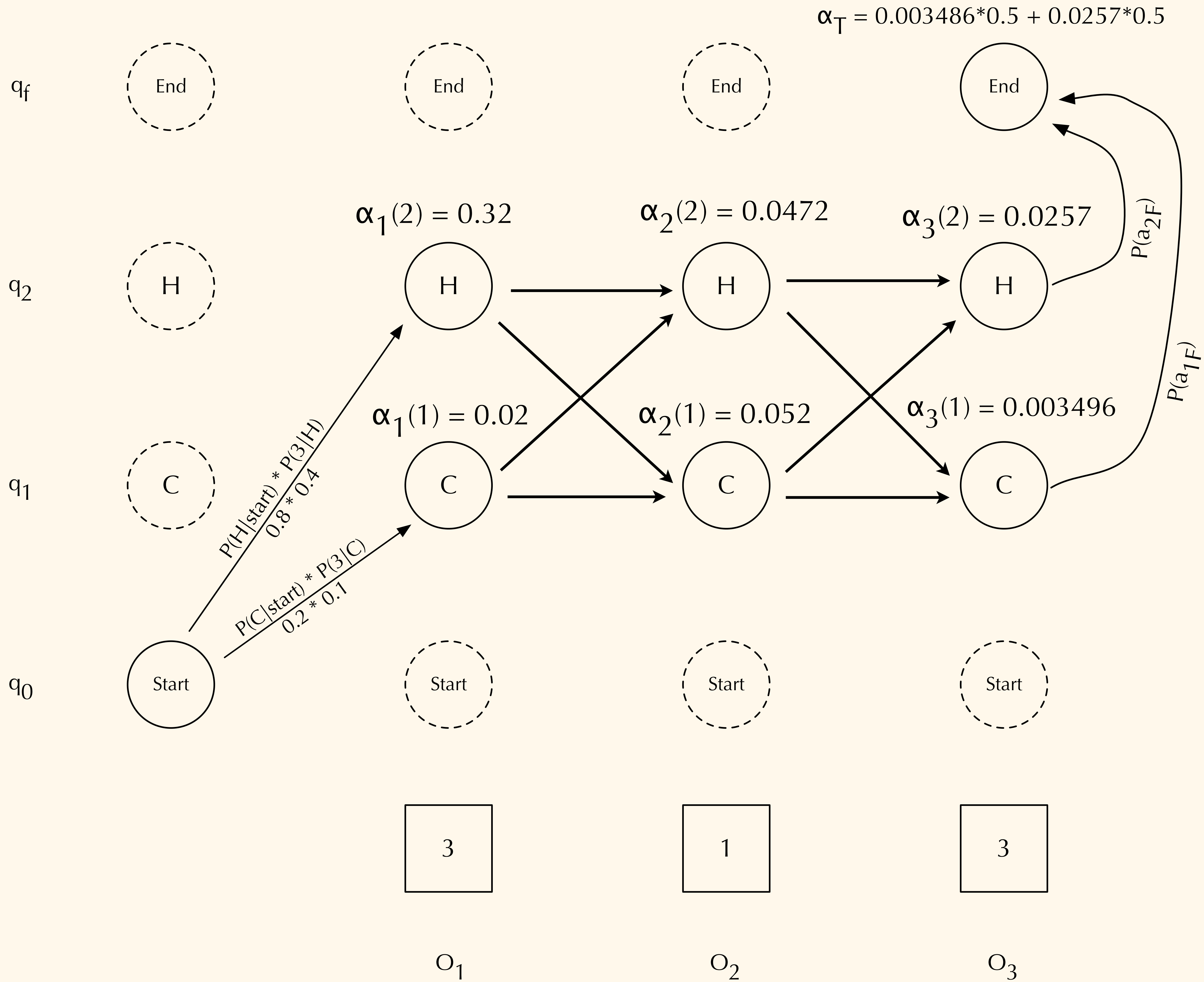
q_f
q_2
q_1
q_0
End
End
End
End
α_1(2) = 0.32
α_2(2) = 0.0472
H
H
H
H
P(H|H) * P(1|H)
0.7 * 0.2
α_1(1) = 0.02
C
C
C
C
P(H|start) * P(3|H)
0.8 * 0.4
P(H|C) * P(1|H)
0.4 * 0.2
P(C|start) * P(3|C)
0.2 * 0.1
Start
Start
Start
Start
3
1
3
O_1
O_2
O_3


