COS 484

Natural Language Processing

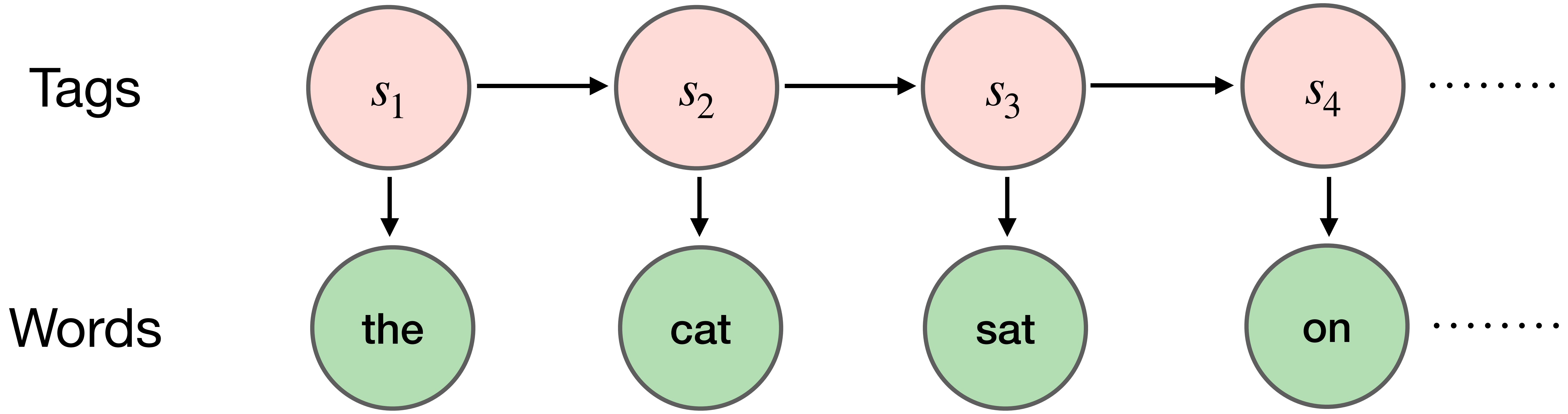
L7: Sequence Models - 2

Spring 2025

Announcements

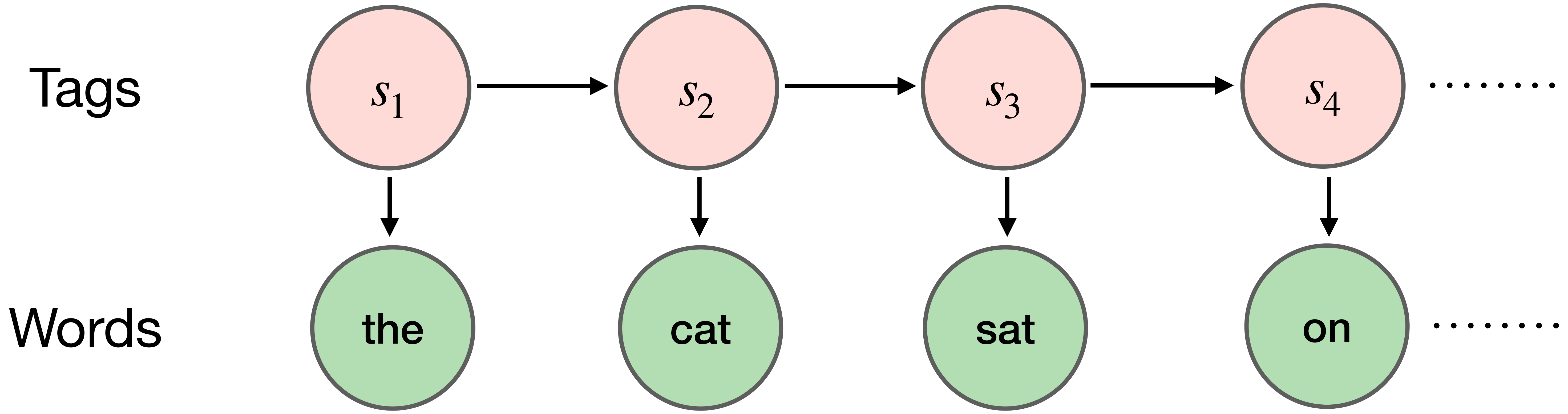
- Reminder! A1 was due today
- A2 will be released later today (due: Mar 3rd)
  - Covering word embedding, HMMs, MEMMs

Recap: Hidden Markov models



1. Set of states S = {1, 2, ..., K} and set of observations $O = \{o_1, \ldots, o_n\}$
2. **Initial state probability distribution** $\pi(s_1)$
3. **Transition probabilities** $P(s_{t+1} \mid s_t)$
4. **Emission probabilities** $P(o_t \mid s_t)$

Recap: Hidden Markov models



1. **Markov assumption**:

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

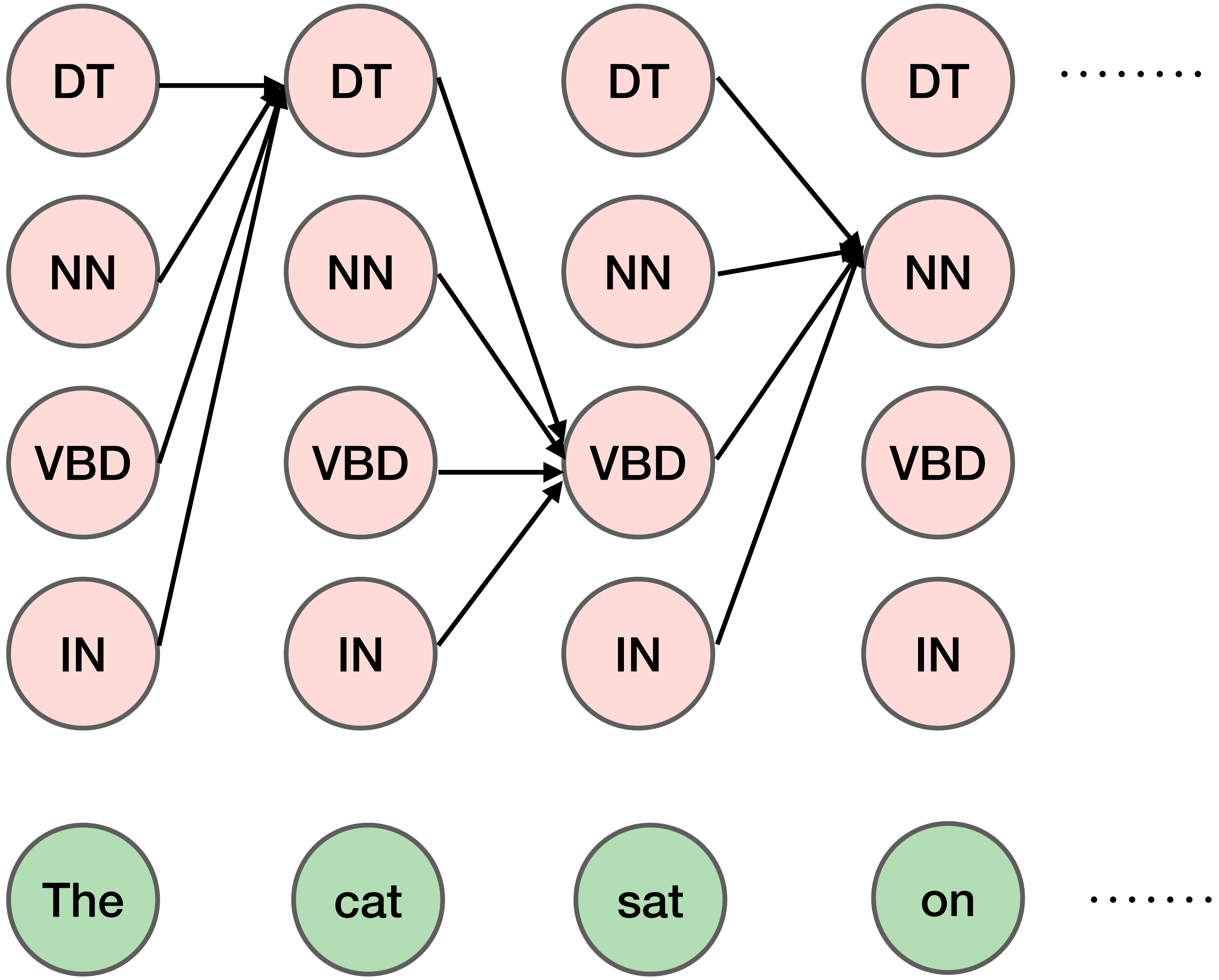
2. **Output independence**:

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

1) assumes (**s)**tate sequences do not have very strong priors/long-range dependencies

2) assumes neighboring (**s)**tates don't affect current (**o)**bservation

Recap: Viterbi decoding



$M[i, j]$ stores joint probability of most probable sequence of states ending with state *j* at time *i*

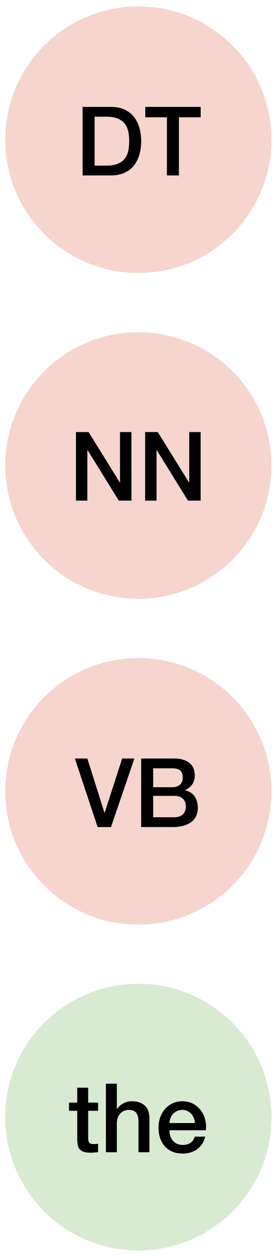
$$M[i, j] = \max_k M[i-1, k]\ P(s_j \mid s_k)\ P(o_i \mid s_j) \quad 1 \leq k \leq K \quad 1 \leq i \leq n$$

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

Viterbi decoding example

- Suppose we want to compute tags for the phrase “the cat runs”
- We need to compute the M and B matrices

$$M[1, s] = P(s \,|\, \varnothing) \times P(o_1 \,|\, s)$$



	DT	NN	VB
$\varnothing$	0.5	0.3	0.2
DT	0.1	0.5	0.4
NN	0.2	0.3	0.5
VB	0.4	0.3	0.3

	the	cat	runs
DT	0.4	0.5	0.1
NN	0.5	0.4	0.1
VB	0.2	0.3	0.5

Viterbi decoding example

- Suppose we want to compute tags for the phrase “the cat runs”
- We need to compute the M and B matrices

$$M[1, s] = P(s \mid \varnothing) \times P(o_1 \mid s)$$

DT $M[1, \text{DT}] = P(\text{DT} \mid \varnothing) \times P(o_1 \mid \text{DT}) = 0.5 \times 0.4$

NN

VB

the

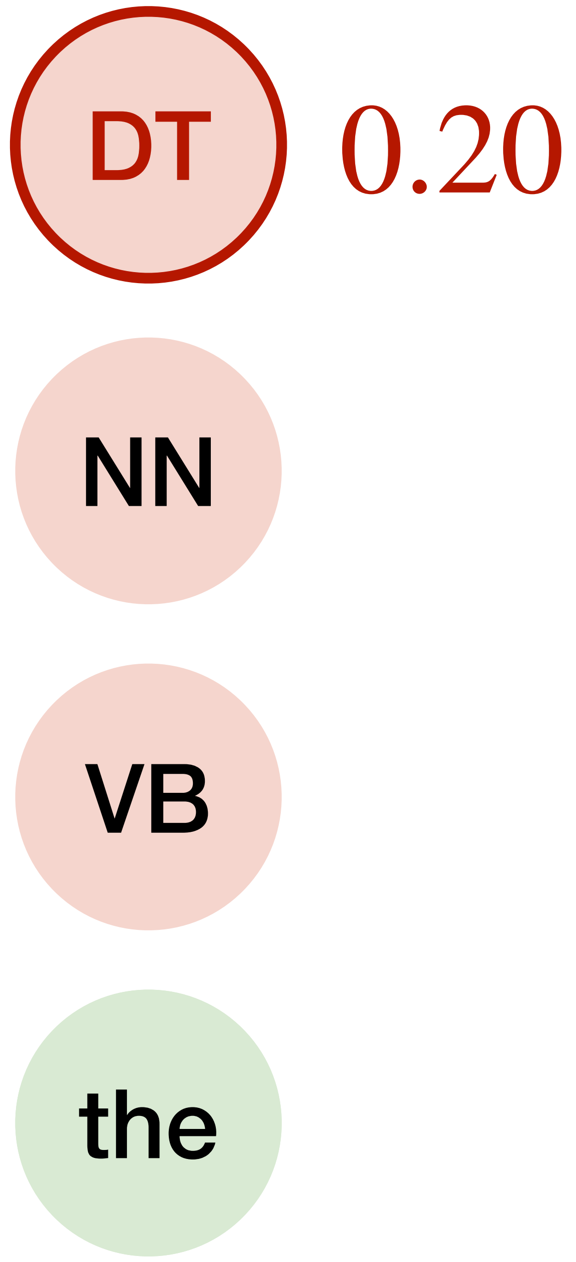
	DT	NN	VB
∅	0.5	0.3	0.2
DT	0.1	0.5	0.4
NN	0.2	0.3	0.5
VB	0.4	0.3	0.3

	the	cat	runs
DT	0.4	0.5	0.1
NN	0.5	0.4	0.1
VB	0.2	0.3	0.5

Viterbi decoding example

- Suppose we want to compute tags for the phrase “the cat runs”
- We need to compute the M and B matrices

$$M[1, s] = P(s \,|\, \varnothing) \times P(o_1 \,|\, s)$$



	DT	NN	VB
∅	0.5	0.3	0.2
DT	0.1	0.5	0.4
NN	0.2	0.3	0.5
VB	0.4	0.3	0.3

	the	cat	runs
DT	0.4	0.5	0.1
NN	0.5	0.4	0.1
VB	0.2	0.3	0.5

Viterbi decoding example

- Suppose we want to compute tags for the phrase “the cat runs”
- We need to compute the M and B matrices

$$M[1,\, s] = P(s \,|\, \varnothing) \times P(o_1 \,|\, s)$$

DT 0.20

NN $M[1, \text{NN}] = P(\text{NN} \,|\, \varnothing) \times P(o_1 \,|\, \text{NN}) = 0.3 \times 0.5$

VB

the

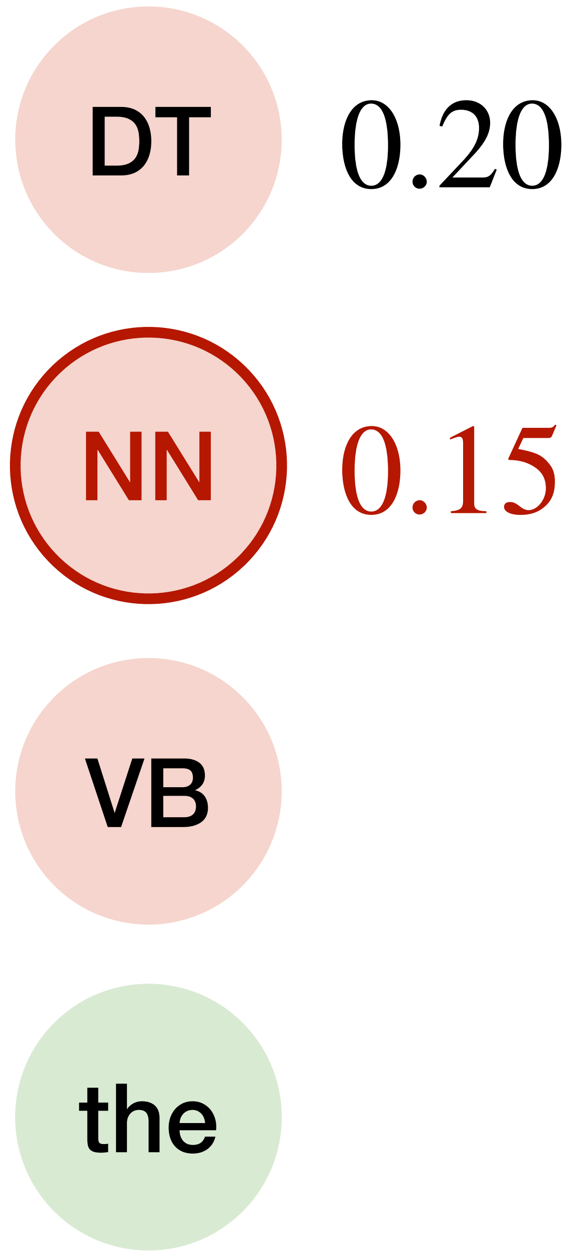
	DT	NN	VB
∅	0.5	0.3	0.2
DT	0.1	0.5	0.4
NN	0.2	0.3	0.5
VB	0.4	0.3	0.3

	the	cat	runs
DT	0.4	0.5	0.1
NN	0.5	0.4	0.1
VB	0.2	0.3	0.5

Viterbi decoding example

- Suppose we want to compute tags for the phrase “the cat runs”
- We need to compute the M and B matrices

$$M[1,\, s] = P(s\,|\,\varnothing) \times P(o_1\,|\,s)$$



	DT	NN	VB
∅	0.5	0.3	0.2
DT	0.1	0.5	0.4
NN	0.2	0.3	0.5
VB	0.4	0.3	0.3

	the	cat	runs
DT	0.4	0.5	0.1
NN	0.5	0.4	0.1
VB	0.2	0.3	0.5

Viterbi decoding example

- Suppose we want to compute tags for the phrase "the cat runs"
- We need to compute the M and B matrices

$$M[1, s] = P(s \mid \varnothing) \times P(o_1 \mid s)$$

DT 0.20

NN 0.15

VB $M[1, \text{VB}] = P(\text{VB} \mid \varnothing) \times P(o_1 \mid \text{VB}) = 0.2 \times 0.2$

the

	DT	NN	VB
∅	0.5	0.3	0.2
DT	0.1	0.5	0.4
NN	0.2	0.3	0.5
VB	0.4	0.3	0.3

	the	cat	runs
DT	0.4	0.5	0.1
NN	0.5	0.4	0.1
VB	0.2	0.3	0.5

Viterbi decoding example

- Suppose we want to compute tags for the phrase “the cat runs”
- We need to compute the M and B matrices

$$M[1, s] = P(s\,|\,\varnothing) \times P(o_1\,|\,s)$$

DT 0.20

NN 0.15

VB 0.04

the

	DT	NN	VB
$\varnothing$	0.5	0.3	0.2
DT	0.1	0.5	0.4
NN	0.2	0.3	0.5
VB	0.4	0.3	0.3

	the	cat	runs
DT	0.4	0.5	0.1
NN	0.5	0.4	0.1
VB	0.2	0.3	0.5

Viterbi decoding example

- Suppose we want to compute tags for the phrase “the cat runs”
- We need to compute the M and B matrices

$$M[1, s] = P(s \,|\, \varnothing) \times P(o_1 \,|\, s)$$

$$M[i, s] = \max_{s_{i-1}} \; P(s \,|\, s_{i-1}) \; P(o_i \,|\, s) M[i-1, s_{i-1}]$$

$$M[2, \text{DT}] = \max_{s_1} \; P(\text{DT} \,|\, s_1) \; P(o_2 \,|\, \text{DT}) M[1, s_1]$$

