

# Lecture 5: Sequence Models II

Alan Ritter

(many slides from Greg Durrett, Dan Klein, Vivek Srikumar, Chris Manning, Yoav Artzi)

# Guest Lecture

- ▶ TAs will meet you at 11:45am in the Coda lobby and leave promptly at 11:50 for the Atrium
- ▶ Also available online
- ▶ See Piazza for an announcement



▶ **Jacob Andreas**  
Massachusetts Institute of  
Technology

**Wed, Feb 12, 2025**  
12-1 pm EST  
CODA 9th Floor Atrium

**GT** Georgia Tech  
Machine Learning Center

# Recall: HMMs

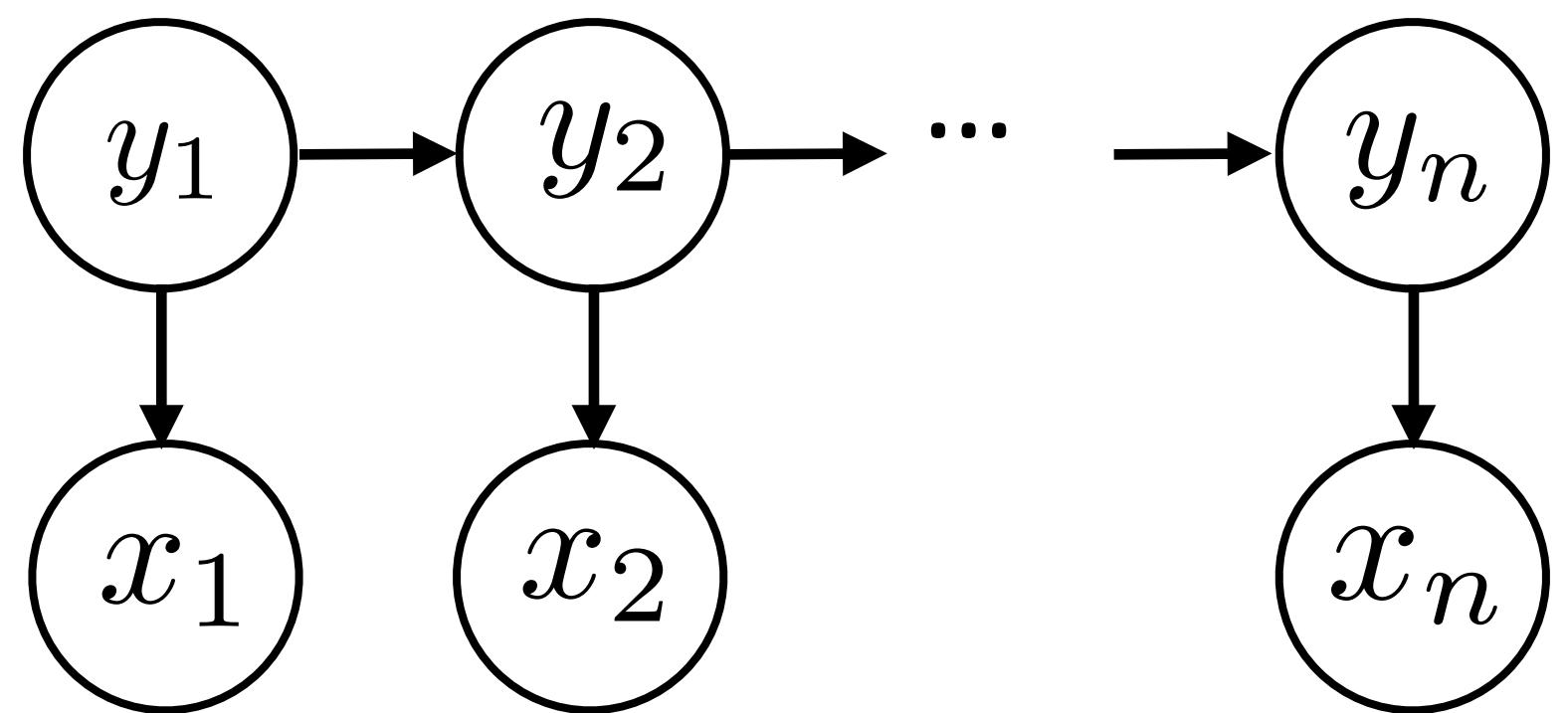
---

- ▶ Input  $\mathbf{x} = (x_1, \dots, x_n)$       Output  $\mathbf{y} = (y_1, \dots, y_n)$

# Recall: HMMs

---

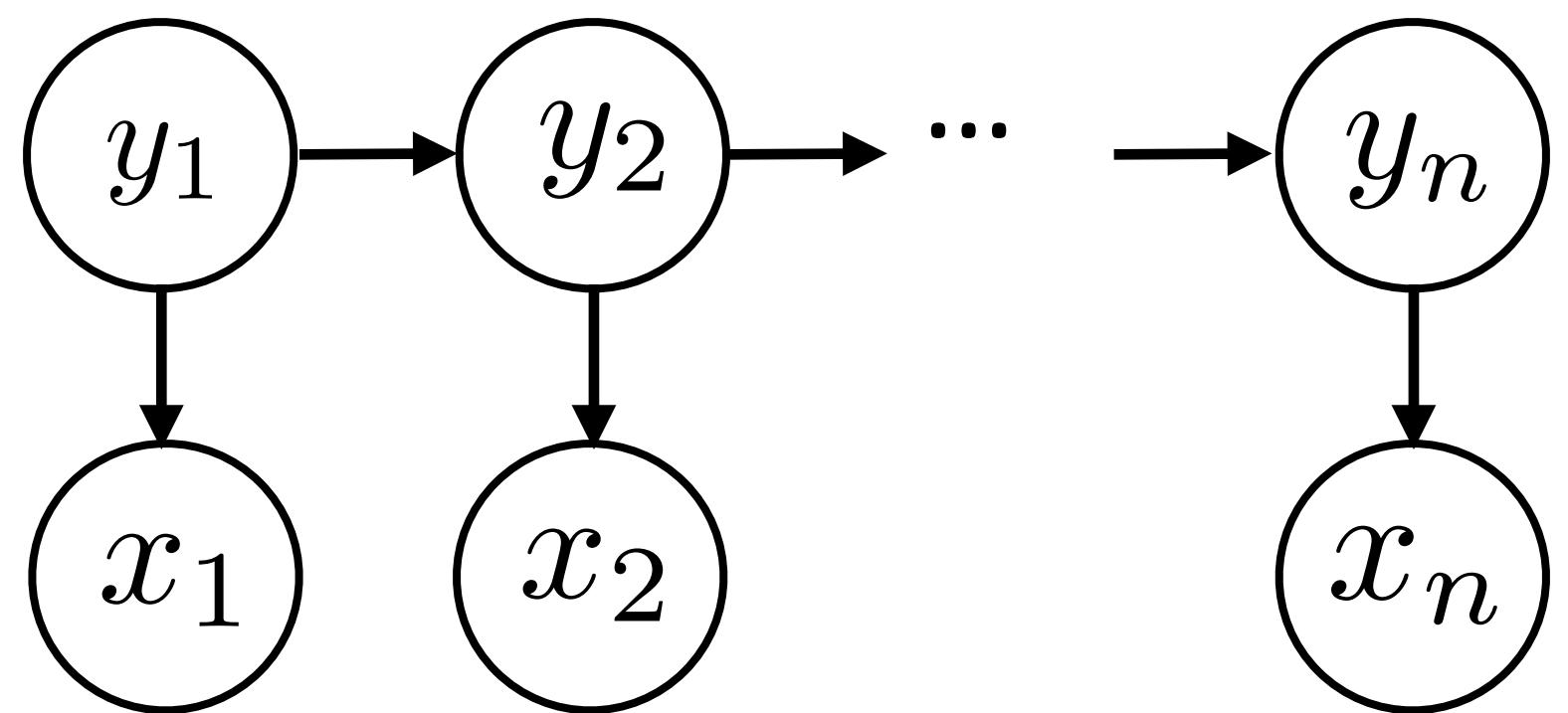
- ▶ Input  $\mathbf{x} = (x_1, \dots, x_n)$       Output  $\mathbf{y} = (y_1, \dots, y_n)$



# Recall: HMMs

---

► Input  $\mathbf{x} = (x_1, \dots, x_n)$



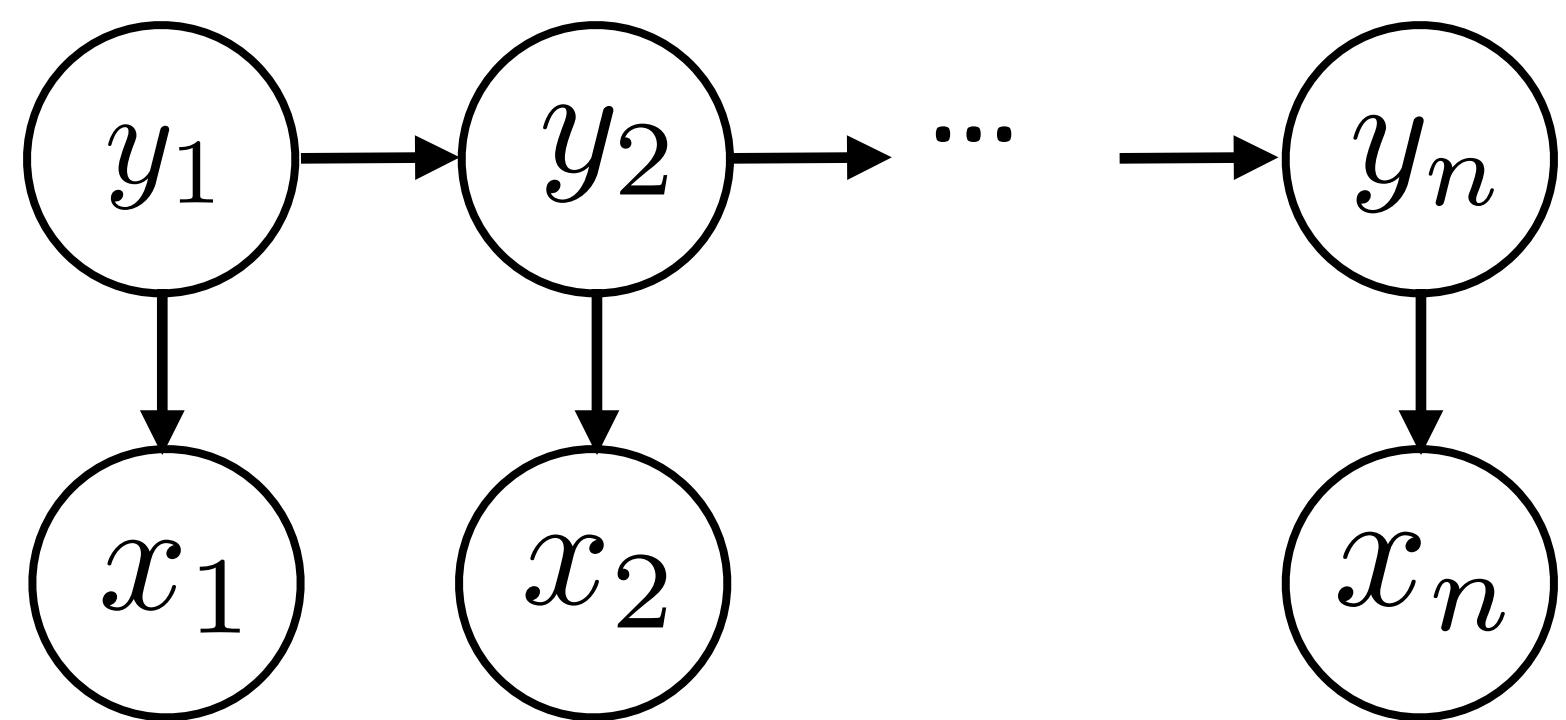
Output  $\mathbf{y} = (y_1, \dots, y_n)$

$$P(\mathbf{y}, \mathbf{x}) = P(y_1) \prod_{i=2}^n P(y_i|y_{i-1}) \prod_{i=1}^n P(x_i|y_i)$$

# Recall: HMMs

---

- ▶ Input  $\mathbf{x} = (x_1, \dots, x_n)$



- ▶ Output  $\mathbf{y} = (y_1, \dots, y_n)$

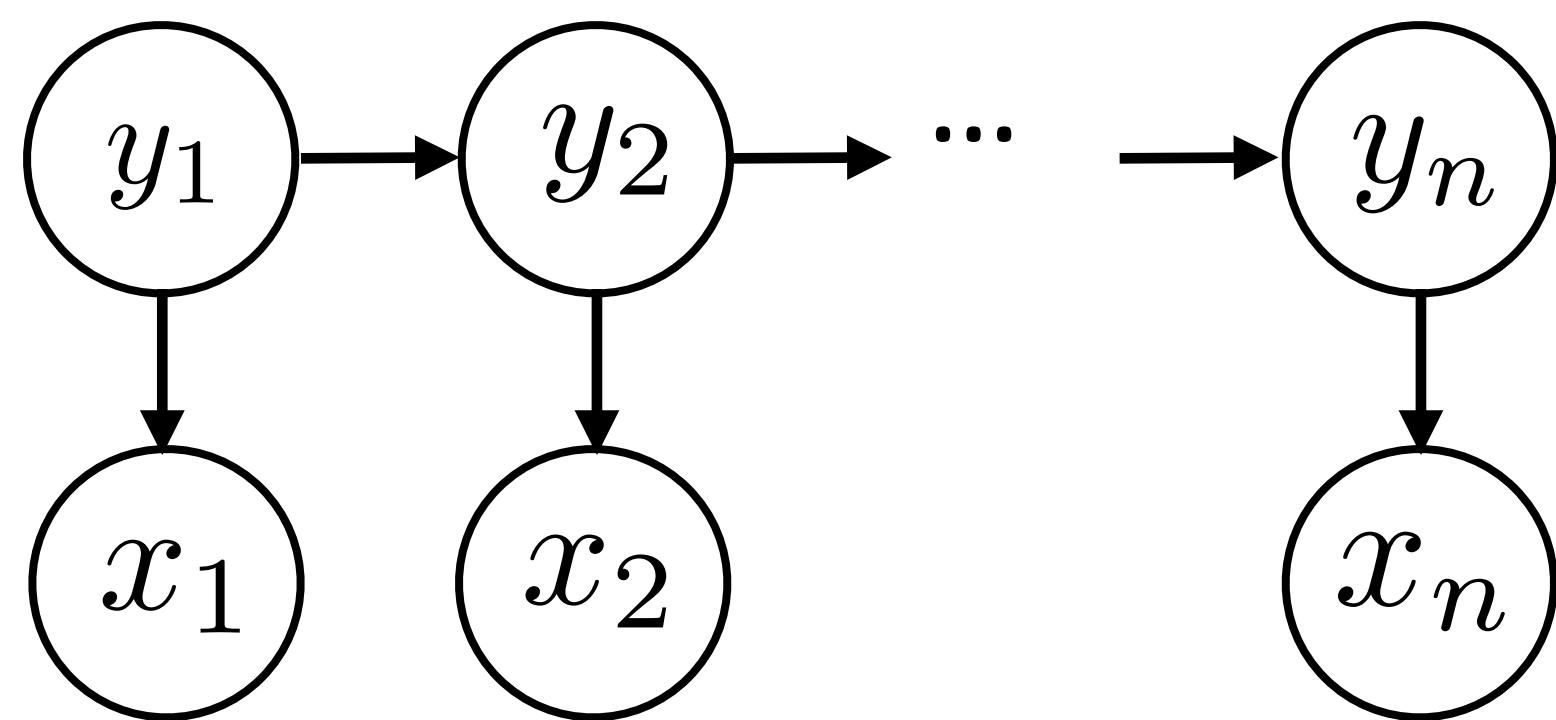
$$P(\mathbf{y}, \mathbf{x}) = P(y_1) \prod_{i=2}^n P(y_i|y_{i-1}) \prod_{i=1}^n P(x_i|y_i)$$

- ▶ Training: maximum likelihood estimation (with smoothing)

# Recall: HMMs

---

- ▶ Input  $\mathbf{x} = (x_1, \dots, x_n)$



- ▶ Output  $\mathbf{y} = (y_1, \dots, y_n)$

$$P(\mathbf{y}, \mathbf{x}) = P(y_1) \prod_{i=2}^n P(y_i|y_{i-1}) \prod_{i=1}^n P(x_i|y_i)$$

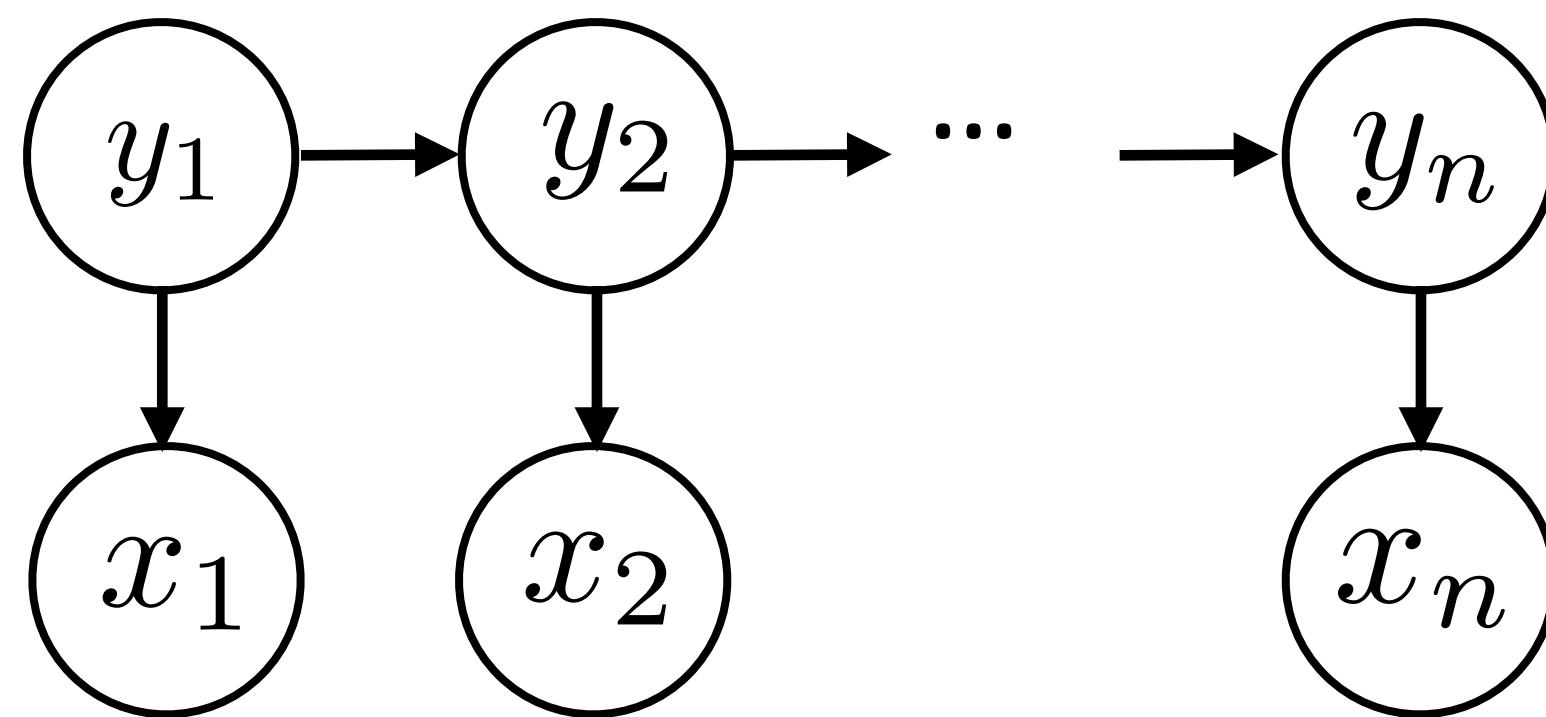
- ▶ Training: maximum likelihood estimation (with smoothing)

- ▶ Inference problem:  $\operatorname{argmax}_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}) = \operatorname{argmax}_{\mathbf{y}} \frac{P(\mathbf{y}, \mathbf{x})}{P(\mathbf{x})}$

# Recall: HMMs

---

- ▶ Input  $\mathbf{x} = (x_1, \dots, x_n)$



- ▶ Output  $\mathbf{y} = (y_1, \dots, y_n)$

$$P(\mathbf{y}, \mathbf{x}) = P(y_1) \prod_{i=2}^n P(y_i|y_{i-1}) \prod_{i=1}^n P(x_i|y_i)$$

- ▶ Training: maximum likelihood estimation (with smoothing)

- ▶ Inference problem:  $\operatorname{argmax}_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}) = \operatorname{argmax}_{\mathbf{y}} \frac{P(\mathbf{y}, \mathbf{x})}{P(\mathbf{x})}$

- ▶ Viterbi:  $\text{score}_i(s) = \max_{y_{i-1}} P(s|y_{i-1}) P(x_i|s) \text{score}_{i-1}(y_{i-1})$

# This Lecture

---

- ▶ CRFs: model (+features for NER), inference, learning
- ▶ Named entity recognition (NER)
- ▶ (if time) Beam search

# Named Entity Recognition

---

# Named Entity Recognition

---

*Barack Obama will travel to Hangzhou today for the G20 meeting .*

# Named Entity Recognition

---

*Barack Obama will travel to Hangzhou today for the G20 meeting .*

PERSON

LOC

ORG

# Named Entity Recognition

---

B-PER I-PER O O O B-LOC O O O B-ORG O O

*Barack Obama* will travel to *Hangzhou* today for the *G20* meeting .

PERSON

LOC

ORG

- ▶ BIO tagset: begin, inside, outside

# Named Entity Recognition

---

B-PER I-PER O O O B-LOC O O O B-ORG O O

*Barack Obama will travel to Hangzhou today for the G20 meeting.*

PERSON

LOC

ORG

- ▶ BIO tagset: begin, inside, outside
- ▶ Sequence of tags – should we use an HMM?
- ▶ Why might an HMM not do so well here?

# Named Entity Recognition

---

B-PER	I-PER	O	O	O	B-LOC	O	O	O	B-ORG	O	O
<i>Barack Obama</i>				<i>will travel to Hangzhou</i>				<i>today for the G20 meeting.</i>			
PERSON				LOC				ORG			

- ▶ BIO tagset: begin, inside, outside
- ▶ Sequence of tags – should we use an HMM?
- ▶ Why might an HMM not do so well here?
  - ▶ Lots of O's, so tags aren't as informative about context

# Named Entity Recognition

---

B-PER	I-PER	O	O	O	B-LOC	O	O	O	B-ORG	O	O
<i>Barack Obama</i>				<i>will travel to Hangzhou</i>				<i>today for the G20 meeting.</i>			
PERSON				LOC				ORG			

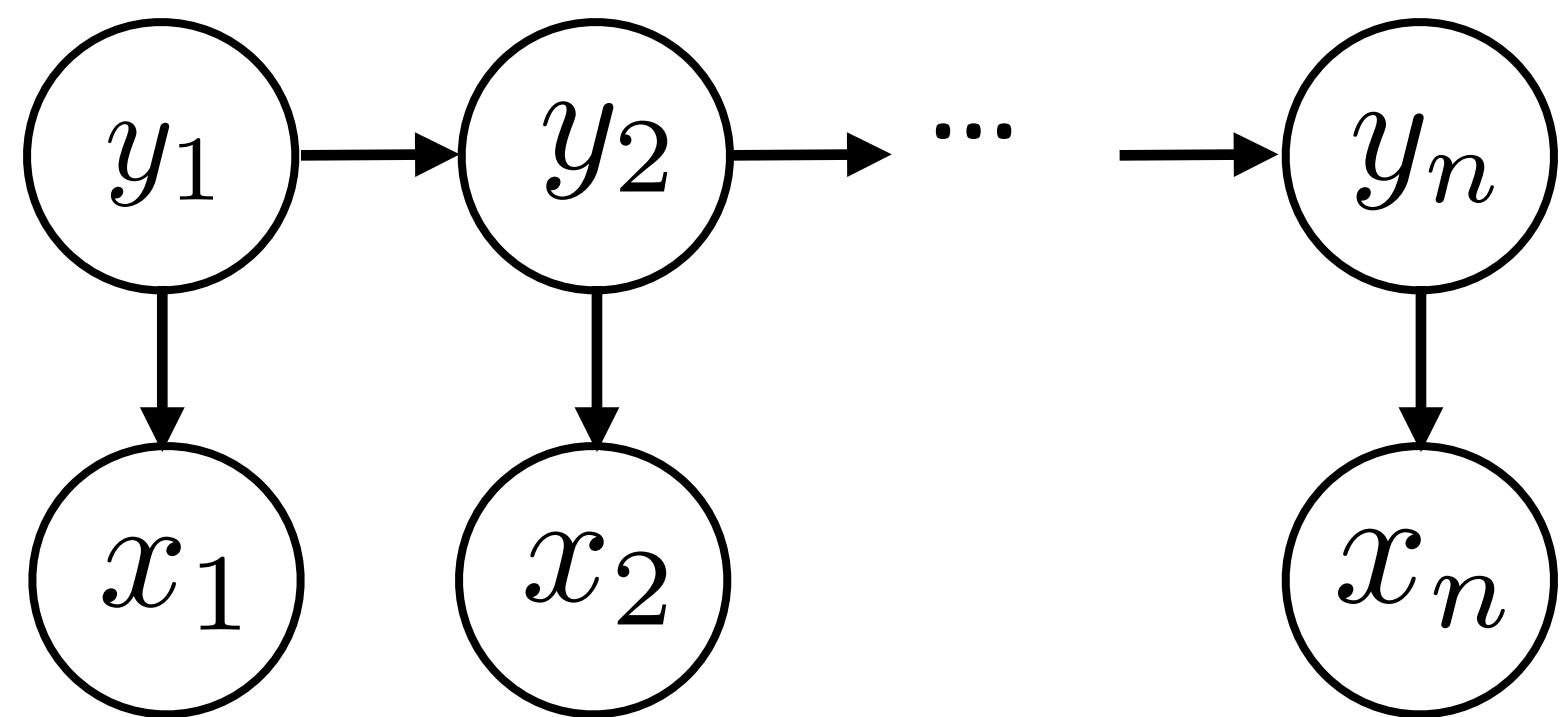
- ▶ BIO tagset: begin, inside, outside
- ▶ Sequence of tags – should we use an HMM?
- ▶ Why might an HMM not do so well here?
  - ▶ Lots of O's, so tags aren't as informative about context
  - ▶ Insufficient features/capacity with multinomials (especially for unks)

# CRFs

# Conditional Random Fields

---

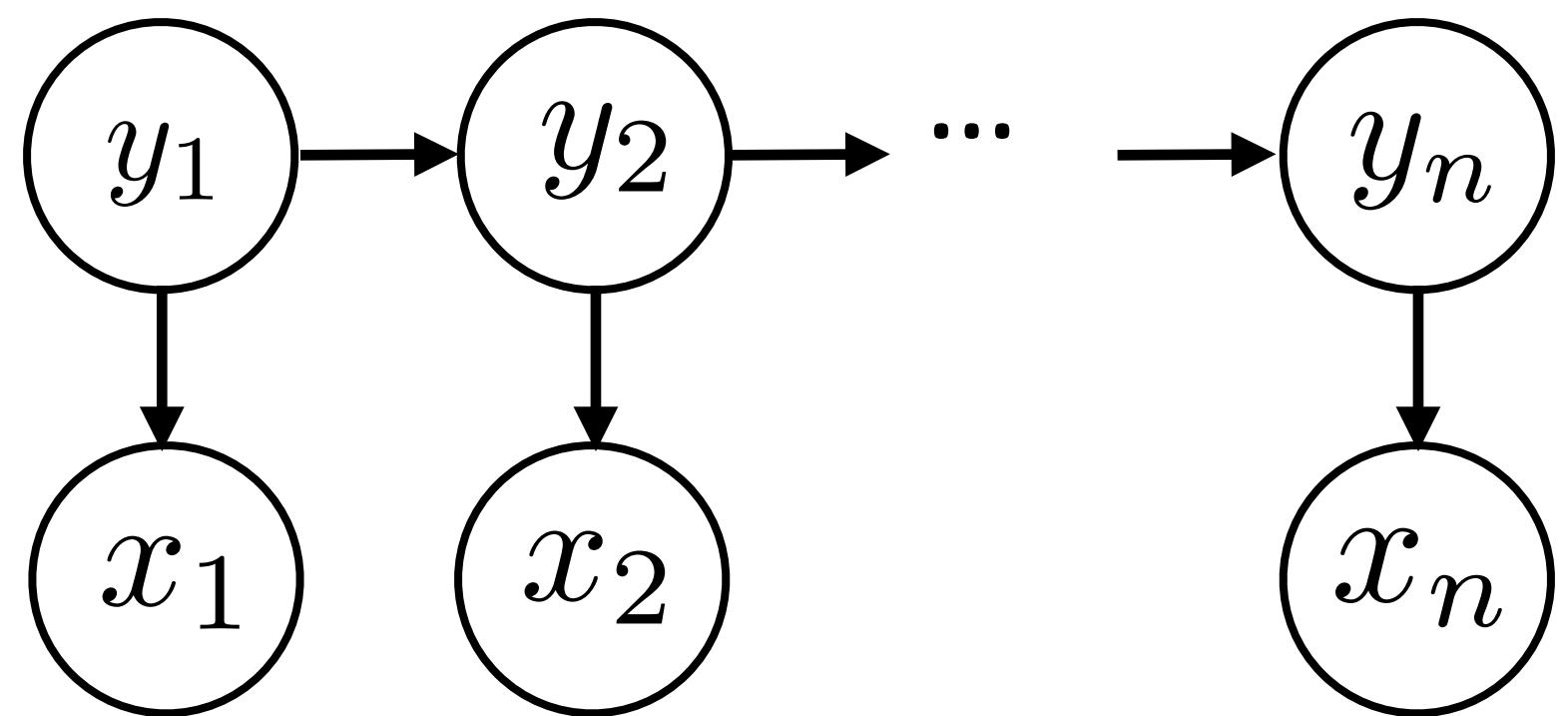
- ▶ HMMs are expressible as Bayes nets (factor graphs)



# Conditional Random Fields

---

- ▶ HMMs are expressible as Bayes nets (factor graphs)



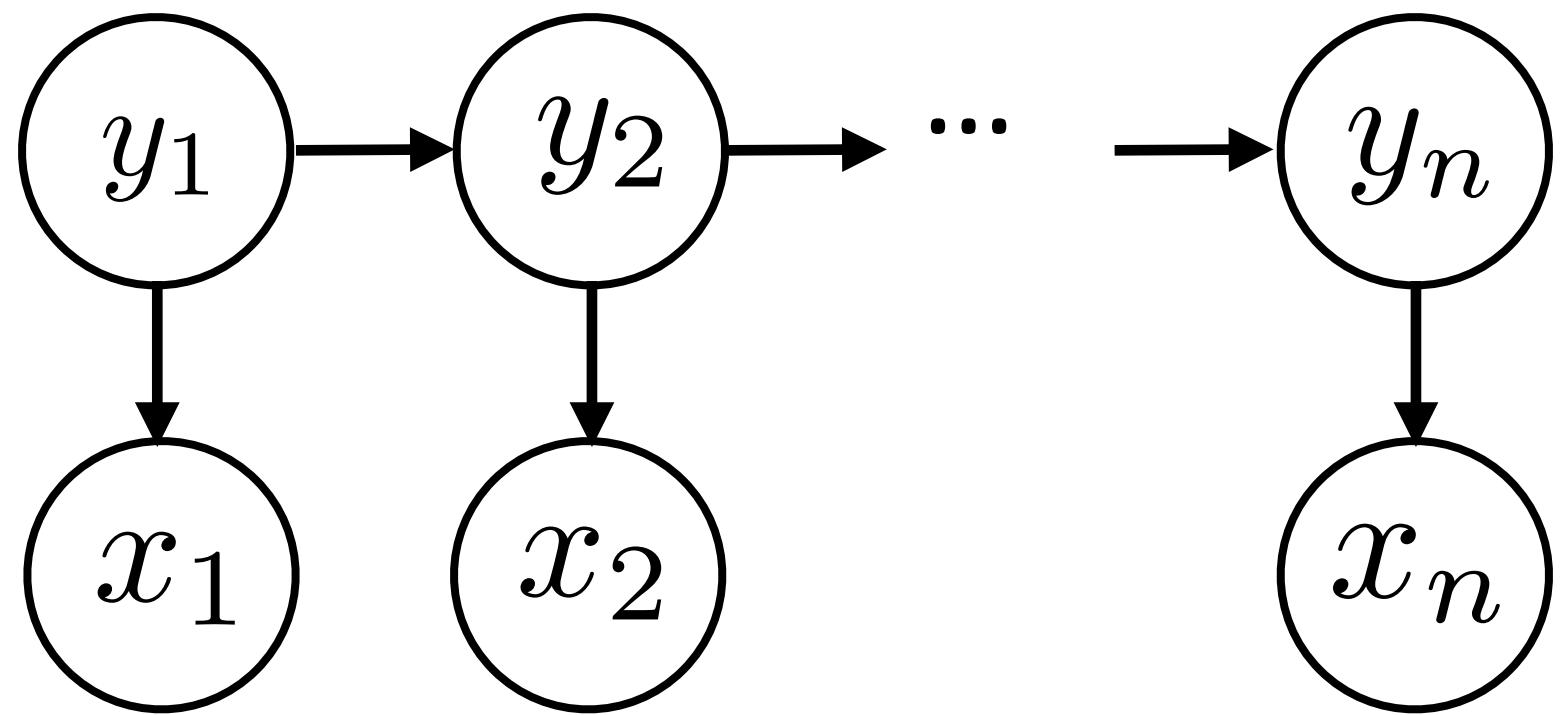
- ▶ This reflects the following decomposition:

$$P(\mathbf{y}, \mathbf{x}) = P(y_1)P(x_1|y_1)P(y_2|y_1)P(x_2|y_2)\dots$$

# Conditional Random Fields

---

- ▶ HMMs are expressible as Bayes nets (factor graphs)



- ▶ This reflects the following decomposition:

$$P(\mathbf{y}, \mathbf{x}) = P(y_1)P(x_1|y_1)P(y_2|y_1)P(x_2|y_2)\dots$$

- ▶ Locally normalized model: each factor is a probability distribution that normalizes

# Conditional Random Fields

---

- HMMs:  $P(\mathbf{y}, \mathbf{x}) = P(y_1)P(x_1|y_1)P(y_2|y_1)P(x_2|y_2)\dots$

# Conditional Random Fields

---

- ▶ HMMs:  $P(\mathbf{y}, \mathbf{x}) = P(y_1)P(x_1|y_1)P(y_2|y_1)P(x_2|y_2) \dots$
- ▶ CRFs: discriminative models with the following globally-normalized form:

# Conditional Random Fields

---

- ▶ HMMs:  $P(\mathbf{y}, \mathbf{x}) = P(y_1)P(x_1|y_1)P(y_2|y_1)P(x_2|y_2)\dots$
- ▶ CRFs: discriminative models with the following globally-normalized form:

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_k \exp(\phi_k(\mathbf{x}, \mathbf{y}))$$

# Conditional Random Fields

---

- ▶ HMMs:  $P(\mathbf{y}, \mathbf{x}) = P(y_1)P(x_1|y_1)P(y_2|y_1)P(x_2|y_2) \dots$
- ▶ CRFs: discriminative models with the following globally-normalized form:

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_k \exp(\phi_k(\mathbf{x}, \mathbf{y}))$$

  
normalizer

# Conditional Random Fields

---

- ▶ HMMs:  $P(\mathbf{y}, \mathbf{x}) = P(y_1)P(x_1|y_1)P(y_2|y_1)P(x_2|y_2) \dots$
- ▶ CRFs: discriminative models with the following globally-normalized form:

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_k \exp(\phi_k(\mathbf{x}, \mathbf{y}))$$

↑ any real-valued scoring function of its arguments

normalizer

# Conditional Random Fields

---

- ▶ HMMs:  $P(\mathbf{y}, \mathbf{x}) = P(y_1)P(x_1|y_1)P(y_2|y_1)P(x_2|y_2) \dots$
- ▶ CRFs: discriminative models with the following globally-normalized form:

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_k \exp(\phi_k(\mathbf{x}, \mathbf{y}))$$

↑  
normalizer                          any real-valued scoring function of its arguments

- ▶ Naive Bayes : logistic regression :: HMMs : CRFs  
local vs. global normalization <-> generative vs. discriminative

# Conditional Random Fields

- HMMs:  $P(\mathbf{y}, \mathbf{x}) = P(y_1)P(x_1|y_1)P(y_2|y_1)P(x_2|y_2)\dots$
  - CRFs: discriminative models with the following globally-normalized form:

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_k \exp(\phi_k(\mathbf{x}, \mathbf{y}))$$

