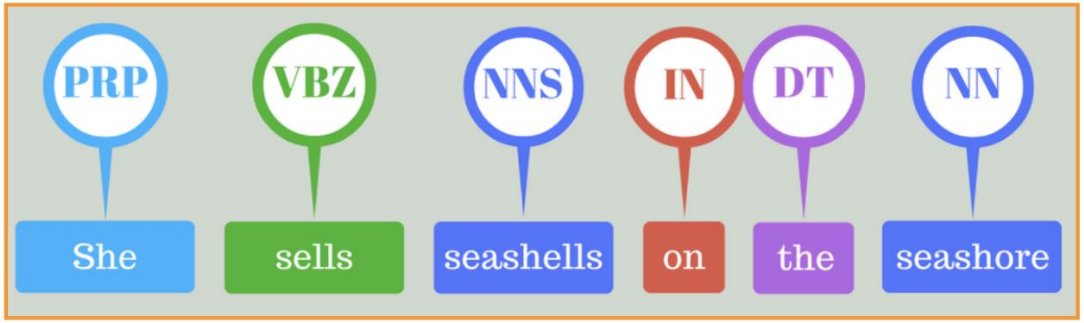
CSE 4392 SPECIAL TOPICS
NATURAL LANGUAGE PROCESSING

Sequence Models

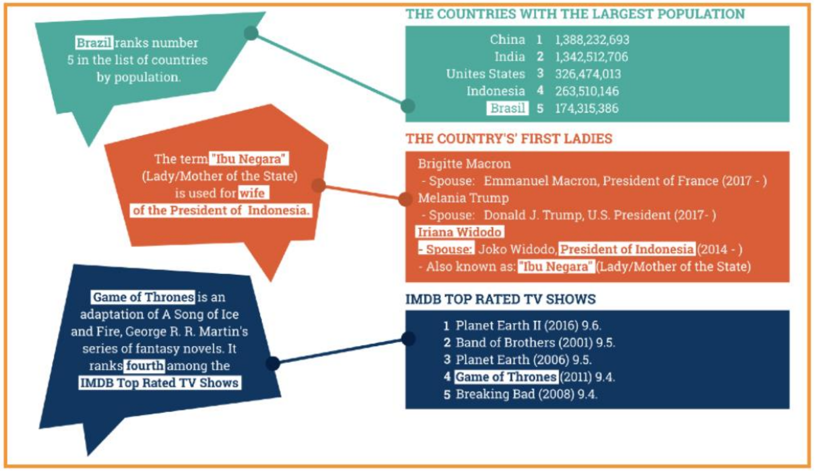
2025 Spring

WHY MODEL SEQUENCES?



Part-of-speech tagging

Name Entity Recognition



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

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

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)

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

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

- Conditional Random Field (CRF)

WHAT ARE POS TAGS?

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

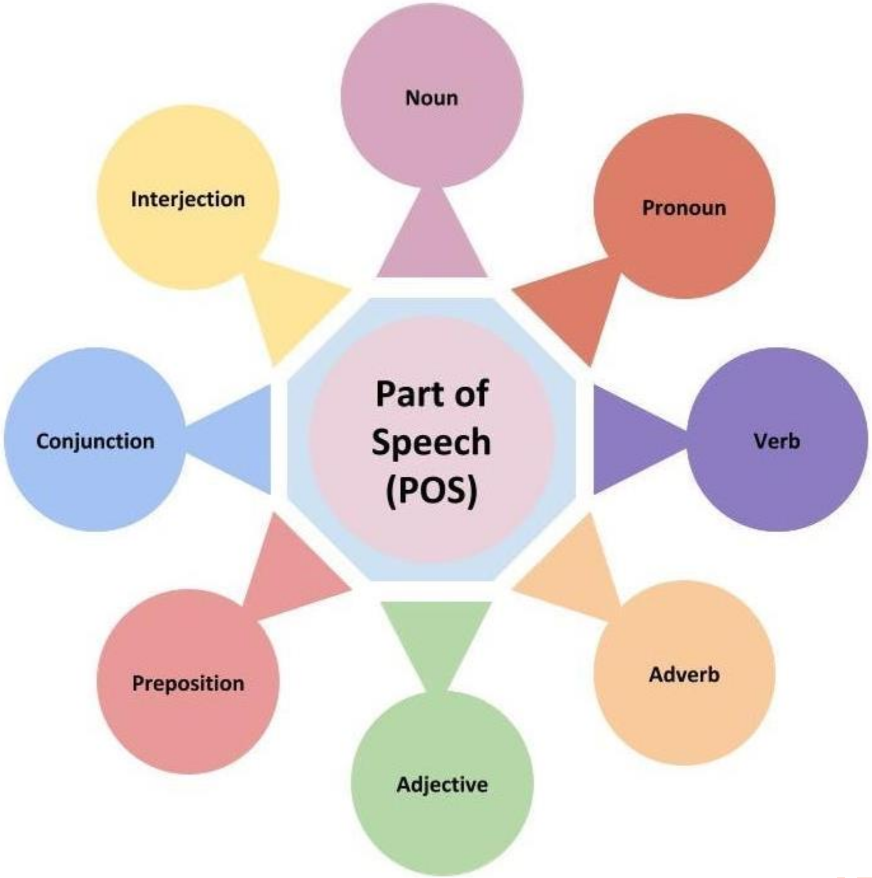
The/DT cat/NN sat/VBD on/IN the/DT mat/NN

