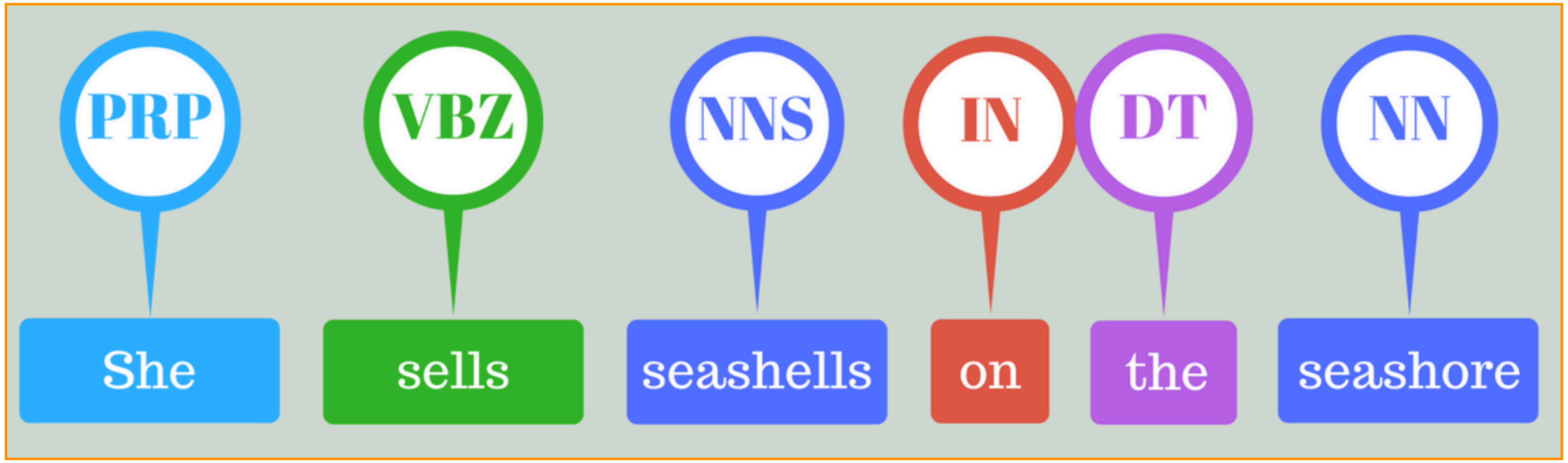
COS 484/584

Sequence Models - I

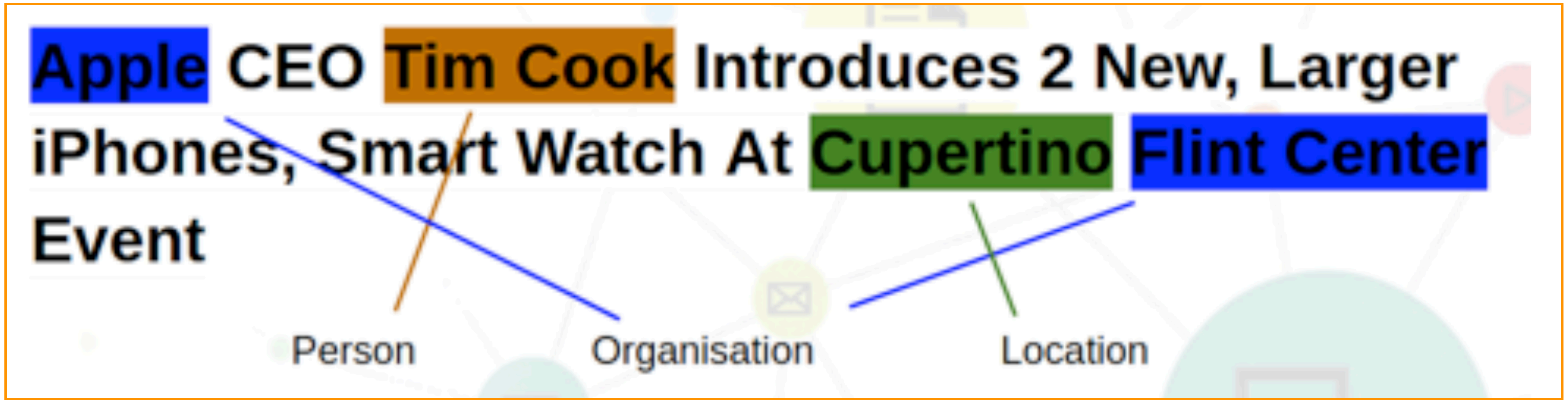
Spring 2021

Why model sequences?



Part of Speech tagging

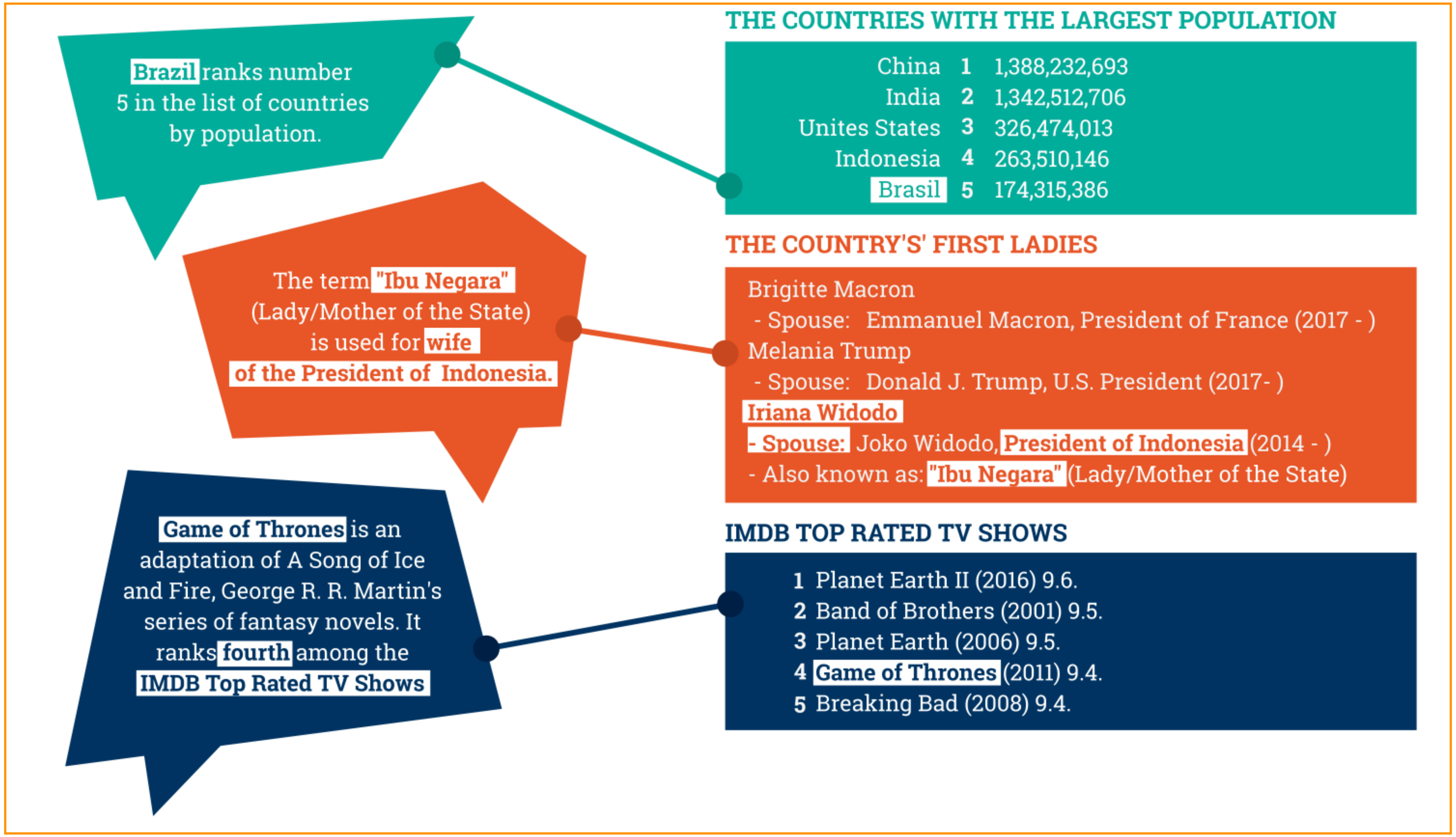
Why model sequences?



Apple CEO Tim Cook Introduces 2 New, Larger iPhones, Smart Watch At Cupertino Flint Center Event

Named Entity recognition

Why model sequences?



Brazil ranks number 5 in the list of countries by population.

The term "Ibu Negara" (Lady/Mother of the State) is used for wife of the President of Indonesia.

Game of Thrones is an adaptation of A Song of Ice and Fire, George R. R. Martin's series of fantasy novels. It ranks fourth among the IMDB Top Rated TV Shows

THE COUNTRIES WITH THE LARGEST POPULATION

China	1	1,388,232,693
India	2	1,342,512,706
Unites States	3	326,474,013
Indonesia	4	263,510,146
Brasil	5	174,315,386

THE COUNTRY'S' FIRST LADIES

Brigitte Macron
- Spouse: Emmanuel Macron, President of France (2017 -)

Melania Trump
- Spouse: Donald J. Trump, U.S. President (2017-)

Iriana Widodo
- Spouse: Joko Widodo, President of Indonesia (2014 -)
- Also known as: "Ibu Negara" (Lady/Mother of the State)

IMDB TOP RATED TV SHOWS

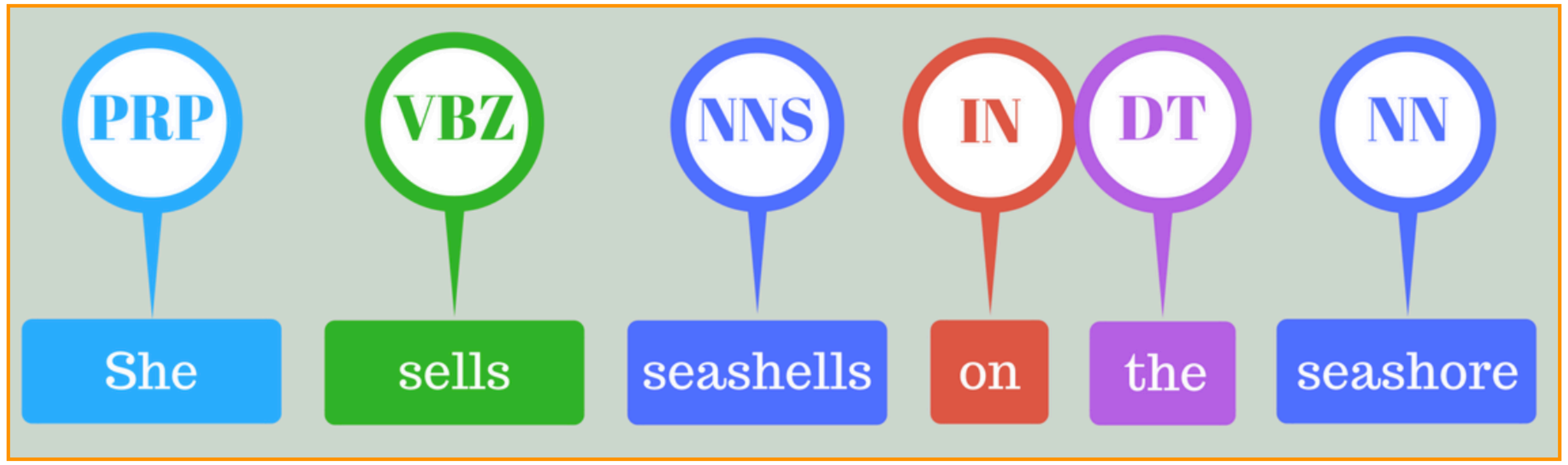
1 Planet Earth II (2016) 9.6.
2 Band of Brothers (2001) 9.5.
3 Planet Earth (2006) 9.5.
4 Game of Thrones (2011) 9.4.
5 Breaking Bad (2008) 9.4.

Information Extraction

Overview

- Hidden markov models (HMM)
- Viterbi algorithm

What are part of speech tags?



