ANLP

10 - CRFs (sequences, part II)

David Schlangen
University of Potsdam, MSc Cognitive Systems
Winter 2019 / 2020

Industry Day 2019

Idea

Every two years, the Applied CL group invites a number of local companies (related to language technology) to Uni Potsdam, so that

- students get an idea on current topics and research/development practices in industry
- contacts for internships, part-time or full-time jobs can be made

The 2019 event: Friday November 22nd, 10am to 4pm

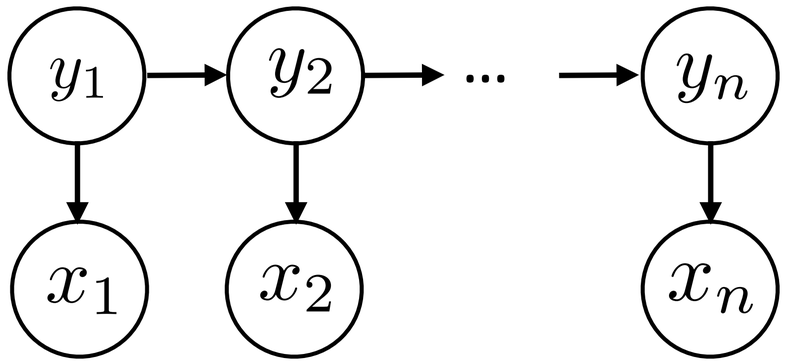
- Venue: Campus Griebnitzsee, Building 6, Room S24 / S25

Participating companies in 2019

- acrolinx
- Carmeq
- DFKI
- ID Berlin
- parlamind
- rasa
- retresco
- zalando research

Recall: HMMs

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



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

- Training: maximum likelihood estimation (with smoothing)
- Inference problem: $\text{argmax}_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}) = \text{argmax}_{\mathbf{y}} \frac{P(\mathbf{y}, \mathbf{x})}{\cancel{P(\mathbf{x})}}$
- 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

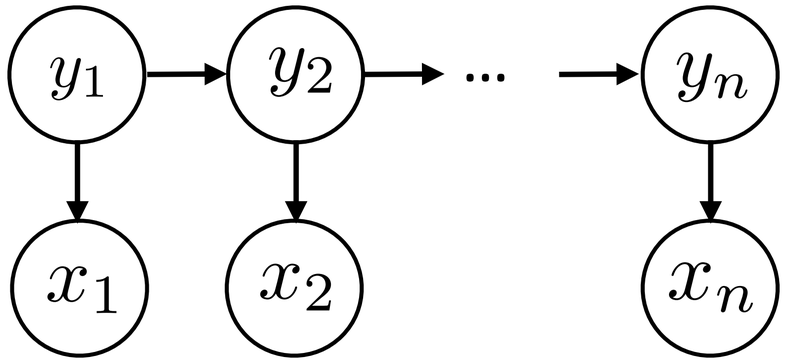
Named Entity Recognition

- 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)



- This reflects the following decomposition:

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

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

Conditional Random Fields

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

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

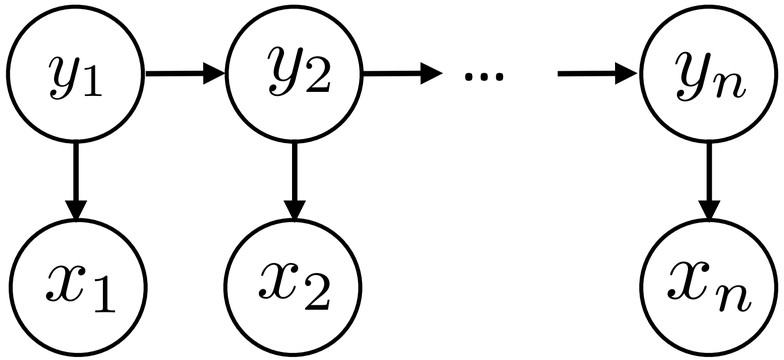
normalizer

any real-valued scoring function of its arguments

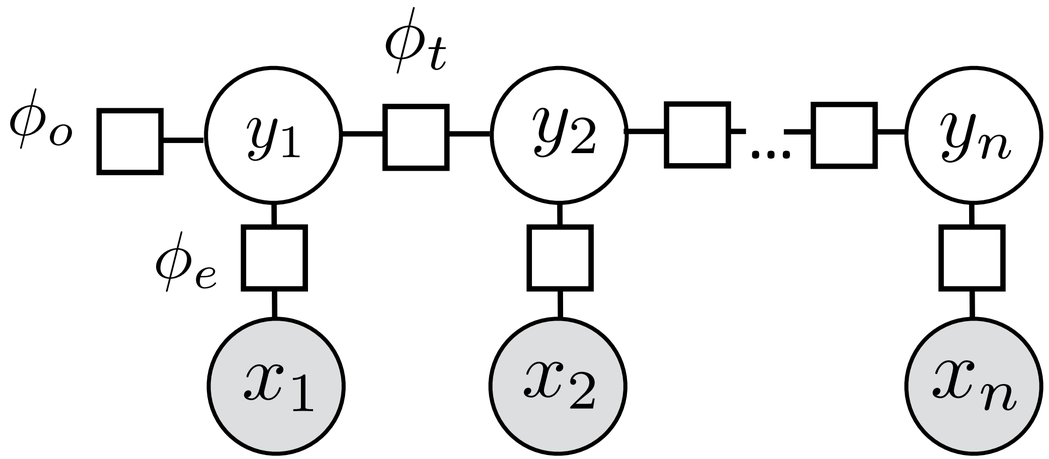
- Naive Bayes : logistic regression :: HMMs : CRFs
  local vs. global normalization <-> generative vs. discriminative
- Locally normalized discriminative models do exist (MEMMs)
- How do we max over ***y***? Intractable in general — can we fix this?

Sequential CRFs

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



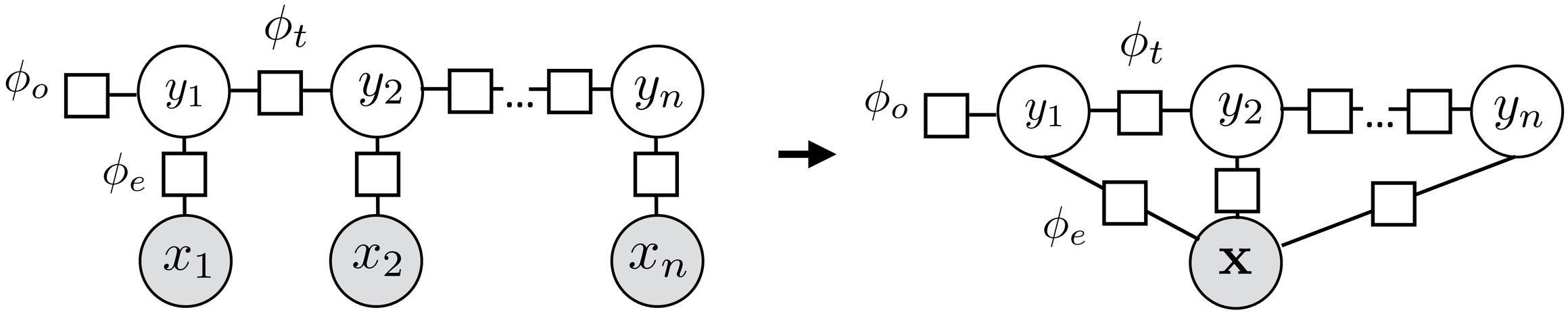
- CRFs:



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

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

Sequential CRFs



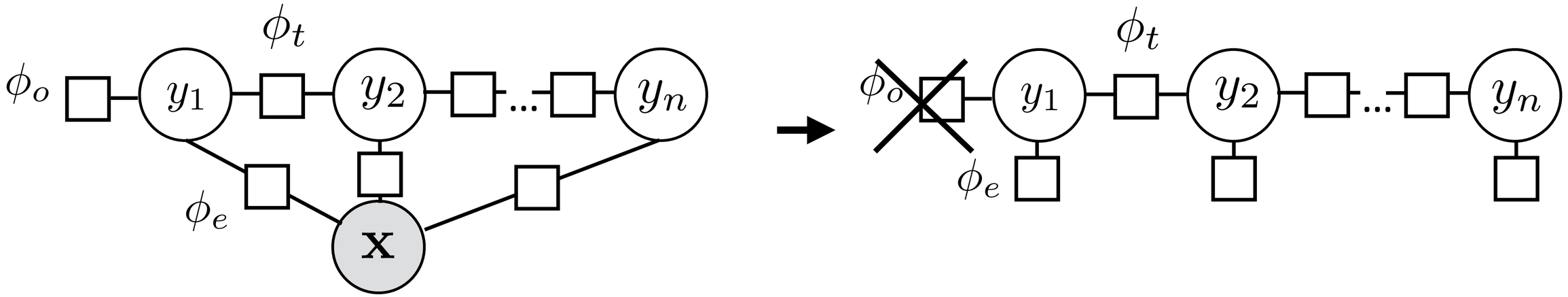
$$P(\mathbf{y}|\mathbf{x}) \propto \exp(\phi_o(y_1)) \prod_{i=2}^{n} \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^{n} \exp(\phi_e(x_i, y_i))$$