Fort/NNP Worth/NNP is/VBZ in/IN Texas/NNP

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

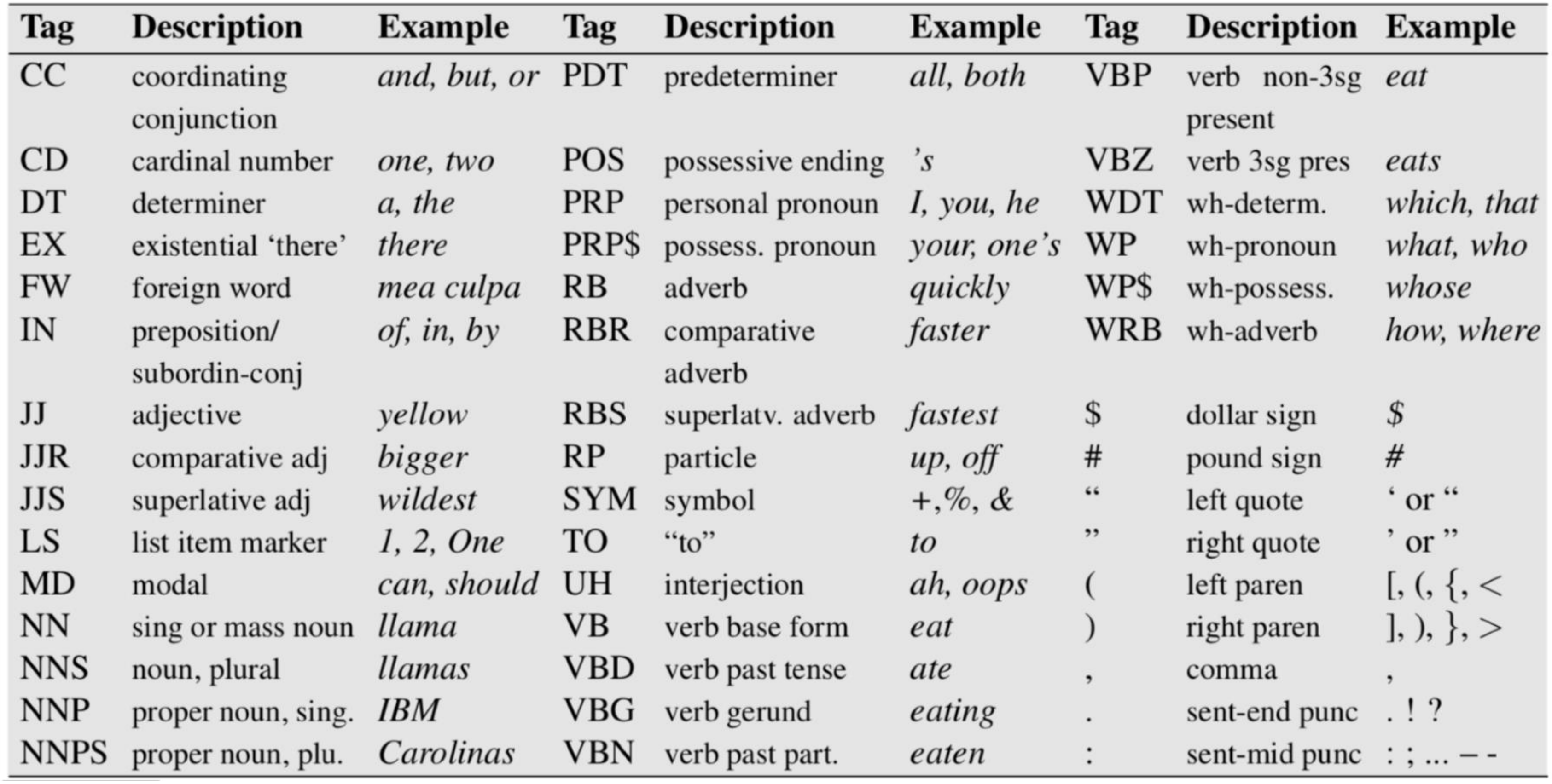
PARTS OF SPEECH

- 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 TAG SET

45 Tags



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	: ; ... – -

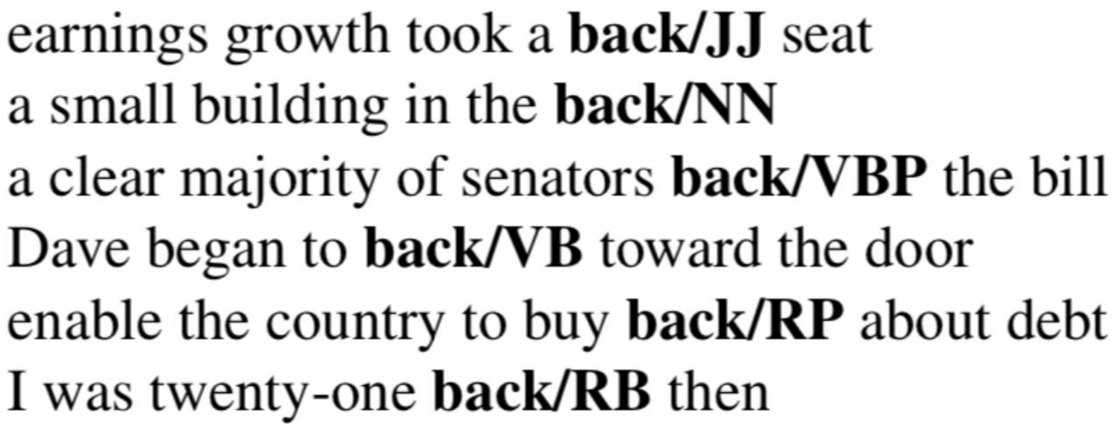
(Marcus et al., 1993)

Other corpora: Brown, WSJ, Switchboard

PART OF SPEECH TAGGING

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

- Some words have MANY functions:



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

A SIMPLE BASELINE

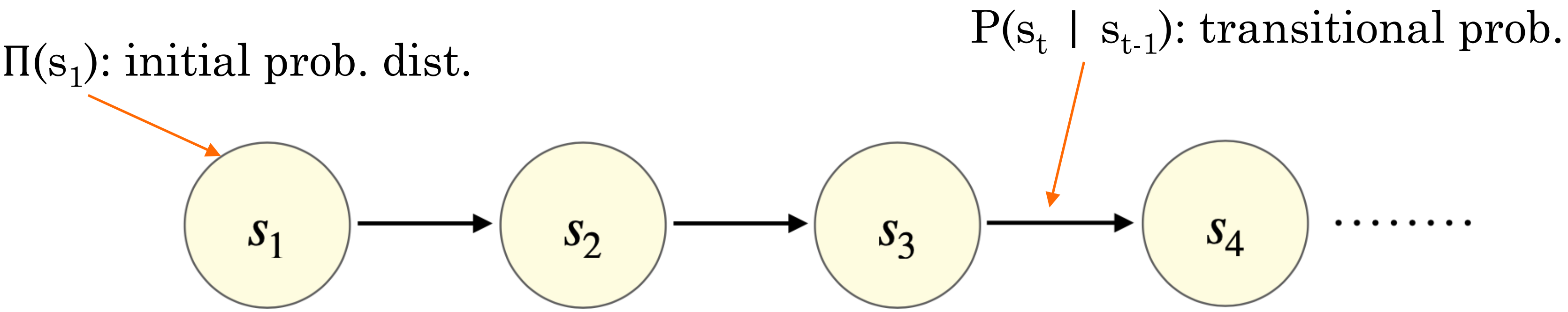
- Most words are easy to disembiguate
- **Most frequence class**: assign each word (token) its most frequently used class in the training set. (e.g., man/NN)
- Accuracy: 92.34% on the Wall Street Journal (WSJ) dataset!
- State of the art: ~ 97%
- Average English sentence: ~ 14 words
  - Sentence level accuracy: $0.92^{14} = 31\%$ 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:

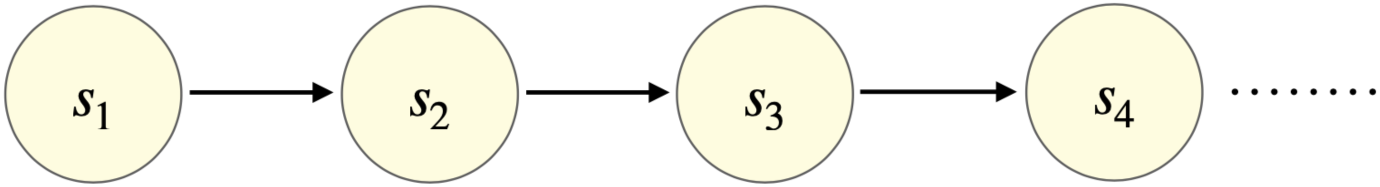
$$P(s_t \mid s_{<t}) \approx P(s_t \mid s_{t-1})$$

QUIZ: MARKOV ASSUMPTION

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

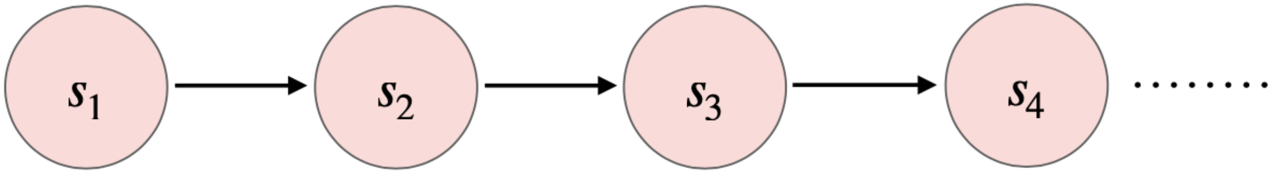
- Where have we seen this before?
  - a) Logistic regression
  - b) Linear regression
  - c) Large language model
  - d) N-gram language model

