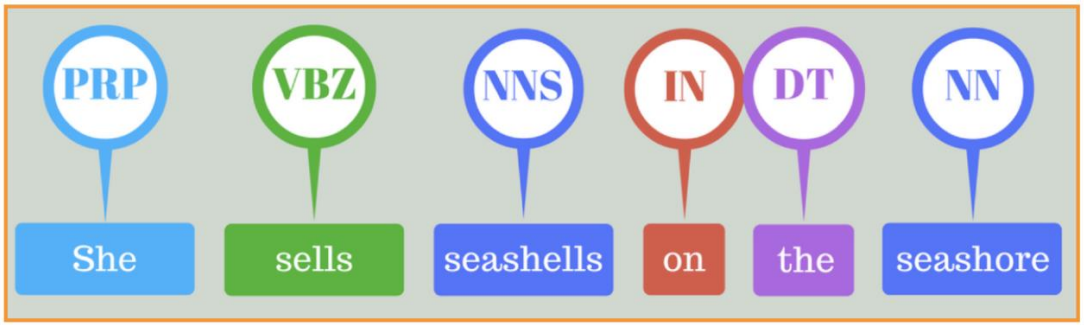
# CSE 4392 Special Topics
# Natural Language Processing

# Sequence Models



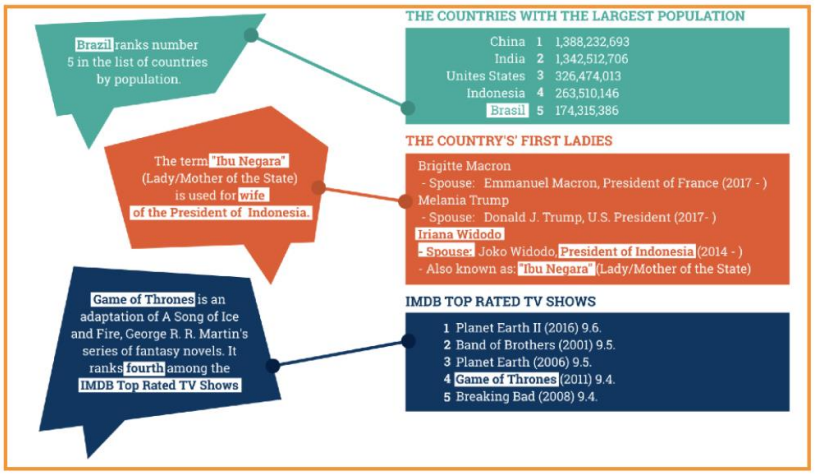
2025 Spring

# Why Model Sequences?




Part-of-speech tagging

Name Entity Recognition


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!)

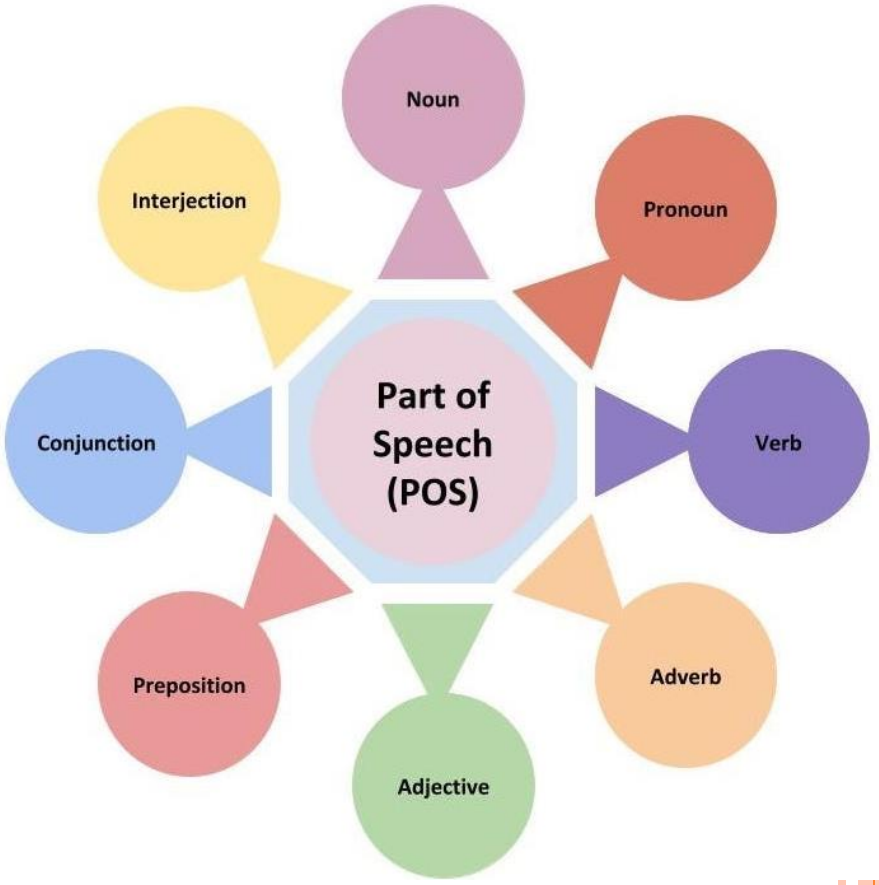
The/DT cat/NN sat/VBD on/IN the/DT mat/NN