q_f
q_2
q_1
q_0
End
End
End
End
$\alpha_1(2) = 0.32$
$\alpha_2(2) = 0.0472$
H
H
H
H
P(C|H) * P(1|C)
0.3 * 0.5
$\alpha_1(1) = 0.02$
$\alpha_2(1) = 0.32 * 0.15 + 0.02 * 0.20$
P(H|start) * P(3|H)
0.8 * 0.4
C
C
C
C
P(C|C) * P(1|C)
0.4 * 0.5
P(C|start) * P(3|C)
0.2 * 0.1
Start
Start
Start
Start
3
1
3
O_1
O_2
O_3

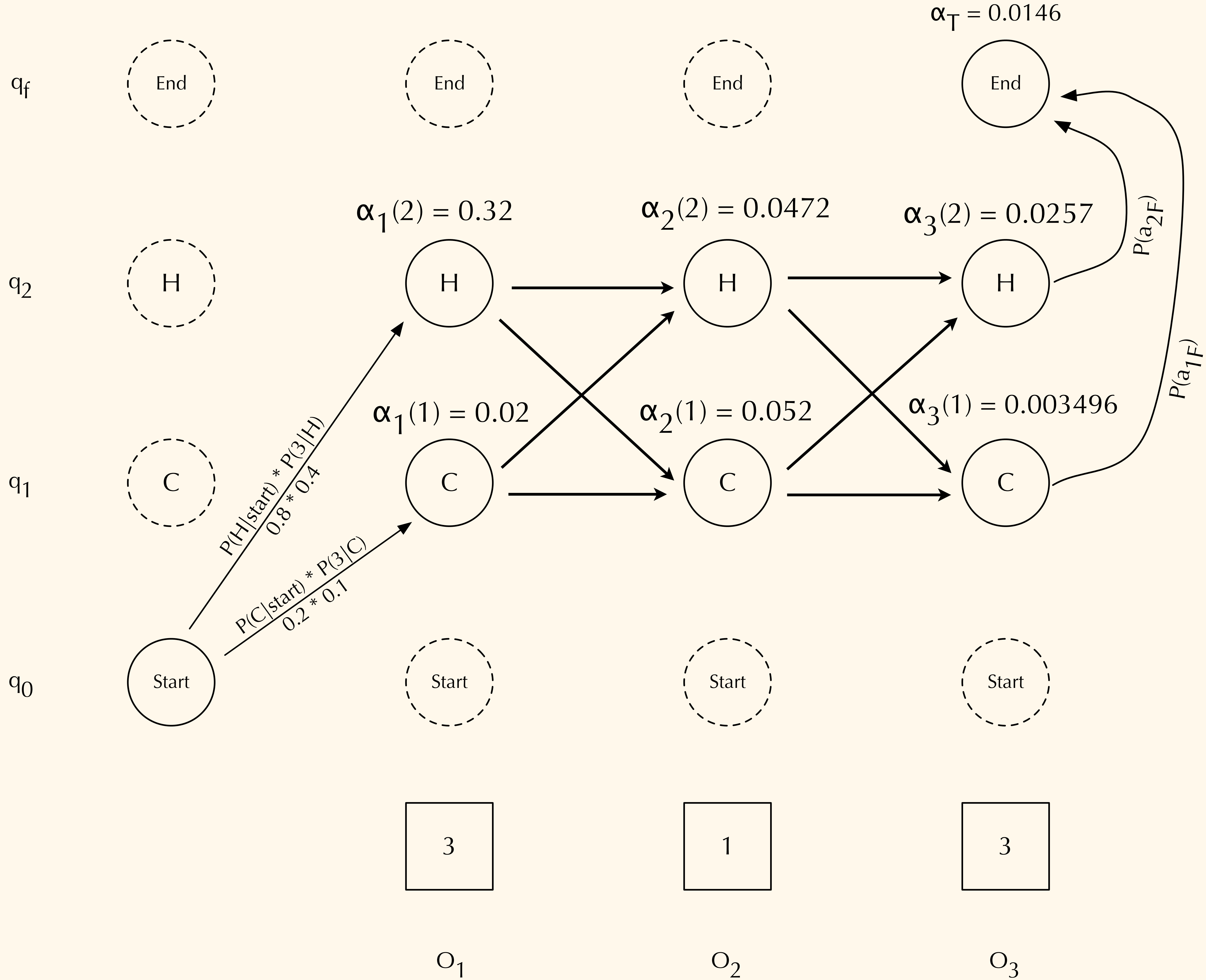


qf
q2
q1
q0
End
End
End
End
α1(2) = 0.32
α2(2) = 0.0472
H
H
H
H
P(C|H) * P(1|C)
0.3 * 0.5
α1(1) = 0.02
α2(1) = 0.052
P(H|start) * P(3|H)
0.8 * 0.4
C
C
C
C
P(C|C) * P(1|C)
0.4 * 0.5
P(C|start) * P(3|C)
0.2 * 0.1
Start
Start
Start
Start
3
1
3
O1
O2
O3


αT = α3(1)*P(a1F) + α3(2)*P(a2F)
qf
q2
q1
q0
End
End
End
End
H
H
H
H
C
C
C
C
Start
Start
Start
Start
α1(2) = 0.32
α2(2) = 0.0472
α3(2) = 0.0257
α1(1) = 0.02
α2(1) = 0.052
α3(1) = 0.003496
P(a2F)
P(a1F)
P(H|start) * P(3|H)
0.8 * 0.4
P(C|start) * P(3|C)
0.2 * 0.1
3
1
3
O1
O2
O3



αT = 0.003486*0.5 + 0.0257*0.5
qf
q2
q1
q0
End
End
End
End
H
H
H
H
C
C
C
C
Start
Start
Start
Start
α1(2) = 0.32
α2(2) = 0.0472
α3(2) = 0.0257
α1(1) = 0.02
α2(1) = 0.052
α3(1) = 0.003496
P(H|start) * P(3|H)
0.8 * 0.4
P(C|start) * P(3|C)
0.2 * 0.1
P(a2F)
P(a1F)
3
1
3
O1
O2
O3



$\alpha_T = 0.0146$
q_f
End
End
End
End
$\alpha_1(2) = 0.32$
$\alpha_2(2) = 0.0472$
$\alpha_3(2) = 0.0257$
$P(a_{2F})$
q_2
H
H
H
H
$P(a_{1F})$
$\alpha_1(1) = 0.02$
$\alpha_2(1) = 0.052$