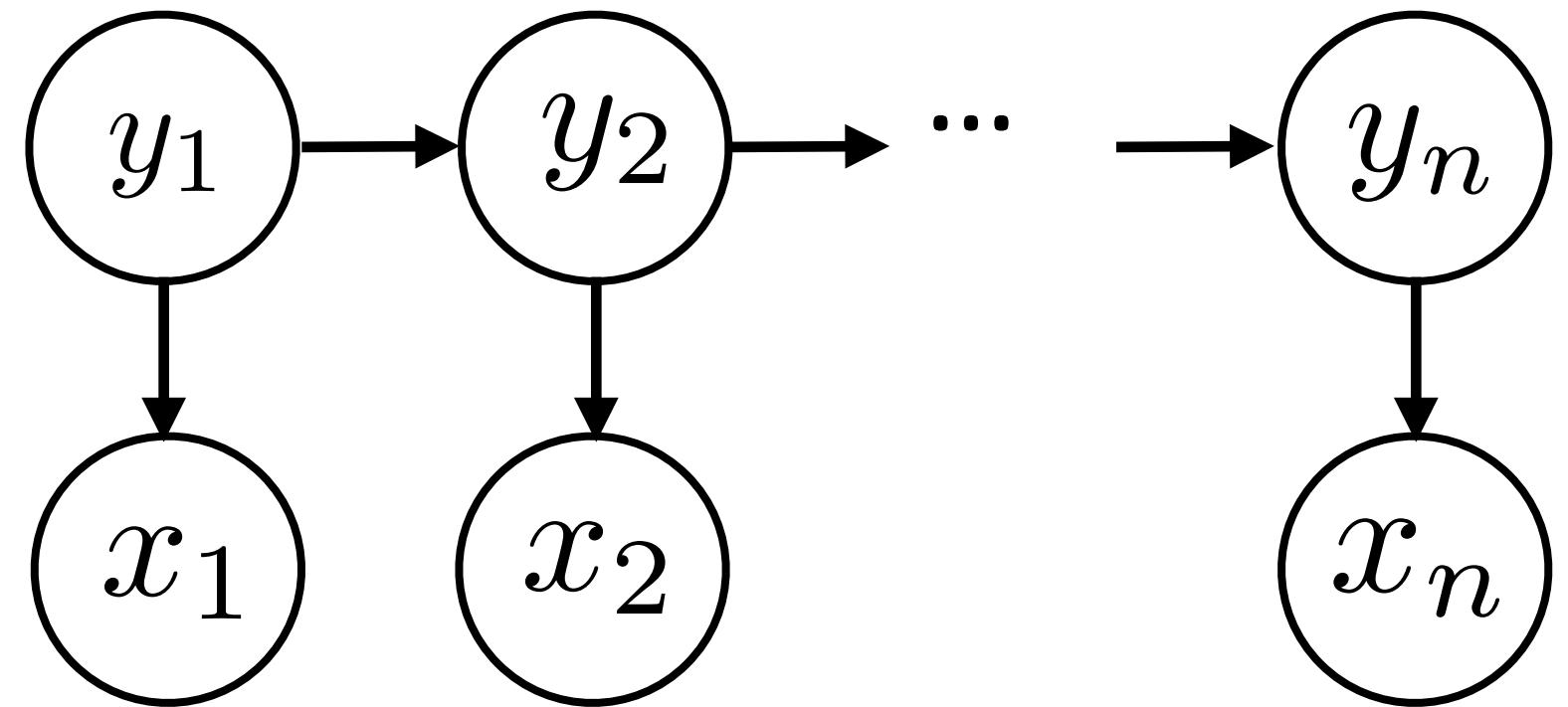
normalizer  any real-valued scoring function of its arguments 

- ▶ Naive Bayes : logistic regression :: HMMs : CRFs  
local vs. global normalization  $\leftrightarrow$  generative vs. discriminative
  - ▶ Locally normalized discriminative models do exist (MEMMs)

# Sequential CRFs

---

- HMMs:  $P(\mathbf{y}, \mathbf{x}) = P(y_1)P(x_1|y_1)P(y_2|y_1)P(x_2|y_2)\dots$



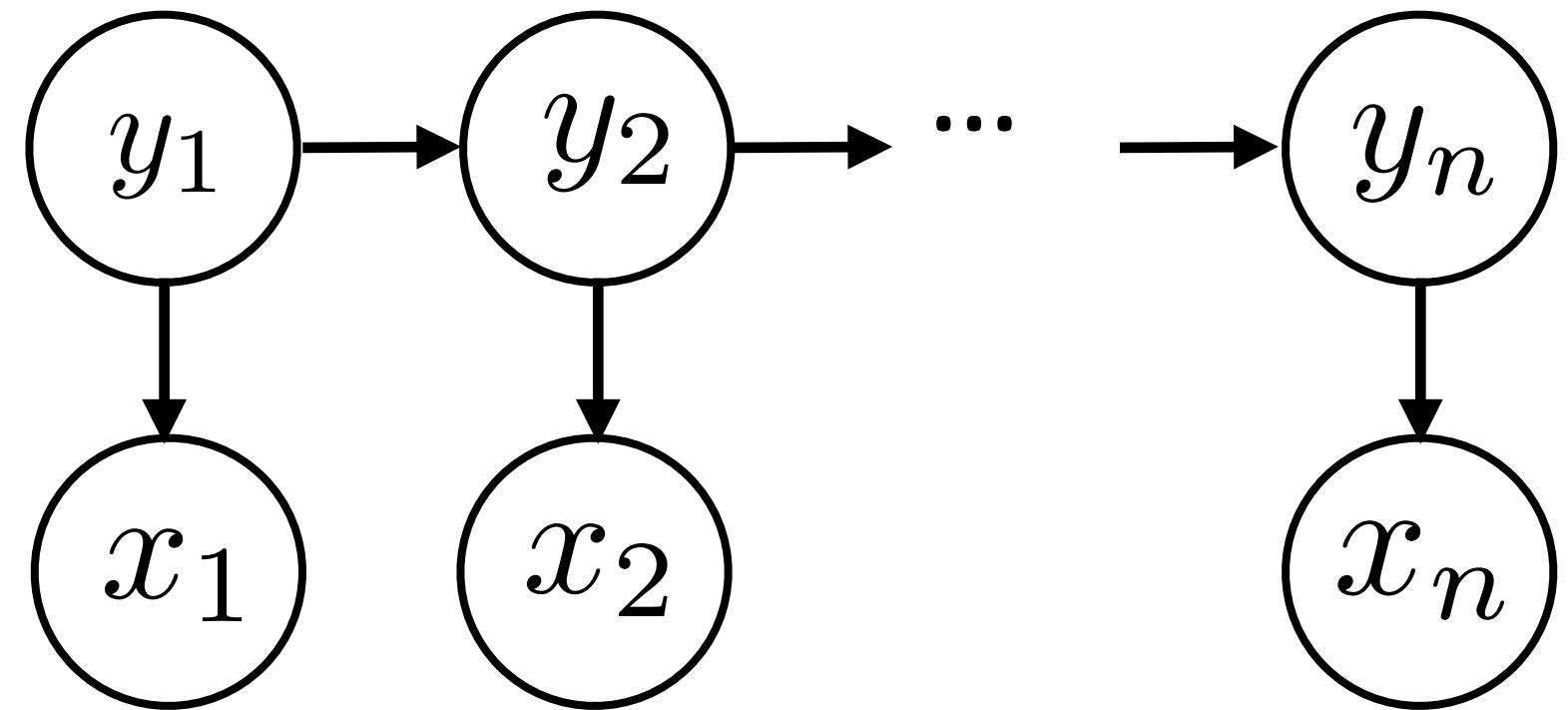
- CRFs:

$$P(\mathbf{y}|\mathbf{x}) \propto \prod_k \exp(\phi_k(\mathbf{x}, \mathbf{y}))$$

# Sequential CRFs

---

- HMMs:  $P(\mathbf{y}, \mathbf{x}) = P(y_1)P(x_1|y_1)P(y_2|y_1)P(x_2|y_2)\dots$



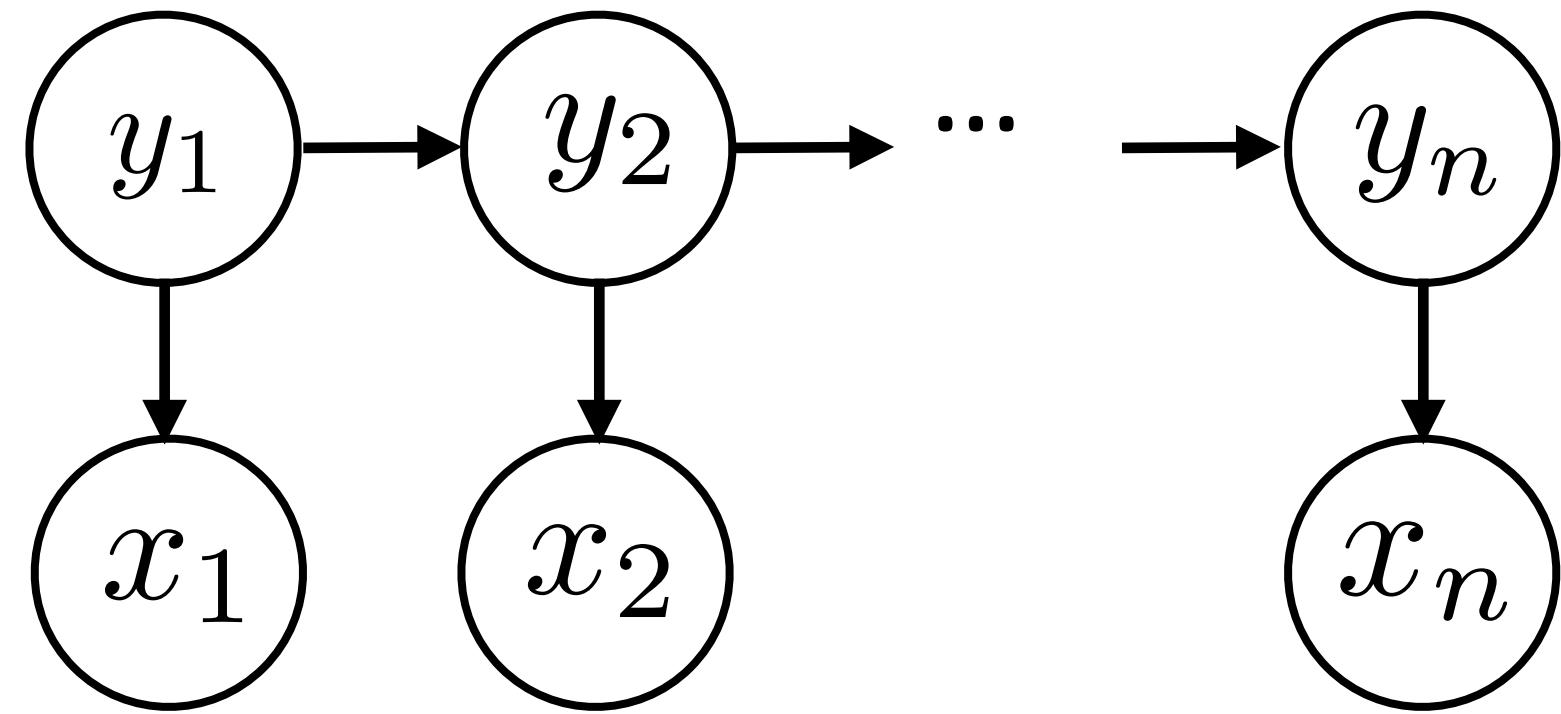
- CRFs:

$$P(\mathbf{y}|\mathbf{x}) \propto \prod_k \exp(\phi_k(\mathbf{x}, \mathbf{y}))$$

$$P(\mathbf{y}|\mathbf{x}) \propto \exp(\phi_o(y_1)) \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(x_i, y_i))$$

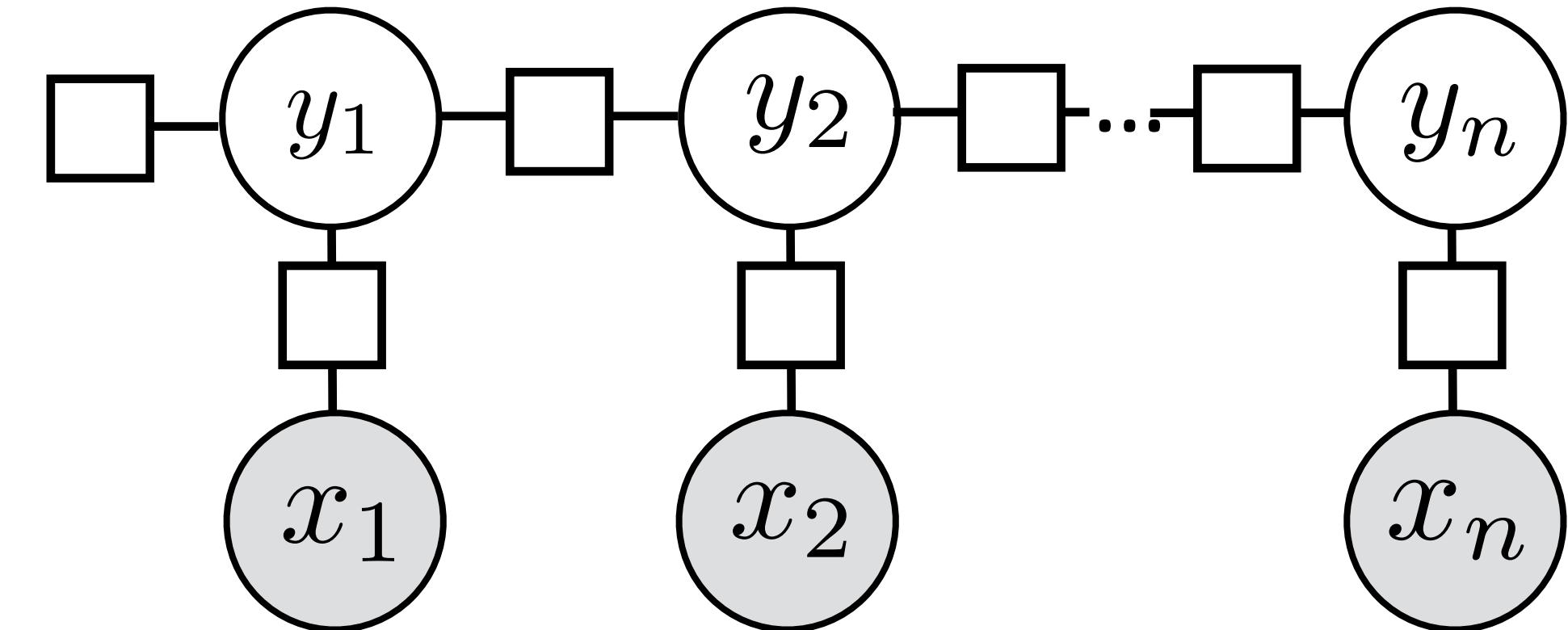
# Sequential CRFs

- HMMs:  $P(\mathbf{y}, \mathbf{x}) = P(y_1)P(x_1|y_1)P(y_2|y_1)P(x_2|y_2) \dots$



- CRFs:

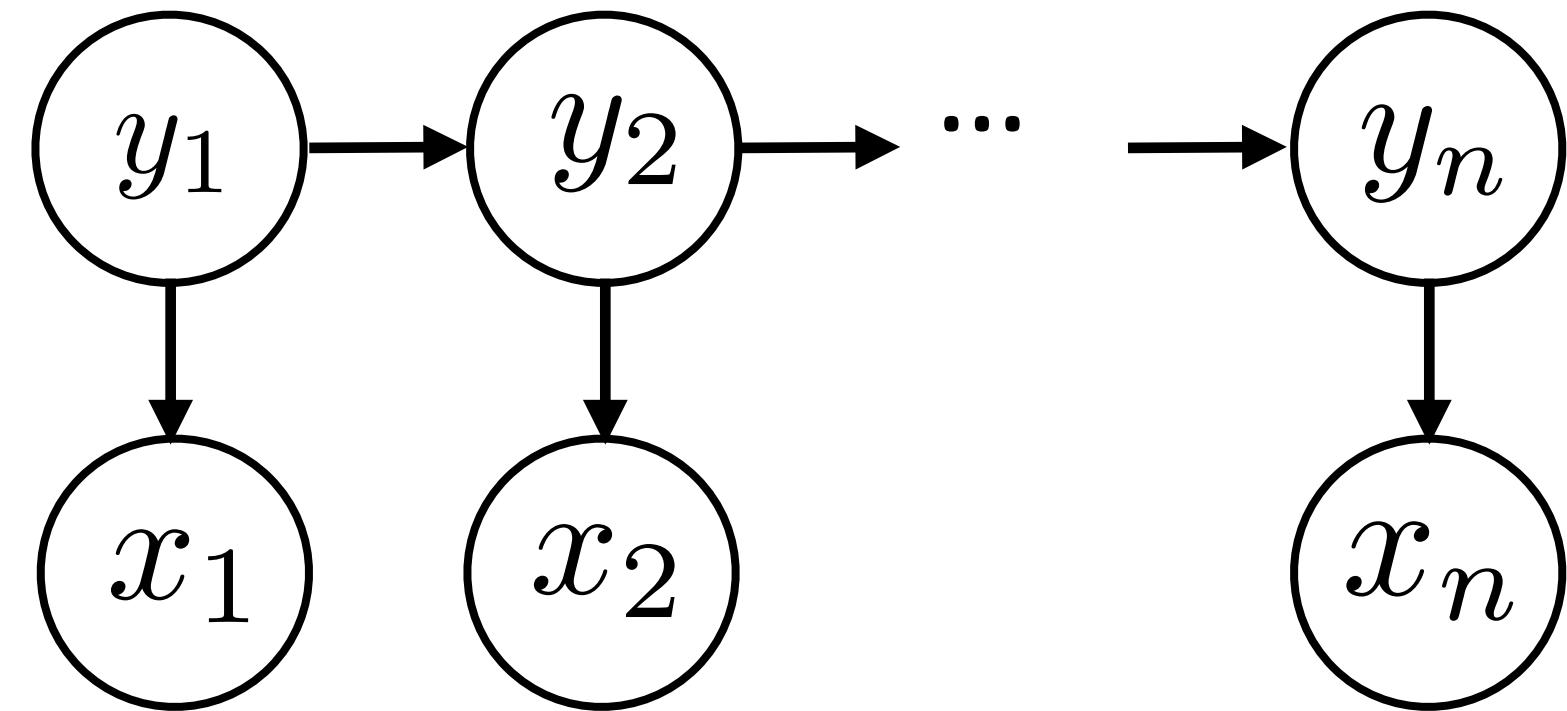
$$P(\mathbf{y}|\mathbf{x}) \propto \prod_k \exp(\phi_k(\mathbf{x}, \mathbf{y}))$$



$$P(\mathbf{y}|\mathbf{x}) \propto \exp(\phi_o(y_1)) \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(x_i, y_i))$$

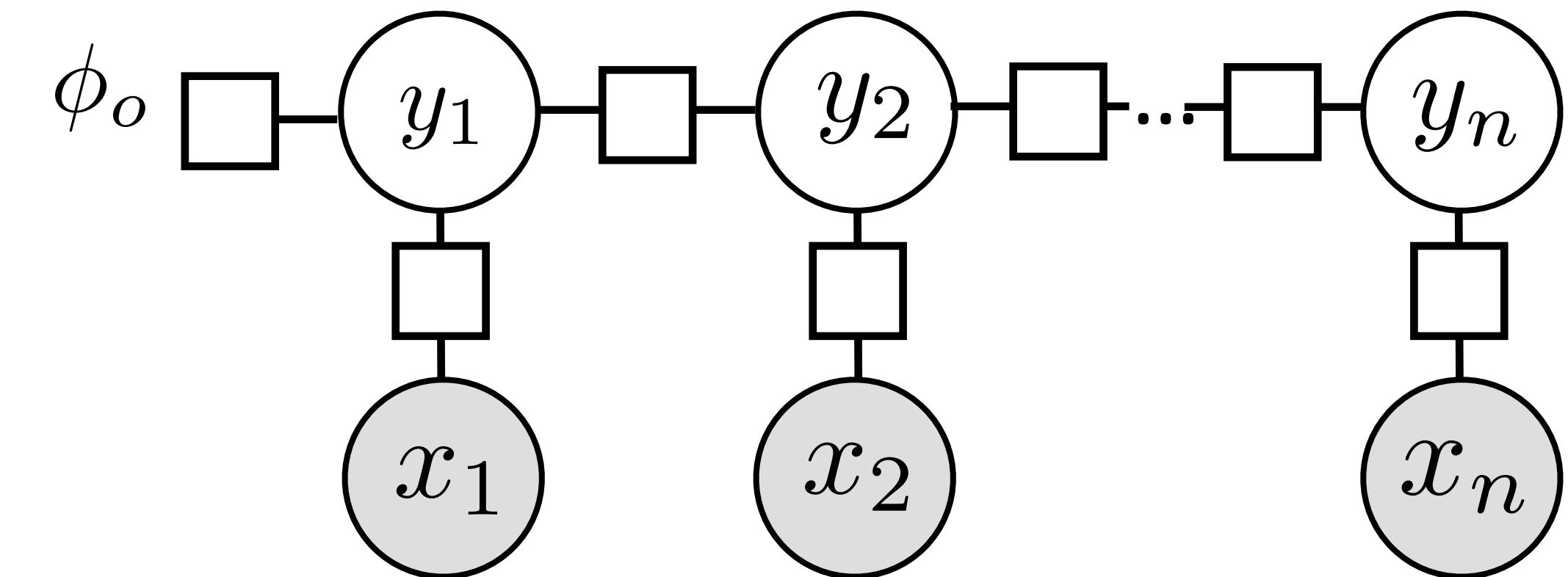
# Sequential CRFs

- HMMs:  $P(\mathbf{y}, \mathbf{x}) = P(y_1)P(x_1|y_1)P(y_2|y_1)P(x_2|y_2) \dots$



- CRFs:

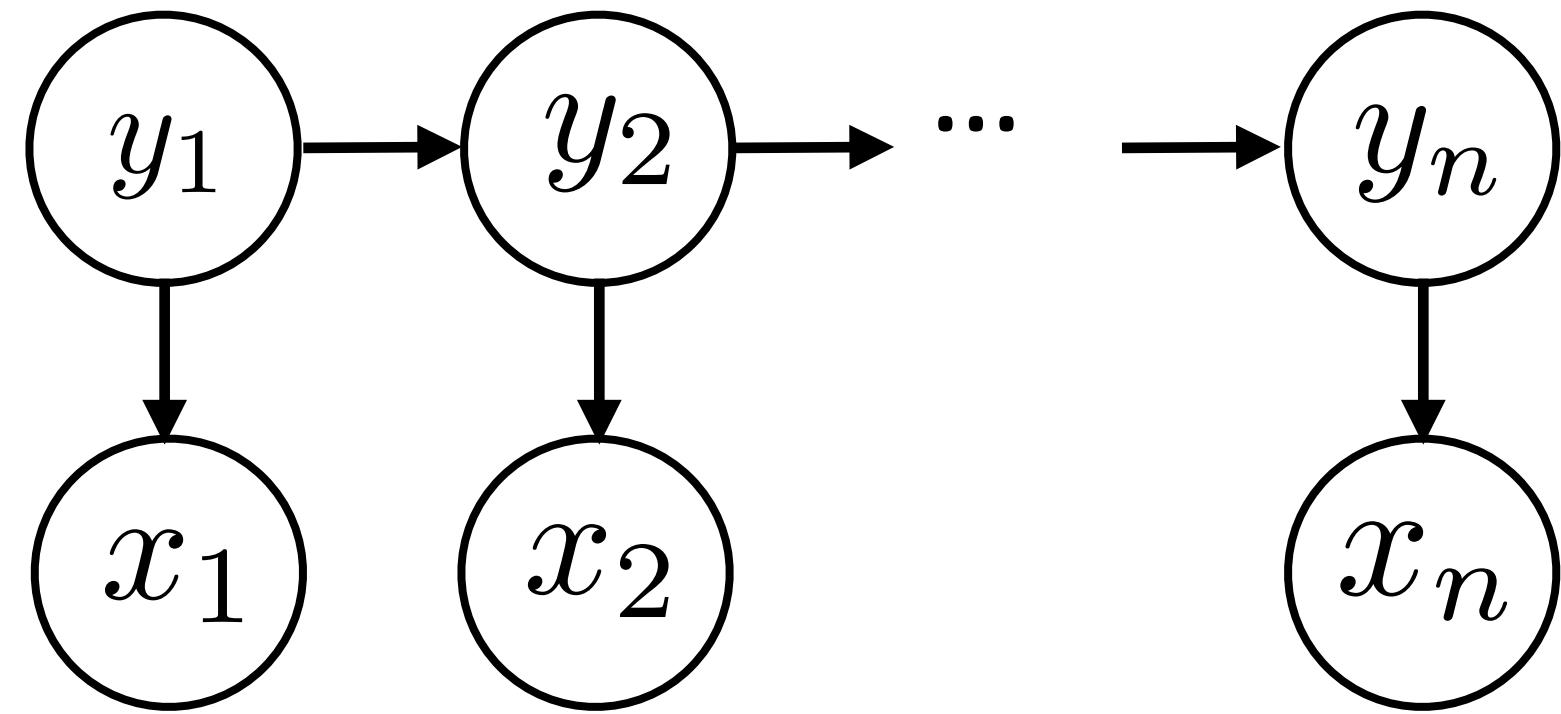
$$P(\mathbf{y}|\mathbf{x}) \propto \prod_k \exp(\phi_k(\mathbf{x}, \mathbf{y}))$$



$$P(\mathbf{y}|\mathbf{x}) \propto \exp(\phi_o(y_1)) \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(x_i, y_i))$$

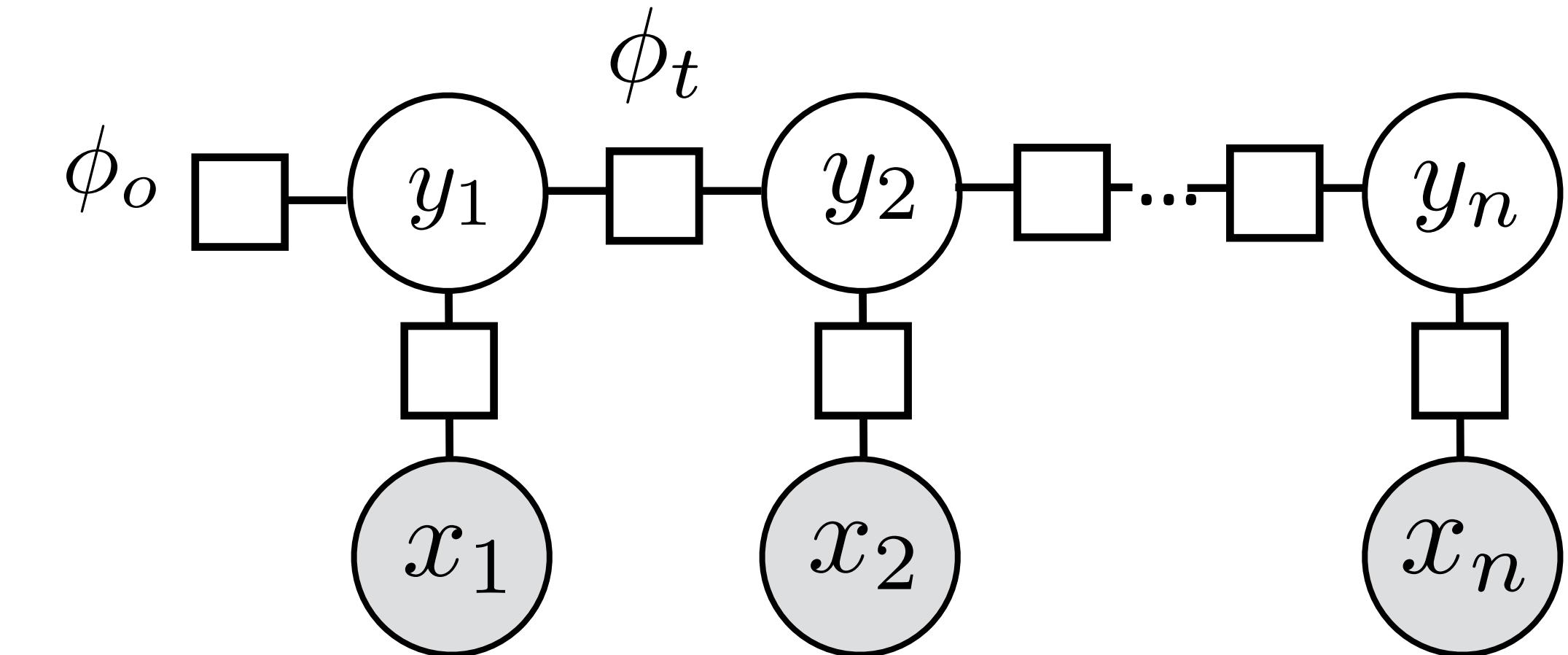
# Sequential CRFs

- HMMs:  $P(\mathbf{y}, \mathbf{x}) = P(y_1)P(x_1|y_1)P(y_2|y_1)P(x_2|y_2) \dots$



- CRFs:

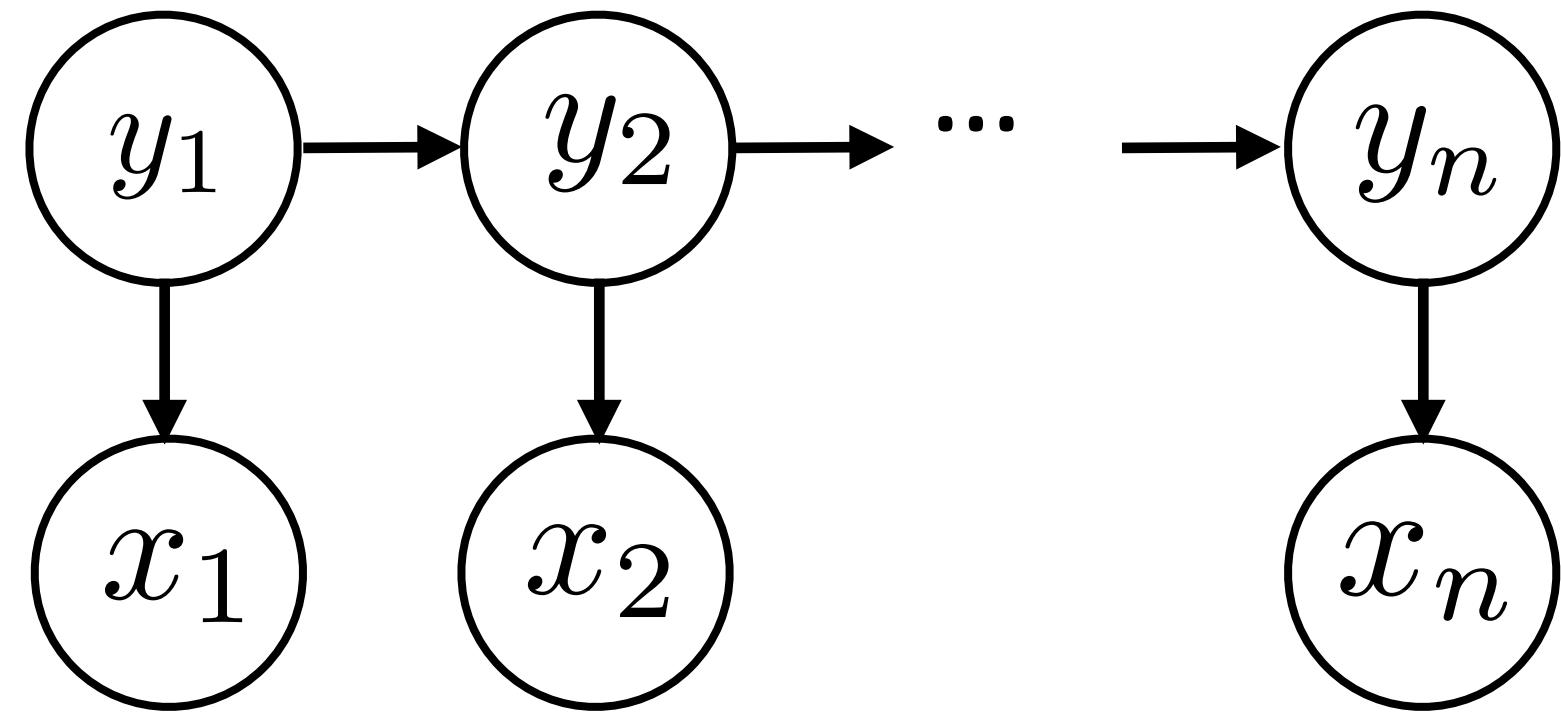
$$P(\mathbf{y}|\mathbf{x}) \propto \prod_k \exp(\phi_k(\mathbf{x}, \mathbf{y}))$$



$$P(\mathbf{y}|\mathbf{x}) \propto \exp(\phi_o(y_1)) \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(x_i, y_i))$$

# Sequential CRFs

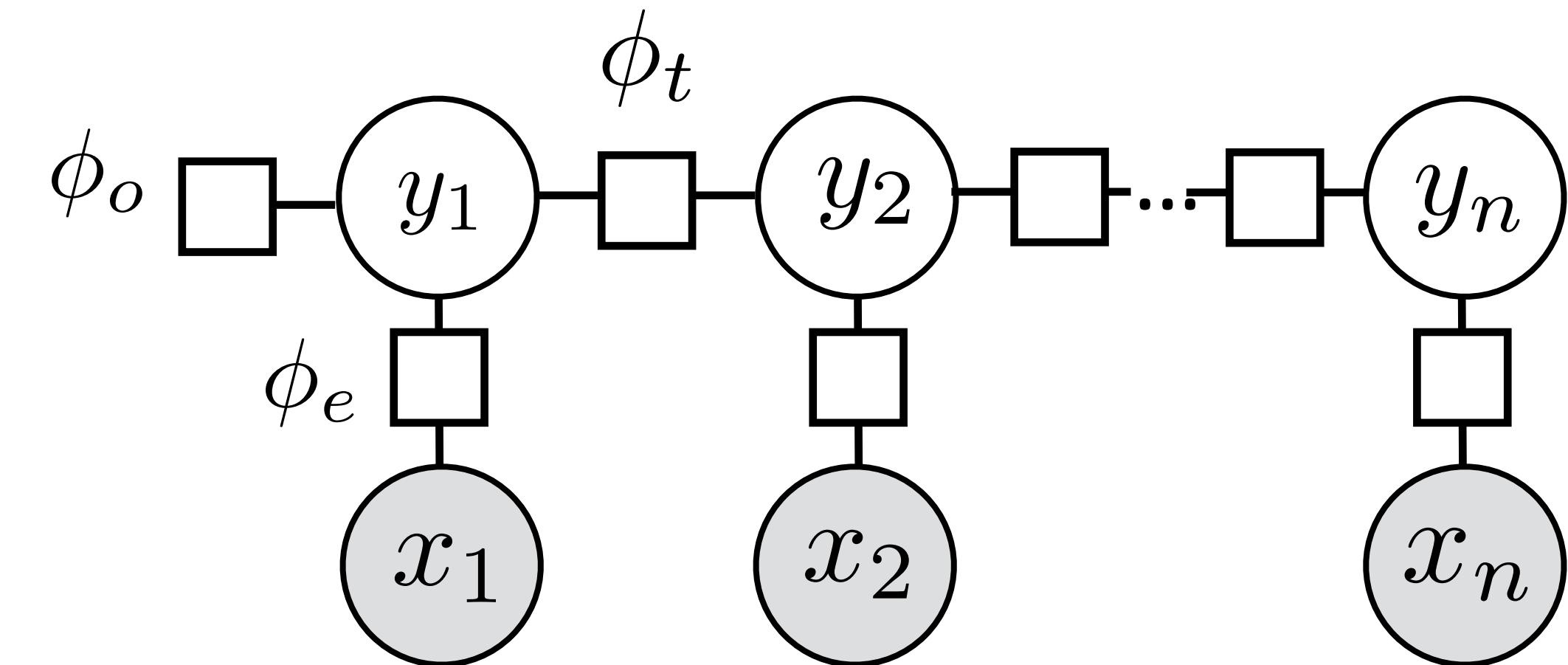
- HMMs:  $P(\mathbf{y}, \mathbf{x}) = P(y_1)P(x_1|y_1)P(y_2|y_1)P(x_2|y_2) \dots$



- CRFs:

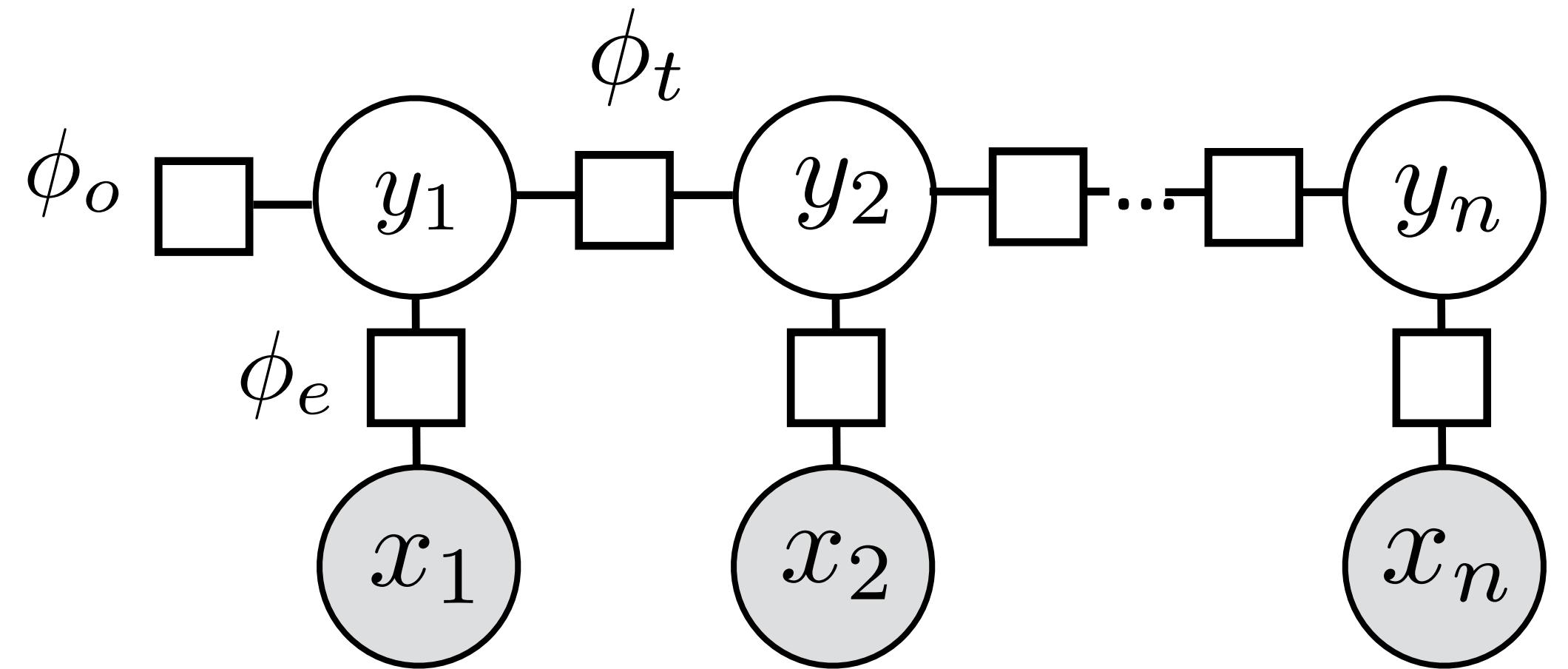
$$P(\mathbf{y}|\mathbf{x}) \propto \prod_k \exp(\phi_k(\mathbf{x}, \mathbf{y}))$$

$$P(\mathbf{y}|\mathbf{x}) \propto \exp(\phi_o(y_1)) \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(x_i, y_i))$$



# Sequential CRFs

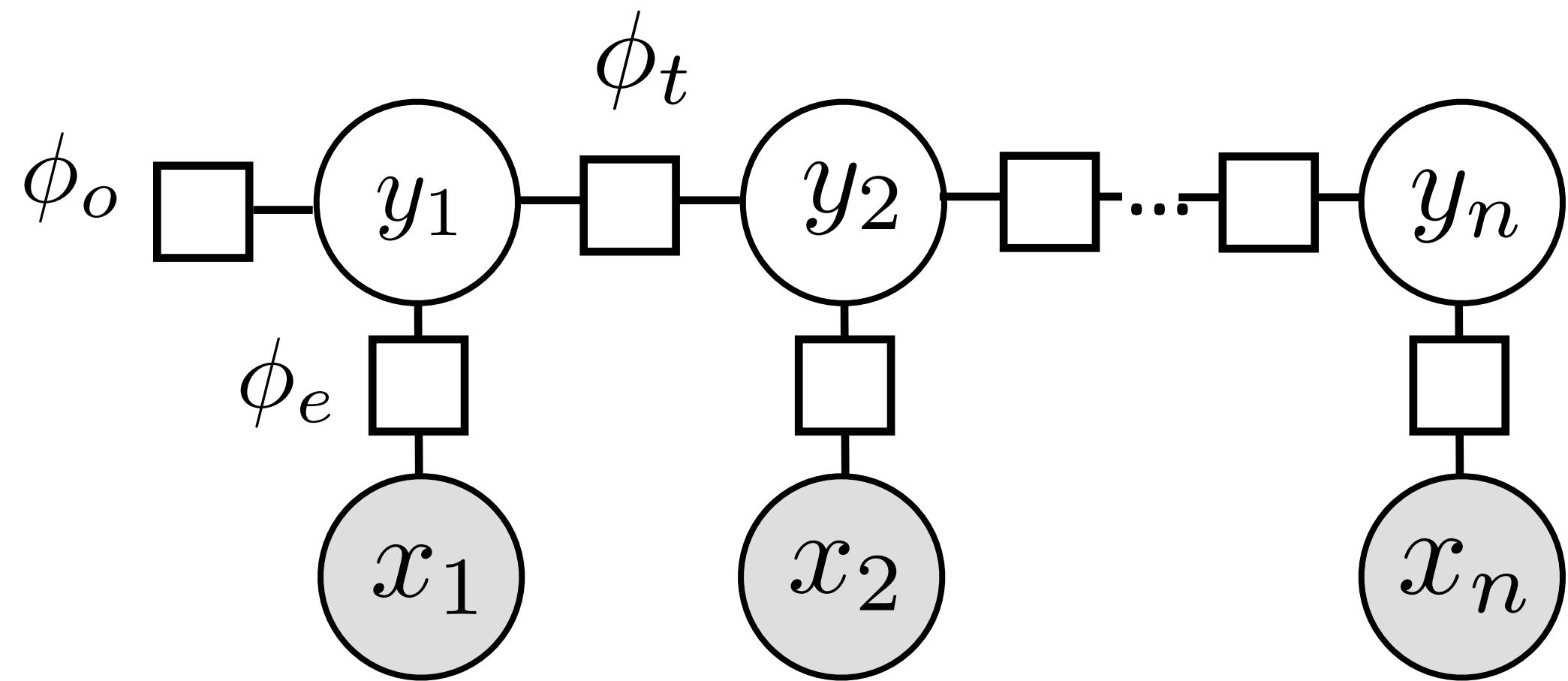
---



$$P(\mathbf{y}|\mathbf{x}) \propto \exp(\phi_o(y_1)) \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(x_i, y_i))$$

# Sequential CRFs

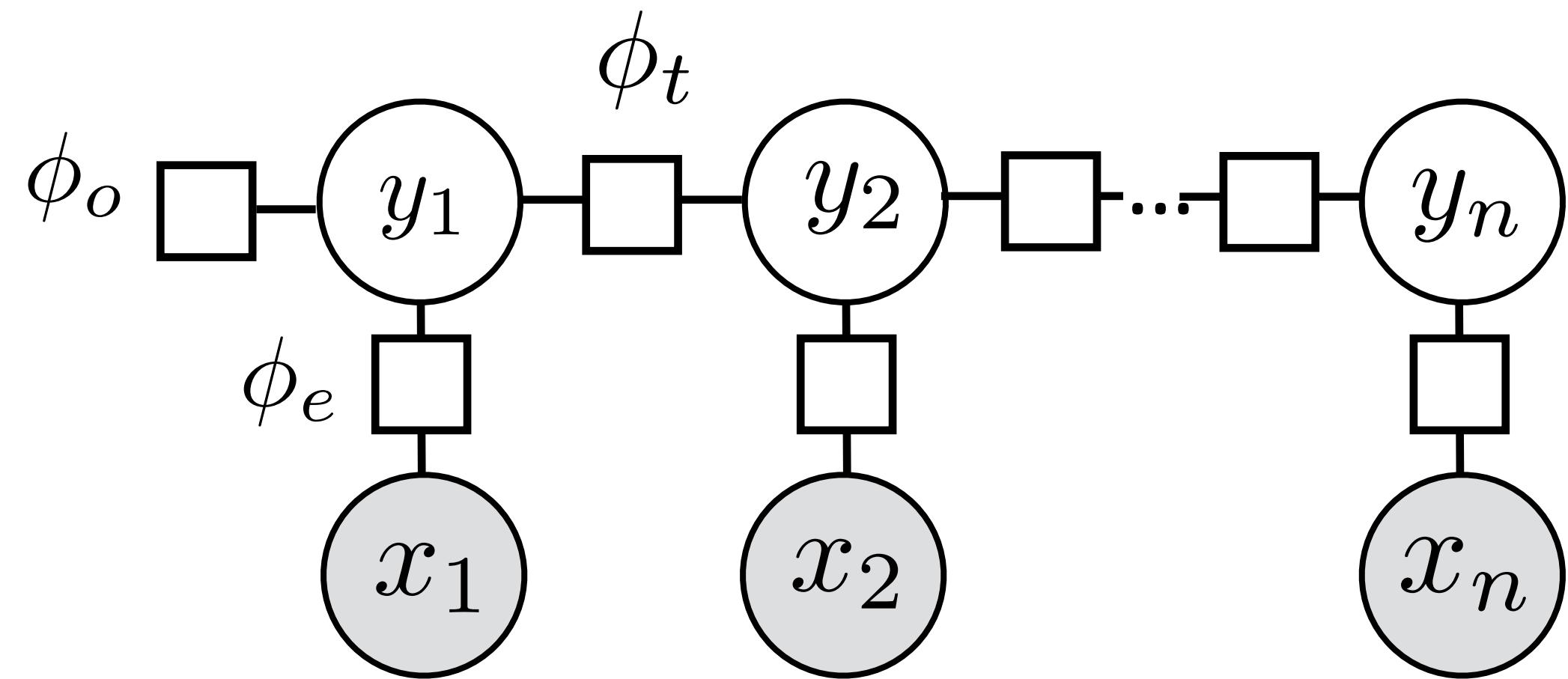
---



$$P(\mathbf{y}|\mathbf{x}) \propto \exp(\phi_o(y_1)) \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(x_i, y_i))$$

- ▶ We condition on  $\mathbf{x}$ , so every factor can depend on all of  $\mathbf{x}$  (including transitions, but we won't do this)

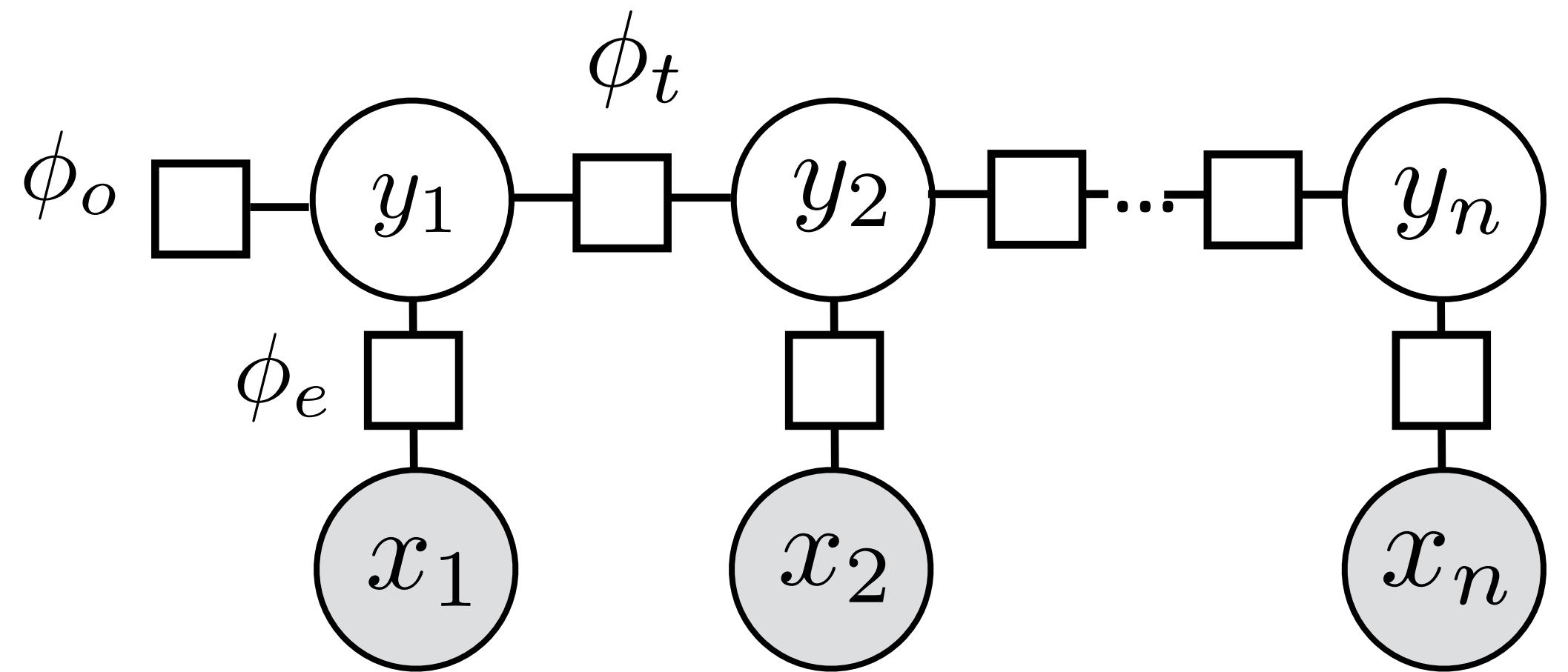
# Sequential CRFs



$$P(\mathbf{y}|\mathbf{x}) \propto \exp(\phi_o(y_1)) \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(x_i, y_i)) \rightarrow \prod_{i=1}^n \exp(\phi_e(y_i, i, \mathbf{x}))$$

- We condition on  $\mathbf{x}$ , so every factor can depend on all of  $\mathbf{x}$  (including transitions, but we won't do this)

# Sequential CRFs



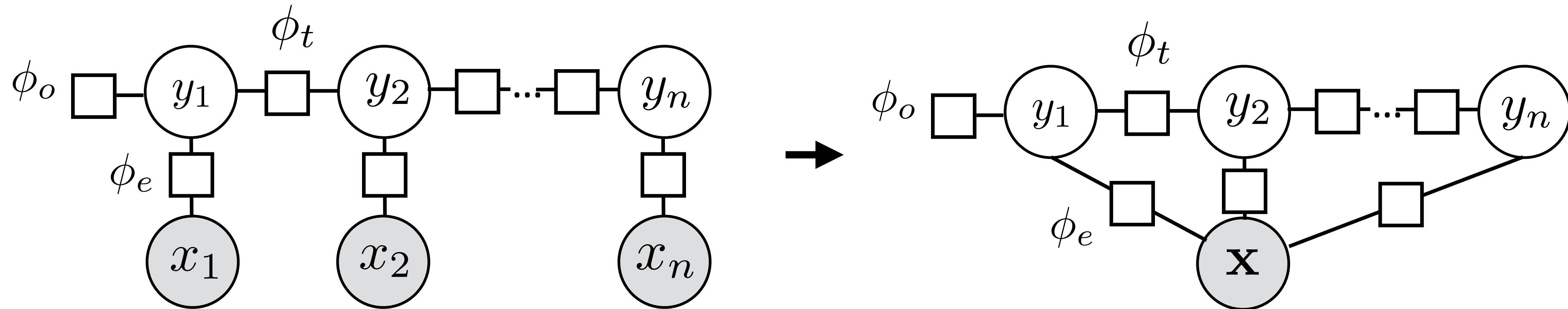
$$P(\mathbf{y}|\mathbf{x}) \propto \exp(\phi_o(y_1)) \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(x_i, y_i))$$

- We condition on  $\mathbf{x}$ , so every factor can depend on all of  $\mathbf{x}$  (including transitions, but we won't do this)

$\prod_{i=1}^n \exp(\phi_e(y_i, i, \mathbf{x}))$

token index — lets us look at current word

# Sequential CRFs



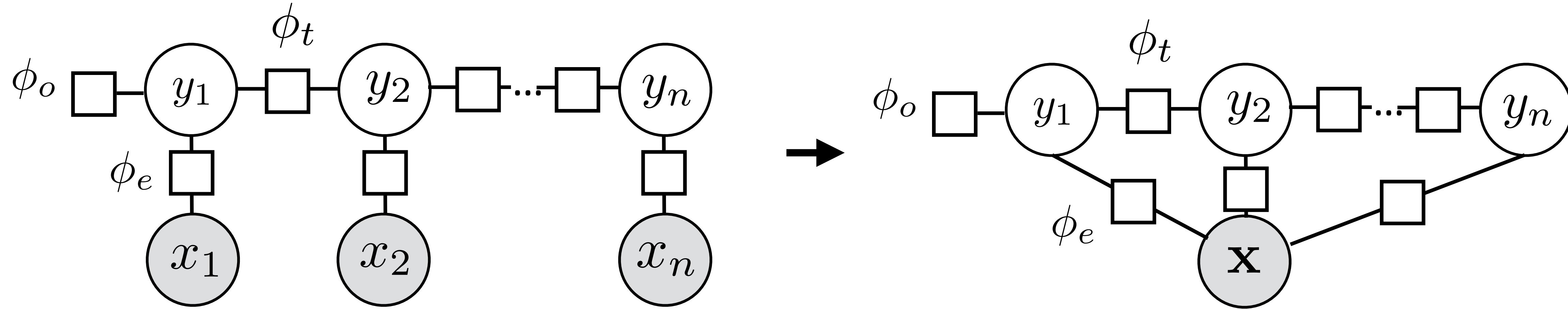
$$P(\mathbf{y}|\mathbf{x}) \propto \exp(\phi_o(y_1)) \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(x_i, y_i))$$

- We condition on  $\mathbf{x}$ , so every factor can depend on all of  $\mathbf{x}$  (including transitions, but we won't do this)

$\prod_{i=1}^n \exp(\phi_e(y_i, i, \mathbf{x}))$

token index — lets us look at current word

# Sequential CRFs



$$P(\mathbf{y}|\mathbf{x}) \propto \exp(\phi_o(y_1)) \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(x_i, y_i))$$

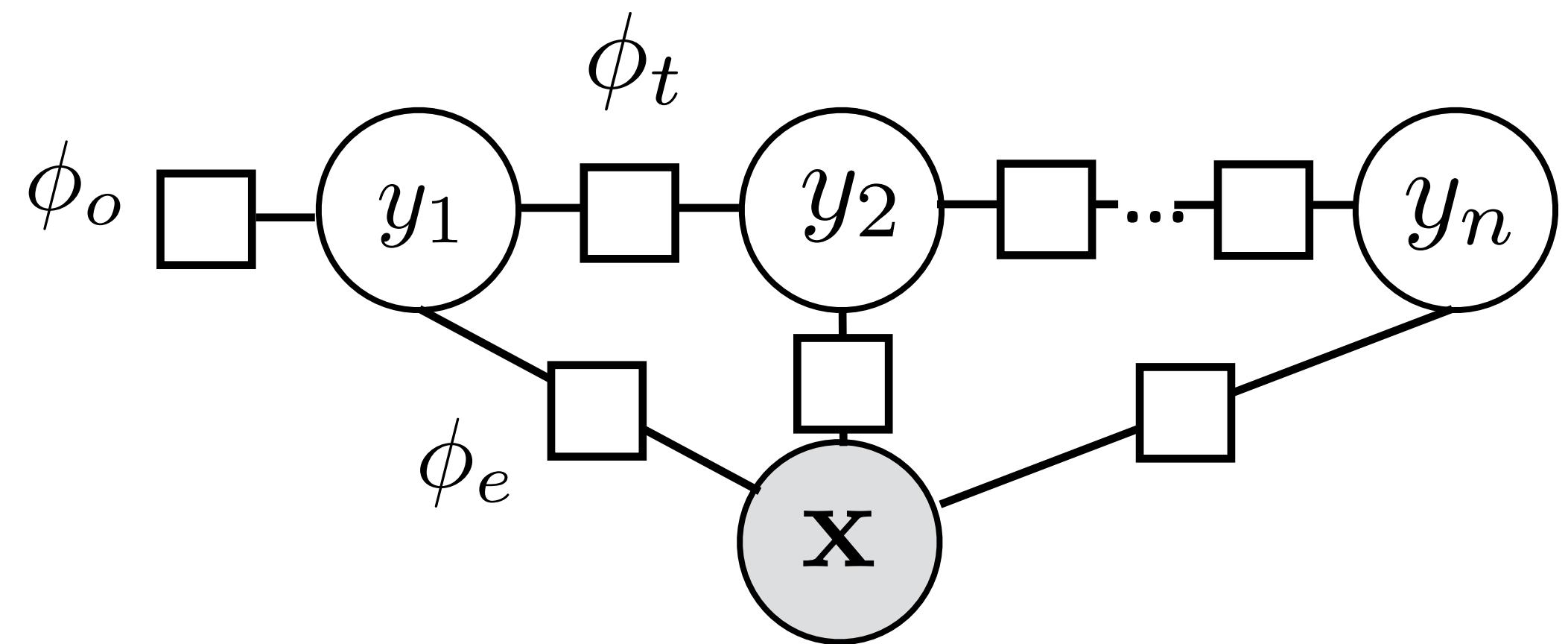
- ▶ We condition on  $\mathbf{x}$ , so every factor can depend on all of  $\mathbf{x}$  (including transitions, but we won't do this)
- ▶  $\mathbf{y}$  can't depend arbitrarily on  $\mathbf{x}$  in a generative model

$$\prod_{i=1}^n \exp(\phi_e(y_i, i, \mathbf{x}))$$

token index — lets us look at current word

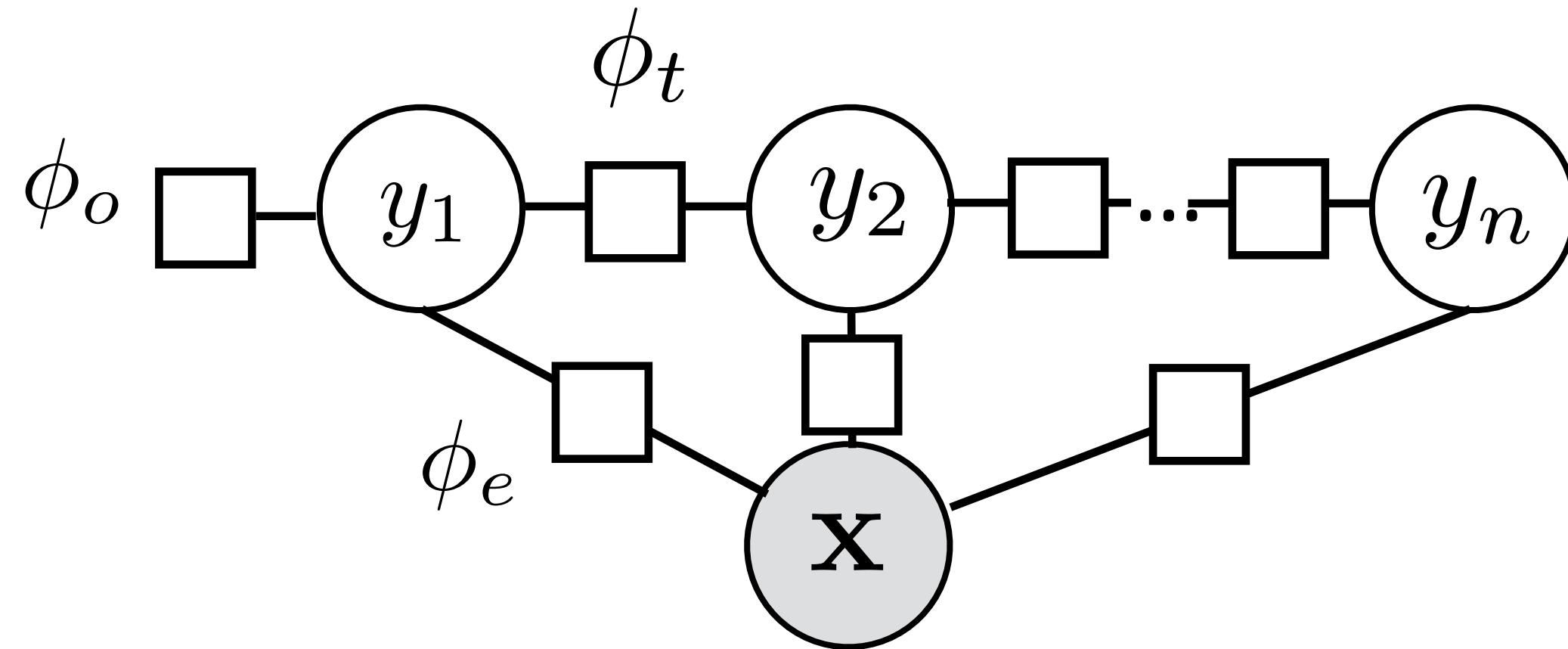
# Sequential CRFs

---



# Sequential CRFs

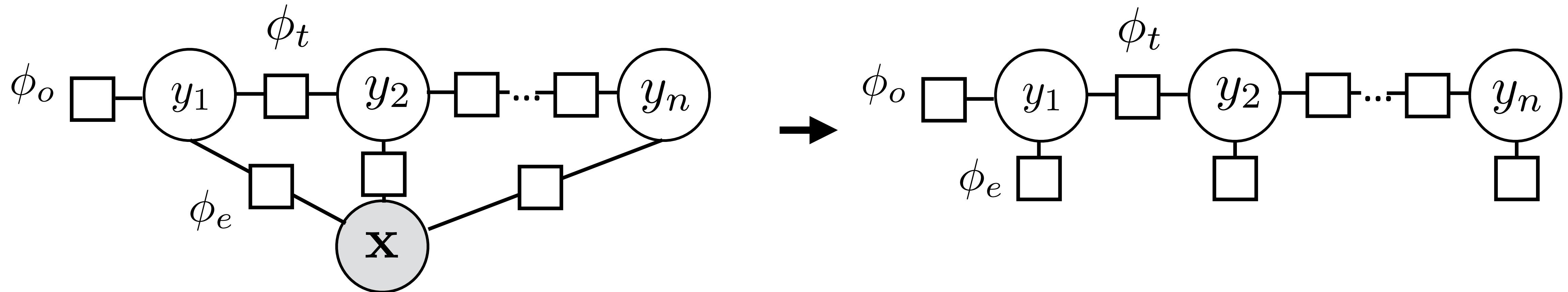
---



- Notation: omit  $x$  from the factor graph entirely (implicit)

# Sequential CRFs

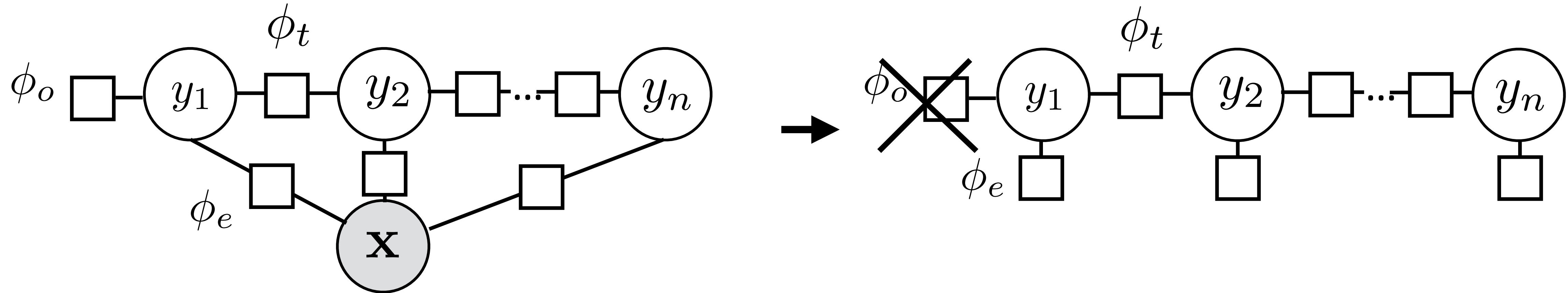
---



- Notation: omit  $x$  from the factor graph entirely (implicit)

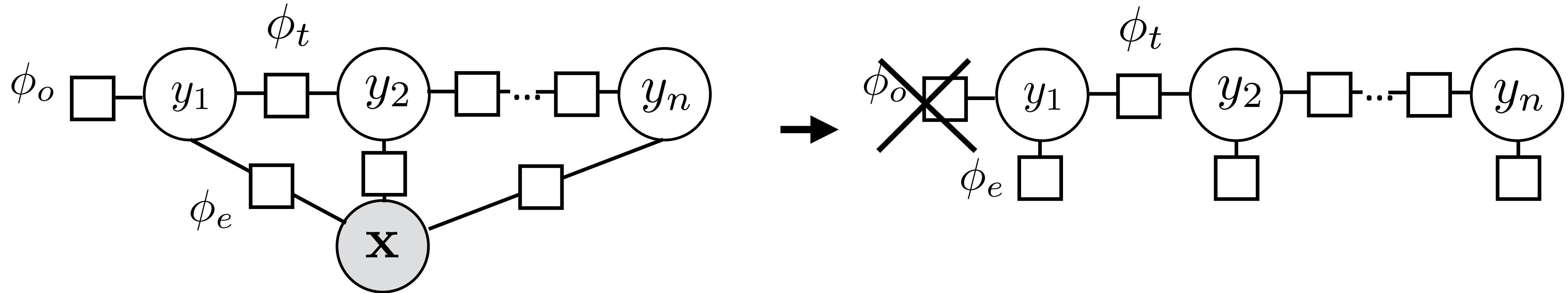
# Sequential CRFs

---



- ▶ Notation: omit  $x$  from the factor graph entirely (implicit)
- ▶ Don't include initial distribution, can bake into other factors

# Sequential CRFs



- Notation: omit  $\mathbf{x}$  from the factor graph entirely (implicit)
- Don't include initial distribution, can bake into other factors

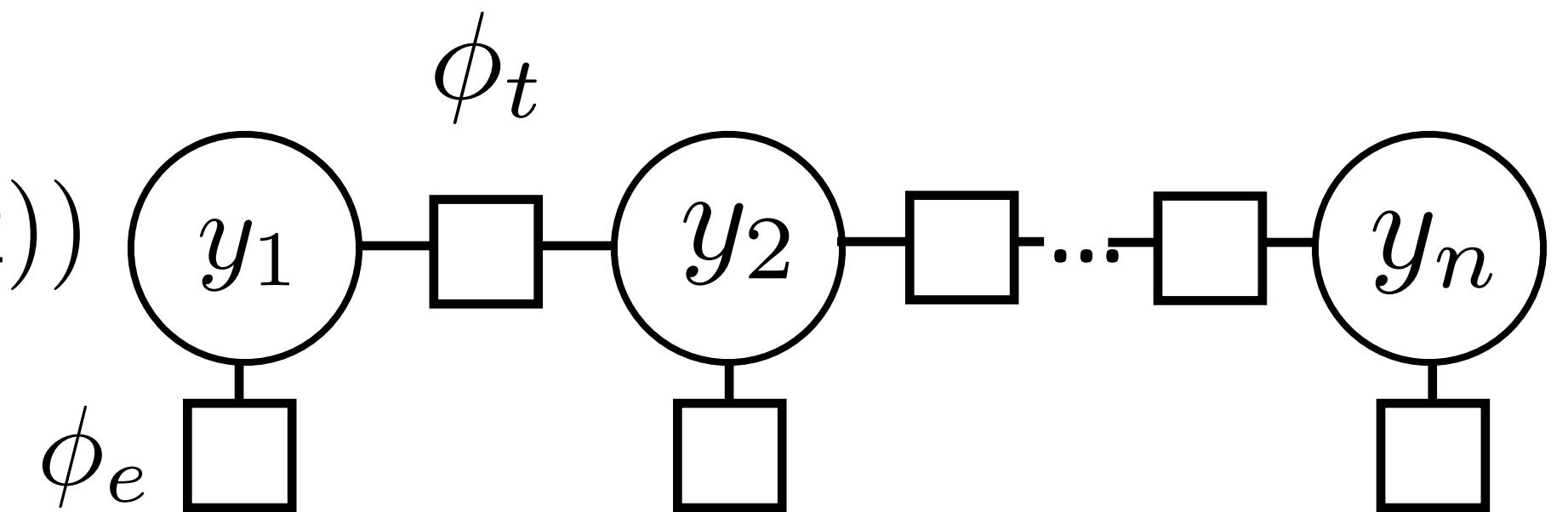
Sequential CRFs:

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(y_i, i, \mathbf{x}))$$

# Feature Functions

---

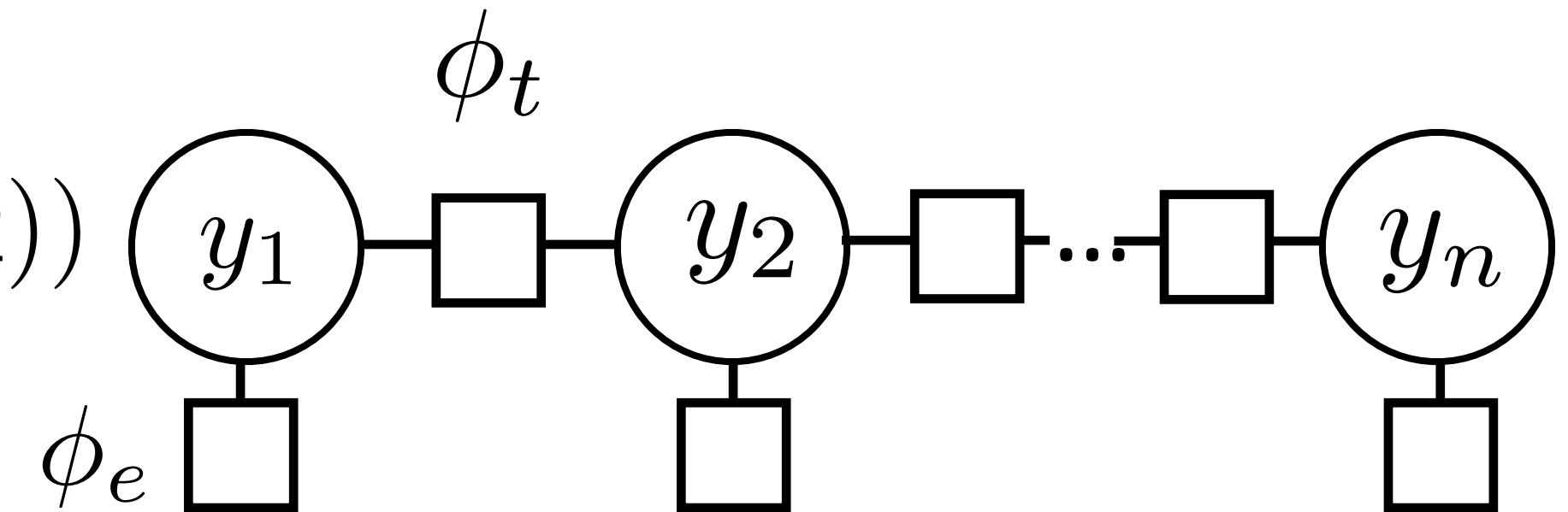
$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(y_i, i, \mathbf{x}))$$