DT	0.20	DT
NN	0.15	NN
VB	0.04	VB
the		cat

	DT	NN	VB
∅	0.5	0.3	0.2
DT	0.1	0.5	0.4
NN	0.2	0.3	0.5
VB	0.4	0.3	0.3

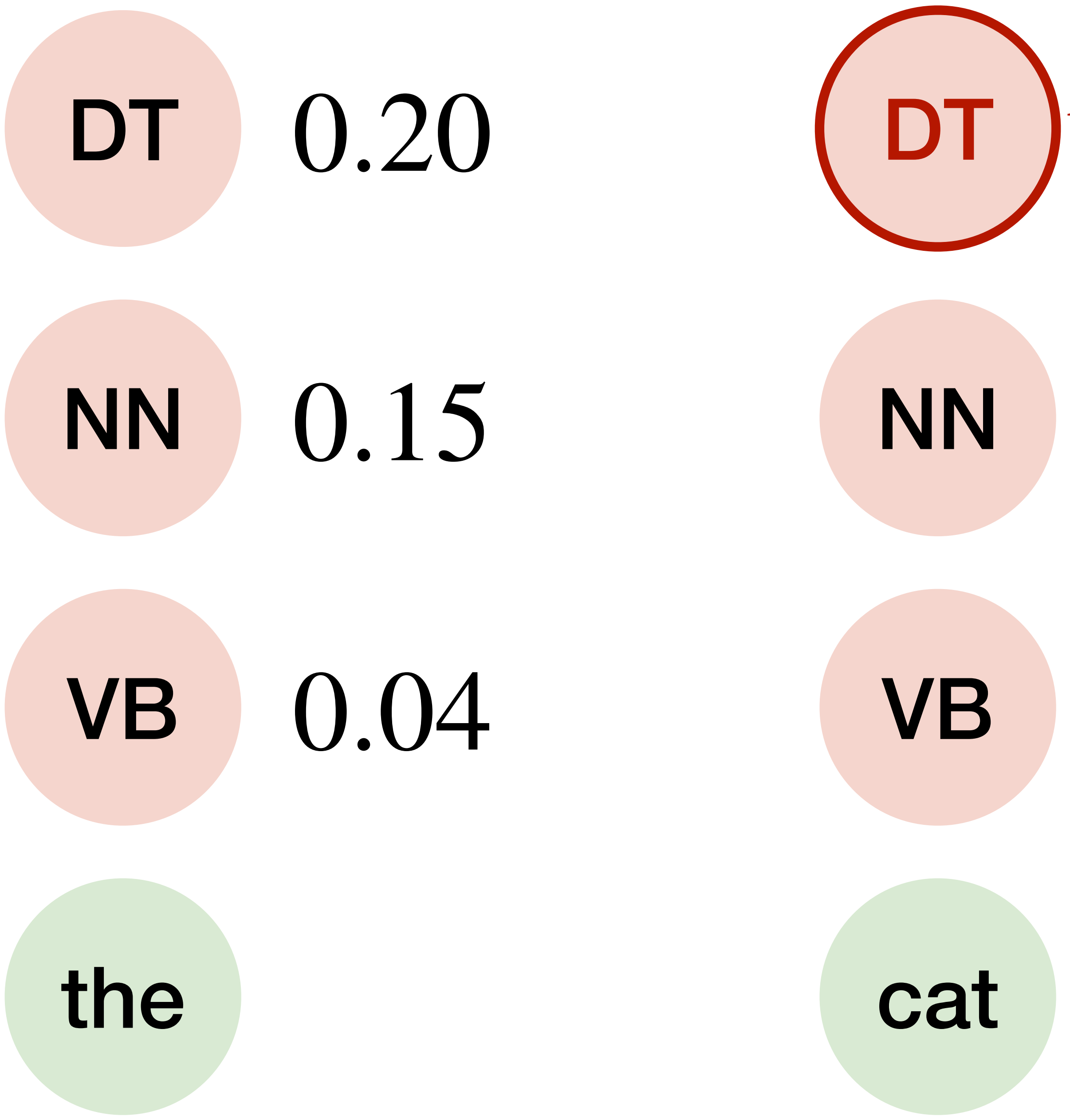
	the	cat	runs
DT	0.4	0.5	0.1
NN	0.5	0.4	0.1
VB	0.2	0.3	0.5

Viterbi decoding example

- Suppose we want to compute tags for the phrase "the cat runs"
- We need to compute the M and B matrices

$$M[1,\, s] = P(s\,|\,\varnothing) \times P(o_1\,|\,s)$$

$$M[i,\, s] = \max_{s_{i-1}}\ P(s\,|\,s_{i-1})\ P(o_i\,|\,s) M[i-1, s_{i-1}]$$



$$M[2,\, \text{DT}] = \max_{s_1}\ P(\text{DT}\,|\,s_1)\ P(o_2\,|\,\text{DT}) M[1,\, s_1]$$

$$= \max(\underbrace{0.1 \times 0.5 \times 0.20,}_{s_1\,=\,\text{DT}}\ 0.2 \times 0.5 \times 0.15,\ 0.4 \times 0.5 \times 0.04)$$

	DT	NN	VB
∅	0.5	0.3	0.2
DT	0.1	0.5	0.4
NN	0.2	0.3	0.5
VB	0.4	0.3	0.3

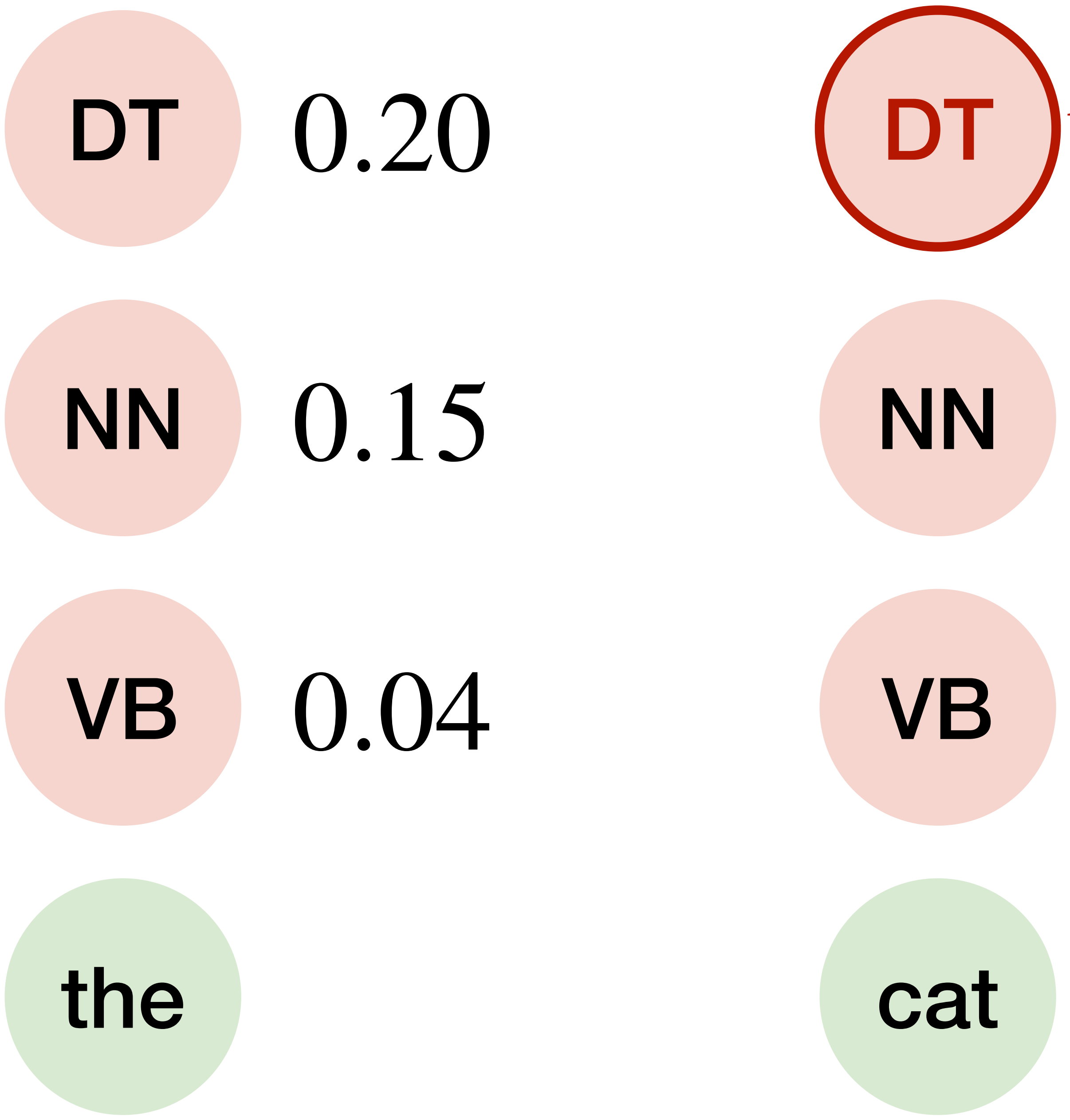
	the	cat	runs
DT	0.4	0.5	0.1
NN	0.5	0.4	0.1
VB	0.2	0.3	0.5

Viterbi decoding example

- Suppose we want to compute tags for the phrase “the cat runs”
- We need to compute the M and B matrices

$$M[1, s] = P(s \mid \varnothing) \times P(o_1 \mid s)$$

$$M[i, s] = \max_{s_{i-1}} \; P(s \mid s_{i-1}) \; P(o_i \mid s) M[i-1, s_{i-1}]$$



$$M[2, \text{DT}] = \max_{s_1} \; P(\text{DT} \mid s_1) \; P(o_2 \mid \text{DT}) M[1, s_1]$$

$$= \max(0.1 \times 0.5 \times 0.20, \boxed{\underset{s_1 = \text{NN}}{0.2 \times 0.5 \times 0.15,}} \; 0.4 \times 0.5 \times 0.04)$$

	DT	NN	VB
∅	0.5	0.3	0.2
DT	0.1	0.5	0.4
NN	0.2	0.3	0.5
VB	0.4	0.3	0.3

	the	cat	runs
DT	0.4	0.5	0.1
NN	0.5	0.4	0.1
VB	0.2	0.3	0.5

Viterbi decoding example

- Suppose we want to compute tags for the phrase "the cat runs"
- We need to compute the M and B matrices

$$M[1,\, s] = P(s \mid \varnothing) \times P(o_1 \mid s)$$

$$M[i,\, s] = \max_{s_{i-1}} \; P(s \mid s_{i-1}) \; P(o_i \mid s) M[i-1, s_{i-1}]$$

DT	0.20	DT	
NN	0.15	NN	
VB	0.04	VB	
the		cat	

$$M[2,\, \text{DT}] = \max_{s_1} \; P(\text{DT} \mid s_1) \; P(o_2 \mid \text{DT}) M[1,\, s_1]$$

$$= \max(0.1 \times 0.5 \times 0.20,\; 0.2 \times 0.5 \times 0.15,\; \boxed{0.4 \times 0.5 \times 0.04})$$

s_1 = VB

	DT	NN	VB
∅	0.5	0.3	0.2
DT	0.1	0.5	0.4
NN	0.2	0.3	0.5
VB	0.4	0.3	0.3

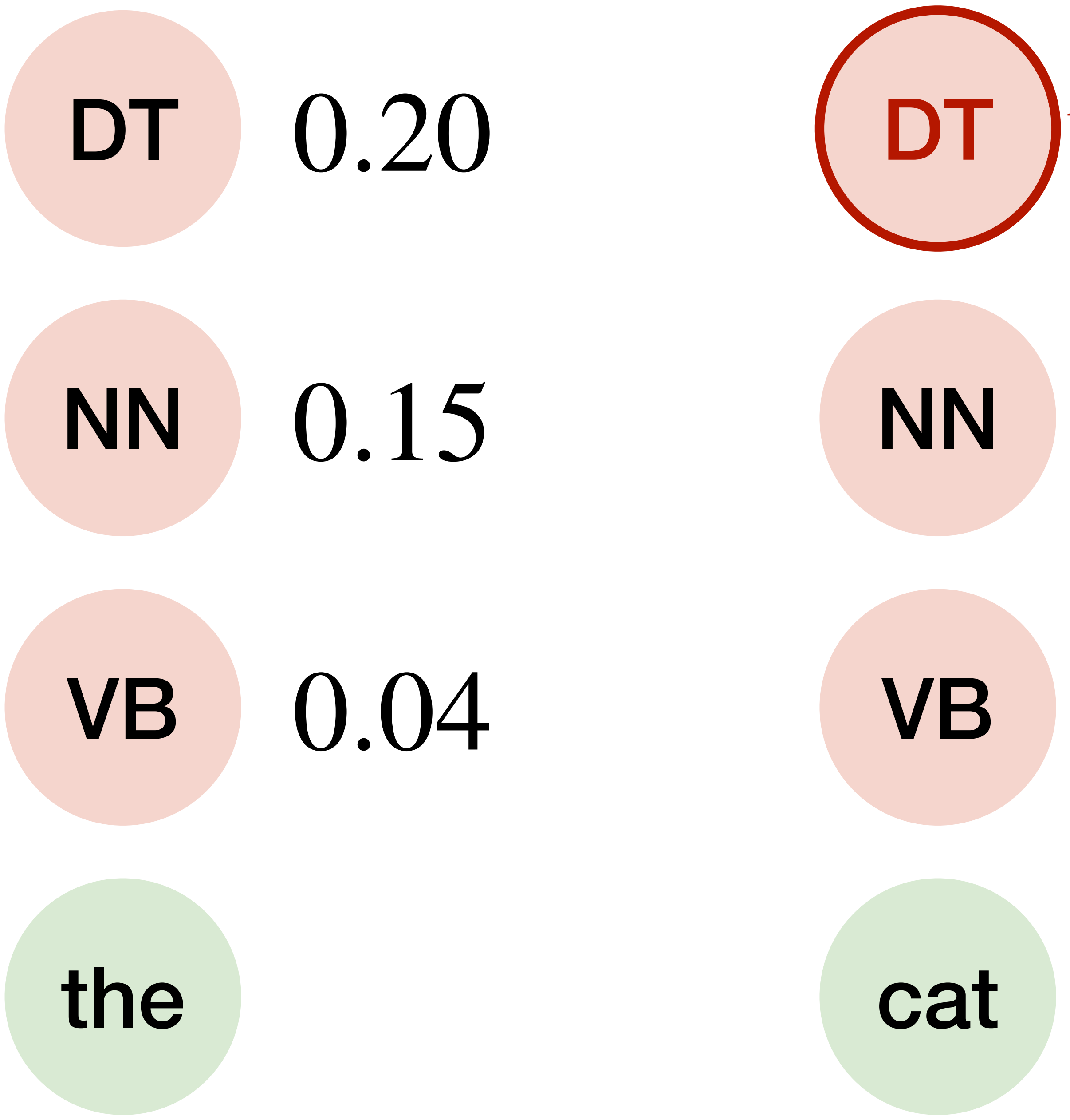
	the	cat	runs
DT	0.4	0.5	0.1
NN	0.5	0.4	0.1
VB	0.2	0.3	0.5

Viterbi decoding example

- Suppose we want to compute tags for the phrase “the cat runs”
- We need to compute the M and B matrices

$$M[1,\, s] = P(s \,|\, \varnothing) \times P(o_1 \,|\, s)$$

$$M[i,\, s] = \max_{s_{i-1}} \; P(s \,|\, s_{i-1}) \; P(o_i \,|\, s) M[i-1, s_{i-1}]$$



$$M[2,\, \text{DT}] = \max_{s_1} \; P(\text{DT} \,|\, s_1) \; P(o_2 \,|\, \text{DT}) M[1,\, s_1]$$
$$= \max(0.1 \times 0.5 \times 0.20,\; 0.2 \times 0.5 \times 0.15,\; 0.4 \times 0.5 \times 0.04)$$
$$= \max(0.01,\; 0.015,\; 0.008)$$

