

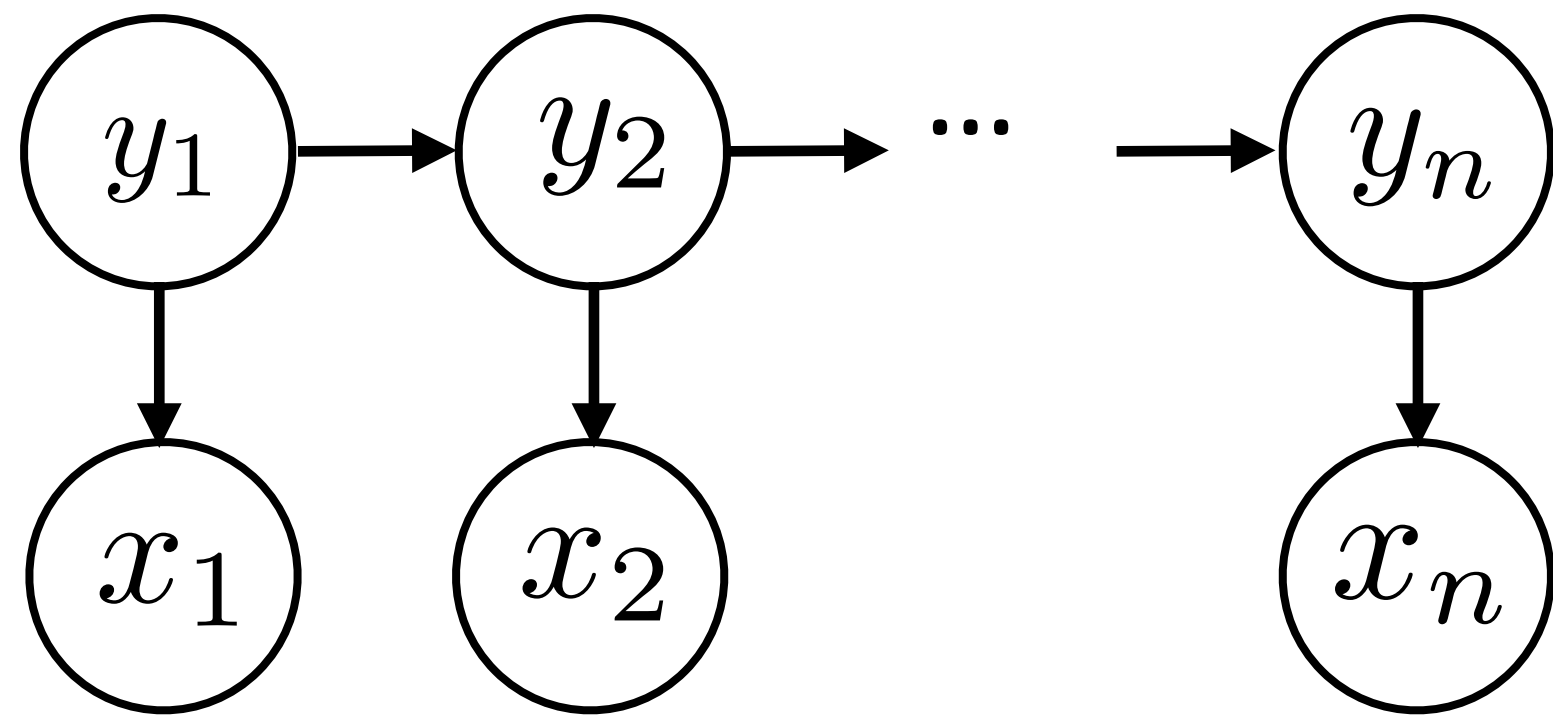
Sequence Models II

Wei Xu

(many slides from Greg Durrett)

HMMs (Recap)

- 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: $\operatorname{argmax}_{\mathbf{y}} P(\mathbf{y} | \mathbf{x}) = \operatorname{argmax}_{\mathbf{y}} \frac{P(\mathbf{y}, \mathbf{x})}{\cancel{P(\mathbf{x})}}$
- Viterbi: $\operatorname{score}_i(s) = \max_{y_{i-1}} P(s | y_{i-1}) P(x_i | s) \operatorname{score}_{i-1}(y_{i-1})$



Andrew Viterbi, 1967

Forward-Backward Algorithm (Recap)

- ▶ What did Viterbi compute? $P(\mathbf{y}_{\max}|\mathbf{x}) = \max_{y_1, \dots, y_n} P(\mathbf{y}|\mathbf{x})$
- ▶ In addition to finding the best path, we may want to compute marginal probabilities of paths $P(y_i = s|\mathbf{x})$

$$P(y_i = s|\mathbf{x}) = \sum_{y_1, \dots, y_{i-1}, y_{i+1}, \dots, y_n} P(\mathbf{y}|\mathbf{x})$$

- ▶ Can compute marginals with dynamic programming as well using an algorithm called forward-backward

This Lecture

- ▶ Named entity recognition (NER)
- ▶ CRFs: model (+features for NER), inference, learning
- ▶ Reading: Eisenstein Chapter 7 & 8.3

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

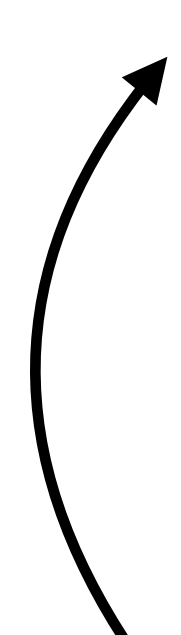
LOC

ORG

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

Where we're going

- ▶ Flexible discriminative model for tagging tasks that can use arbitrary features of the input. Similar to logistic regression, but *structured*

B-PER I-PER
 *Barack Obama* will travel to *Hangzhou* today for the *G20* meeting .

Curr_word=Barack & **Label=B-PER**

Next_word=Obama & **Label=B-PER**

Curr_word_starts_with_capital=True & **Label=B-PER**

Posn_in_sentence=1st & **Label=B-PER**

Label=B-PER & Next-Label = I-PER

...

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

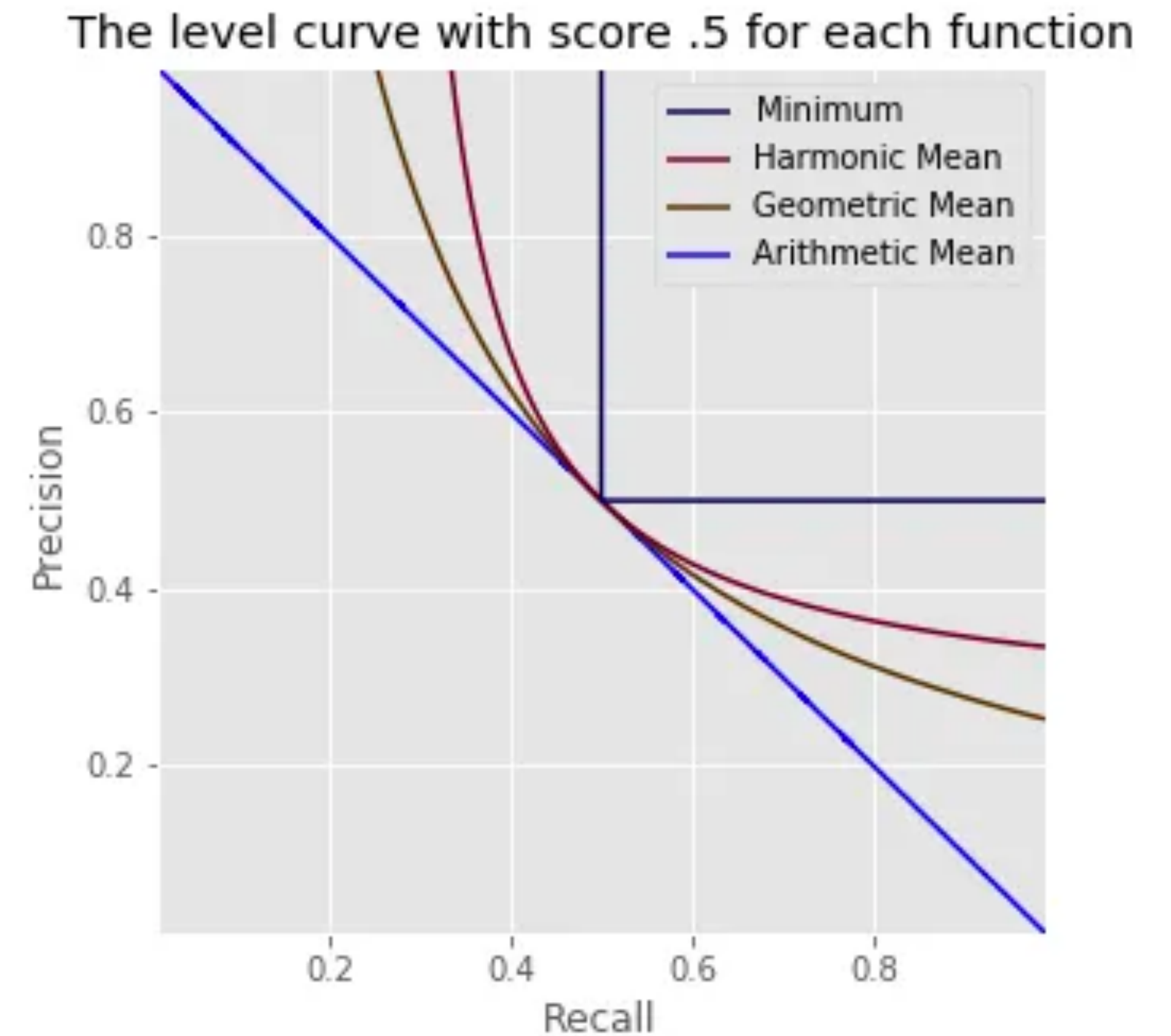
LOC

ORG

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

Harmonic Mean (Math Review)

- ▶ Arithmetic Mean = $(P + R) / 2$
- ▶ Geometric Mean = $\sqrt{P \times R}$
- ▶ Harmonic Mean = $2 \times P \times R / (P + R)$



Personal Disclosure Detection

Category	#Spans	Avg Len	Example
<i>Demographic Attributes</i>			
LOCATION	525	5.70±3.85	I live in the UK and a diagnosis is really expensive, even with health insurance
AGE	308	2.93±1.72	I am a 23-year-old who is currently going through the last leg of undergraduate school
RELATIONSHIP STATUS	287	6.72±5.97	My partner has not helped at all, and I'm bed ridden now
AGE/GENDER	248	1.42±0.71	For some context, I (20F), still live with my parents
PET	192	6.93±7.31	Hi, I have two musk turtles and have never had any health problems before at all
APPEARANCE	173	6.96±6.25	Same here. I am 6'2. No one can sit behind me.
HUSBAND/BF	148	6.89 ±7.24	My husband and I vote for different parties
WIFE/GF	144	5.24±4.42	My gf and I applied, we're new but fairly active!
GENDER	110	3.28±3.10	Am I insane? Eh. I'm just a girl who wants to look on the outside how I feel on the inside.
RACE/NATIONALITY	99	3.63±2.37	As Italian I hope tonight you will won the world cup
SEXUAL ORIENTATION	58	6.52±7.47	I'm a straight man but I do wanna say this
NAME	21	3.81±3.48	Hello guys, my name is xxx and I love travelling
CONTACT	14	5.69±3.56	xxx is my ig
<i>Personal Experiences</i>			
HEALTH	783	10.36±9.78	I am pretty sure I have autism, but I don't want to get an official diagnosis.
FAMILY	543	9.27±8.73	My little brother (9M) is my pride and joy
OCCUPATION	428	8.90±6.60	I'm a motorcycle tourer (by profession), but when I'm off the saddle I'm mostly bored
MENTAL HEALTH	285	16.86±16.28	I get asked this pretty regularly.. but I struggle with depression and ADHD
EDUCATION	229	9.92±7.71	Hi there, I got accepted to UCLA (IS), which I'm pumped about.
FINANCE	153	12.00±9.19	Yes. I was making \$68k a year and had around \$19k in debt

Table 1: Statistics and examples for each self-disclosure category in our dataset, sorted by decreasing frequency. Personal identifiable information are redacted as ‘xxx’ to be shown here.

