CMPT 825: Natural Language Processing

Sequence Models

Spring 2020

Adapted from slides from Danqi Chen and Karthik Narasimhan
(Princeton COS 484)

Overview

- Hidden markov models (HMM)
- Viterbi algorithm
- Maximum entropy markov models (MEMM)

Sequence Tagging

Input: sequence of words; Output: sequence of labels

Input	British	left	waffles	on	Falkland	Islands
Output1	N	N	V	P	N	N
Output2	N	V	N	P	N	N
⋮						

N Noun, e.g. islands

V Verb, e.g. leave, left

P Preposition, e.g. on

What are POS tags

- Word classes or syntactic categories
  - Reveal useful information about a word (and its neighbors!)

The/DT cat/NN sat/VBD on/IN the/DT mat/NN

British/NNP left/NN waffles/NNS on/IN Falkland/NNP Islands/NNP

The/DT old/NN man/VB the/DT boat/NN

Parts of Speech

- Different words have different functions

- Closed class: fixed membership, **function words**

  - e.g. prepositions (*in, on, of*), determiners (*the, a*)

- Open class: New words get added frequently

  - e.g. nouns (Twitter, Facebook), verbs (google), adjectives, adverbs

Penn Tree Bank tagset

Tag	Description	Example	Tag	Description	Example	Tag	Description	Example
CC	coordinating conjunction	*and, but, or*	PDT	predeterminer	*all, both*	VBP	verb non-3sg present	*eat*
CD	cardinal number	*one, two*	POS	possessive ending	*'s*	VBZ	verb 3sg pres	*eats*
DT	determiner	*a, the*	PRP	personal pronoun	*I, you, he*	WDT	wh-determ.	*which, that*
EX	existential 'there'	*there*	PRP$	possess. pronoun	*your, one's*	WP	wh-pronoun	*what, who*
FW	foreign word	*mea culpa*	RB	adverb	*quickly*	WP$	wh-possess.	*whose*
IN	preposition/ subordin-conj	*of, in, by*	RBR	comparative adverb	*faster*	WRB	wh-adverb	*how, where*
JJ	adjective	*yellow*	RBS	superlatv. adverb	*fastest*	$	dollar sign	*$*
JJR	comparative adj	*bigger*	RP	particle	*up, off*	#	pound sign	*#*
JJS	superlative adj	*wildest*	SYM	symbol	*+,%, &*	‘‘	left quote	‘ or “
LS	list item marker	*1, 2, One*	TO	"to"	*to*	’’	right quote	’ or ”
MD	modal	*can, should*	UH	interjection	*ah, oops*	(	left paren	[, (, {, <
NN	sing or mass noun	*llama*	VB	verb base form	*eat*	)	right paren	],), }, >
NNS	noun, plural	*llamas*	VBD	verb past tense	*ate*	,	comma	,
NNP	proper noun, sing.	*IBM*	VBG	verb gerund	*eating*	.	sent-end punc	. ! ?
NNPS	proper noun, plu.	*Carolinas*	VBN	verb past part.	*eaten*	:	sent-mid punc	: ; ... – -

Figure 8.1 Penn Treebank part-of-speech tags (including punctuation).

[45 tags]

(Marcus et al., 1993)

Other corpora: Brown, WSJ, Switchboard

Part of Speech Tagging

- Disambiguation task: each word might have different senses/functions
  - The/DT man/NN bought/VBD a/DT boat/NN
  - The/DT old/NN man/VB the/DT boat/NN

		WSJ	Brown
Types:			
Unambiguous	(1 tag)	44,432 (**86%**)	45,799 (**85%**)
Ambiguous	(2+ tags)	7,025 (**14%**)	8,050 (**15%**)
Tokens:			
Unambiguous	(1 tag)	577,421 (**45%**)	384,349 (**33%**)
Ambiguous	(2+ tags)	711,780 (**55%**)	786,646 (**67%**)

Figure 8.2 Tag ambiguity for word types in Brown and WSJ, using Treebank-3 (45-tag) tagging. Punctuation were treated as words, and words were kept in their original case.

Part of Speech Tagging

- Disambiguation task: each word might have different senses/functions
  - The/DT man/NN bought/VBD a/DT boat/NN
  - The/DT old/NN man/VB the/DT boat/NN

earnings growth took a **back/JJ** seat
a small building in the **back/NN**
a clear majority of senators **back/VBP** the bill
Dave began to **back/VB** toward the door
enable the country to buy **back/RP** about debt
I was twenty-one **back/RB** then

Some words have many functions!

A simple baseline

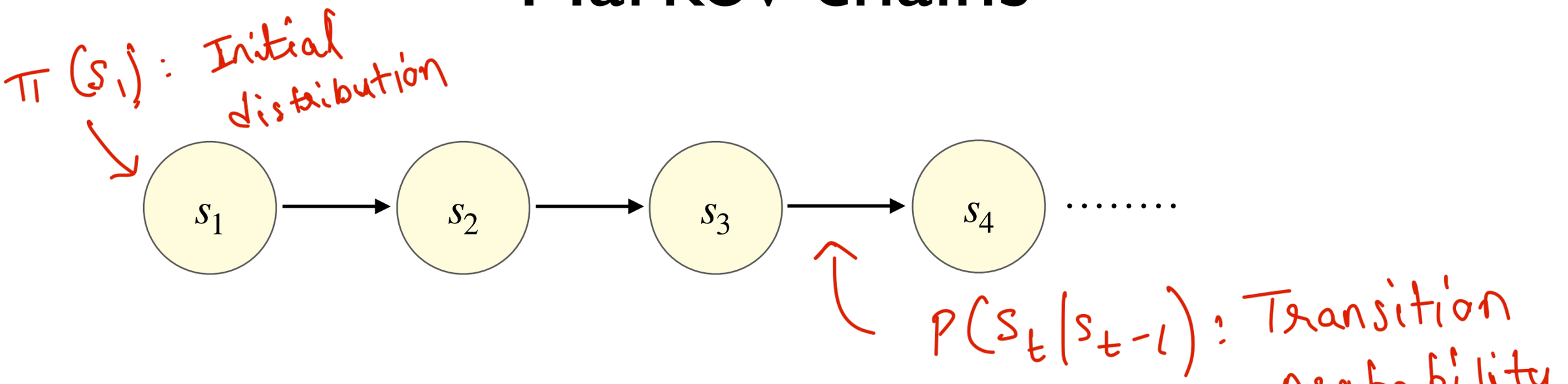
- Many words might be easy to disambiguate
- **Most frequent class:** Assign each token (word) to the class it occurred most in the training set. (e.g. man/NN)
- Accurately tags **92.34%** of word tokens on Wall Street Journal (WSJ)!
- State of the art ~ 97%
- Average English sentence ~ 14 words
    - Sentence level accuracies: 0.92^{14} = **31%** vs 0.97^{14} = **65%**
- POS tagging not solved yet!

Hidden Markov Models

Some observations

- The function (or POS) of a word depends on its context
  - The/DT old/NN man/VB the/DT boat/NN
  - The/DT old/JJ man/NN bought/VBD the/DT boat/NN
- Certain POS combinations are extremely unlikely
  - *<JJ, DT>* or *<DT, IN>*
- Better to make decisions on entire sequences instead of individual words (Sequence modeling!)

Markov chains



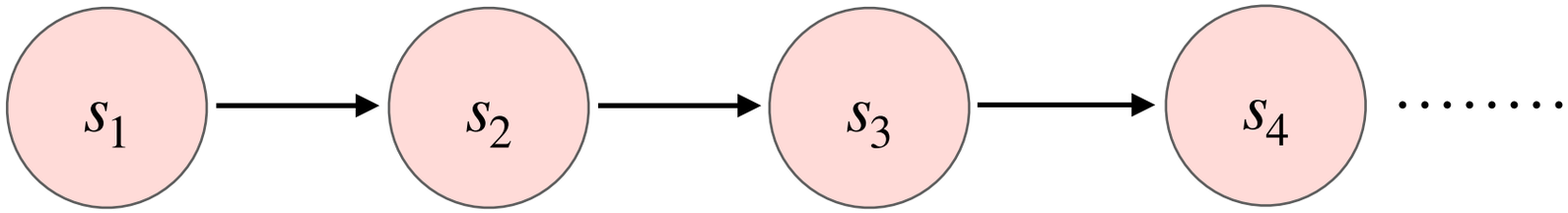
- Model probabilities of sequences of variables

- Each state can take one of K values ({1, 2, ..., K} for simplicity)

- Markov assumption: $P(s_t \mid s_{<t}) \approx P(s_t \mid s_{t-1})$

Where have we seen this before?

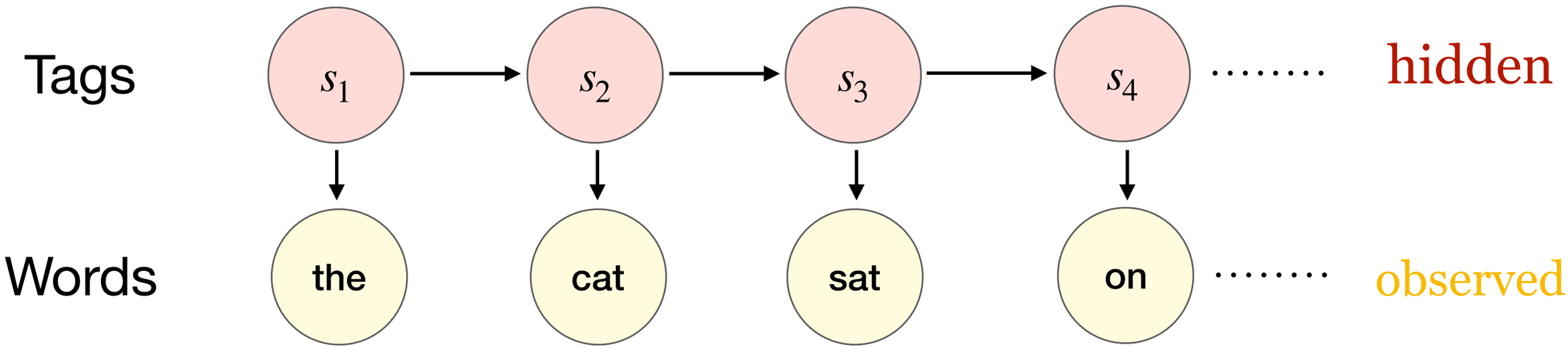
Markov chains



The/?? cat/?? sat/?? on/?? the/?? mat/??

- We don’t observe POS tags at test time

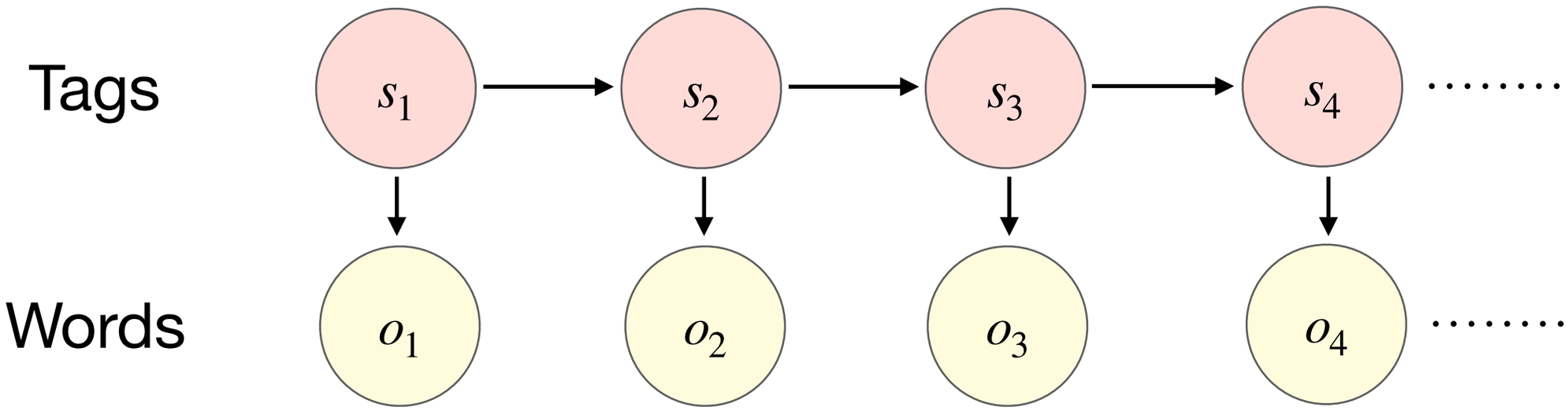
Hidden Markov Model (HMM)



The/?? cat/?? sat/?? on/?? the/?? mat/??