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

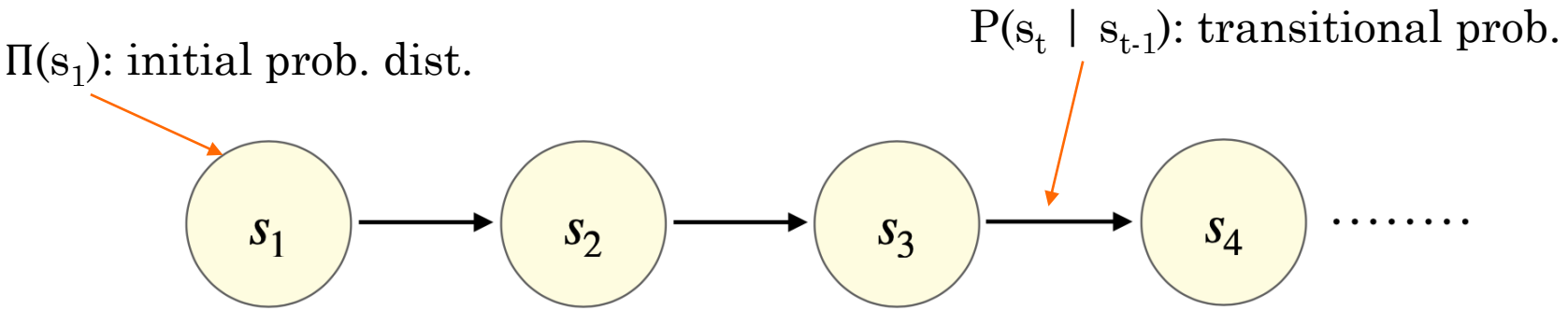
- Most words are easy to disembiguate
- **Most frequence class**: assign each word (token) its most frequently used class in the training set. (e.g., man/NN)
- Accuracy: 92.34% on the Wall Street Journal (WSJ) dataset!
- State of the art: ~ 97%
- Average English sentence: ~ 14 words
  - Sentence level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