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

token index — lets us look at current word

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

Sequential CRFs



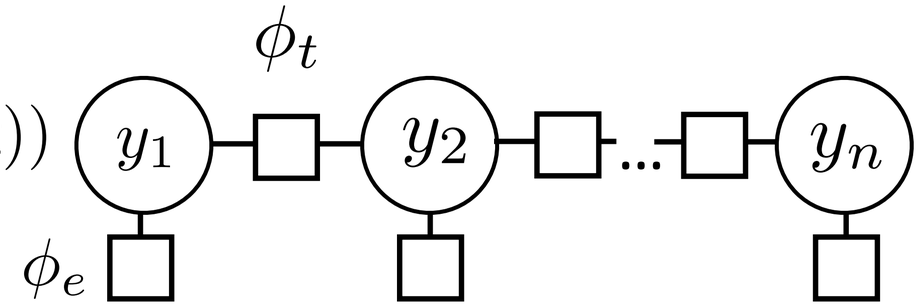
- Notation: omit **x** from the factor graph entirely (implicit)
- Don't include initial distribution, can bake into other factors

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

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



- Phis 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

$$
\begin{aligned}
\boldsymbol{f}(\boldsymbol{w} = \textit{they can fish}, \boldsymbol{y} = \mathrm{N\ V\ V}\ ) &= \sum_{m=1}^{M+1} \boldsymbol{f}(\boldsymbol{w}, y_m, y_{m-1}, m) \\
&= \boldsymbol{f}(\boldsymbol{w}, \mathrm{N}, \lozenge, 1) \\
&\quad + \boldsymbol{f}(\boldsymbol{w}, \mathrm{V}, \mathrm{N}, 2) \\
&\quad + \boldsymbol{f}(\boldsymbol{w}, \mathrm{V}, \mathrm{V}, 3) \\
&\quad + \boldsymbol{f}(\boldsymbol{w}, \blacklozenge, \mathrm{V}, 4) \\
&= (w_m = \textit{they}, y_m = \mathrm{N}) + (y_m = \mathrm{N}, y_{m-1} = \lozenge) \\
&\quad + (w_m = \textit{can}, y_m = \mathrm{V}) + (y_m = \mathrm{V}, y_{m-1} = \mathrm{N}) \\
&\quad + (w_m = \textit{fish}, y_m = \mathrm{V}) + (y_m = \mathrm{V}, y_{m-1} = \mathrm{V}) \\
&\quad + (y_m = \blacklozenge, y_{m-1} = \mathrm{V}).
\end{aligned}
$$

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

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

Ind[B-LOC & Prev word = *to*]

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

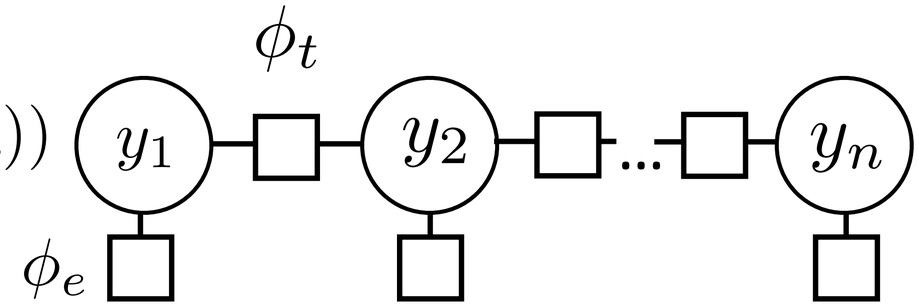
- Model: $P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_{i=2}^{n} \exp(\phi_t(y_{i-1}, y_i)) \prod_{i=1}^{n} \exp(\phi_e(y_i, i, \mathbf{x}))$

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

- Inference
- Learning

Computing (arg)maxes

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



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

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

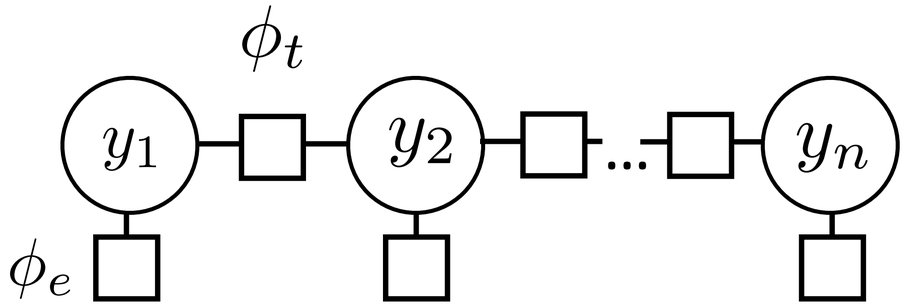
$$= \max_{y_2,\ldots,y_n} e^{\phi_t(y_{n-1},y_n)} e^{\phi_e(y_n,n,\mathbf{x})} \ldots e^{\phi_e(y_2,2,\mathbf{x})} \boxed{\max_{y_1} e^{\phi_t(y_1,y_2)}} \underbrace{e^{\phi_e(y_1,1,\mathbf{x})}}$$