- We don’t observe POS tags at test time
- But we do observe the words!
- HMM allows us to *jointly reason* over both hidden and observed events.

Components of an HMM



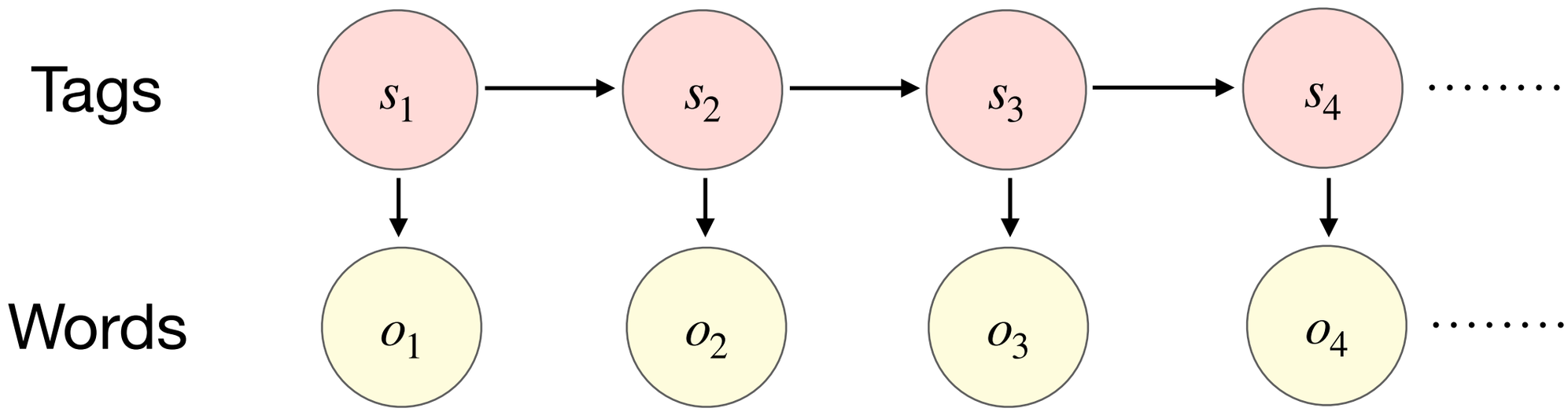
1. Set of states S = {1, 2, ..., K} and observations O

2. Initial state probability distribution $\pi(s_1)$

3. Transition probabilities $P(s_{t+1} \mid s_t)$

4. Emission probabilities $P(o_t \mid s_t)$

Assumptions



1. Markov assumption:

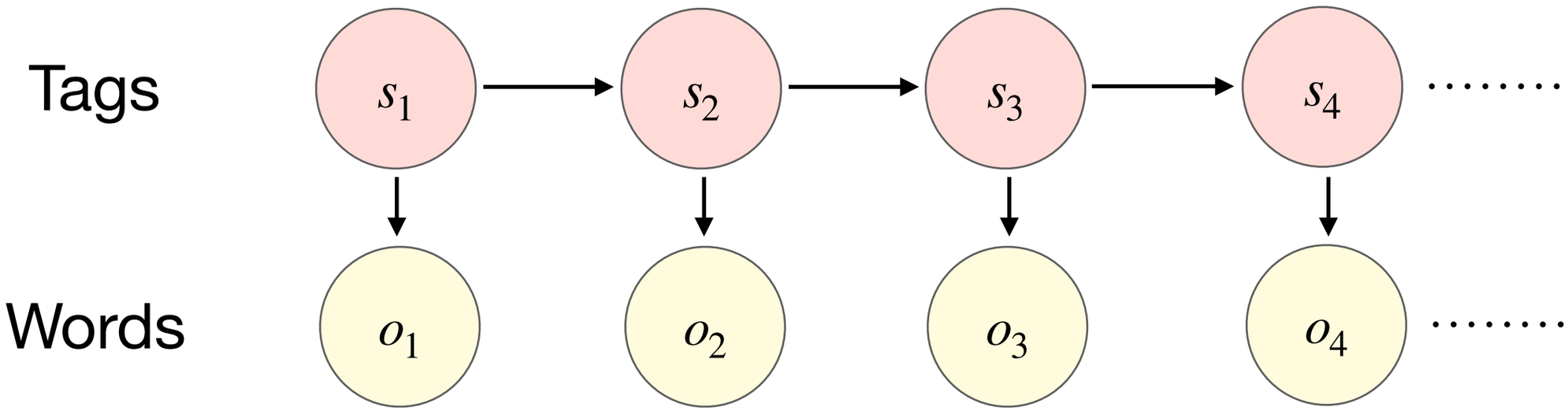
$$P(s_{t+1} \mid s_1, \ldots, s_t) = P(s_{t+1} \mid s_t)$$

2. Output independence:

$$P(o_t \mid s_1, \ldots, s_t) = P(o_t \mid s_t)$$

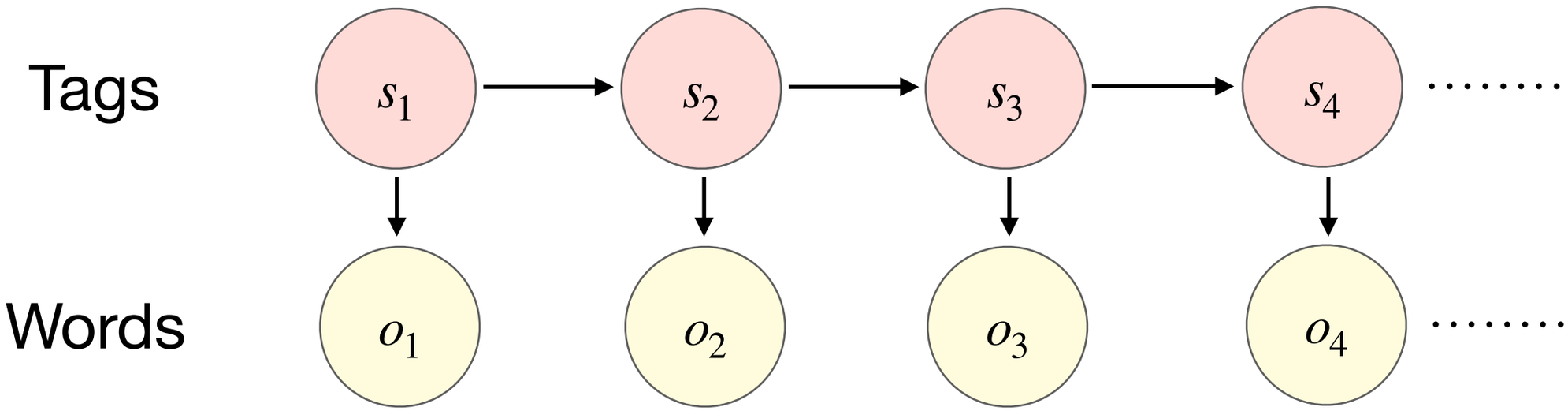
Which is a stronger assumption?

Sequence likelihood



$$P(S, O) = P(s_1, s_2 \cdots s_n, o_1, o_2 \cdots o_n)$$

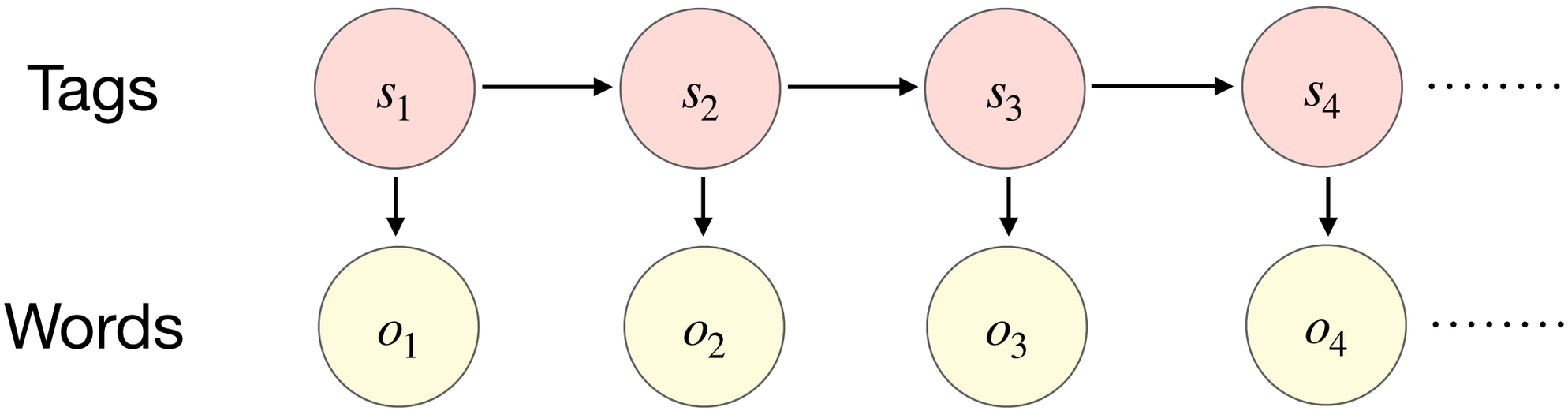
Sequence likelihood



$$P(S, O) = P(s_1, s_2 \dots s_n, o_1, o_2 \dots o_n)$$

$$= \pi(s_1) P(o_1 | s_1) \prod_{i=2}^{n} P(s_i, o_i | s_{i-1})$$

Sequence likelihood



$$P(S, O) = P(s_1, s_2 \dots s_n, o_1, o_2 \dots o_n)$$

$$= \pi(s_1)\, P(o_1 | s_1) \prod_{i=2}^{n} P(s_i, o_i | s_{i-1})$$

$$= \pi(s_1)\, P(o_1 | s_1) \prod_{i=2}^{n} P(s_i | s_{i-1}) P(o_i | s_i)$$

Learning

Training set:

1 Pierre/NNP Vinken/NNP ,/, 61/CD years/NNS old/JJ ,/, will/MD join/VB the/DT board/NN as/IN a/DT nonexecutive/JJ director/NN Nov./NNP 29/CD ./.

2 Mr./NNP Vinken/NNP is/VBZ chairman/NN of/IN Elsevier/NNP N.V./NNP ,/, the/DT Dutch/NNP publishing/VBG group/NN ./.

3 Rudolph/NNP Agnew/NNP ,/, 55/CD years/NNS old/JJ and/CC chairman/NN of/IN Consolidated/NNP Gold/NNP Fields/NNP PLC/NNP ,/, was/VBD named/VBN a/DT nonexecutive/JJ director/NN of/IN this/DT British/JJ industrial/JJ conglomerate/NN ./.

...

38,219 It/PRP is/VBZ also/RB pulling/VBG 20/CD people/NNS out/IN of/IN Puerto/NNP Rico/NNP ,/, who/WP were/VBD helping/VBG Huricane/NNP Hugo/NNP victims/NNS ,/, and/CC sending/VBG them/PRP to/TO San/NNP Francisco/NNP instead/RB ./.

Learning

Training set:

1 Pierre/NNP Vinken/NNP ,/, 61/CD year
join/VB the/DT board/NN as/IN a/DT no
Nov./NNP 29/CD ./.
2 Mr./NNP Vinken/NNP is/VBZ chairman
N.V./NNP ,/, the/DT Dutch/NNP publish
3 Rudolph/NNP Agnew/NNP ,/, 55/CD ye
chairman/NN of/IN Consolidated/NNP Go
,/, was/VBD named/VBN a/DT nonexecut
this/DT British/JJ industrial/JJ conglomer
...
38,219 It/PRP is/VBZ also/RB pulling/VB
of/IN Puerto/NNP Rico/NNP ,/, who/WP
Huricane/NNP Hugo/NNP victims/NNS ,/
them/PRP to/TO San/NNP Francisco/NN

Maximum likelihood estimate:

$$P(s_i \mid s_j) = \frac{C(s_j, s_i)}{C(s_j)}$$

$$P(o \mid s) = \frac{C(s, o)}{C(s)}$$

Example: POS tagging

the/?? cat/?? sat/?? on/?? the/?? mat/??