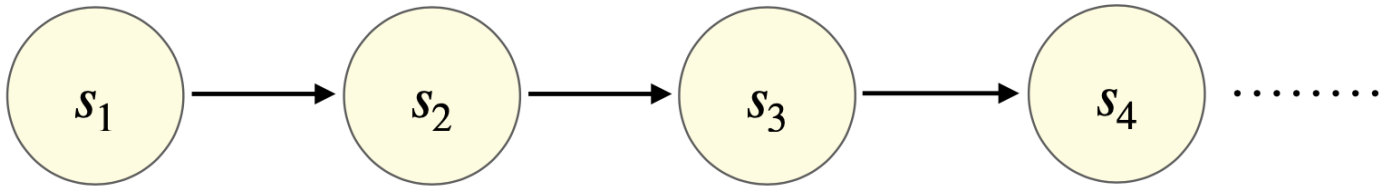
  a) Logistic regression

  b) Linear regression

  c) Large language model

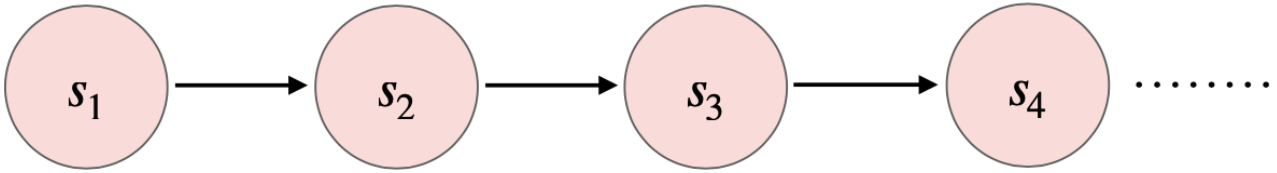
  d) N-gram language model

# Markov Chains




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

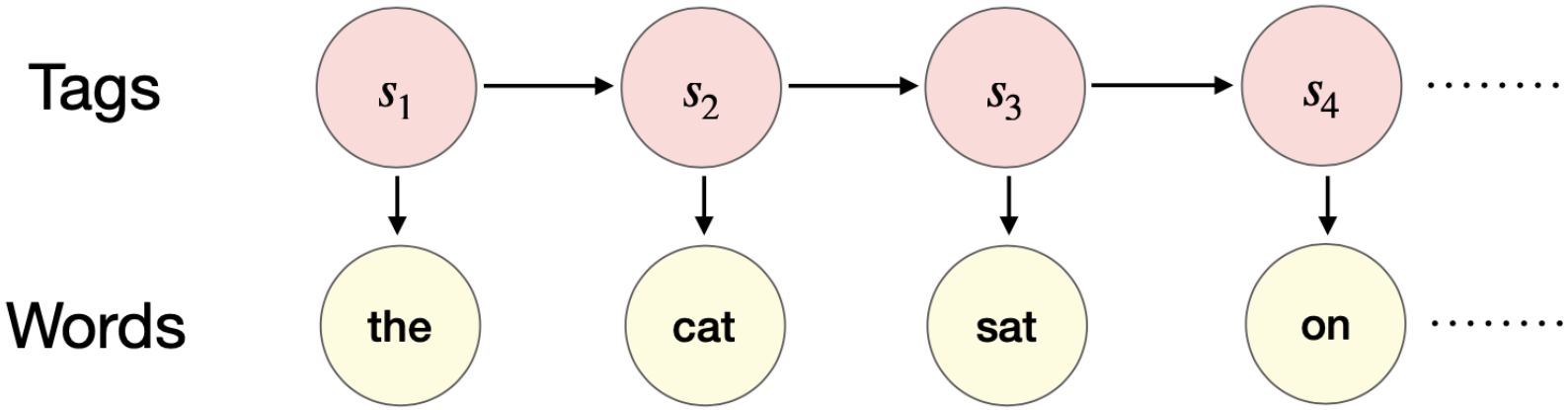
# Markov Chains




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

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

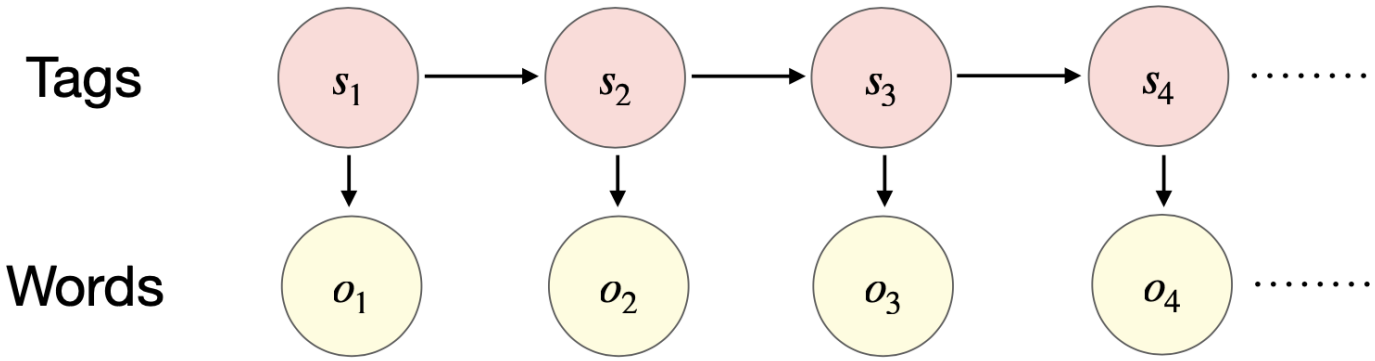
# Markov Chains




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