Personal Disclosure Detection

Class (#Spans)	RoBERTa-large			DeBERTaV3-large			GPT-4		
	Span F_1	Partial F_1	Token F_1	Span F_1	Partial F_1	Token F_1	Span F_1	Partial F_1	Token F_1
AGE (35)	60.87	72.46	<u>87.04</u>	64.62	70.77	84.97	<i>65.71</i>	80.0	80.54
AGE&GENDER (17)	73.68	84.21	<u>81.63</u>	54.05	70.27	75.47	69.77	74.42	70.97
RACE/NATIONALITY (8)	60.0	60.0	64.94	82.35	82.35	<u>81.25</u>	70.59	70.59	71.64
GENDER (17)	61.11	61.11	56.47	72.73	72.73	61.73	57.14	57.14	<u>62.50</u>
LOCATION (41)	52.87	71.26	76.60	57.78	73.33	<u>83.61</u>	41.30	54.35	69.63
APPEARANCE (31)	<i>44.07</i>	64.41	<u>78.14</u>	35.48	67.74	76.27	12.77	42.55	54.71
WIFE/GF (30)	53.33	66.67	74.19	65.52	75.86	<u>78.36</u>	48.39	64.52	74.32
FINANCE (33)	35.82	68.66	76.57	<i>40.0</i>	71.43	<u>77.68</u>	31.17	54.55	69.49
OCCUPATION (44)	40.0	64.44	72.39	45.65	65.22	<u>75.08</u>	39.56	52.75	68.21
FAMILY (44)	<i>50.0</i>	58.70	66.52	35.29	49.02	<u>69.23</u>	46.60	58.25	64.68
HEALTH (40)	44.21	56.84	<u>76.54</u>	<i>45.10</i>	58.82	75.94	26.45	38.02	50.17
MENTAL HEALTH (46)	39.22	64.71	<u>81.18</u>	<i>40.0</i>	63.16	75.75	36.36	52.73	66.51
HUSBAND/BF (14)	68.75	75.0	<u>71.91</u>	64.71	70.59	71.11	55.17	68.97	68.49
EDUCATION (21)	55.32	68.09	85.23	65.38	69.23	<u>86.73</u>	42.55	51.06	67.37
PET (15)	46.15	46.15	56.88	48.28	55.17	<u>61.54</u>	48.28	48.28	56.25
RELATIONSHIP STATUS (31)	38.36	41.10	58.80	36.92	43.08	<u>60.11</u>	34.29	42.86	57.58
SEXUAL ORIENTATION (12)	28.57	76.19	68.0	25.0	58.33	66.0	<i>52.17</i>	69.57	<u>71.29</u>
AVERAGE	48.32	64.71	72.53	48.88	65.71	<u>74.17</u>	41.93	57.68	66.14

Table 8: Test performance per class for fine-tuned models and prompted GPT-4-0125-preview. *Italic* highlights the best model for each class on span F_1 . **Bold** on partial span F_1 . Underline on token F_1 .

CRFs

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

John Lafferty^{†*}
Andrew McCallum^{*†}
Fernando Pereira^{*‡}

LAFFERTY@CS.CMU.EDU
MCCALLUM@WHIZBANG.COM
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

Abstract

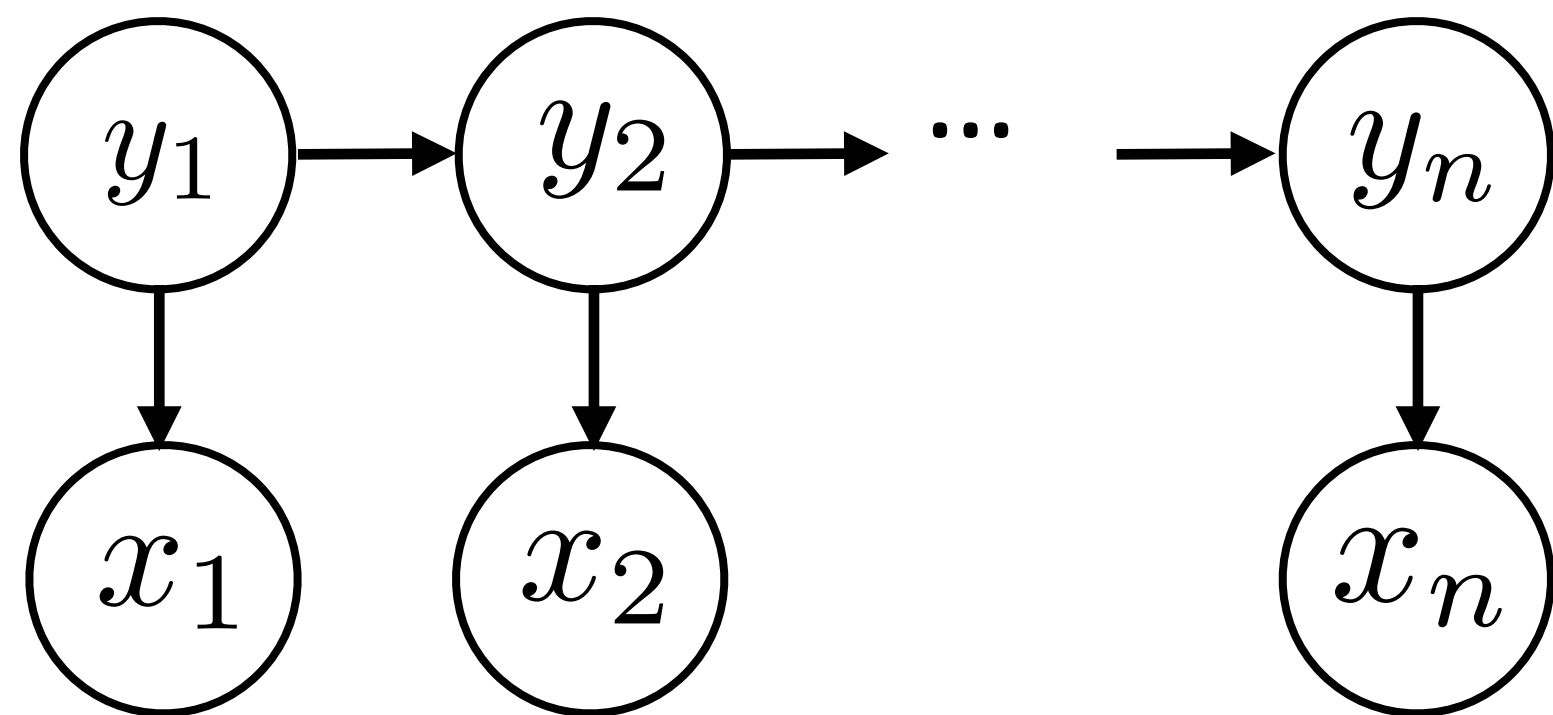
We present *conditional random fields*, a framework for building probabilistic models to segment and label sequence data. Conditional random fields offer several advantages over hidden Markov models and stochastic grammars for such tasks, including the ability to relax strong independence assumptions made in those models. Conditional random fields also avoid a fundamental limitation of maximum entropy Markov models (MEMMs) and other discriminative Markov models based on directed graphical models, which can be biased towards states

minimize the joint likelihood of training examples. To define a joint probability over observation and label sequences, a generative model needs to enumerate all possible observation sequences, typically requiring a representation in which observations are task-appropriate atomic entities, such as words or nucleotides. In particular, it is not practical to represent multiple interacting features or long-range dependencies of the observations, since the inference problem for such models is intractable.

This difficulty is one of the main motivations for looking at conditional models as an alternative. A conditional model specifies the probabilities of possible label sequences given an observation sequence. Therefore, it does not expend

HMMs, Formally

- ▶ 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: $P(\mathbf{y}, \mathbf{x}) = P(y_1)P(x_1|y_1)P(y_2|y_1)P(x_2|y_2) \dots$
- ▶ 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 $\nearrow$ $\uparrow$ any real-valued scoring function of its arguments

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

▶ Special case: linear feature-based potentials $\phi_k(\mathbf{x}, \mathbf{y}) = w^\top f_k(\mathbf{x}, \mathbf{y})$

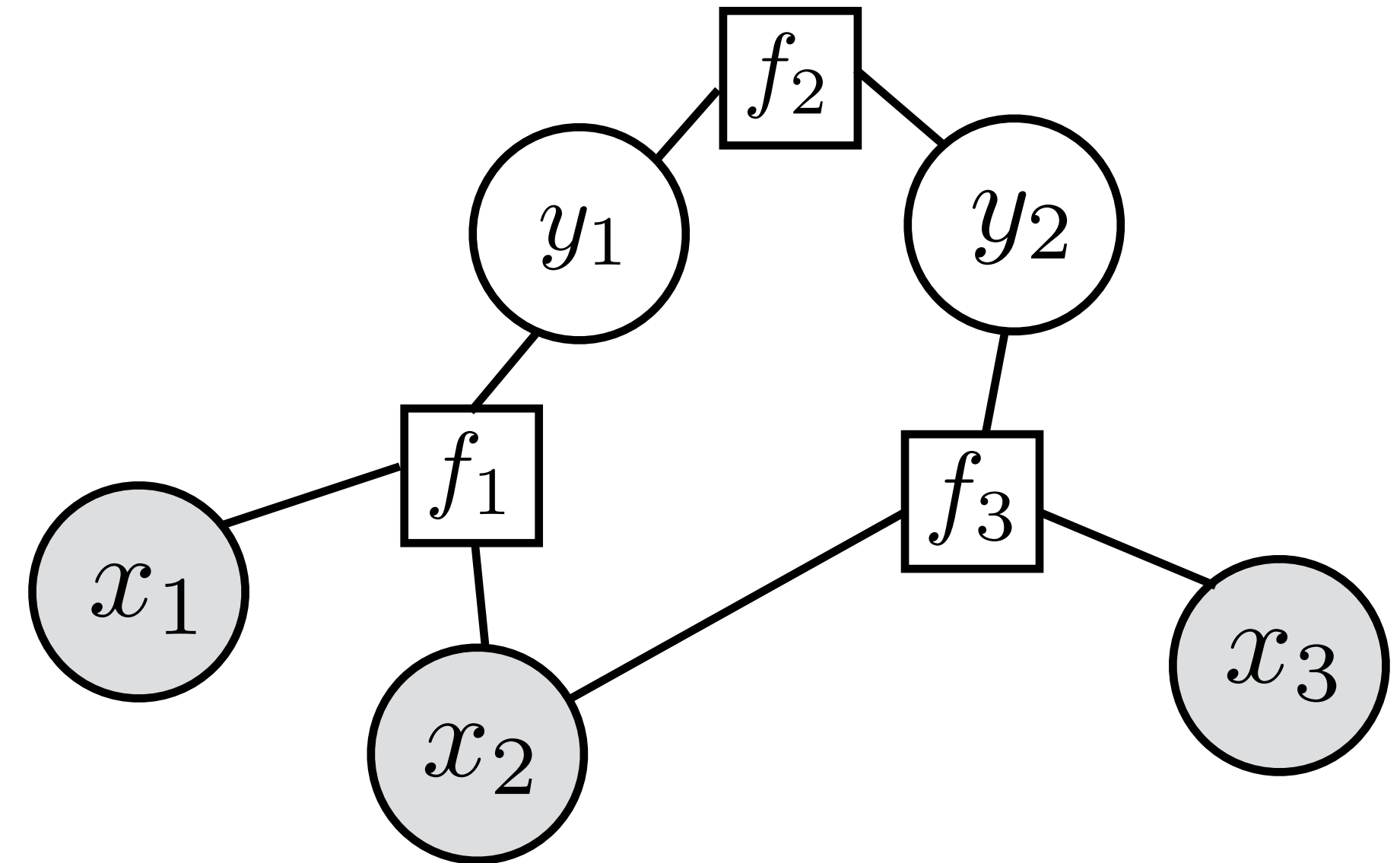
$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \exp \left(\sum_{k=1}^n w^\top f_k(\mathbf{x}, \mathbf{y}) \right)$$

▶ Looks like our single weight vector multiclass logistic regression model

HMMs vs. CRFs

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \exp \left(\sum_{k=1}^n w^\top f_k(\mathbf{x}, \mathbf{y}) \right)$$