$\alpha_3(1) = 0.003496$
q_1
C
C
C
C
P(H|start) * P(3|H)
0.8 * 0.4
P(C|start) * P(3|C)
0.2 * 0.1
q_0
Start
Start
Start
Start
3
1
3
O_1
O_2
O_3

There are three fundamental kinds of questions that we can ask with an HMM:

1. ***Likelihood:* Given a sequence of states, what is the most likely observed sequence? *or* How likely is a given observation sequence?**

2. *Decoding:* Given an observation sequence and a fully-specified HMM, what is the most likely sequence of states to have produced that observation?

3. *Learning:* Given an observation sequence and a set of states, what are the likely transition and emission probabilities (A and B)?

Applications:

Part-of-speech tagging

Speech recognition (observed: MFCC, hidden: phoneme)

Bioinformatics (observed: nucleotide sequence, hidden: coding/non-coding region, etc.)

There are three fundamental kinds of questions that we can ask with an HMM:

1. *Likelihood*: Given a sequence of states, what is the most likely observed sequence? or How likely is a given observation sequence?

2. *Decoding:* Given an observation sequence and a fully-specified HMM, what is the most likely sequence of states to have produced that observation?

3. *Learning:* Given an observation sequence and a set of states, what are the likely transition and emission probabilities (A and B)?