	DT	NN	VB
∅	0.5	0.3	0.2
DT	0.1	0.5	0.4
NN	0.2	0.3	0.5
VB	0.4	0.3	0.3

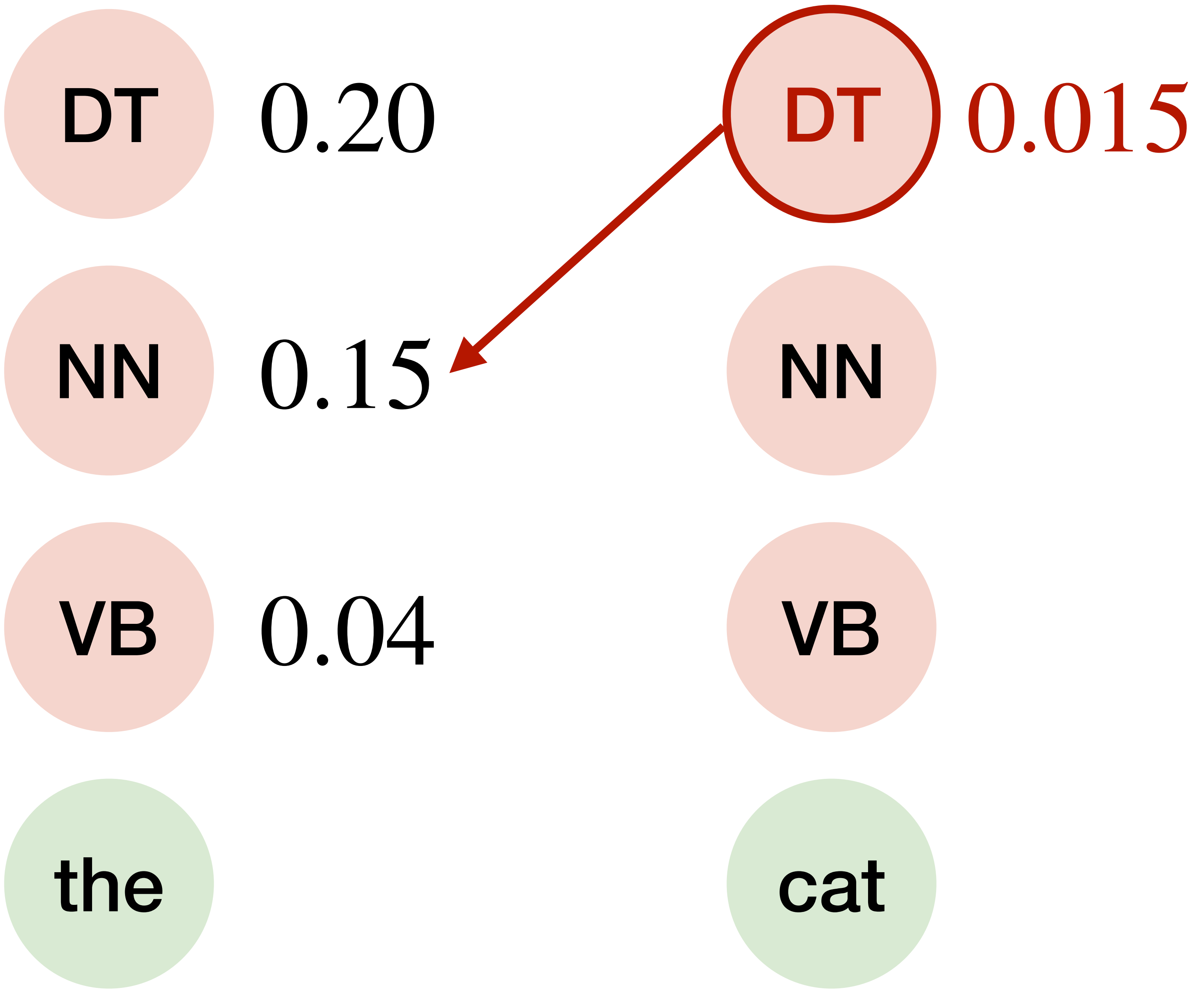
	the	cat	runs
DT	0.4	0.5	0.1
NN	0.5	0.4	0.1
VB	0.2	0.3	0.5

Viterbi decoding example

- Suppose we want to compute tags for the phrase “the cat runs”
- We need to compute the M and B matrices

$$M[1,\, s] = P(s\,|\,\varnothing) \times P(o_1\,|\,s)$$

$$M[i,\, s] = \max_{s_{i-1}}\; P(s\,|\,s_{i-1})\; P(o_i\,|\,s) M[i-1, s_{i-1}]$$



We also want to store the s_1 that gave us the maximum

	DT	NN	VB
$\varnothing$	0.5	0.3	0.2
DT	0.1	0.5	0.4
NN	0.2	0.3	0.5
VB	0.4	0.3	0.3

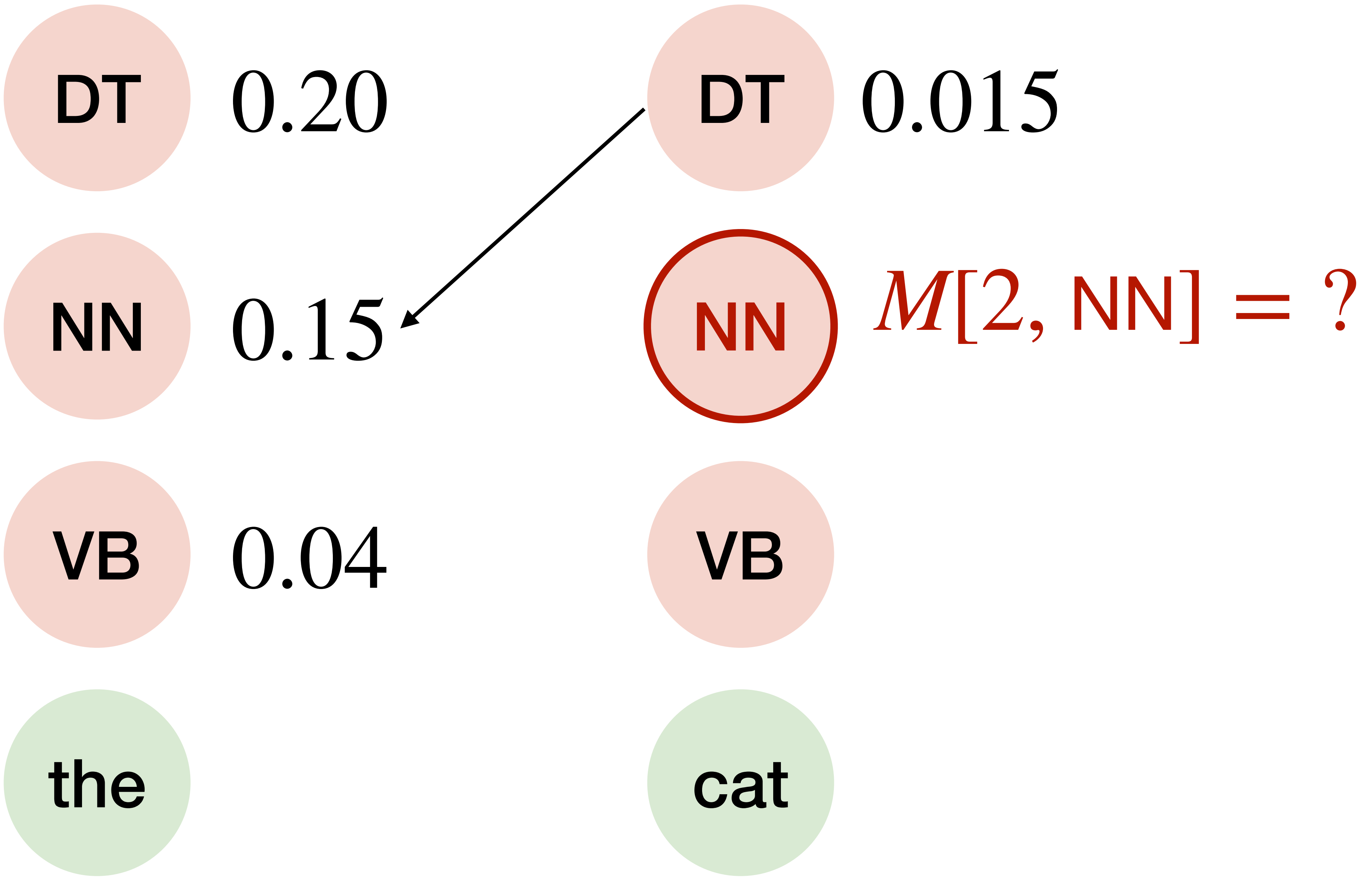
	the	cat	runs
DT	0.4	0.5	0.1
NN	0.5	0.4	0.1
VB	0.2	0.3	0.5

Viterbi decoding example

- Suppose we want to compute tags for the phrase “the cat runs”
- We need to compute the M and B matrices

$$M[1,\, s] = P(s \mid \varnothing) \times P(o_1 \mid s)$$

$$M[i,\, s] = \max_{s_{i-1}} \; P(s \mid s_{i-1})\; P(o_i \mid s) M[i-1, s_{i-1}]$$



	DT	NN	VB
∅	0.5	0.3	0.2
DT	0.1	0.5	0.4
NN	0.2	0.3	0.5
VB	0.4	0.3	0.3

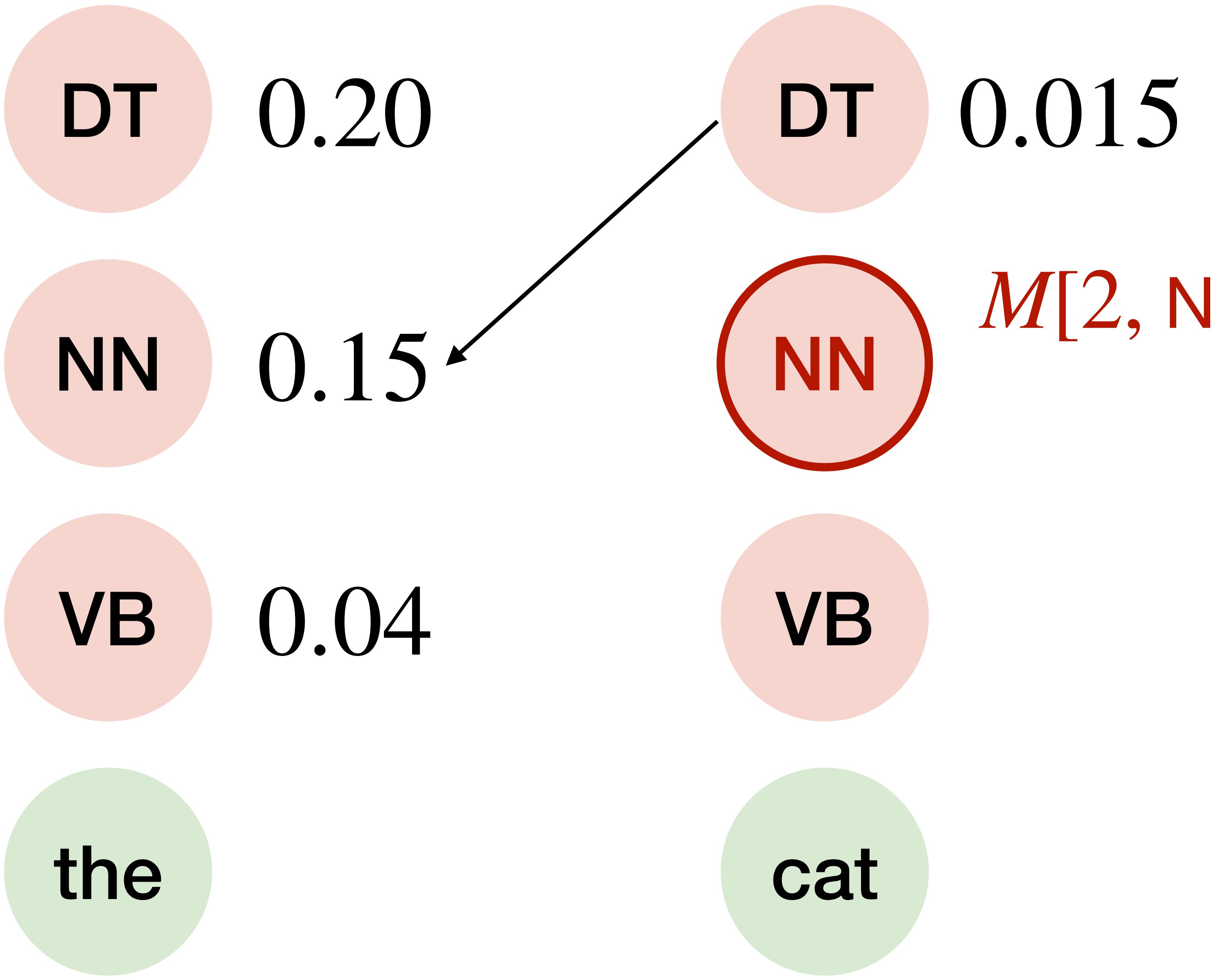
	the	cat	runs
DT	0.4	0.5	0.1
NN	0.5	0.4	0.1
VB	0.2	0.3	0.5

Viterbi decoding example

- Suppose we want to compute tags for the phrase “the cat runs”
- We need to compute the M and B matrices

$$M[1,\, s] = P(s \,|\, \varnothing) \times P(o_1 \,|\, s)$$

$$M[i,\, s] = \max_{s_{i-1}}\; P(s \,|\, s_{i-1})\; P(o_i \,|\, s) M[i-1, s_{i-1}]$$



$$M[2,\, \text{NN}] = \max_{s_1}\; P(\text{NN} \,|\, s_1)\; P(o_2 \,|\, \text{NN}) M[1,\, s_1]$$

$$= \max(0.5 \times 0.4 \times 0.20,\; 0.3 \times 0.4 \times 0.15,\; 0.3 \times 0.4 \times 0.04)$$

$$= \max(0.04,\; 0.018,\; 0.0048)$$

	DT	NN	VB
$\varnothing$	0.5	0.3	0.2
DT	0.1	0.5	0.4
NN	0.2	0.3	0.5
VB	0.4	0.3	0.3

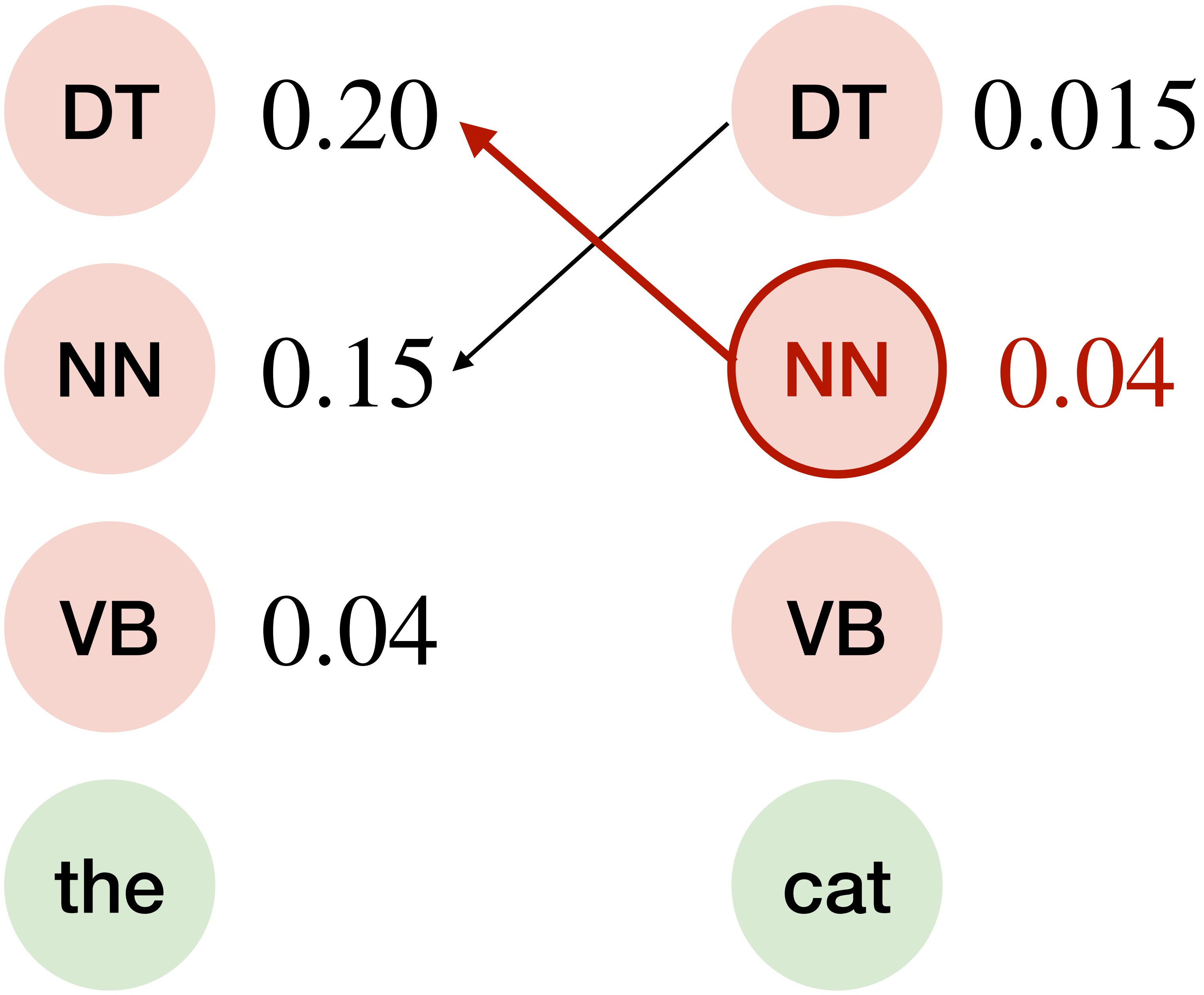
NN	0.5	0.4	0.1
VB	0.2	0.3	0.5

Viterbi decoding example

- Suppose we want to compute tags for the phrase “the cat runs”
- We need to compute the M and B matrices

$$M[1,\, s] = P(s \mid \varnothing) \times P(o_1 \mid s)$$

$$M[i,\, s] = \max_{s_{i-1}} \; P(s \mid s_{i-1})\; P(o_i \mid s) M[i-1, s_{i-1}]$$