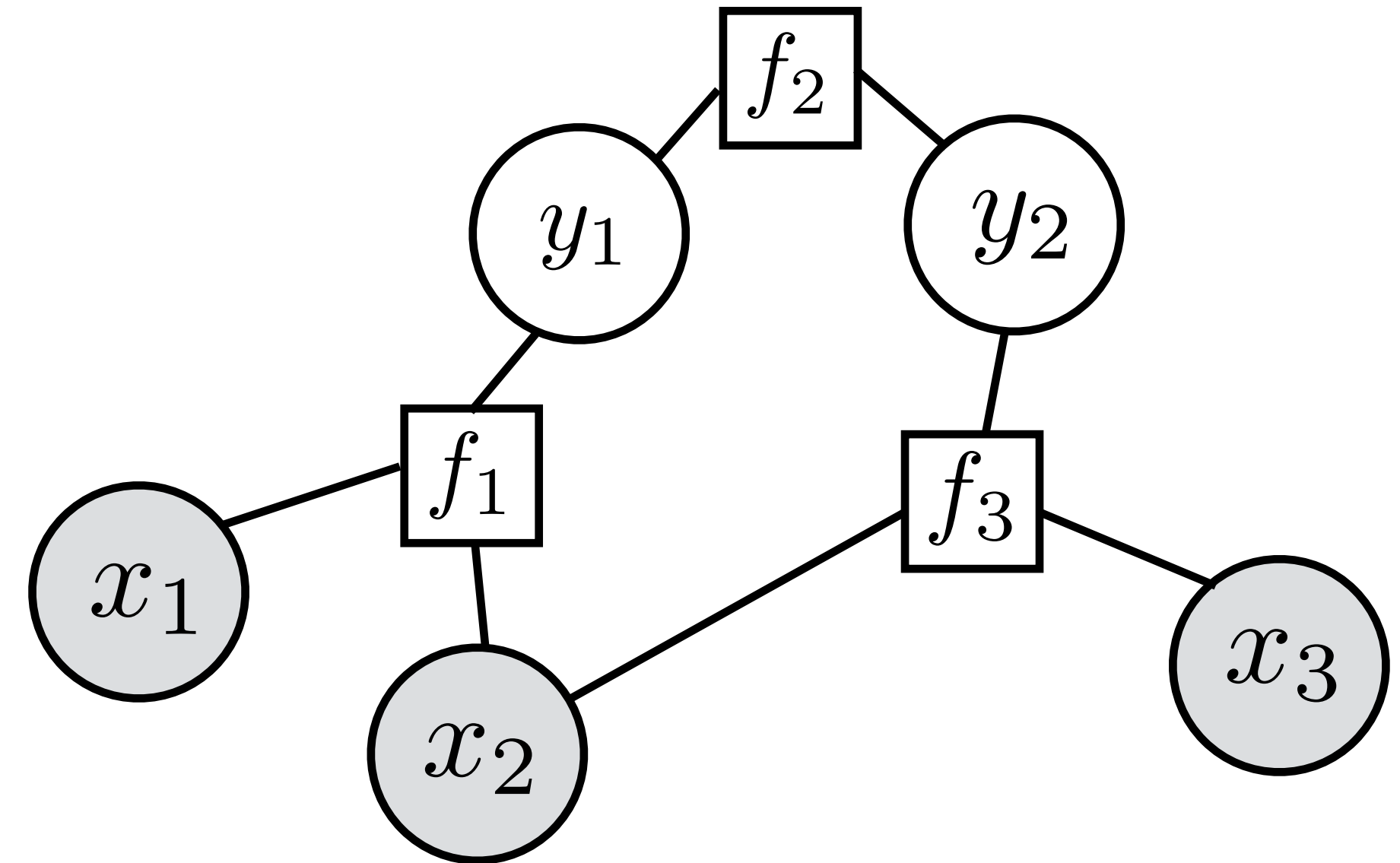
- Conditional model: $\mathbf{x}$'s are observed



- Each square represent a potential (feature function) and touches the $\mathbf{y}$'s and $\mathbf{x}$'s it depends on

HMMs vs. CRFs

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \exp \left(\sum_{k=1}^n w^\top f_k(\mathbf{x}, \mathbf{y}) \right)$$

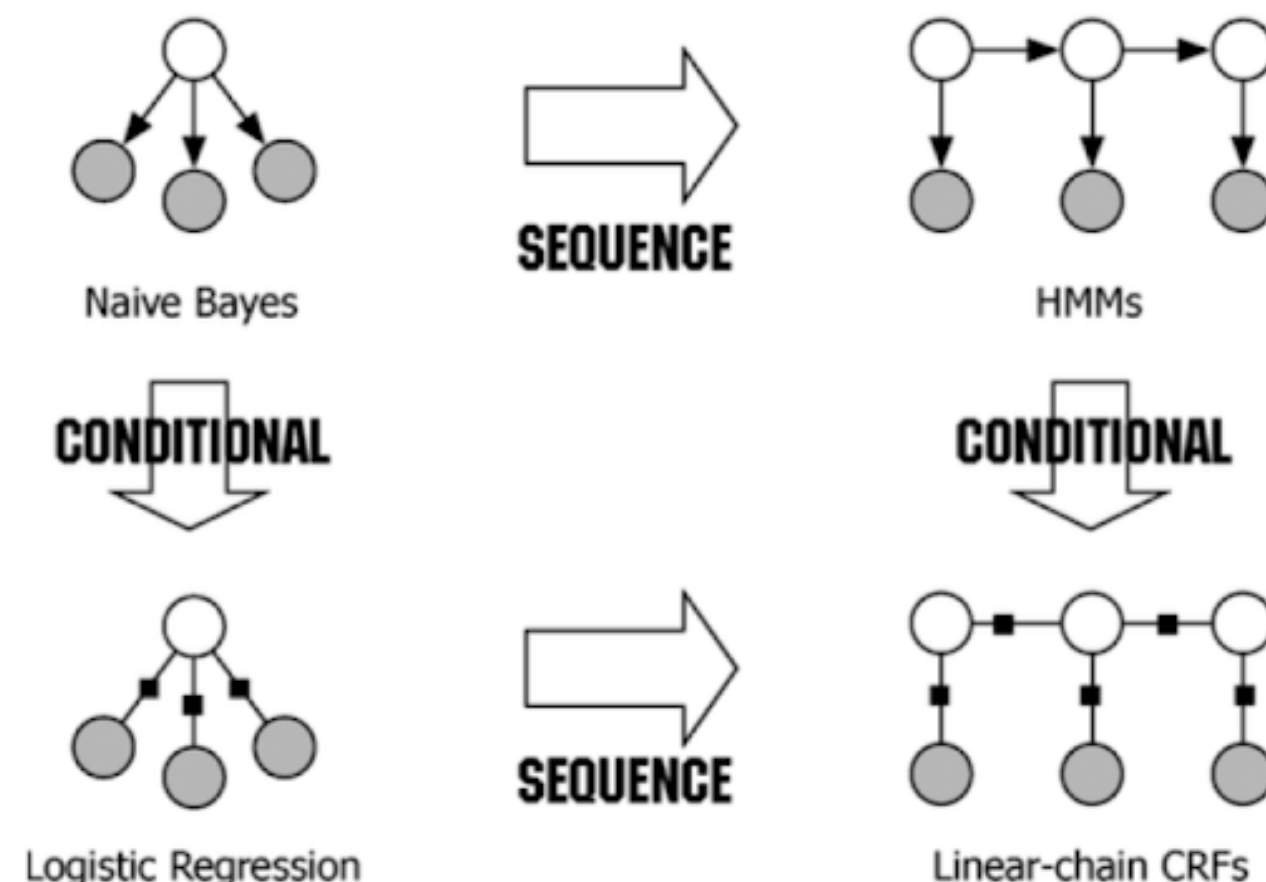


- Conditional model: x's are observed

- Naive Bayes : logistic regression :: HMMs : CRFs

local vs. global normalization <-> generative vs. discriminative

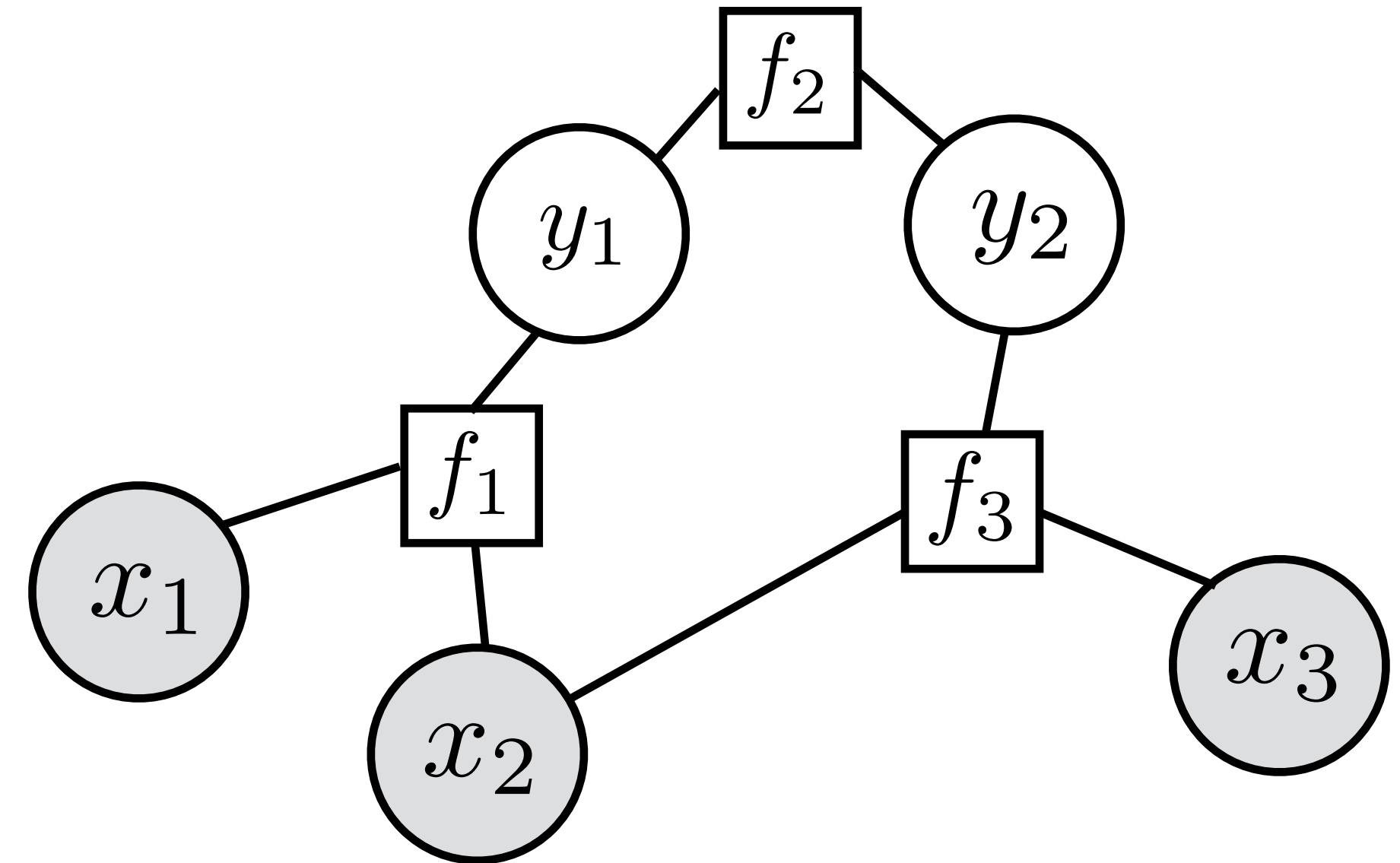
(locally normalized discriminative models do exist (MEMMs))



- Each square represent a potential (feature function) and touches the y's and x's it depends on

HMMs vs. CRFs

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \exp \left(\sum_{k=1}^n w^\top f_k(\mathbf{x}, \mathbf{y}) \right)$$



- ▶ Conditional model: $\mathbf{x}$'s are observed
- ▶ Naive Bayes : logistic regression :: HMMs : CRFs
local vs. global normalization $\leftrightarrow$ generative vs. discriminative
(locally normalized discriminative models do exist (MEMMs))
- ▶ HMMs: in the standard setup, emissions consider one word at a time
- ▶ CRFs: features over many words simultaneously, non-independent features (e.g., suffixes and prefixes), doesn't have to be a generative model

Problem with CRFs

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \exp \left(\sum_{k=1}^n w^\top f_k(\mathbf{x}, \mathbf{y}) \right)$$

- ▶ Normalizing constant

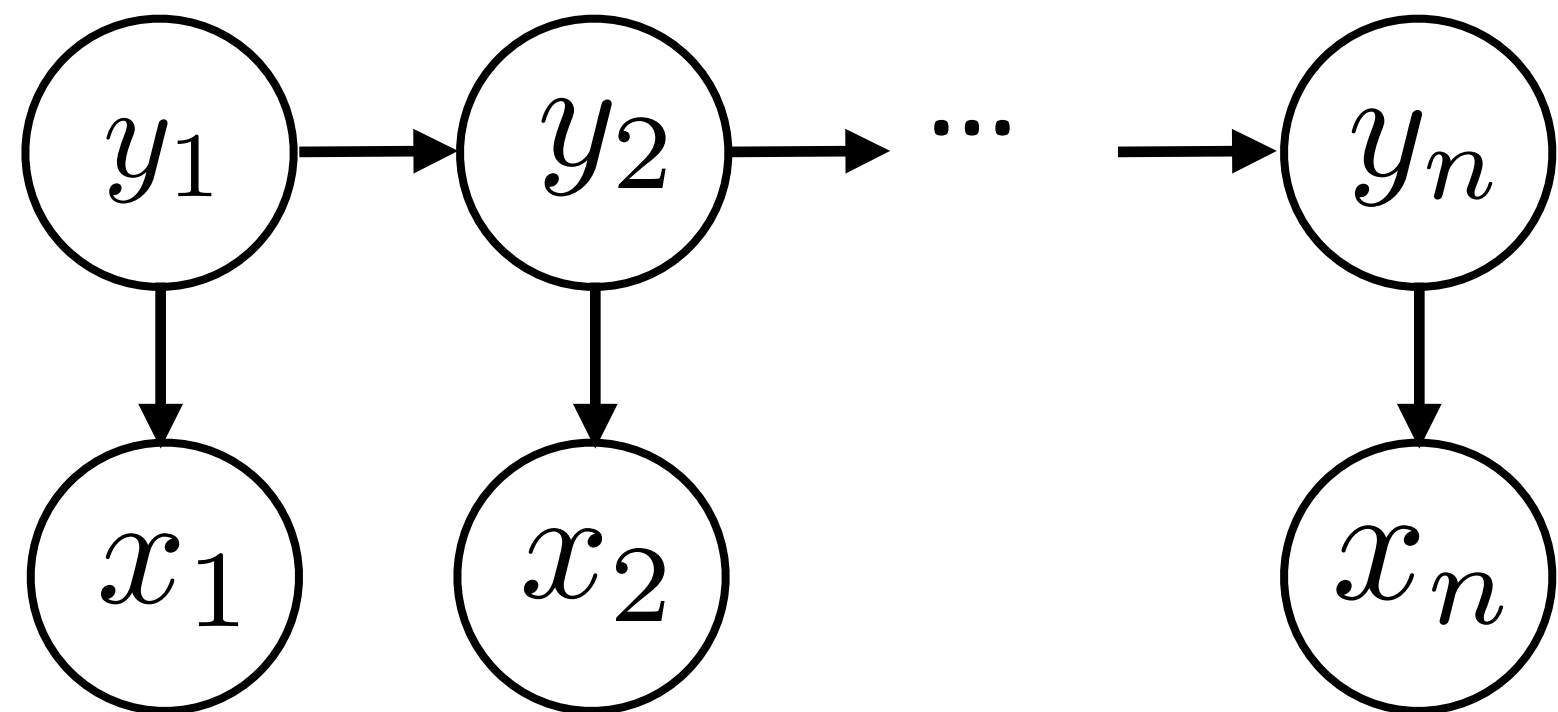
$$Z = \sum_{\mathbf{y}'} \exp \left(\sum_{k=1}^n w^\top f_k(\mathbf{x}, \mathbf{y}') \right)$$