# Feature Functions

---

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(y_i, i, \mathbf{x}))$$

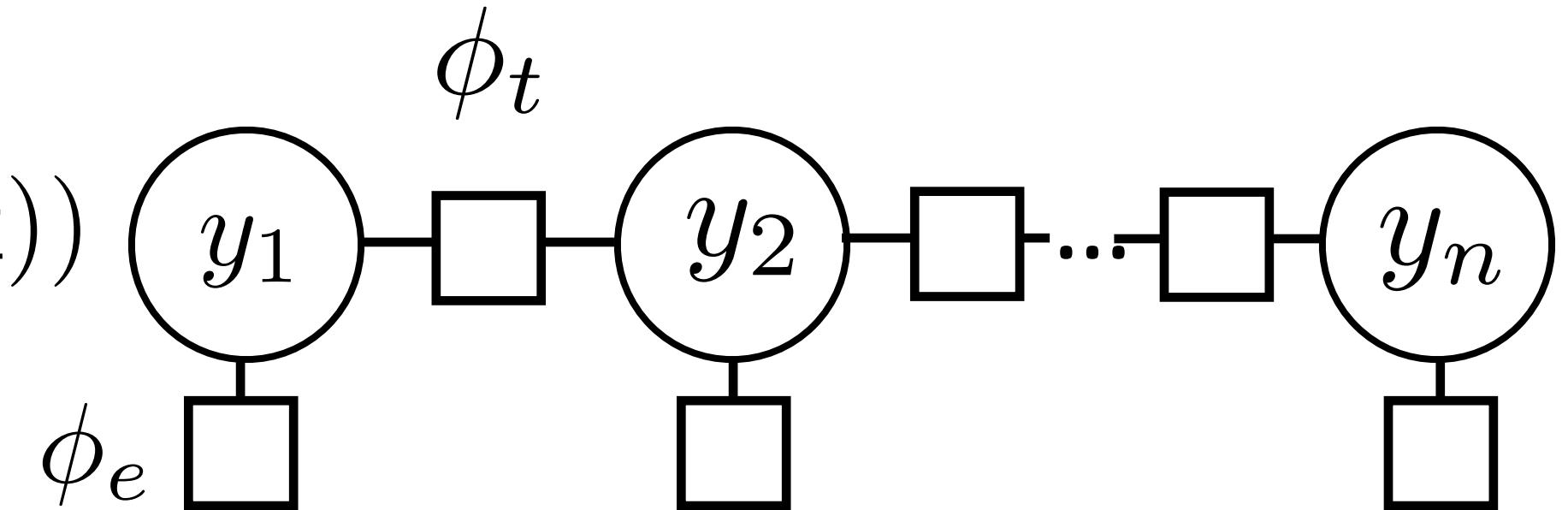


- ▶ This can be almost anything! Here we use linear functions of sparse features

# Feature Functions

---

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(y_i, i, \mathbf{x}))$$



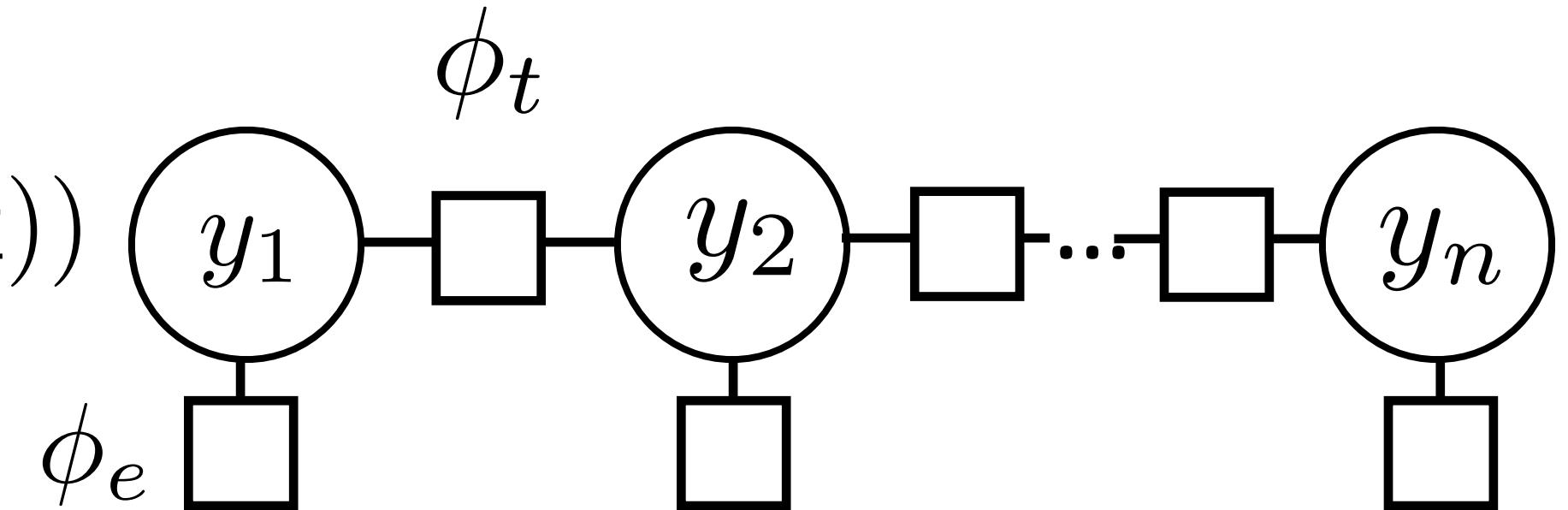
- ▶ This can be almost anything! Here we use linear functions of sparse features

$$\phi_e(y_i, i, \mathbf{x}) = w^\top f_e(y_i, i, \mathbf{x})$$

# Feature Functions

---

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(y_i, i, \mathbf{x}))$$



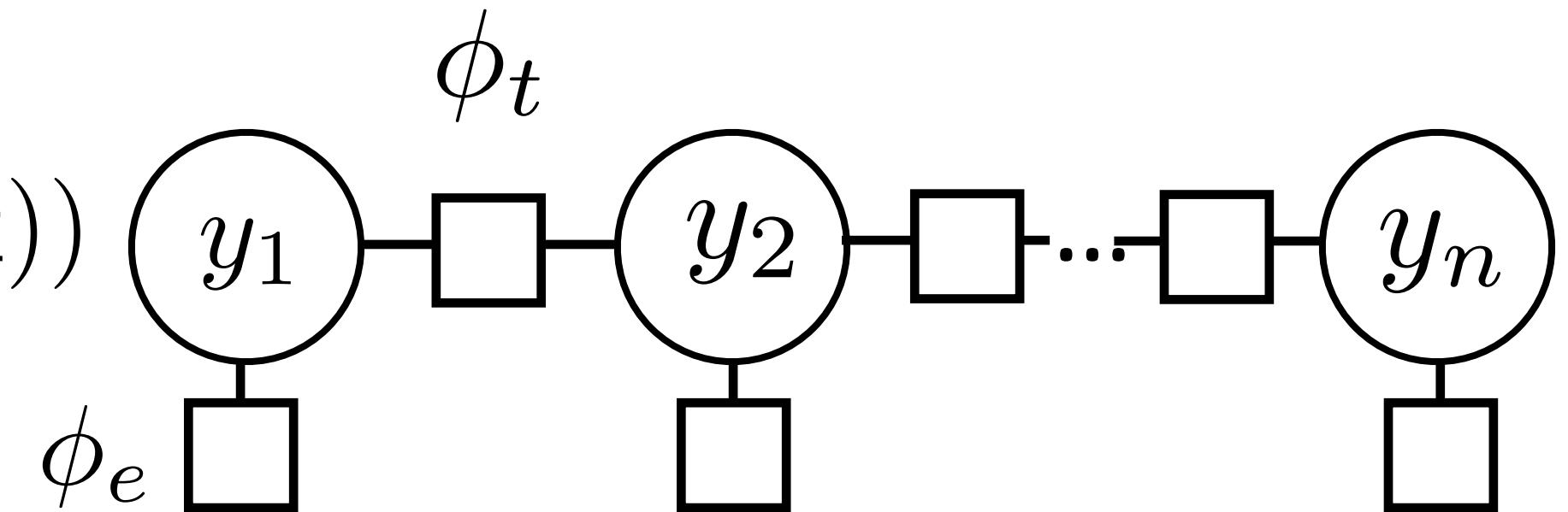
- ▶ This can be almost anything! Here we use linear functions of sparse features

$$\phi_e(y_i, i, \mathbf{x}) = w^\top f_e(y_i, i, \mathbf{x}) \quad \phi_t(y_{i-1}, y_i) = w^\top f_t(y_{i-1}, y_i)$$

# Feature Functions

---

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(y_i, i, \mathbf{x}))$$



- This can be almost anything! Here we use linear functions of sparse features

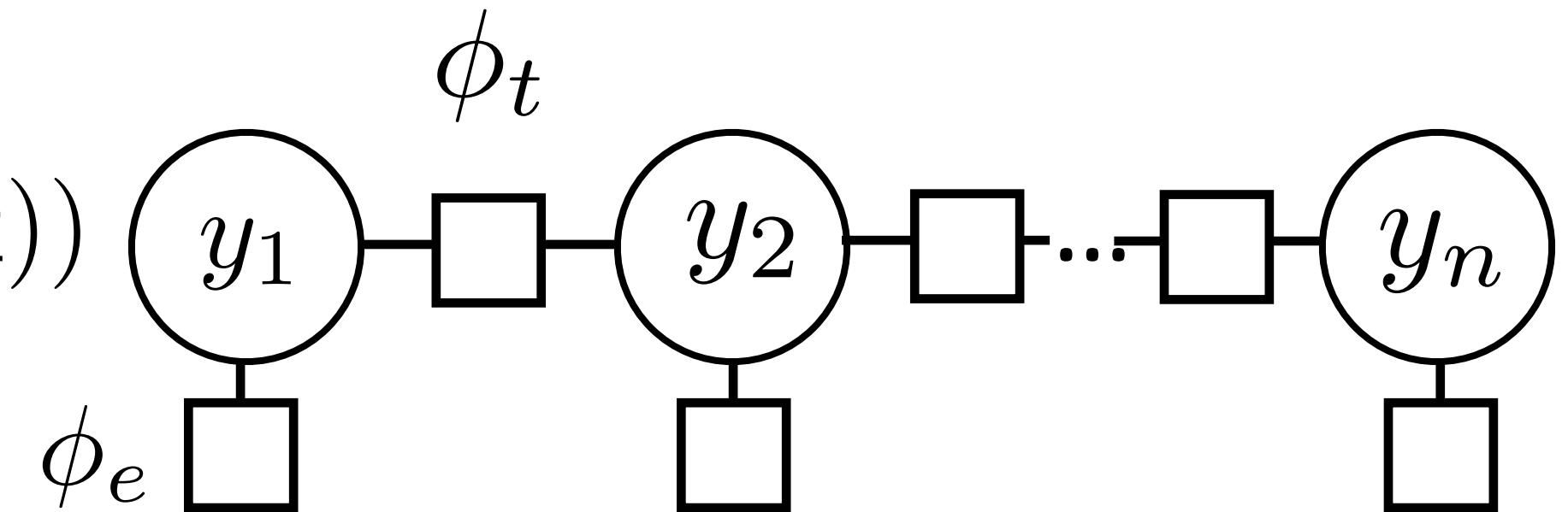
$$\phi_e(y_i, i, \mathbf{x}) = w^\top f_e(y_i, i, \mathbf{x}) \quad \phi_t(y_{i-1}, y_i) = w^\top f_t(y_{i-1}, y_i)$$

$$P(\mathbf{y}|\mathbf{x}) \propto \exp w^\top \left[ \sum_{i=2}^n f_t(y_{i-1}, y_i) + \sum_{i=1}^n f_e(y_i, i, \mathbf{x}) \right]$$

# Feature Functions

---

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(y_i, i, \mathbf{x}))$$



- ▶ This can be almost anything! Here we use linear functions of sparse features

$$\phi_e(y_i, i, \mathbf{x}) = w^\top f_e(y_i, i, \mathbf{x}) \quad \phi_t(y_{i-1}, y_i) = w^\top f_t(y_{i-1}, y_i)$$

$$P(\mathbf{y}|\mathbf{x}) \propto \exp w^\top \left[ \sum_{i=2}^n f_t(y_{i-1}, y_i) + \sum_{i=1}^n f_e(y_i, i, \mathbf{x}) \right]$$

- ▶ Looks like our single weight vector multiclass logistic regression model

# Basic Features for NER

---

$$P(\mathbf{y}|\mathbf{x}) \propto \exp w^\top \left[ \sum_{i=2}^n f_t(y_{i-1}, y_i) + \sum_{i=1}^n f_e(y_i, i, \mathbf{x}) \right]$$

*Barack Obama will travel to Hangzhou today for the G20 meeting .*

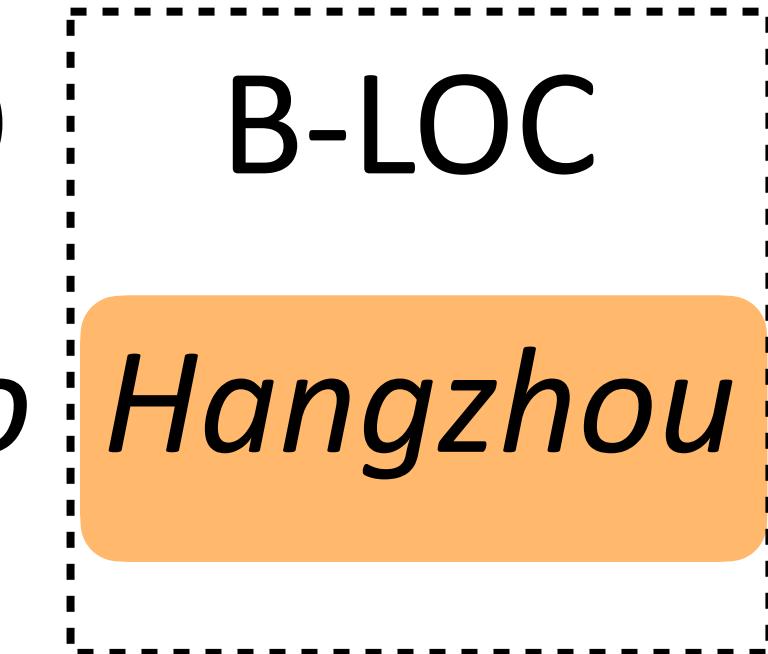
# Basic Features for NER

---

$$P(\mathbf{y}|\mathbf{x}) \propto \exp w^\top \left[ \sum_{i=2}^n f_t(y_{i-1}, y_i) + \sum_{i=1}^n f_e(y_i, i, \mathbf{x}) \right]$$

*Barack Obama will travel to **Hangzhou** today for the G20 meeting .*

O      B-LOC



# Basic Features for NER

---

$$P(\mathbf{y}|\mathbf{x}) \propto \exp w^\top \left[ \sum_{i=2}^n f_t(y_{i-1}, y_i) + \sum_{i=1}^n f_e(y_i, i, \mathbf{x}) \right]$$

O      B-LOC  
*Barack Obama will travel to Hangzhou today for the G20 meeting .*

Transitions:  $f_t(y_{i-1}, y_i) = \text{Ind}[y_{i-1} \& y_i] = \text{Ind}[O - B-LOC]$

# Basic Features for NER

---

$$P(\mathbf{y}|\mathbf{x}) \propto \exp w^\top \left[ \sum_{i=2}^n f_t(y_{i-1}, y_i) + \sum_{i=1}^n f_e(y_i, i, \mathbf{x}) \right]$$

O      B-LOC  
*Barack Obama will travel to **Hangzhou** today for the G20 meeting .*

Transitions:  $f_t(y_{i-1}, y_i) = \text{Ind}[y_{i-1} \& y_i] = \text{Ind}[O - B-LOC]$

Emissions:  $f_e(y_6, 6, \mathbf{x}) =$

# Basic Features for NER

---

$$P(\mathbf{y}|\mathbf{x}) \propto \exp w^\top \left[ \sum_{i=2}^n f_t(y_{i-1}, y_i) + \sum_{i=1}^n f_e(y_i, i, \mathbf{x}) \right]$$

O      B-LOC  
*Barack Obama will travel to Hangzhou today for the G20 meeting .*

Transitions:  $f_t(y_{i-1}, y_i) = \text{Ind}[y_{i-1} \& y_i] = \text{Ind}[O - B-LOC]$

Emissions:  $f_e(y_6, 6, \mathbf{x}) = \text{Ind}[B-LOC \& \text{Current word} = Hangzhou]$

# Basic Features for NER

---

$$P(\mathbf{y}|\mathbf{x}) \propto \exp w^\top \left[ \sum_{i=2}^n f_t(y_{i-1}, y_i) + \sum_{i=1}^n f_e(y_i, i, \mathbf{x}) \right]$$

O      B-LOC  
*Barack Obama will travel to **Hangzhou** today for the G20 meeting .*

Transitions:  $f_t(y_{i-1}, y_i) = \text{Ind}[y_{i-1} \& y_i] = \text{Ind}[O - B-LOC]$

Emissions:  $f_e(y_6, 6, \mathbf{x}) = \text{Ind}[B-LOC \& \text{Current word} = Hangzhou]$   
 $\text{Ind}[B-LOC \& \text{Prev word} = to]$

# Features for NER

---

LOC

*Leicestershire is a nice place to visit...*

$\phi_e(y_i, i, \mathbf{x})$

PER

*Leonardo DiCaprio won an award...*

LOC

*I took a vacation to Boston*

ORG

*Apple released a new version...*

LOC

*Texas governor Greg Abbott said*

PER

*According to the New York Times...*

ORG

# Features for NER

---

- ▶ Word features (can use in HMM)

- ▶ Capitalization

- ▶ Word shape

- ▶ Prefixes/suffixes

- ▶ Lexical indicators

- ▶ Context features (can't use in HMM!)

- ▶ Words before/after

- ▶ Tags before/after

- ▶ Word clusters

- ▶ Gazetteers

*Leicestershire*

*Boston*

*Apple released a new version...*

*According to the New York Times...*

# CRFs Outline

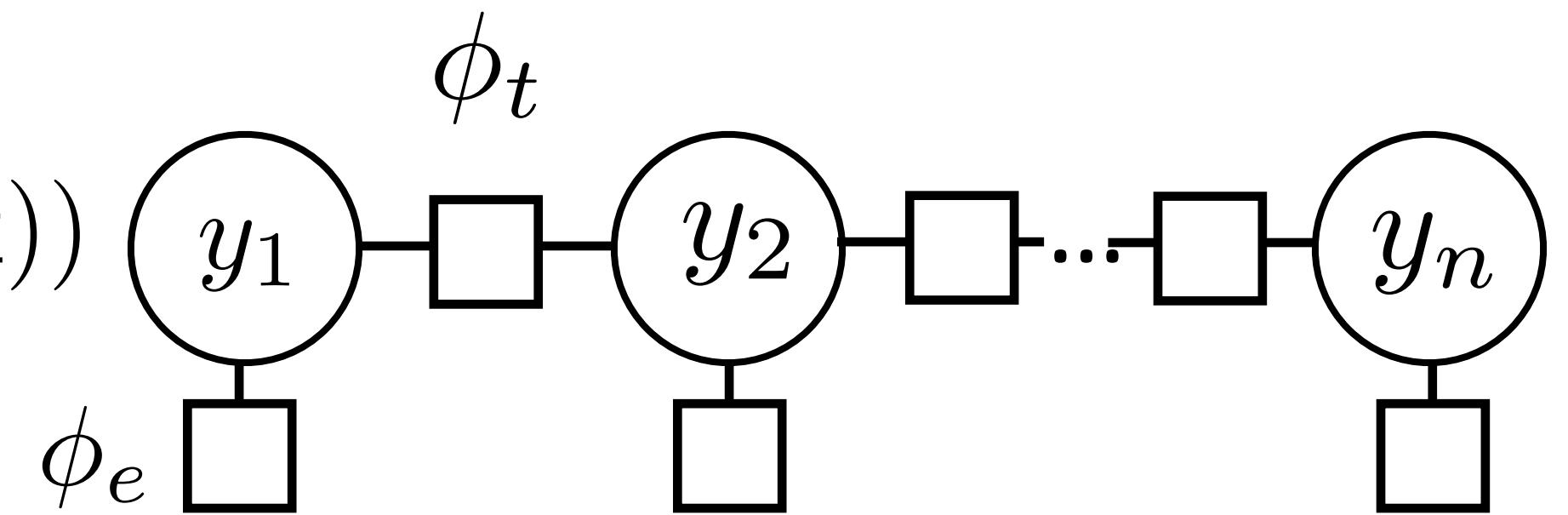
---

- Model:  $P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(y_i, i, \mathbf{x}))$   
$$P(\mathbf{y}|\mathbf{x}) \propto \exp w^\top \left[ \sum_{i=2}^n f_t(y_{i-1}, y_i) + \sum_{i=1}^n f_e(y_i, i, \mathbf{x}) \right]$$
- Inference
- Learning

# Computing (arg)maxes

---

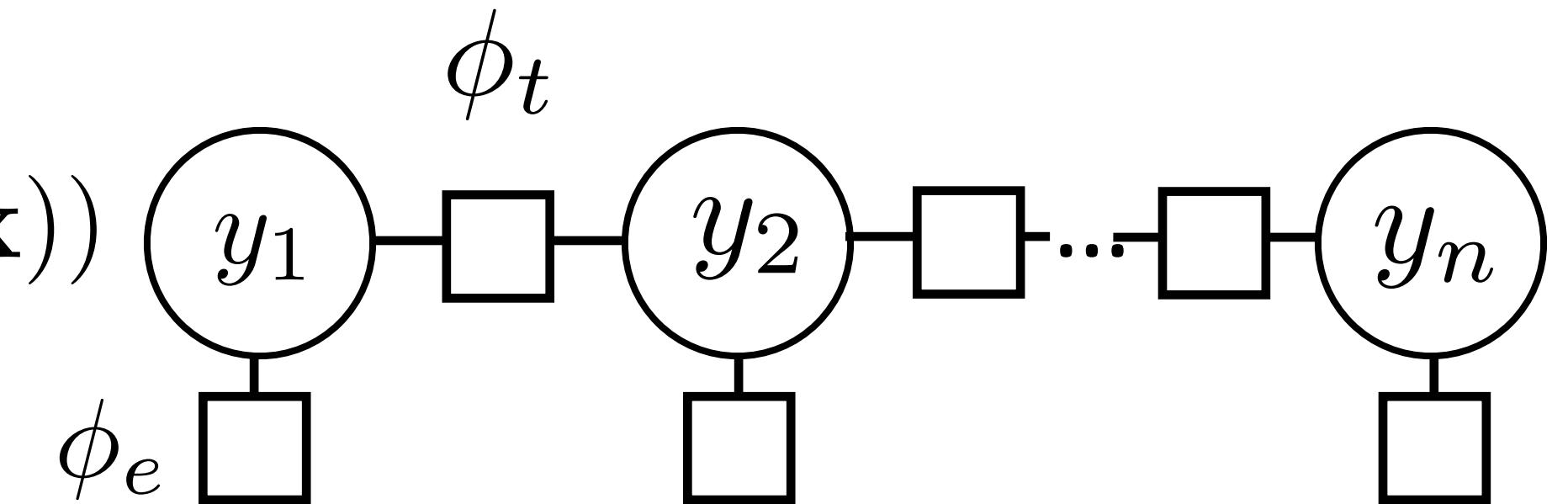
$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(y_i, i, \mathbf{x}))$$



# Computing (arg)maxes

---

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(y_i, i, \mathbf{x}))$$

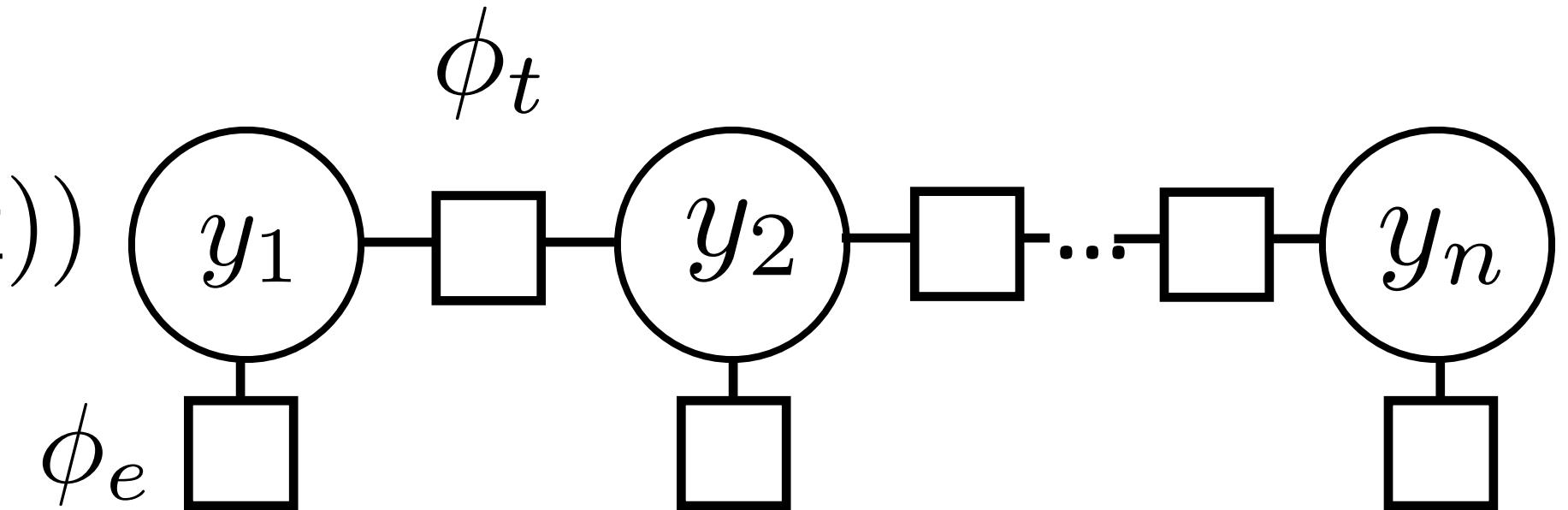


- $\text{argmax}_{\mathbf{y}} P(\mathbf{y}|\mathbf{x})$ : can use Viterbi exactly as in HMM case

# Computing (arg)maxes

---

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(y_i, i, \mathbf{x}))$$



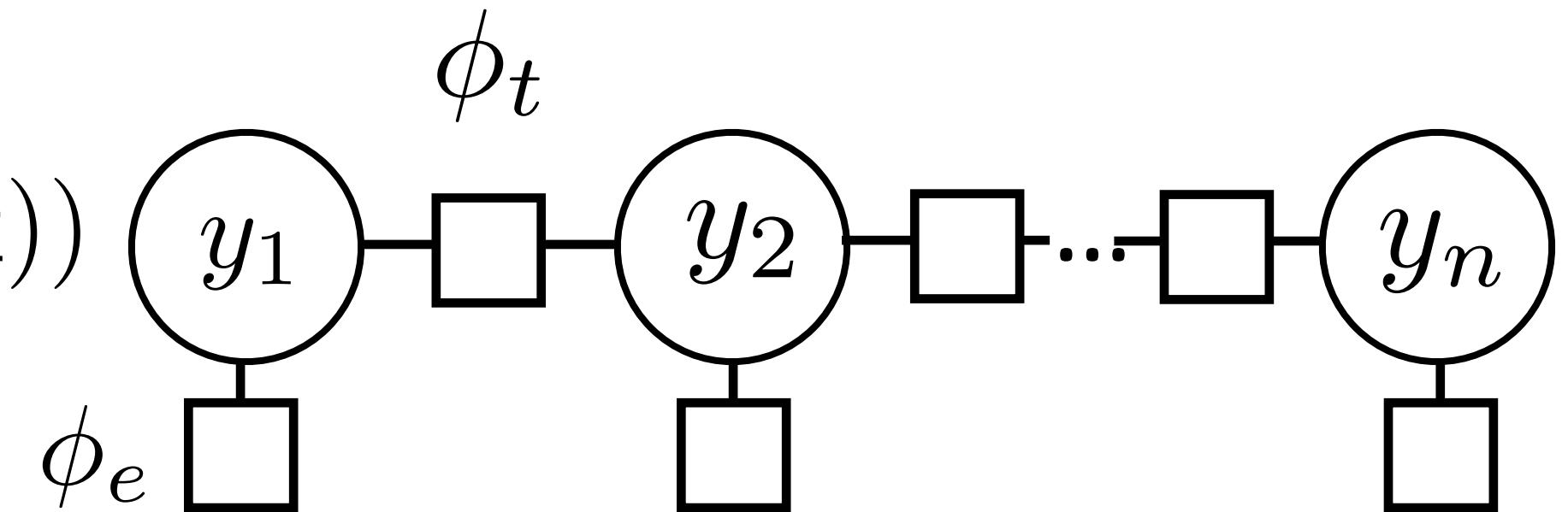
- $\text{argmax}_{\mathbf{y}} P(\mathbf{y}|\mathbf{x})$ : can use Viterbi exactly as in HMM case

$$\max_{y_1, \dots, y_n} e^{\phi_t(y_{n-1}, y_n)} e^{\phi_e(y_n, n, \mathbf{x})} \dots e^{\phi_e(y_2, 2, \mathbf{x})} e^{\phi_t(y_1, y_2)} e^{\phi_e(y_1, 1, \mathbf{x})}$$

# Computing (arg)maxes

---

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(y_i, i, \mathbf{x}))$$



- $\text{argmax}_{\mathbf{y}} P(\mathbf{y}|\mathbf{x})$ : can use Viterbi exactly as in HMM case

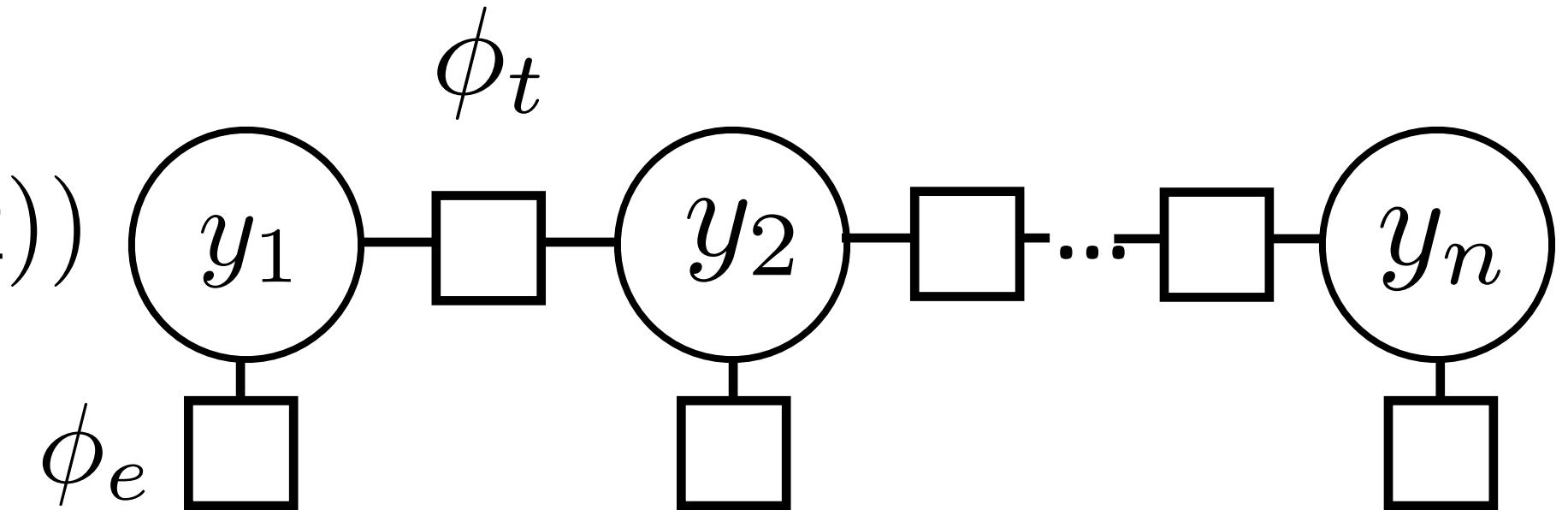
$$\max_{y_1, \dots, y_n} e^{\phi_t(y_{n-1}, y_n)} e^{\phi_e(y_n, n, \mathbf{x})} \dots e^{\phi_e(y_2, 2, \mathbf{x})} e^{\phi_t(y_1, y_2)} e^{\phi_e(y_1, 1, \mathbf{x})}$$

$$= \max_{y_2, \dots, y_n} e^{\phi_t(y_{n-1}, y_n)} e^{\phi_e(y_n, n, \mathbf{x})} \dots e^{\phi_e(y_2, 2, \mathbf{x})} \max_{y_1} e^{\phi_t(y_1, y_2)} e^{\phi_e(y_1, 1, \mathbf{x})}$$

# Computing (arg)maxes

---

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(y_i, i, \mathbf{x}))$$



- $\text{argmax}_{\mathbf{y}} P(\mathbf{y}|\mathbf{x})$ : can use Viterbi exactly as in HMM case

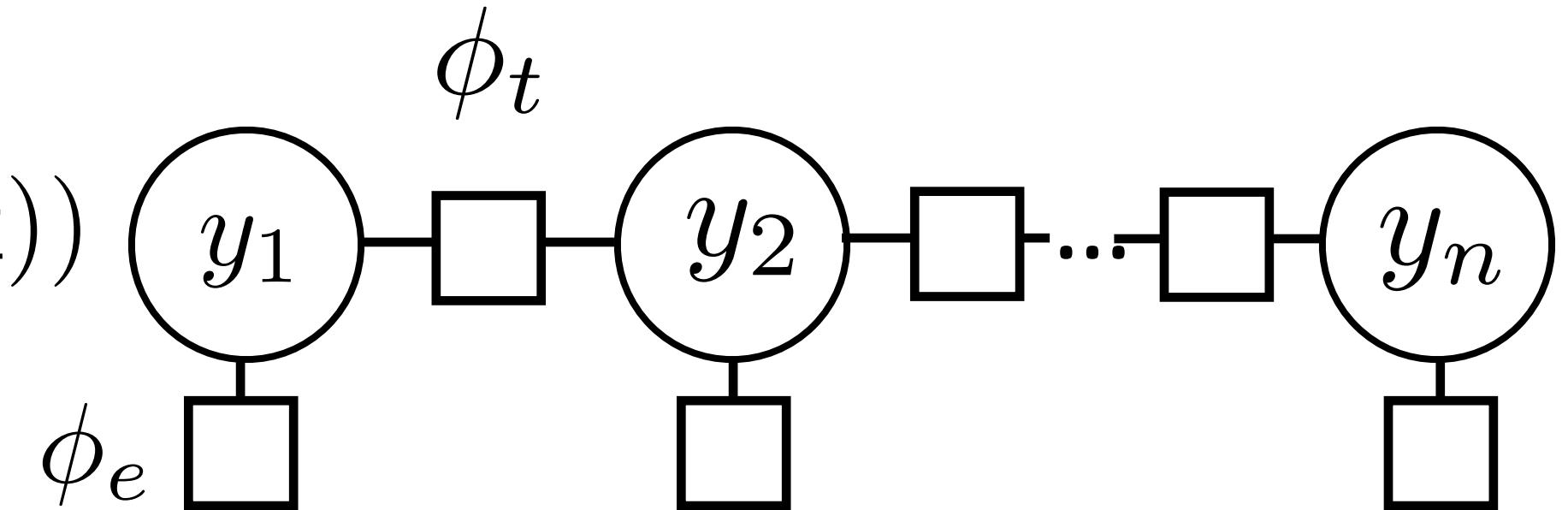
$$\max_{y_1, \dots, y_n} e^{\phi_t(y_{n-1}, y_n)} e^{\phi_e(y_n, n, \mathbf{x})} \dots e^{\phi_e(y_2, 2, \mathbf{x})} e^{\phi_t(y_1, y_2)} e^{\phi_e(y_1, 1, \mathbf{x})}$$

$$= \max_{y_2, \dots, y_n} e^{\phi_t(y_{n-1}, y_n)} e^{\phi_e(y_n, n, \mathbf{x})} \dots e^{\phi_e(y_2, 2, \mathbf{x})} \boxed{\max_{y_1} e^{\phi_t(y_1, y_2)} e^{\phi_e(y_1, 1, \mathbf{x})}}$$

# Computing (arg)maxes

---

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(y_i, i, \mathbf{x}))$$



- $\text{argmax}_{\mathbf{y}} P(\mathbf{y}|\mathbf{x})$ : can use Viterbi exactly as in HMM case

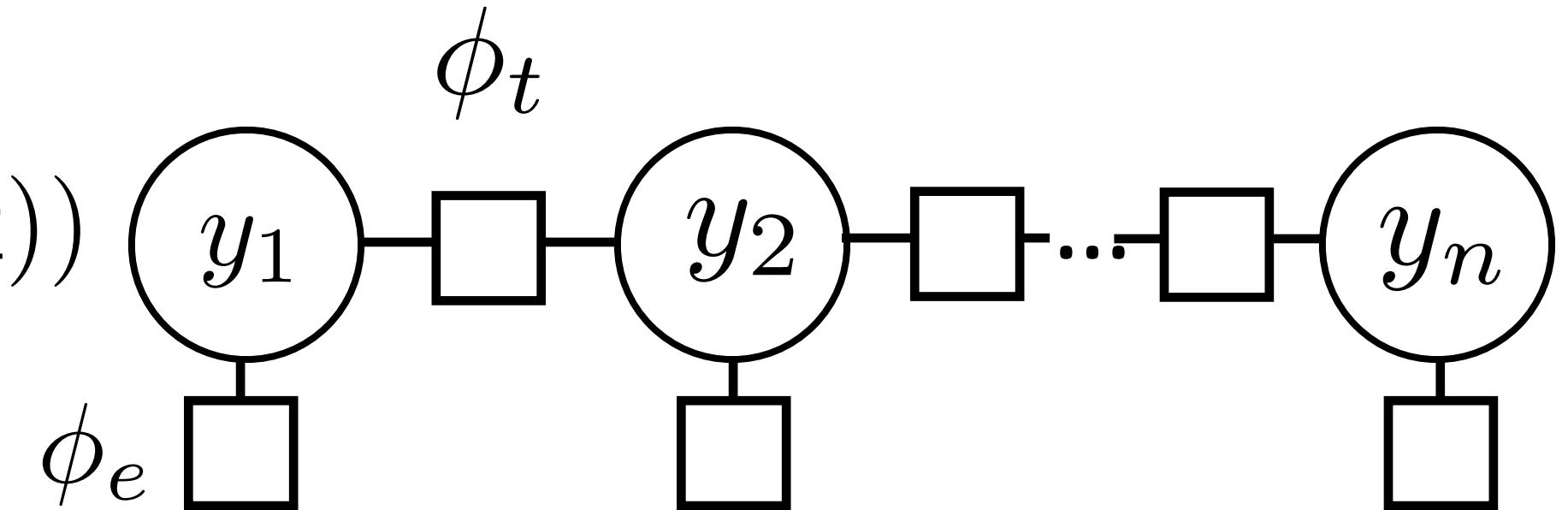
$$\max_{y_1, \dots, y_n} e^{\phi_t(y_{n-1}, y_n)} e^{\phi_e(y_n, n, \mathbf{x})} \dots e^{\phi_e(y_2, 2, \mathbf{x})} e^{\phi_t(y_1, y_2)} e^{\phi_e(y_1, 1, \mathbf{x})}$$

$$= \max_{y_2, \dots, y_n} e^{\phi_t(y_{n-1}, y_n)} e^{\phi_e(y_n, n, \mathbf{x})} \dots e^{\phi_e(y_2, 2, \mathbf{x})} \boxed{\max_{y_1} e^{\phi_t(y_1, y_2)} \underbrace{e^{\phi_e(y_1, 1, \mathbf{x})}}_{}}$$