$\pi(DT) = 0.8$ s_{t+1} o_t

s_t	DT	NN	IN	VBD
DT	0.5	0.8	0.05	0.1
NN	0.05	0.2	0.15	0.6
IN	0.5	0.2	0.05	0.25
VBD	0.3	0.3	0.3	0.1

	the	cat	sat	on	mat
DT	0.5	0	0	0	0
NN	0.01	0.2	0.01	0.01	0.2
IN	0	0	0	0.4	0
VBD	0	0.01	0.1	0.01	0.01

Example: POS tagging

the/?? cat/?? sat/?? on/?? the/?? mat/??

$\pi(DT) = 0.8$ s_{t+1} o_t

	DT	NN	IN	VBD
DT	0.5	0.8	0.05	0.1
NN	0.05	0.2	0.15	0.6
IN	0.5	0.2	0.05	0.25
VBD	0.3	0.3	0.3	0.1

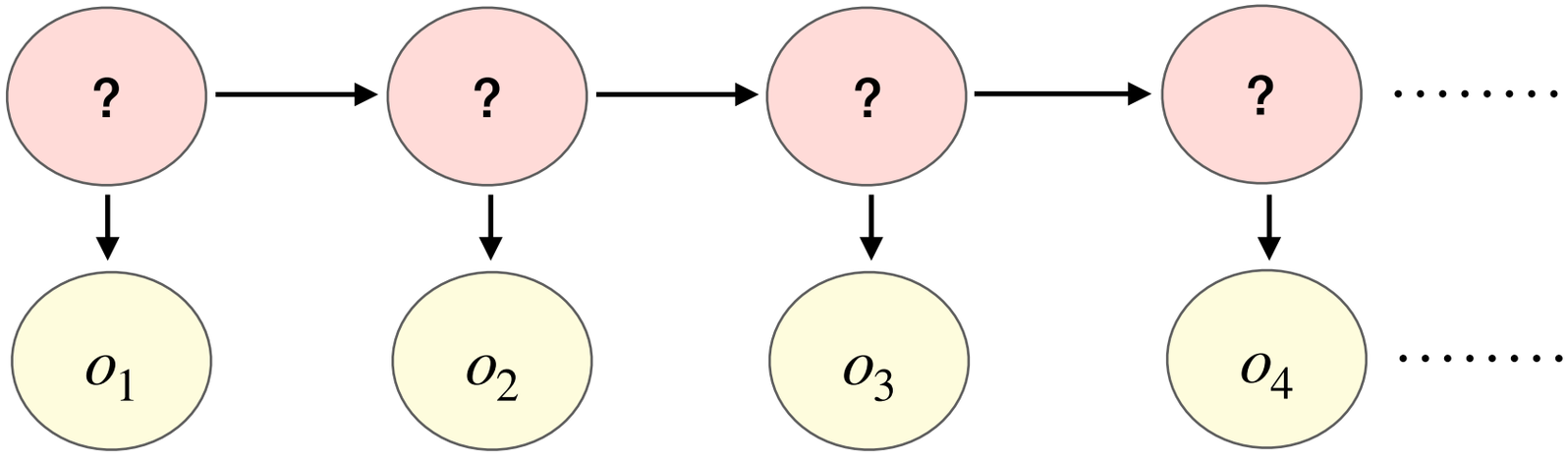
s_t

	the	cat	sat	on	mat
DT	0.5	0	0	0	0
NN	0.01	0.2	0.01	0.01	0.2
IN	0	0	0	0.4	0
VBD	0	0.01	0.1	0.01	0.01

P(the/DT, cat/NN, sat/VBD, on/IN, the/DT, mat/NN)

$= 1.84 * 10^{-5}$

Decoding with HMMs

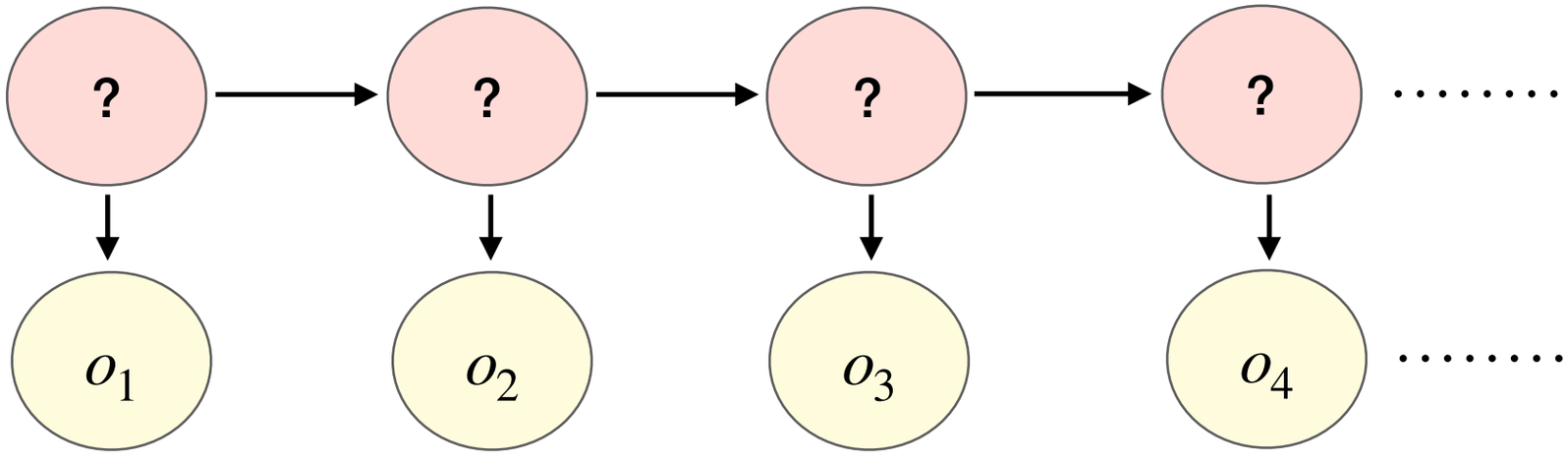


- **Task:** Find the most probable sequence of states $\langle s_1, s_2, \ldots, s_n \rangle$ given the observations $\langle o_1, o_2, \ldots, o_n \rangle$

$$\hat{S} = \underset{S}{\operatorname{argmax}}\, P(S|O) = \underset{S}{\operatorname{argmax}}\, \frac{P(S)\,P(O|S)}{P(O)}$$

[Bayes]

Decoding with HMMs

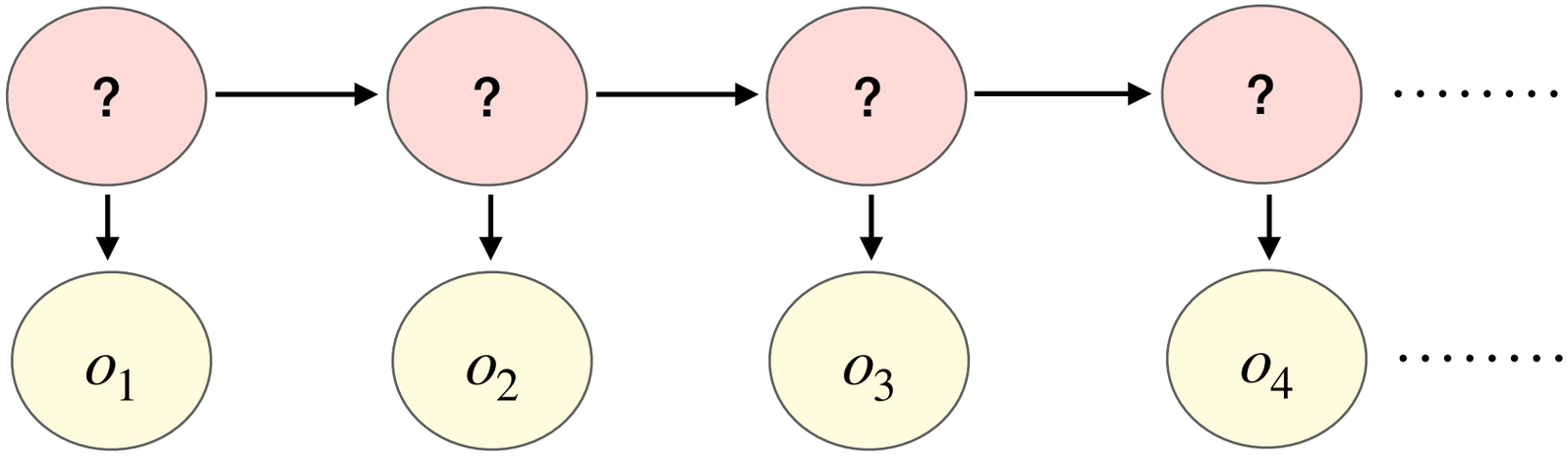


- **Task:** Find the most probable sequence of states $\langle s_1, s_2, \ldots, s_n \rangle$ given the observations $\langle o_1, o_2, \ldots, o_n \rangle$

$$\hat{S} = \underset{S}{\operatorname{argmax}}\, P(S|O) = \underset{S}{\operatorname{argmax}}\, \frac{P(S)\,P(O|S)}{P(O)} \quad \text{[Bayes]}$$

$$= \underset{S}{\operatorname{argmax}}\, P(S)\,P(O|S)$$

Decoding with HMMs



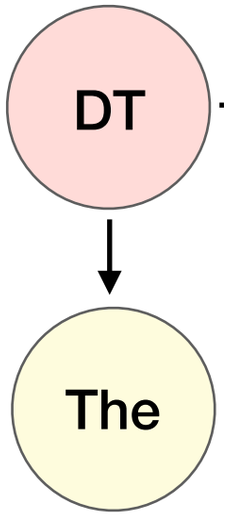
- **Task:** Find the most probable sequence of states $\langle s_1, s_2, \ldots, s_n \rangle$ given the observations $\langle o_1, o_2, \ldots, o_n \rangle$

$$\hat{S} = \underset{S}{\operatorname{argmax}}\ P(S)\,P(O|S)$$

$$= \underset{S}{\operatorname{argmax}} \prod_{i=1}^{n} P(s_i|s_{i-1})\ \underbrace{P(o_i|s_i)}_{\text{Emission}}$$

Transition

Greedy decoding

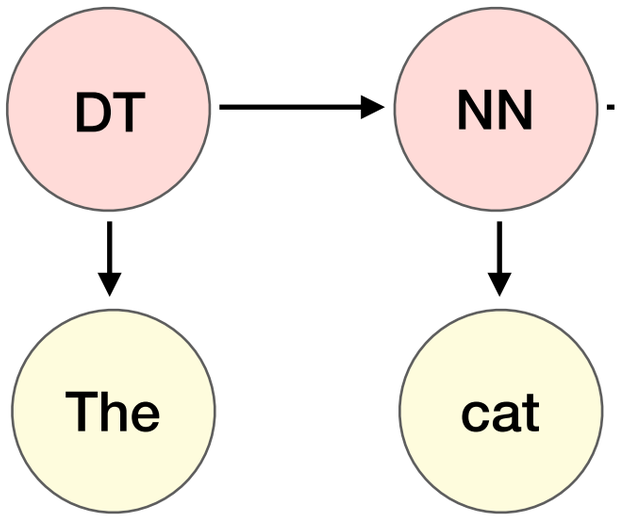


$$\underset{s}{\operatorname{argmax}}\ \pi(s_1 = s)\, P(\text{The} \mid s) = \text{`DT'}$$

$$\hat{s} = \underset{s}{\operatorname{argmax}}\ P(s)\, P(o \mid s)$$

$$= \underset{s}{\operatorname{argmax}} \prod_{i=1}^{n} \underbrace{P(s_i \mid s_{i-1})}_{\text{Transition}}\ \underbrace{P(o_i \mid s_i)}_{\text{Emission}}$$