- ▶ Inference: $\mathbf{y}_{\text{best}} = \operatorname{argmax}_{\mathbf{y}'} \exp \left(\sum_{k=1}^n w^\top f_k(\mathbf{x}, \mathbf{y}') \right)$

- ▶ If $\mathbf{y}$ consists of 5 variables with 30 values each, how expensive are these?
- ▶ Need to constrain the form of our CRFs to make it tractable

Sequential CRFs

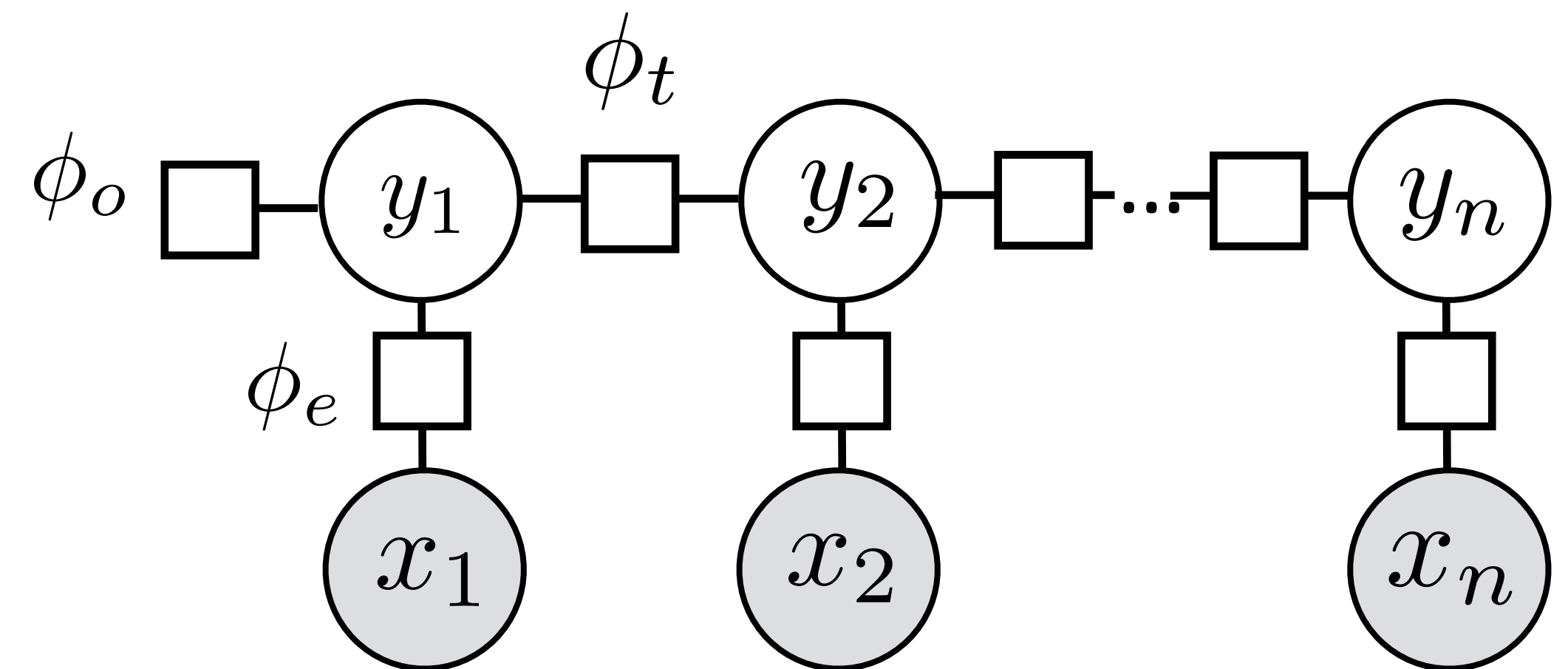
- ▶ HMMs: $P(\mathbf{y}, \mathbf{x}) = P(y_1)P(x_1|y_1)P(y_2|y_1)P(x_2|y_2) \dots$



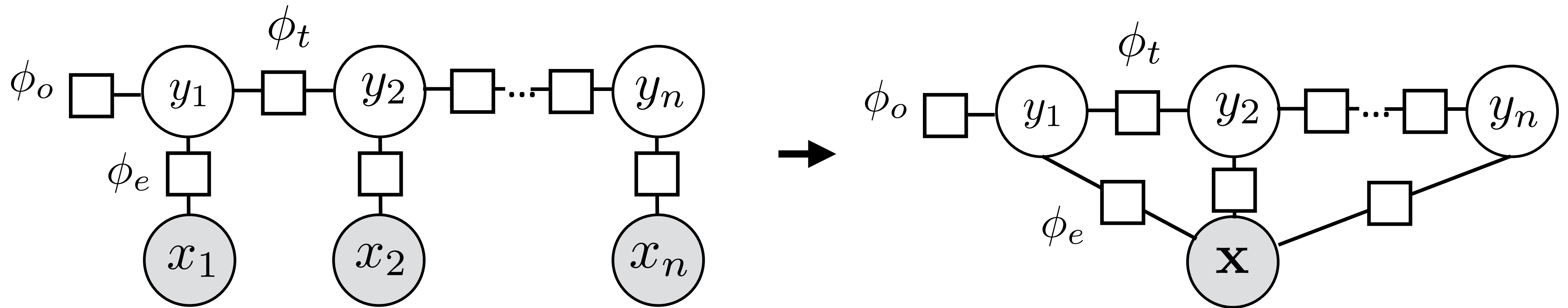
- ▶ CRFs:

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

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



Sequential CRFs

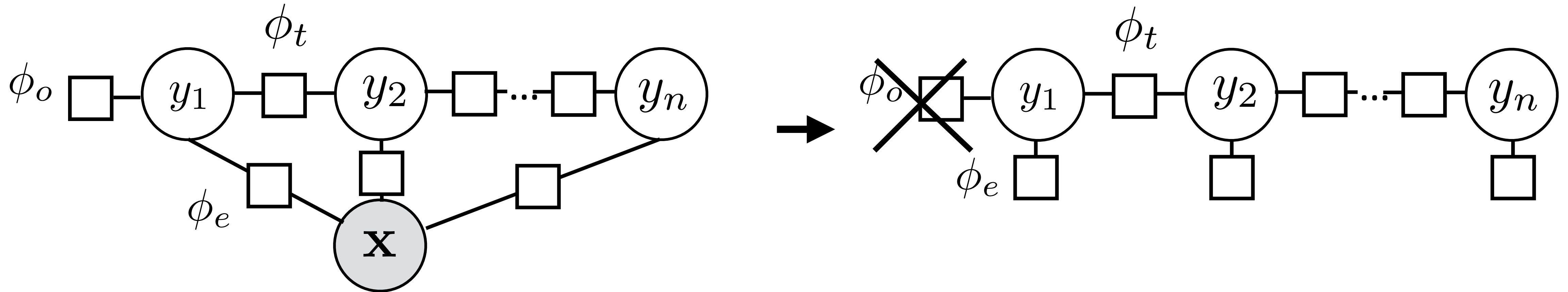


$$P(\mathbf{y}|\mathbf{x}) \propto \exp(\phi_o(y_1)) \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(x_i, y_i))$$

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

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

Sequential CRFs



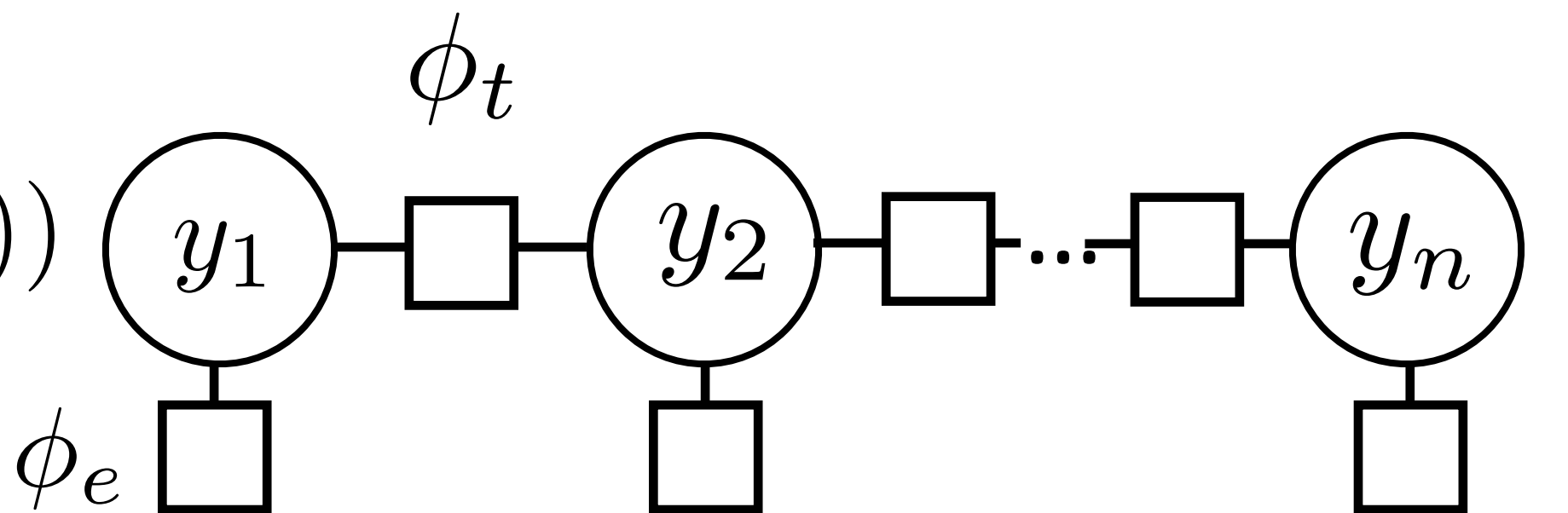
- ▶ Notation: omit $\mathbf{x}$ from the factor graph entirely (implicit)
- ▶ Don't include initial distribution, can bake into other factors

Sequential CRFs (aka linear-chain CRFs):

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

Features for NER

Feature Functions

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


The diagram illustrates a Markov chain structure. It consists of a sequence of circular nodes labeled $y_1, y_2, \dots, y_n$ connected by horizontal lines. Between each pair of consecutive circular nodes, there is a square node. Additionally, below each circular node y_i , there is another square node. The label ϕ_t is positioned above the square node between y_1 and y_2 . The label ϕ_e is positioned to the left of the square node below y_1 .