# Computing (arg)maxes

---

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(y_i, i, \mathbf{x}))$$



- $\text{argmax}_{\mathbf{y}} P(\mathbf{y}|\mathbf{x})$ : can use Viterbi exactly as in HMM case

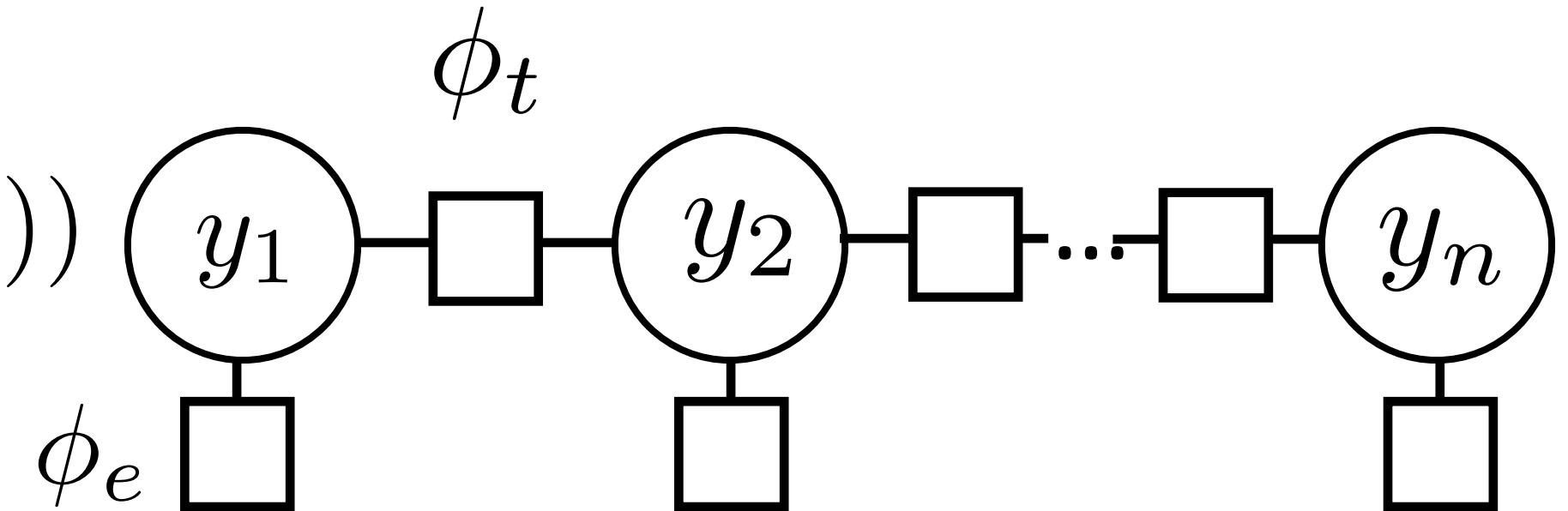
$$\max_{y_1, \dots, y_n} e^{\phi_t(y_{n-1}, y_n)} e^{\phi_e(y_n, n, \mathbf{x})} \dots e^{\phi_e(y_2, 2, \mathbf{x})} e^{\phi_t(y_1, y_2)} e^{\phi_e(y_1, 1, \mathbf{x})}$$

$$= \max_{y_2, \dots, y_n} e^{\phi_t(y_{n-1}, y_n)} e^{\phi_e(y_n, n, \mathbf{x})} \dots e^{\phi_e(y_2, 2, \mathbf{x})} \boxed{\max_{y_1} e^{\phi_t(y_1, y_2)} e^{\phi_e(y_1, 1, \mathbf{x})}}$$

$$= \max_{y_3, \dots, y_n} e^{\phi_t(y_{n-1}, y_n)} e^{\phi_e(y_n, n, \mathbf{x})} \dots \max_{y_2} e^{\phi_t(y_2, y_3)} e^{\phi_e(y_2, 2, \mathbf{x})} \max_{y_1} e^{\phi_t(y_1, y_2)} \underbrace{\text{score}_1(y_1)}$$

# Computing (arg)maxes

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(y_i, i, \mathbf{x}))$$



- $\text{argmax}_{\mathbf{y}} P(\mathbf{y}|\mathbf{x})$ : can use Viterbi exactly as in HMM case

$$\max_{y_1, \dots, y_n} e^{\phi_t(y_{n-1}, y_n)} e^{\phi_e(y_n, n, \mathbf{x})} \dots e^{\phi_e(y_2, 2, \mathbf{x})} e^{\phi_t(y_1, y_2)} e^{\phi_e(y_1, 1, \mathbf{x})}$$

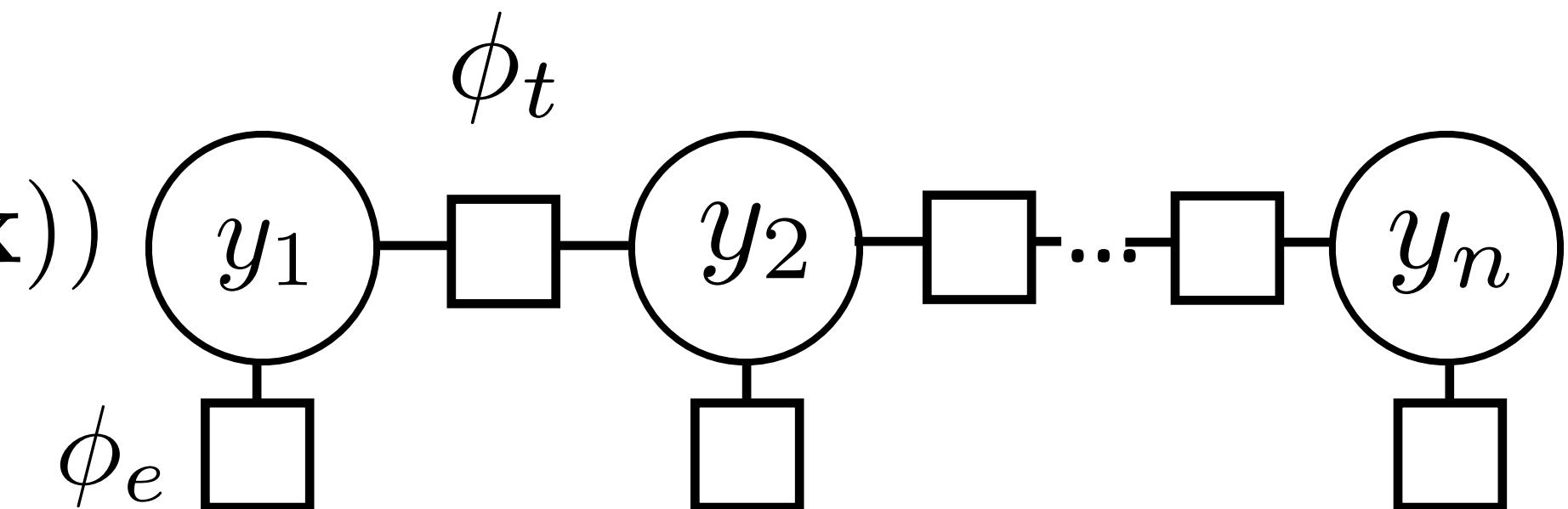
$$= \max_{y_2, \dots, y_n} e^{\phi_t(y_{n-1}, y_n)} e^{\phi_e(y_n, n, \mathbf{x})} \dots e^{\phi_e(y_2, 2, \mathbf{x})} \boxed{\max_{y_1} e^{\phi_t(y_1, y_2)} e^{\phi_e(y_1, 1, \mathbf{x})}}$$

$$= \max_{y_3, \dots, y_n} e^{\phi_t(y_{n-1}, y_n)} e^{\phi_e(y_n, n, \mathbf{x})} \dots \underbrace{\max_{y_2} e^{\phi_t(y_2, y_3)} e^{\phi_e(y_2, 2, \mathbf{x})}}_{\underbrace{\max_{y_1} e^{\phi_t(y_1, y_2)} \text{score}_1(y_1)}}$$

# Computing (arg)maxes

---

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(y_i, i, \mathbf{x}))$$



- $\text{argmax}_{\mathbf{y}} P(\mathbf{y}|\mathbf{x})$ : can use Viterbi exactly as in HMM case

$$\max_{y_1, \dots, y_n} e^{\phi_t(y_{n-1}, y_n)} e^{\phi_e(y_n, n, \mathbf{x})} \dots e^{\phi_e(y_2, 2, \mathbf{x})} e^{\phi_t(y_1, y_2)} e^{\phi_e(y_1, 1, \mathbf{x})}$$

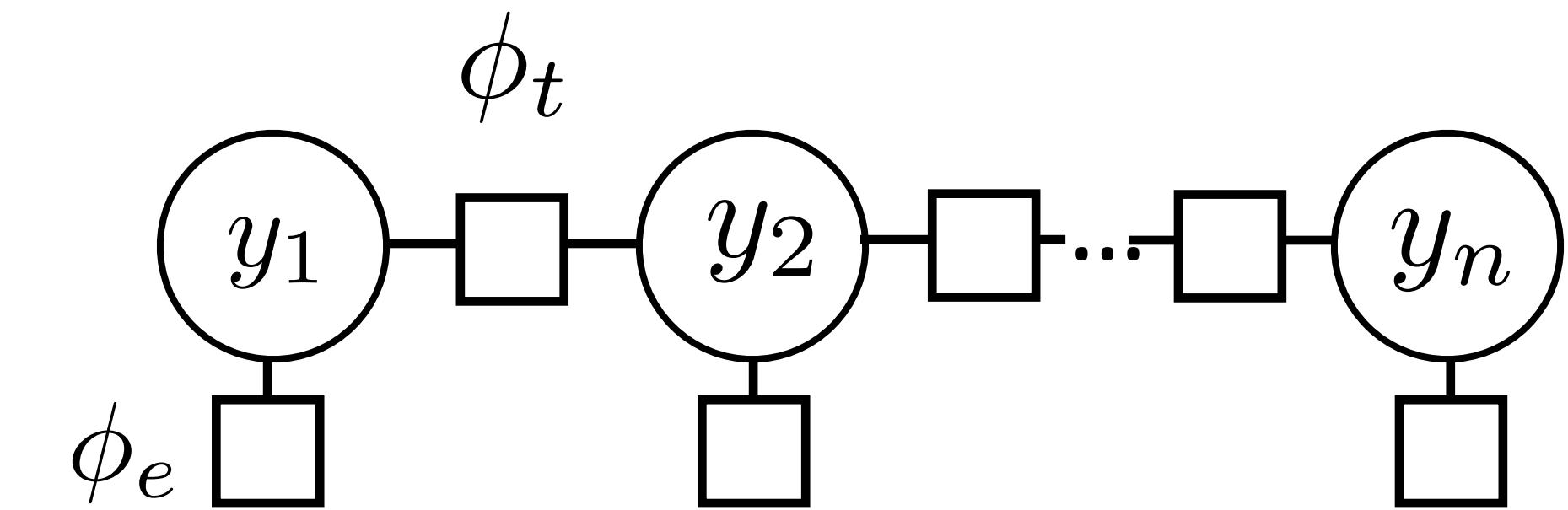
$$= \max_{y_2, \dots, y_n} e^{\phi_t(y_{n-1}, y_n)} e^{\phi_e(y_n, n, \mathbf{x})} \dots e^{\phi_e(y_2, 2, \mathbf{x})} \boxed{\max_{y_1} e^{\phi_t(y_1, y_2)} \underbrace{e^{\phi_e(y_1, 1, \mathbf{x})}}_{\text{score}_1(y_1)}}$$

$$= \max_{y_3, \dots, y_n} e^{\phi_t(y_{n-1}, y_n)} e^{\phi_e(y_n, n, \mathbf{x})} \dots \max_{y_2} e^{\phi_t(y_2, y_3)} e^{\phi_e(y_2, 2, \mathbf{x})} \underbrace{\max_{y_1} e^{\phi_t(y_1, y_2)} \text{score}_1(y_1)}$$

- $\exp(\phi_t(y_{i-1}, y_i))$  and  $\exp(\phi_e(y_i, i, \mathbf{x}))$  play the role of the Ps now, same dynamic program

# Inference in General CRFs

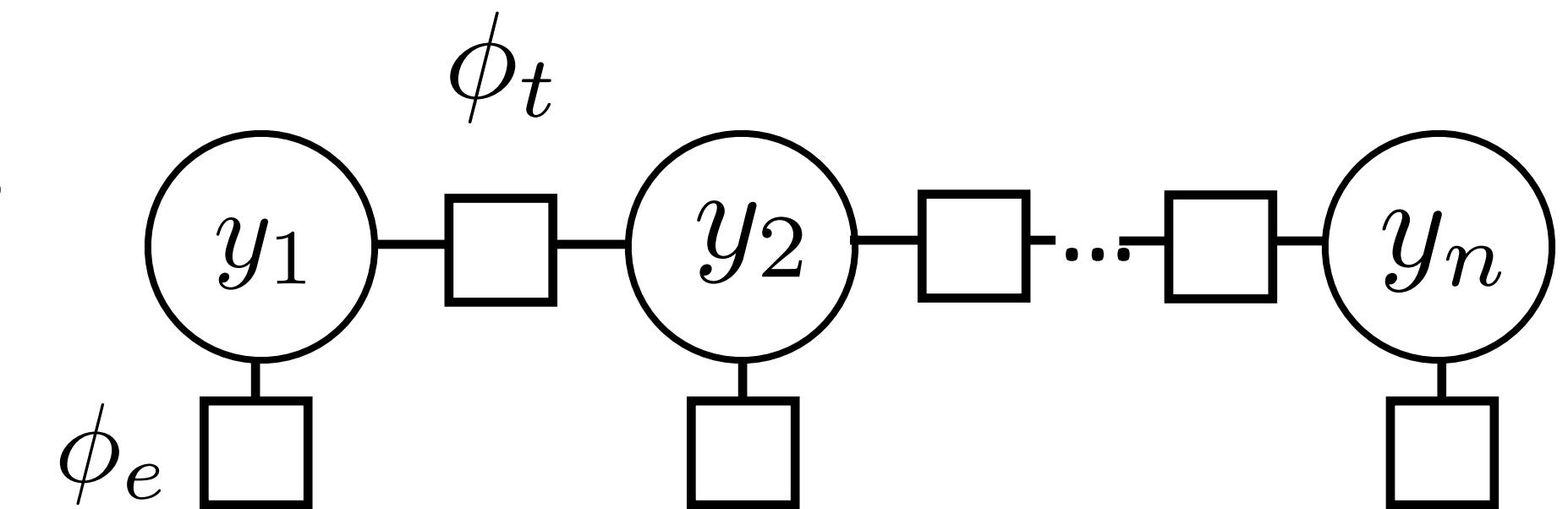
---



# Inference in General CRFs

---

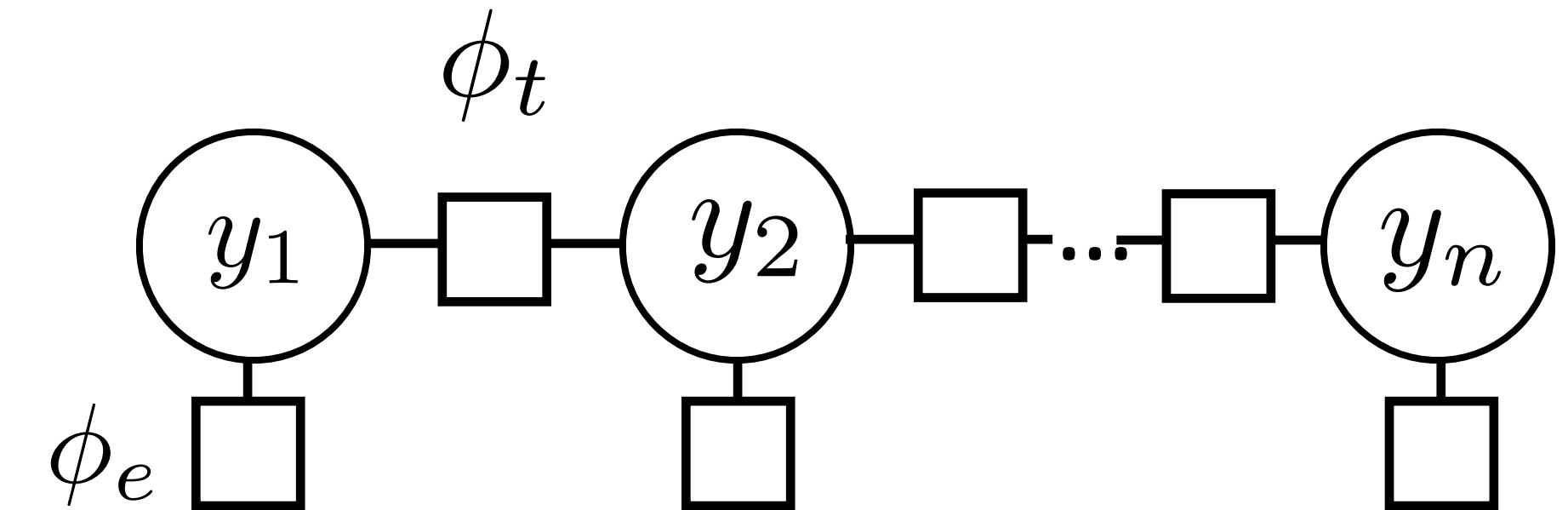
- ▶ Can do inference in any tree-structured CRF



# Inference in General CRFs

---

- ▶ Can do inference in any tree-structured CRF



- ▶ Max-product algorithm: generalization of Viterbi to arbitrary tree-structured graphs (sum-product is generalization of forward-backward)

# CRFs Outline

---

- Model:  $P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(y_i, i, \mathbf{x}))$   
$$P(\mathbf{y}|\mathbf{x}) \propto \exp w^\top \left[ \sum_{i=2}^n f_t(y_{i-1}, y_i) + \sum_{i=1}^n f_e(y_i, i, \mathbf{x}) \right]$$
- Inference: argmax  $P(\mathbf{y}|\mathbf{x})$  from Viterbi
- Learning

# Training CRFs

---

$$P(\mathbf{y}|\mathbf{x}) \propto \exp w^\top \left[ \sum_{i=2}^n f_t(y_{i-1}, y_i) + \sum_{i=1}^n f_e(y_i, i, \mathbf{x}) \right]$$

# Training CRFs

---

$$P(\mathbf{y}|\mathbf{x}) \propto \exp w^\top \left[ \sum_{i=2}^n f_t(y_{i-1}, y_i) + \sum_{i=1}^n f_e(y_i, i, \mathbf{x}) \right]$$

- Logistic regression:  $P(y|x) \propto \exp w^\top f(x, y)$

# Training CRFs

---

$$P(\mathbf{y}|\mathbf{x}) \propto \exp w^\top \left[ \sum_{i=2}^n f_t(y_{i-1}, y_i) + \sum_{i=1}^n f_e(y_i, i, \mathbf{x}) \right]$$

- ▶ Logistic regression:  $P(y|x) \propto \exp w^\top f(x, y)$
- ▶ Maximize  $\mathcal{L}(\mathbf{y}^*, \mathbf{x}) = \log P(\mathbf{y}^*|\mathbf{x})$

# Training CRFs

---

$$P(\mathbf{y}|\mathbf{x}) \propto \exp w^\top \left[ \sum_{i=2}^n f_t(y_{i-1}, y_i) + \sum_{i=1}^n f_e(y_i, i, \mathbf{x}) \right]$$

- ▶ Logistic regression:  $P(y|x) \propto \exp w^\top f(x, y)$
- ▶ Maximize  $\mathcal{L}(\mathbf{y}^*, \mathbf{x}) = \log P(\mathbf{y}^*|\mathbf{x})$
- ▶ Gradient is completely analogous to logistic regression:

# Training CRFs

---

$$P(\mathbf{y}|\mathbf{x}) \propto \exp w^\top \left[ \sum_{i=2}^n f_t(y_{i-1}, y_i) + \sum_{i=1}^n f_e(y_i, i, \mathbf{x}) \right]$$

- ▶ Logistic regression:  $P(y|x) \propto \exp w^\top f(x, y)$
- ▶ Maximize  $\mathcal{L}(\mathbf{y}^*, \mathbf{x}) = \log P(\mathbf{y}^*|\mathbf{x})$
- ▶ Gradient is completely analogous to logistic regression:

$$\begin{aligned} \frac{\partial}{\partial w} \mathcal{L}(\mathbf{y}^*, \mathbf{x}) &= \sum_{i=2}^n f_t(y_{i-1}^*, y_i^*) + \sum_{i=1}^n f_e(y_i^*, i, \mathbf{x}) \\ &\quad - \mathbb{E}_{\mathbf{y}} \left[ \sum_{i=2}^n f_t(y_{i-1}, y_i) + \sum_{i=1}^n f_e(y_i, i, \mathbf{x}) \right] \end{aligned}$$

# Training CRFs

---

$$P(\mathbf{y}|\mathbf{x}) \propto \exp w^\top \left[ \sum_{i=2}^n f_t(y_{i-1}, y_i) + \sum_{i=1}^n f_e(y_i, i, \mathbf{x}) \right]$$

- ▶ Logistic regression:  $P(y|x) \propto \exp w^\top f(x, y)$
- ▶ Maximize  $\mathcal{L}(\mathbf{y}^*, \mathbf{x}) = \log P(\mathbf{y}^*|\mathbf{x})$
- ▶ Gradient is completely analogous to logistic regression:

$$\frac{\partial}{\partial w} \mathcal{L}(\mathbf{y}^*, \mathbf{x}) = \sum_{i=2}^n f_t(y_{i-1}^*, y_i^*) + \sum_{i=1}^n f_e(y_i^*, i, \mathbf{x})$$

intractable!  $\xrightarrow{-\mathbb{E}_{\mathbf{y}} \left[ \sum_{i=2}^n f_t(y_{i-1}, y_i) + \sum_{i=1}^n f_e(y_i, i, \mathbf{x}) \right]}$

# Training CRFs

---

$$\begin{aligned} \frac{\partial}{\partial w} \mathcal{L}(\mathbf{y}^*, \mathbf{x}) &= \sum_{i=2}^n f_t(y_{i-1}^*, y_i^*) + \sum_{i=1}^n f_e(y_i^*, i, \mathbf{x}) \\ &\quad - \mathbb{E}_{\mathbf{y}} \left[ \sum_{i=2}^n f_t(y_{i-1}, y_i) + \sum_{i=1}^n f_e(y_i, i, \mathbf{x}) \right] \end{aligned}$$

# Training CRFs

---

$$\begin{aligned}\frac{\partial}{\partial w} \mathcal{L}(\mathbf{y}^*, \mathbf{x}) &= \sum_{i=2}^n f_t(y_{i-1}^*, y_i^*) + \sum_{i=1}^n f_e(y_i^*, i, \mathbf{x}) \\ &\quad - \mathbb{E}_{\mathbf{y}} \left[ \sum_{i=2}^n f_t(y_{i-1}, y_i) + \sum_{i=1}^n f_e(y_i, i, \mathbf{x}) \right]\end{aligned}$$

- ▶ Let's focus on emission feature expectation

# Training CRFs

---

$$\begin{aligned}\frac{\partial}{\partial w} \mathcal{L}(\mathbf{y}^*, \mathbf{x}) &= \sum_{i=2}^n f_t(y_{i-1}^*, y_i^*) + \sum_{i=1}^n f_e(y_i^*, i, \mathbf{x}) \\ &\quad - \mathbb{E}_{\mathbf{y}} \left[ \sum_{i=2}^n f_t(y_{i-1}, y_i) + \sum_{i=1}^n f_e(y_i, i, \mathbf{x}) \right]\end{aligned}$$

- ▶ Let's focus on emission feature expectation

$$\mathbb{E}_{\mathbf{y}} \left[ \sum_{i=1}^n f_e(y_i, i, \mathbf{x}) \right]$$

# Training CRFs

---

$$\begin{aligned} \frac{\partial}{\partial w} \mathcal{L}(\mathbf{y}^*, \mathbf{x}) &= \sum_{i=2}^n f_t(y_{i-1}^*, y_i^*) + \sum_{i=1}^n f_e(y_i^*, i, \mathbf{x}) \\ &\quad - \mathbb{E}_{\mathbf{y}} \left[ \sum_{i=2}^n f_t(y_{i-1}, y_i) + \sum_{i=1}^n f_e(y_i, i, \mathbf{x}) \right] \end{aligned}$$

- ▶ Let's focus on emission feature expectation

$$\mathbb{E}_{\mathbf{y}} \left[ \sum_{i=1}^n f_e(y_i, i, \mathbf{x}) \right] = \sum_{\mathbf{y} \in \mathcal{Y}} P(\mathbf{y} | \mathbf{x}) \left[ \sum_{i=1}^n f_e(y_i, i, \mathbf{x}) \right]$$

# Training CRFs

---

$$\begin{aligned}\frac{\partial}{\partial w} \mathcal{L}(\mathbf{y}^*, \mathbf{x}) &= \sum_{i=2}^n f_t(y_{i-1}^*, y_i^*) + \sum_{i=1}^n f_e(y_i^*, i, \mathbf{x}) \\ &\quad - \mathbb{E}_{\mathbf{y}} \left[ \sum_{i=2}^n f_t(y_{i-1}, y_i) + \sum_{i=1}^n f_e(y_i, i, \mathbf{x}) \right]\end{aligned}$$

- ▶ Let's focus on emission feature expectation

$$\mathbb{E}_{\mathbf{y}} \left[ \sum_{i=1}^n f_e(y_i, i, \mathbf{x}) \right] = \sum_{\mathbf{y} \in \mathcal{Y}} P(\mathbf{y}|\mathbf{x}) \left[ \sum_{i=1}^n f_e(y_i, i, \mathbf{x}) \right] = \sum_{i=1}^n \sum_{\mathbf{y} \in \mathcal{Y}} P(\mathbf{y}|\mathbf{x}) f_e(y_i, i, \mathbf{x})$$

# Training CRFs

---

$$\begin{aligned} \frac{\partial}{\partial w} \mathcal{L}(\mathbf{y}^*, \mathbf{x}) &= \sum_{i=2}^n f_t(y_{i-1}^*, y_i^*) + \sum_{i=1}^n f_e(y_i^*, i, \mathbf{x}) \\ &\quad - \mathbb{E}_{\mathbf{y}} \left[ \sum_{i=2}^n f_t(y_{i-1}, y_i) + \sum_{i=1}^n f_e(y_i, i, \mathbf{x}) \right] \end{aligned}$$

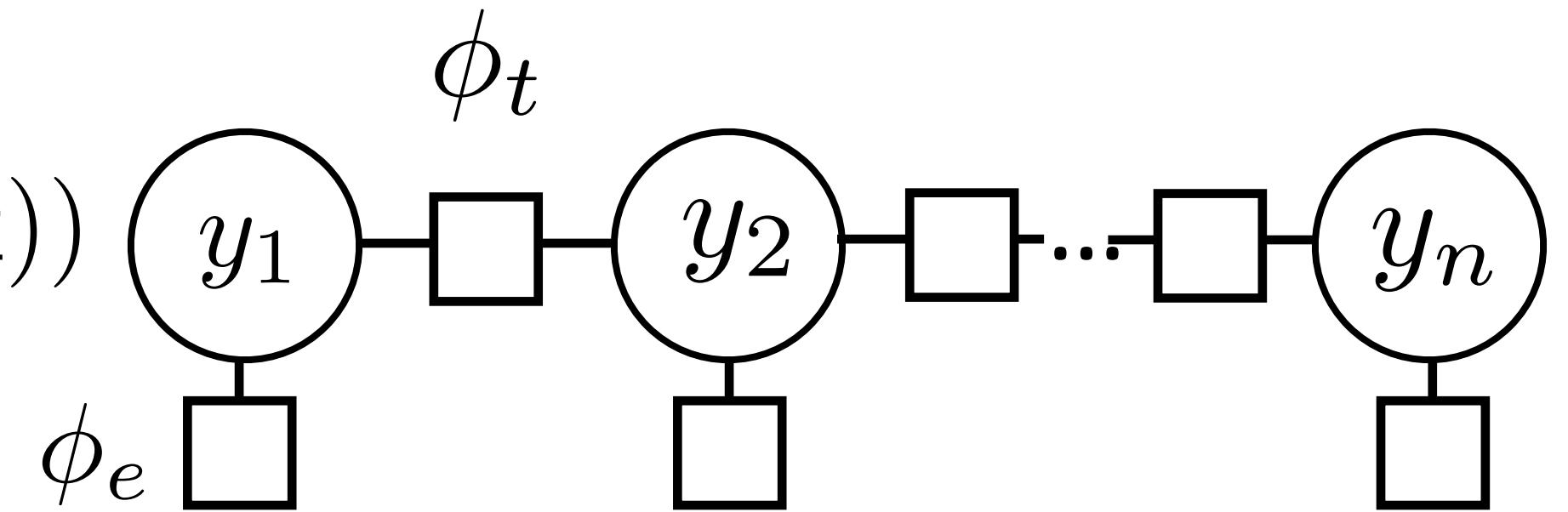
- ▶ Let's focus on emission feature expectation

$$\begin{aligned} \mathbb{E}_{\mathbf{y}} \left[ \sum_{i=1}^n f_e(y_i, i, \mathbf{x}) \right] &= \sum_{\mathbf{y} \in \mathcal{Y}} P(\mathbf{y} | \mathbf{x}) \left[ \sum_{i=1}^n f_e(y_i, i, \mathbf{x}) \right] = \sum_{i=1}^n \sum_{\mathbf{y} \in \mathcal{Y}} P(\mathbf{y} | \mathbf{x}) f_e(y_i, i, \mathbf{x}) \\ &= \sum_{i=1}^n \sum_s P(y_i = s | \mathbf{x}) f_e(s, i, \mathbf{x}) \end{aligned}$$

# Computing Marginals

---

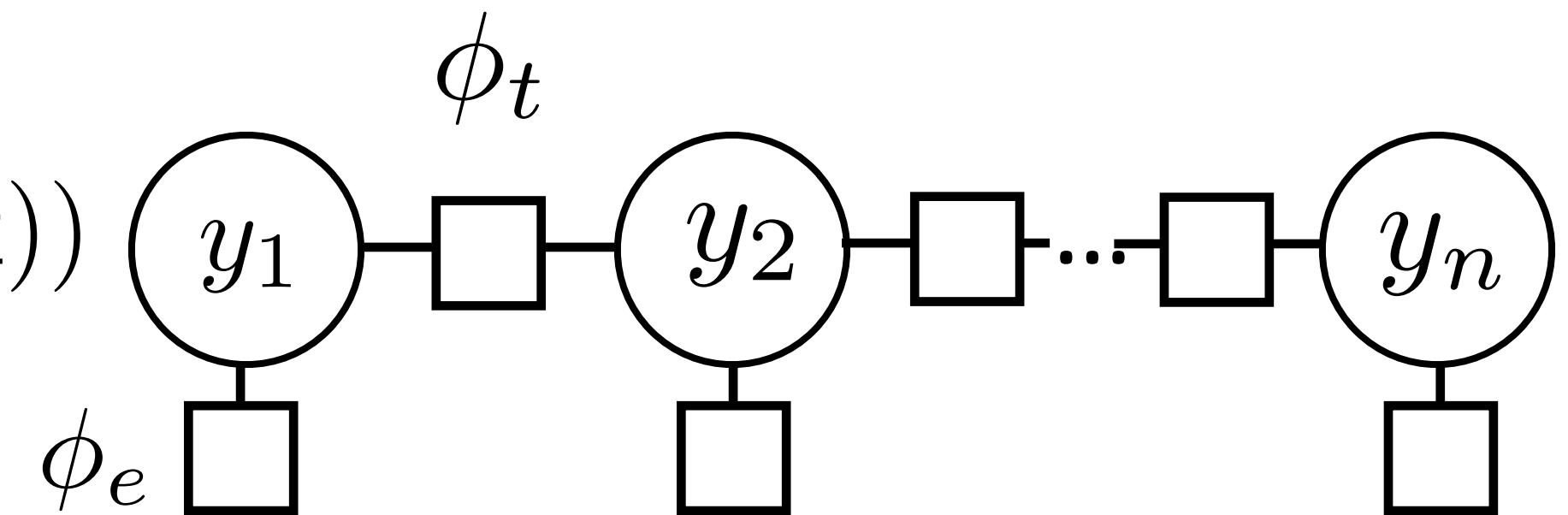
$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(y_i, i, \mathbf{x}))$$



# Computing Marginals

---

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(y_i, i, \mathbf{x}))$$

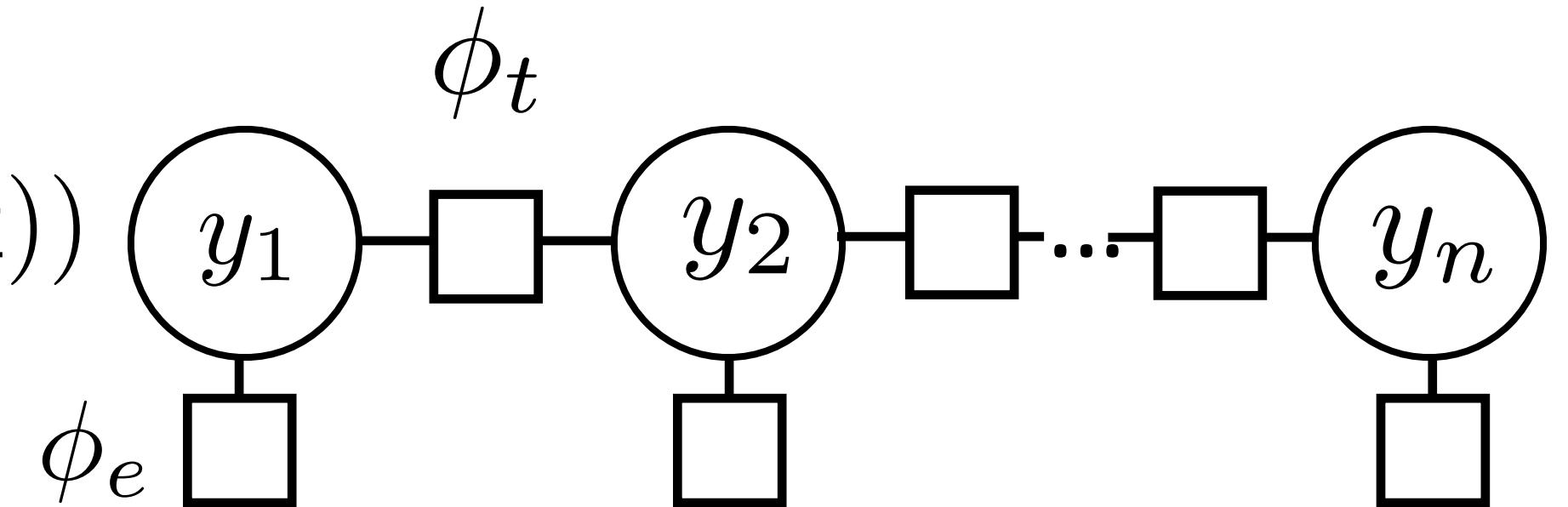


- ▶ Normalizing constant  $Z = \sum_{\mathbf{y}} \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(y_i, i, \mathbf{x}))$