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

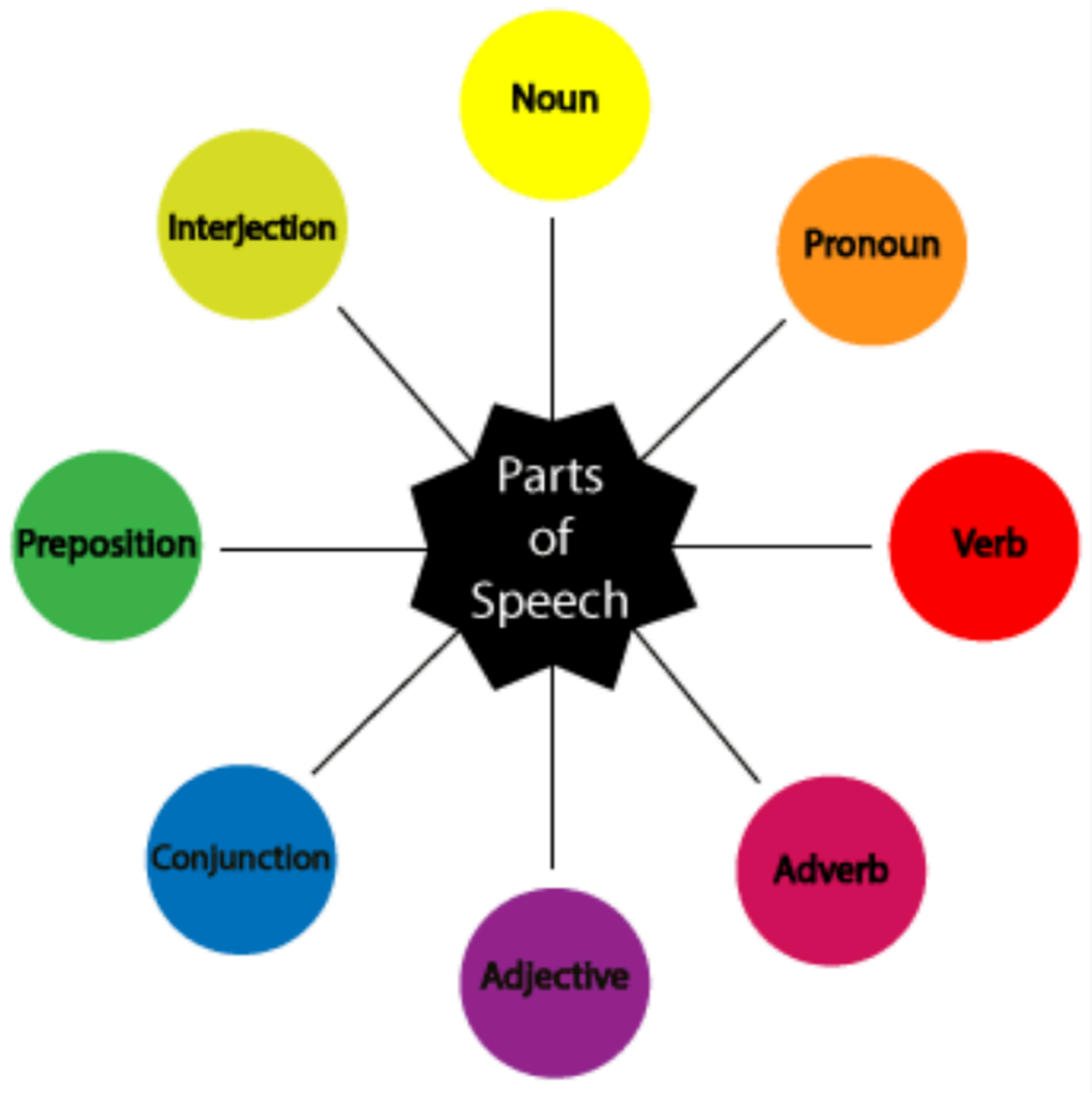
1. The/DT cat/NN sat/VBD on/IN the/DT mat/NN

2. Princeton/NNP is/VBZ in/IN New/NNP Jersey/NNP

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

Parts of Speech

- Different words have different functions
- Can be roughly divided into two classes
- **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



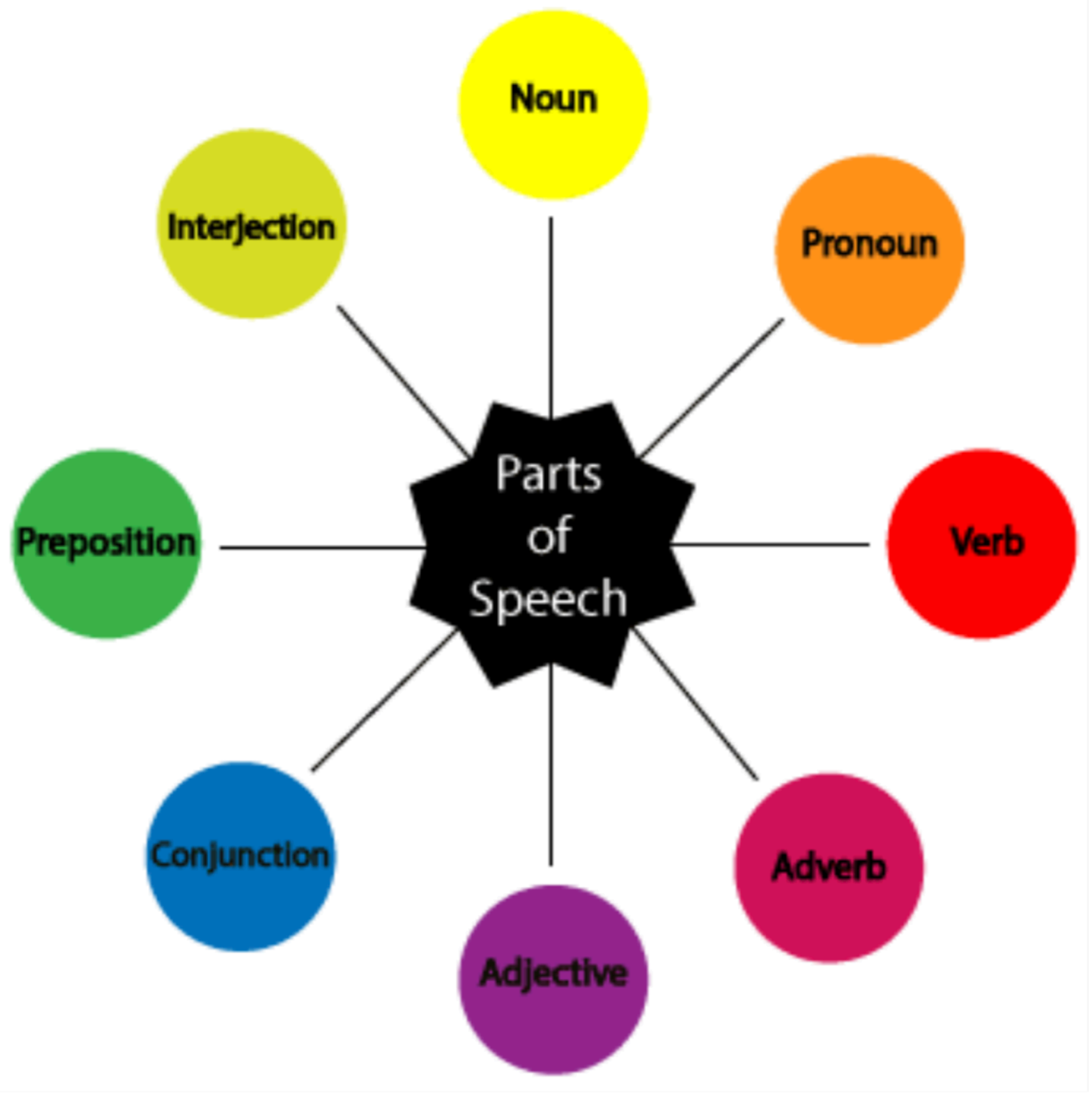
Parts of Speech

- How many part of speech tags do you think English has?

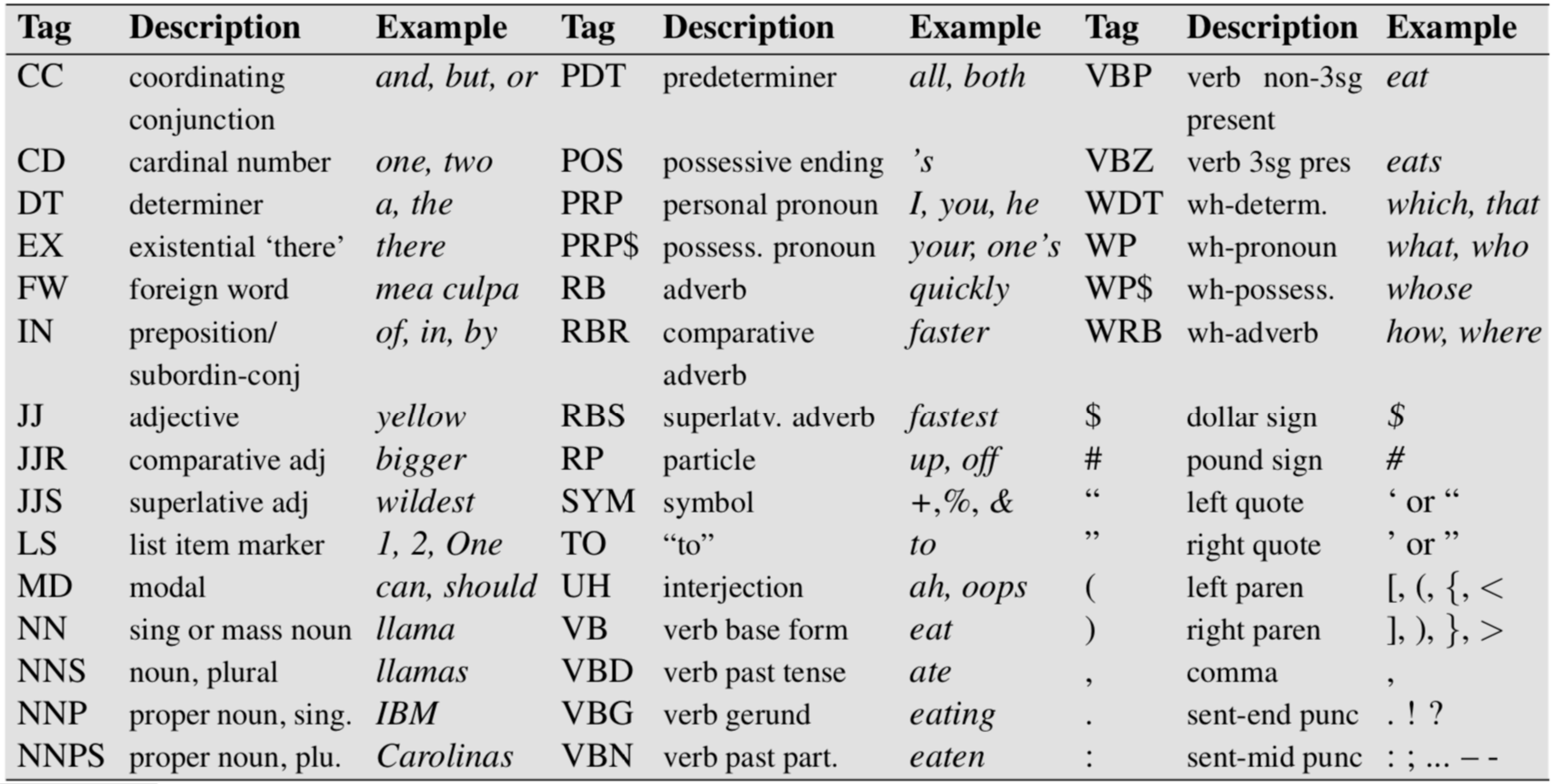
A. < 10

B. 10 - 30

C. >30



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

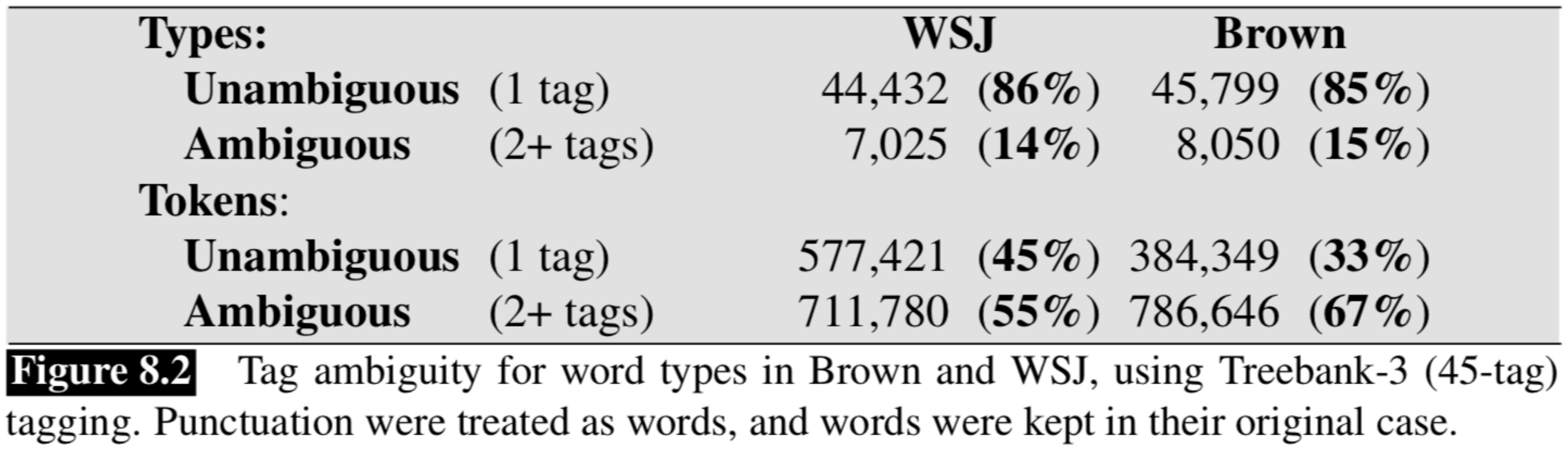
(Marcus et al., 1993)

Other corpora: Brown, WSJ, Switchboard

Part of Speech Tagging

- Tag each word with its part of speech
- 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

Same word, different tags



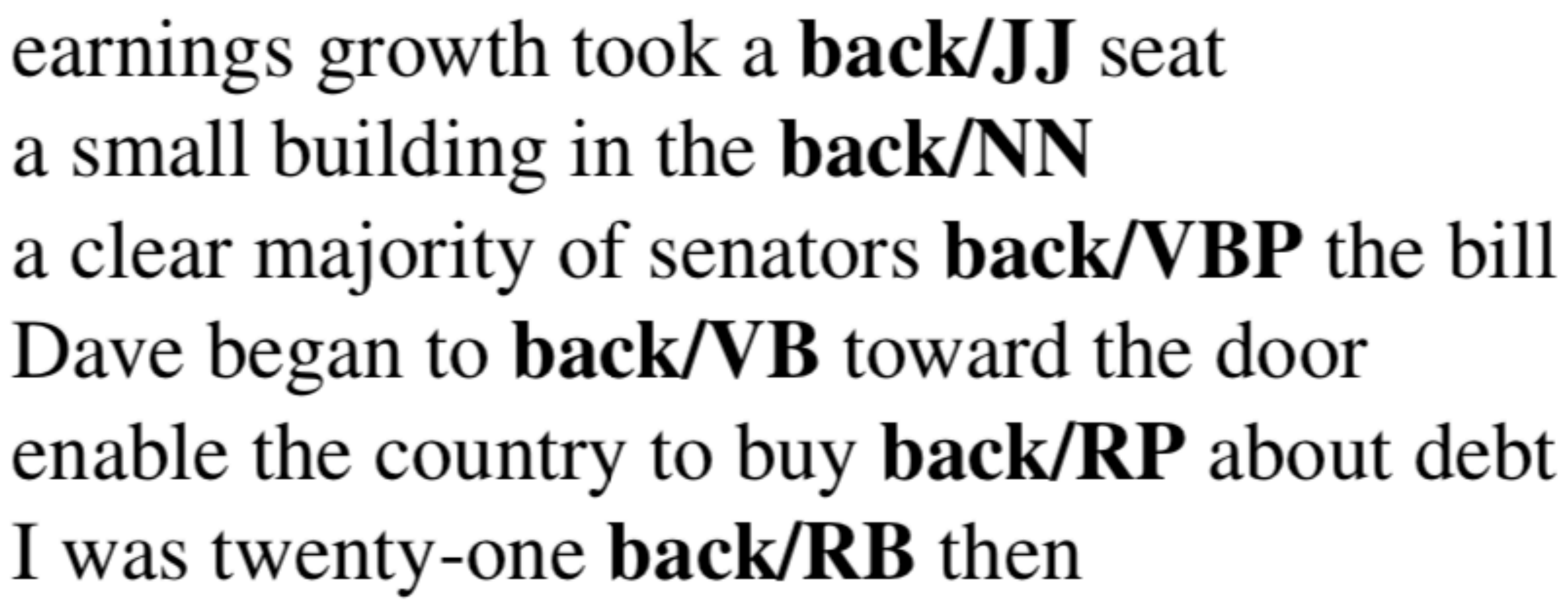
		WSJ		Brown	
Types:					
Unambiguous	(1 tag)	44,432	(**86%**)	45,799	(**85%**)
Ambiguous	(2+ tags)	7,025	(**14%**)	8,050	(**15%**)
Tokens:					
Unambiguous	(1 tag)	577,421	(**45%**)	384,349	(**33%**)
Ambiguous	(2+ tags)	711,780	(**55%**)	786,646	(**67%**)

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

- Tag each word with its part of speech
- 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

Same word, different tags



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!