- ▶ This can be almost anything! (e.g., use a neural network to compute them)
Here we use linear functions of sparse features:

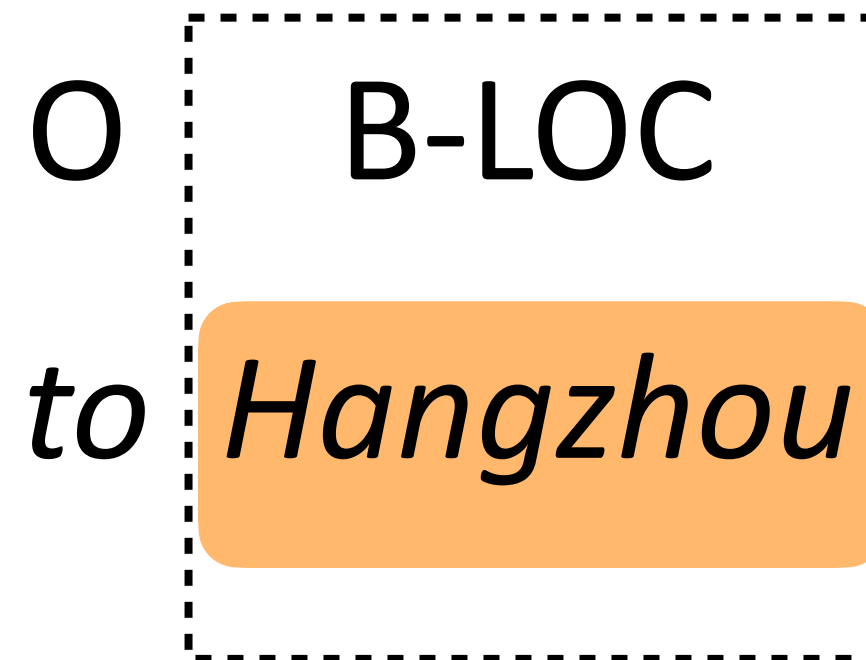
$$\phi_e(y_i, i, \mathbf{x}) = w^\top f_e(y_i, i, \mathbf{x}) \quad \phi_t(y_{i-1}, y_i) = w^\top f_t(y_{i-1}, y_i)$$

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

- ▶ 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 .

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

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

Emission 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

PER

Greg Abbott said

ORG

According to the New York Times...

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
 - ▶ POS Tags before/after (if we run a POS tagger first)
- ▶ Word clusters
- ▶ Gazetteers

Leicestershire

Boston

Apple released a new version...

According to the *New York Times*...

Inference and Learning in CRFs

CRFs in General (recap)

- ▶ CRFs: discriminative model with the following 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

- ▶ Our special case: linear feature-based potentials $\phi_k(\mathbf{x}, \mathbf{y}) = w^\top f_k(\mathbf{x}, \mathbf{y})$

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \exp \left(\sum_{k=1}^n w^\top f_k(\mathbf{x}, \mathbf{y}) \right)$$

- ▶ Problem: intractable inference in the general case! Computing Z requires an exponent sum

Linear-chain 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

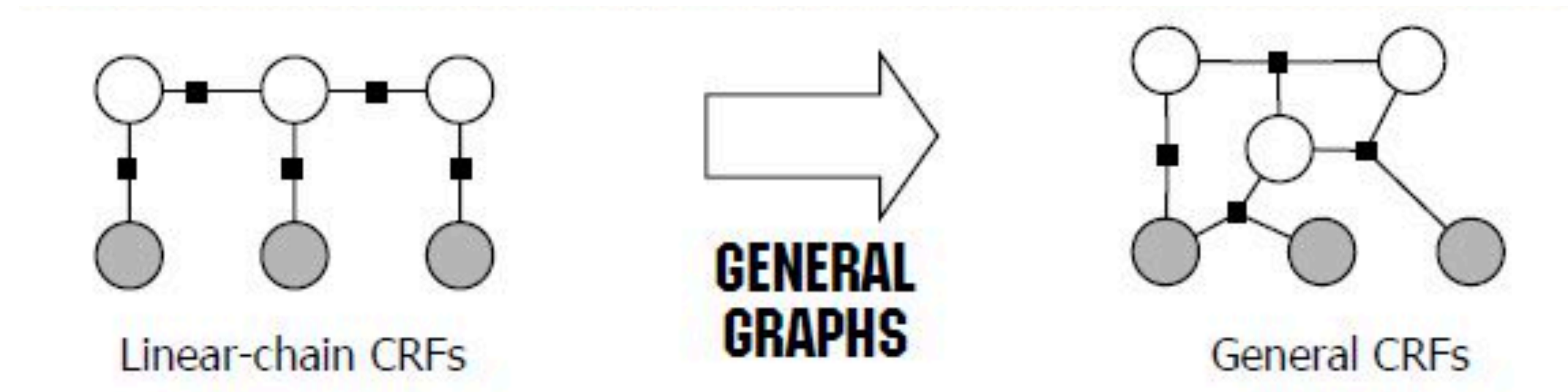
$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(y_i, i, \mathbf{x}))$$

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

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

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

Inference in General CRFs



- ▶ 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: $\operatorname{argmax} P(\mathbf{y}|\mathbf{x})$ from Viterbi
- Learning

Training CRFs

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

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

Training CRFs

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

- Let's focus on emission feature expectation

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

Forward-Backward Algorithm

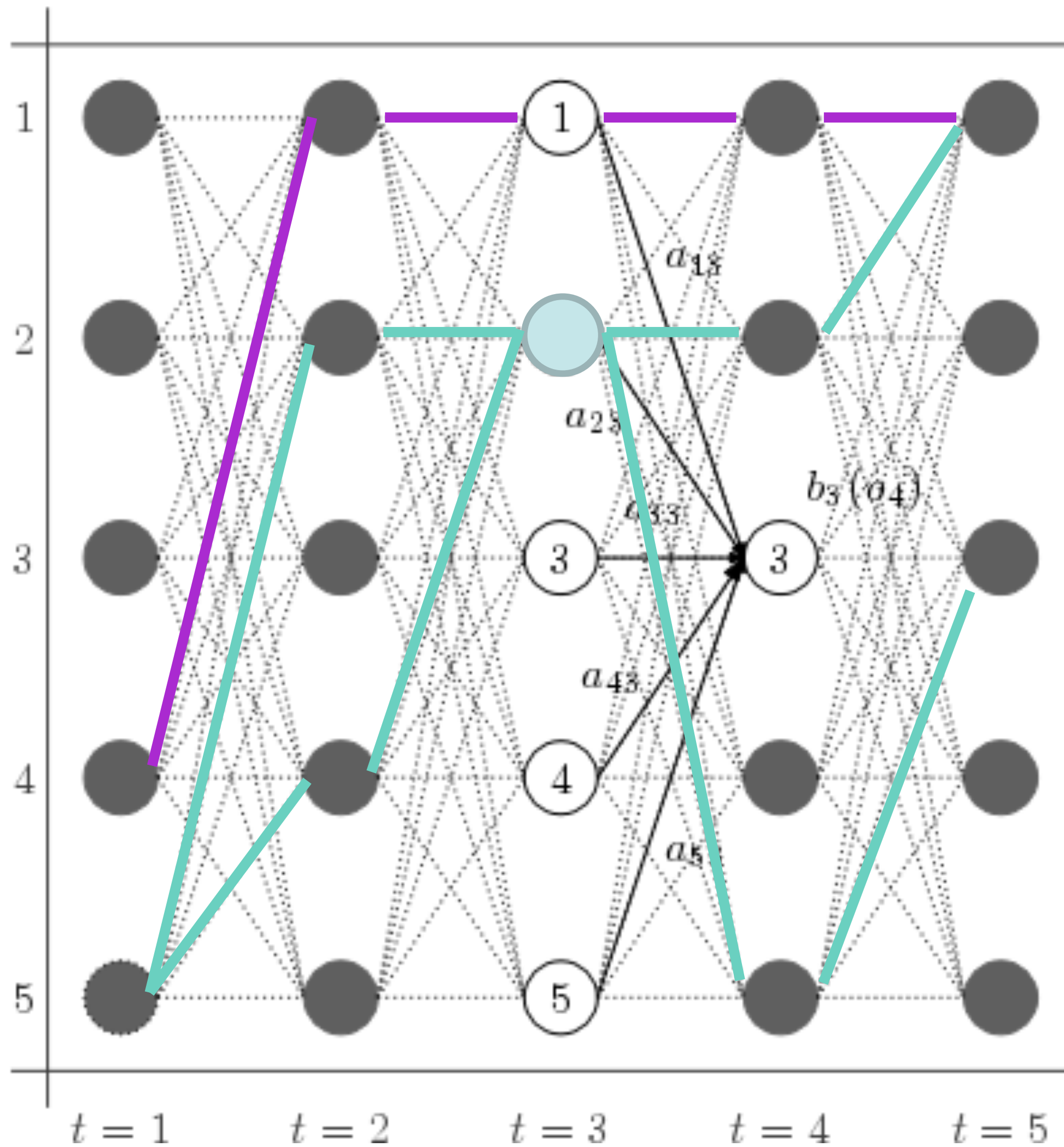
- ▶ How do we compute these marginals $P(y_i = s | \mathbf{x})$?

$$P(y_i = s | \mathbf{x}) = \sum_{y_1, \dots, y_{i-1}, y_{i+1}, \dots, y_n} P(\mathbf{y} | \mathbf{x})$$

- ▶ What did Viterbi compute? $P(\mathbf{y}_{\max} | \mathbf{x}) = \max_{y_1, \dots, y_n} P(\mathbf{y} | \mathbf{x})$

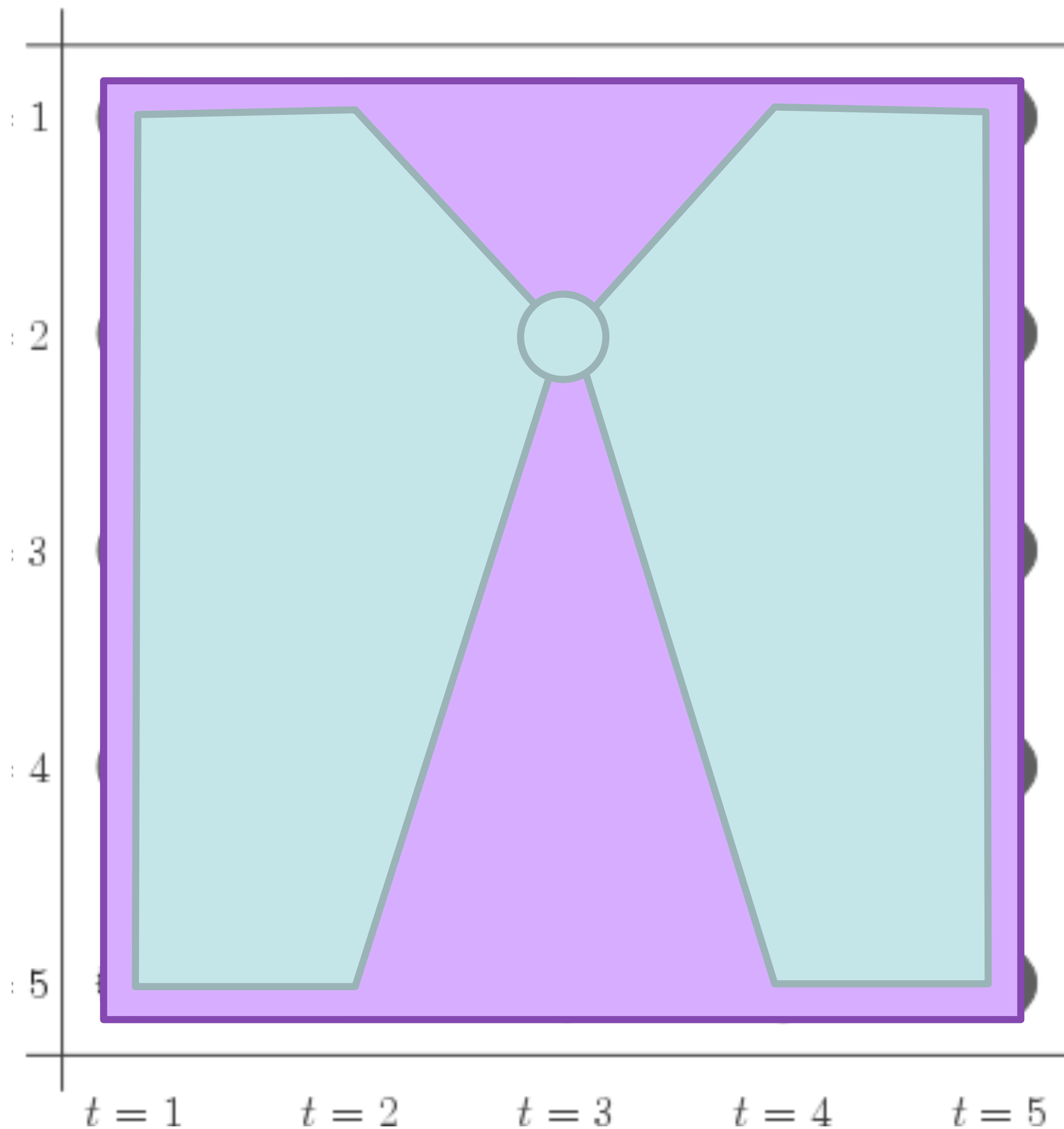
- ▶ Can compute marginals with dynamic programming as well using the forward-backward algorithm

Forward-Backward Algorithm



$$P(y_3 = 2 | \mathbf{x}) = \frac{\text{sum of all paths through state 2 at time 3}}{\text{sum of all paths}}$$