Greedy decoding

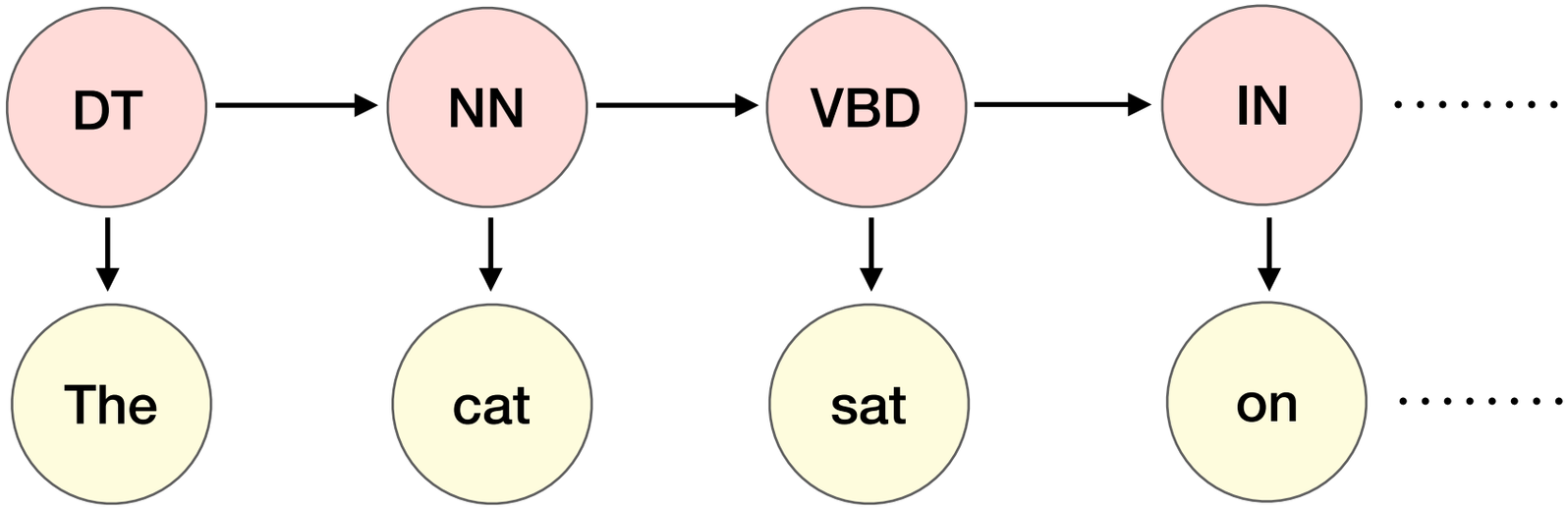


$$\underset{s}{\operatorname{argmax}}\ P(s_2 = s \mid DT)\, P(\text{cat} \mid s) = \text{'NN'}$$

$$\hat{s} = \underset{s}{\operatorname{argmax}}\ P(s)\, P(o \mid s)$$

$$= \underset{s}{\operatorname{argmax}} \prod_{i=1}^{n} \underbrace{P(s_i \mid s_{i-1})}_{\text{Transition}} \underbrace{P(o_i \mid s_i)}_{\text{Emission}}$$

Greedy decoding



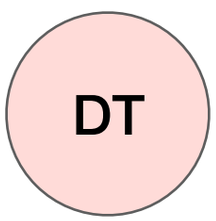
$$\forall t, \quad \hat{s}_{t+1} = \underset{s}{\operatorname{argmax}} \; P(s \mid \hat{s}_t) \, P(o_{t+1} \mid s)$$

- Not guaranteed to be optimal!
  - Local decisions

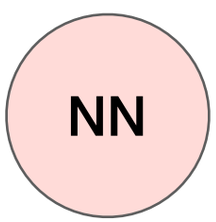
Viterbi decoding

- Use dynamic programming!
- Probability lattice, $M[T, K]$
  - T : Number of time steps
  - K : Number of states
- $M[i, j]$: Most probable sequence of states ending with state **j** at time **i**

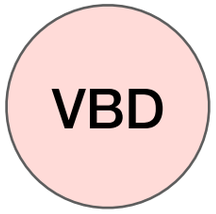
Viterbi decoding



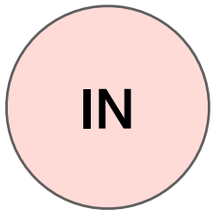
$$M[1,DT] = \pi(DT)\ P(\textbf{the}\,|\,DT)$$



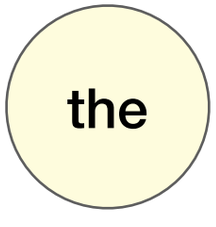
$$M[1,NN] = \pi(NN)\ P(\textbf{the}\,|\,NN)$$



$$M[1,VBD] = \pi(VBD)\ P(\textbf{the}\,|\,VBD)$$

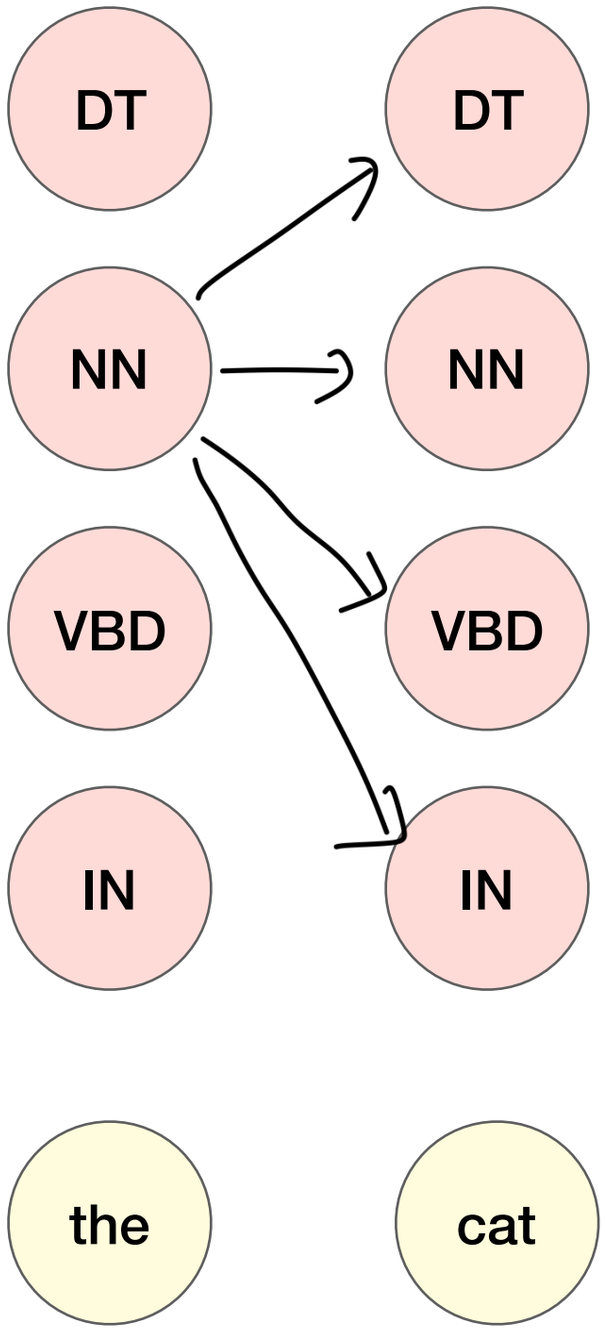


$$M[1,IN] = \pi(IN)\ P(\textbf{the}\,|\,IN)$$



Forward

Viterbi decoding



$$M[2,DT] = \max_k M[1,k]\ P(DT|k)\ P(\textbf{cat}|DT)$$

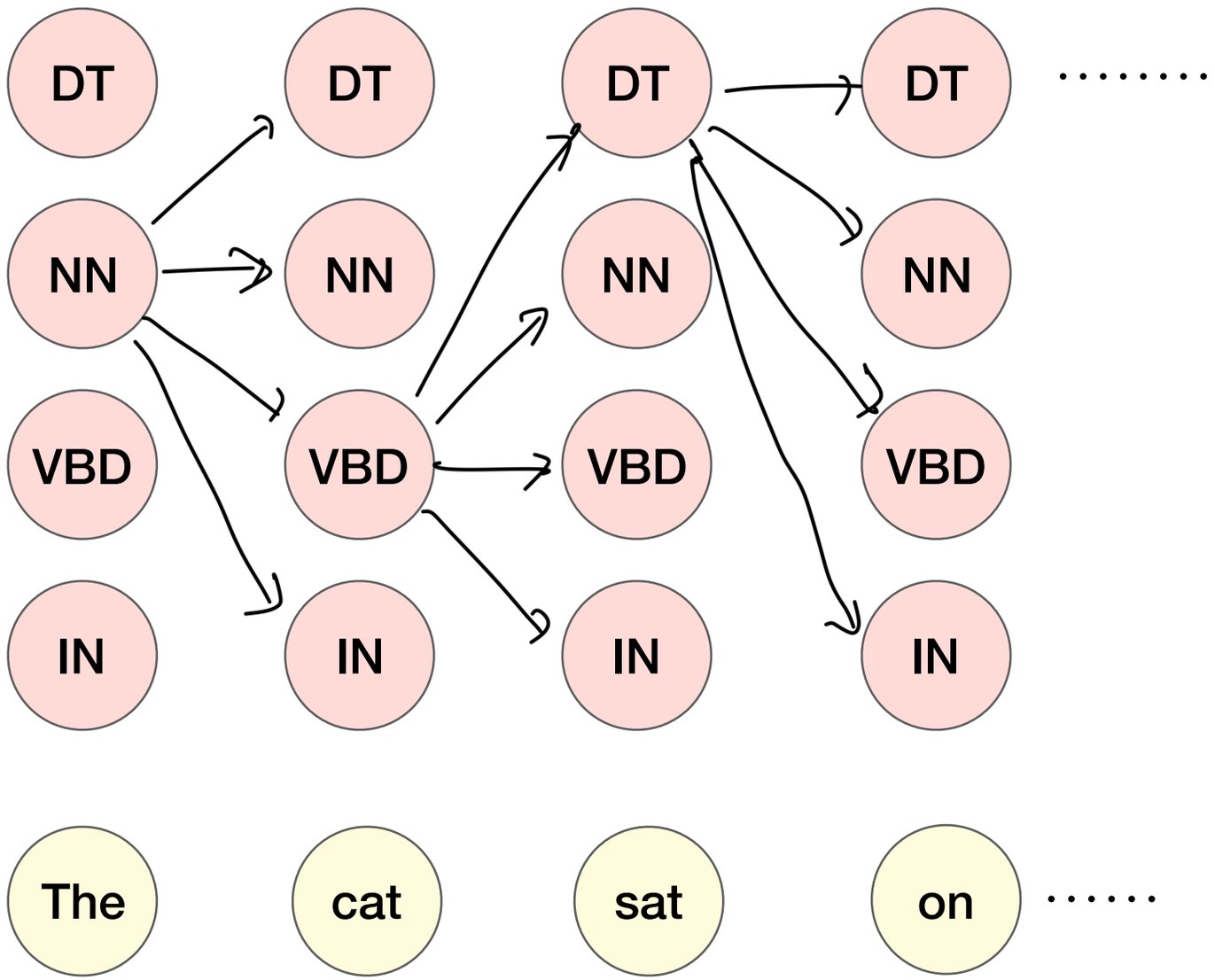
$$M[2,NN] = \max_k M[1,k]\ P(NN|k)\ P(\textbf{cat}|NN)$$