	DT	NN	VB
∅	0.5	0.3	0.2
DT	0.1	0.5	0.4
NN	0.2	0.3	0.5
VB	0.4	0.3	0.3

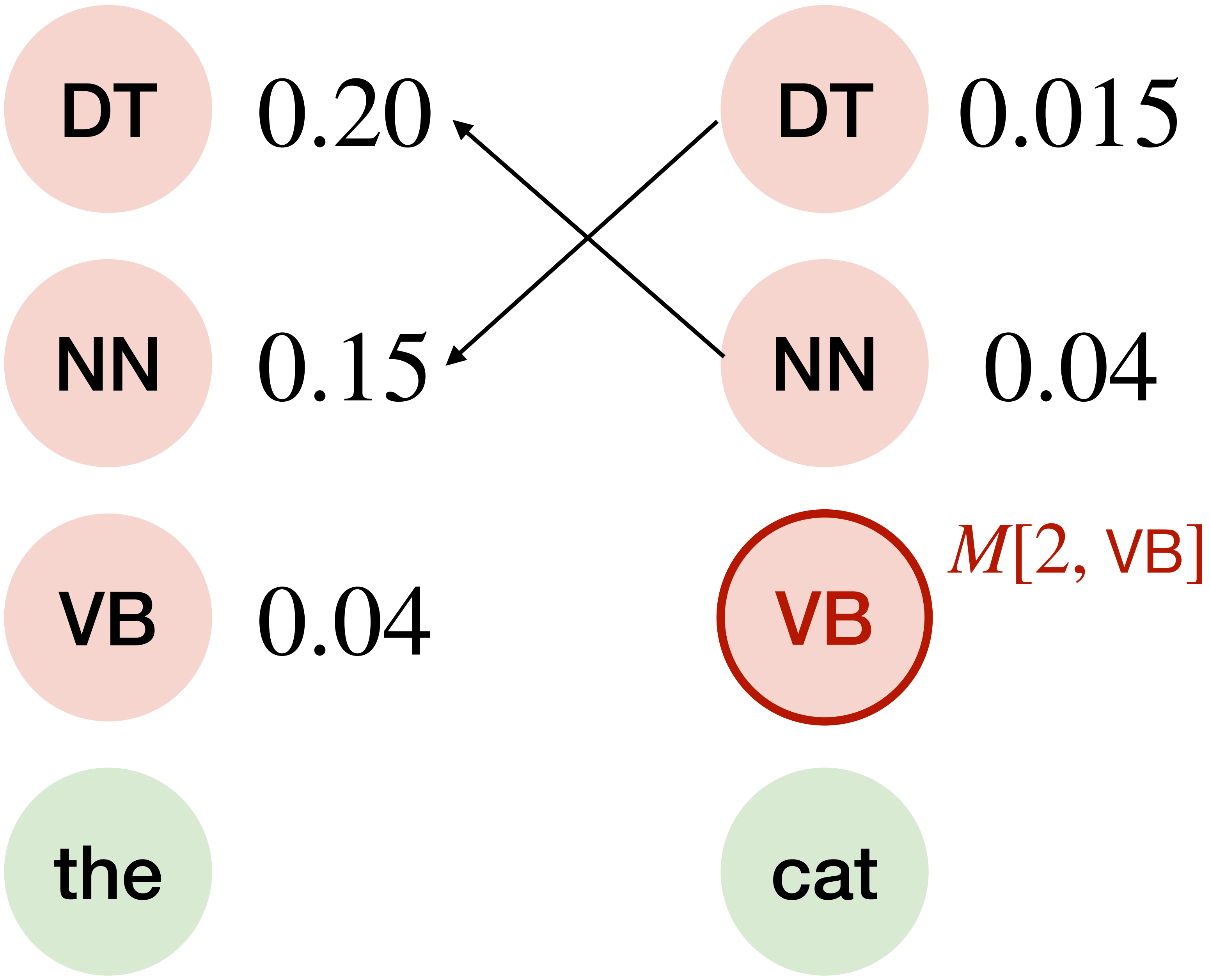
	the	cat	runs
DT	0.4	0.5	0.1
NN	0.5	0.4	0.1
VB	0.2	0.3	0.5

Viterbi decoding example

- Suppose we want to compute tags for the phrase “the cat runs”
- We need to compute the M and B matrices

$$M[1,\, s] = P(s \mid \varnothing) \times P(o_1 \mid s)$$

$$M[i,\, s] = \max_{s_{i-1}}\ P(s \mid s_{i-1})\ P(o_i \mid s) M[i-1, s_{i-1}]$$



$$M[2, \text{VB}] = \max_{s_1}\ P(\text{VB} \mid s_1)\ P(o_2 \mid \text{VB}) M[1,\, s_1]$$
$$= \max(0.4 \times 0.3 \times 0.20,\ 0.5 \times 0.3 \times 0.15,\ 0.3 \times 0.3 \times 0.04)$$
$$= \max(0.024,\ 0.0225,\ 0.0036)$$

	DT	NN	VB
∅	0.5	0.3	0.2
DT	0.1	0.5	0.4
NN	0.2	0.3	0.5
VB	0.4	0.3	0.3

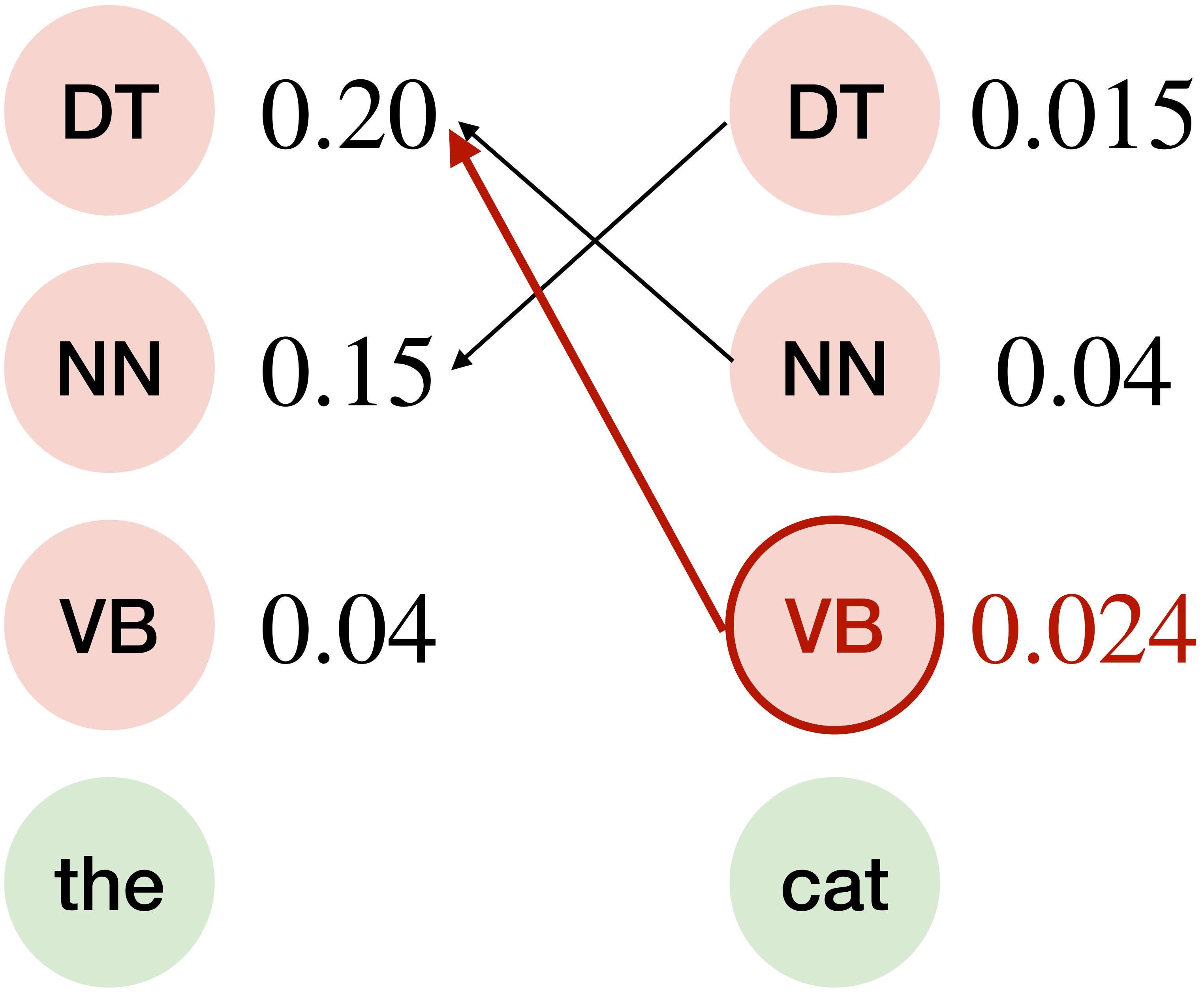
	the	cat	runs
		0.5	0.1
		0.4	0.1
VB	0.2	0.3	0.5

Viterbi decoding example

- Suppose we want to compute tags for the phrase "the cat runs"
- We need to compute the M and B matrices

$$M[1,\, s] = P(s \mid \varnothing) \times P(o_1 \mid s)$$

$$M[i,\, s] = \max_{s_{i-1}} \; P(s \mid s_{i-1})\; P(o_i \mid s) M[i-1, s_{i-1}]$$



	DT	NN	VB
$\varnothing$	0.5	0.3	0.2
DT	0.1	0.5	0.4
NN	0.2	0.3	0.5
VB	0.4	0.3	0.3

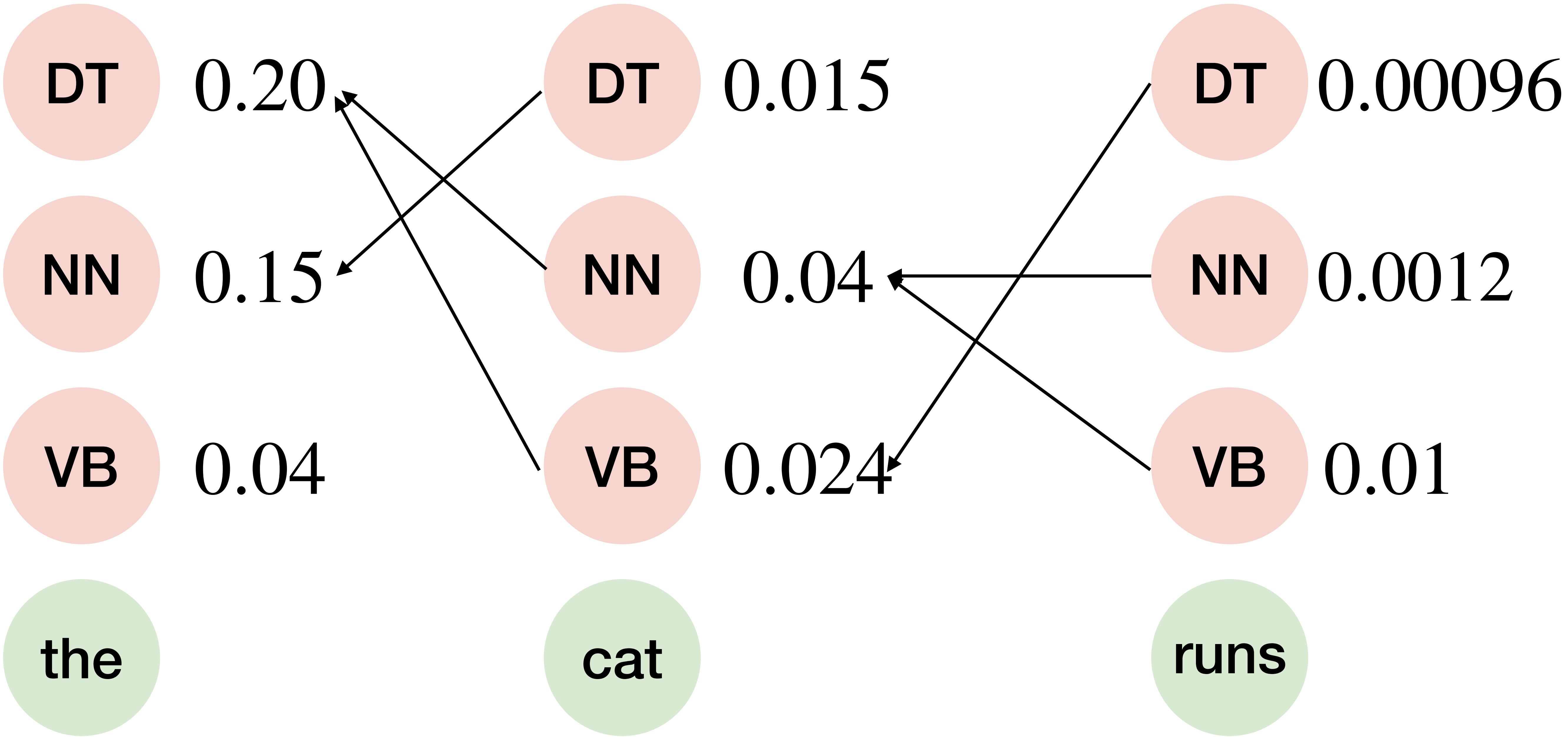
	the	cat	runs
DT	0.4	0.5	0.1
NN	0.5	0.4	0.1
VB	0.2	0.3	0.5

Viterbi decoding example

- Suppose we want to compute tags for the phrase “the cat runs”
- We need to compute the M and B matrices

$$M[1,\, s] = P(s \,|\, \varnothing) \times P(o_1 \,|\, s)$$

$$M[i,\, s] = \max_{s_{i-1}} \; P(s \,|\, s_{i-1})\; P(o_i \,|\, s) M[i-1, s_{i-1}]$$



	DT	NN	VB
∅	0.5	0.3	0.2
DT	0.1	0.5	0.4
NN	0.2	0.3	0.5
VB	0.4	0.3	0.3

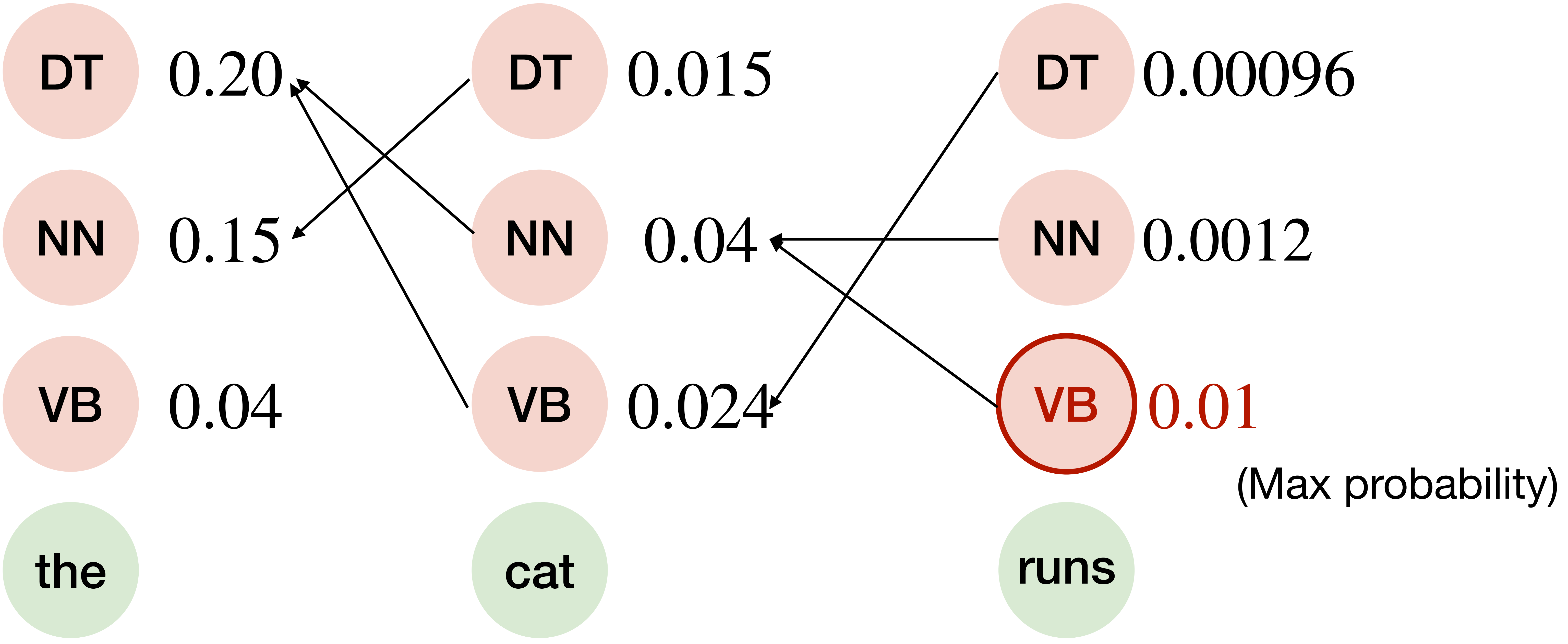
	the	cat	runs
DT	0.4	0.5	0.1
NN	0.5	0.4	0.1
VB	0.2	0.3	0.5

Viterbi decoding example

- Suppose we want to compute tags for the phrase “the cat runs”
- We need to compute the M and B matrices

$$M[1, s] = P(s \mid \varnothing) \times P(o_1 \mid s)$$

$$M[i, s] = \max_{s_{i-1}} \; P(s \mid s_{i-1}) \; P(o_i \mid s) M[i-1, s_{i-1}]$$