Forward-Backward Algorithm



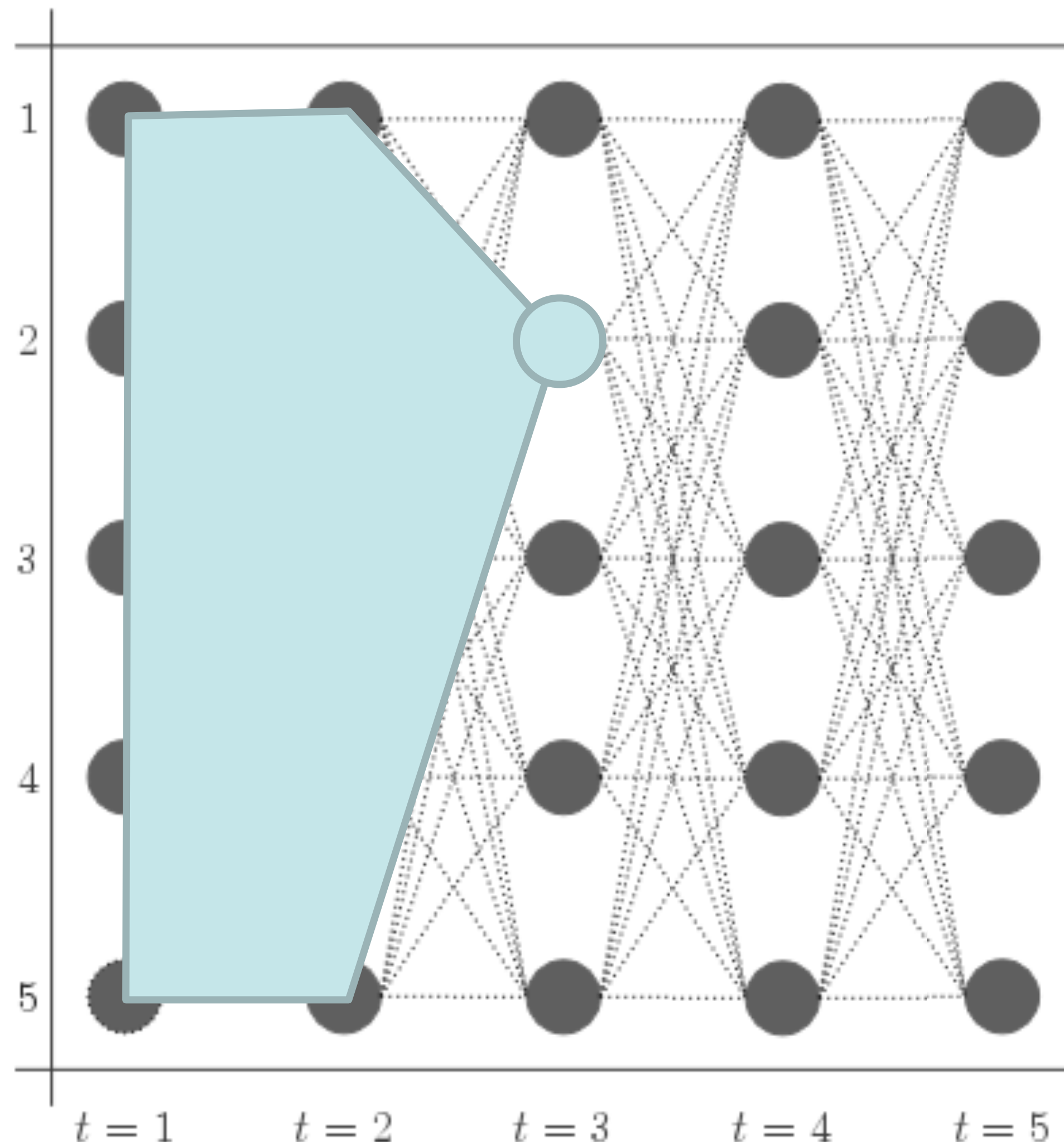
$$P(y_3 = 2 | \mathbf{x}) =$$

sum of all paths through state 2 at time 3
sum of all paths

$$= \frac{\text{light blue shape}}{\text{purple shape}}$$

- ▶ Easiest and most flexible to do one pass to compute and one to compute

Forward-Backward Algorithm



- ▶ Initial:

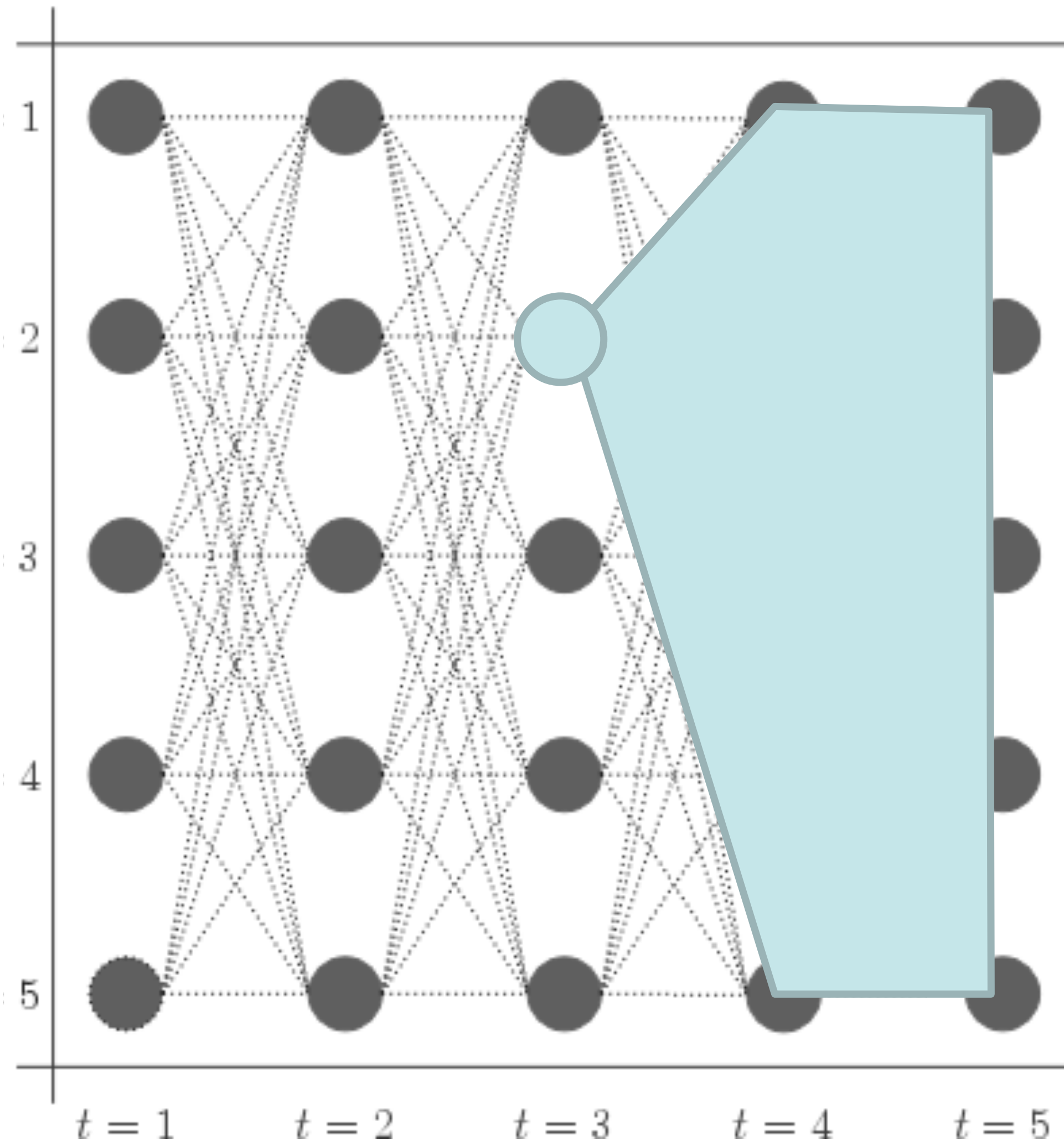
$$\alpha_1(s) = \exp(\phi_e(s, 1, \mathbf{x}))$$

- ▶ Recurrence:

$$\alpha_t(s_t) = \sum_{s_{t-1}} \alpha_{t-1}(s_{t-1}) \exp(\phi_e(s_t, t, \mathbf{x})) \exp(\phi_t(s_{t-1}, s_t))$$

- ▶ Same as Viterbi but summing instead of maxing!
- ▶ These quantities get very small!
Store everything as log probabilities

Forward-Backward Algorithm



- ▶ Initial:

$$\beta_n(s) = 1$$

- ▶ Recurrence:

$$\beta_t(s_t) = \sum_{s_{t+1}} \beta_{t+1}(s_{t+1}) \exp(\phi_e(s_{t+1}, t+1, \mathbf{x})) \exp(\phi_t(s_t, s_{t+1}))$$

- ▶ Big differences: count emission for the *next* timestep (not current one)

Forward-Backward Algorithm

$$\alpha_1(s) = \exp(\phi_e(s, 1, \mathbf{x}))$$

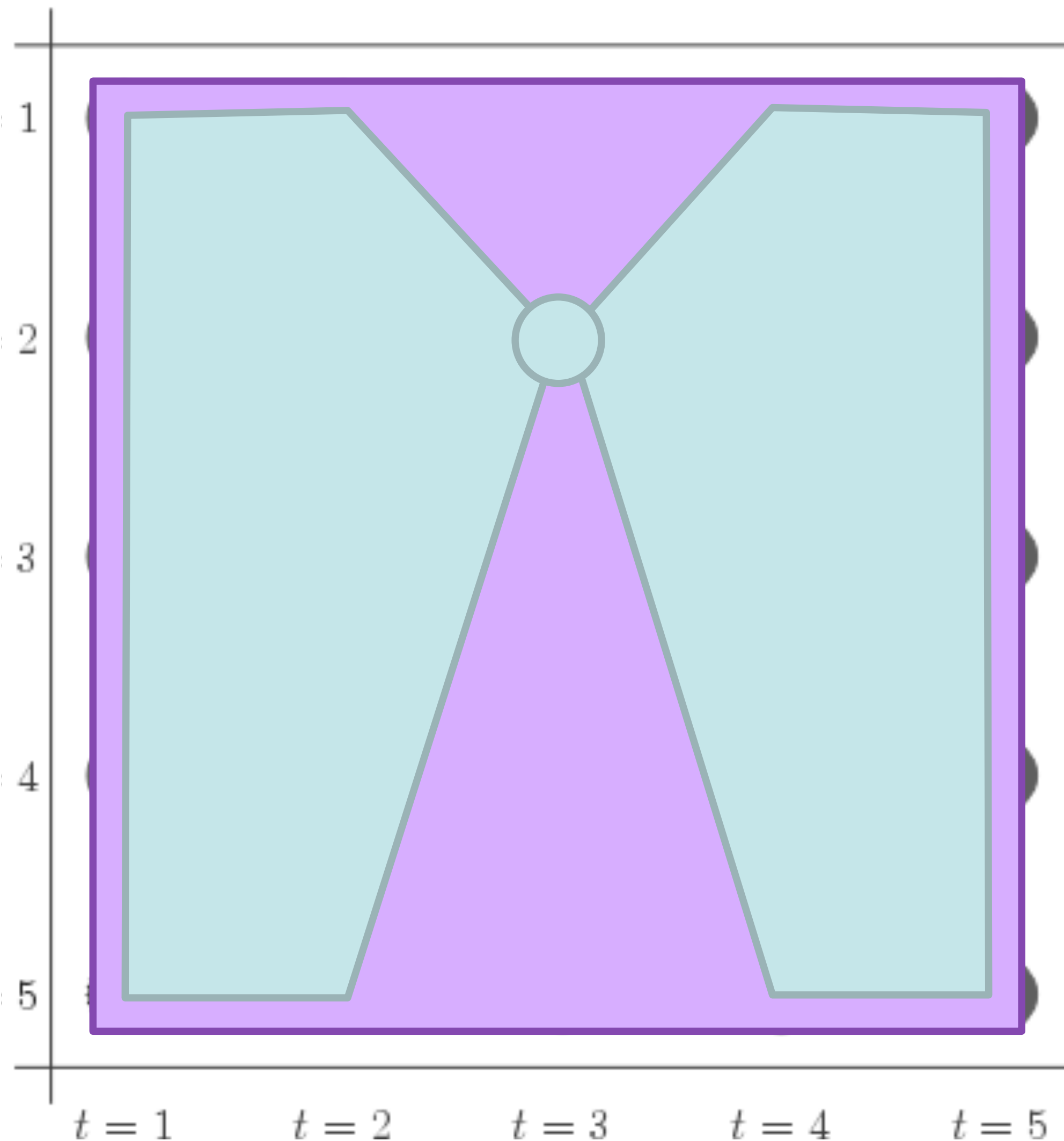
$$\alpha_t(s_t) = \sum_{s_{t-1}} \alpha_{t-1}(s_{t-1}) \exp(\phi_e(s_t, t, \mathbf{x})) \exp(\phi_t(s_{t-1}, s_t))$$