MARKOV CHAINS



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

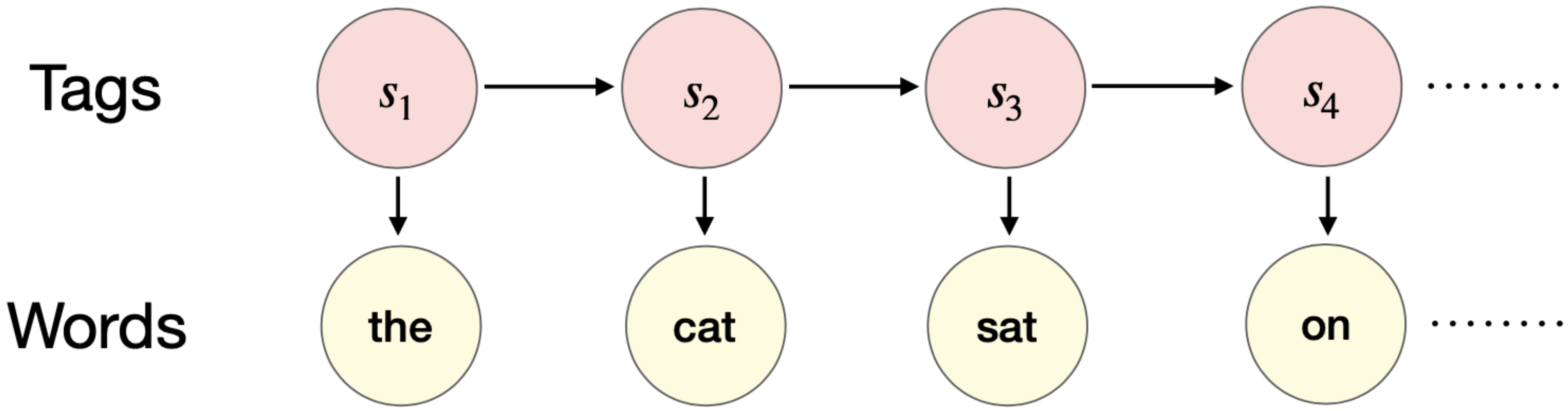
MARKOV CHAINS



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

- We don't know the tags in the corpus.

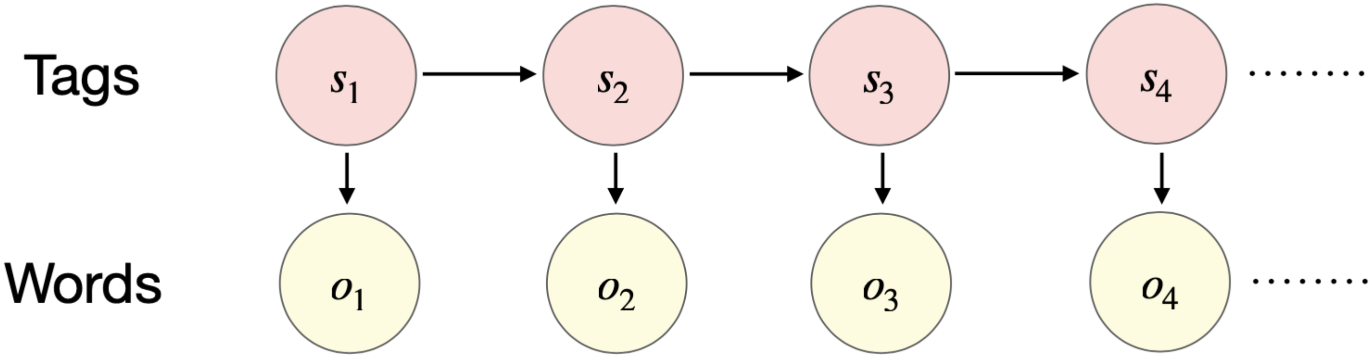
MARKOV CHAINS



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

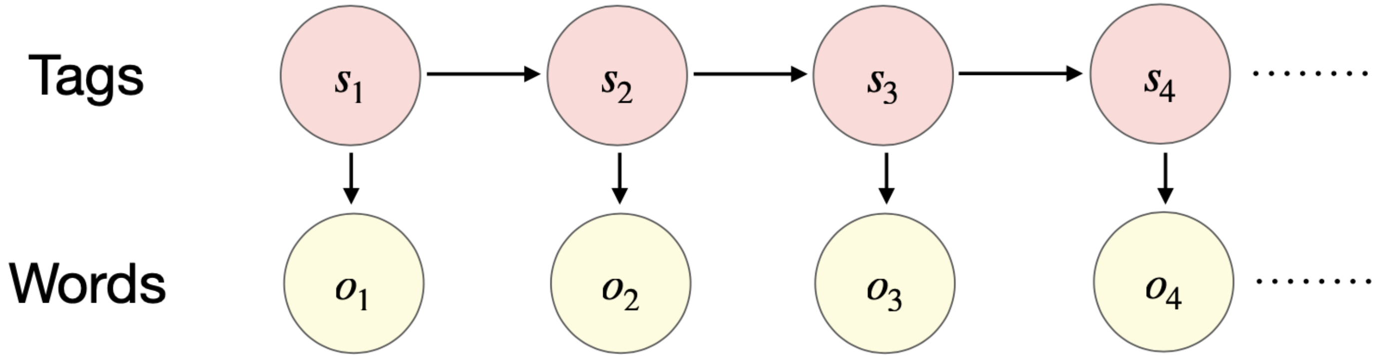
- We don't know the tags in the corpus.
- But we do observe the words!
- HMM allows us to jointly reason over both ***hidden*** and ***observed*** events.

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:

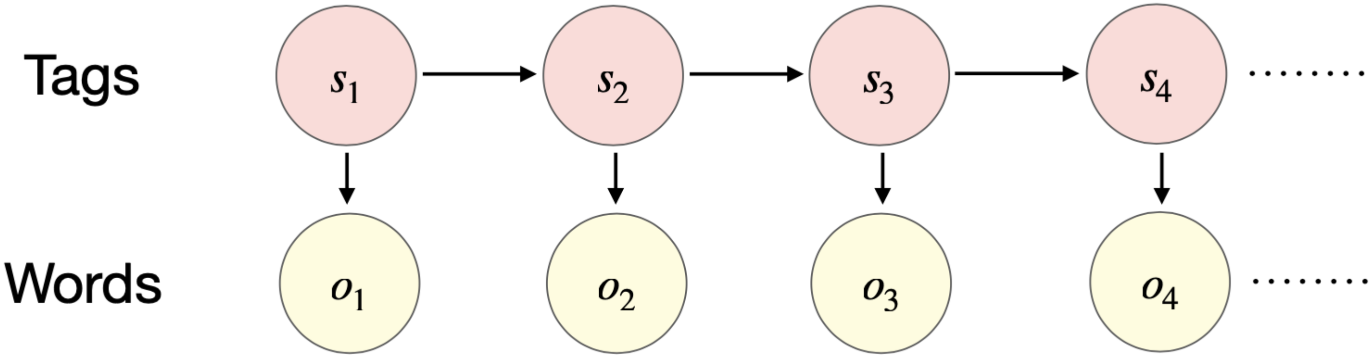
$$P(s_{t+1} \mid s_1, \ldots, s_t) = P(s_{t+1} \mid s_t)$$

2. Output independence assumption:

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

Quiz: Which one of the two assumptions is stronger, and why?