	DT	NN	VB
$\varnothing$	0.5	0.3	0.2
DT	0.1	0.5	0.4
NN	0.2	0.3	0.5
VB	0.4	0.3	0.3

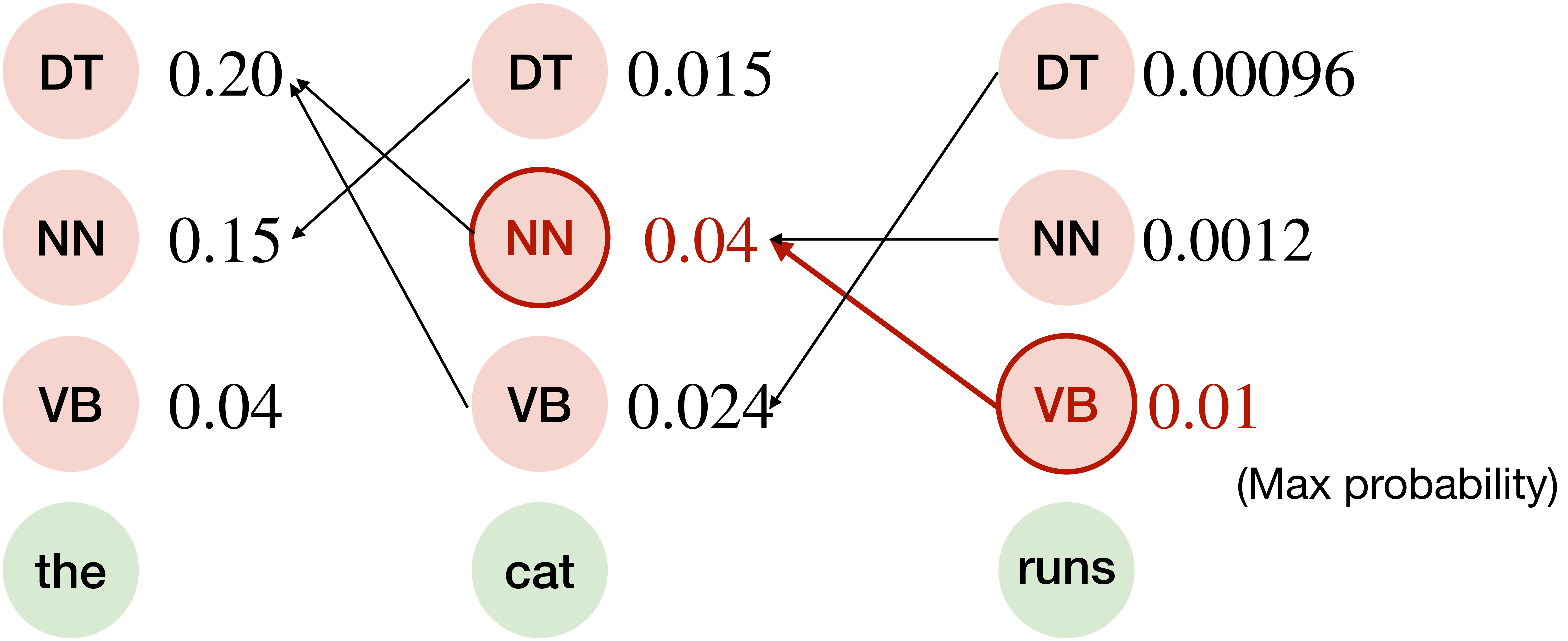
	the	cat	runs
DT	0.4	0.5	0.1
NN	0.5	0.4	0.1
VB	0.2	0.3	0.5

Viterbi decoding example

- Suppose we want to compute tags for the phrase “the cat runs”
- We need to compute the M and B matrices

$$M[1,\, s] = P(s \,|\, \varnothing) \times P(o_1 \,|\, s)$$

$$M[i,\, s] = \max_{s_{i-1}} \; P(s \,|\, s_{i-1}) \; P(o_i \,|\, s) M[i-1, s_{i-1}]$$



	DT	NN	VB
$\varnothing$	0.5	0.3	0.2
DT	0.1	0.5	0.4
NN	0.2	0.3	0.5
VB	0.4	0.3	0.3

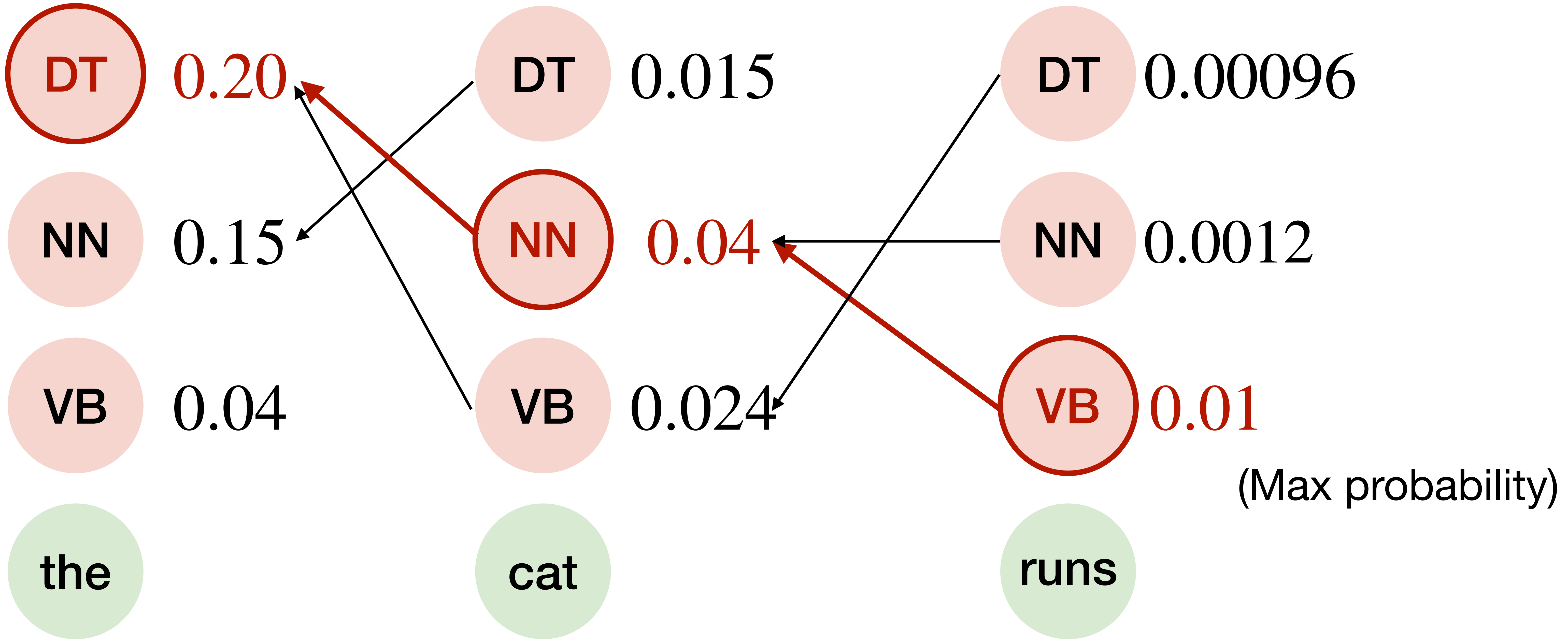
	the	cat	runs
DT	0.4	0.5	0.1
NN	0.5	0.4	0.1
VB	0.2	0.3	0.5

Viterbi decoding example

- Suppose we want to compute tags for the phrase "the cat runs"
- We need to compute the M and B matrices

$$M[1,\ s] = P(s\,|\,\varnothing) \times P(o_1\,|\,s)$$

$$M[i,\ s] = \max_{s_{i-1}}\ P(s\,|\,s_{i-1})\ P(o_i\,|\,s) M[i-1, s_{i-1}]$$

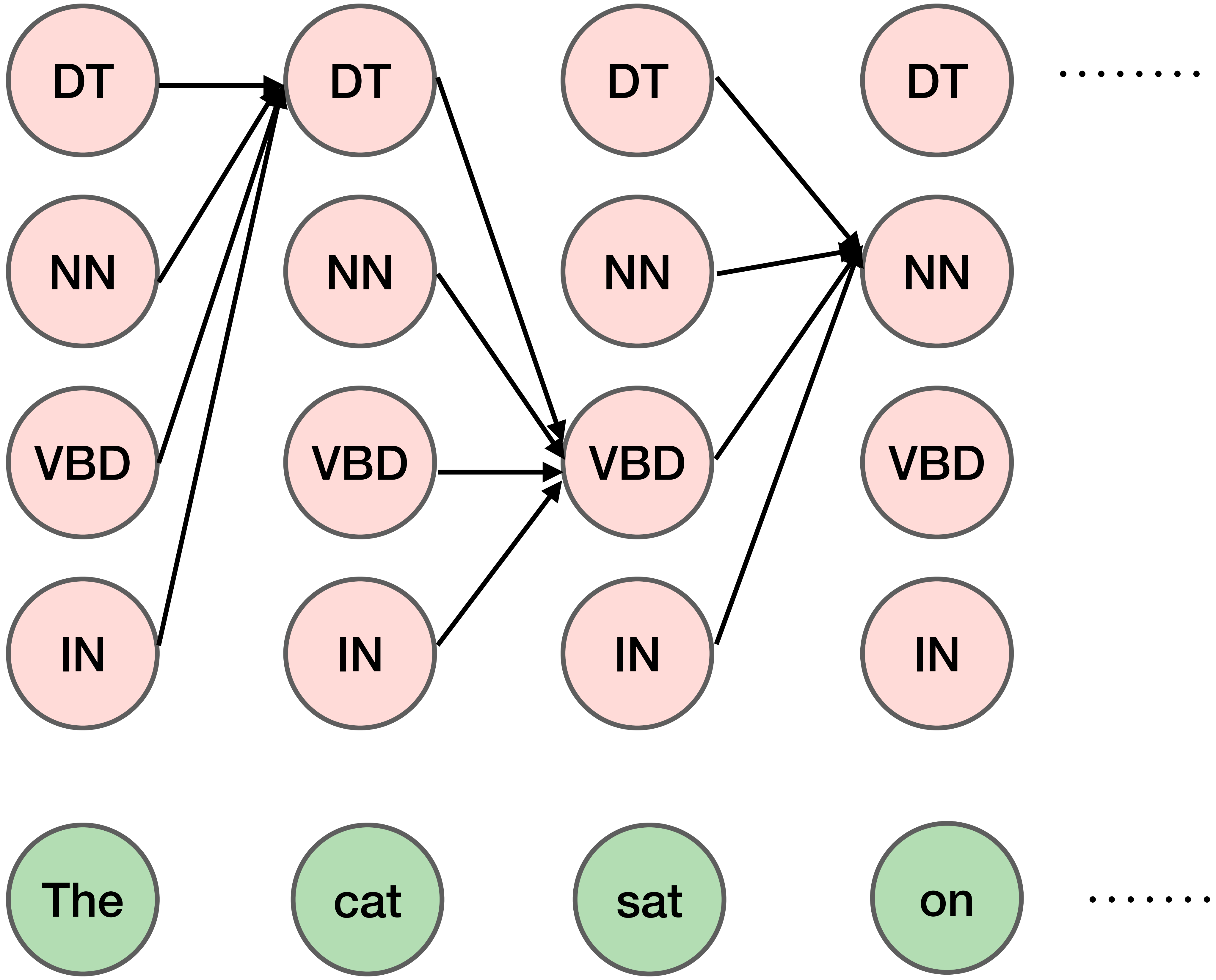


	DT	NN	VB
∅	0.5	0.3	0.2
DT	0.1	0.5	0.4
NN	0.2	0.3	0.5
VB	0.4	0.3	0.3

	the	cat	runs
DT	0.4	0.5	0.1
NN	0.5	0.4	0.1
VB	0.2	0.3	0.5

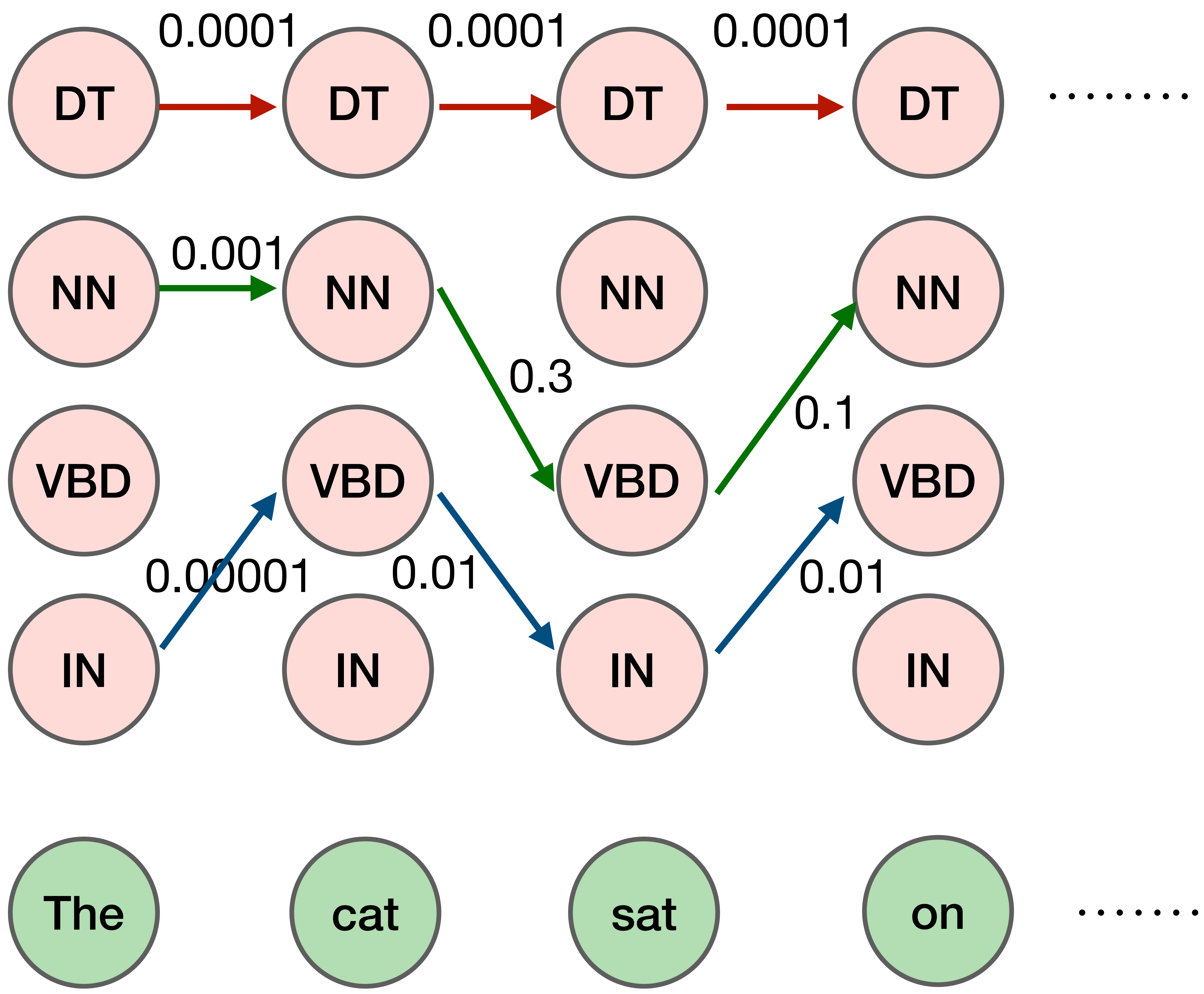
Beam search: Recap

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



Beam search: Recap

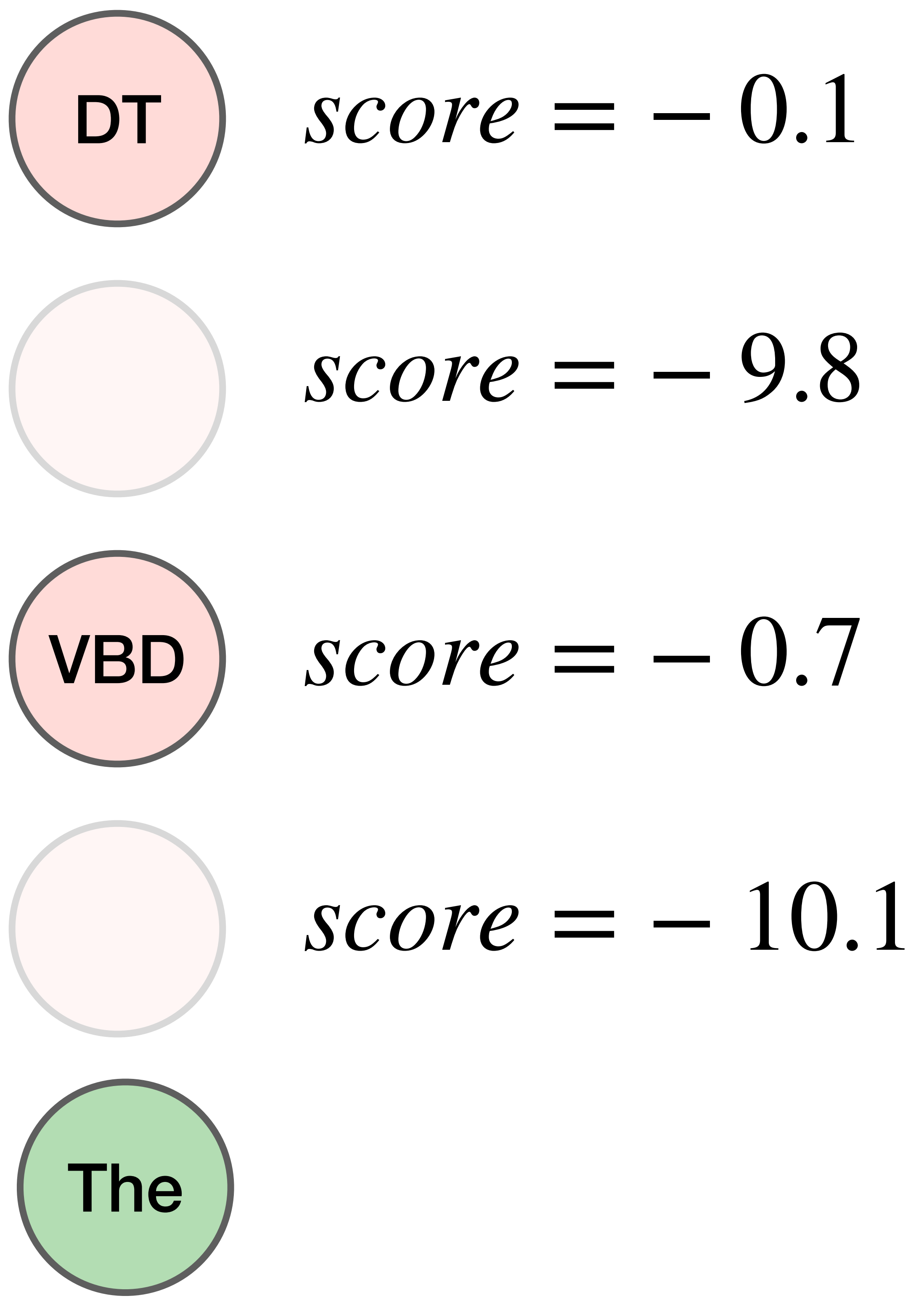
- If K (number of possible hidden states) is too large, Viterbi is too expensive!
- **Observation:** Many paths have very low likelihood!