$$\beta_n(s) = 1$$

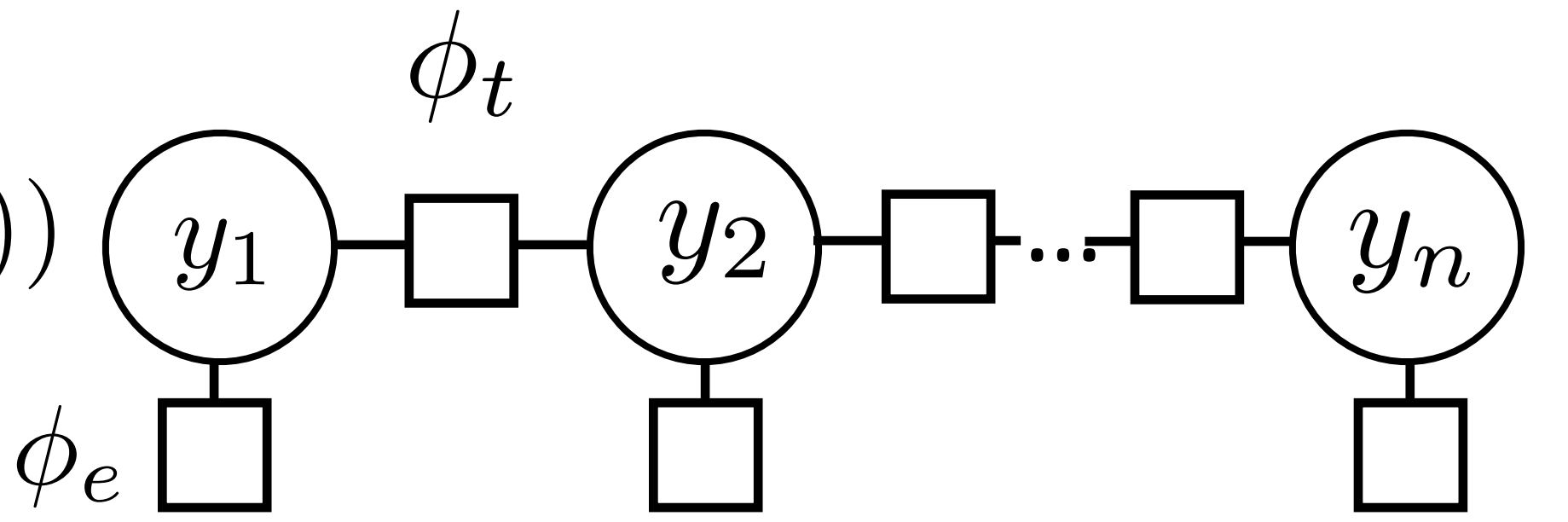
$$\beta_t(s_t) = \sum_{s_{t+1}} \beta_{t+1}(s_{t+1}) \exp(\phi_e(s_{t+1}, t+1, \mathbf{x})) \exp(\phi_t(s_t, s_{t+1}))$$

$$P(s_3 = 2 | \mathbf{x}) = \frac{\alpha_3(2)\beta_3(2)}{\sum_i \alpha_3(i)\beta_3(i)} = \frac{\text{Diagram of state 2 at } t=3}{Z}$$

► What is the denominator here? Z



Computing Marginals

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


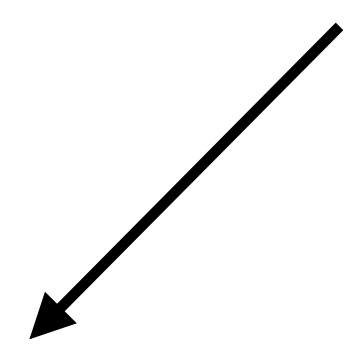
► Normalizing constant $Z = \sum_{\mathbf{y}} \prod_{i=2}^n \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^n \exp(\phi_e(y_i, i, \mathbf{x}))$

► Analogous to $P(\mathbf{x})$ for HMMs

► For both HMMs and CRFs:

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

Z for CRFs,
P(**x**) for HMMs



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
- ▶ ... but, you can build a pretty good system without learned transition features (e.g., use heuristic weights, or just enforce constraints like B-PER → I-ORG is illegal)

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: $\operatorname{argmax} P(\mathbf{y}|\mathbf{x})$ 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

- extract features on each emission and transition (look up in cache)
- compute potentials ϕ based on features + weights
- compute marginal probabilities with forward-backward
- accumulate gradient over all emissions and transitions
- apply the gradient update

Implementation Tips for CRFs

- ▶ Caching is your friend! Cache feature vectors especially, and reduce redundant computation
- ▶ Exploit sparsity in feature vectors where possible, especially in feature vectors and gradients
- ▶ Do all dynamic program computation in log space to avoid underflow
- ▶ If things are too slow, run a profiler and see where time is being spent. Forward-backward should take most of the time

Debugging Tips for CRFs

- ▶ Hard to know whether inference, learning, or the model is broken!
- ▶ Compute the objective — is optimization working?
 - ▶ **Inference:** check gradient computation (most likely place for bugs)
 - ▶ Is $\sum_s \text{forward}_i(s) \text{backward}_i(s)$ the same for all i ?
 - ▶ Do probabilities normalize correctly + look “reasonable”? (Nearly uniform when untrained, then slowly converging to the right thing)
 - ▶ **Learning:** is the objective going down? Can you fit a small training set (of 1 or 10 examples)? Are you applying the gradient correctly?
- ▶ If objective is going down but model performance is bad:
 - ▶ **Inference:** check performance if you decode the training set

Application in NER

NER Results

- ▶ CRF with lexical features can get around 85 F1 on CoNLL 2003: 4 classes (PER, ORG, LOC, MISC) on newswire data
- ▶ What else do we need to capture?
- ▶ World knowledge:

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

Tanjug

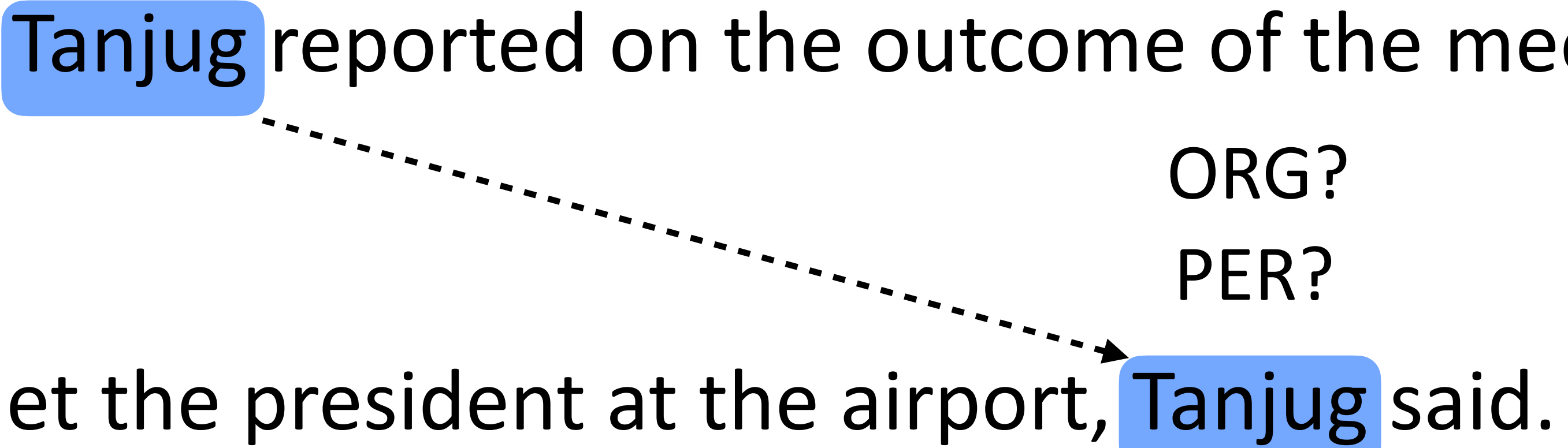
From Wikipedia, the free encyclopedia

Tanjug (/ˈtʌnjʊɡ/) ([Serbian Cyrillic](#): Танјуг) is a Serbian state news agency based in [Belgrade](#).^[2]

Nonlocal Features

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

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

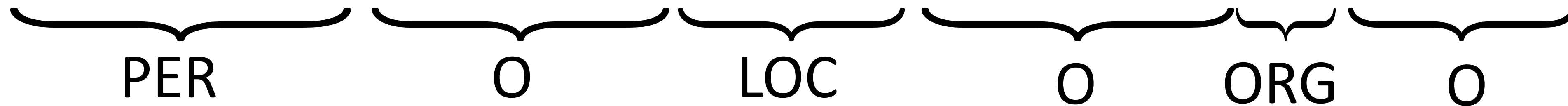


ORG?
PER?

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

Semi-Markov CRF Models

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



- ▶ Chunk-level (n -gram) prediction rather than token-level BIO
- ▶ $\mathbf{y}$ 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

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

Ratinov and Roth (2009)

Lample et al. (2016)

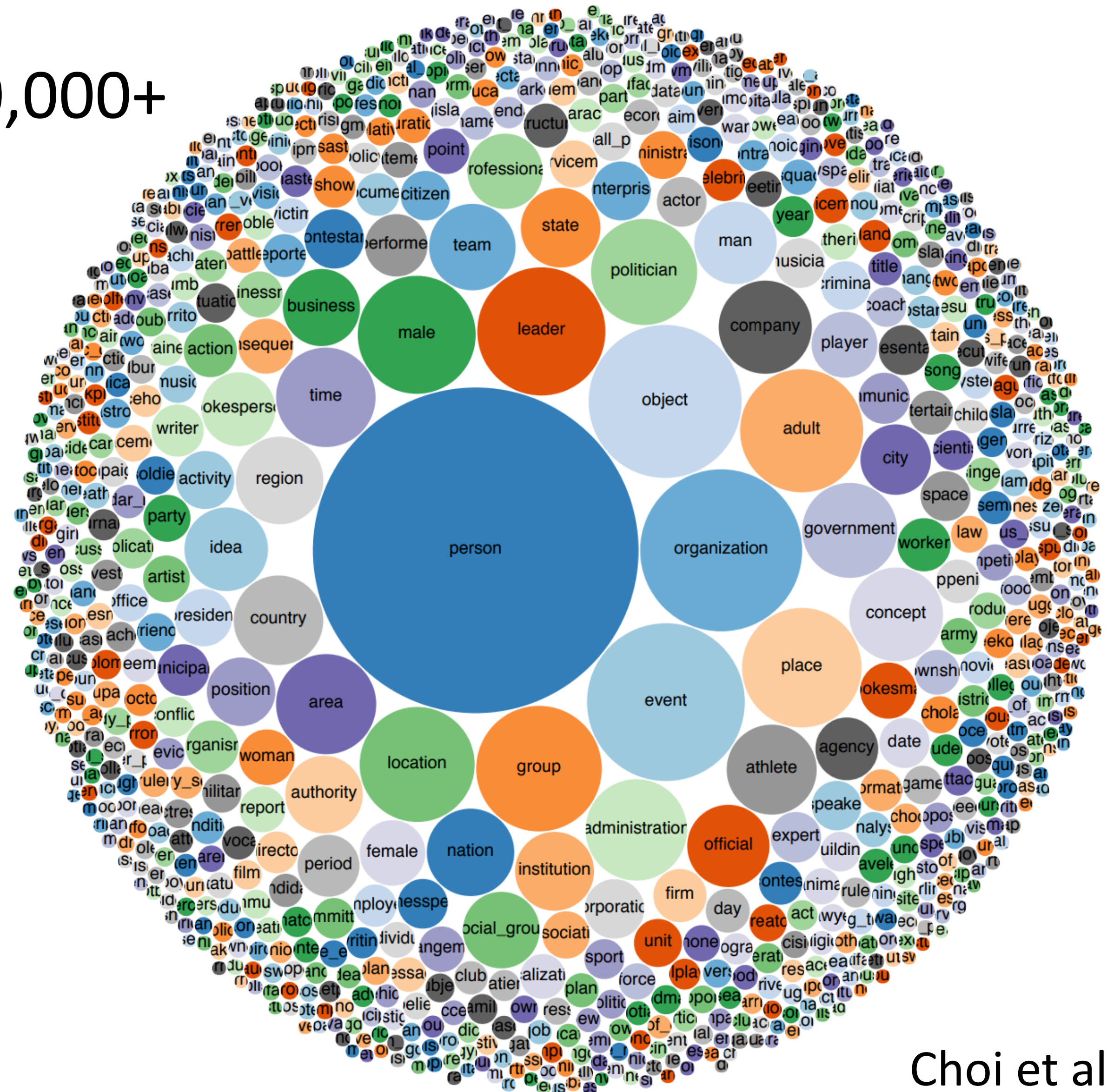
LSTM-CRF (no char)	90.20
LSTM-CRF	90.94
S-LSTM (no char)	87.96
S-LSTM	90.33

BiLSTM-CRF + ELMo
Peters et al. (2018) **92.2**

Fine-tuning approach	Dev F1	Test F1
BERT _{LARGE}	96.6	92.8
BERT _{BASE}	96.4	92.4
Devlin et al. (2019)		

Ultra-fine Entity Typing

- ▶ Scaling up to larger typesets 10,000+
- ▶ Person/official/president ...