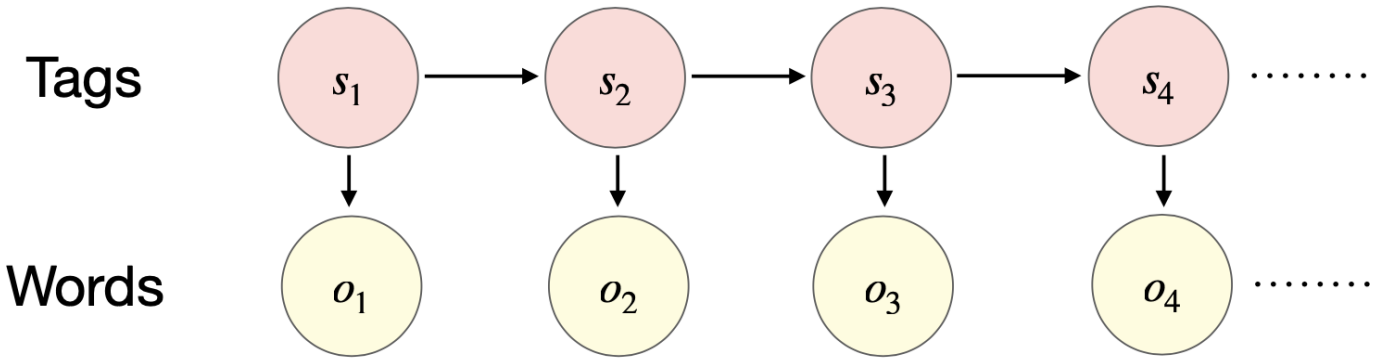
- We don't know the tags in the corpus.
- But we do observe the words!
- HMM allows us to jointly reason over both ***hidden*** and ***observed*** events.

# 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:

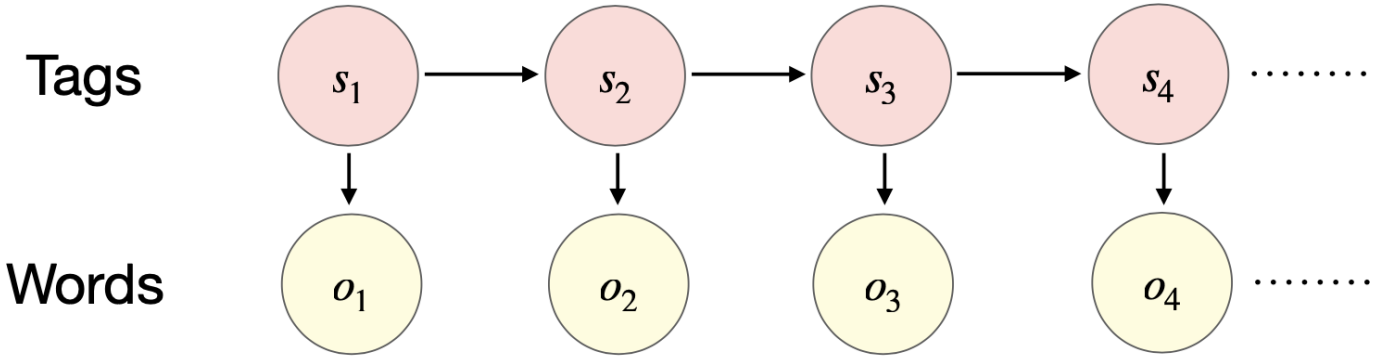
$$P(s_{t+1} \mid s_1, \ldots, s_t) = P(s_{t+1} \mid s_t)$$

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




$$\mathrm{P}(\mathrm{S},\mathrm{O}) = \mathrm{P}(\mathrm{s}_1,\mathrm{s}_2,\ldots,\mathrm{s}_\mathrm{n},\mathrm{o}_1,\mathrm{o}_2,\ldots,\mathrm{o}_\mathrm{n})$$