$$M[2,VBD] = \max_k M[1,k]\ P(VBD|k)\ P(\textbf{cat}|VBD)$$

$$M[2,IN] = \max_k M[1,k]\ P(IN|k)\ P(\textbf{cat}|IN)$$

Forward

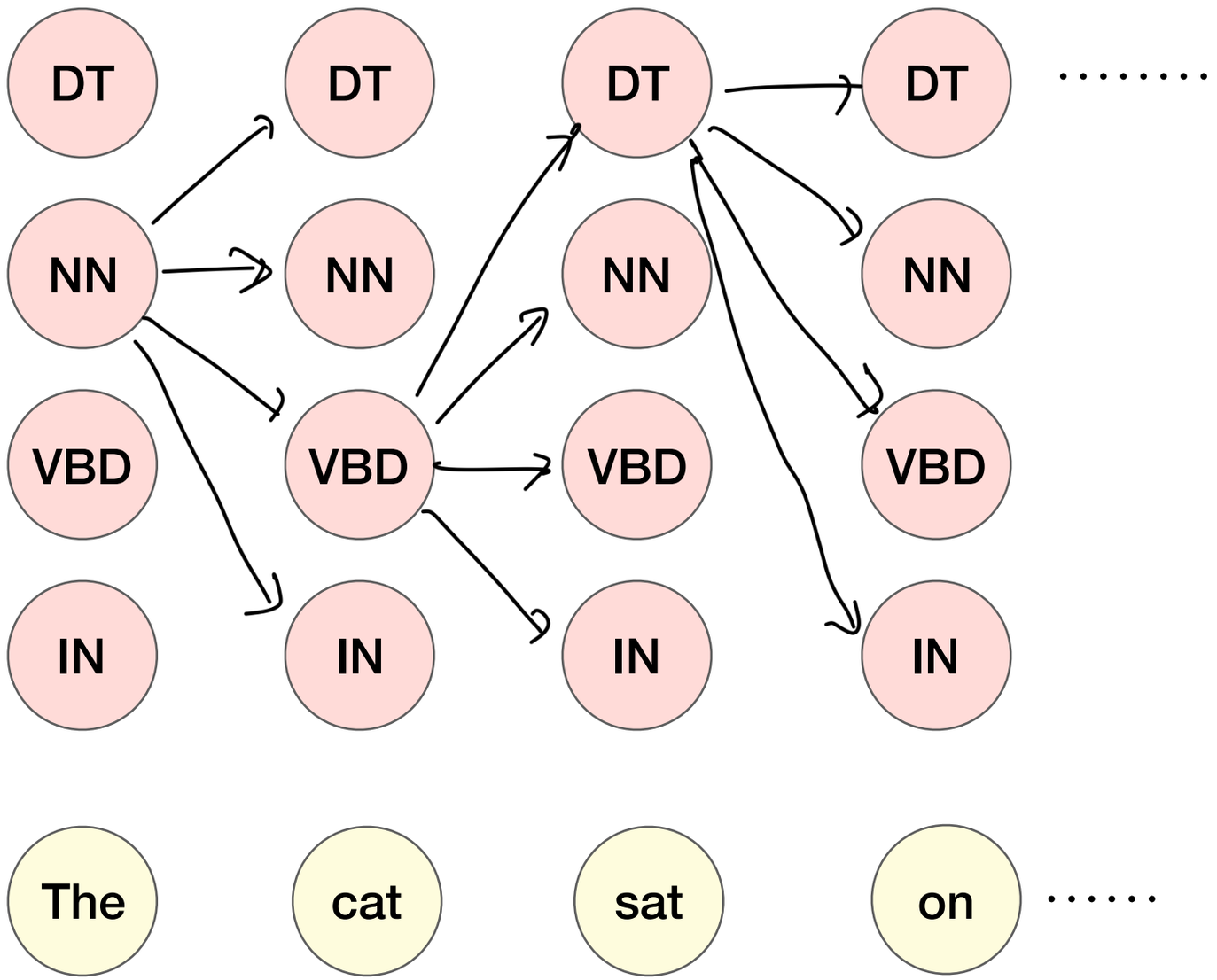
Viterbi decoding



$$M[i,j] = \max_k M[i-1,k]\ P(s_j \mid s_k)\ P(o_i \mid s_j) \quad 1 \le k \le K \quad 1 \le i \le n$$

Backward: Pick $\max_k M[n,k]$ and backtrack

Viterbi decoding



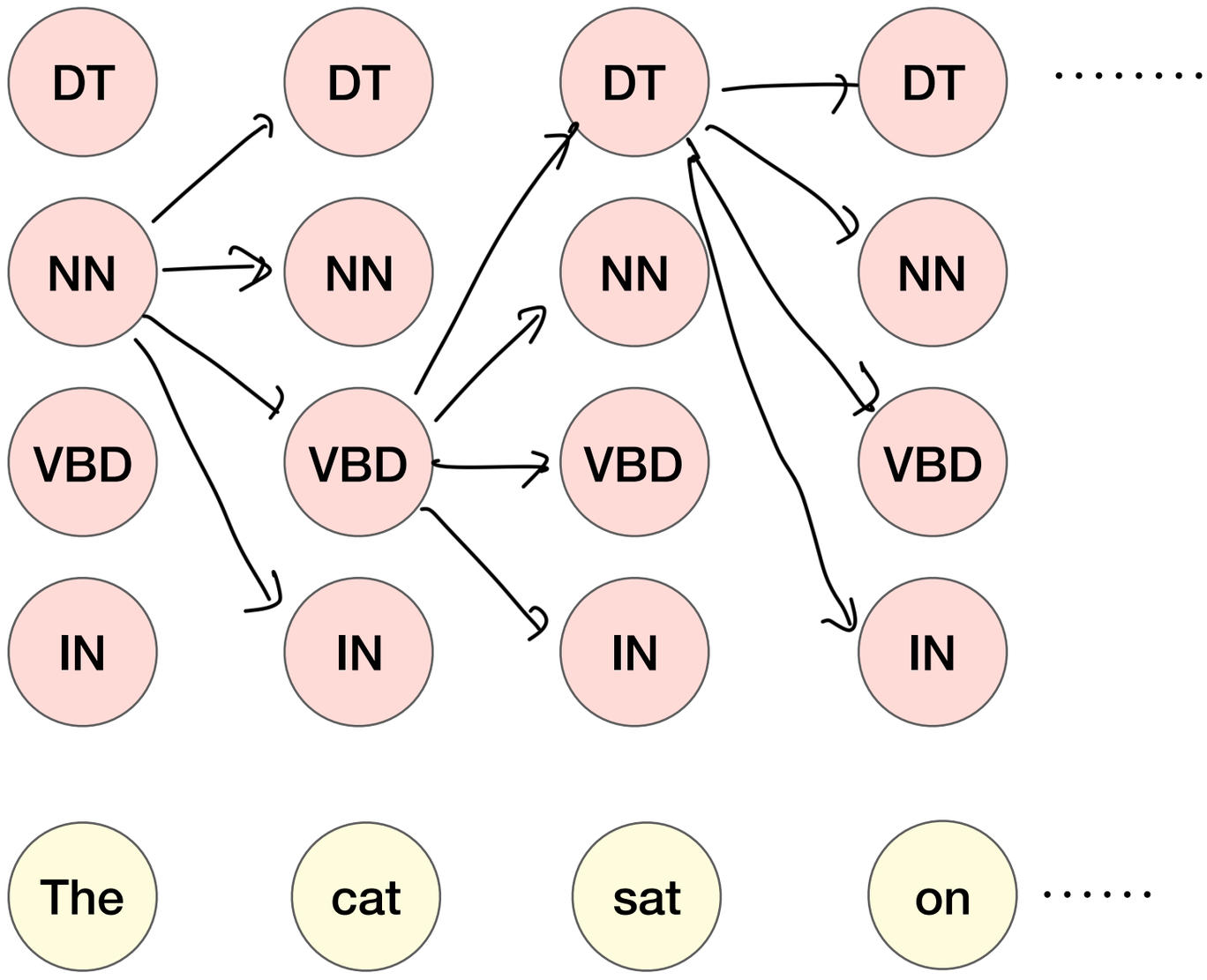
Time complexity?

$$M[i,j] = \max_k M[i-1,k]\ P(s_j \mid s_k)\ P(o_i \mid s_j) \quad 1 \le k \le K \quad 1 \le i \le n$$

Backward: Pick $\max_k M[n,k]$ and backtrack

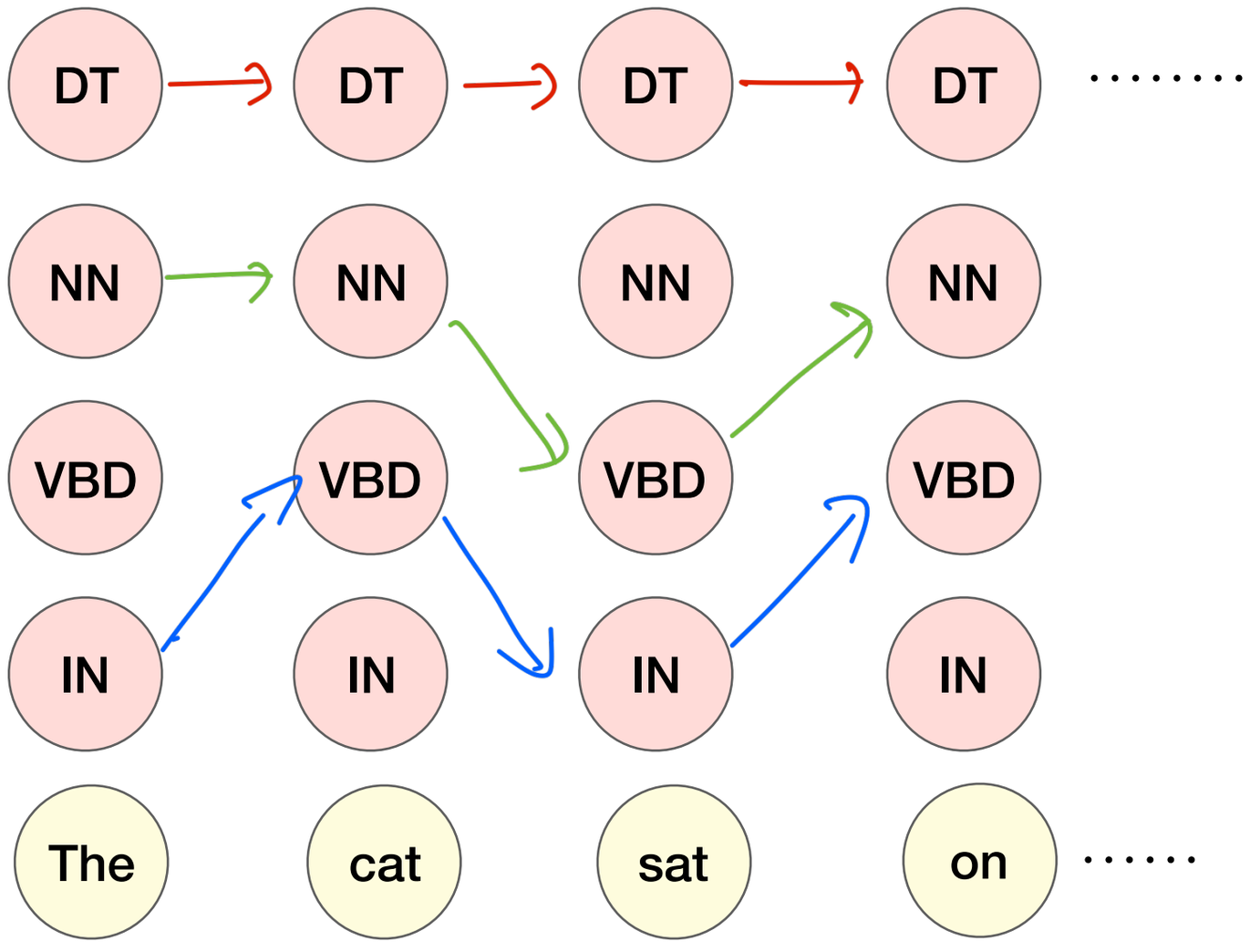
Beam Search

- If K (number of states) is too large, Viterbi is too expensive!



Beam Search

- If K (number of states) is too large, Viterbi is too expensive!



Many paths have very low likelihood!

Beam Search

- If K (number of states) is too large, Viterbi is too expensive!