How accurate do you think this baseline would be at tagging words?

A) <50%

B) 50-75%

C) 75-90%

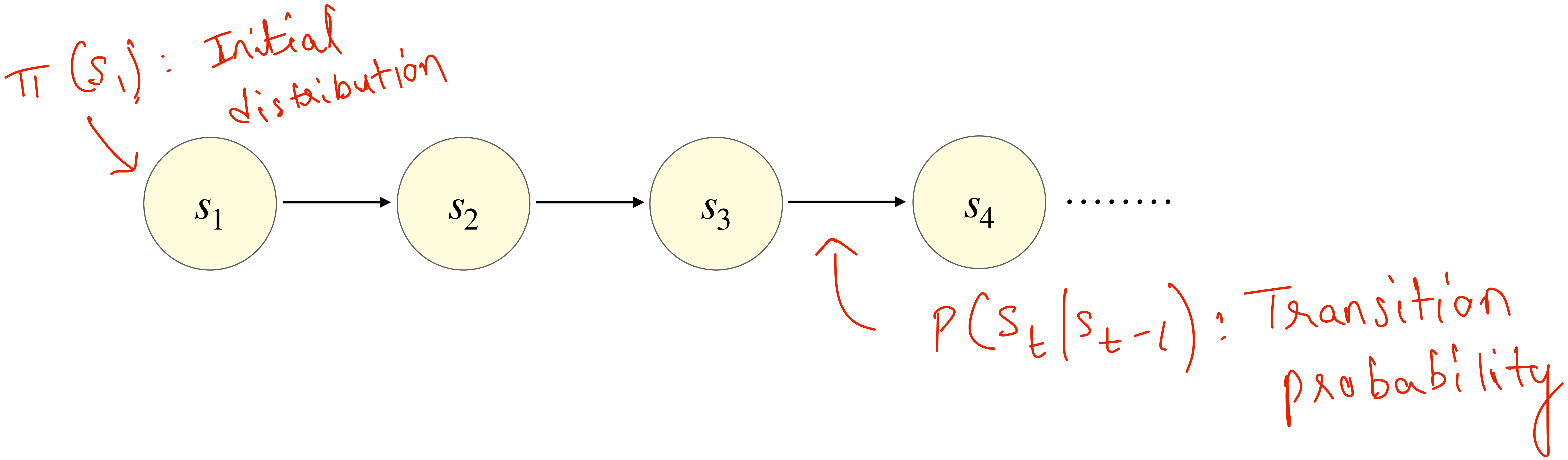
D) >90%

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>* ("good the") or *<DT, IN>* *(*"the in"*)*
- Better to make decisions on entire sentences instead of individual words (Sequence modeling!)

Hidden Markov Models

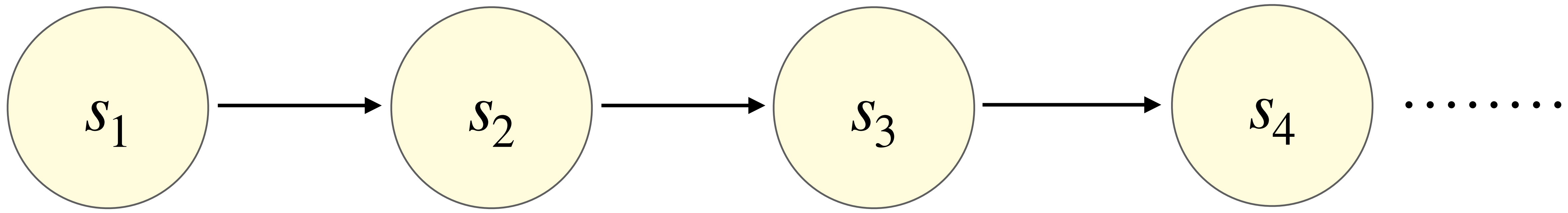
Markov chains


$\pi(s_1)$: Initial distribution
s_1
s_2
s_3
s_4
....
$P(s_t | s_{t-1})$: Transition probability

- Model probabilities of sequences of variables
- Each state can take one of K values (can assume {1, 2, ..., K} for simplicity)
- Markov assumption: $P(s_t | s_{<t}) \approx P(s_t | s_{t-1})$

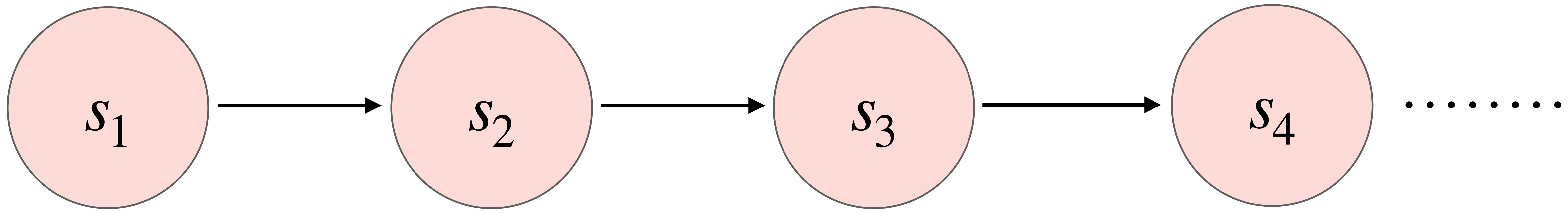
Where have we seen this before? Language models!

Markov chains



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

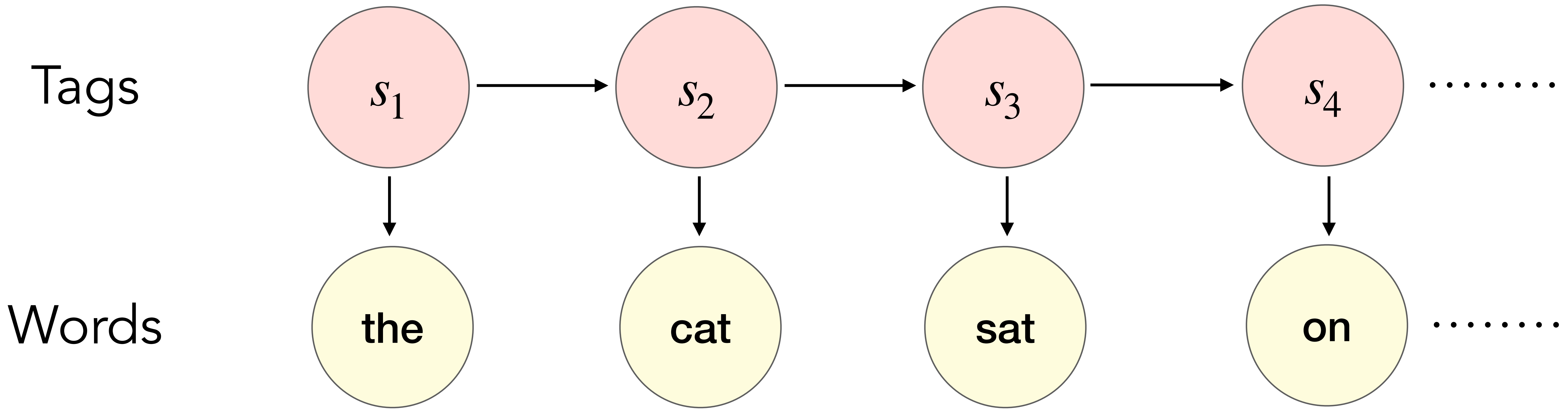
Markov chains



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

- We don't normally see sequences of POS tags in text

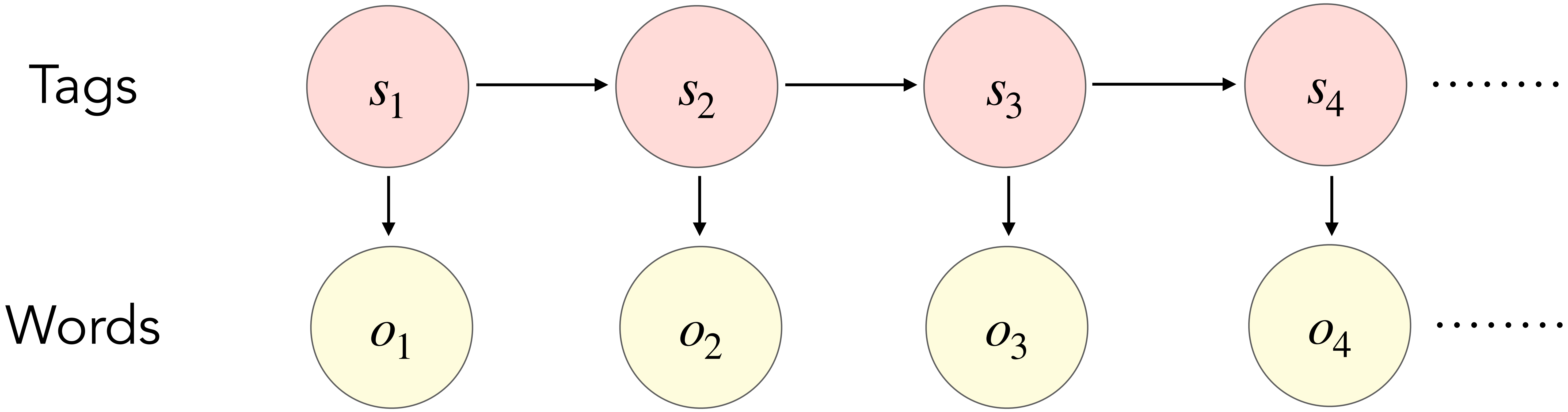
Hidden Markov Model (HMM)



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

- We don’t normally see sequences of POS tags in text
- But we do observe the words!
- HMM allows us to *jointly reason* over both hidden and observed events.
  - Assume that each position has a tag that generates a word

Components of an HMM



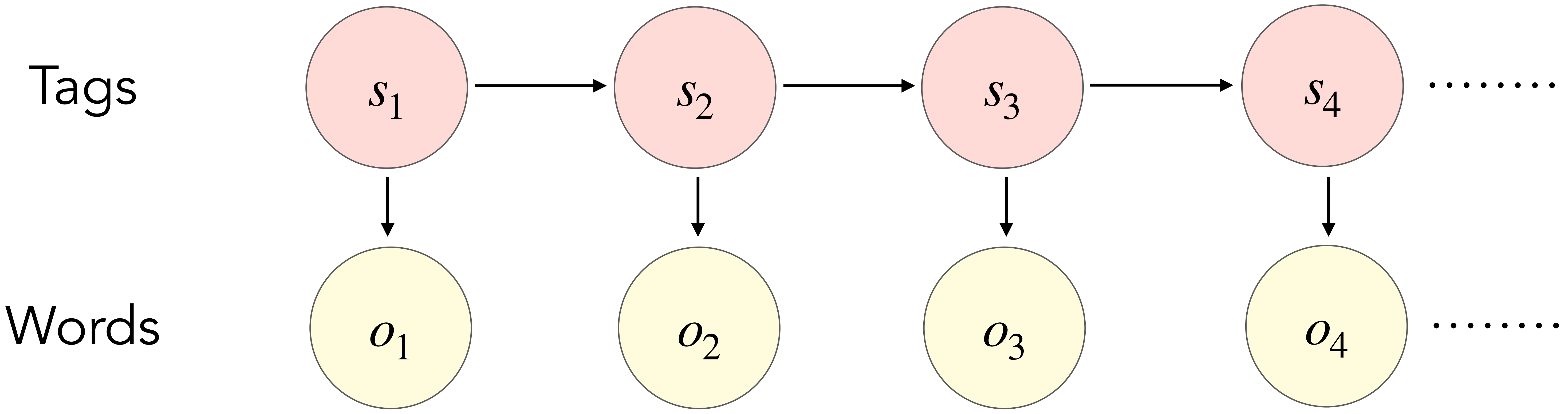
1. Set of states S = {1, 2, ..., K} and set of observations O