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

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

# Learning

- Training Set:

1 Pierre/NNP Vinken/NNP ,/, 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 ./.

Maximum likelihood estimate:

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

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

# EXAMPLE: POS TAGGING

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

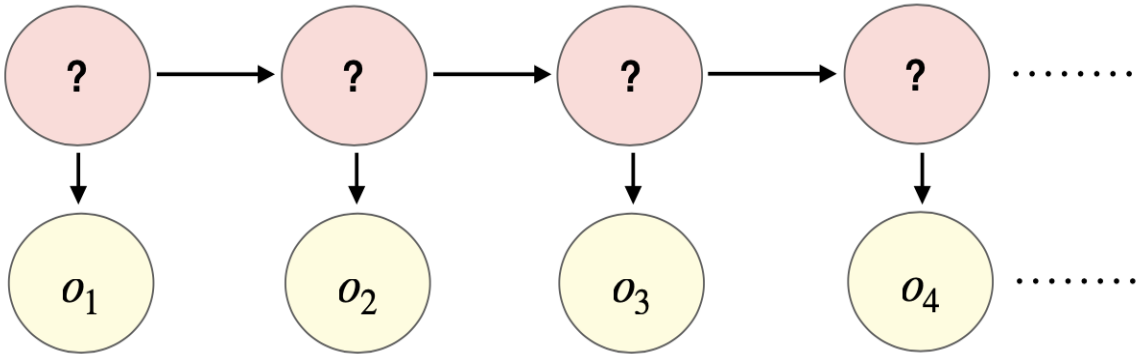
$\pi(DT) = 0.8$ $s_{t+1}$ $o_t$

| $s_t$ | 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 |

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 $\langle o_1, o_2, \ldots, o_n \rangle$

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

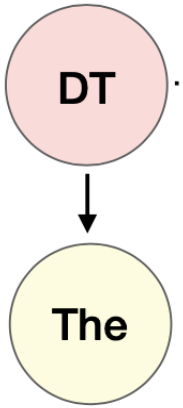
constant

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

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

transition emission

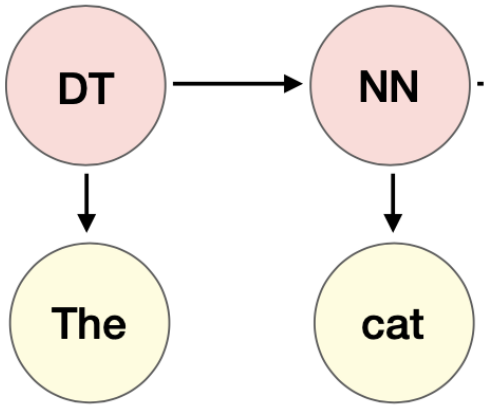
# Greedy Decoding




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

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

# Greedy Decoding

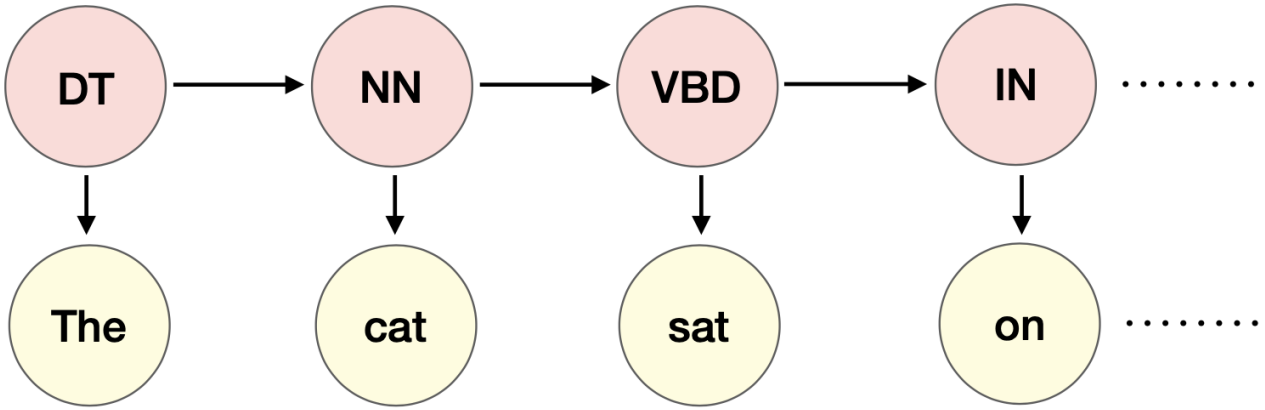



$$\underset{s}{argmax}\ \Pi(s_1 = s)P(The\ |s) = 'DT'$$

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

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

# Greedy Decoding




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

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

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

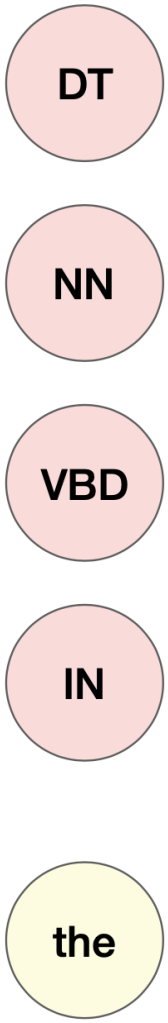
Not guaranteed to be optimal: local decision only!