# Computing Marginals

---

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(y_i, i, \mathbf{x}))$$

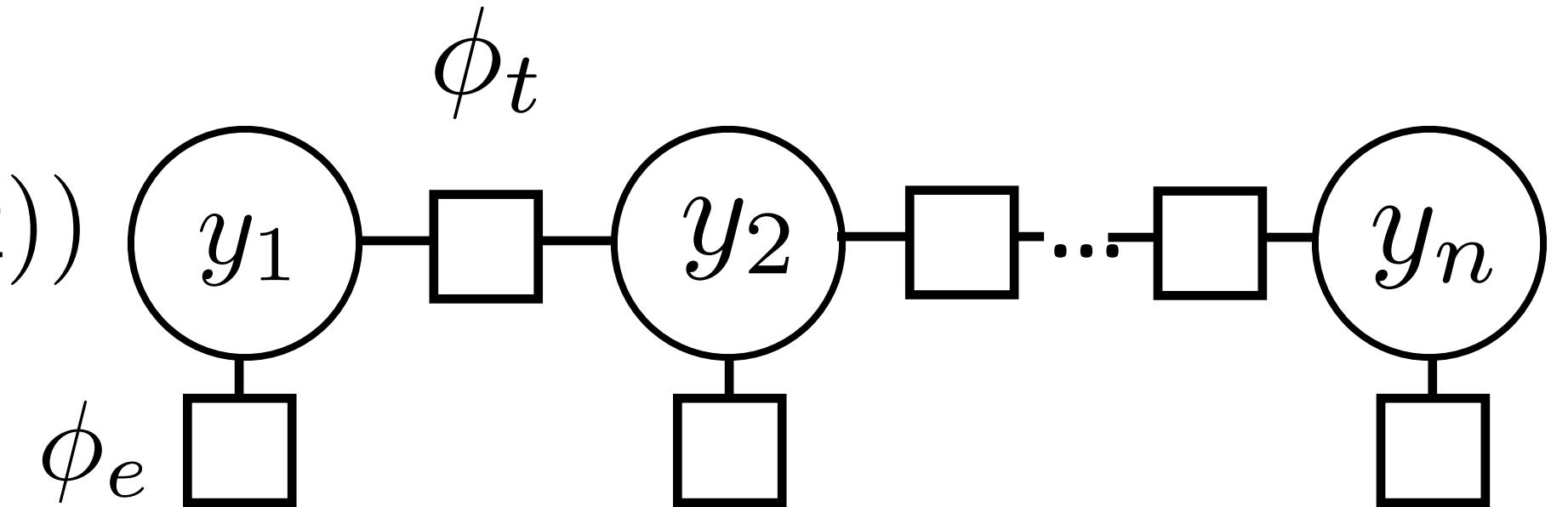


- ▶ Normalizing constant  $Z = \sum_{\mathbf{y}} \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(y_i, i, \mathbf{x}))$
- ▶ Analogous to  $P(\mathbf{x})$  for HMMs

# Computing Marginals

---

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(y_i, i, \mathbf{x}))$$

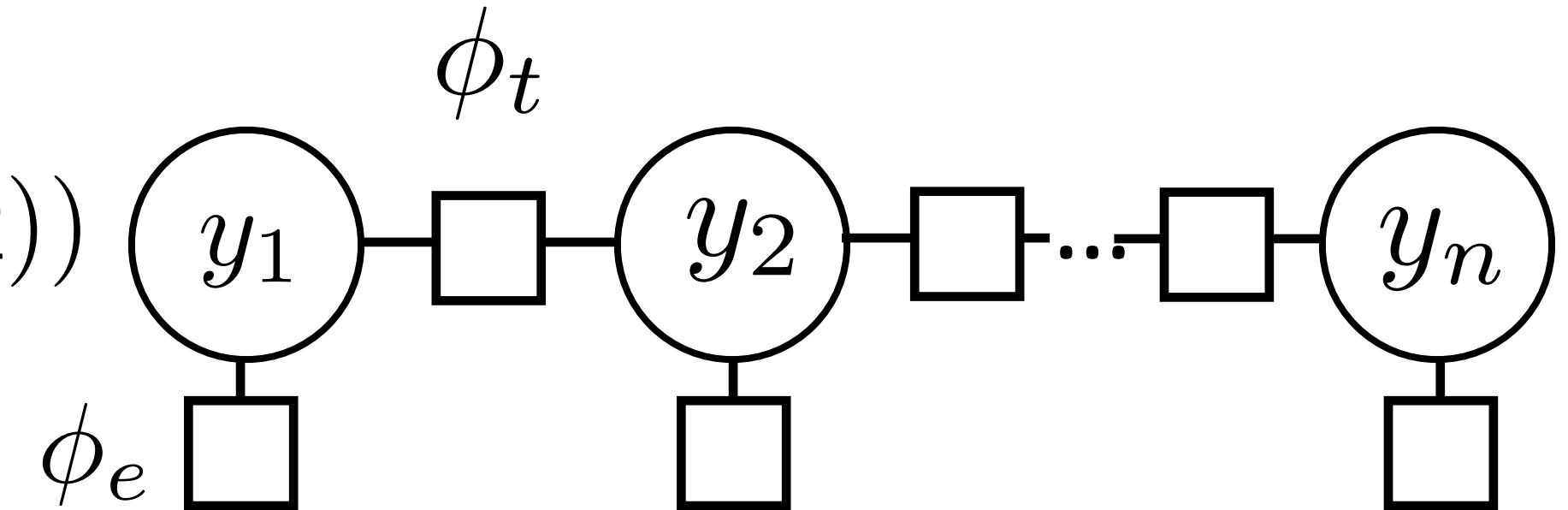


- ▶ Normalizing constant  $Z = \sum_{\mathbf{y}} \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(y_i, i, \mathbf{x}))$
- ▶ Analogous to  $P(\mathbf{x})$  for HMMs
- ▶ For both HMMs and CRFs:

$$P(y_i = s | \mathbf{x}) = \frac{\text{forward}_i(s) \text{backward}_i(s)}{\sum_{s'} \text{forward}_i(s') \text{backward}_i(s')}$$

# Computing Marginals

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(y_i, i, \mathbf{x}))$$



- ▶ Normalizing constant  $Z = \sum_{\mathbf{y}} \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(y_i, i, \mathbf{x}))$
- ▶ Analogous to  $P(\mathbf{x})$  for HMMs
- ▶ For both HMMs and CRFs:

$$P(y_i = s | \mathbf{x}) = \frac{\text{forward}_i(s) \text{backward}_i(s)}{\sum_{s'} \text{forward}_i(s') \text{backward}_i(s')}$$

for HMMs

# Posteriors vs. Probabilities

---

$$P(y_i = s | \mathbf{x}) = \frac{\text{forward}_i(s) \text{backward}_i(s)}{\sum_{s'} \text{forward}_i(s') \text{backward}_i(s')}$$

- ▶ Posterior is *derived* from the parameters and the data (conditioned on  $\mathbf{x}$ !)

# Posteriors vs. Probabilities

---

$$P(y_i = s | \mathbf{x}) = \frac{\text{forward}_i(s) \text{backward}_i(s)}{\sum_{s'} \text{forward}_i(s') \text{backward}_i(s')}$$

- ▶ Posterior is *derived* from the parameters and the data (conditioned on  $\mathbf{x}$ !)

$$P(x_i | y_i), P(y_i | y_{i-1})$$

HMM      Model parameter (usually  
              multinomial distribution)

# Posteriors vs. Probabilities

---

$$P(y_i = s | \mathbf{x}) = \frac{\text{forward}_i(s) \text{backward}_i(s)}{\sum_{s'} \text{forward}_i(s') \text{backward}_i(s')}$$

- ▶ Posterior is *derived* from the parameters and the data (conditioned on  $\mathbf{x}$ !)

HMM

Model parameter (usually  
multinomial distribution)

$$P(x_i | y_i), P(y_i | y_{i-1})$$

$$P(y_i | \mathbf{x}), P(y_{i-1}, y_i | \mathbf{x})$$

# Posteriors vs. Probabilities

---

$$P(y_i = s | \mathbf{x}) = \frac{\text{forward}_i(s) \text{backward}_i(s)}{\sum_{s'} \text{forward}_i(s') \text{backward}_i(s')}$$

- ▶ Posterior is *derived* from the parameters and the data (conditioned on  $\mathbf{x}$ !)

HMM

$$P(x_i | y_i), P(y_i | y_{i-1})$$

Model parameter (usually  
multinomial distribution)

$$P(y_i | \mathbf{x}), P(y_{i-1}, y_i | \mathbf{x})$$

Inferred quantity from  
forward-backward

# Posteriors vs. Probabilities

---

$$P(y_i = s | \mathbf{x}) = \frac{\text{forward}_i(s) \text{backward}_i(s)}{\sum_{s'} \text{forward}_i(s') \text{backward}_i(s')}$$

- ▶ Posterior is *derived* from the parameters and the data (conditioned on  $\mathbf{x}$ !)

HMM

$$P(x_i | y_i), P(y_i | y_{i-1})$$

Model parameter (usually  
multinomial distribution)

$$P(y_i | \mathbf{x}), P(y_{i-1}, y_i | \mathbf{x})$$

Inferred quantity from  
forward-backward

CRF

# Posteriors vs. Probabilities

---

$$P(y_i = s | \mathbf{x}) = \frac{\text{forward}_i(s) \text{backward}_i(s)}{\sum_{s'} \text{forward}_i(s') \text{backward}_i(s')}$$

- ▶ Posterior is *derived* from the parameters and the data (conditioned on  $\mathbf{x}$ !)

HMM

$$P(x_i | y_i), P(y_i | y_{i-1})$$

Model parameter (usually  
multinomial distribution)

$$P(y_i | \mathbf{x}), P(y_{i-1}, y_i | \mathbf{x})$$

Inferred quantity from  
forward-backward

CRF

Undefined (model is by  
definition conditioned on  $\mathbf{x}$ )

# Posteriors vs. Probabilities

---

$$P(y_i = s | \mathbf{x}) = \frac{\text{forward}_i(s) \text{backward}_i(s)}{\sum_{s'} \text{forward}_i(s') \text{backward}_i(s')}$$

- ▶ Posterior is *derived* from the parameters and the data (conditioned on  $\mathbf{x}$ !)

HMM

$P(x_i | y_i), P(y_i | y_{i-1})$   
Model parameter (usually  
multinomial distribution)

CRF

Undefined (model is by  
definition conditioned on  $\mathbf{x}$ )

$P(y_i | \mathbf{x}), P(y_{i-1}, y_i | \mathbf{x})$

Inferred quantity from  
forward-backward

Inferred quantity from  
forward-backward

# Training CRFs

---

- ▶ For emission features:

$$\frac{\partial}{\partial w} \mathcal{L}(\mathbf{y}^*, \mathbf{x}) = \sum_{i=1}^n f_e(y_i^*, i, \mathbf{x}) - \sum_{i=1}^n \sum_s P(y_i = s | \mathbf{x}) f_e(s, i, \mathbf{x})$$

# Training CRFs

---

- ▶ For emission features:

$$\frac{\partial}{\partial w} \mathcal{L}(\mathbf{y}^*, \mathbf{x}) = \sum_{i=1}^n f_e(y_i^*, i, \mathbf{x}) - \sum_{i=1}^n \sum_s P(y_i = s | \mathbf{x}) f_e(s, i, \mathbf{x})$$

gold features – expected features under model

# Training CRFs

---

- ▶ For emission features:

$$\frac{\partial}{\partial w} \mathcal{L}(\mathbf{y}^*, \mathbf{x}) = \sum_{i=1}^n f_e(y_i^*, i, \mathbf{x}) - \sum_{i=1}^n \sum_s P(y_i = s | \mathbf{x}) f_e(s, i, \mathbf{x})$$

gold features – expected features under model

- ▶ Transition features: need to compute  $P(y_i = s_1, y_{i+1} = s_2 | \mathbf{x})$  using forward-backward as well

# CRFs Outline

---

- Model:  $P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(y_i, i, \mathbf{x}))$   
$$P(\mathbf{y}|\mathbf{x}) \propto \exp w^\top \left[ \sum_{i=2}^n f_t(y_{i-1}, y_i) + \sum_{i=1}^n f_e(y_i, i, \mathbf{x}) \right]$$
- Inference: argmax  $P(\mathbf{y}|\mathbf{x})$  from Viterbi
- Learning: run forward-backward to compute posterior probabilities; then

$$\frac{\partial}{\partial w} \mathcal{L}(\mathbf{y}^*, \mathbf{x}) = \sum_{i=1}^n f_e(y_i^*, i, \mathbf{x}) - \sum_{i=1}^n \sum_s P(y_i = s | \mathbf{x}) f_e(s, i, \mathbf{x})$$

# Pseudocode

---

for each epoch

    for each example

# Pseudocode

---

for each epoch

    for each example

        extract features on each emission and transition (look up in cache)

# Pseudocode

---

```
for each epoch
    for each example
        extract features on each emission and transition (look up in cache)
        compute potentials phi based on features + weights
```

# Pseudocode

---

for each epoch

    for each example

        extract features on each emission and transition (look up in cache)

        compute potentials phi based on features + weights

        compute marginal probabilities with forward-backward

# Pseudocode

---

```
for each epoch
    for each example
        extract features on each emission and transition (look up in cache)
        compute potentials phi based on features + weights
        compute marginal probabilities with forward-backward
        accumulate gradient over all emissions and transitions
```

# How to “Cheat” with Automatic Differentiation

---

for each epoch

    for each example

        extract features on each emission and transition (look up in cache)

        compute potentials phi based on features + weights

        compute marginal probabilities with forward-backward

        accumulate gradient over all emissions and transitions

# How to “Cheat” with Automatic Differentiation

---

for each epoch

for each example

extract features on each emission and transition (look up in cache)

compute potentials phi based on features + weights

compute marginal probabilities with forward-backward

accumulate gradient over all emissions and transitions

# How to “Cheat” with Automatic Differentiation

---

for each epoch

for each example

extract features on each emission and transition (look up in cache)

compute potentials  $\phi$  based on features + weights

compute marginal probabilities with forward-backward

accumulate gradient over all emissions and transitions

- 
- Compute  $P(Y|X)$ , using the forward algorithm to get  $Z(X)$
  - Use auto-diff through the computation graph of the dynamic program, to compute gradients.

# Structured SVM / Structured Perceptron

# Structured Perceptron

---

- ▶ Structured Perceptron Update:

$$\hat{y} = \operatorname{argmax}_{y \in \mathcal{Y}} w^\top f(x, y)$$

$$w = w + f(x, y^*) - f(x, \hat{y})$$

# Structured Perceptron

---

- ▶ Structured Perceptron Update:

$$\hat{y} = \operatorname{argmax}_{y \in \mathcal{Y}} w^\top f(x, y) \quad \xleftarrow{\text{Viterbi Algorithm}}$$

$$w = w + f(x, y^*) - f(x, \hat{y})$$

# Structured Perceptron

---

- ▶ Structured Perceptron Update:

$$\hat{y} = \operatorname{argmax}_{y \in \mathcal{Y}} w^\top f(x, y) \quad \xleftarrow{\text{Viterbi Algorithm}}$$

$$w = w + f(x, y^*) - f(x, \hat{y})$$

- ▶ Compare to gradient of CRF:

$$\begin{aligned} \frac{\partial}{\partial w} \mathcal{L}(\mathbf{y}^*, \mathbf{x}) &= \sum_{i=2}^n f_t(y_{i-1}^*, y_i^*) + \sum_{i=1}^n f_e(y_i^*, i, \mathbf{x}) \\ &\quad - \mathbb{E}_{\mathbf{y}} \left[ \sum_{i=2}^n f_t(y_{i-1}, y_i) + \sum_{i=1}^n f_e(y_i, i, \mathbf{x}) \right] \end{aligned}$$

# Structured Perceptron

---

- ▶ Structured Perceptron Update:

$$\hat{y} = \operatorname{argmax}_{y \in \mathcal{Y}} w^\top f(x, y) \quad \xleftarrow{\text{Viterbi Algorithm}}$$

$$w = w + f(x, y^*) - f(x, \hat{y})$$

- ▶ Compare to gradient of CRF:

$$\frac{\partial}{\partial w} \mathcal{L}(y^*, \mathbf{x}) = \sum_{i=2}^n f_t(y_{i-1}^*, y_i^*) + \sum_{i=1}^n f_e(y_i^*, i, \mathbf{x})$$

Replaces Expectation  
With argmax

$$- \mathbb{E}_y \left[ \sum_{i=2}^n f_t(y_{i-1}, y_i) + \sum_{i=1}^n f_e(y_i, i, \mathbf{x}) \right]$$

# Structured SVM

---

# Structured SVM

---

- CRF:  $\log P(\mathbf{y}|\mathbf{x}) \propto \sum_{i=2}^n w^\top f_t(y_{i-1}, y_i) + \sum_{i=1}^n w^\top f_e(x_i, y_i)$

# Structured SVM

---

- CRF:  $\log P(\mathbf{y}|\mathbf{x}) \propto \sum_{i=2}^n w^\top f_t(y_{i-1}, y_i) + \sum_{i=1}^n w^\top f_e(x_i, y_i)$
- We can formulate an SVM using the same features

# Structured SVM

---

- CRF:  $\log P(\mathbf{y}|\mathbf{x}) \propto \sum_{i=2}^n w^\top f_t(y_{i-1}, y_i) + \sum_{i=1}^n w^\top f_e(x_i, y_i)$
- We can formulate an SVM using the same features

$$w^\top f(\mathbf{x}, \mathbf{y}) = \sum_{i=2}^n w^\top f_t(y_{i-1}, y_i) + \sum_{i=1}^n w^\top f_e(x_i, y_i)$$

# Structured SVM

---

- CRF:  $\log P(\mathbf{y}|\mathbf{x}) \propto \sum_{i=2}^n w^\top f_t(y_{i-1}, y_i) + \sum_{i=1}^n w^\top f_e(x_i, y_i)$
- We can formulate an SVM using the same features

$$w^\top f(\mathbf{x}, \mathbf{y}) = \sum_{i=2}^n w^\top f_t(y_{i-1}, y_i) + \sum_{i=1}^n w^\top f_e(x_i, y_i)$$

$$\text{Minimize } \lambda \|w\|_2^2 + \sum_{j=1}^m \xi_j$$

# Structured SVM

---

- CRF:  $\log P(\mathbf{y}|\mathbf{x}) \propto \sum_{i=2}^n w^\top f_t(y_{i-1}, y_i) + \sum_{i=1}^n w^\top f_e(x_i, y_i)$
- We can formulate an SVM using the same features

$$w^\top f(\mathbf{x}, \mathbf{y}) = \sum_{i=2}^n w^\top f_t(y_{i-1}, y_i) + \sum_{i=1}^n w^\top f_e(x_i, y_i)$$

$$\begin{aligned} & \text{Minimize } \lambda \|w\|_2^2 + \sum_{j=1}^m \xi_j \\ \text{s.t. } & \forall j \quad \xi_j \geq 0 \\ & \forall j \forall \mathbf{y} \in \mathcal{Y} \quad w^\top f(\mathbf{x}_j, \mathbf{y}_j^*) \geq w^\top f(\mathbf{x}_j, \mathbf{y}) + \ell(\mathbf{y}, \mathbf{y}_j^*) - \xi_j \end{aligned}$$

# Structured SVM

---

$$w^\top f(\mathbf{x}, \mathbf{y}) = \sum_{i=2}^n w^\top f_t(y_{i-1}, y_i) + \sum_{i=1}^n w^\top f_e(x_i, y_i)$$

$$\begin{aligned} & \text{Minimize} \quad \lambda \|w\|_2^2 + \sum_{j=1}^m \xi_j \\ \text{s.t.} \quad & \forall j \quad \xi_j \geq 0 \\ & \forall j \forall \mathbf{y} \in \mathcal{Y} \quad w^\top f(\mathbf{x}_j, \mathbf{y}_j^*) \geq w^\top f(\mathbf{x}_j, \mathbf{y}) + \ell(\mathbf{y}, \mathbf{y}_j^*) - \xi_j \end{aligned}$$

# Structured SVM

$$w^\top f(\mathbf{x}, \mathbf{y}) = \sum_{i=2}^n w^\top f_t(y_{i-1}, y_i) + \sum_{i=1}^n w^\top f_e(x_i, y_i)$$

$$\begin{aligned} & \text{Minimize } \lambda \|w\|_2^2 + \sum_{j=1}^m \xi_j \\ \text{s.t. } & \forall j \quad \xi_j \geq 0 \\ & \forall j \forall \mathbf{y} \in \mathcal{Y} \quad w^\top f(\mathbf{x}_j, \mathbf{y}_j^*) \geq w^\top f(\mathbf{x}_j, \mathbf{y}) + \ell(\mathbf{y}, \mathbf{y}_j^*) - \xi_j \end{aligned}$$


- Exponentially large state space! Use Viterbi for loss-augmented decode

# Structured SVM

$$w^\top f(\mathbf{x}, \mathbf{y}) = \sum_{i=2}^n w^\top f_t(y_{i-1}, y_i) + \sum_{i=1}^n w^\top f_e(x_i, y_i)$$

$$\begin{aligned} & \text{Minimize } \lambda \|w\|_2^2 + \sum_{j=1}^m \xi_j \\ \text{s.t. } & \forall j \quad \xi_j \geq 0 \\ & \forall j \forall \mathbf{y} \in \mathcal{Y} \quad w^\top f(\mathbf{x}_j, \mathbf{y}_j^*) \geq w^\top f(\mathbf{x}_j, \mathbf{y}) + \ell(\mathbf{y}, \mathbf{y}_j^*) - \xi_j \end{aligned}$$

- Exponentially large state space! Use Viterbi for loss-augmented decode
- Same as normal Viterbi but boost wrong labels' scores by 1 (if using Hamming loss)

# Structured SVM

$$w^\top f(\mathbf{x}, \mathbf{y}) = \sum_{i=2}^n w^\top f_t(y_{i-1}, y_i) + \sum_{i=1}^n w^\top f_e(x_i, y_i)$$

$$\begin{aligned} & \text{Minimize } \lambda \|w\|_2^2 + \sum_{j=1}^m \xi_j \\ \text{s.t. } & \forall j \quad \xi_j \geq 0 \\ & \forall j \forall \mathbf{y} \in \mathcal{Y} \quad w^\top f(\mathbf{x}_j, \mathbf{y}_j^*) \geq w^\top f(\mathbf{x}_j, \mathbf{y}) + \ell(\mathbf{y}, \mathbf{y}_j^*) - \xi_j \end{aligned}$$

- Exponentially large state space! Use Viterbi for loss-augmented decode
- Same as normal Viterbi but boost wrong labels' scores by 1 (if using Hamming loss)
- Only need Viterbi, not forward-backward...hmm...

**NER**

# NER

---

# NER

---

- ▶ CRF with lexical features can get around 85 F1 on this problem

# NER

---

- ▶ CRF with lexical features can get around 85 F1 on this problem
- ▶ Other pieces of information that many systems capture

# NER

---

- ▶ CRF with lexical features can get around 85 F1 on this problem
- ▶ Other pieces of information that many systems capture
- ▶ World knowledge:

# NER

---

- ▶ CRF with lexical features can get around 85 F1 on this problem
- ▶ Other pieces of information that many systems capture
- ▶ World knowledge:

The delegation met the president at the airport, Tanjug said.

# NER

---

- ▶ CRF with lexical features can get around 85 F1 on this problem
- ▶ Other pieces of information that many systems capture
- ▶ World knowledge:

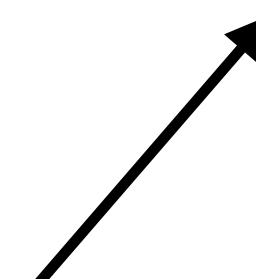
The delegation met the president at the airport, **Tanjug** said.

**Tanjug**

---

From Wikipedia, the free encyclopedia

**Tanjug** (/tʌnjʊg/) ([Serbian Cyrillic](#): Танјуг) is a Serbian state news agency based in [Belgrade](#).<sup>[2]</sup>



# Nonlocal Features

---

The delegation met the president at the airport, Tanjug said.

# Nonlocal Features

---

ORG?  
PER?

The delegation met the president at the airport, Tanjug said.

# Nonlocal Features

---

The news agency Tanjug reported on the outcome of the meeting.

ORG?  
PER?

The delegation met the president at the airport, Tanjug said.

# Nonlocal Features

---

The news agency Tanjug reported on the outcome of the meeting.

ORG?  
PER?

The delegation met the president at the airport, Tanjug said.

# Nonlocal Features

---

The news agency Tanjug reported on the outcome of the meeting.

ORG?  
PER?

The delegation met the president at the airport, Tanjug said.

- More complex factor graph structures can let you capture this, or just decode sentences in order and use features on previous sentences

# Semi-Markov Models

---

*Barack Obama will travel to Hangzhou today for the G20 meeting .*

# Semi-Markov Models

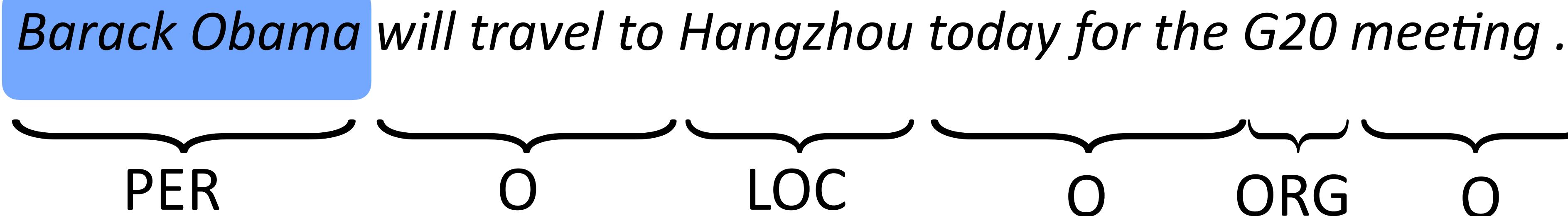
---

*Barack Obama will travel to Hangzhou today for the G20 meeting .*

- ▶ Chunk-level prediction rather than token-level BIO

# Semi-Markov Models

---

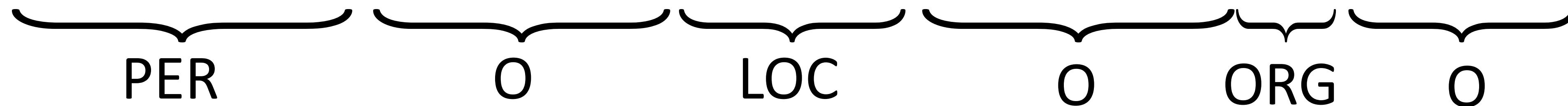


- ▶ Chunk-level prediction rather than token-level BIO

# Semi-Markov Models

---

*Barack Obama will travel to Hangzhou today for the G20 meeting .*

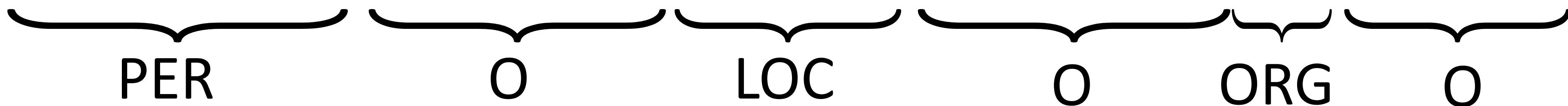


- ▶ Chunk-level prediction rather than token-level BIO
- ▶  $y$  is a set of touching spans of the sentence

# Semi-Markov Models

---

*Barack Obama will travel to Hangzhou today for the G20 meeting .*

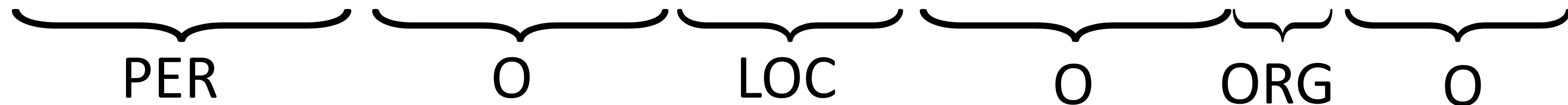


- ▶ Chunk-level prediction rather than token-level BIO
- ▶  $y$  is a set of touching spans of the sentence
- ▶ Pros: features can look at whole span at once

# Semi-Markov Models

---

*Barack Obama will travel to Hangzhou today for the G20 meeting .*



- ▶ Chunk-level prediction rather than token-level BIO
- ▶  $y$  is a set of touching spans of the sentence
- ▶ Pros: features can look at whole span at once
- ▶ Cons: there's an extra factor of  $n$  in the dynamic programs

# Evaluating NER

---

B-PER I-PER O O O B-LOC O O O B-ORG O O

*Barack Obama will travel to Hangzhou today for the G20 meeting .*

PERSON

LOC

ORG

# Evaluating NER

---

B-PER I-PER O O O B-LOC O O O B-ORG O O

*Barack Obama will travel to Hangzhou today for the G20 meeting .*

PERSON

LOC

ORG

- ▶ Prediction of all Os still gets 66% accuracy on this example!

# Evaluating NER

---

B-PER	I-PER	0	0	0	B-LOC	0	0	0	B-ORG	0	0
-------	-------	---	---	---	-------	---	---	---	-------	---	---

*Barack Obama will travel to Hangzhou today for the G20 meeting .*

PERSON

LOC

ORG

- ▶ Prediction of all Os still gets 66% accuracy on this example!
- ▶ What we really want to know: how many named entity *chunk* predictions did we get right?

# Evaluating NER

---

B-PER	I-PER	0	0	0	B-LOC	0	0	0	B-ORG	0	0
-------	-------	---	---	---	-------	---	---	---	-------	---	---

*Barack Obama will travel to Hangzhou today for the G20 meeting .*

PERSON

LOC

ORG

- ▶ Prediction of all Os still gets 66% accuracy on this example!
- ▶ What we really want to know: how many named entity *chunk* predictions did we get right?
- ▶ Precision: of the ones we predicted, how many are right?

# Evaluating NER

---

B-PER	I-PER	0	0	0	B-LOC	0	0	0	B-ORG	0	0
-------	-------	---	---	---	-------	---	---	---	-------	---	---

*Barack Obama will travel to Hangzhou today for the G20 meeting .*

PERSON

LOC

ORG

- ▶ Prediction of all Os still gets 66% accuracy on this example!
- ▶ What we really want to know: how many named entity *chunk* predictions did we get right?
- ▶ Precision: of the ones we predicted, how many are right?
- ▶ Recall: of the gold named entities, how many did we find?

# Evaluating NER

---

B-PER	I-PER	0	0	0	B-LOC	0	0	0	B-ORG	0	0
-------	-------	---	---	---	-------	---	---	---	-------	---	---

*Barack Obama will travel to Hangzhou today for the G20 meeting .*

PERSON

LOC

ORG

- ▶ Prediction of all Os still gets 66% accuracy on this example!
- ▶ What we really want to know: how many named entity *chunk* predictions did we get right?
- ▶ Precision: of the ones we predicted, how many are right?
- ▶ Recall: of the gold named entities, how many did we find?
- ▶ F-measure: harmonic mean of these two

# How well do NER systems do?

---

	System	Resources Used	$F_1$
+	LBJ-NER	Wikipedia, Nonlocal Features, Word-class Model	90.80
-	(Suzuki and Isozaki, 2008)	Semi-supervised on 1G-word unlabeled data	89.92
-	(Ando and Zhang, 2005)	Semi-supervised on 27M-word unlabeled data	89.31
-	(Kazama and Torisawa, 2007a)	Wikipedia	88.02
-	(Krishnan and Manning, 2006)	Non-local Features	87.24
-	(Kazama and Torisawa, 2007b)	Non-local Features	87.17
+	(Finkel et al., 2005)	Non-local Features	86.86

# How well do NER systems do?

	System	Resources Used	$F_1$	
+	LBJ-NER	Wikipedia, Nonlocal Features, Word-class Model	90.80	Lample et al. (2016)
-	(Suzuki and Isozaki, 2008)	Semi-supervised on 1G-word unlabeled data	89.92	LSTM-CRF (no char) 90.20
-	(Ando and Zhang, 2005)	Semi-supervised on 27M-word unlabeled data	89.31	LSTM-CRF 90.94
-	(Kazama and Torisawa, 2007a)	Wikipedia	88.02	S-LSTM (no char) 87.96
-	(Krishnan and Manning, 2006)	Non-local Features	87.24	S-LSTM 90.33
-	(Kazama and Torisawa, 2007b)	Non-local Features	87.17	
+	(Finkel et al., 2005)	Non-local Features	86.86	

# How well do NER systems do?

	System	Resources Used	$F_1$	
+	LBJ-NER	Wikipedia, Nonlocal Features, Word-class Model	90.80	Lample et al. (2016)
-	(Suzuki and Isozaki, 2008)	Semi-supervised on 1G-word unlabeled data	89.92	LSTM-CRF (no char) 90.20 LSTM-CRF <b>90.94</b>
-	(Ando and Zhang, 2005)	Semi-supervised on 27M-word unlabeled data	89.31	S-LSTM (no char) 87.96 S-LSTM 90.33
-	(Kazama and Torisawa, 2007a)	Wikipedia	88.02	BiLSTM-CRF + ELMo 92.2
-	(Krishnan and Manning, 2006)	Non-local Features	87.24	Peters et al. (2018)
-	(Kazama and Torisawa, 2007b)	Non-local Features	87.17	
+	(Finkel et al., 2005)	Non-local Features	86.86	

# How well do NER systems do?

	System	Resources Used	$F_1$	
+	LBJ-NER	Wikipedia, Nonlocal Features, Word-class Model	90.80	Lample et al. (2016)
-	(Suzuki and Isozaki, 2008)	Semi-supervised on 1G-word unlabeled data	89.92	LSTM-CRF (no char) 90.20 LSTM-CRF <b>90.94</b>
-	(Ando and Zhang, 2005)	Semi-supervised on 27M-word unlabeled data	89.31	S-LSTM (no char) 87.96 S-LSTM 90.33
-	(Kazama and Torisawa, 2007a)	Wikipedia	88.02	BiLSTM-CRF + ELMo <b>92.2</b>
-	(Krishnan and Manning, 2006)	Non-local Features	87.24	Peters et al. (2018)
-	(Kazama and Torisawa, 2007b)	Non-local Features	87.17	BERT-Large
+	(Finkel et al., 2005)	Non-local Features	86.86	Devlin et al. (2019) <b>92.8</b>

# Beam Search

# Viterbi Time Complexity

---

VBD

VBN VBZ

NNP NNS

VB

VBP VBZ

NN NNS CD NN

Fed raises interest rates 0.5 percent

# Viterbi Time Complexity

---

VBD              VB  
VBN VBZ      VBP      VBZ  
NNP NNS      NN      NNS CD NN

Fed raises interest rates 0.5 percent

- ▶ n word sentence, s tags to consider – what is the time complexity?

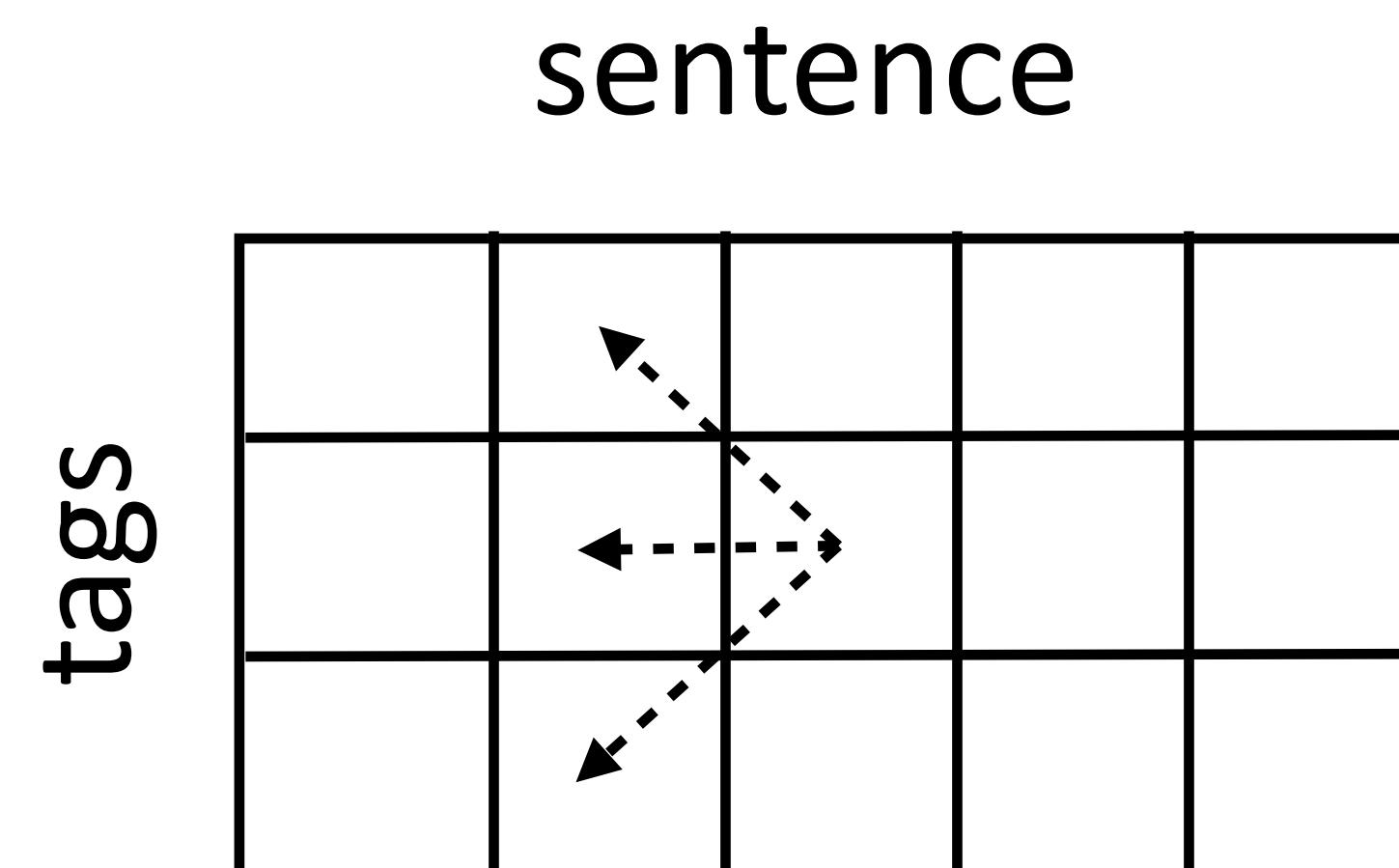
# Viterbi Time Complexity

---

VBD              VB  
VBN **VBZ**      VBP      VBZ  
**NNP** NNS      **NN**      **NNS** **CD** **NN**

Fed raises interest rates 0.5 percent

- ▶ n word sentence, s tags to consider – what is the time complexity?