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

3. Transition probabilities $P(s_{t+1} | s_t)$ (OR $\theta_{s_t \rightarrow s_{t+1}}$)

4. Emission probabilities $P(o_t | s_t)$ (OR $\phi_{s_t \rightarrow o_t}$)

Assumptions



1. Markov assumption:

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

2. Output independence:

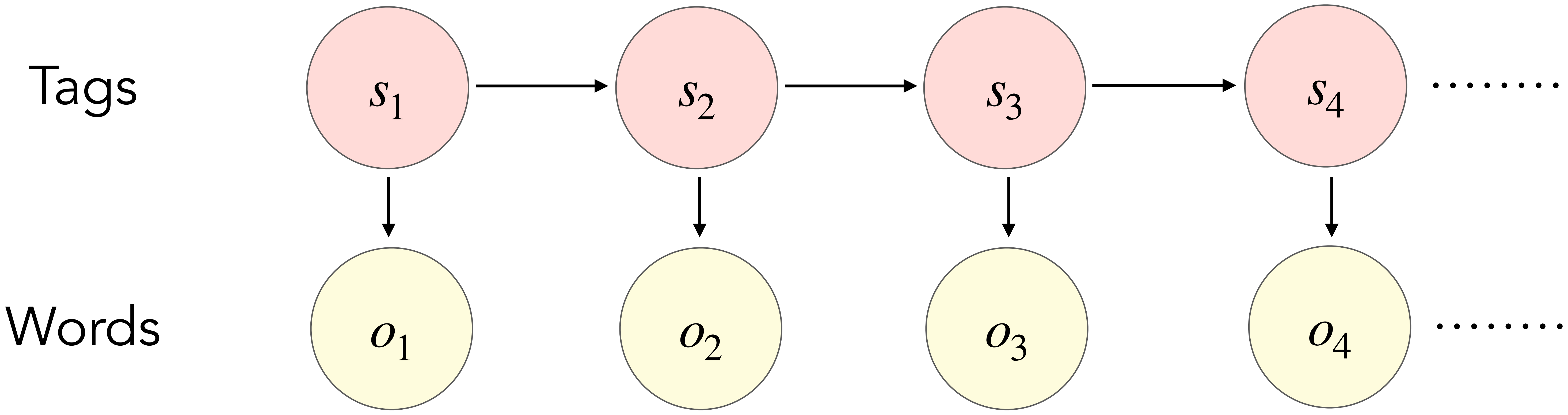
$$P(o_t \mid s_1, \ldots, s_t) \approx P(o_t \mid s_t)$$

Depends on language!
1) assumes POS tag sequences do not have very strong priors/ long-range dependencies
2) assumes neighboring tags don't affect current word

Which do you think is a stronger assumption?

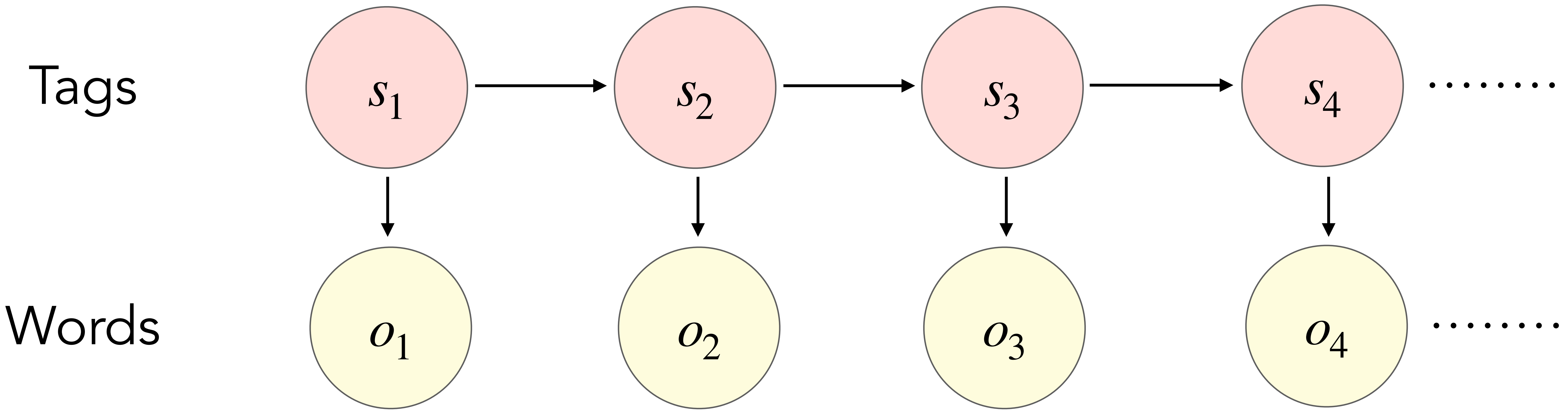
A) Markov assumption
B) Output independence

Sequence likelihood



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

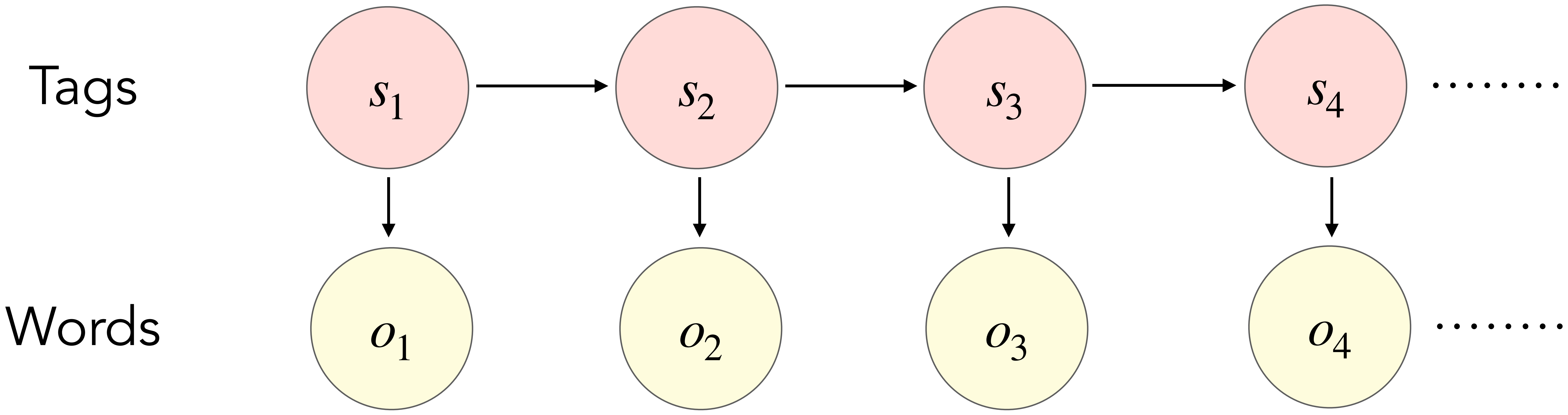
Sequence likelihood



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

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

Sequence likelihood

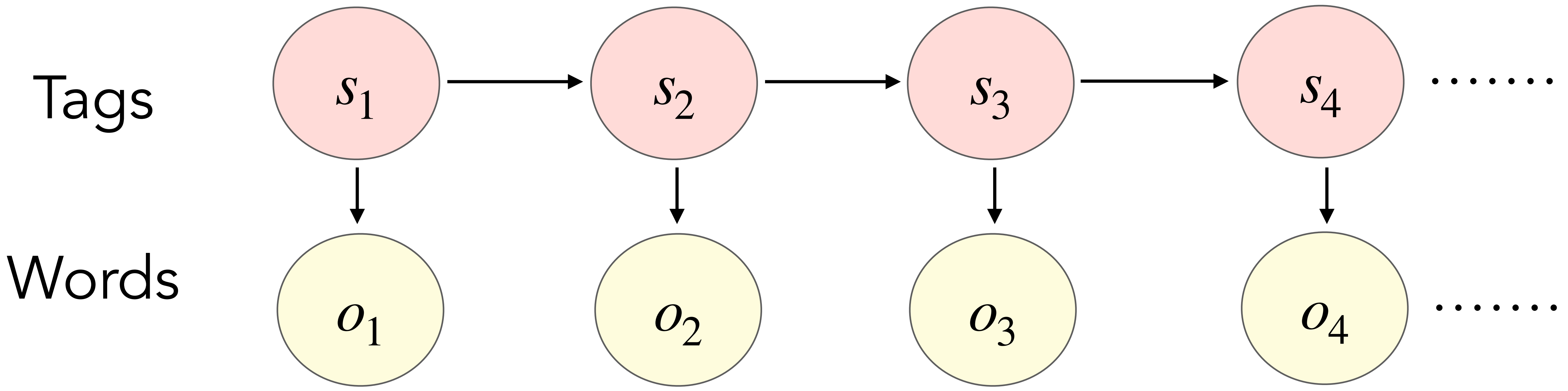


$$P(S, O) = P(s_1, s_2 \ldots s_n, o_1, o_2 \ldots o_n)$$

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

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

Example: Sequence likelihood



Dummy start state

s_{t+1}

s_t	DT	NN
∅	0.8	0.2
DT	0.2	0.8
NN	0.3	0.7

o_t

	the	cat
DT	0.9	0.1
NN	0.5	0.5

What is the joint probability $P($*the cat, DT NN*$)$?

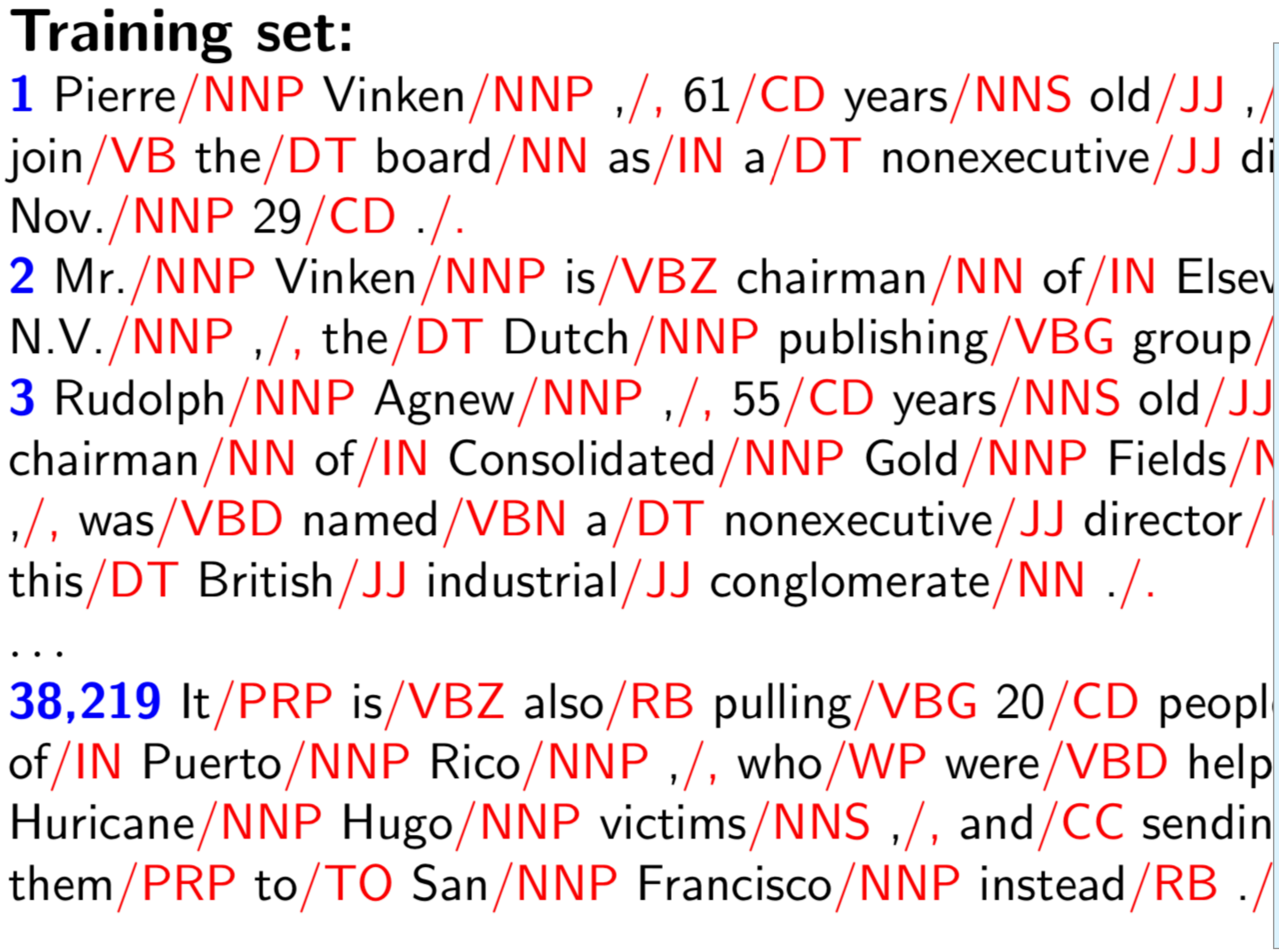
A) $(0.8 * 0.8) * (0.9 * 0.5)$

B) $(0.2 * 0.8) * (0.9 * 0.5)$

C) $(0.3 * 0.7) * (0.5 * 0.5)$

Ans: A

Learning



Training set:

1 Pierre/NNP Vinken/NNP ,/, 61/CD years/NNS old/JJ ,/
join/VB the/DT board/NN as/IN a/DT nonexecutive/JJ di
Nov./NNP 29/CD ./.
2 Mr./NNP Vinken/NNP is/VBZ chairman/NN of/IN Elsev
N.V./NNP ,/, the/DT Dutch/NNP publishing/VBG group/
3 Rudolph/NNP Agnew/NNP ,/, 55/CD years/NNS old/JJ
chairman/NN of/IN Consolidated/NNP Gold/NNP Fields/N
,/, was/VBD named/VBN a/DT nonexecutive/JJ director/
this/DT British/JJ industrial/JJ conglomerate/NN ./.
. . .
38,219 It/PRP is/VBZ also/RB pulling/VBG 20/CD peopl
of/IN Puerto/NNP Rico/NNP ,/, who/WP were/VBD help
Huricane/NNP Hugo/NNP victims/NNS ,/, and/CC sendin
them/PRP to/TO San/NNP Francisco/NNP instead/RB ./