SEQUENCE LIKELIHOOD



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

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

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

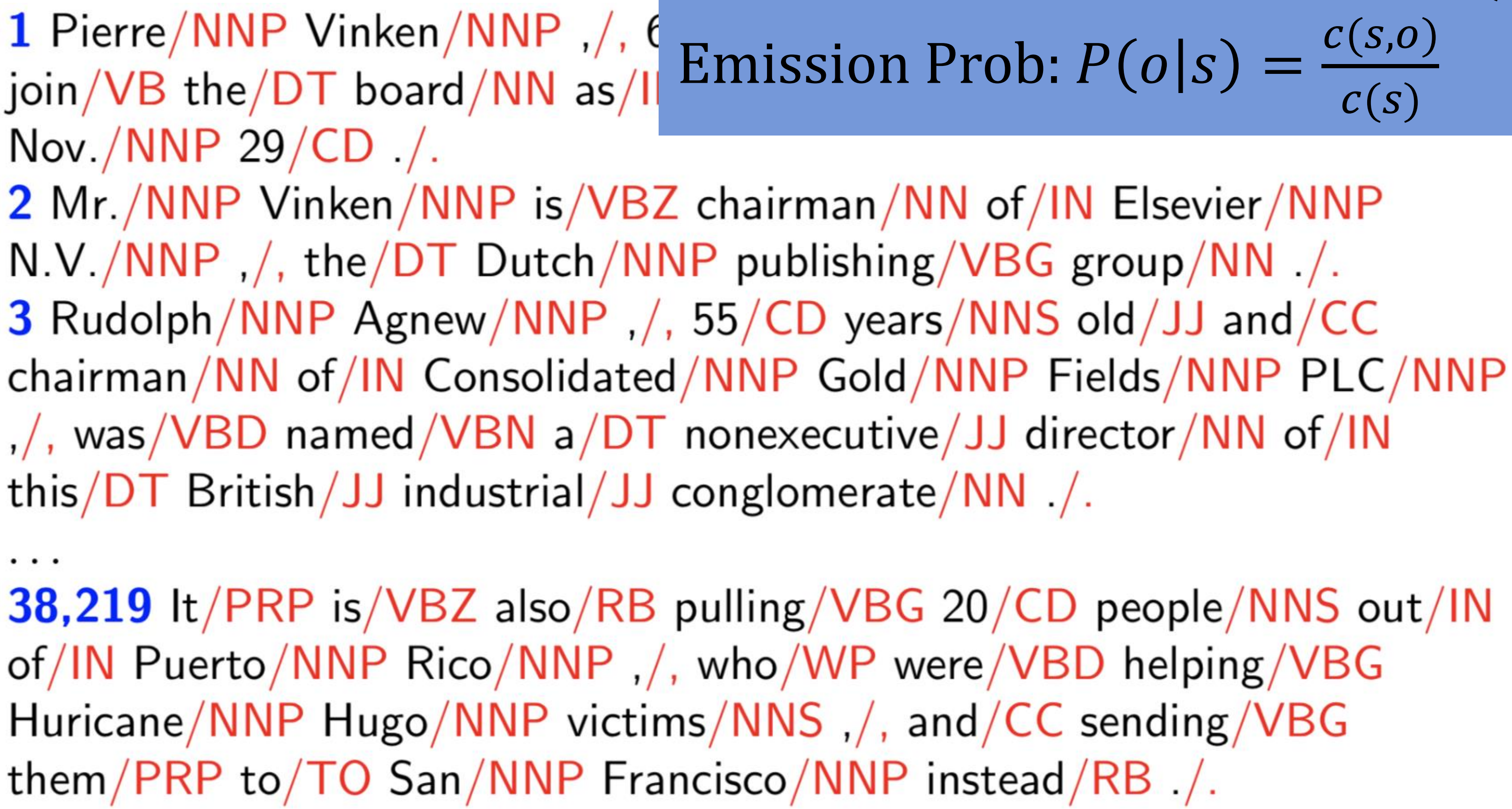
LEARNING

Maximum likelihood estimate:

Transition prob: $P(s_i|s_j) = \frac{c(s_i,s_j)}{c(s_j)}$

Emission Prob: $P(o|s) = \frac{c(s,o)}{c(s)}$

- Training Set:



1 Pierre/NNP Vinken/NNP ,/, 6
join/VB the/DT board/NN as/I
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 ./.

EXAMPLE: POS TAGGING

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

$\pi(DT) = 0.8$

s_{t+1}

s_t

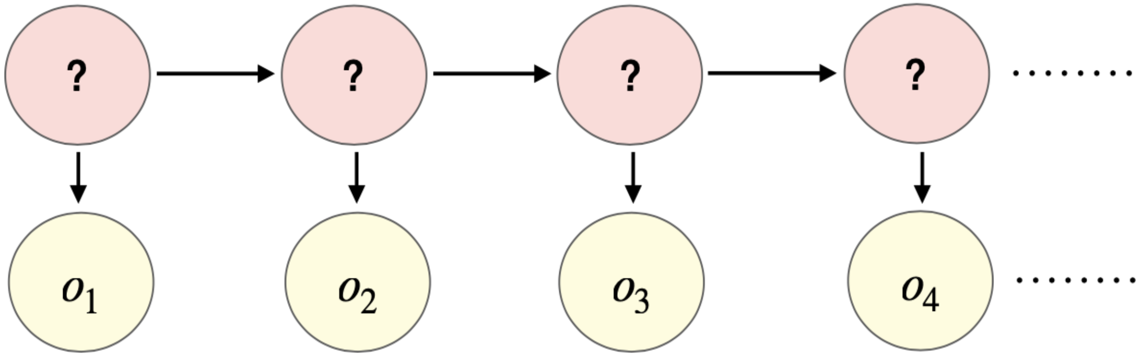
	DT	NN	IN	VBD
DT	0.5	0.8	0.05	0.1
NN	0.05	0.2	0.15	0.6
IN	0.5	0.2	0.05	0.25
VBD	0.3	0.3	0.3	0.1

o_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



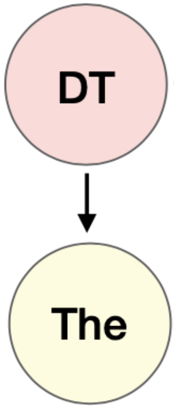
- **Task:** Find the most probable sequence of states $\langle s_1, s_2, \ldots, s_n \rangle$, given the observations $\langle o_1, o_2, \ldots, o_n \rangle$

$$\hat{S} = \underset{S}{argmax}\, P(S|O) = \underset{S}{argmax} \frac{P(S)P(O|S)}{\underset{\nwarrow}{P(O)}} \quad \text{constant}$$