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

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

Inference in General CRFs

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

Training CRFs

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

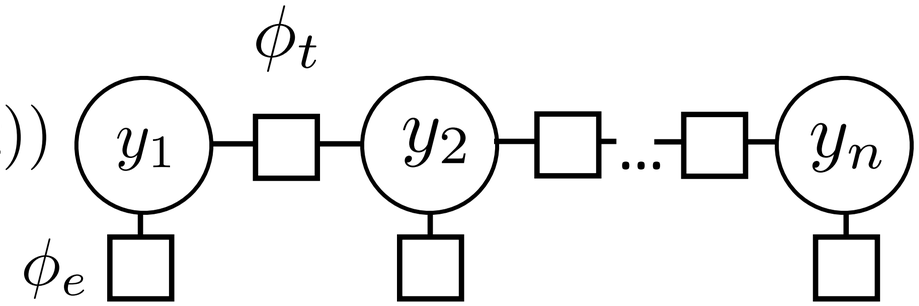
- Let's focus on emission feature expectation

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

$$= \sum_{i=1}^{n}\sum_{s} P(y_i = s|\mathbf{x}) f_e(s, i, \mathbf{x})$$

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(**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

backward$_i$(s) = probability / score for generating rest of observation seq., from here

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 transition features

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(**y**|**x**) from Viterbi

- Learning: run forward-backward to compute posterior probabilities; then

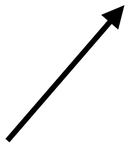
$$\frac{\partial}{\partial w} \mathcal{L}(\mathbf{y}^*, \mathbf{x}) = \sum_{i=1}^{n} f_e(y_i^*, i, \mathbf{x}) - \sum_{i=1}^{n} \sum_{s} P(y_i = s|\mathbf{x}) f_e(s, i, \mathbf{x})$$

NER

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.