- Maximum likelihood estimate:

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

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

Learning Example

1. the/DT cat/NN sat/VBD on/IN the/DT mat/NN

2. Princeton/NNP is/VBZ in/IN New/NNP Jersey/NNP

3. the/DT old/NN man/VB the/DT boats/NNS

$$P(NN \mid DT) = \frac{3}{4}$$

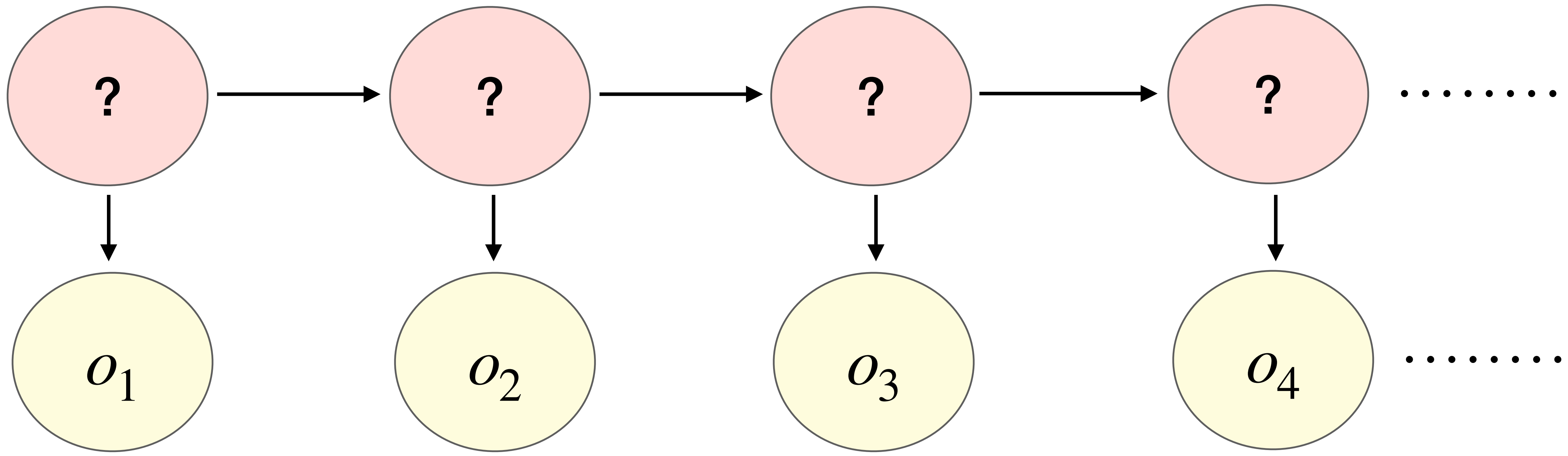
$$P(cat \mid NN) = \frac{1}{3}$$

- Maximum likelihood estimate:

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

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

Decoding with HMMs

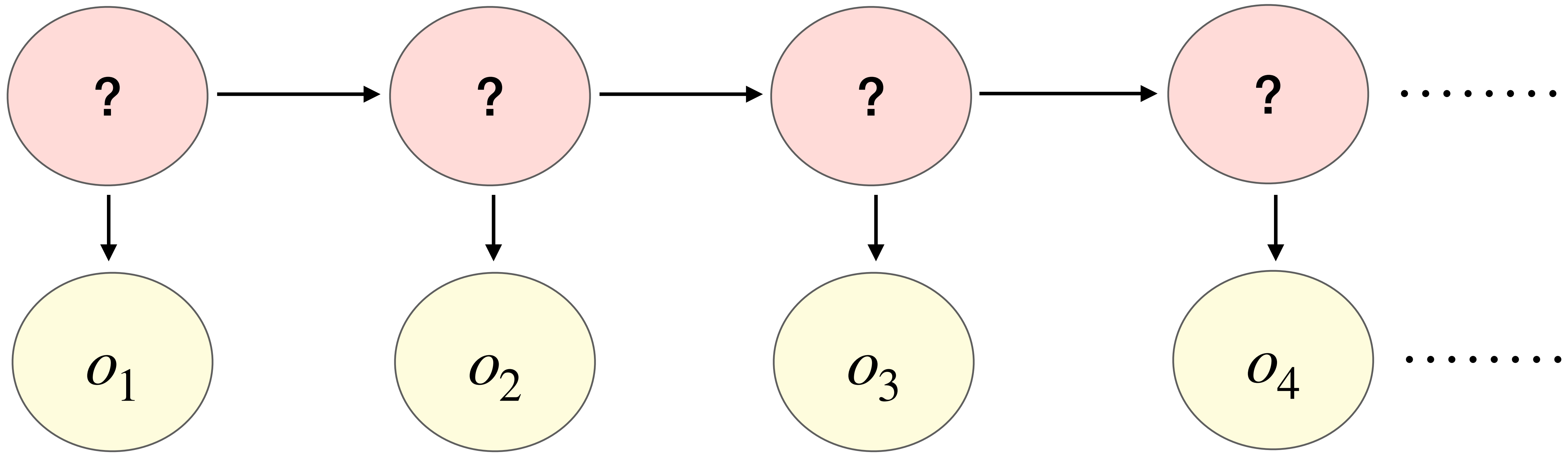


Task: Find the most probable sequence of states $\langle s_1, s_2, \ldots, s_n \rangle$ given the observations $\langle o_1, o_2, \ldots, o_n \rangle$

$$\hat{S} = \arg\max_{S} P(S|O) = \arg\max_{S} \frac{P(S)\,P(O|S)}{P(O)}$$

[Bayes]

Decoding with HMMs

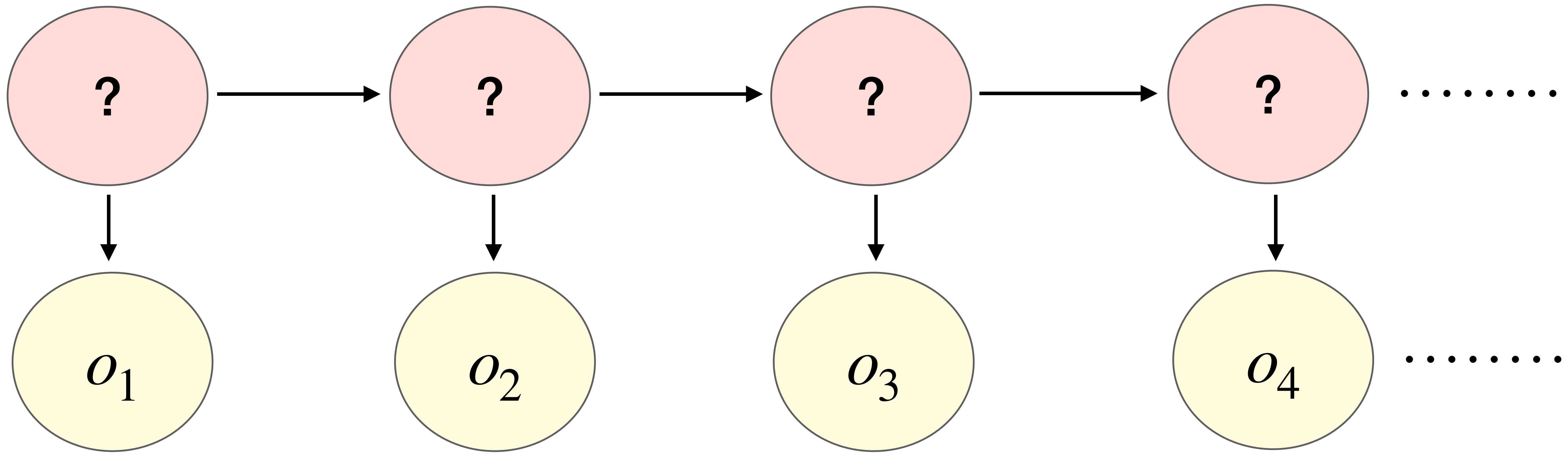


Task: Find the most probable sequence of states $\langle s_1, s_2, \ldots, s_n \rangle$ given the observations $\langle o_1, o_2, \ldots, o_n \rangle$

$$\hat{S} = \arg\max_S P(S|O) = \arg\max_S \frac{P(S)\,P(O|S)}{P(O)} \quad [\text{Bayes}]$$

$$= \arg\max_S P(S)\,P(O|S)$$

Decoding with HMMs



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

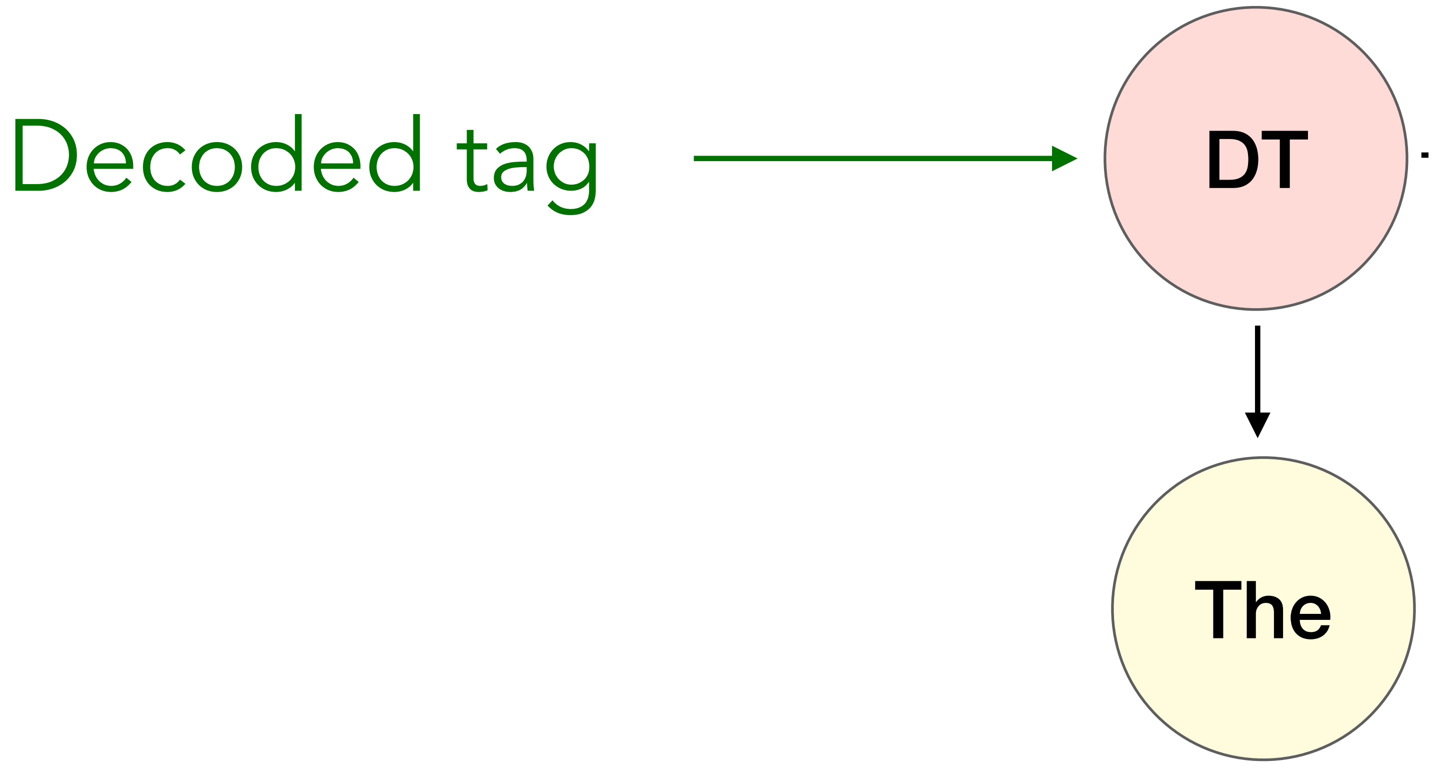
$$\hat{S} = \underset{S}{\arg\max}\ P(S)\, P(O|S)$$

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

How can we maximize this?

Search over all state sequences?