Beam search: Recap

- If K (number of possible hidden states) is too large, Viterbi is too expensive!
- **Observation:** Many paths have very low likelihood!
- Keep a fixed number of hypotheses at each point
  - Beam width = β

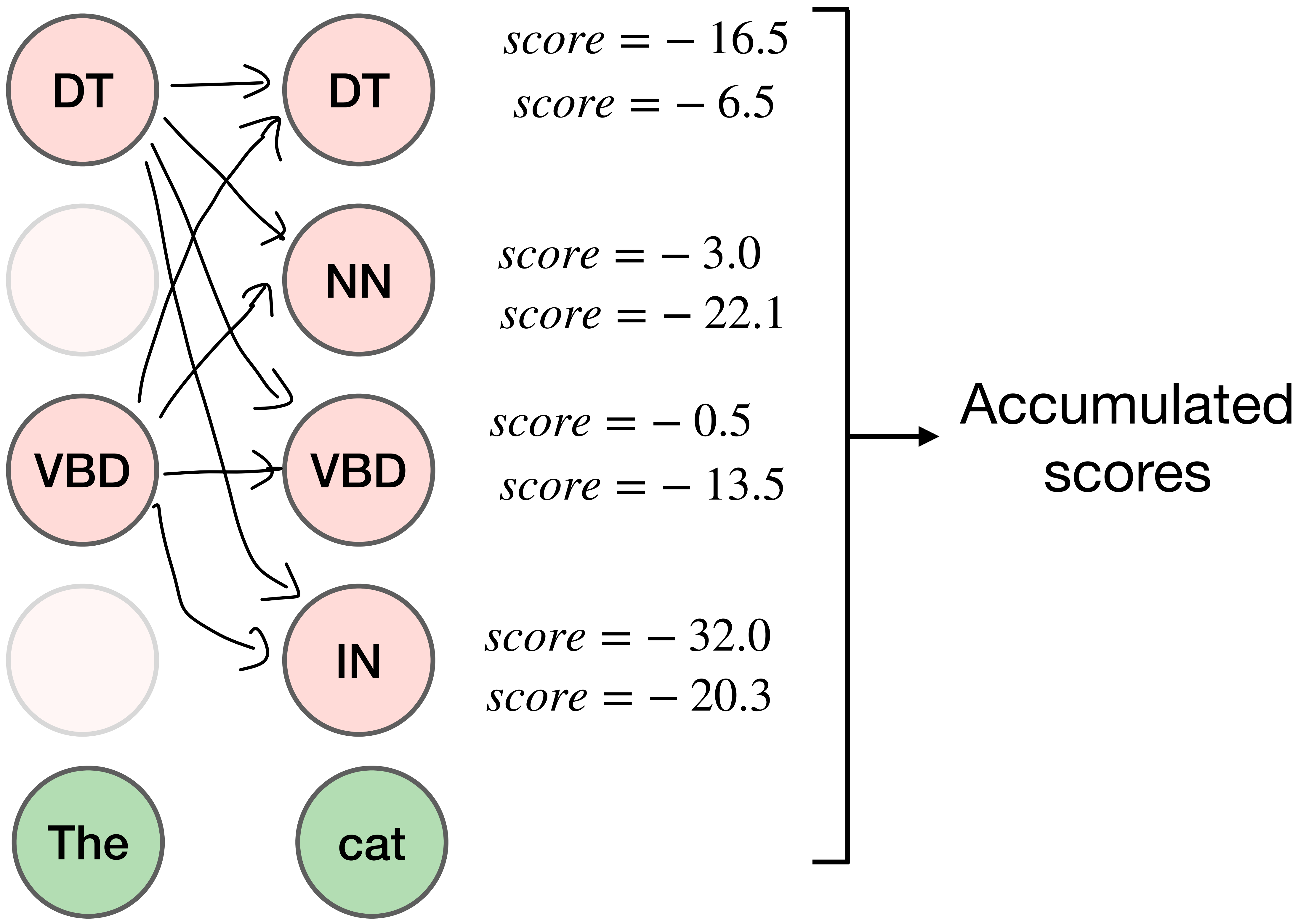
log probabilities



$\beta = 2$

Beam search: Recap

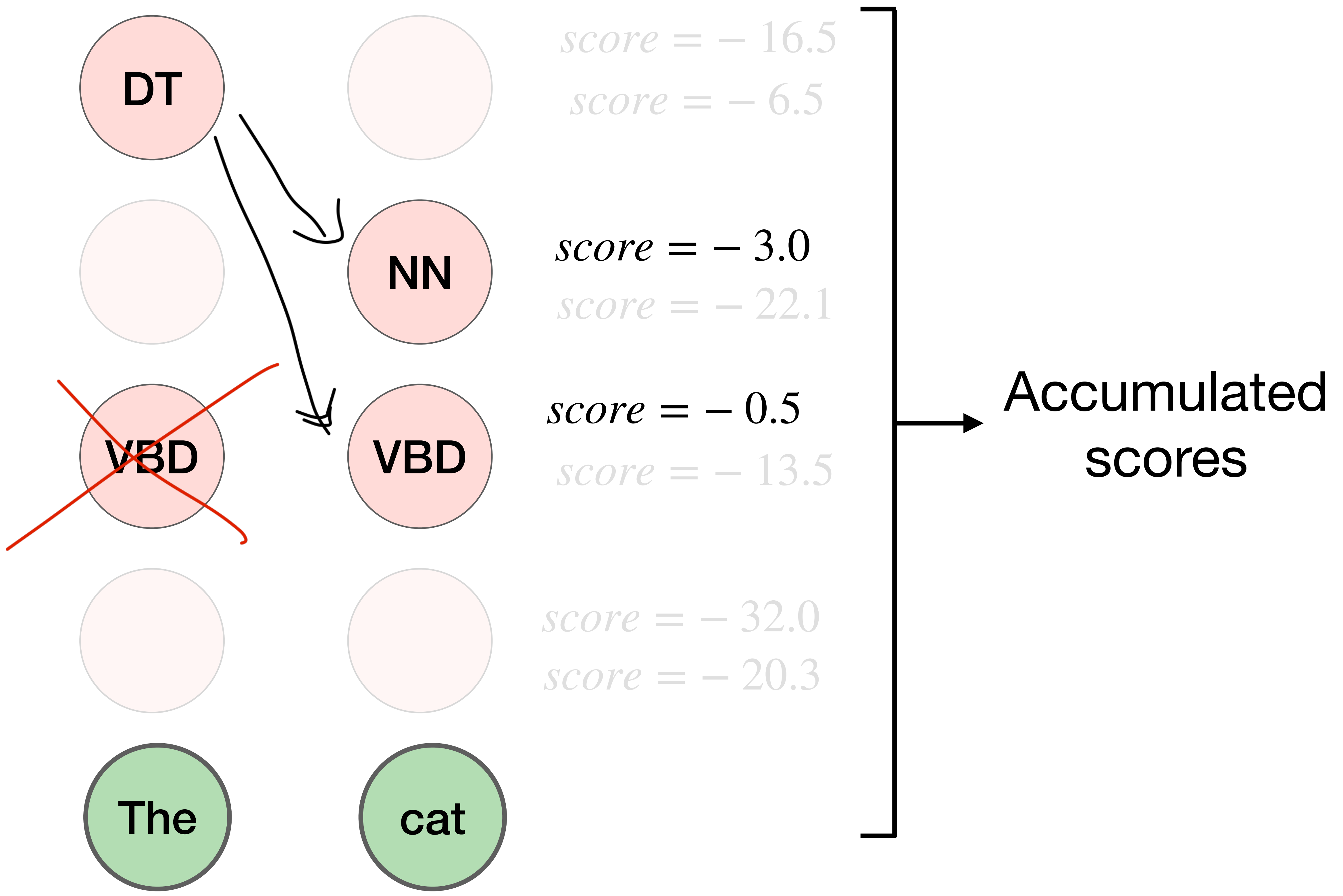
- If K (number of possible hidden states) is too large, Viterbi is too expensive!
- **Observation:** Many paths have very low likelihood!
- Keep a fixed number of hypotheses at each point
    - Beam width = β



Step 1: Expand all partial sequences in current beam

Beam search: Recap

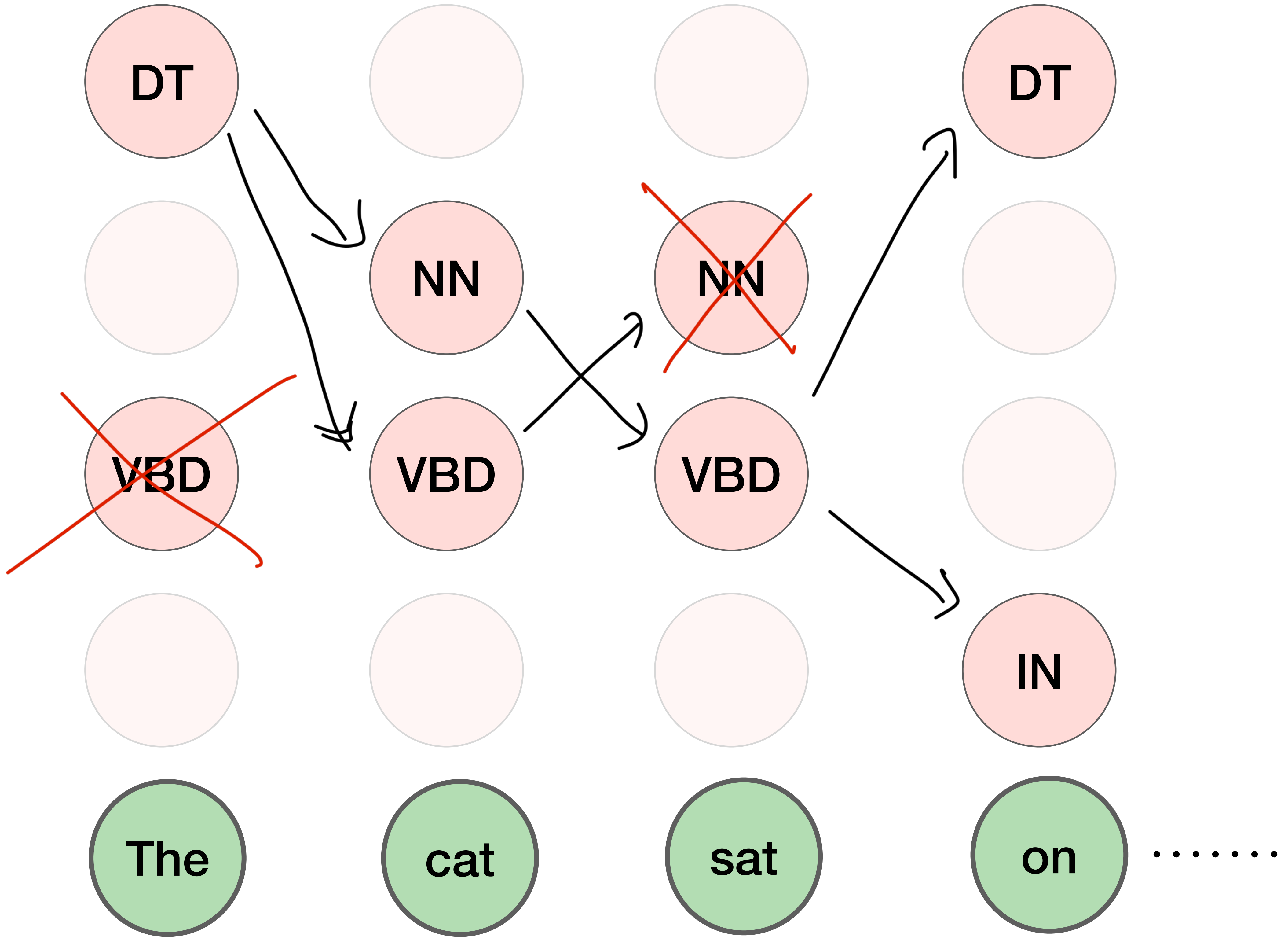
- If K (number of possible hidden states) is too large, Viterbi is too expensive!
- **Observation:** Many paths have very low likelihood!
- Keep a fixed number of hypotheses at each point
  - Beam width = β



Step 2: Prune back to top β scores (sort and select) … repeat!

Beam search: Recap

- If K (number of possible hidden states) is too large, Viterbi is too expensive!
- **Observation:** Many paths have very low likelihood!
- Keep a fixed number of hypotheses at each point
  - Beam width = β



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

Trigram hidden Markov models

- What we have seen so far is also called bigram HMM
- Can be extended to trigram, 4-gram etc: $P(S, O) = \prod_{i=1}^{n} P(s_i \mid s_{i-1}, s_{i-2}) P(o_i \mid s_i)$
- MLE estimate: $P(s_i \mid s_{i-1}, s_{i-2}) = \dfrac{\text{Count}(s_i, s_{i-1}, s_{i-2})}{\text{Count}(s_{i-1}, s_{i-2})}$
  - Can add smoothing techniques to avoid zero probabilities!
- Viterbi:

$$M[i, j, k] = \max_r M[i-1, k, r] \ P(s_j \mid s_k, s_r) \ P(o_i \mid s_j) \quad 1 \leq j, k, r \leq K \ \ 1 \leq i \leq n$$

  - most probable sequence of states ending with state *j* at time *i*, and state *k* at *i-1*
  - Time complexity $= O(nK^3)$

2 min stretch break

Maximum Entropy Markov Models (MEMMs)

ICML 2000

Maximum Entropy Markov Models
for Information Extraction and Segmentation

Andrew McCallum MCCALLUM@JUSTRESEARCH.COM
Dayne Freitag DAYNE@JUSTRESEARCH.COM
Just Research, 4616 Henry Street, Pittsburgh, PA 15213 USA

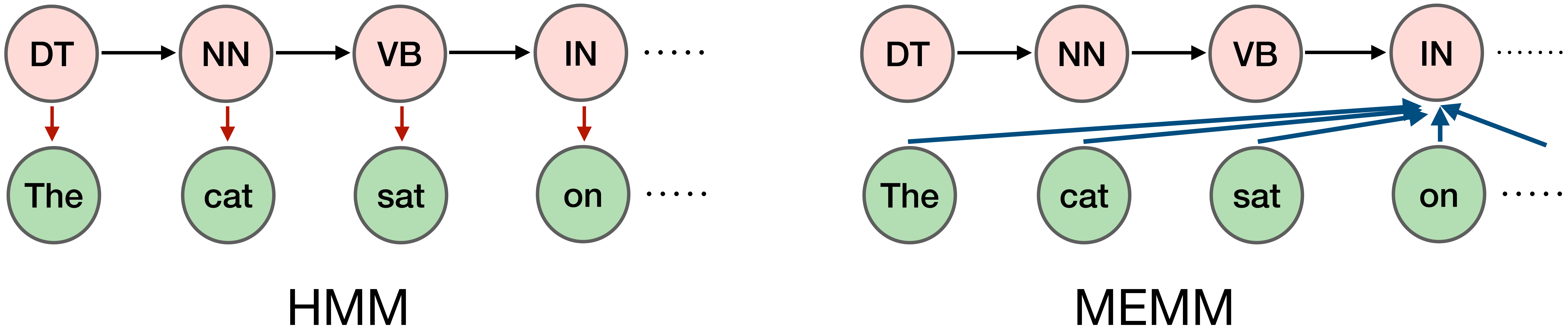
Fernando Pereira PEREIRA@RESEARCH.ATT.COM
AT&T Labs - Research, 180 Park Ave, Florham Park, NJ 07932 USA

Generative vs discriminative models

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

	Generative	**Discriminative**
Text classification	Naive Bayes: $P(c)\,P(d \mid c)$	Logistic Regression: $P(c \mid d)$
Sequence prediction	HMM: $P(s_1, \ldots s_n)\,P(o_1, \ldots o_n \mid s_1, \ldots s_n)$	MEMM: $P(s_1, \ldots s_n \mid o_1, \ldots o_n)$

Maximum entropy Markov model (MEMM)



$$P(S \mid O) = \prod_{i=1}^{n} P(s_i \mid s_{i-1}, s_{i-2}, \ldots, s_1, O)$$

$$= \prod_{i=1}^{n} P(s_i \mid s_{i-1}, O)$$

Markov assumption:
Bigram MEMM

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

weights

features

Important: you can define features over entire word sequence O!