Choi et al. (2018)

Sentence Alignment

► Neural CRF for Sentence Alignment

NEWSELA

WAR & PEACE SCIENCE KIDS MONEY HEALTH

SCIENCE 1738 SHARE

Archaeologist may have found remains of ancient Egyptian Queen Nefertiti

By Robert Gebelhoff, Washington Post. 08.17.15



The 3,330-year-old bust of Nefertiti sits in an exhibition in the Kulturforum in Berlin, Germany, March 1, 2005. Photo: AP/Herbert Knosowski

Nefertiti — she's an ancient Egyptian queen and the source of a fantastic mystery regarding the iconic remnants of long-lost royalty.

For decades, archaeologists have speculated on the location of the queen's remains, the last royal mummy missing from the dynasty of the famous King Tutankhamun, better known as King Tut. But now, an archaeologist claims that he has found her

MAX
1140L

960L
720L
420L

WRITE
QUIZ

NEWSELA

WAR & PEACE SCIENCE KIDS MONEY LAW HEALTH

SCIENCE 1738 SHARE

Mystery of ancient Egypt solved? Tomb of queen may be hidden near King Tut

By Washington Post, adapted by Newsela staff. 08.17.15



The 3,330-year-old bust of Nefertiti sits in an exhibition in the Kulturforum in Berlin, Germany, March 1, 2005. Photo: AP/Herbert Knosowski

The ancient Egyptian Queen Nefertiti has long been at the center of a mystery. For years, archaeologists have wondered where her tomb might be hidden. Nefertiti belonged to the family line of the famous King Tutankhamun, better known as King Tut. Indeed, some believe she was Tut's mother. While the other royals in her line are

1140L

960L

720L

420L

WRITE
QUIZ

Professional editors rewrite news articles into 4 different readability levels for grade 3-12 students.

Sentence Alignment

► Neural CRF for Sentence Alignment

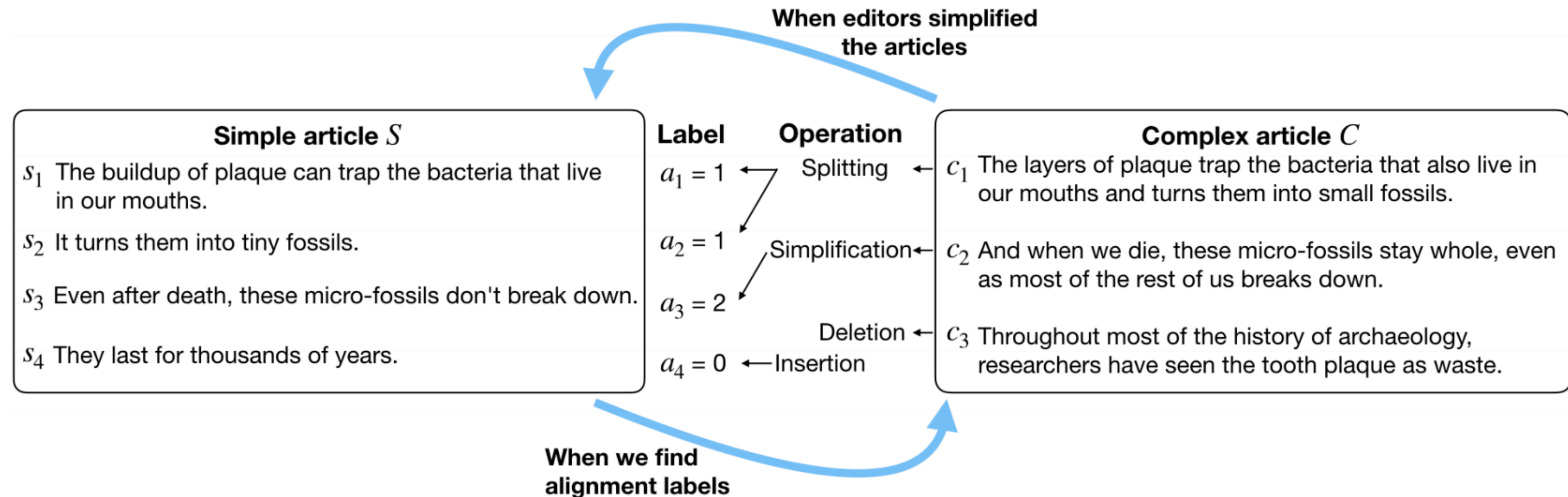
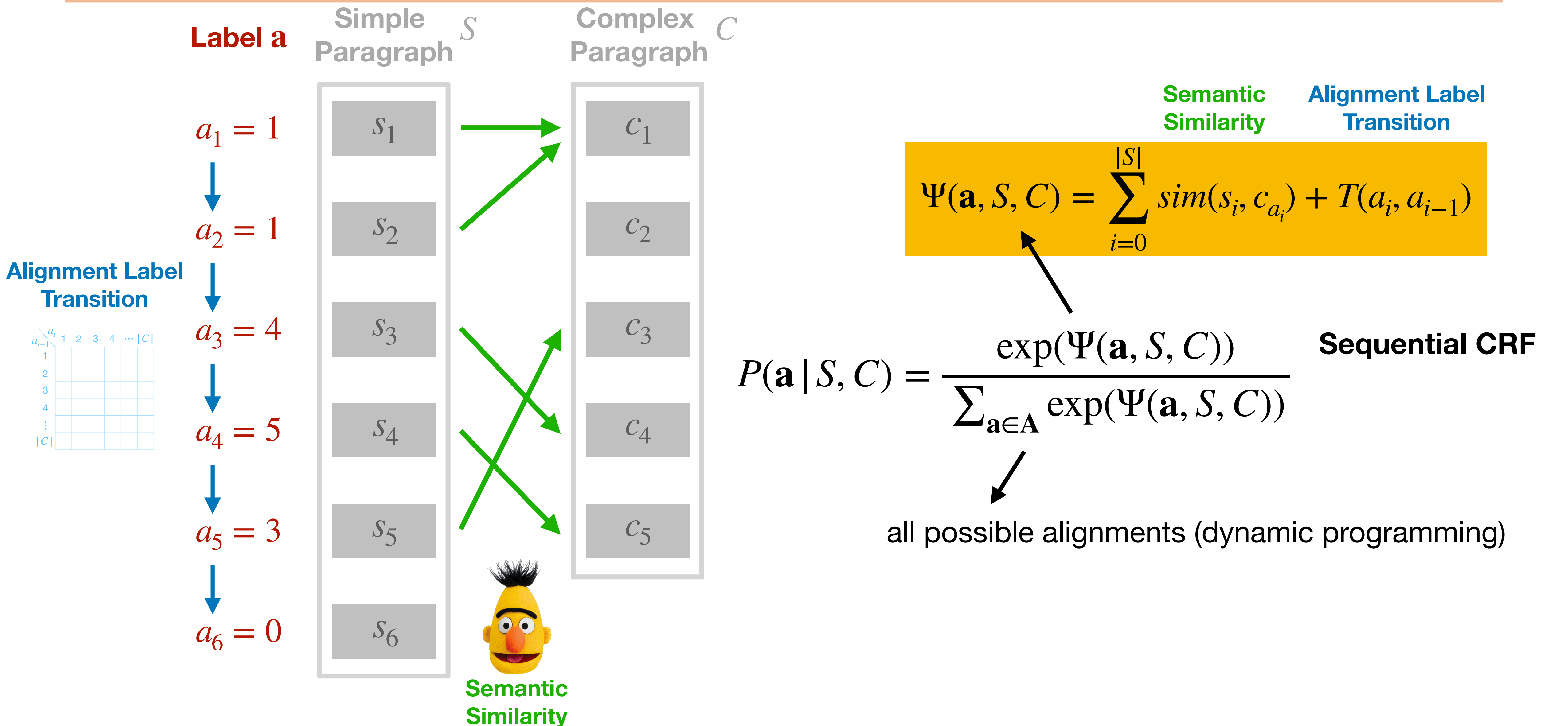


Figure 1: An example of sentence alignment between an original news article (right) and its simplified version (left) in Newsela. The label a_i for each simple sentence s_i is the index of complex sentence c_{a_i} it aligns to.

Sentence Alignment



Sentence Alignment

- Structure prediction + BERT_{finetune} → A neural CRF alignment model.

		aligned + partial vs. others*		
		Precision	Recall	F1
Greedy	JaccardAlign (Xu et al., 2015)	98.66	67.58	80.22
Dynamic Programming	MASSAlign (Paetzold et al., 2017)	95.49	82.27	88.39
Greedy	CATS (Štajner et al., 2018)	88.56	91.31	89.92
Threshold	BERT _{finetune}	94.99	89.62	92.22
Threshold	BERT _{finetune} + paragraph alignment	98.05	88.63	93.10
CRF	Our CRF aligner	97.86	91.31	95.59

+5.7

* Results are on the manually annotated Newsela dataset.

Next Up

- ▶ More sequential models
 - ▶ Recurrent Neural Network
 - ▶ Neural CRF model

Sequential Models

$$\mathbf{x} = (x_1, x_2, \dots, x_n) \quad , \quad \mathbf{y} = (y_1, y_2, \dots, y_n)$$

$$\hat{\mathbf{y}} = \underset{\mathbf{y} \in \mathcal{Y}(\mathbf{x})}{\operatorname{argmax}} \quad \psi(\mathbf{x}, \mathbf{y})$$

$$\begin{cases} \mathcal{Y}(\mathbf{x}) = \mathcal{Y}^n \\ \text{where } \mathcal{Y} = \{NV, VB, \dots\} \\ |\mathcal{Y}(\mathbf{x})| = |\mathcal{Y}|^n \end{cases}$$

scoring function on pairs of sequences.

$$\begin{matrix} \text{Vocabulary} \rightarrow & \mathcal{V}^n \times \mathcal{Y}^n & \rightarrow & \mathbb{R} \\ & \text{label space} & & \text{real number} \end{matrix}$$

$$\psi(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n+1} \left\{ \psi(\mathbf{x}, y_i, y_{i-1}, i) \right\} = \mathbf{w}^T \cdot \mathbf{f}(\mathbf{x}, y_i, y_{i-1}, i)$$

linear model
feature-based

decompose into a local scoring function
to make the inference more tractable.

making a series of inter connected labeling decisions.

HMM

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\operatorname{argmax}} \quad P(\mathbf{y} | \mathbf{x})$$

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\operatorname{argmax}} \quad \{P(\mathbf{x}, \mathbf{y})\}$$

$$= \underset{\mathbf{y}}{\operatorname{argmax}} \quad \prod_{i=2}^n P(y_i | y_{i-1}) \prod_{i=1}^n P(x_i | y_i)$$

CRF

normalizing
constant
(partition
function)

$$P(y|x) = \frac{\exp(\psi(x, y))}{\sum_{y' \in Y(x)} \exp(\psi(x, y'))} = Z$$

almost identical to LR,

except that the label space is now sequence of tags.
requiring efficient algorithm for both:

- decoding: search for best tag seq. y^* , given x and θ
- normalization: sum over all tag sequences $y(x)$

$$\hat{y} = \operatorname{argmax}_y \log P(y|x)$$

$$= \operatorname{argmax}_y \log \frac{1}{Z} \exp(\psi(x, y))$$

$$= \operatorname{argmax}_y \psi(x, y)$$

$$= \operatorname{argmax}_y w^T f(x, y_{i-1}, y_i, i)$$

LR training

M training examples

$$L(x, y^*) = \sum_{j=1}^M \log P(y^{(j)*} | x^{(j)})$$

$$= \sum_{j=1}^M \left(W^T f(x^{(j)}, y^{(j)*}) - \log \sum_{y \in \mathcal{Y}} \exp(W^T f(x^{(j)}, y)) \right)$$

~~$L(x, y)$~~

$$\frac{\partial}{\partial W} L(x, y^*) = f(x^{(j)}, y^{(j)*}) - \sum_{y^*} f(x^{(j)}, y^*) P_W(y^* | x^{(j)})$$

$$= f(x^{(j)}, y^{(j)*}) - \mathbb{E}_y [f(x^{(j)}, y)]$$

gold feature value

model's expectation of feature value.

CRF training

M training examples
sequences
sentences.

$$L(x, y^*) = \sum_{j=1}^M \log P(y^{(j)*} | x^{(j)})$$

$$\frac{\partial}{\partial W} L(x, y^*) = f(x^{(j)}, y^{(j)*}) - \mathbb{E}_y [f(x^{(j)}, y)]$$

$$= f(x^{(j)}, y^{(j)*}) - \sum_{y \leftarrow \text{intractable}} f(x^{(j)}, y) P_W(y | x^{(j)})$$