Greedy decoding



Decode/reveal one state at a time

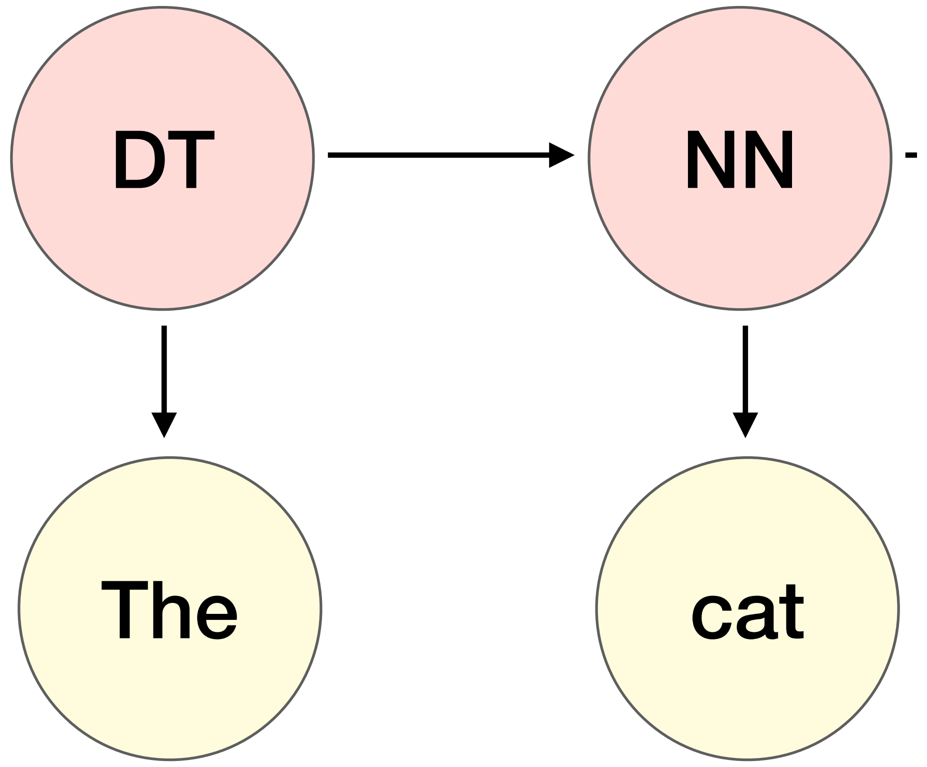
$$\operatorname*{argmax}_{s} \pi(s_1 = s)\, P(\text{The} \mid s)$$

$$= \text{`DT'}$$

$$\hat{S} = \operatorname*{argmax}_{S} P(S)\, P(O \mid S)$$

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

Greedy decoding



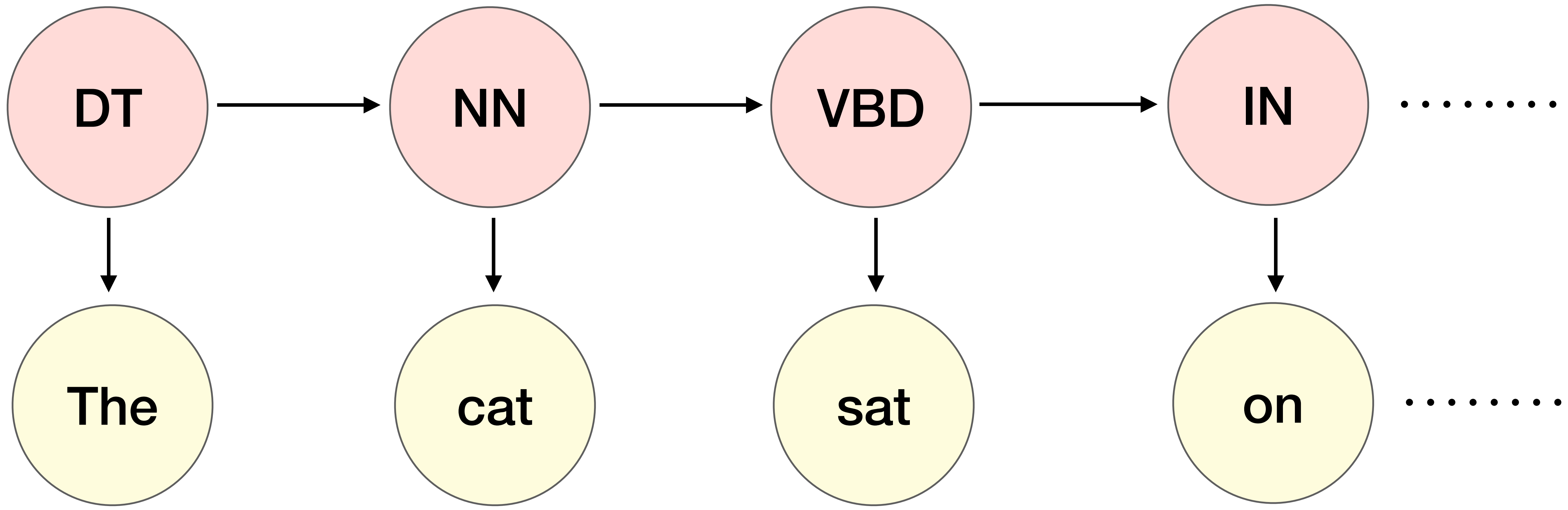
$$\underset{s}{\operatorname{argmax}}\ P(s_2=s \mid DT)\, P(cat \mid s)$$

$$= \text{'NN'}$$

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

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

Greedy decoding



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

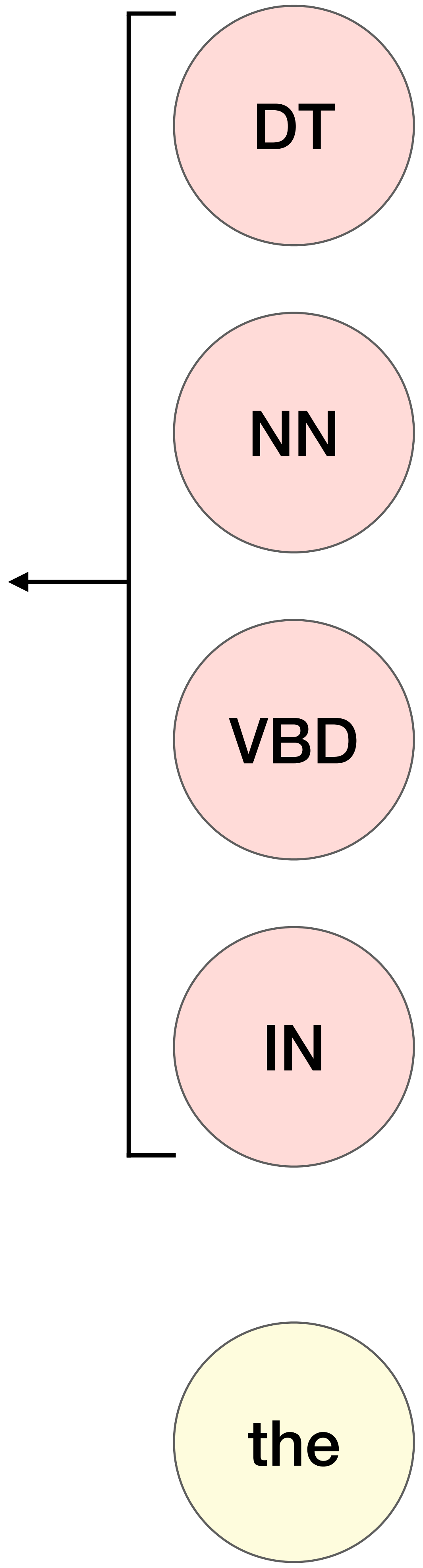
- Not guaranteed to produce the overall optimal sequence

- Local decisions

Viterbi decoding

- Use dynamic programming!
- Maintain some extra data structures
- Probability lattice, $M[T, K]$ and backtracking matrix, $B[T, K]$
  - T : Number of time steps
  - K : Number of states
- $M[i, j]$ stores most probable sequence of states ending with state **j** at time **i**
- $B[i, j]$ is the tag at time **i-1** in the most probable sequence ending with tag **j** at time **i**

Viterbi decoding



4 possible POS tags

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

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

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

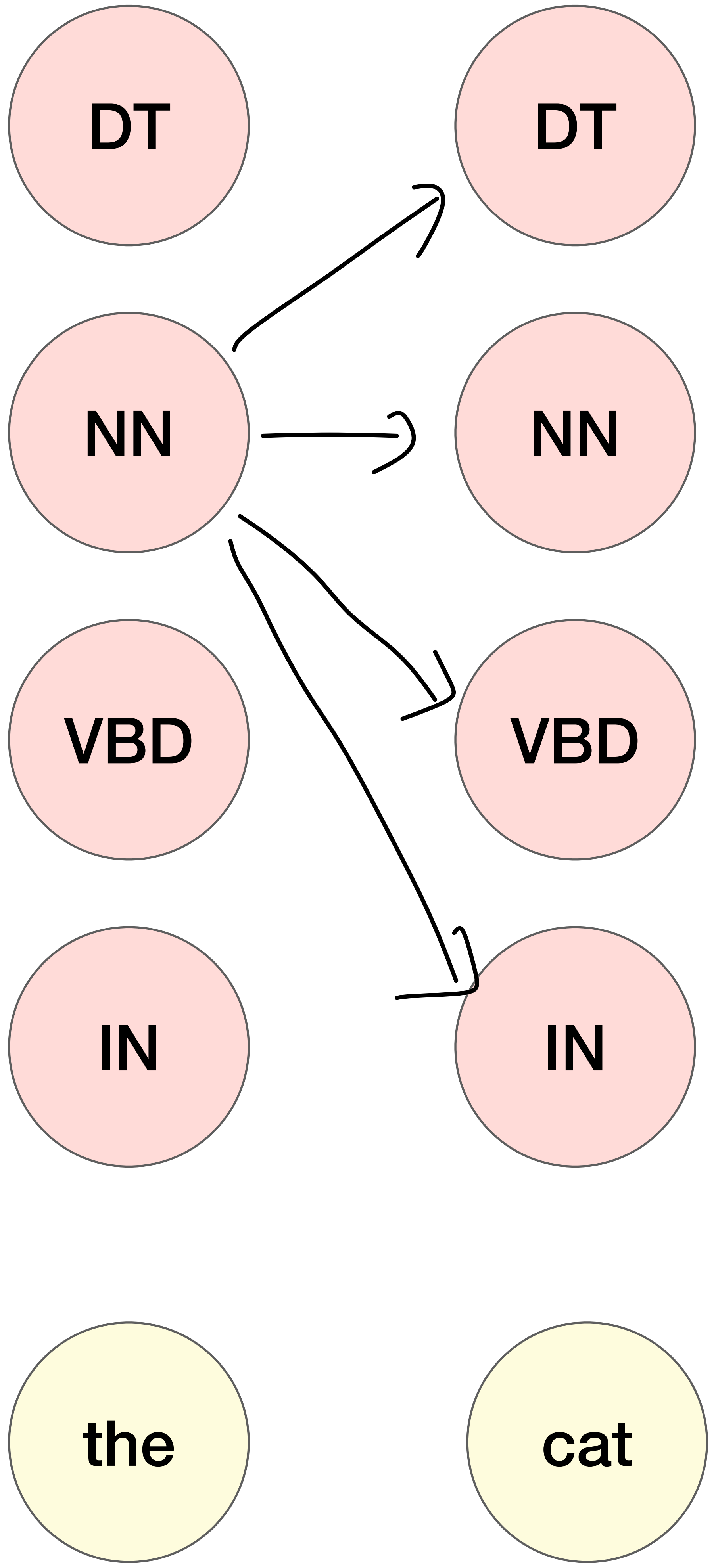
$$M[1,IN] = \pi(IN)\ P(\text{the} \mid IN)$$

Initialize the table

Forward

Viterbi decoding

Consider all possible previous tags



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

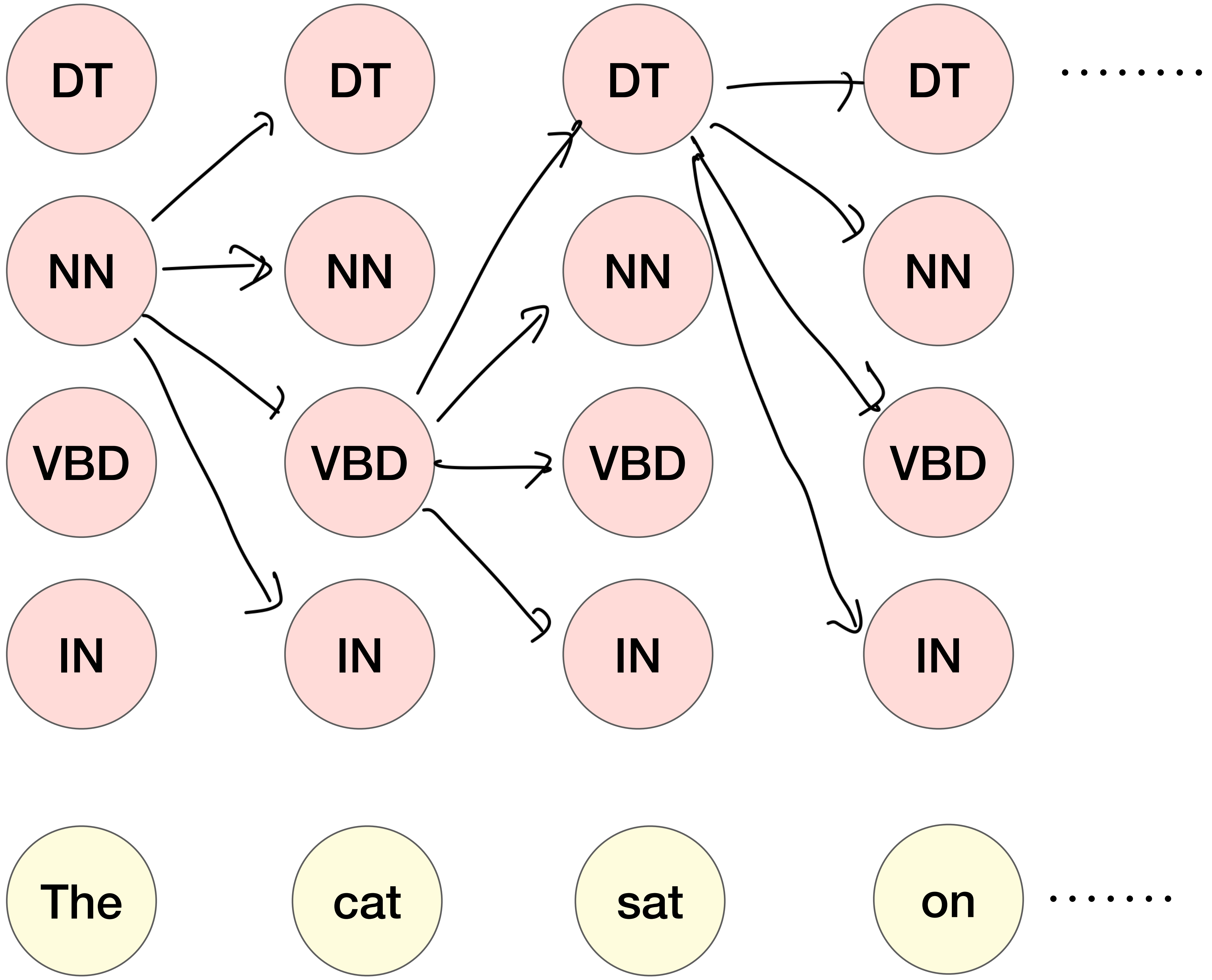
$$M[2,NN] = \max_k M[1,k]\ P(NN|k)\ P(\text{cat}|NN)$$

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

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

Forward

Viterbi decoding



What is the time complexity of this algorithm?