Use features and weights: $P(s_i = s \mid s_{i-1}, O) \propto \exp(\mathbf{w} \cdot \mathbf{f}(s_i = s, s_{i-1}, O, i))$. *Which of the following is the correct way to calculate this probability?*

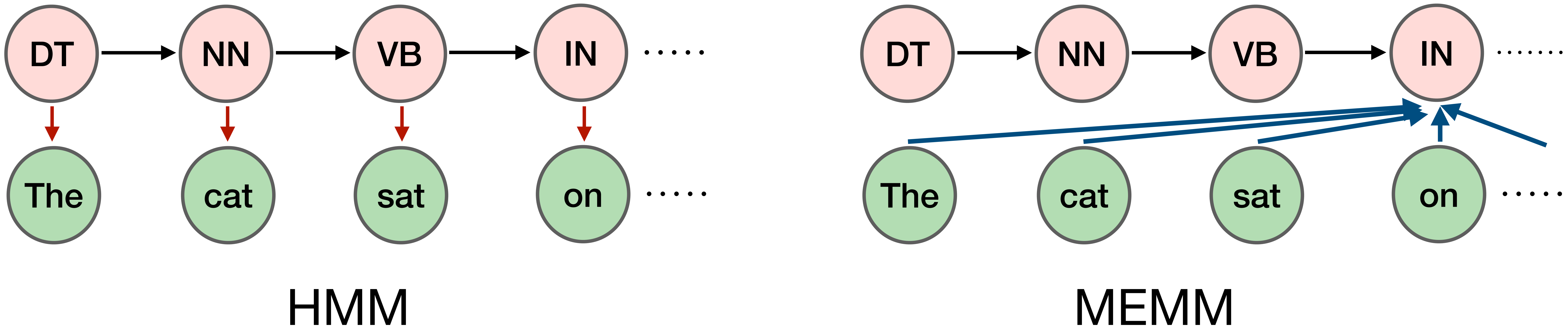
A) $$P(s_i = s \mid s_{i-1}, O) = \frac{\exp(\mathbf{w} \cdot \mathbf{f}(s_i = s, s_{i-1}, O, i))}{\sum_{\boxed{s'=1}}^{\boxed{K}} \exp(\mathbf{w} \cdot \mathbf{f}(s_i = s, \boxed{s_{i-1} = s',} O, i))}$$

B) $$P(s_i = s \mid s_{i-1}, O) = \frac{\exp(\mathbf{w} \cdot \mathbf{f}(s_i = s, s_{i-1}, O, i))}{\sum_{\boxed{s'=1}}^{\boxed{K}} \exp(\mathbf{w} \cdot \mathbf{f}(\boxed{s_i = s',} s_{i-1}, O, i))}$$

C) $$P(s_i = s \mid s_{i-1}, O) = \frac{\exp(\mathbf{w} \cdot \mathbf{f}(s_i = s, s_{i-1}, O, i))}{\sum_{\boxed{O'}} \exp(\mathbf{w} \cdot \mathbf{f}(s_i = s, s_{i-1}, \boxed{O',} i))}$$

The answer is (B)

Maximum entropy Markov model (MEMM)



- Bigram MEMM: $O = \langle o_1, o_2, \ldots, o_n \rangle$

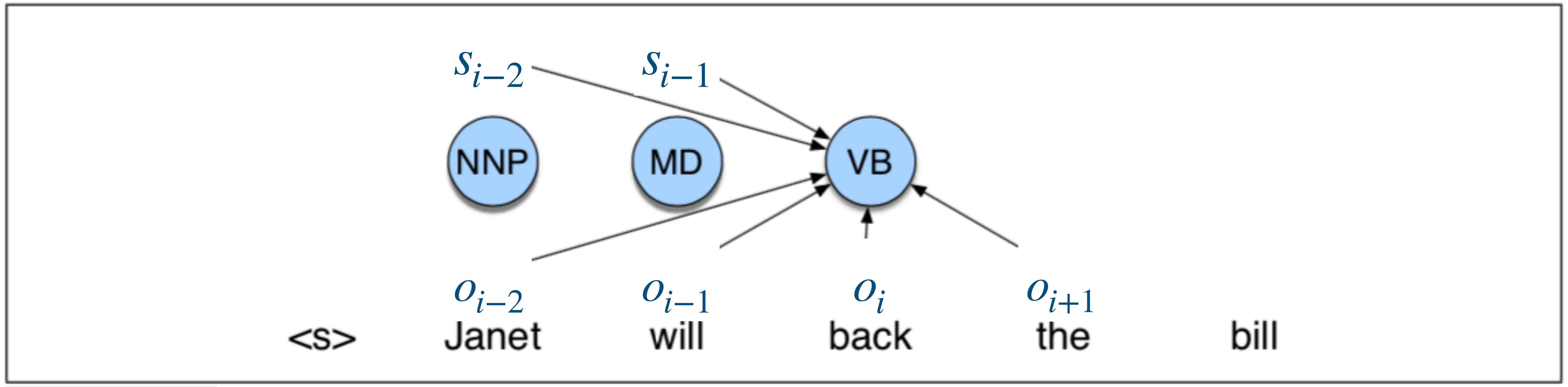
$$\boxed{P(s_i = s \mid s_{i-1}, O) = \frac{\exp(\mathbf{w} \cdot \mathbf{f}(s_i = s, s_{i-1}, O, i))}{\sum_{s'=1}^{K} \exp(\mathbf{w} \cdot \mathbf{f}(s_i = s', s_{i-1}, O, i))}}$$

- Can be easily extended to trigram MEMM, 4-gram MEMM..

$$P(s_i = s \mid s_{i-1}, s_{i-2}, O) = \frac{\exp(\mathbf{w} \cdot \mathbf{f}(s_i = s, s_{i-1}, s_{i-2}, O, i))}{\sum_{s'=1}^{K} \exp(\mathbf{w} \cdot \mathbf{f}(s_i = s', s_{i-1}, s_{i-2}, O, i))}$$

How to define features?

$$\mathbf{f}(s_i = s', s_{i-1}, s_{i-2}, O, i)$$



$\langle s_i, o_{i-2}\rangle$, $\langle s_i, o_{i-1}\rangle$, $\langle s_i, o_i\rangle$, $\langle s_i, o_{i+1}\rangle$, $\langle s_i, o_{i+2}\rangle$
$\langle s_i, s_{i-1}\rangle$, $\langle s_i, s_{i-1}, s_{i-2}\rangle$

Feature templates

s_i = VB and o_{i-2}= Janet
s_i = VB and o_{i-1}= will
s_i = VB and o_i= back
s_i = VB and s_{i-1}= MD
s_i = VB and s_{i-1}= MD and s_{i-2} = NNP

Features (binary)

Features in an MEMM

Incorrect	DT	JJ	NN	DT	NN
Correct	DT	NN	VB	DT	NN
	The	old	man	the	boat
	o_{i-1}	o_i	o_{i+1}	o_{i+2}	o_{i+3}

Which of these feature templates would help most to tag ‘old’ correctly?

(A) $\langle s_i, s_{i-1}, o_i, o_{i-1}, o_{i+1} \rangle$

(B) $\langle s_i, s_{i-1}, o_i, o_{i-1} \rangle$

(C) $\langle s_i, o_i, o_{i-1}, o_{i+1} \rangle$

(D) $\langle s_i, o_i, o_{i-1}, o_{i+1}, o_{i+2} \rangle$

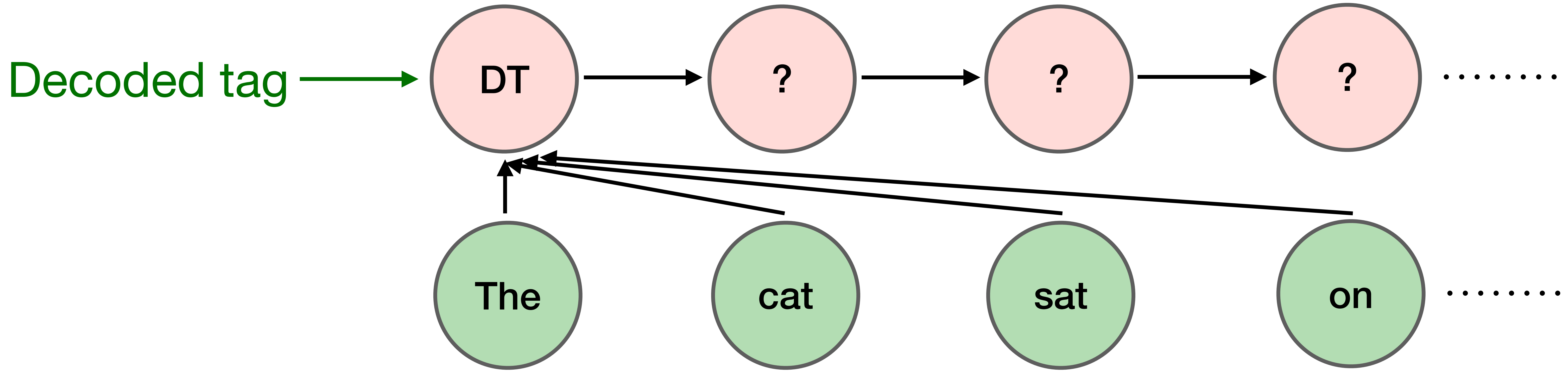
The answer is (D)

MEMMs: Decoding

- Bigram MEMM:

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

- Greedy decoding:



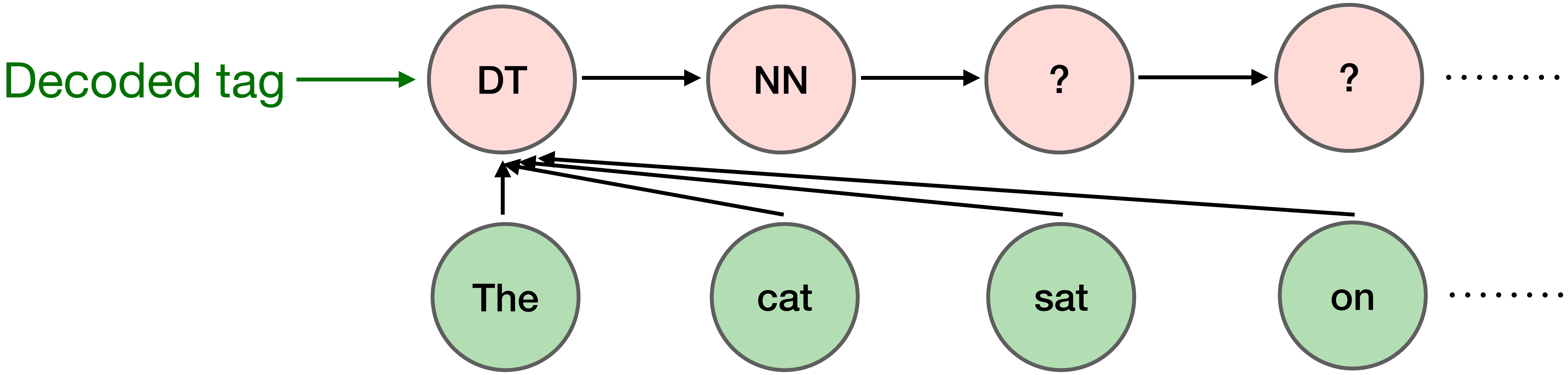
$$\hat{s}_1 = \arg\max_{s} P(s_i = s \mid \varnothing, O) = \boxed{\arg\max_{s} \mathbf{w} \cdot \mathbf{f}(s_i = s, s_{i-1} = \varnothing, O)} = \text{DT}$$

MEMMs: Decoding

- Bigram MEMM:

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

- Greedy decoding:



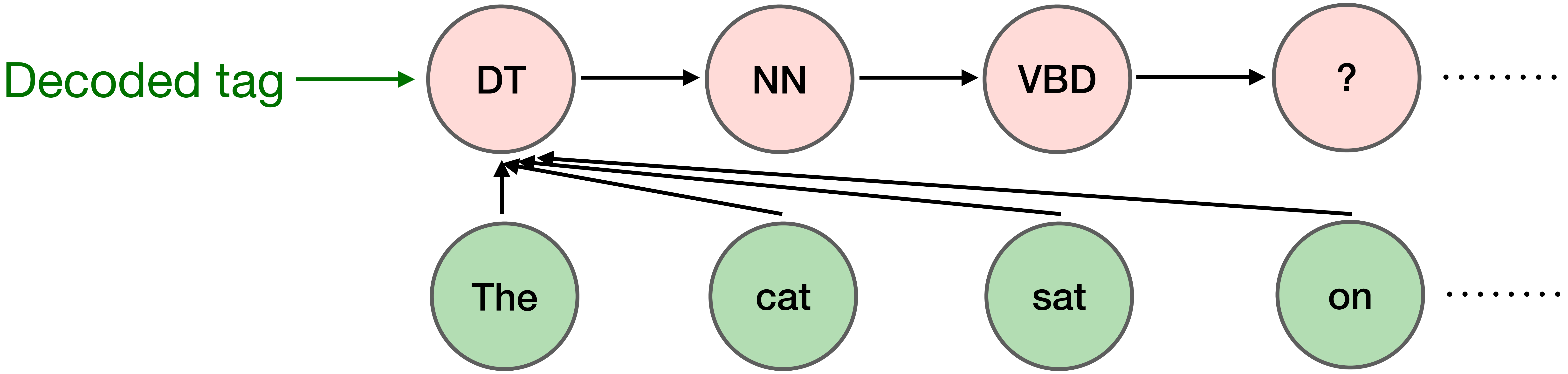
$$\hat{s}_2 = \arg\max_{s} P(s_i = s \mid \text{DT}, O) = \text{NN}$$

MEMMs: Decoding

- Bigram MEMM:

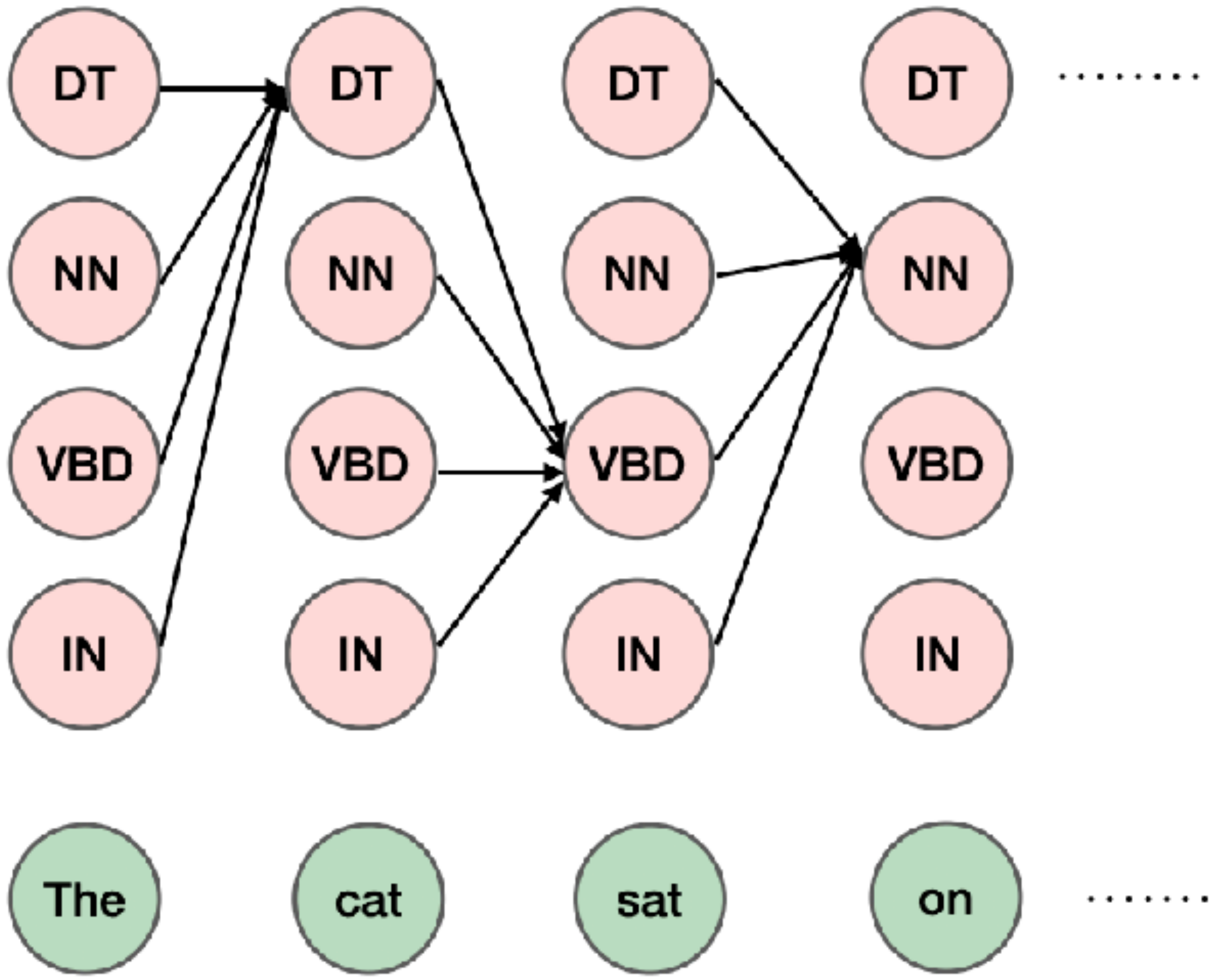
$$\hat{S} = \arg\max_{S} P(S \mid O) = \arg\max_{S} \Pi_i P(s_i \mid s_{i-1}, O)$$

- Greedy decoding:



$$\hat{s}_i = \arg\max_{s} P(s_i = s \mid \hat{s}_{i-1}, O)$$

Viterbi decoding for MEMMs



$M[i, j]$ stores joint probability of most probable sequence of states ending with state j at time i

$$M[i,j] = \max_k M[i-1,k] \boxed{P(s_i = j \mid s_{i-1} = k, O)} \quad 1 \leq k \leq K \quad 1 \leq i \leq n$$

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

MEMM: Decoding

How would you compare the computational complexity of Viterbi decoding for bigram MEMMs compared to decoding for bigram HMMs?