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

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

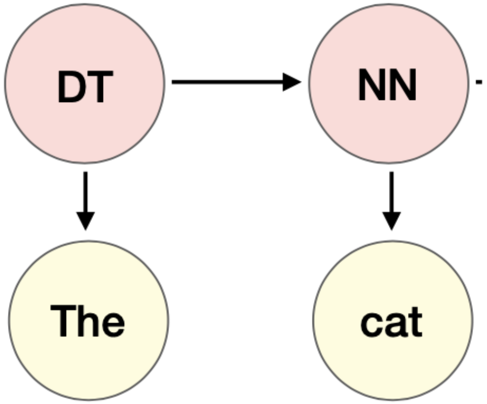
GREEDY DECODING



$$\underset{s}{argmax}\,\Pi(s_1 = s)P(The\,|s) = 'DT'$$

$$\hat{S} = \underset{S}{argmax} \prod_{i=1}^{n} P(s_i|s_{i-1})P(o_i|s_i)$$

GREEDY DECODING

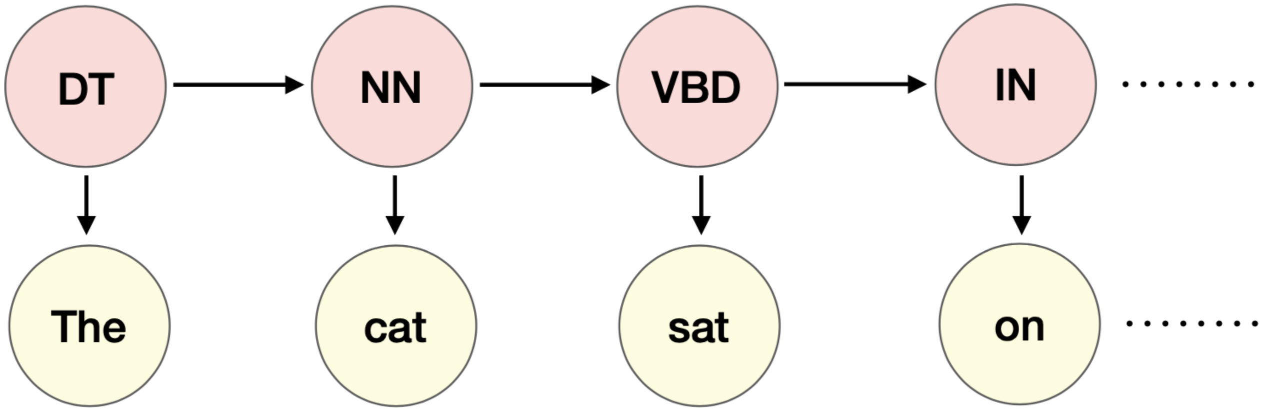


$$\underset{s}{argmax}\,\Pi(s_1 = s)P(The\,|s) = 'DT'$$

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

$$\hat{S} = \underset{S}{argmax} \prod_{i=1}^{n} P(s_i|s_{i-1})P(o_i|s_i)$$

GREEDY DECODING



$$\underset{s}{argmax}\, \Pi(s_1 = s)P(The\ |s) = 'DT'$$

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

$$\forall i, \hat{s}_{i+1} = \underset{s}{argmax}\, P(s|\hat{s}_i)P(o_{i+1}|s)$$

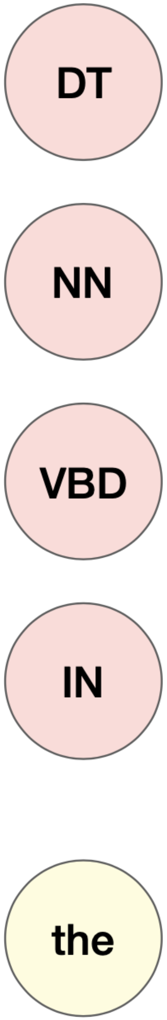
Not guaranteed to be optimal: local decision only!

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 $\boldsymbol{j}$ at time $\boldsymbol{i}$

VITERBI DECODING



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

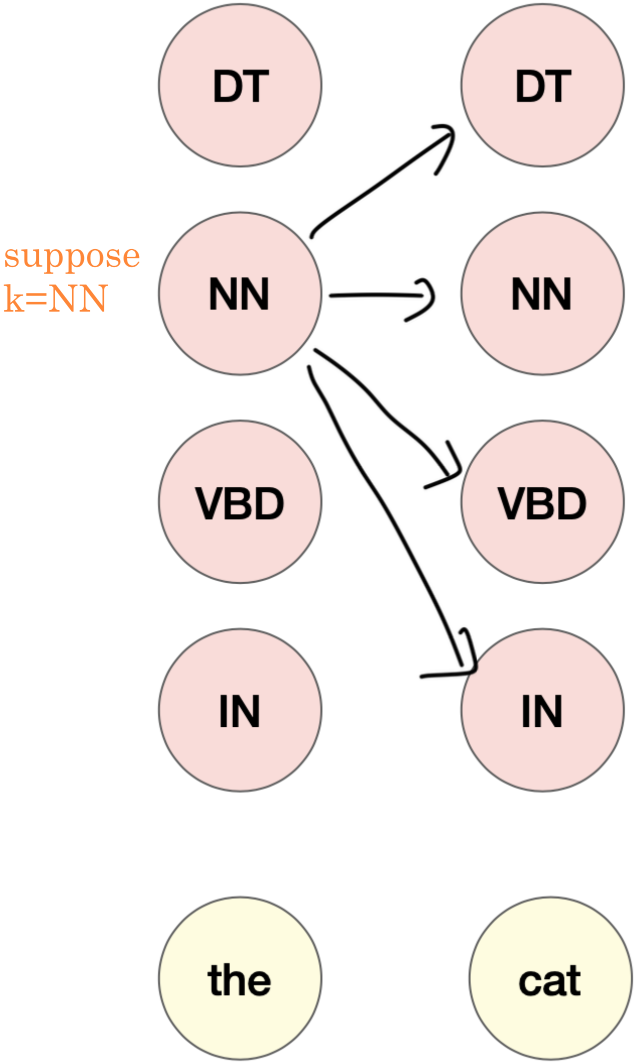
$$M[1,NN] = \pi(NN)\ P(\textbf{the} \mid NN)$$

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

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

Forward →

VITERBI DECODING



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

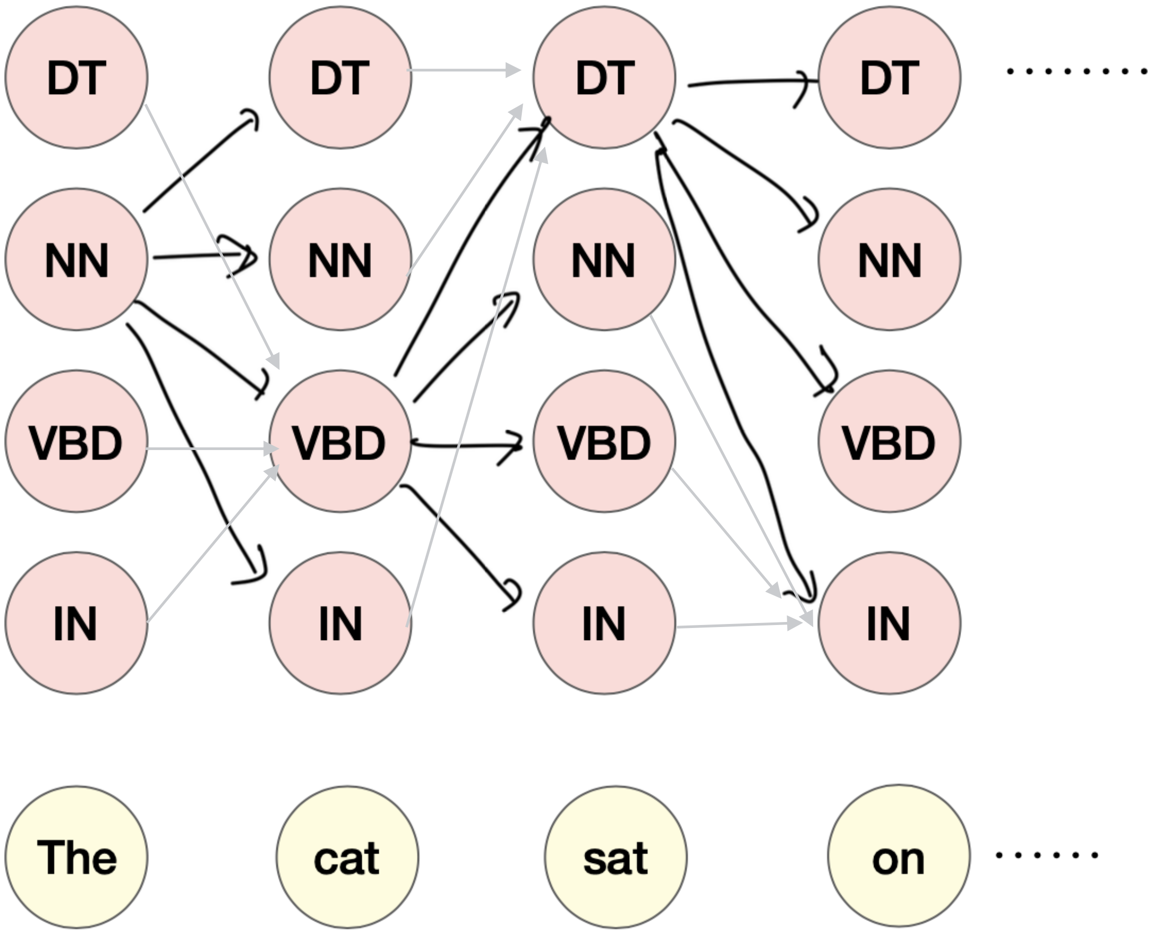
$$M[2,NN] = \max_k M[1,k]\ P(NN\,|\,k)\ P(\textbf{cat}\,|\,NN)$$

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

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

Forward →

VITERBI DECODING



This is a recursive process!