$O(nK^2)$

A) $O(n)$

B) $O(nK)$

C) $O(nK^2)$

D) $O(n^2K)$

n = number of timesteps

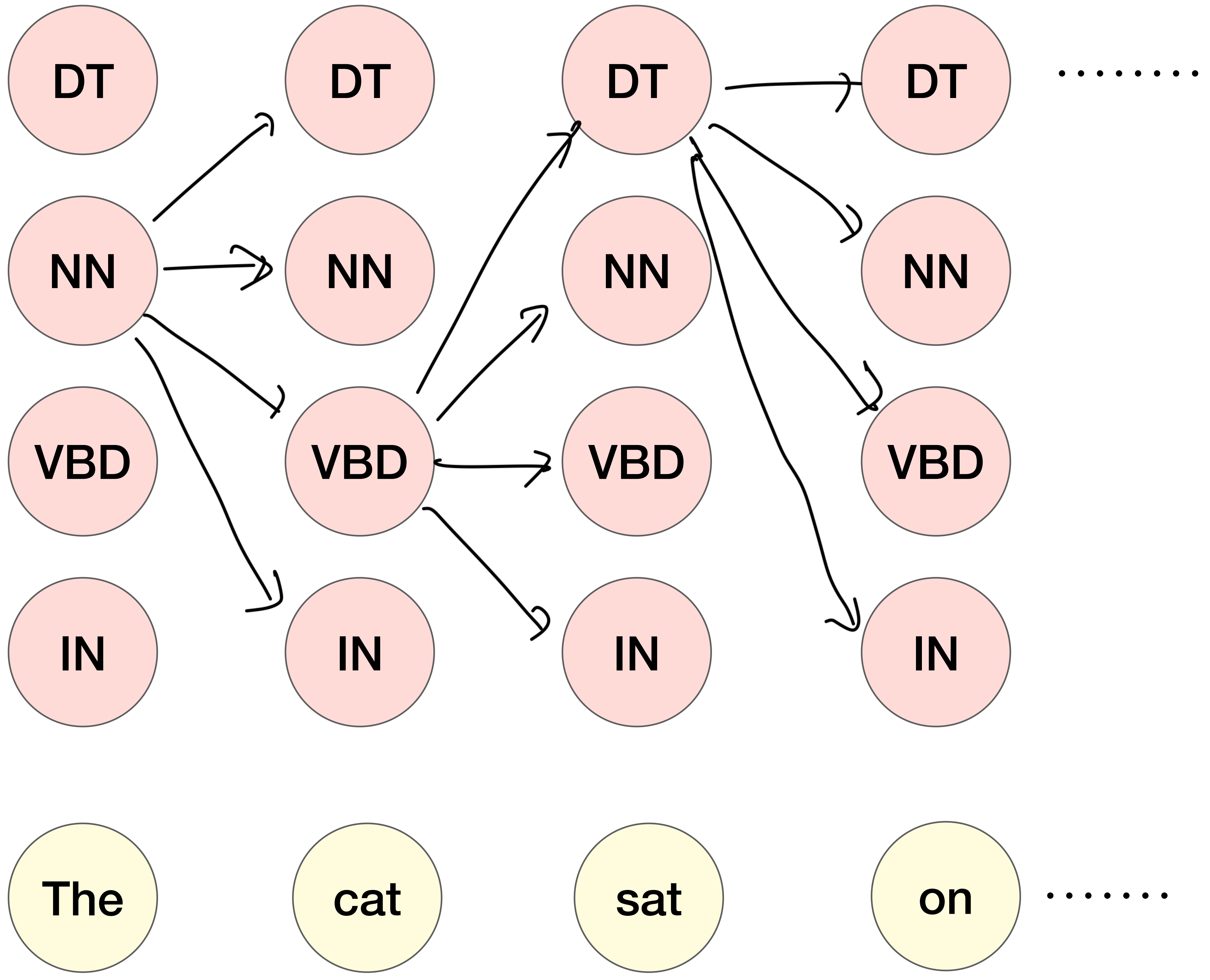
K = number of states

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

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

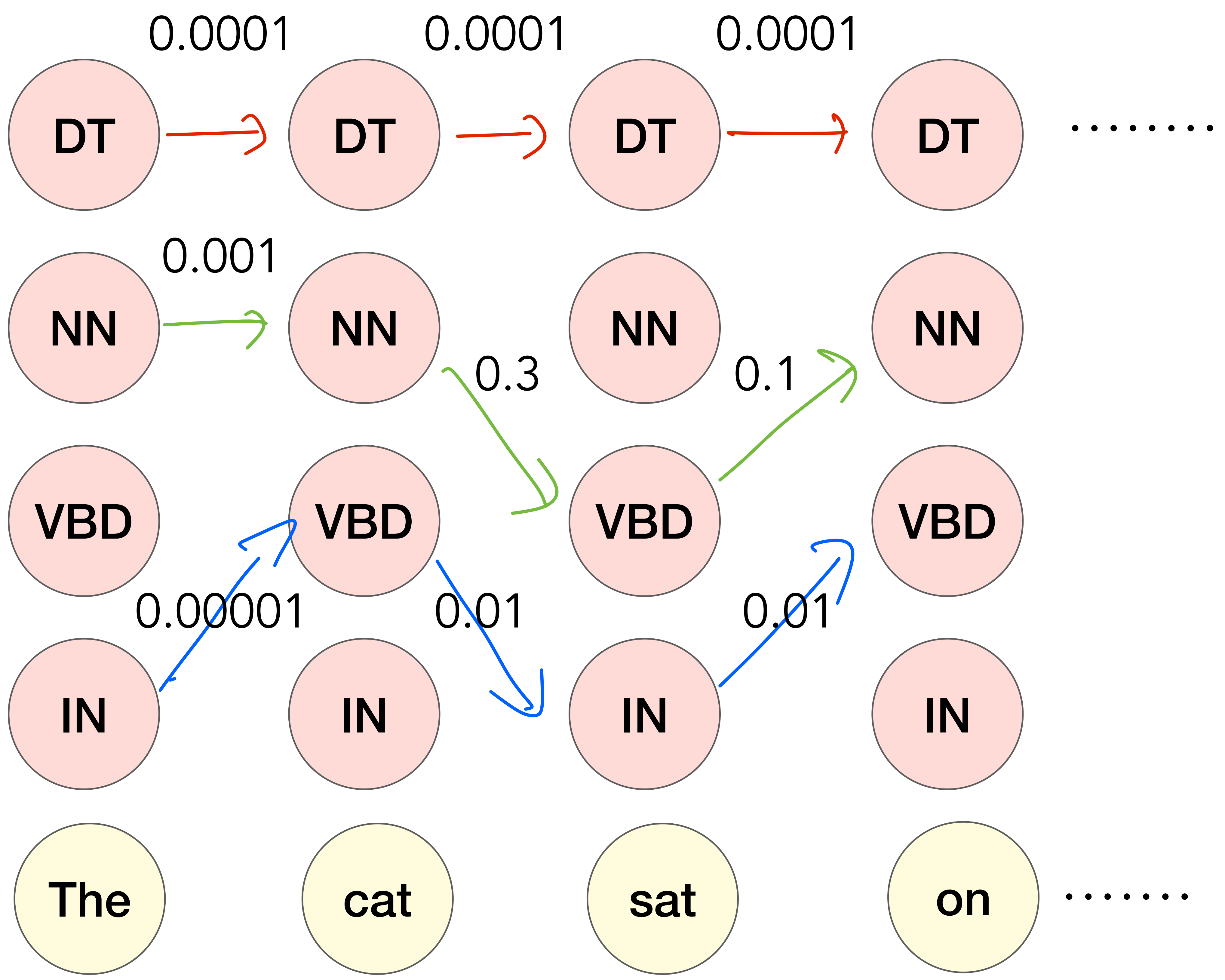
Beam Search

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



Beam Search

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



Observation: *Many paths have very low likelihood!*

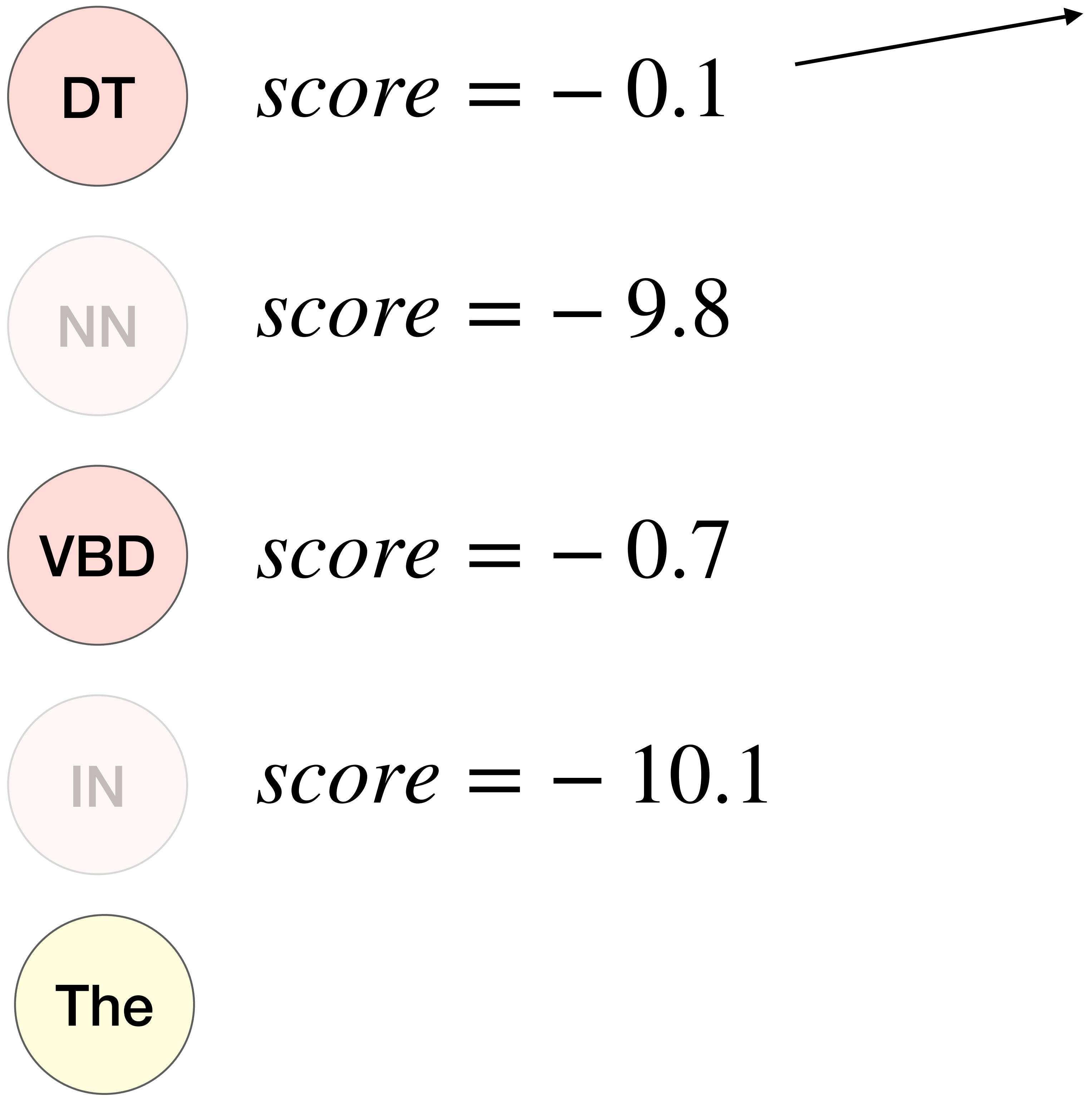
Beam Search

- 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

$\beta = 2$



$score = -0.1$

$score = -9.8$

$score = -0.7$

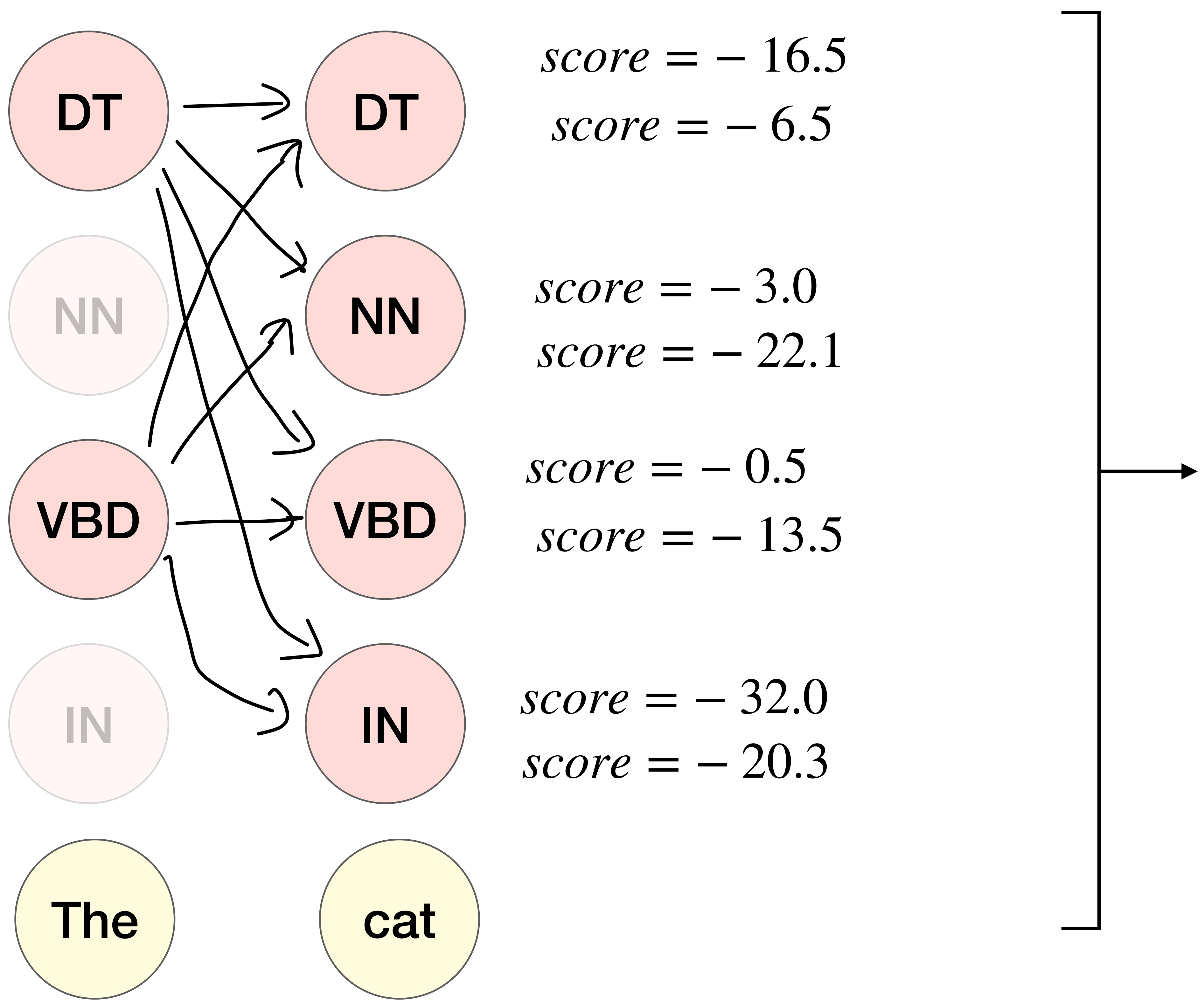
$score = -10.1$

log probabilities

Beam Search

- Keep a fixed number of hypotheses at each point

$\beta = 2$



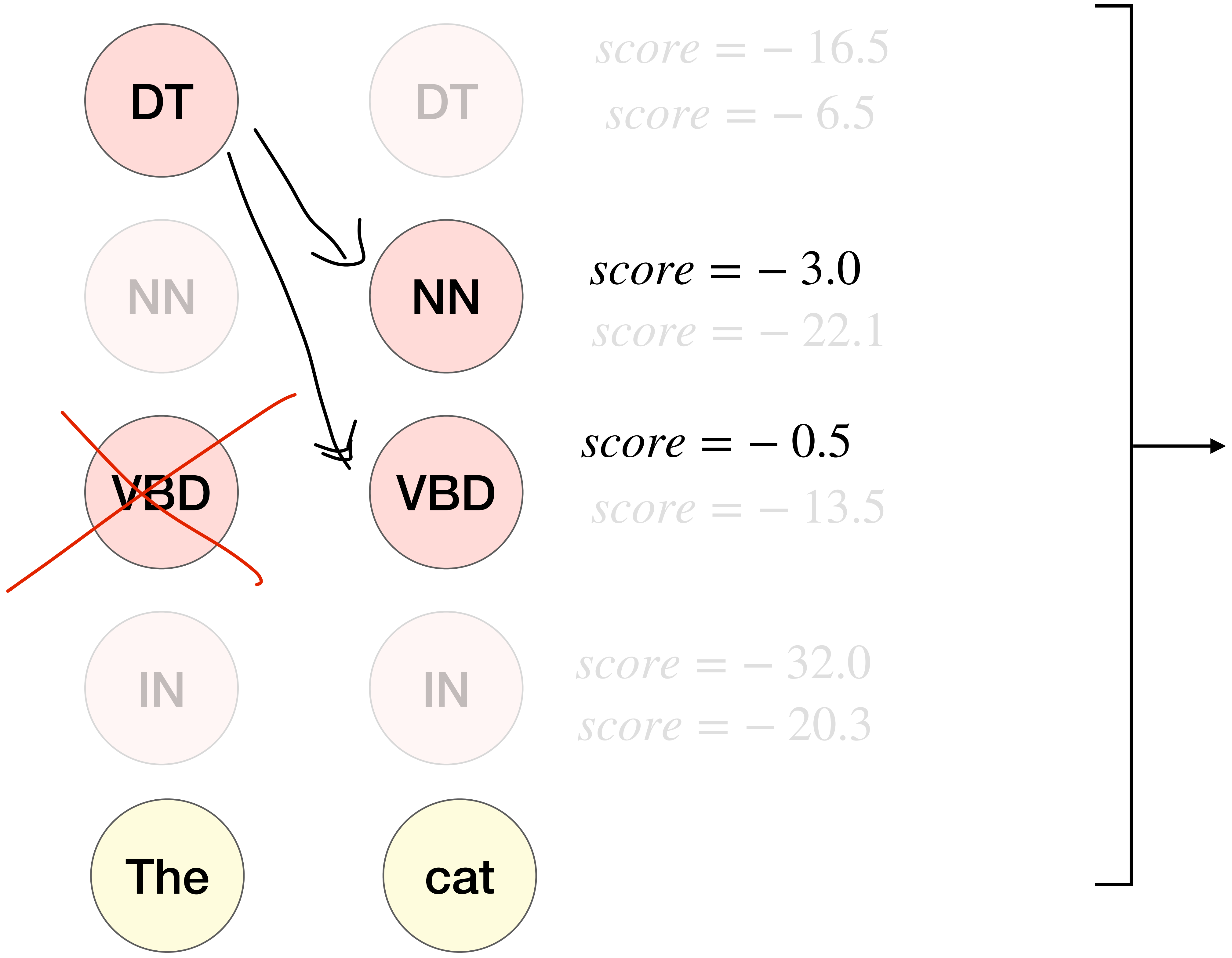