# 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 ***j*** at time ***i***

# Viterbi Decoding




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

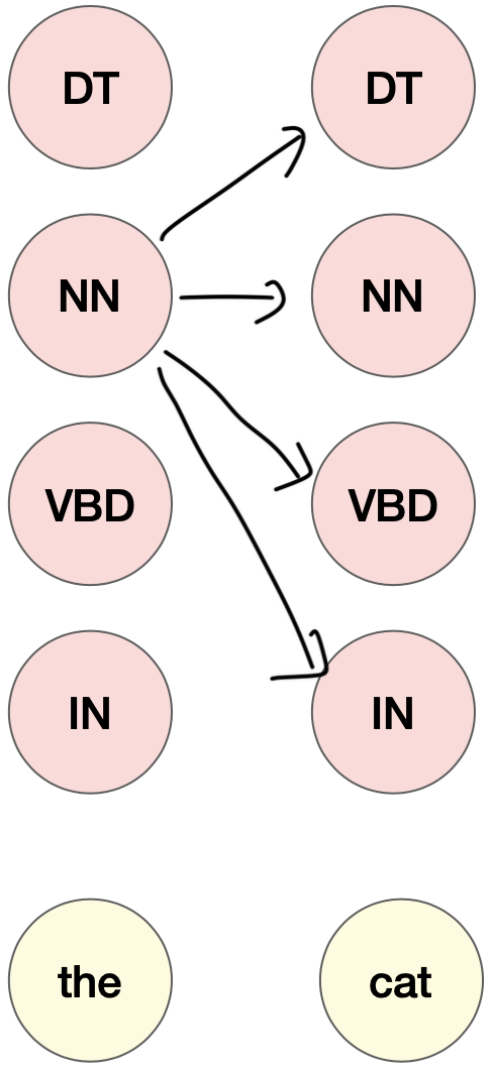
$$M[1,NN] = \pi(NN)\ P(\textbf{the}\,|\,NN)$$

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

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

Forward →

# Viterbi Decoding




suppose k=NN

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

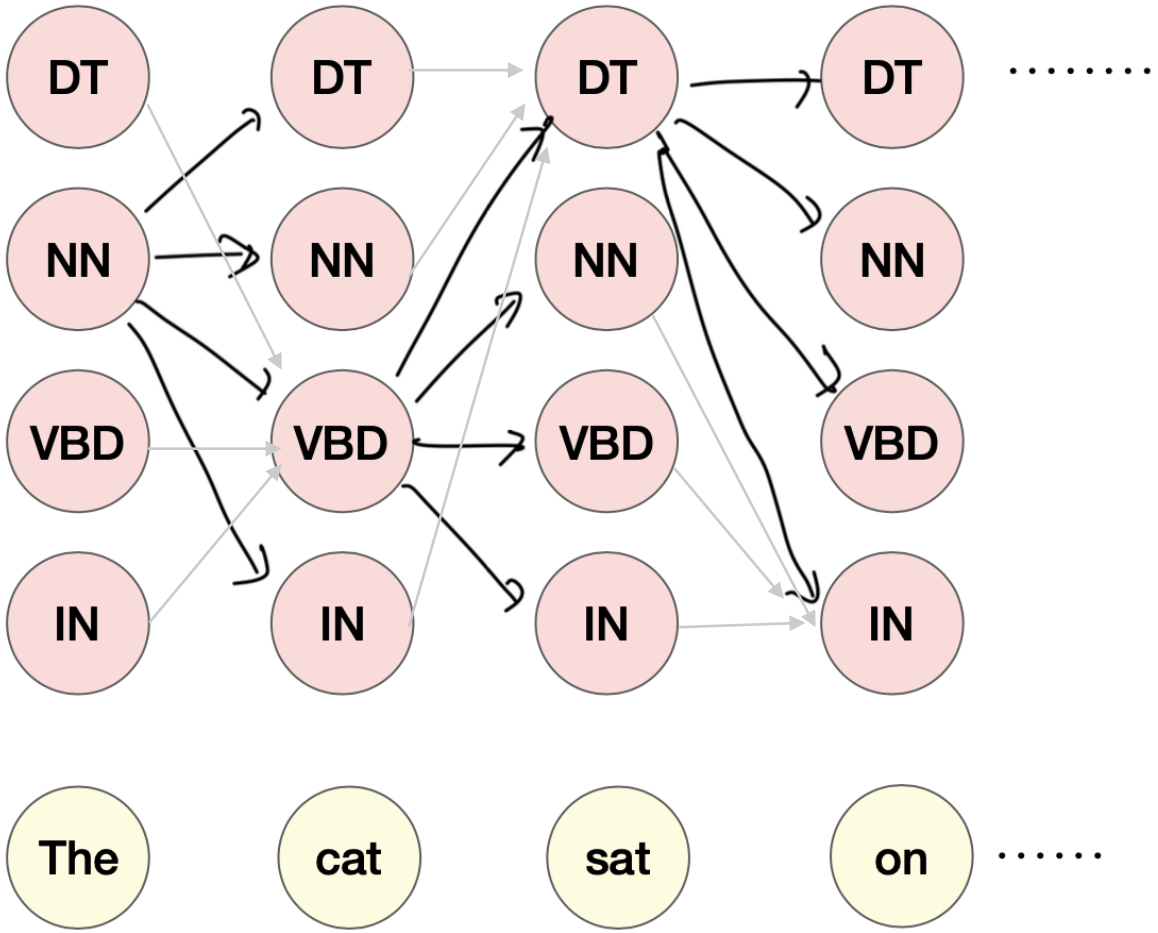
$$M[2,NN] = \max_k M[1,k]\ P(NN|k)\ P(\textbf{cat}|NN)$$

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

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

Forward →

# Viterbi Decoding




This is a recursive process!