Viterbi Algorithm needs to backtrack.

$$M[i,j] = \max_k M[i-1,k]P(s_j|s_k)P(o_i|s_j) \;\; 1 \le k \le K, 1 \le i \le N$$

QUIZ: VITERBI ALGORITHM

Assume

T : Number of time steps (sequence length)

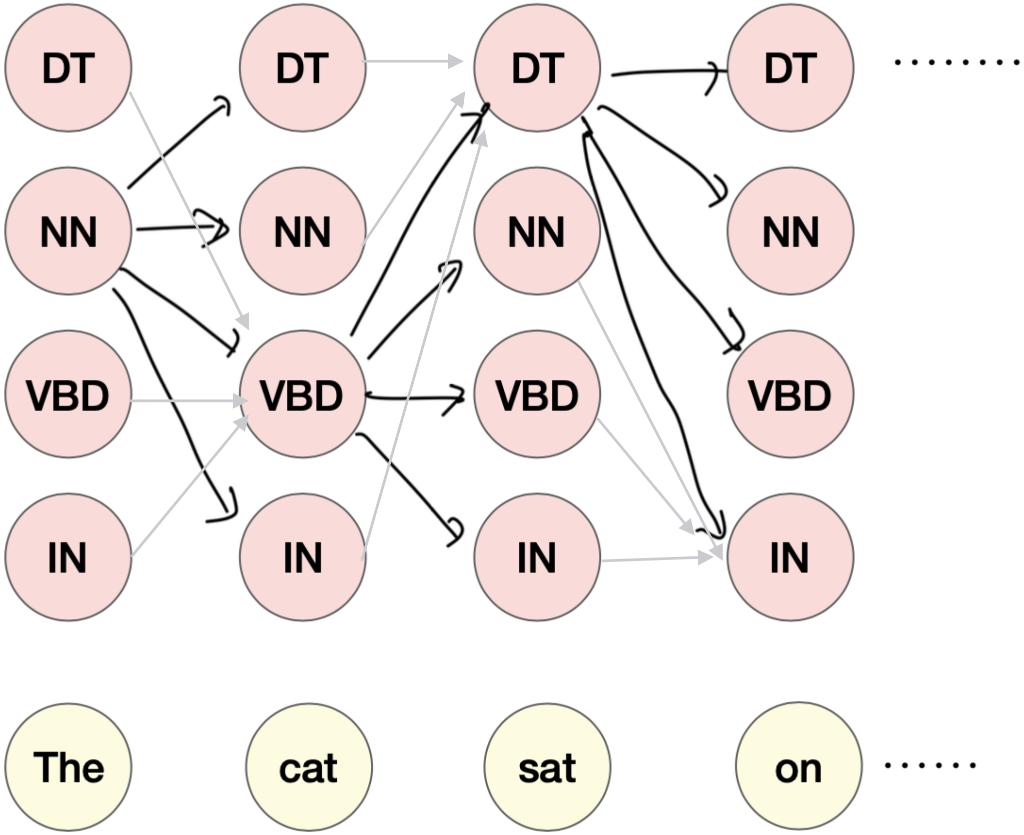
K : Number of states

What is the time complexity of the Viterbi algorithm (in Big O)?

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

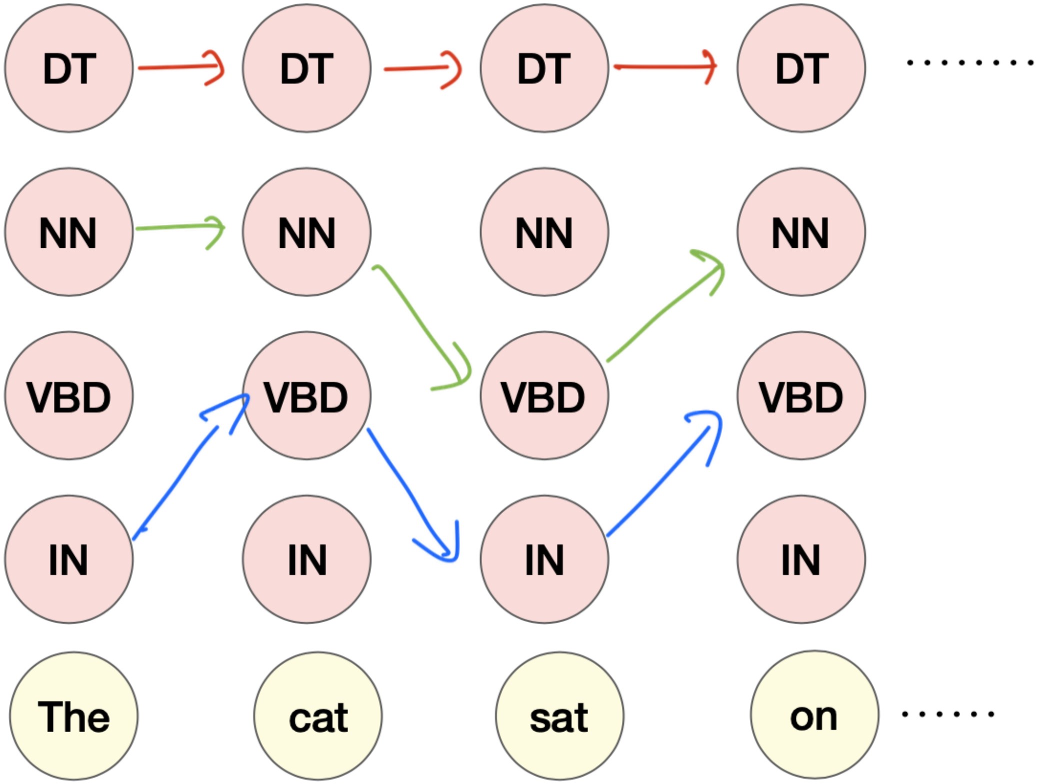
BEAM SEARCH

- When K (the number of states) is large, Viterbi algorithm is very expensive!



BEAM SEARCH

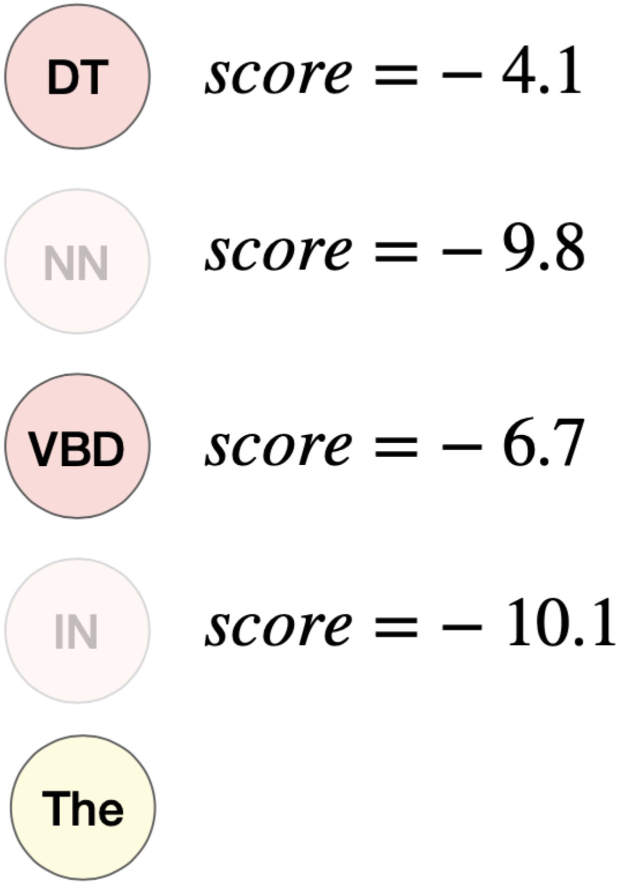
- But many paths have very low likelihood!