Accumulated scores

Step 1: Expand all partial sequences in current beam

Beam Search

- Keep a fixed number of hypotheses at each point

$\beta = 2$



Accumulated scores

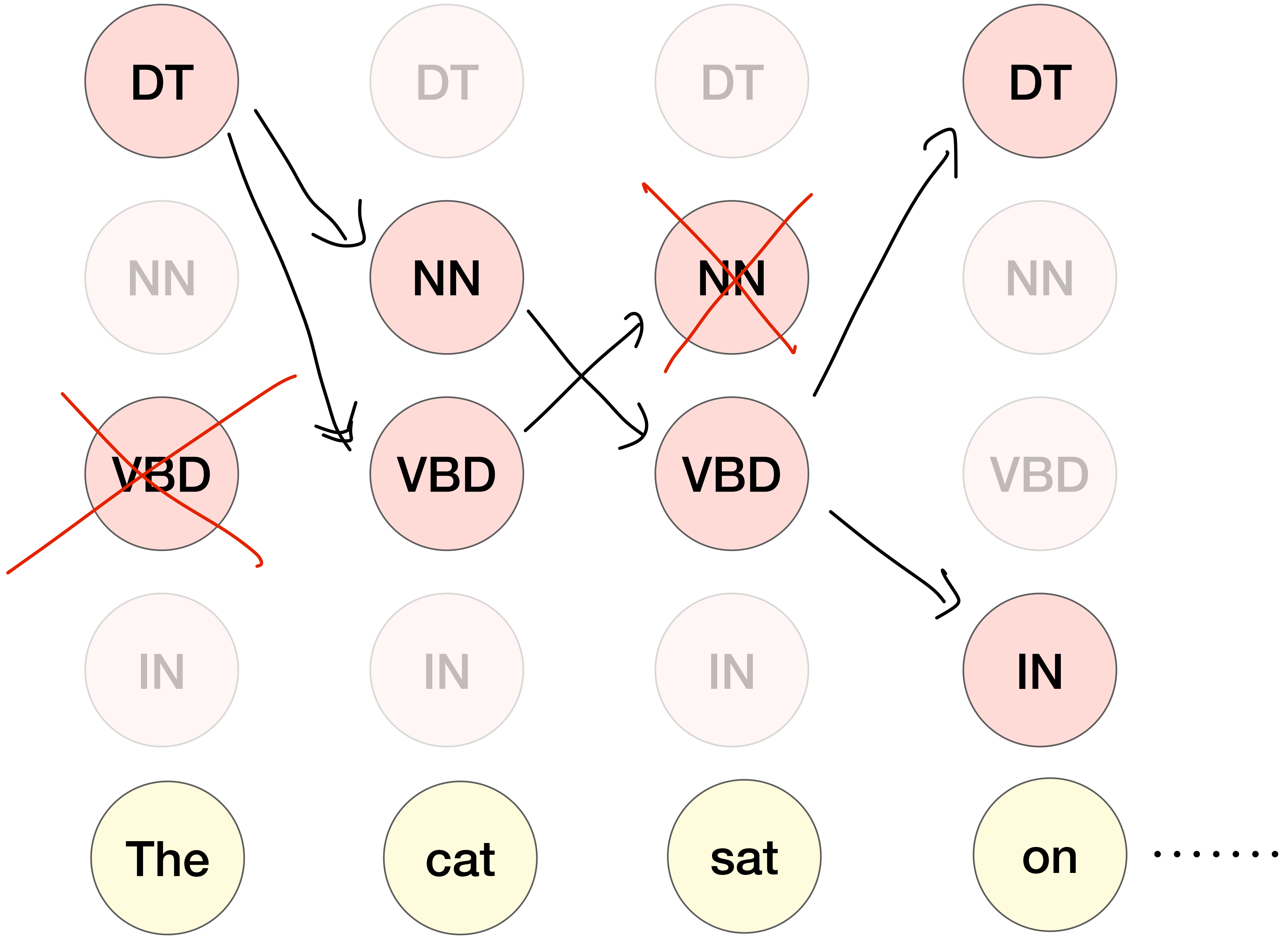
Step 2: Prune set back to top β sequences (sort and select)

… and Repeat!

Beam Search

- Keep a fixed number of hypotheses at each point

$\beta = 2$



What is the time complexity of this algorithm?

n = number of timesteps
K = number of states
β = beam width

Pick $\max_{k} M[n, k]$ from within beam and backtrack

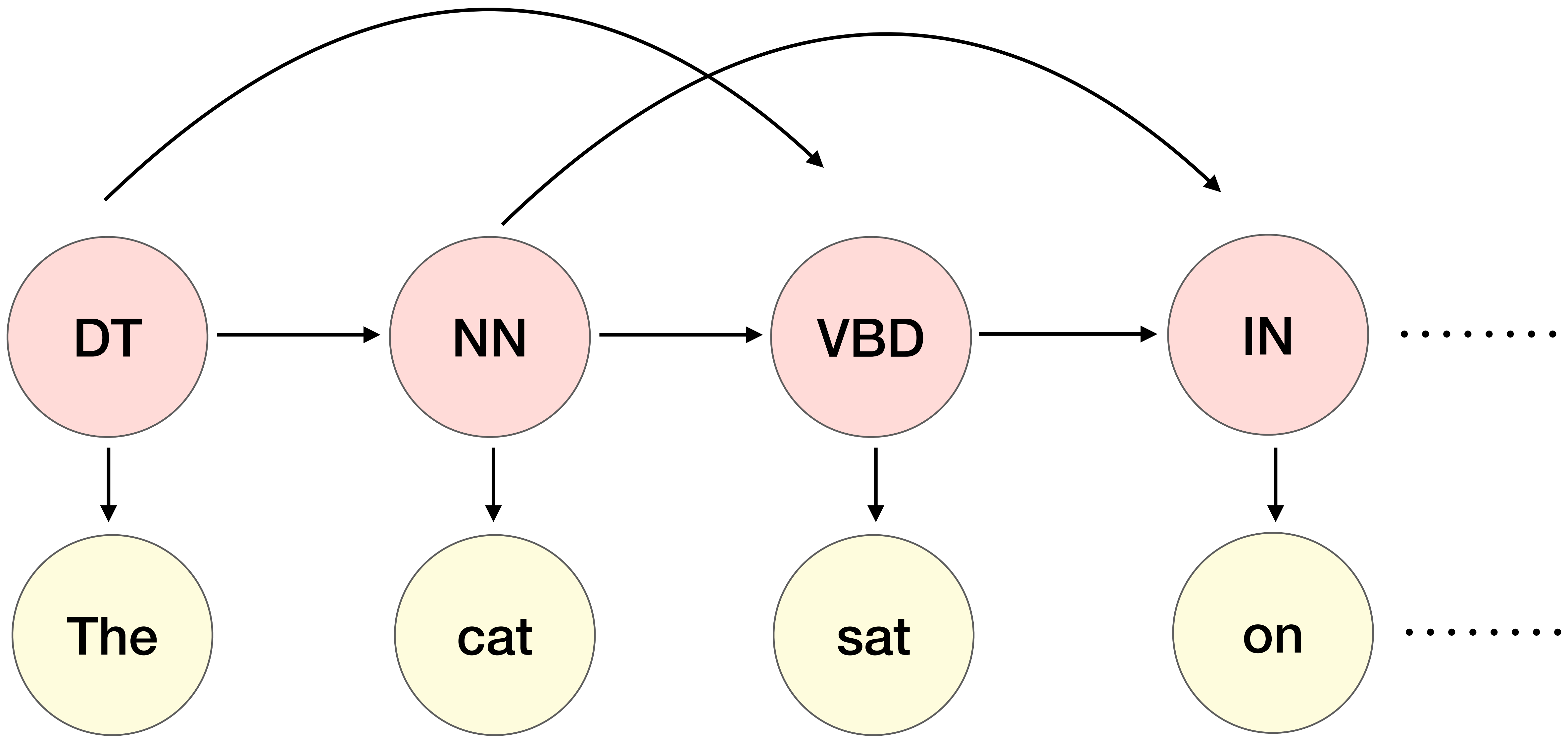
Beam Search

- 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 (some) accuracy for computational savings

Beyond bigrams (Advanced)

- 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?

Give us feedback!

https://forms.gle/D5Fw1tqmWNrNYEzKA