- Keep a fixed number of hypotheses at each point
  - Beam width, β

Beam Search

- Keep a fixed number of hypotheses at each point

$\beta = 2$

DT $score = -4.1$

NN $score = -9.8$

VBD $score = -6.7$

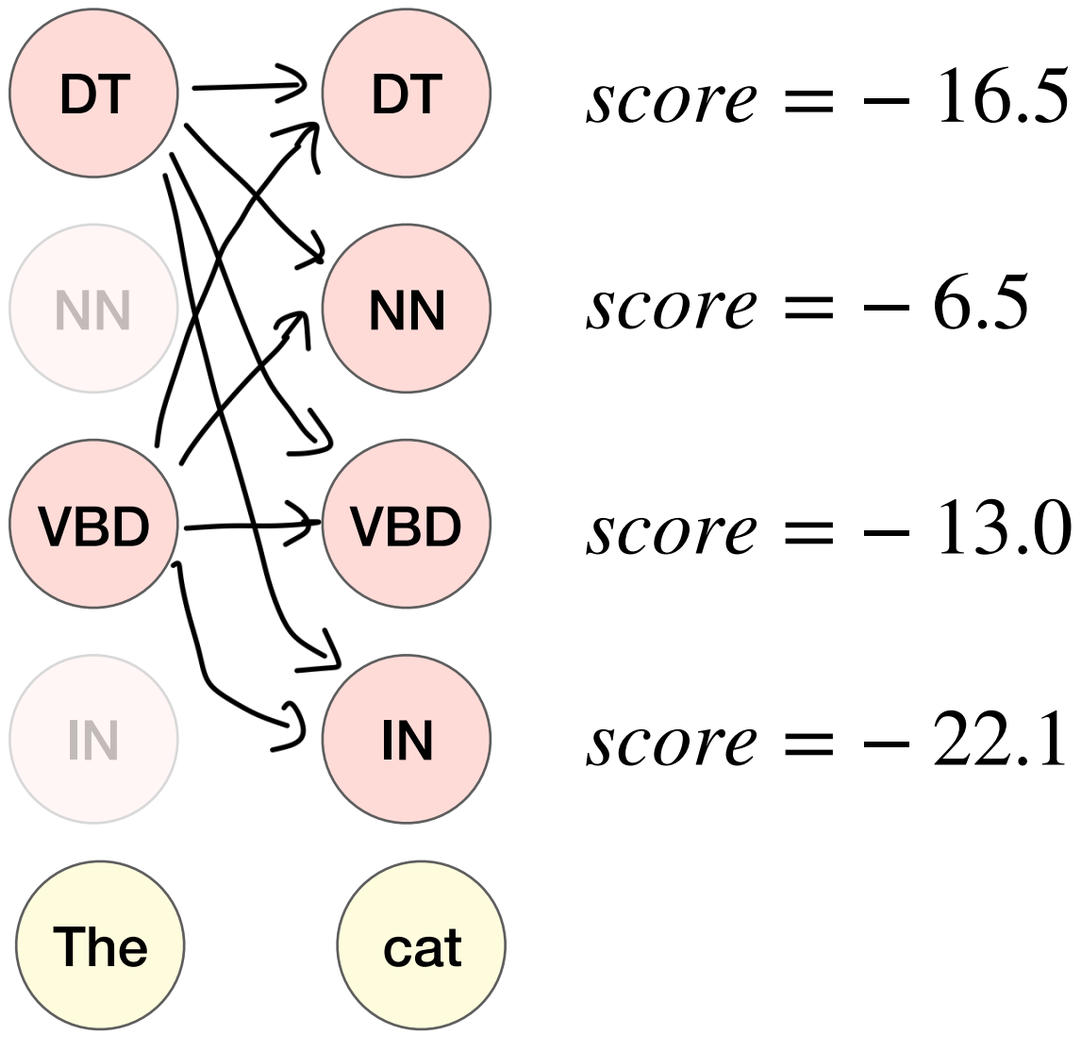
IN $score = -10.1$

The

Beam Search

- Keep a fixed number of hypotheses at each point

$\beta = 2$



$score = -16.5$

$score = -6.5$

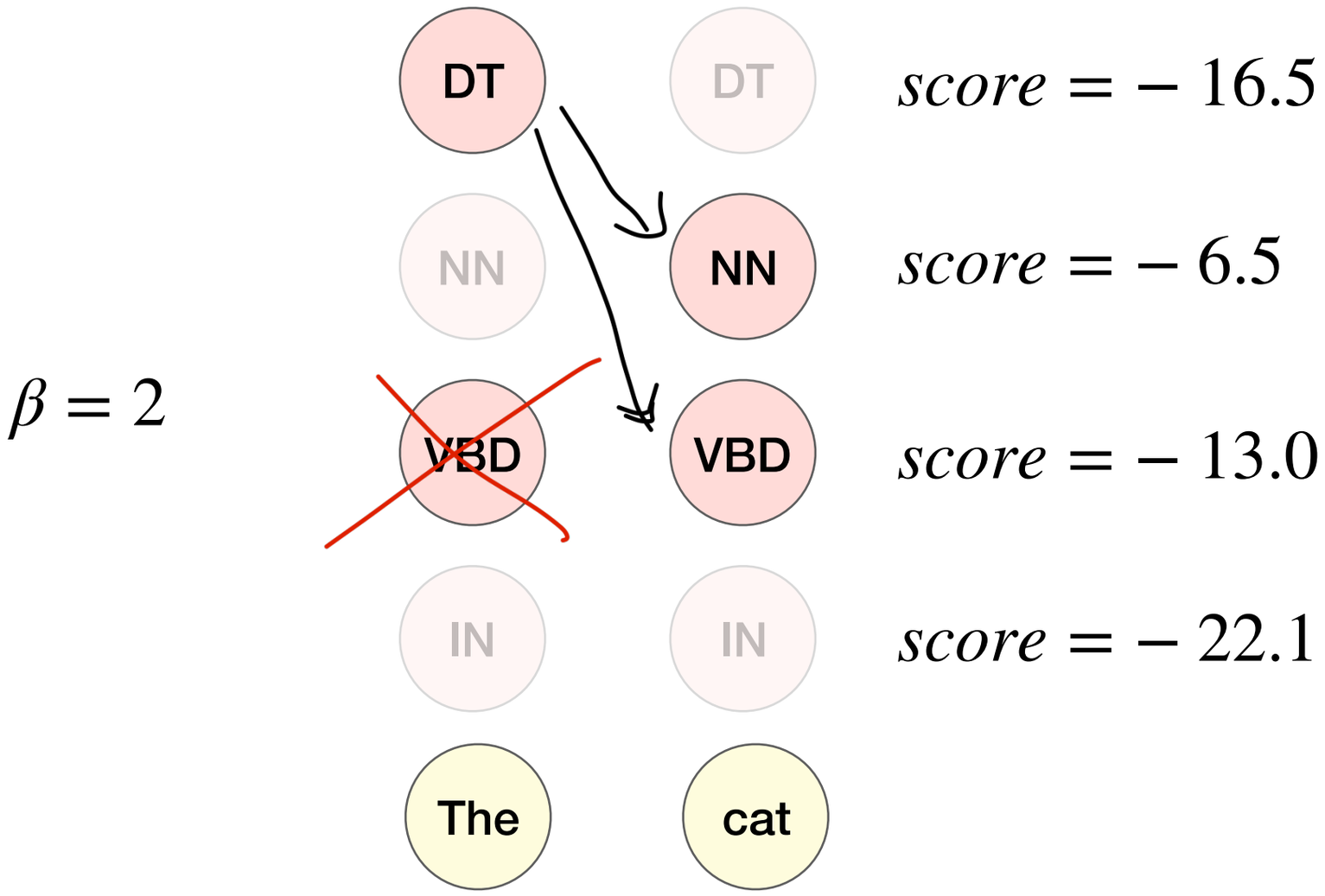
$score = -13.0$

$score = -22.1$

Step 1: Expand all partial sequences in current beam

Beam Search

- Keep a fixed number of hypotheses at each point

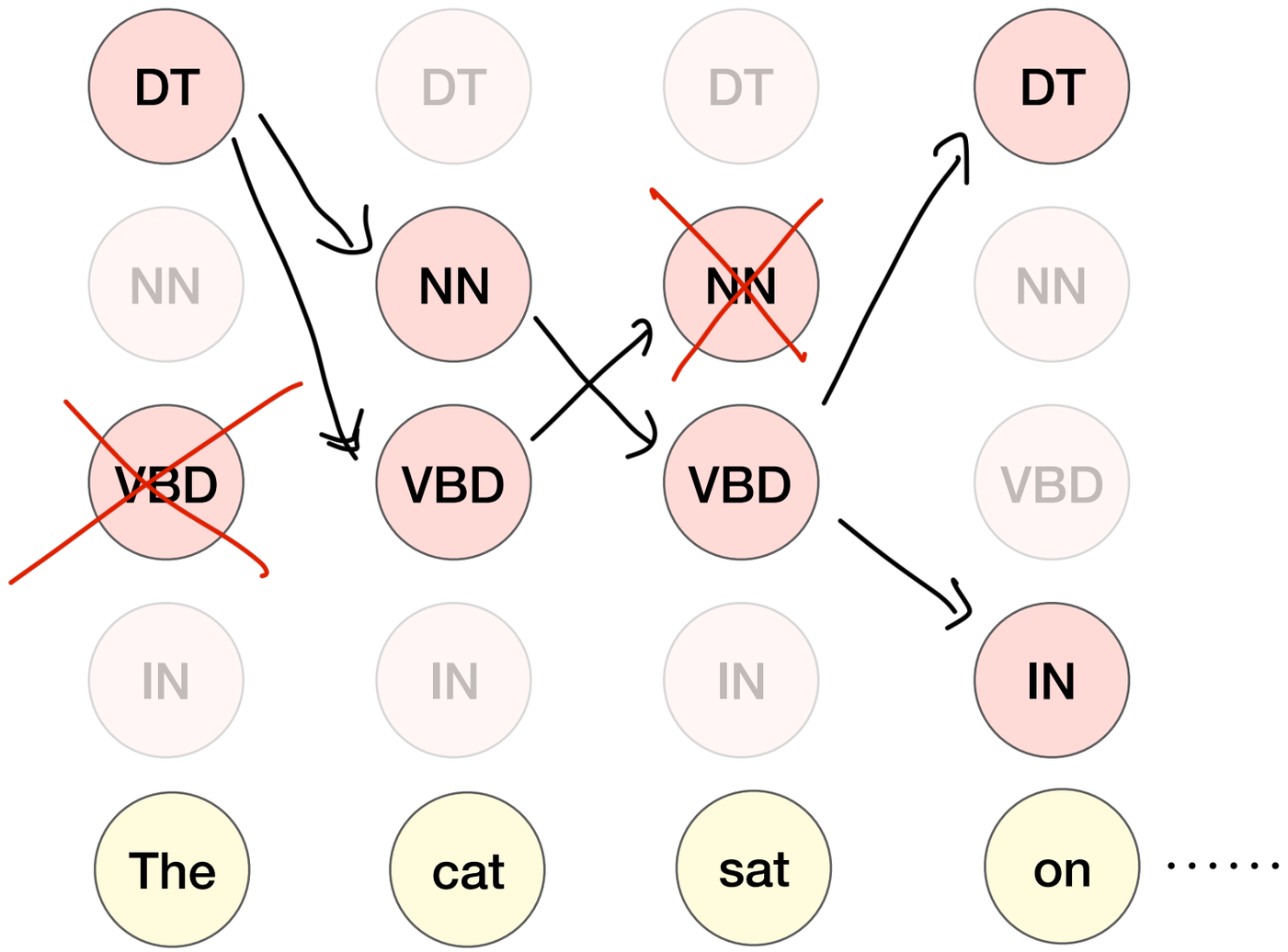


Step 2: Prune set back to top β sequences

Beam Search

- Keep a fixed number of hypotheses at each point

$\beta = 2$



Pick $\max_k M[n, k]$ from within beam and backtrack

Beam Search

- If K (number of states) is too large, Viterbi is too expensive!

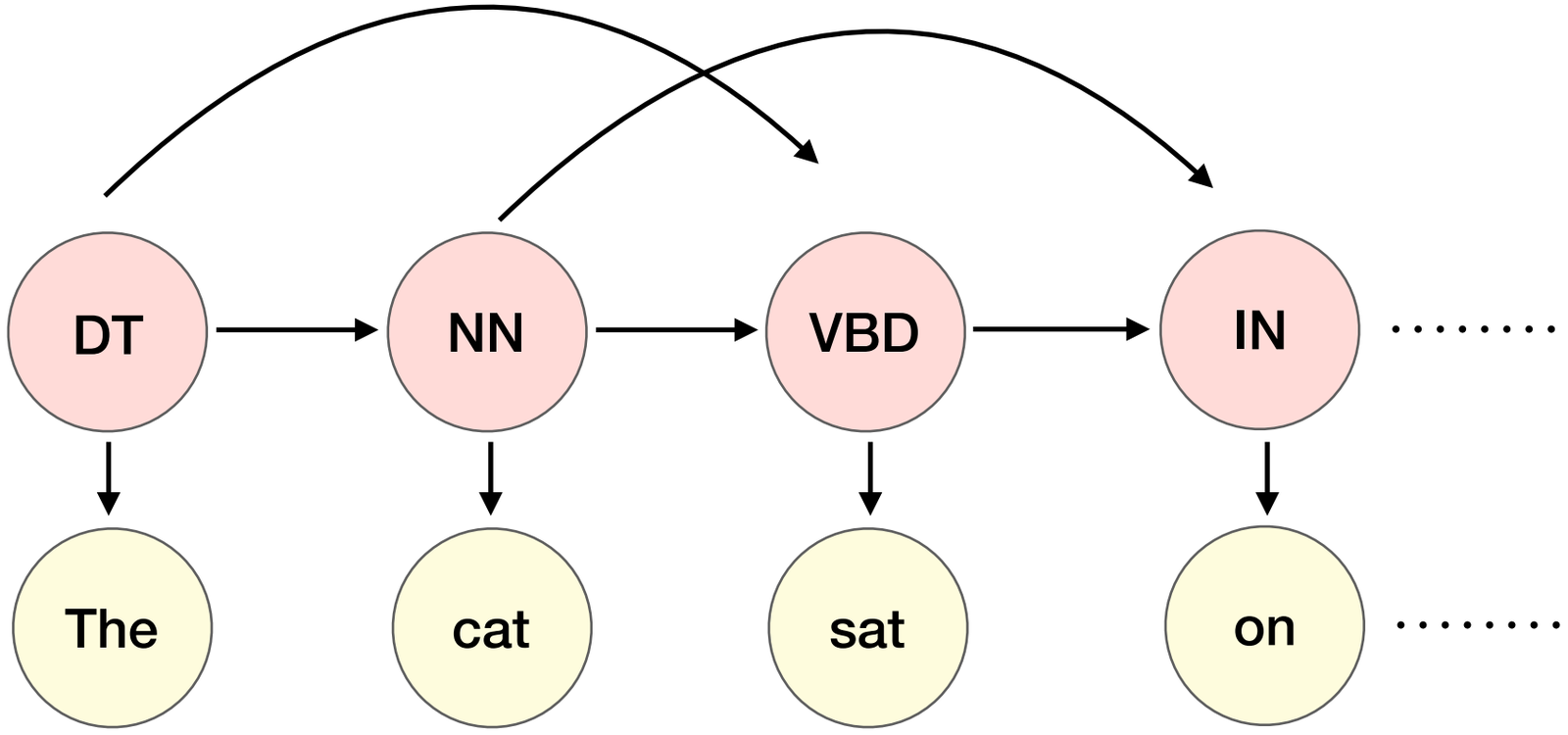
- Keep a fixed number of hypotheses at each point
  - Beam width, β

- Trade-off computation for (some) accuracy

Time complexity?