BEAM SEARCH

- Keep a fix number β of hypotheses at each stage:

$\beta = 2$



$score = -4.1$

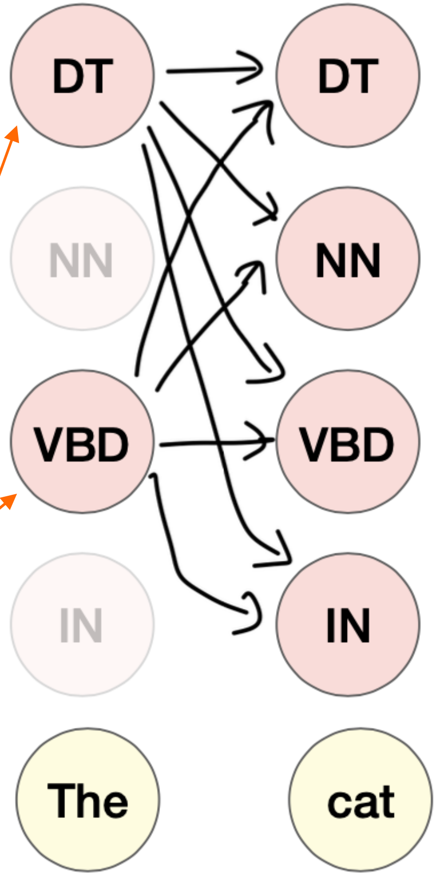
$score = -9.8$

$score = -6.7$

$score = -10.1$

BEAM SEARCH

- Keep a fix number β of hypotheses at each stage:



$score = -16.5$

$score = -6.5$

$score = -13.0$

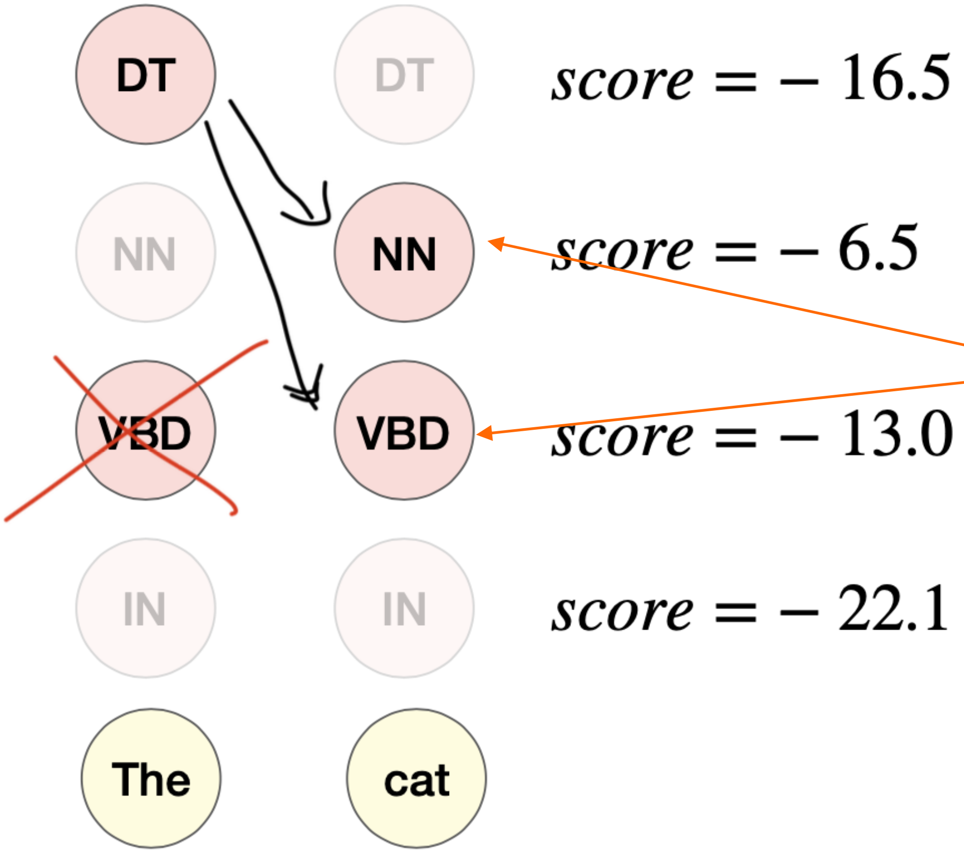
$score = -22.1$

$\beta = 2$

Step 1: Expand all partial sequences in current beam

BEAM SEARCH

- Keep a fix number β of hypotheses at each stage:



$score = -16.5$

$score = -6.5$

$\beta = 2$

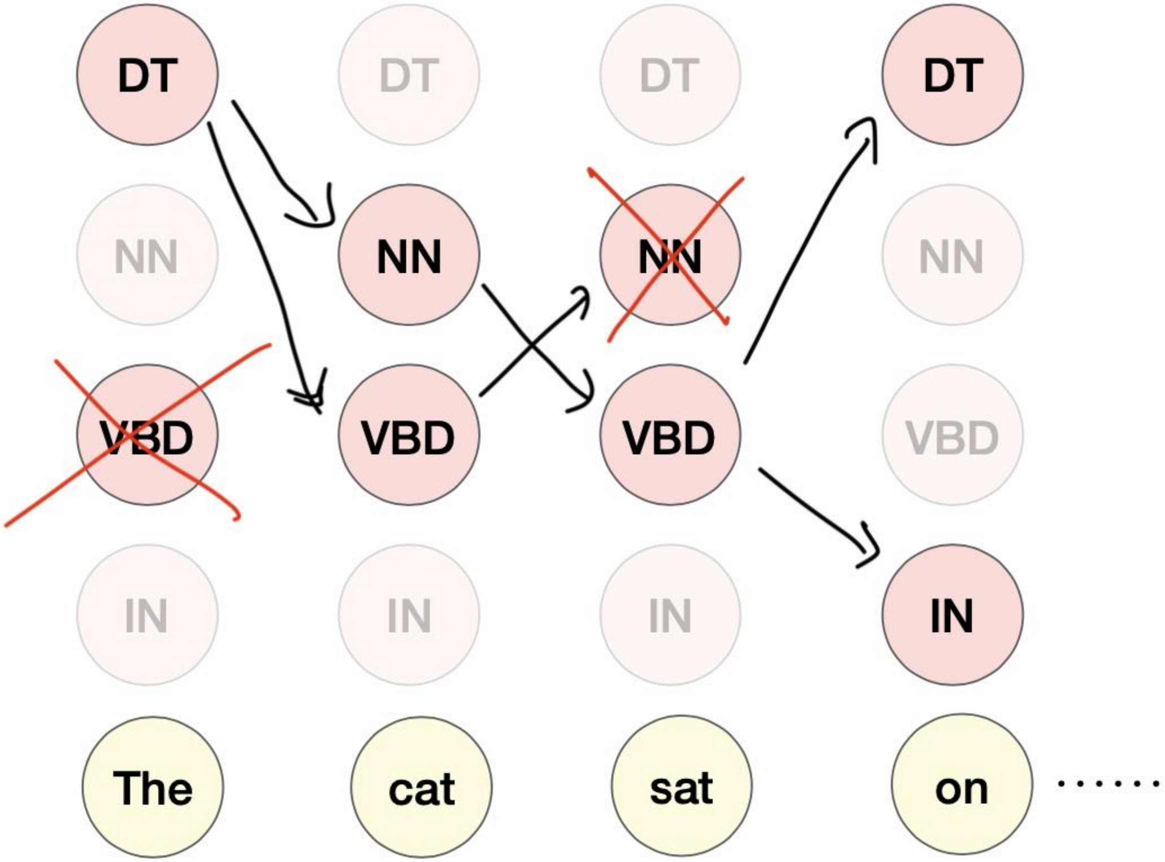
$score = -13.0$

$score = -22.1$

Step 2: Prune set back to top β sequences

BEAM SEARCH

- Keep a fix number β of hypotheses at each stage:



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

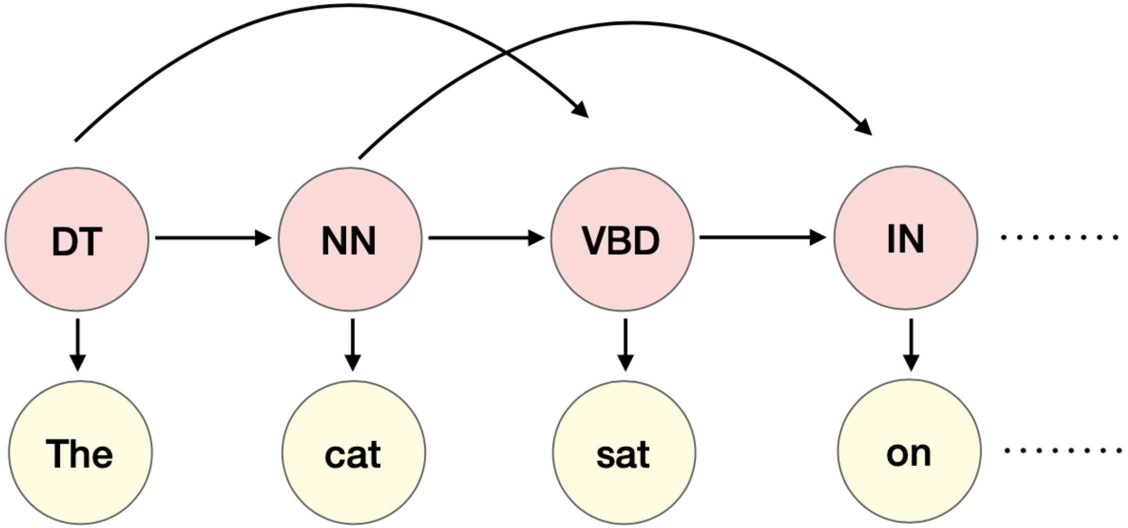
BEAM SEARCH

- If K (number of states) is too large, Viterbi algorithm is too expensive!
- Keep a fixed number of hypotheses at each stage
- Beam width β
- Trade-off (some) accuracy for efficiency

Quiz: What is the time complexity of Beam Search Viterbi Algorithm, given sequence length T, number of states K, and β?

BEYOND BIGRAMS

- Real-world HMM taggers have more relaxed assumptions.
- Tri-gram HMM: $P(s_{t+1}|s_1, s_2, \dots, s_t) = P(s_{t+1}|s_{t-1}, s_t)$



Pros?

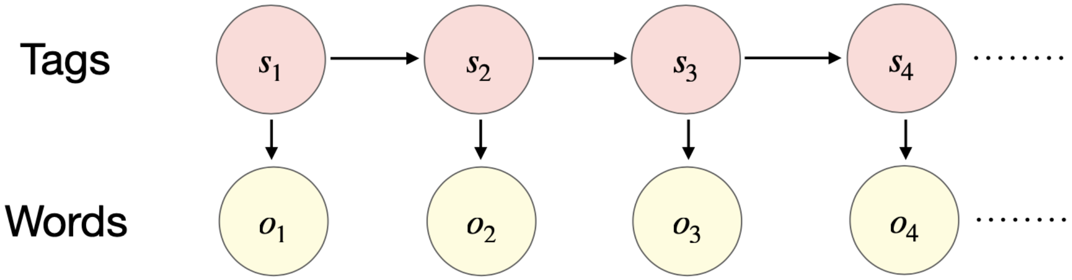
Cons?

LIMITATIONS OF HMM

- HMM is a generative model: $P(O \mid S)$
- Unknown (OOV) words happen often
- HMM relies on a fixed vocabulary (fixed-size emission probability matrix)
- Can't add arbitrary features easily
- Remember log-linear models (LR) can combine arbitrary models?
- But LR is is not a sequential model
- Enter the *Conditional Random Field*!
  - Discriminative model: $P(S \mid O)$

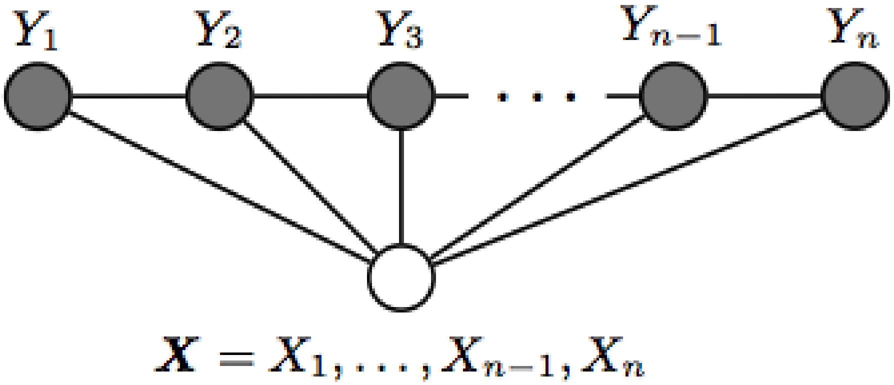
LINEAR CHAIN CRF

- HMM:



$$\hat{S} = \underset{S}{argmax} \prod_{i=1}^{n} P(s_i|s_{i-1})P(o_i|s_i)$$

- Linear chain CRF:



$$\hat{Y} = \underset{Y \in \mathcal{Y}}{\text{argmax}}\, P(Y|X)$$

y is the set of all possible tag sequences

LINEAR CHAIN CRF

- Assigns a probability of the entire tag sequence Y, out of all possible sequences $\mathcal{Y}$.
- A giant version of multinomial logistic regression for a single token.

$$p(Y|X) = \frac{\exp\left(\sum_{k=1}^{K} w_k F_k(X,Y)\right)}{\sum_{Y' \in \mathcal{Y}} \exp\left(\sum_{k=1}^{K} w_k F_k(X,Y')\right)}$$

- F_k is the kth feature function mapping X→Y
- K is total number of features

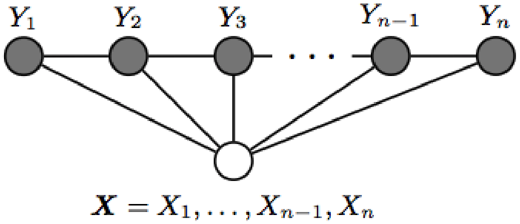
LINEAR CHAIN CRF

- Rename the denominator as a function Z(X):

$$
\begin{aligned}
p(Y|X) &= \frac{1}{Z(X)} \exp\left(\sum_{k=1}^{K} w_k F_k(X,Y)\right) \\
Z(X) &= \sum_{Y' \in \mathcal{Y}} \exp\left(\sum_{k=1}^{K} w_k F_k(X,Y')\right)
\end{aligned}
$$

- Global feature $F_k(X,Y)$ can be decomposed into a sequence of local features, where n is the length of the token sequence:

$$F_k(X,Y) = \sum_{i=1}^{n} \underbrace{f_k(y_{i-1}, y_i, X, i)}_{\text{linear chain! But no directions!}}$$



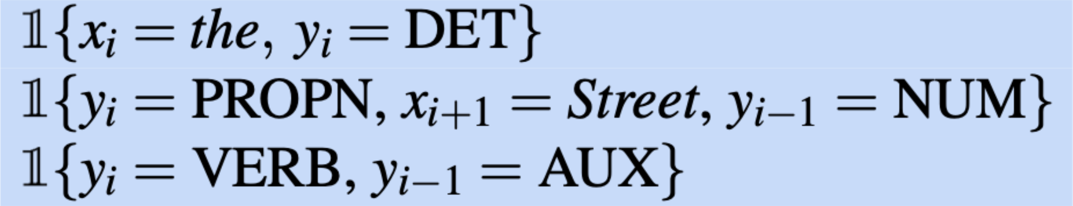
Sum: no directions

FEATURES IN LINEAR CHAIN CRF

- Discriminative models allow for many features
- Each feature f_k depends on any info from

$$(y_{i\text{-}1}, y_i, X, i)$$

- Example features:



$$\mathbb{1}\{x_i = \textit{the},\ y_i = \text{DET}\}$$
$$\mathbb{1}\{y_i = \text{PROPN},\ x_{i+1} = \textit{Street},\ y_{i-1} = \text{NUM}\}$$
$$\mathbb{1}\{y_i = \text{VERB},\ y_{i-1} = \text{AUX}\}$$

- Use feature template to extract features for each position i:

$$\langle y_i, x_i\rangle, \langle y_i, y_{i-1}\rangle, \langle y_i, x_{i-1}, x_{i+2}\rangle$$