Viterbi Algorithm needs to backtrack.

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

# Quiz: Viterbi Algorithm

Assume

$T$ : Number of time steps (sequence length)

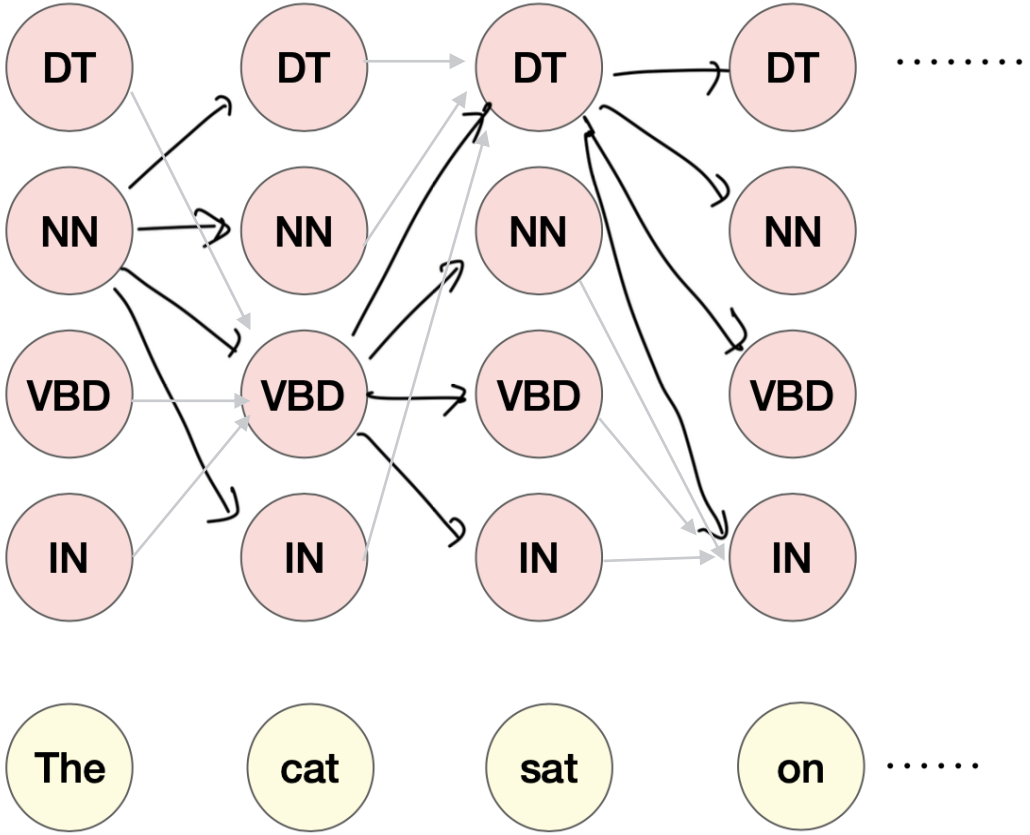
$K$ : Number of states

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

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

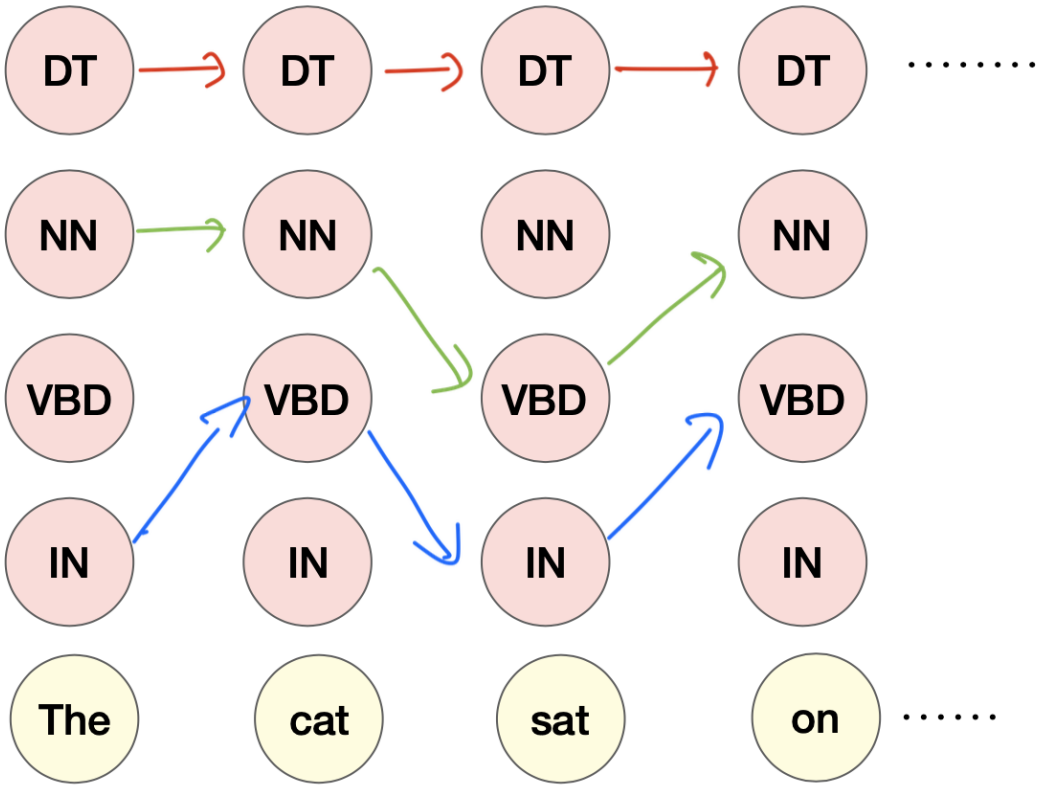
# Beam Search

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



# Beam Search

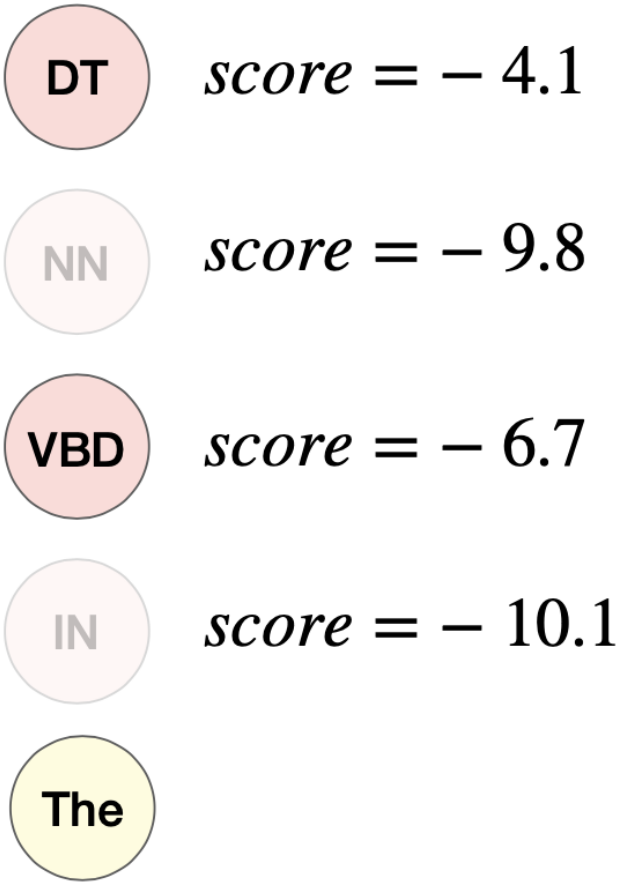
- But many paths have very low likelihood!



# Beam Search

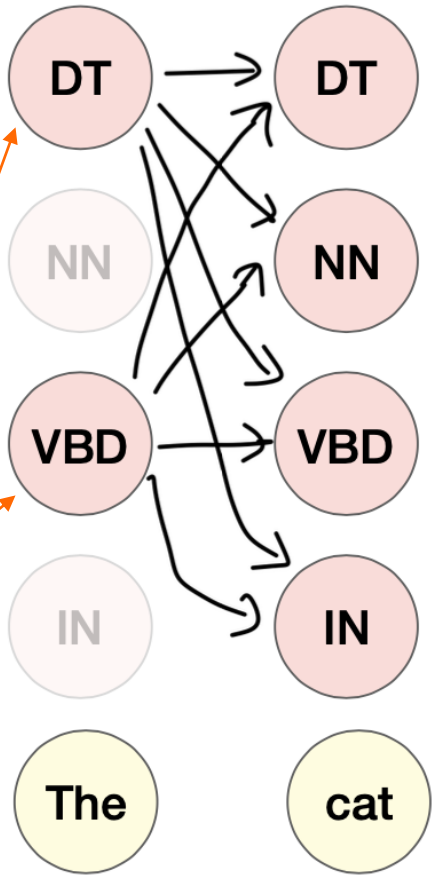
- Keep a fix number $\beta$ of hypotheses at each stage:

$\beta = 2$



# Beam Search

- Keep a fix number $\beta$ of hypotheses at each stage:




$\beta = 2$

$score = -16.5$

$score = -6.5$

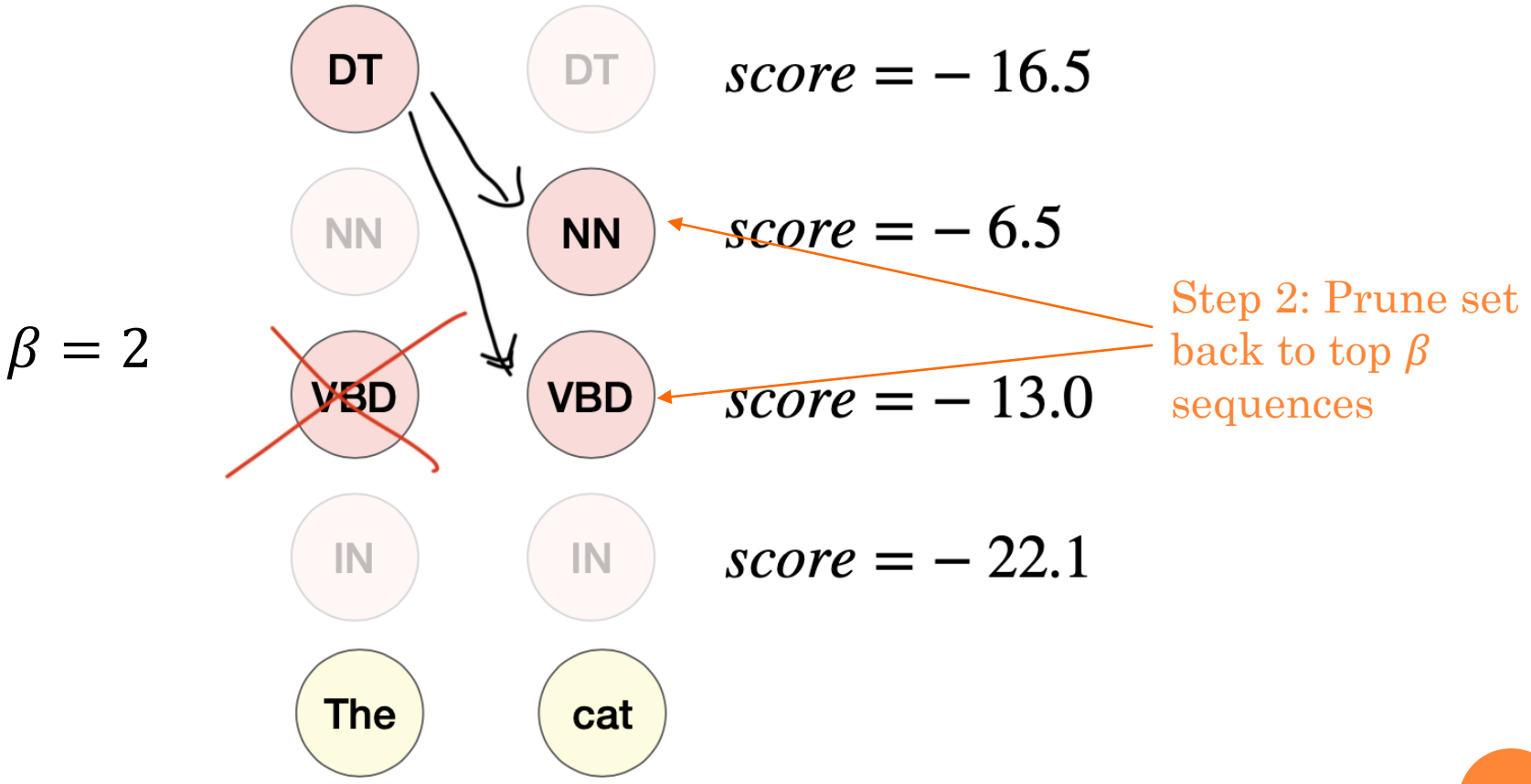
$score = -13.0$

$score = -22.1$

Step 1: Expand all partial sequences in current beam

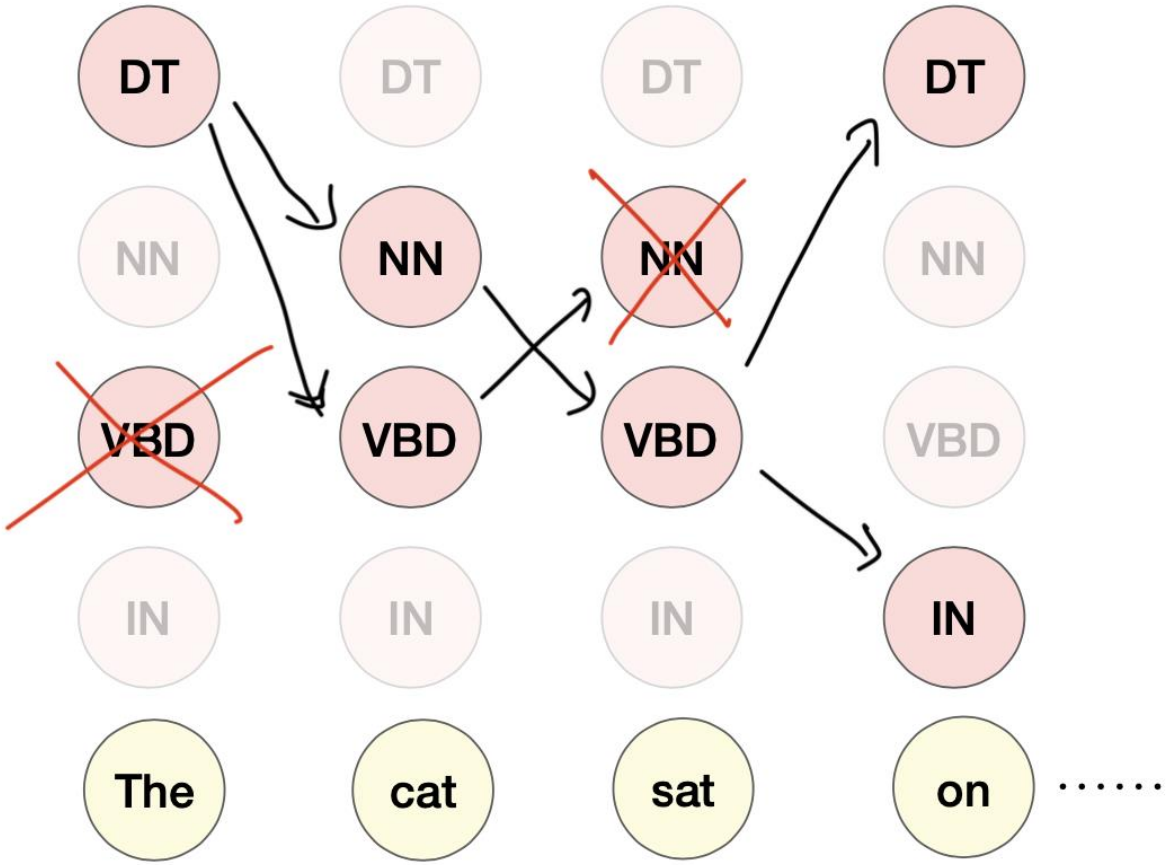
# Beam Search

- Keep a fix number $\beta$ of hypotheses at each stage:



# Beam Search

- Keep a fix number $\beta$ of hypotheses at each stage:




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