Beyond bigrams

- Real-world HMM taggers have more relaxed assumptions
- Trigram HMM: $P(s_{t+1} \mid s_1, s_2, \ldots, s_t) \approx P(s_{t+1} \mid s_{t-1}, s_t)$



Pros?

Cons?

Maximum Entropy Markov Models

Generative vs Discriminative

- HMM is a *generative* model
- Can we model $P(s_1, \ldots, s_n \mid o_1, \ldots, o_n)$ directly?

Generative

Naive Bayes:

$$P(c)P(d \mid c)$$

HMM:

$$P(s_1, \ldots, s_n)P(o_1, \ldots, o_n \mid s_1, \ldots, s_n)$$

Discriminative

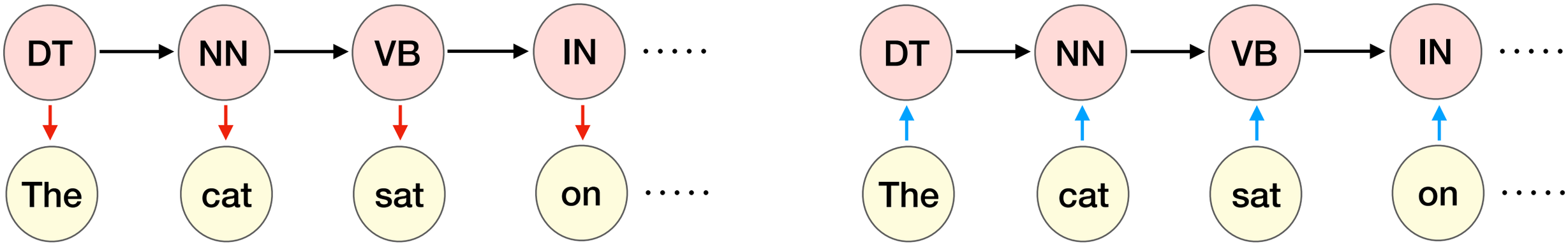
Logistic Regression:

$$P(c \mid d)$$

MEMM:

$$P(s_1, \ldots, s_n \mid o_1, \ldots, o_n)$$

MEMM



- Compute the posterior directly:

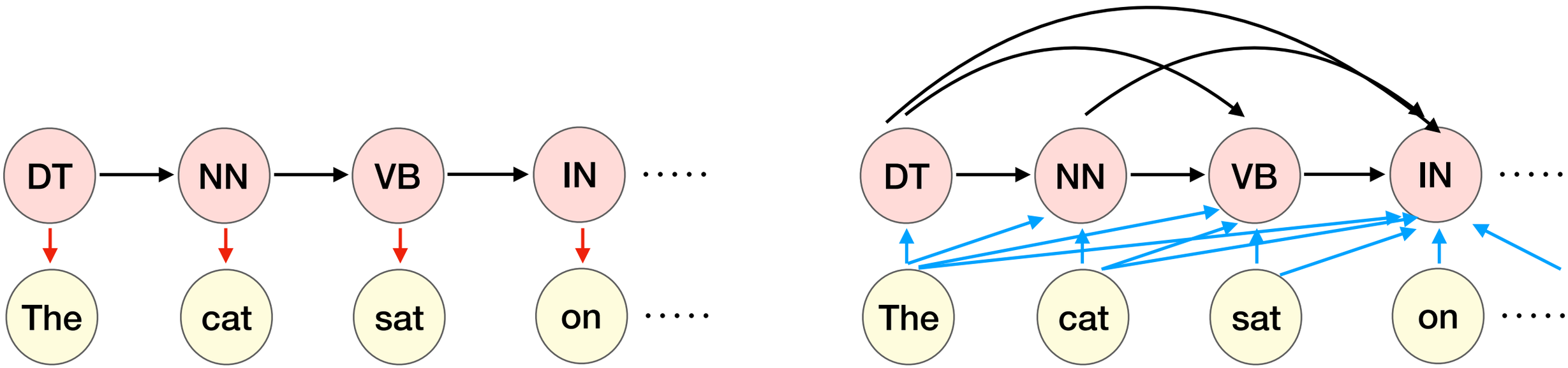
- $$\hat{S} = \arg\max_S P(S \mid O) = \arg\max_S \prod_i P(s_i \mid o_i, s_{i-1})$$

- Use features: $P(s_i \mid o_i, s_{i-1}) \propto \exp(w \cdot f(s_i, o_i, s_{i-1}))$

Features

weights

MEMM



- In general, we can use all observations and all previous states:

$$\hat{S} = \arg\max_{S} P(S \mid O) = \arg\max_{S} \prod_{i} P(s_i \mid o_n, o_{i-1}, \ldots, o_1, s_{i-1}, \ldots, s_1)$$

$$P(s_i \mid s_{i-1}, \ldots, s_1, O) \propto \exp(w \cdot f(s_i, s_{i-1}, \ldots, s_1, O)$$

Features in an MEMM

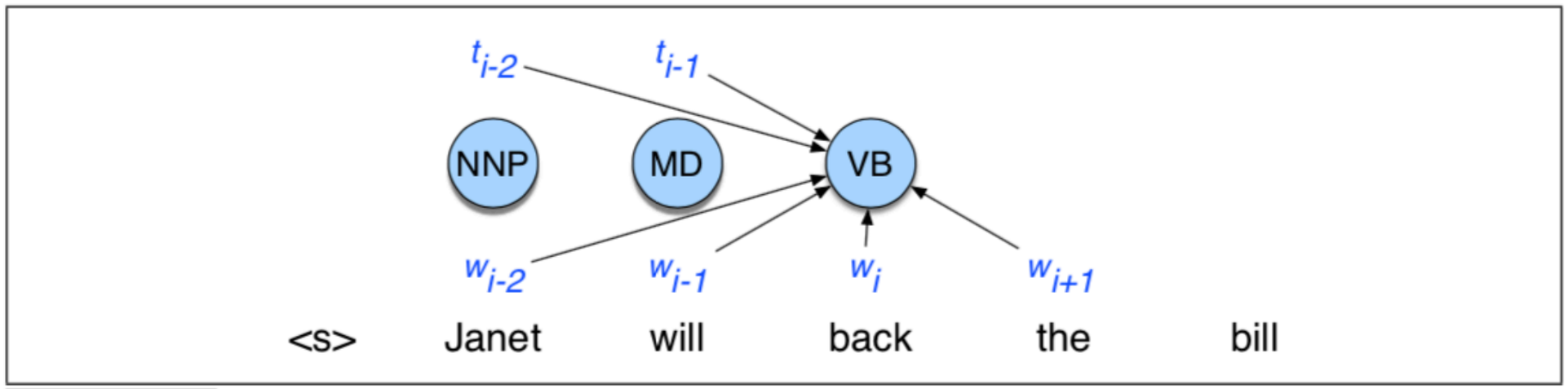


Figure 8.13 An MEMM for part-of-speech tagging showing the ability to condition on more features.

$$\langle t_i, w_{i-2}\rangle, \langle t_i, w_{i-1}\rangle, \langle t_i, w_i\rangle, \langle t_i, w_{i+1}\rangle, \langle t_i, w_{i+2}\rangle$$
$$\langle t_i, t_{i-1}\rangle, \langle t_i, t_{i-2}, t_{i-1}\rangle,$$
$$\langle t_i, t_{i-1}, w_i\rangle, \langle t_i, w_{i-1}, w_i\rangle \langle t_i, w_i, w_{i+1}\rangle,$$

Feature templates

t_i = VB and w_{i-2} = Janet
t_i = VB and w_{i-1} = will
t_i = VB and w_i = back
t_i = VB and w_{i+1} = the
t_i = VB and w_{i+2} = bill
t_i = VB and t_{i-1} = MD
t_i = VB and t_{i-1} = MD and t_{i-2} = NNP
t_i = VB and w_i = back and w_{i+1} = the

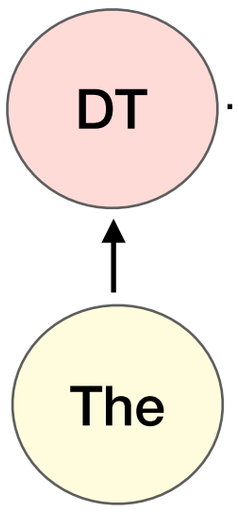
Features

MEMMs: Decoding

$$\hat{S} = \arg\max_{S} P(S \mid O) = \arg\max_{S} \Pi_i P(s_i \mid o_i, s_{i-1})$$

(assume features only on previous time step and current obs)

- Greedy decoding:



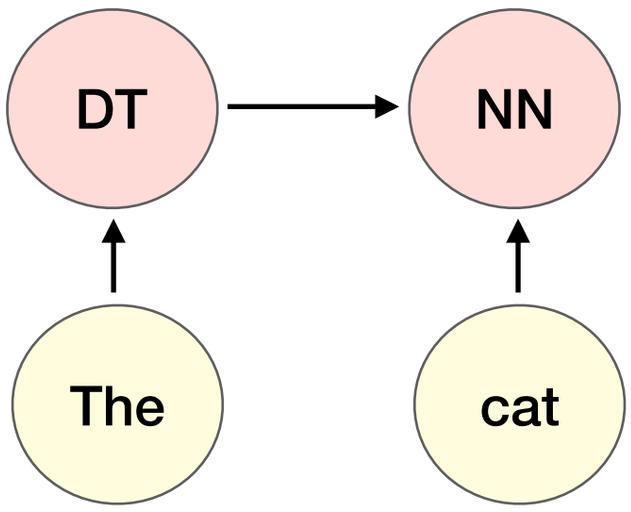
$$\hat{S}_1 = \underset{S}{\text{argmax}}\ P(S \mid \text{The})$$

$$= DT$$

MEMMs: Decoding

$$\hat{S} = \arg\max_{S} P(S \mid O) = \arg\max_{S} \Pi_i P(s_i \mid o_i, s_{i-1})$$

- Greedy decoding:



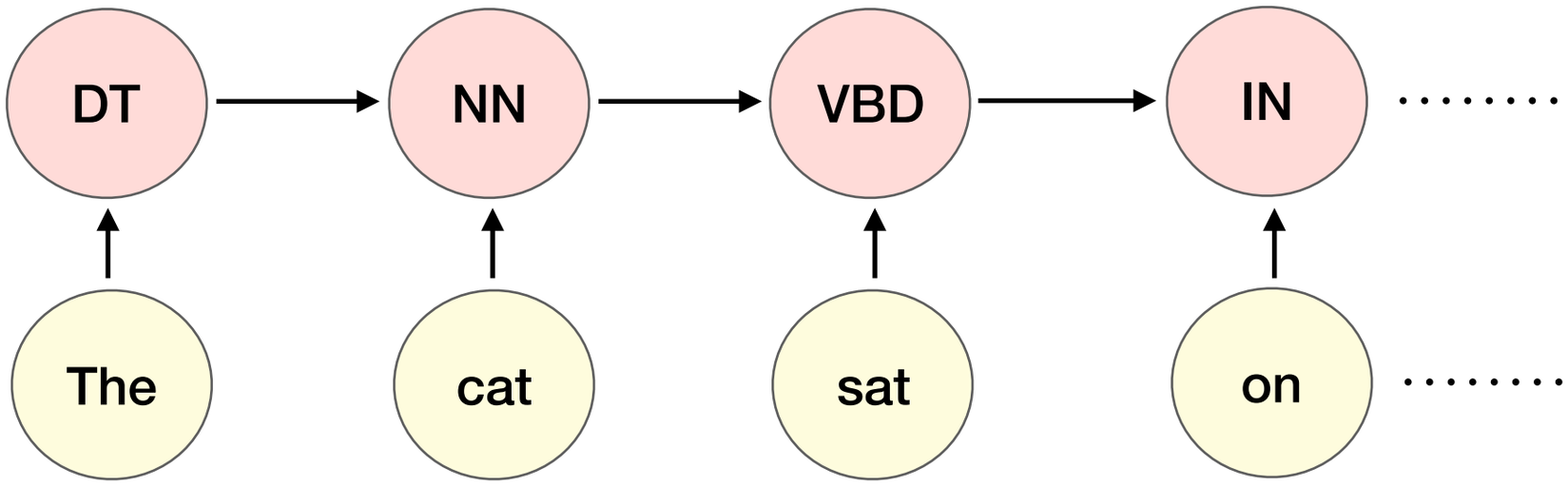
$$\hat{S}_2 = \arg\max_{S} P(S \mid \text{cat}, \text{DT})$$

$$= \text{NN}$$

MEMMs: Decoding

$$\hat{S} = \arg\max_{S} P(S \mid O) = \arg\max_{S} \Pi_i P(s_i \mid o_i, s_{i-1})$$

- Greedy decoding:



$$\forall t, \quad \hat{s}_{t+1} = \arg\max_{s} P(s \mid o_{t+1}, \hat{s}_t)$$

MEMMs: Decoding

$$\hat{S} = \arg\max_{S} P(S \mid O) = \arg\max_{S} \Pi_i P(s_i \mid o_i, s_{i-1})$$

- Greedy decoding

- Viterbi decoding:

$$M[i,j] = \max_{k} M[i-1,k]\ P(s_j \mid o_i, s_k) \quad 1 \le k \le K \quad 1 \le i \le n$$

DP Lattice

states

timesteps

MEMM: Learning

- Gradient descent: similar to logistic regression!

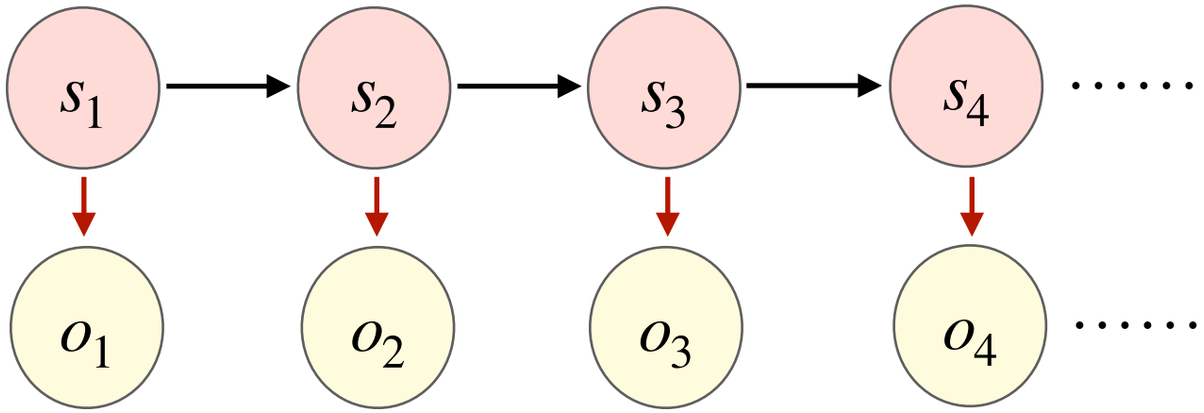
$$P(s_i \mid s_1, \ldots, s_{i-1}, O) \propto \exp(w \cdot f(s_1, \ldots, s_i, O))$$

- Given: pairs of (S, O) where each $S = \langle s_1, s_2, \ldots, s_n \rangle$

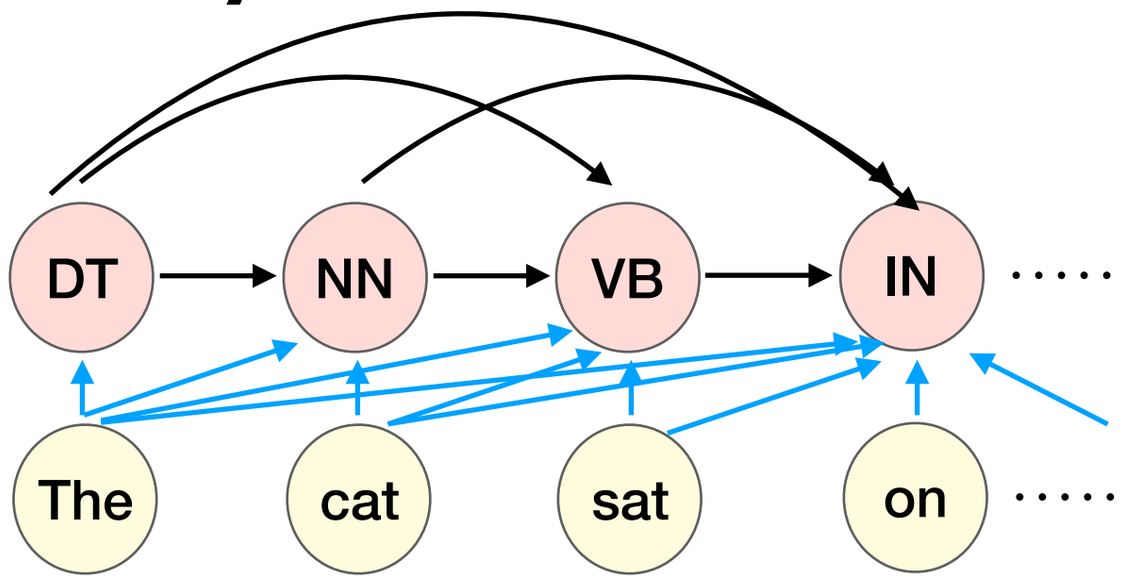
Loss for one sequence, $L = -\sum_i \log P(s_i \mid s_1, \ldots, s_{i-1}, O)$

- Compute gradients with respect to weights w and update

Bidirectionality



HMM

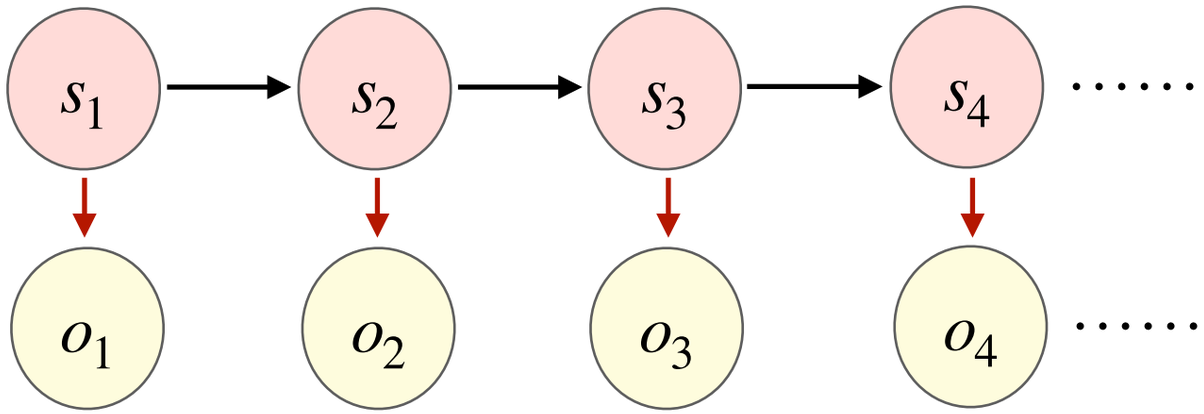


MEMM

Both HMM and MEMM assume left-to-right processing

Why can this be undesirable?

Bidirectionality



HMM

The/? old/? man/? the/? boat/?

$$P(JJ|DT)\,P(\textbf{old}|JJ)\,P(NN|JJ)\,P(\textbf{man}|NN)\,P(DT|NN)$$

$$P(NN|DT)\,P(\textbf{old}|NN)\,P(VB|NN)\,P(\textbf{man}|VB)\,P(DT|VB)$$

Observation bias

Stanford Parser

Please enter a sentence to be parsed:

The old man the boat

Language: English Sample Sentence Parse

Your query

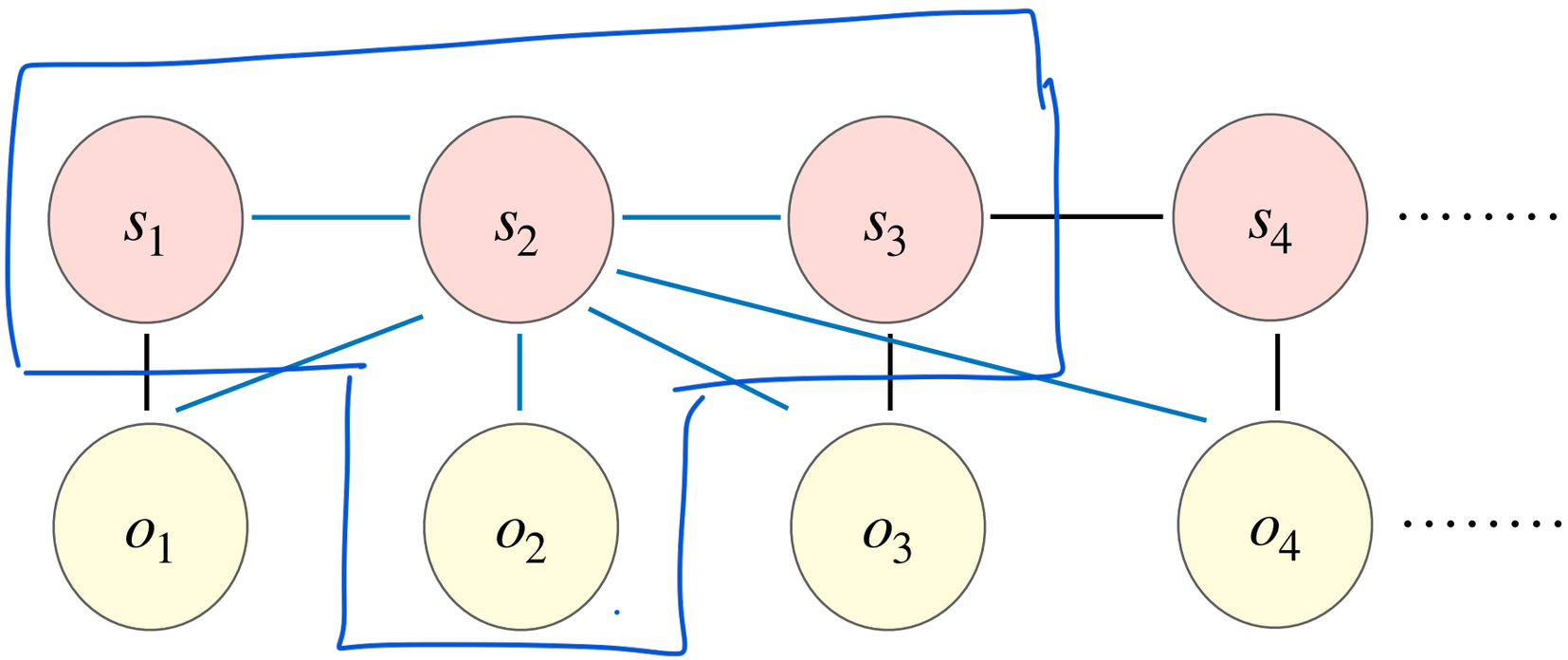
The old man the boat

Tagging

```
The/DT  old/JJ  man/NN  the/DT  boat/NN
```

Observation bias

Conditional Random Field (advanced)



- Compute log-linear functions over cliques
- Lesser independence assumptions
- Ex: $P(s_t \mid \text{everything else}) \propto \exp(w \cdot f(s_{t-1}, s_t, s_{t+1}, O))$