(A) More operations in MEMM
(B) More operations in HMM
(C) Equal
(D) Depends on number of features in MEMM

The answer is (D)

MEMM: $$M[i,j] = \max_k M[i-1,k] \boxed{P(s_i = j \mid s_{i-1} = k, O)} \quad 1 \leq k \leq K \quad 1 \leq i \leq n$$

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

MEMM: Learning

$$P(s_i = s \,|\, s_{i-1}, O) = \frac{\exp(\mathbf{w} \cdot \mathbf{f}(s_i = s, s_{i-1}, O, i))}{\sum_{s'} \exp(\mathbf{w} \cdot \mathbf{f}(s_i = s', s_{i-1}, O, i))}$$

- **Given:** annotated pairs of (S, O) where each $S = \langle s_1, s_2, \ldots, s_n \rangle$
- **Gradient descent:** similar to logistic regression!
- Compute gradients with respect to weights $\mathbf{w}$ and update:
  - Loss for one sequence, $L = -\sum_{i=1}^{n} \log P(s_i \,|\, s_{i-1}, O)$

MEMM vs HMM

- HMM models the joint $P(S, O)$ while MEMM models the required prediction $P(S \mid O)$
- MEMM has more expressivity
  - accounts for dependencies between neighboring states and **entire observation** sequence
  - allows for **more flexible features**
- HMM may hold an advantage if the dataset is small

Conditional Random Fields (CRFs)

ICML 2001

Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data

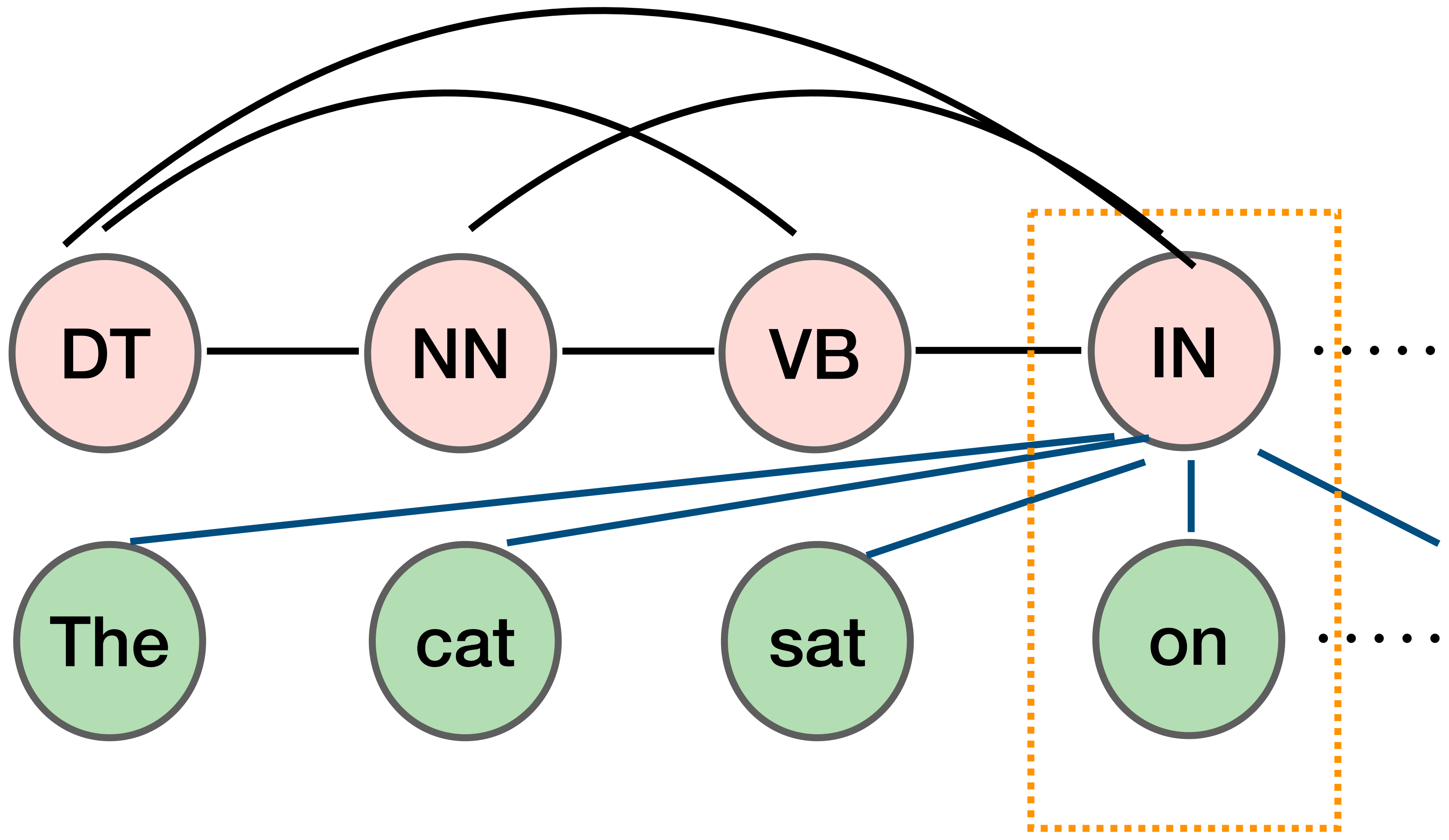
---

John Lafferty[†,*] LAFFERTY@CS.CMU.EDU
Andrew McCallum[*,†] MCCALLUM@WHIZBANG.COM
Fernando Pereira[*,‡] FPEREIRA@WHIZBANG.COM

[*]WhizBang! Labs–Research, 4616 Henry Street, Pittsburgh, PA 15213 USA
[†]School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 USA
[‡]Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104 USA

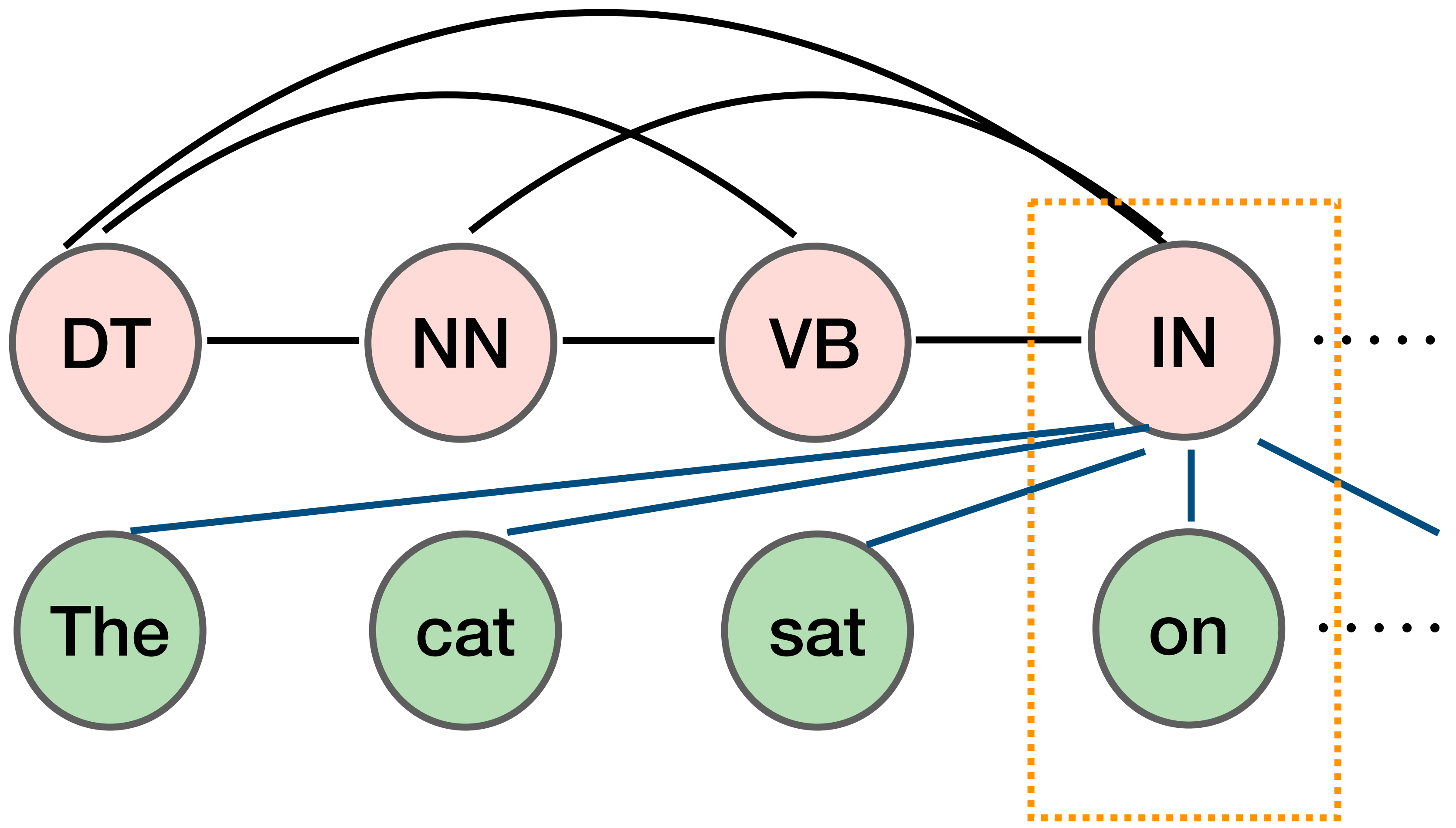
Conditional Random Field

- Model $P(s_1, \ldots, s_n \mid o_1, \ldots, o_n)$ directly
- No Markov assumption
  - Map entire sequence of states S and observations O to a **global** feature vector
    - Normalize over entire sequences



$$P(S \mid O) = \frac{\exp(\mathbf{w} \cdot \mathbf{f}(S, O))}{\sum_{S'} \exp(\mathbf{w} \cdot \mathbf{f}(S', O))} = \frac{\exp(\mathbf{w} \cdot \mathbf{f}(S, O))}{Z(O)}$$

Features



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

$$P(S\,|\,O) = \frac{\exp(\sum_{k=1}^{m} w_k \cdot F_k(S, O))}{\sum_{S'} \exp(\sum_{k=1}^{m} w_k \cdot F_k(S', O))}$$

- Each F_k in $\mathbf{f}$ is a **global** feature function
- Can be computed as a combination of local features: $F_k = \sum_{i=1}^{n} f_k(s_{i-1}, s_i, O, i)$
- Each local feature only depends on previous and current states

CRF: Decoding

- $$\hat{S} = \arg\max_S P(S \mid O) = \arg\max_S \frac{\exp(\mathbf{w} \cdot \mathbf{f}(S, O))}{Z(O)}$$

$$= \arg\max_S \exp(\mathbf{w} \cdot \mathbf{f}(S, O))$$

$$= \arg\max_S \sum_{k=1}^{m} \sum_{i=1}^{n} w_k f_k(s_{i-1}, s_i, O, i)$$

- Use Viterbi similar to HMM and MEMM

CRF: Learning

Log-Linear Models, MEMMs, and CRFs

Michael Collins

$$P(S\,|\,O) = \frac{\exp(\sum_{k=1}^{m}\sum_{i=1}^{n} w_k f_k(s_{i-1}, s_i, O, i))}{Z(O)}$$

$$= \frac{\exp(\sum_{k=1}^{m}\sum_{i=1}^{n} w_k f_k(s_{i-1}, s_i, O, i))}{\sum_{s'_1,\ldots,s'_n}\exp(\sum_{k=1}^{m}\sum_{i=1}^{n} w_k f_k(s'_{i-1}, s'_i, O, i))}$$

$$-\log P(S \mid O) = -\sum_{k=1}^{m}\sum_{i=1}^{n} w_k f_k(s_{i-1}, s_i, O, i)) + \log \sum_{s'_1,\ldots,s'_n} \exp(\sum_{k=1}^{m}\sum_{i=1}^{n} w_k f_k(s'_{i-1}, s'_i, O, i))$$

$\dfrac{-\partial \log P(S \mid O)}{\partial w_k}$ can be done efficiently using dynamic programming

CRF vs MEMM

- MEMM models the required prediction $P(S \mid O)$ using the Markov assumption, while the CRF does not
- CRF uses global features while MEMM features are localized
- Feature design is flexible in both models
- CRF is computationally more complex

CRFs in deep learning era

- Use CRFs on top of neural representations (instead of features and weights)
- Joint sequence prediction without the need for defining features!
- Recent architectures such as seq2seq w/ attention or Transformer may implicitly do the job

Conditional Random Fields as Recurrent Neural Networks

Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, Philip H. S. Torr; Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1529-1537

Neural Architectures for Named Entity Recognition

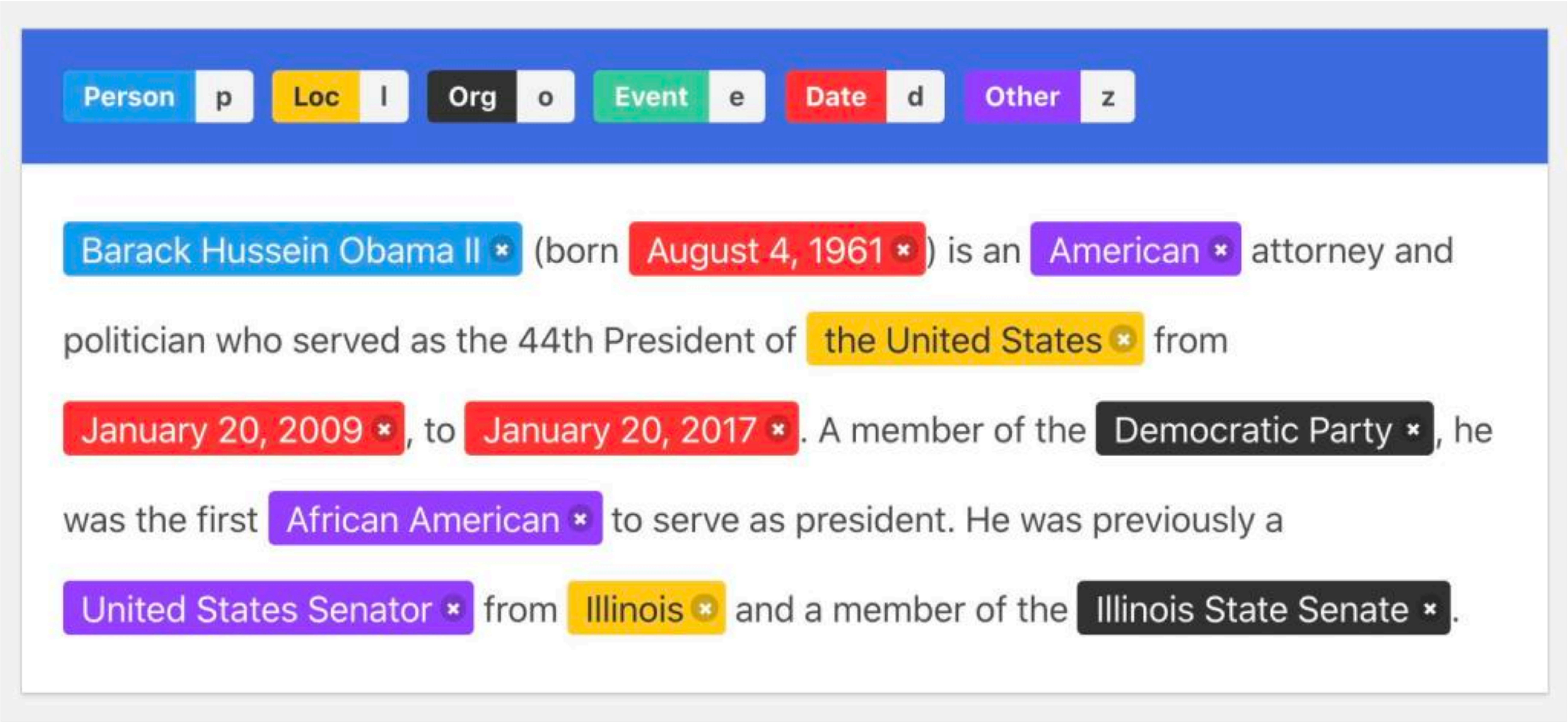
Guillaume Lample[♠] **Miguel Ballesteros**[♣♠]
Sandeep Subramanian[♠] **Kazuya Kawakami**[♠] **Chris Dyer**[♠]
[♠]Carnegie Mellon University [♣]NLP Group, Pompeu Fabra University
{glample,sandeeps,kkawakam,cdyer}@cs.cmu.edu,
miguel.ballesteros@upf.edu

Bidirectional LSTM-CRF Models for Sequence Tagging

Zhiheng Huang Baidu research huangzhiheng@baidu.com
Wei Xu Baidu research xuwei06@baidu.com
Kai Yu Baidu research yukai@baidu.com

Named entity recognition (NER)

Named entity recognition



Named entities

- Named entity, in its core usage, means anything that can be referred to with a proper name.
- NER is the task of 1) finding spans of text that constitute proper names; 2) tagging the type of the entity
- Most common 4 tags:
  - **PER** (Person): “Marie Curie”
  - **LOC** (Location): “New York City”
  - **ORG** (Organization): “Princeton University”
  - **MISC** (Miscellaneous): nationality, events, ..

Steve	Jobs	founded	Apple	with	Steve	Wozniak	.	
PER	PER	O	ORG	O	PER	PER	.	

Only	France	and	Britain	backed	Fischler	’s	proposal	.
O	LOC	O	LOC	O	PER	O	O	O

O = not an entity

If multiple words constitute a named entity, they will be labeled with the same tag.

NER: BIO Tagging

[PER Jane Villanueva] of [ORG United] , a unit of [ORG United Airlines Holding] , said the fare applies to the [LOC Chicago] route.

Words	BIO Label
Jane	B-PER
Villanueva	I-PER
of	O
United	B-ORG
Airlines	I-ORG
Holding	I-ORG
discussed	O
the	O
Chicago	B-LOC
route	O
.	O

B: token that begins a span

I: tokens that inside a span

O: tokens outside of a span