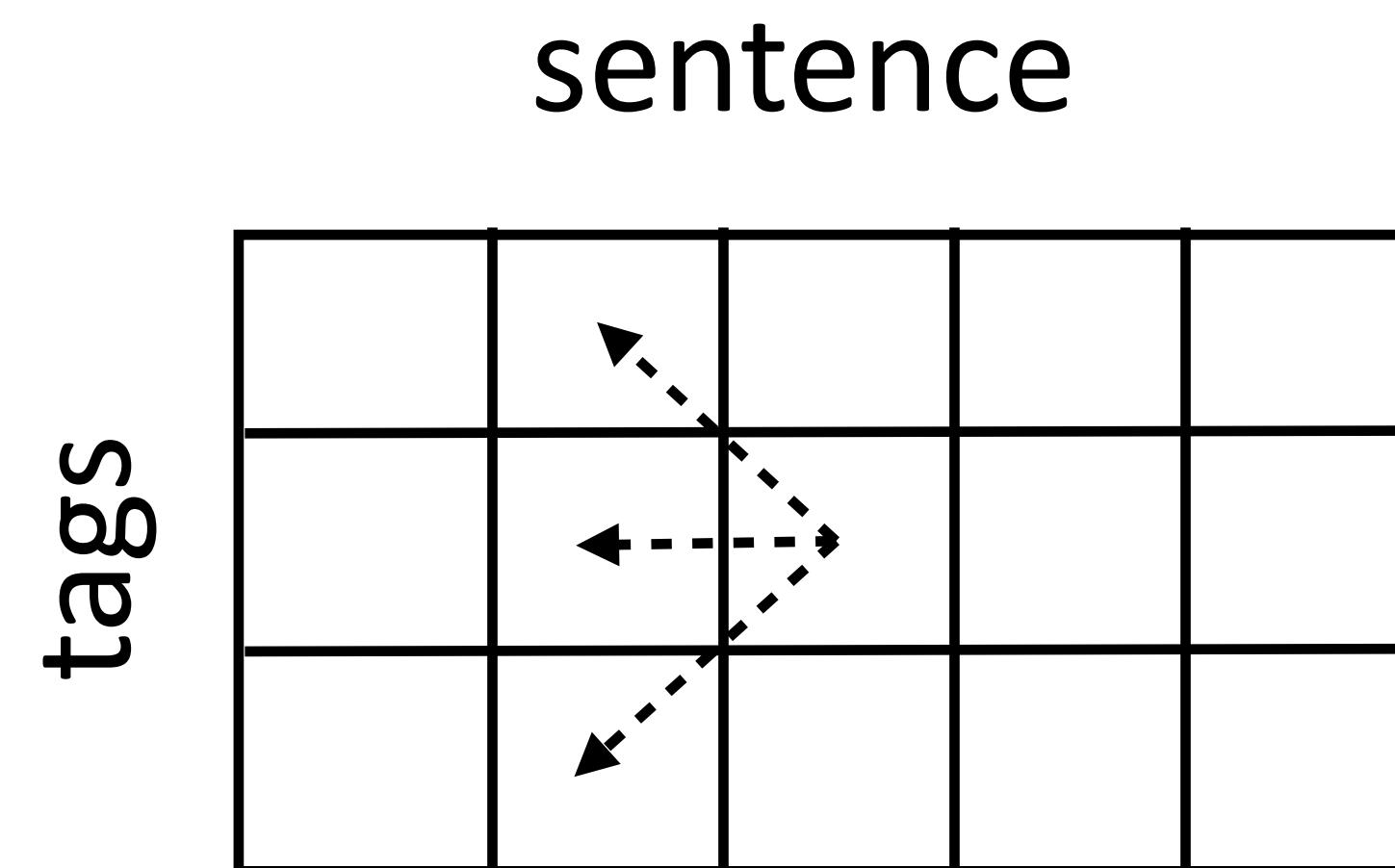
# Viterbi Time Complexity

---

VBD              VB  
VBN **VBZ**      VBP      VBZ  
**NNP** NNS      **NN**      **NNS** **CD** **NN**

Fed raises interest rates 0.5 percent

- ▶ n word sentence, s tags to consider – what is the time complexity?



- ▶  $O(ns^2)$  – s is ~40 for POS, n is ~20

# Viterbi Time Complexity

---

VBD

VBN VBZ

NNP NNS

VB

VBP VBZ

NN NNS CD NN

Fed raises interest rates 0.5 percent

# Viterbi Time Complexity

---

VBD              VB  
VBN VBZ      VBP      VBZ  
NNP NNS      NN      NNS CD NN

Fed raises interest rates 0.5 percent

- ▶ Many tags are totally implausible

# Viterbi Time Complexity

---

VBD              VB  
VBN VBZ      VBP      VBZ  
NNP NNS      NN      NNS CD NN

Fed raises interest rates 0.5 percent

- ▶ Many tags are totally implausible
- ▶ Can any of these be:
  - ▶ Determiners?
  - ▶ Prepositions?
  - ▶ Adjectives?

# Viterbi Time Complexity

---

VBD              VB  
VBN VBZ      VBP      VBZ  
NNP NNS      NN      NNS CD NN  
Fed raises interest rates 0.5 percent

- ▶ Many tags are totally implausible
- ▶ Can any of these be:
  - ▶ Determiners?
  - ▶ Prepositions?
  - ▶ Adjectives?
- ▶ Features quickly eliminate many outcomes from consideration — don't need to consider these going forward

# Beam Search

---

# Beam Search

---

- ▶ Maintain a beam of  $k$  plausible states at the current timestep

# Beam Search

---

- ▶ Maintain a beam of  $k$  plausible states at the current timestep
- ▶ Expand all states, only keep  $k$  top hypotheses at new timestep

# Beam Search

---

- ▶ Maintain a beam of  $k$  plausible states at the current timestep
- ▶ Expand all states, only keep  $k$  top hypotheses at new timestep

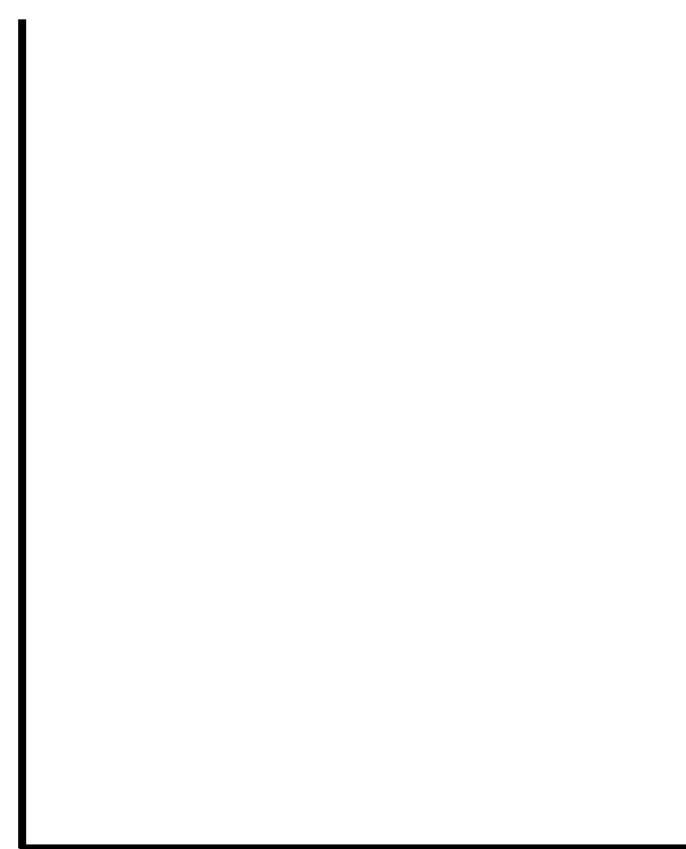
Fed

raises

# Beam Search

---

- ▶ Maintain a beam of  $k$  plausible states at the current timestep
- ▶ Expand all states, only keep  $k$  top hypotheses at new timestep



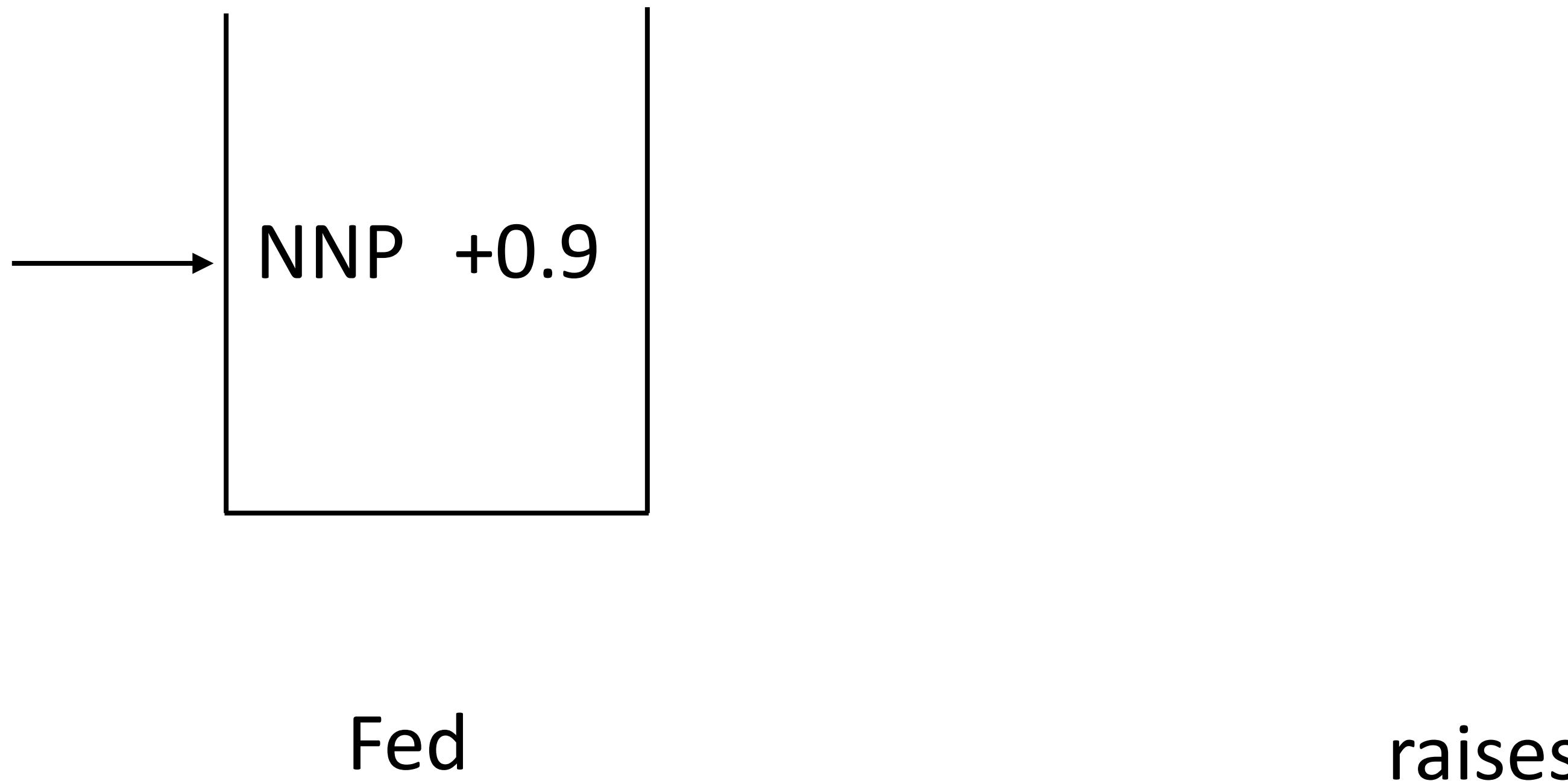
Fed

raises

# Beam Search

---

- ▶ Maintain a beam of  $k$  plausible states at the current timestep
- ▶ Expand all states, only keep  $k$  top hypotheses at new timestep



# Beam Search

---

- ▶ Maintain a beam of  $k$  plausible states at the current timestep
- ▶ Expand all states, only keep  $k$  top hypotheses at new timestep

→	VBD	+1.2
→	NNP	+0.9

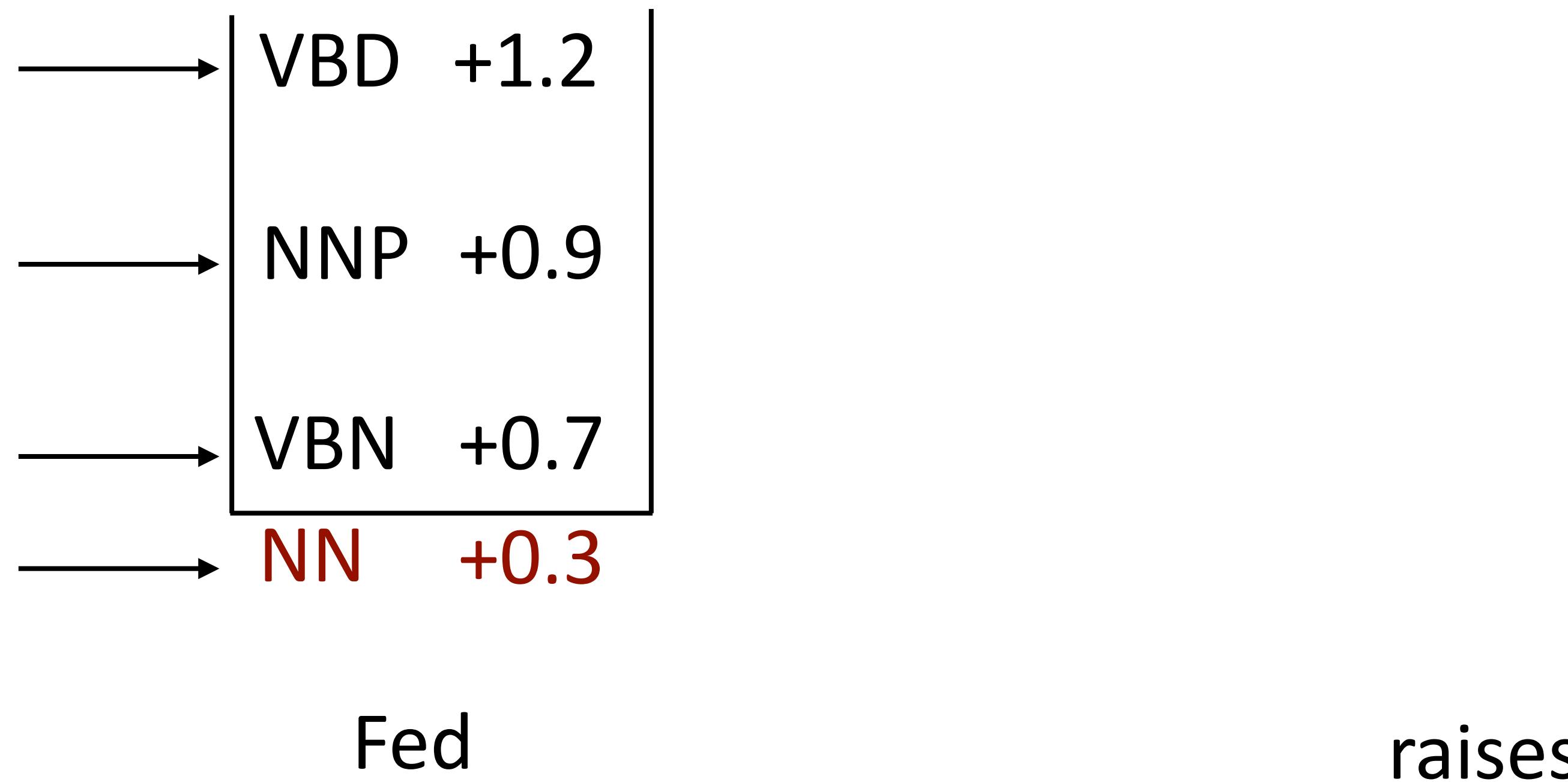
Fed

raises

# Beam Search

---

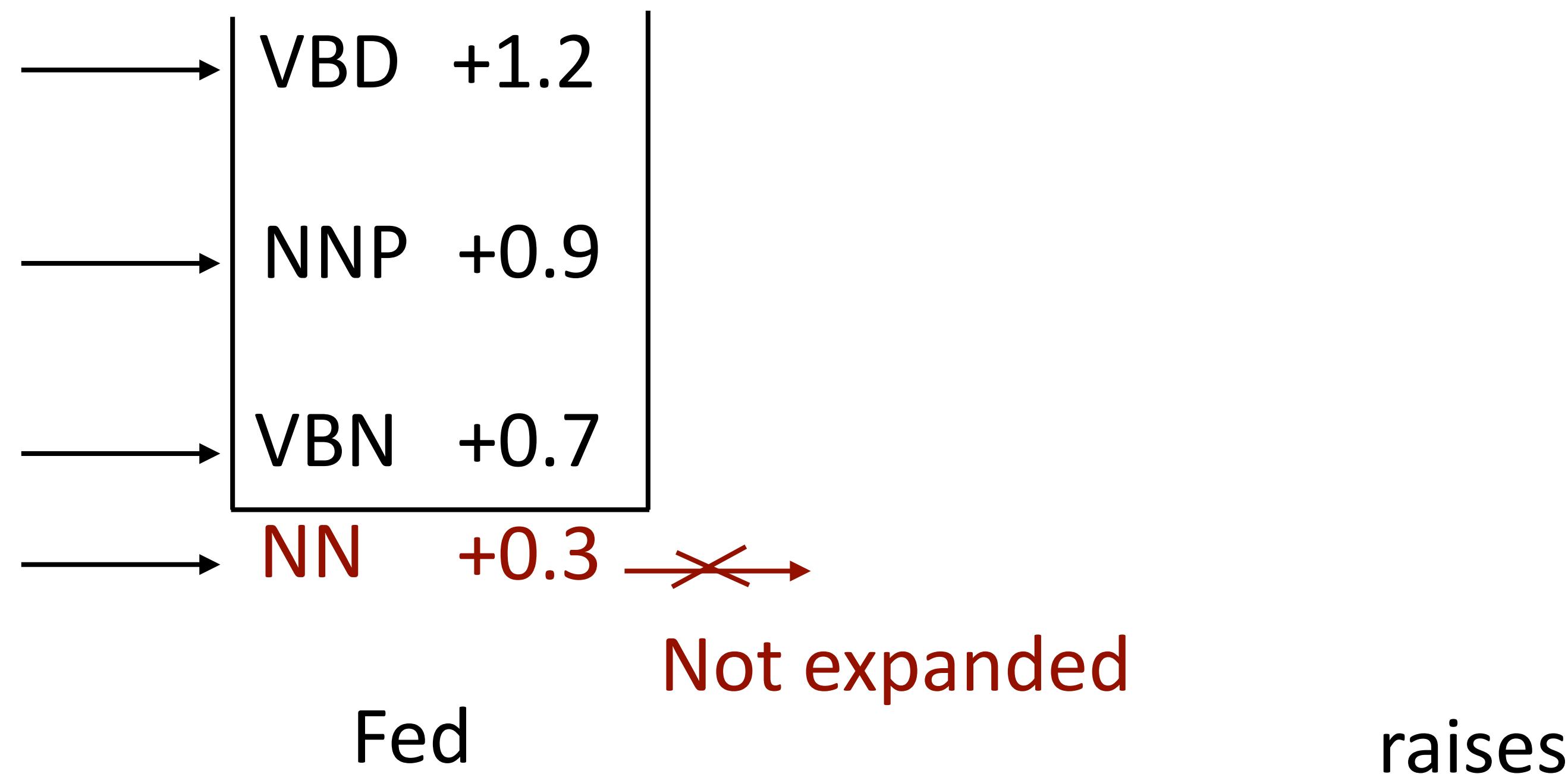
- ▶ Maintain a beam of  $k$  plausible states at the current timestep
- ▶ Expand all states, only keep  $k$  top hypotheses at new timestep



# Beam Search

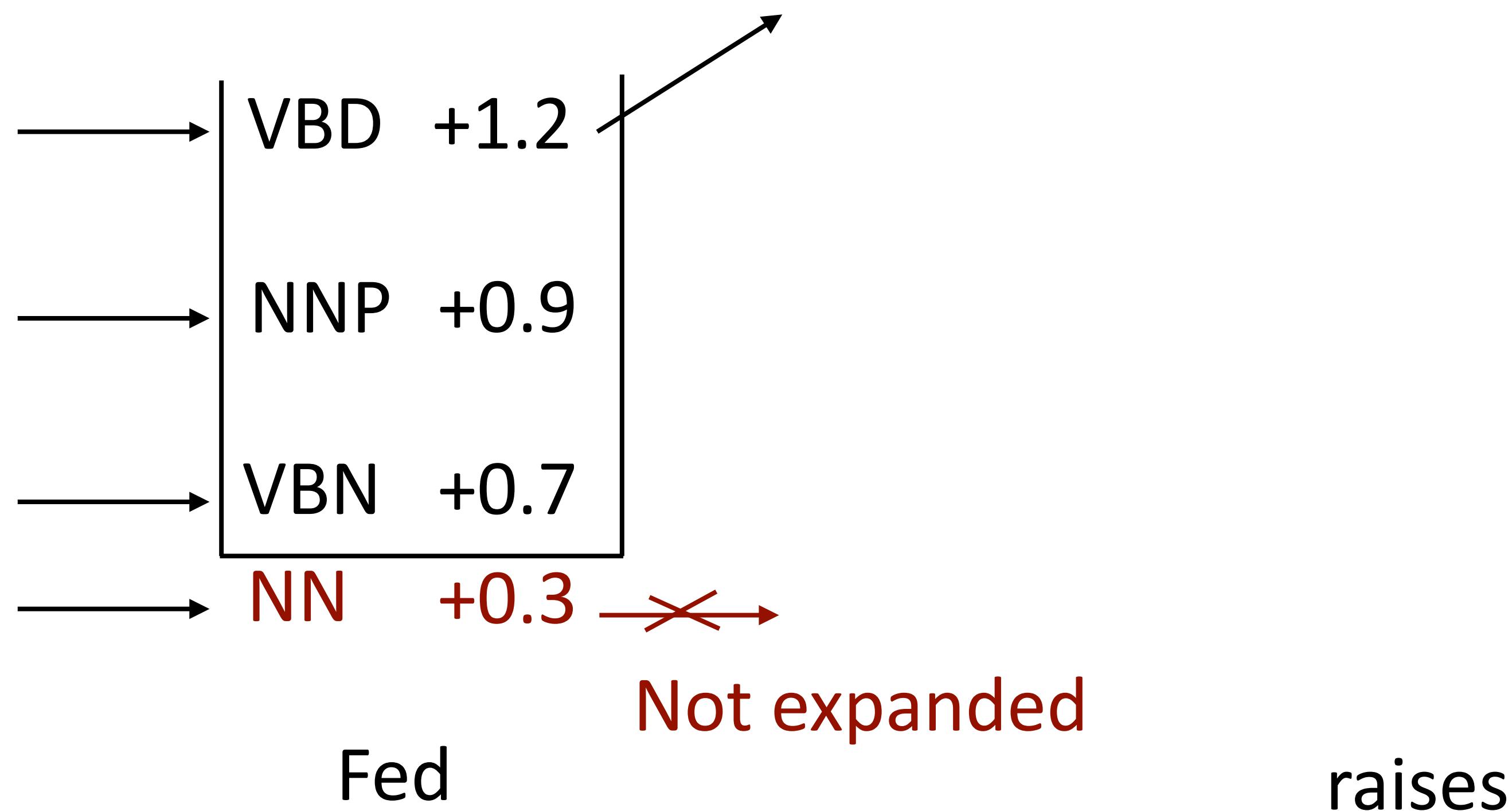
---

- ▶ Maintain a beam of  $k$  plausible states at the current timestep
- ▶ Expand all states, only keep  $k$  top hypotheses at new timestep



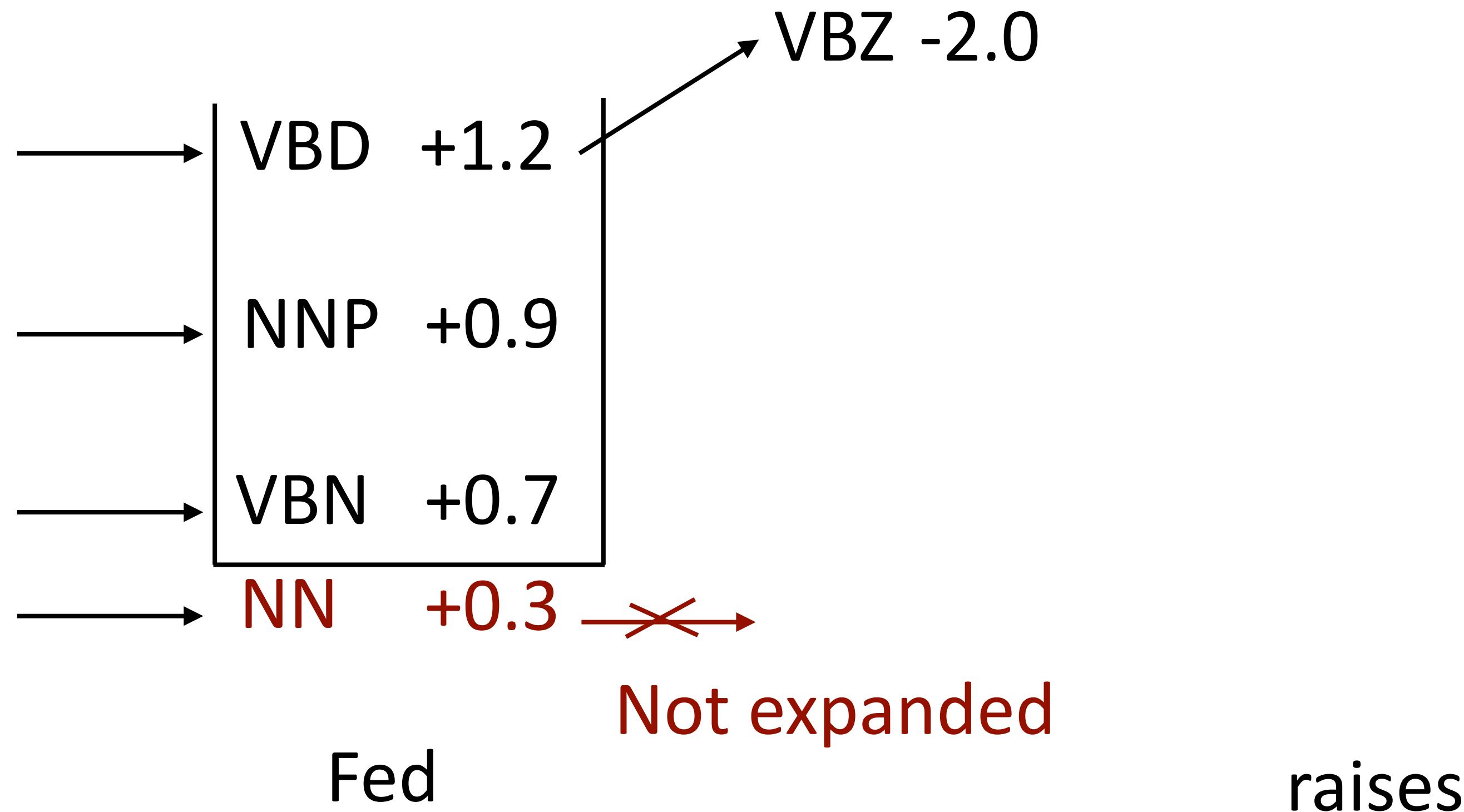
# Beam Search

- ▶ Maintain a beam of  $k$  plausible states at the current timestep
  - ▶ Expand all states, only keep  $k$  top hypotheses at new timestep



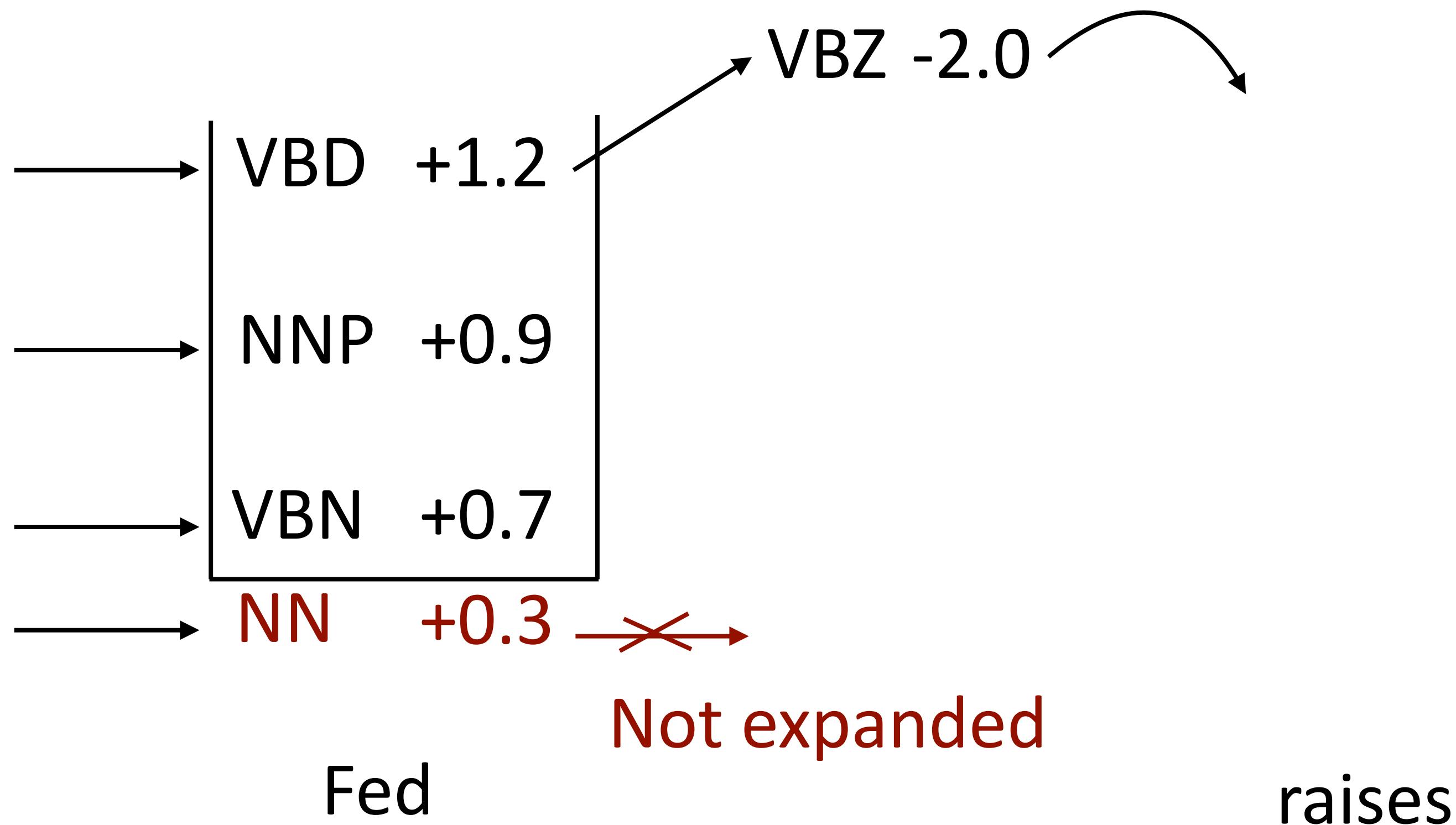
# Beam Search

- ▶ Maintain a beam of  $k$  plausible states at the current timestep
- ▶ Expand all states, only keep  $k$  top hypotheses at new timestep



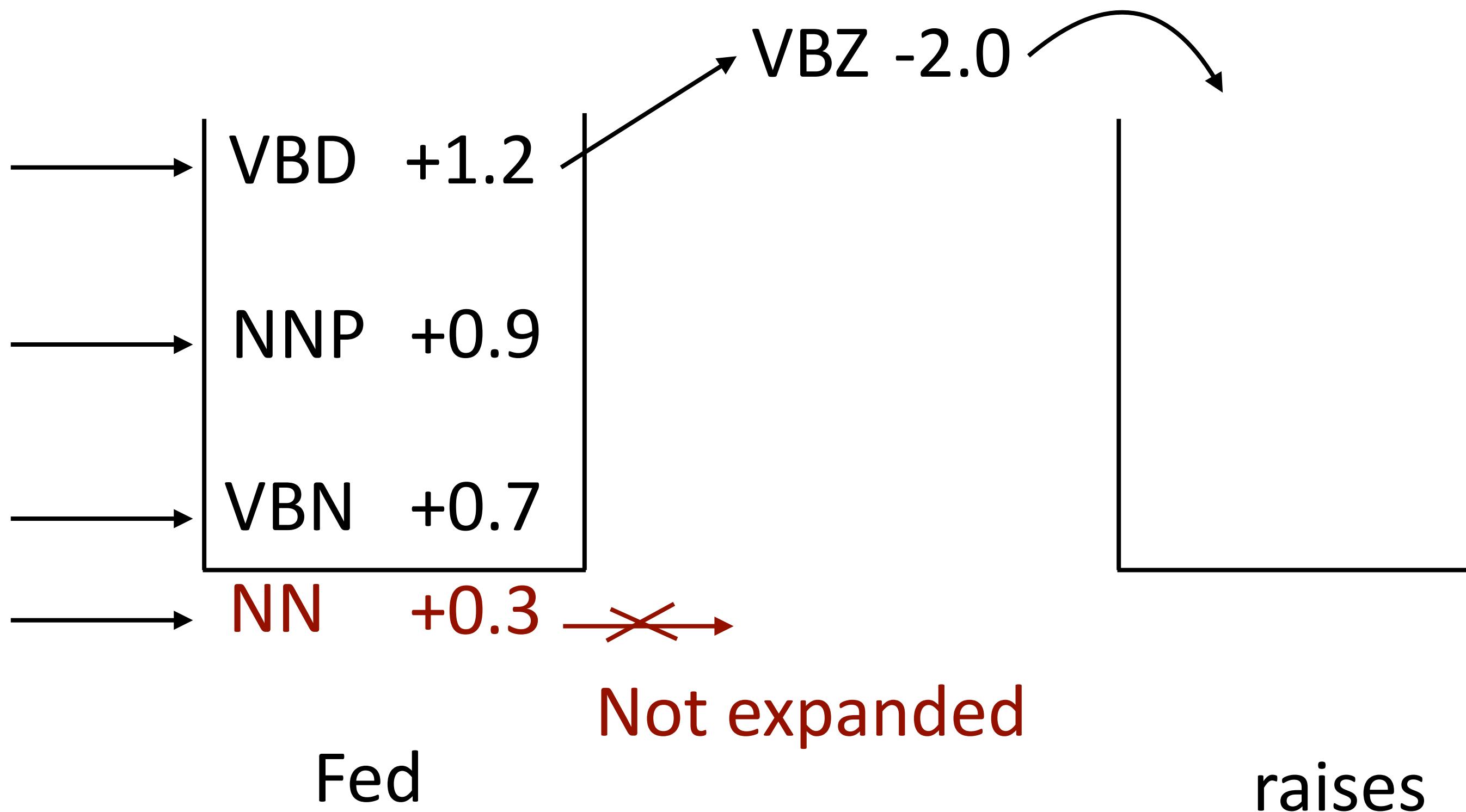
# Beam Search

- ▶ Maintain a beam of  $k$  plausible states at the current timestep
  - ▶ Expand all states, only keep  $k$  top hypotheses at new timestep



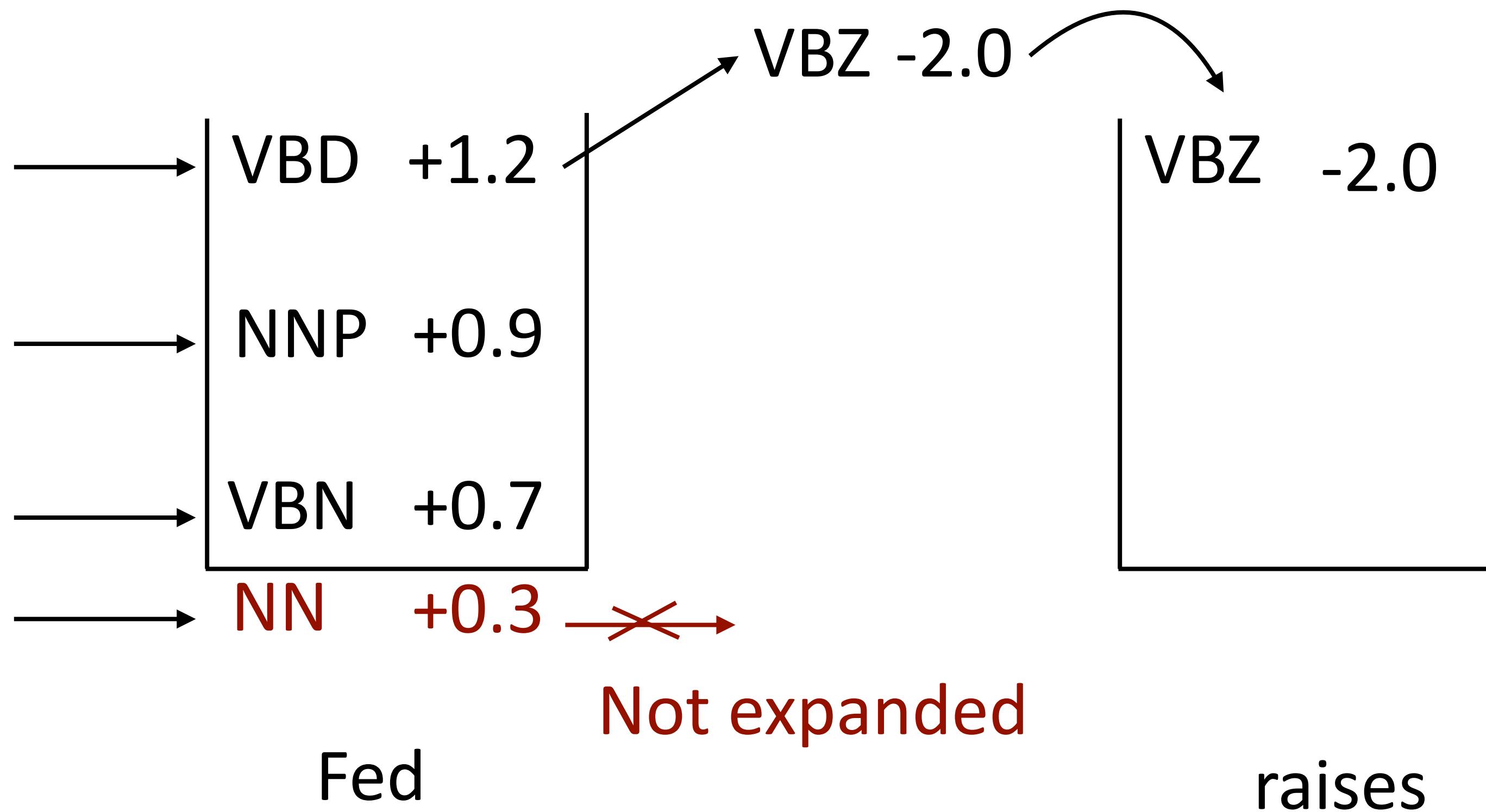
# Beam Search

- ▶ Maintain a beam of  $k$  plausible states at the current timestep
- ▶ Expand all states, only keep  $k$  top hypotheses at new timestep



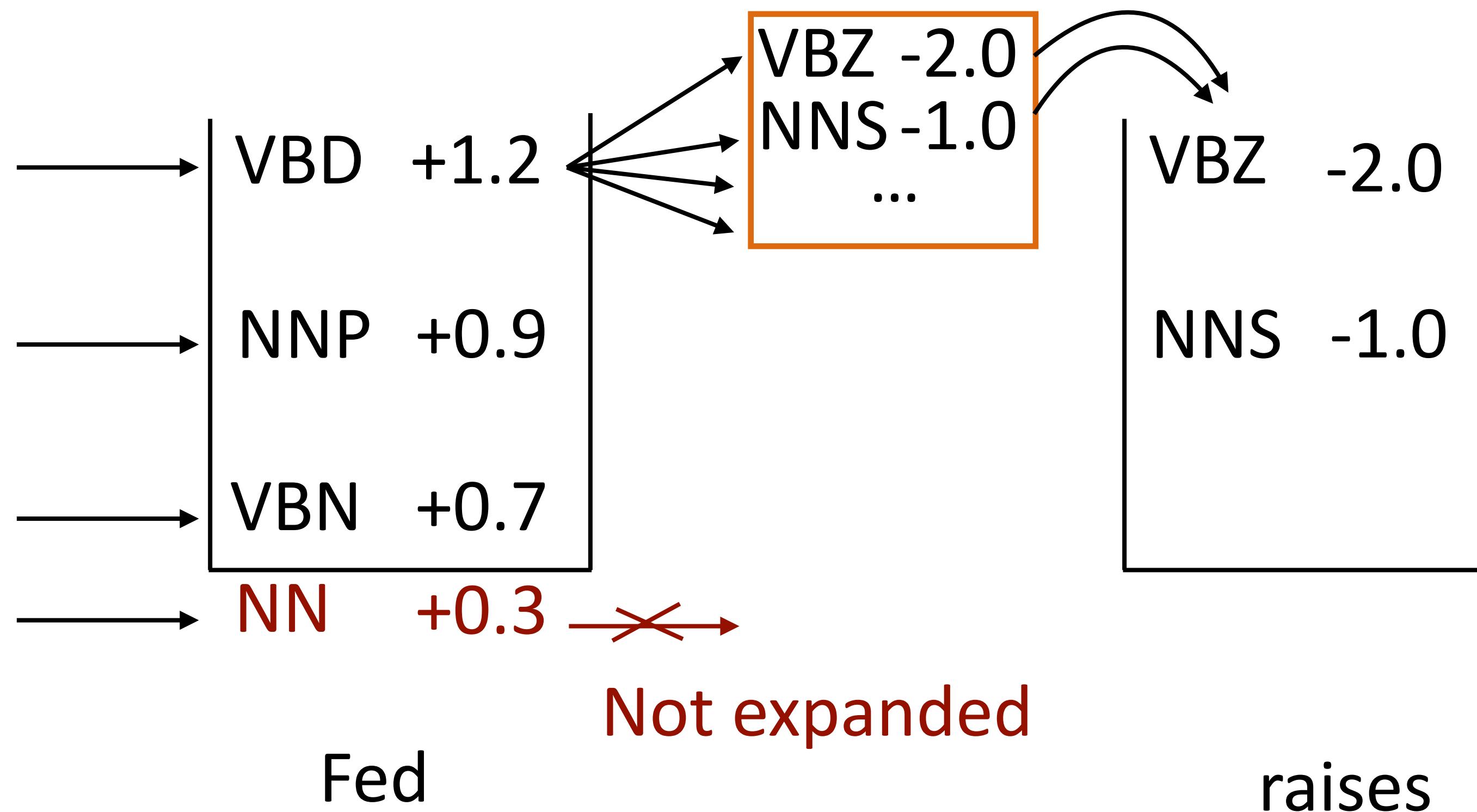
# Beam Search

- ▶ Maintain a beam of  $k$  plausible states at the current timestep
- ▶ Expand all states, only keep  $k$  top hypotheses at new timestep



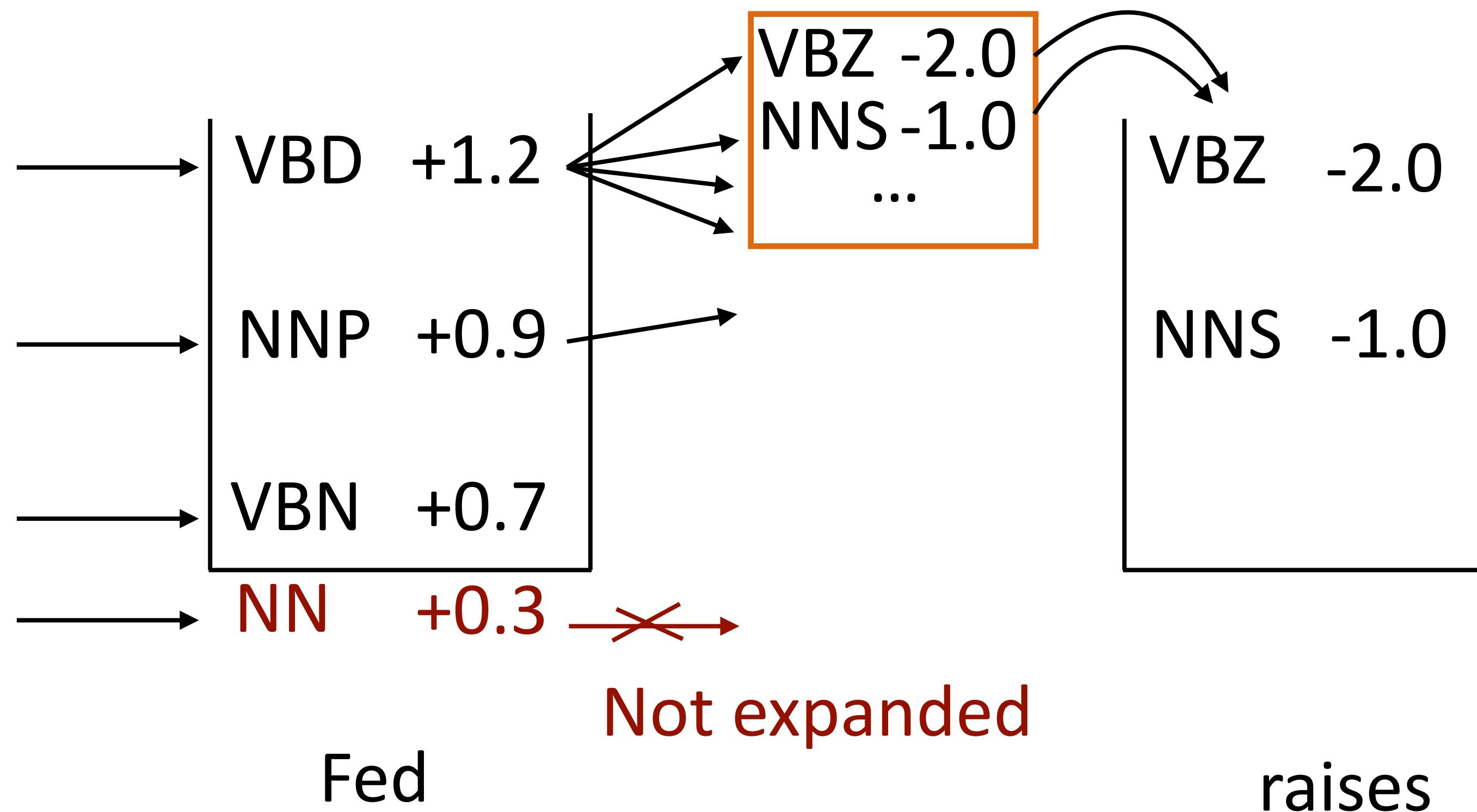
# Beam Search

- ▶ Maintain a beam of  $k$  plausible states at the current timestep
- ▶ Expand all states, only keep  $k$  top hypotheses at new timestep



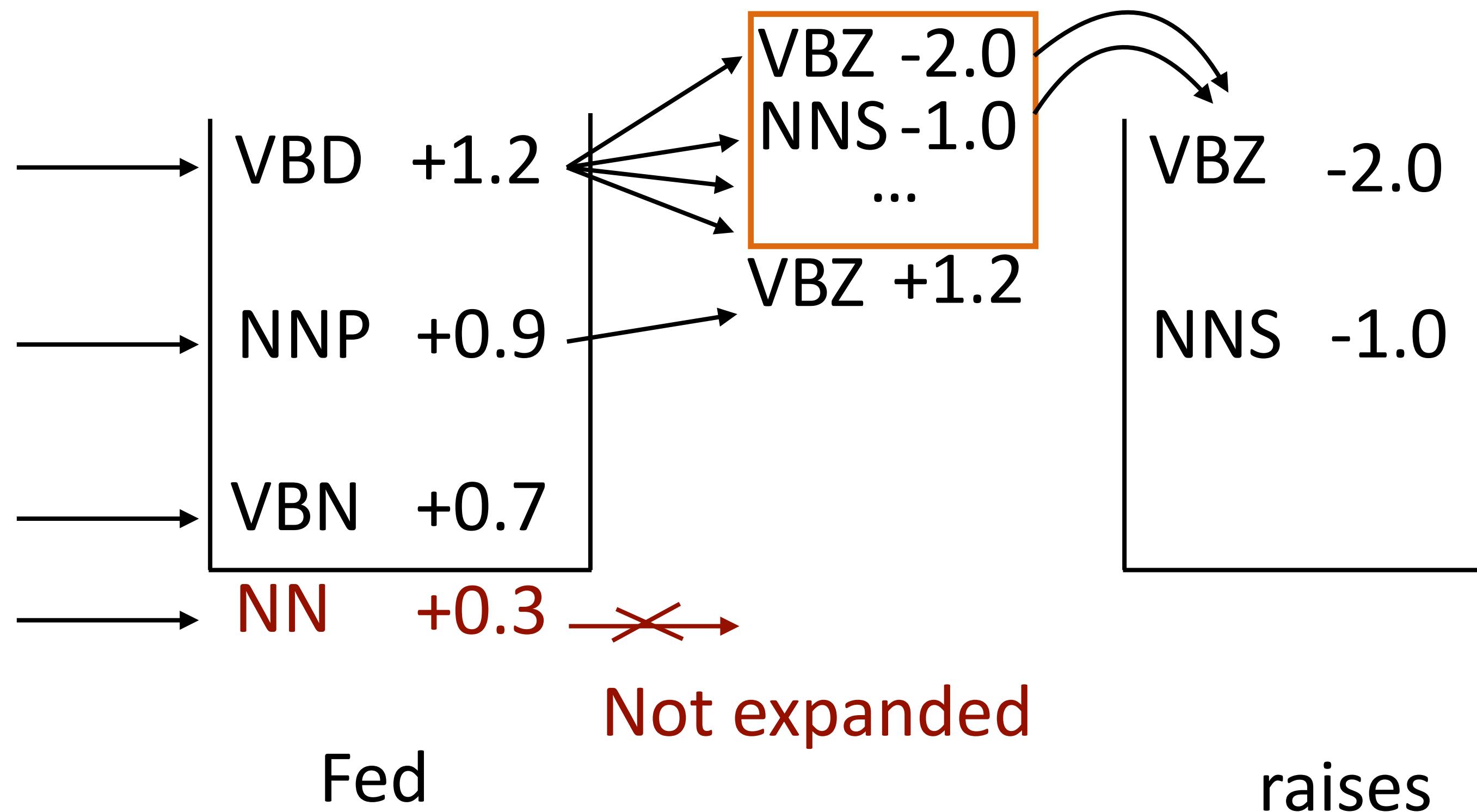
# Beam Search

- ▶ Maintain a beam of  $k$  plausible states at the current timestep
- ▶ Expand all states, only keep  $k$  top hypotheses at new timestep



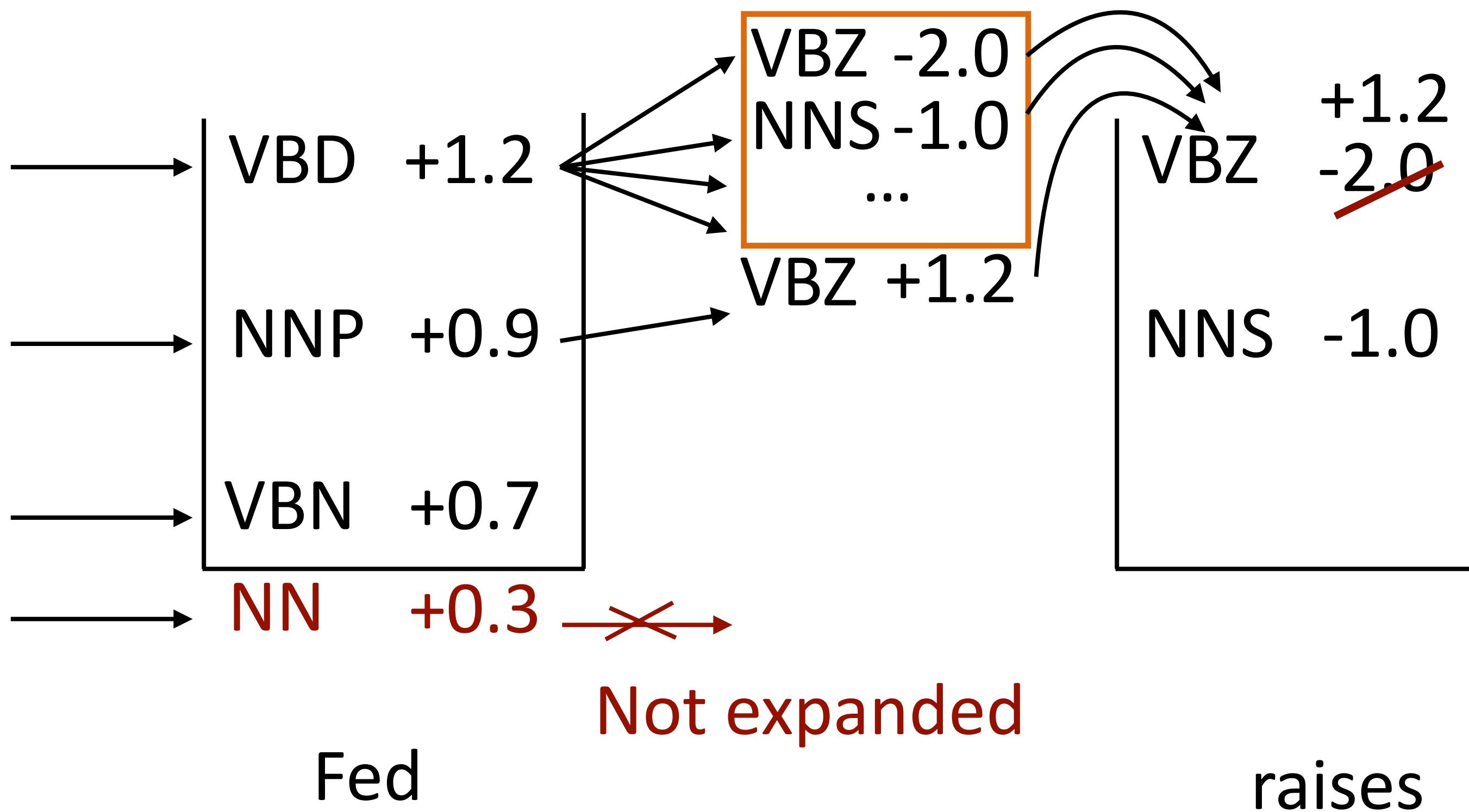
# Beam Search

- ▶ Maintain a beam of  $k$  plausible states at the current timestep
- ▶ Expand all states, only keep  $k$  top hypotheses at new timestep



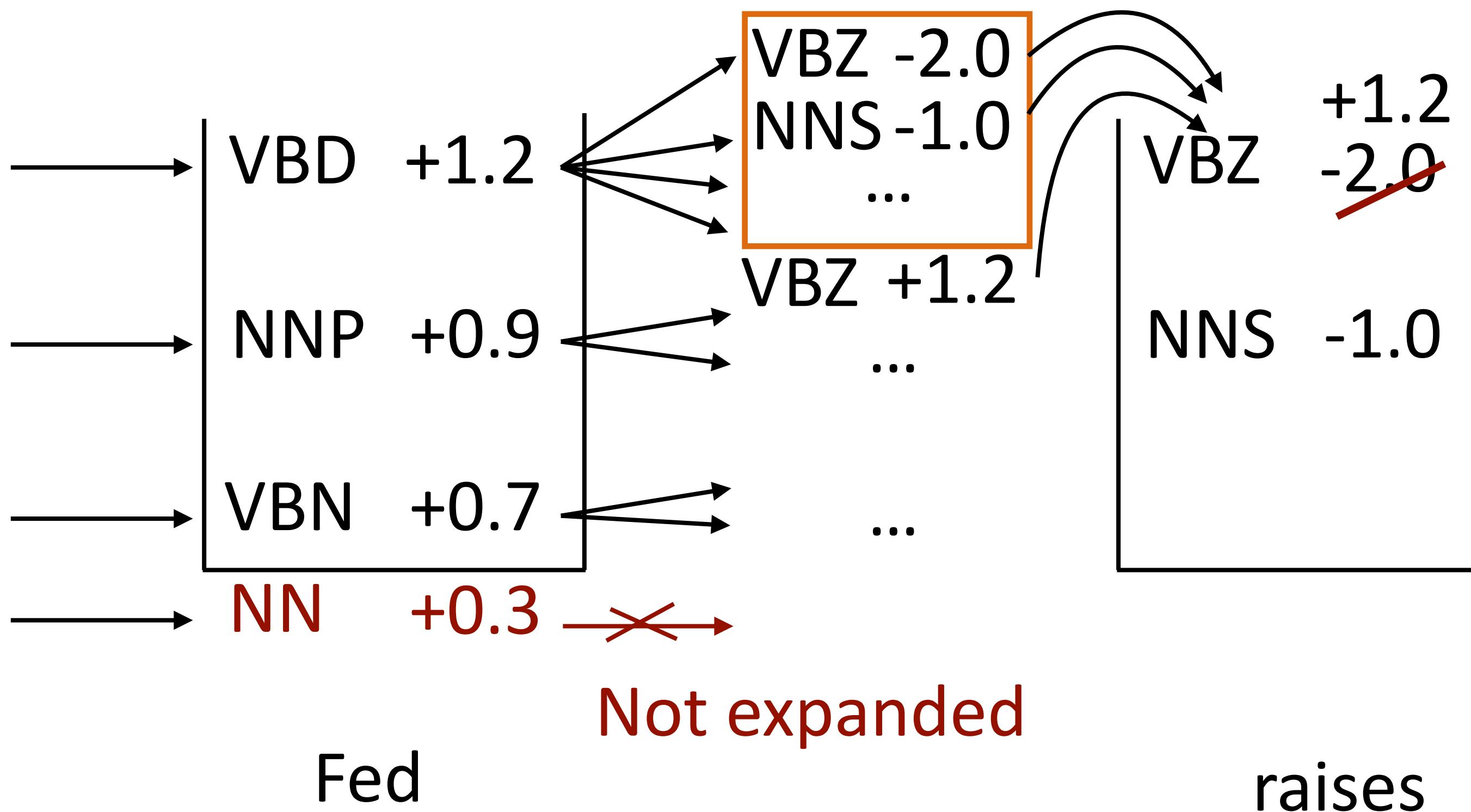
# Beam Search

- ▶ Maintain a beam of  $k$  plausible states at the current timestep
- ▶ Expand all states, only keep  $k$  top hypotheses at new timestep



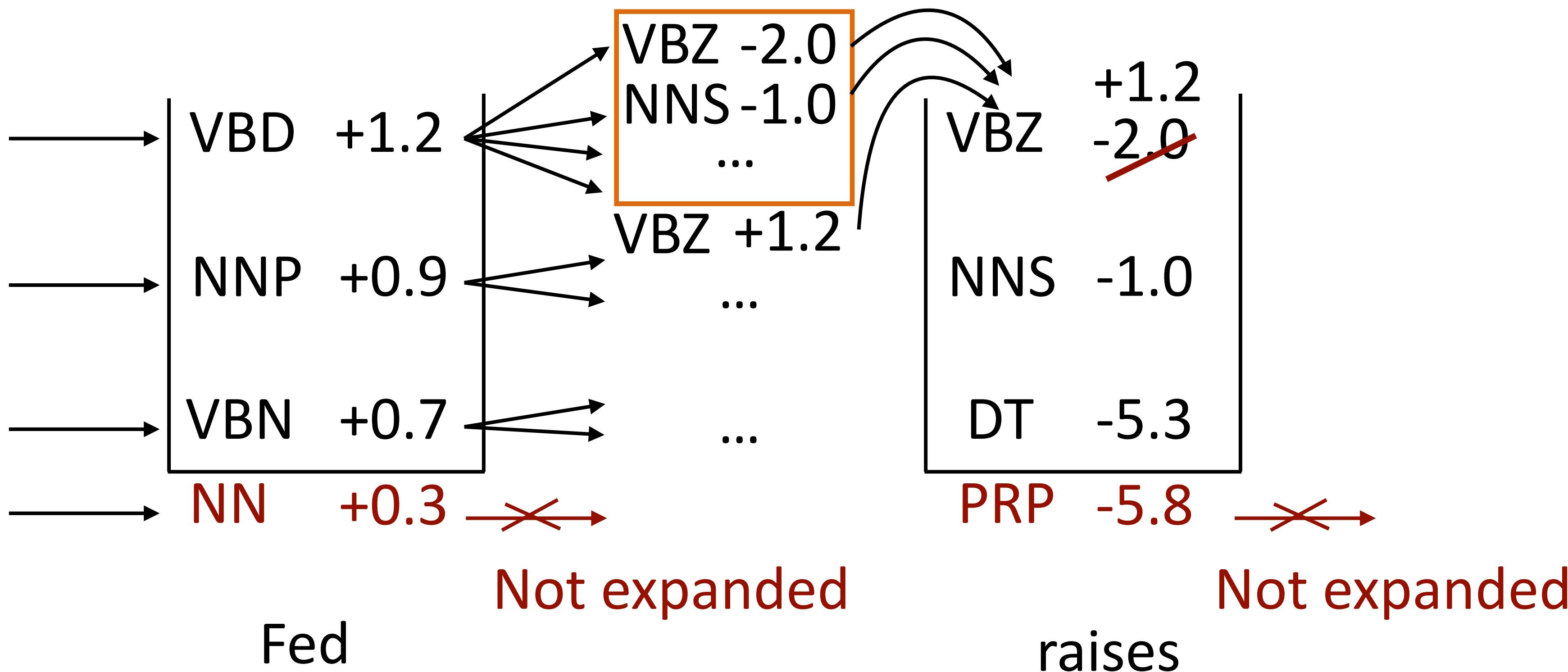
# Beam Search

- ▶ Maintain a beam of  $k$  plausible states at the current timestep
- ▶ Expand all states, only keep  $k$  top hypotheses at new timestep



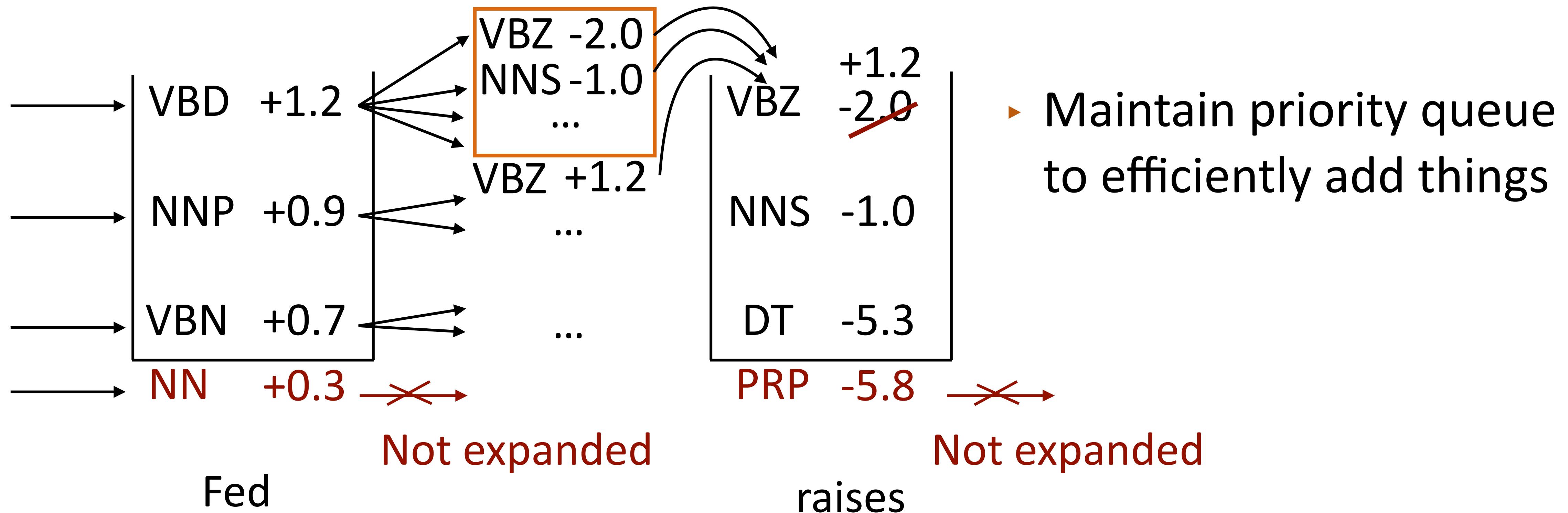
# Beam Search

- ▶ Maintain a beam of  $k$  plausible states at the current timestep
- ▶ Expand all states, only keep  $k$  top hypotheses at new timestep



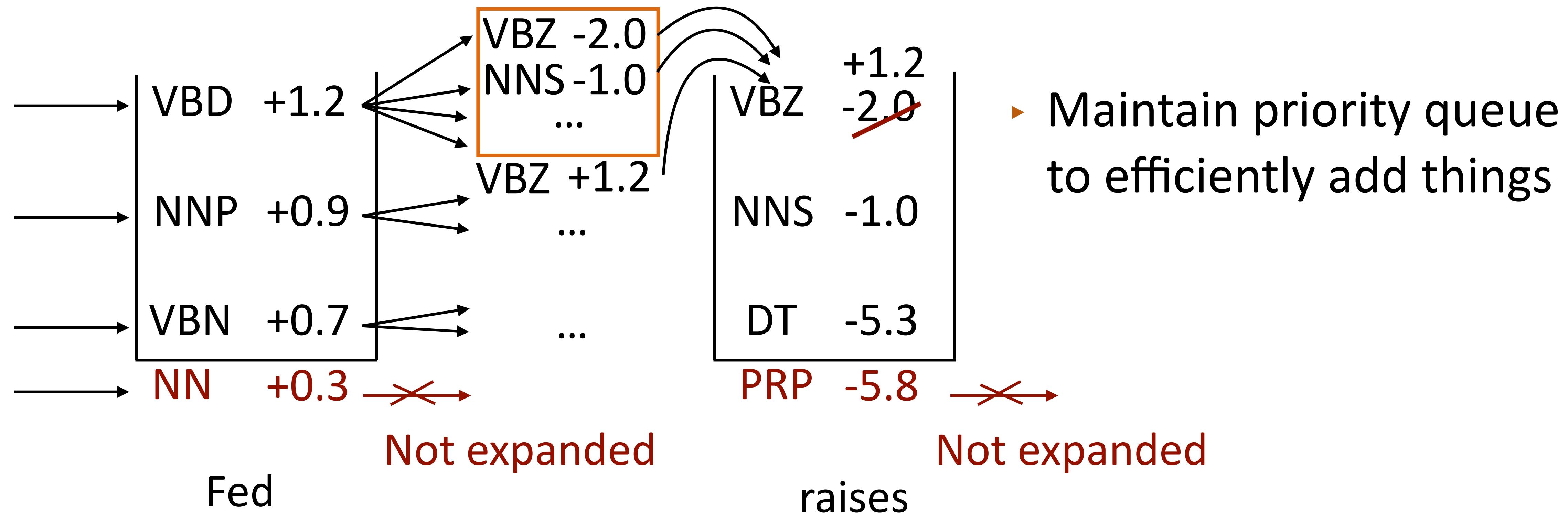
# Beam Search

- ▶ Maintain a beam of  $k$  plausible states at the current timestep
- ▶ Expand all states, only keep  $k$  top hypotheses at new timestep



# Beam Search

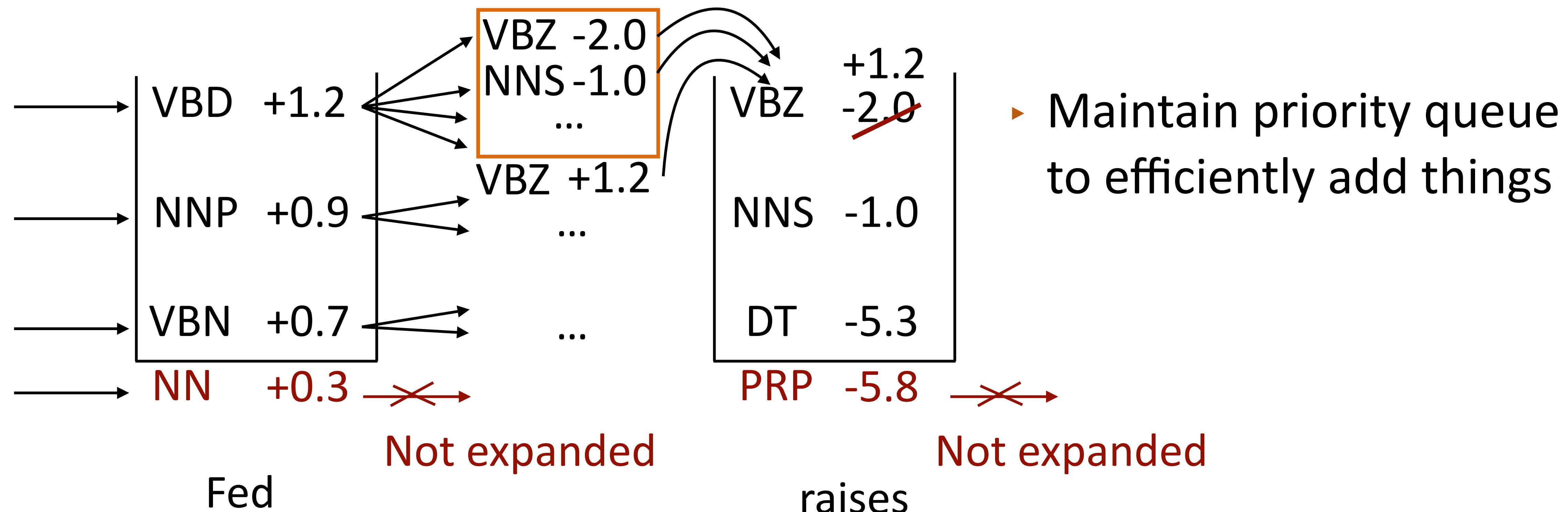
- ▶ Maintain a beam of  $k$  plausible states at the current timestep
- ▶ Expand all states, only keep  $k$  top hypotheses at new timestep



- ▶ Beam size of  $k$ , time complexity

# Beam Search

- ▶ Maintain a beam of  $k$  plausible states at the current timestep
- ▶ Expand all states, only keep  $k$  top hypotheses at new timestep



- ▶ Beam size of  $k$ , time complexity  $O(nks \log(ks))$

# How good is beam search?

---

# How good is beam search?

---

- ▶  $k=1$ : greedy search

# How good is beam search?

---

- ▶  $k=1$ : greedy search
- ▶ Choosing beam size:

# How good is beam search?

---

- ▶  $k=1$ : greedy search
- ▶ Choosing beam size:
  - ▶ 2 is usually better than 1

# How good is beam search?

---

- ▶  $k=1$ : greedy search
- ▶ Choosing beam size:
  - ▶ 2 is usually better than 1
  - ▶ Usually don't use larger than 50

# How good is beam search?

---

- ▶  $k=1$ : greedy search
- ▶ Choosing beam size:
  - ▶ 2 is usually better than 1
  - ▶ Usually don't use larger than 50
  - ▶ Depends on problem structure

# How good is beam search?

---

- ▶  $k=1$ : greedy search
- ▶ Choosing beam size:
  - ▶ 2 is usually better than 1
  - ▶ Usually don't use larger than 50
  - ▶ Depends on problem structure
- ▶ If beam search is much faster than computing full sums, can use structured perceptron instead of CRFs
- ▶ Very similar to structured SVM