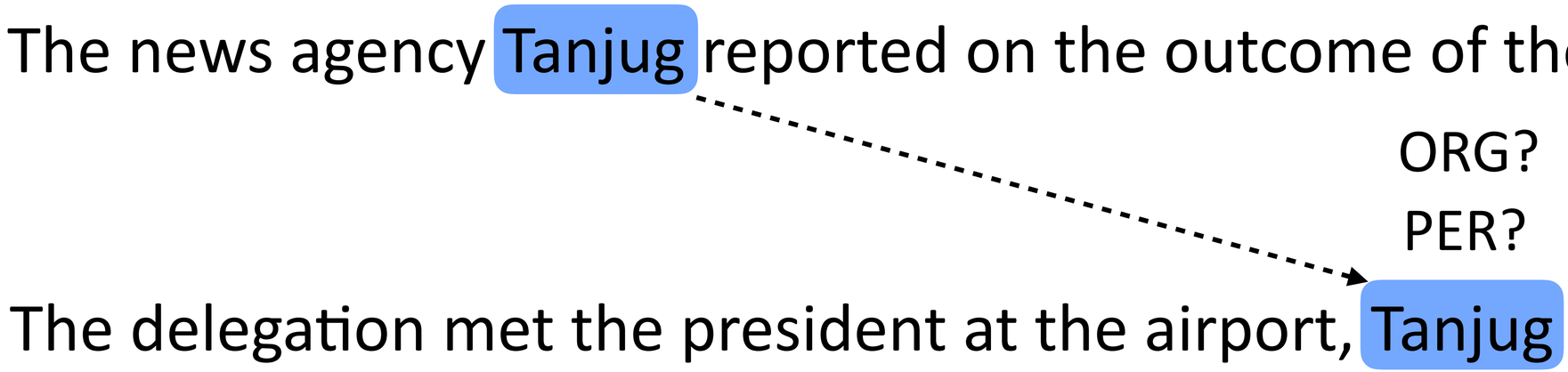


Tanjug

From Wikipedia, the free encyclopedia

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

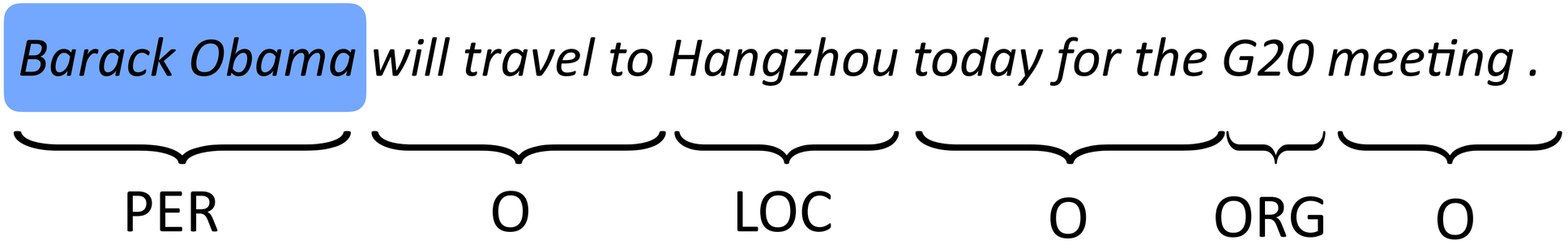
Nonlocal Features



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

Finkel and Manning (2008), Ratinov and Roth (2009)

Semi-Markov Models



- 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

Sarawagi and Cohen (2004)

Evaluating NER

- 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

Ratinov and Roth (2009)

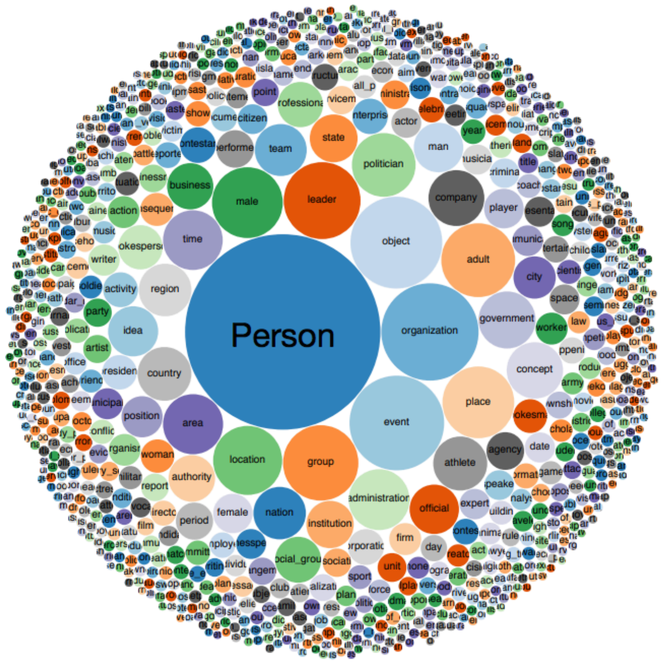
Lample et al. (2016)

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**

BERT
Devlin et al. (2019) **92.8**

Modern Entity Typing



a) Our Dataset | b) OntoNotes | c) FIGER

- More and more classes (17 -> 112 -> 10,000+)

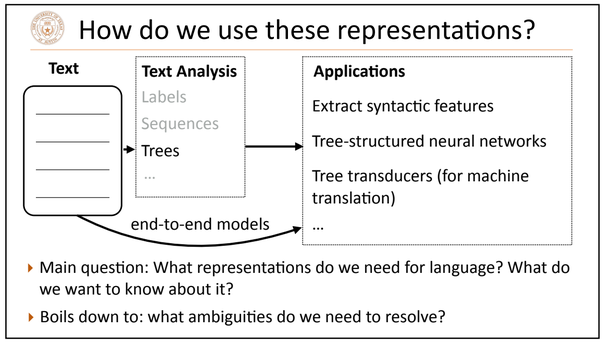
Choi et al. (2018)

Next Time

- Neural networks

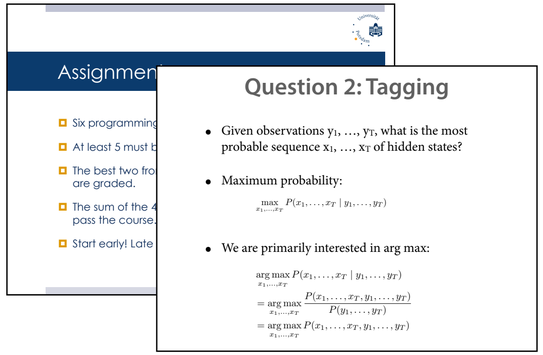
slide credits

slides that look like this



come from

CS388 given by Greg Durrett at U Texas, Austin



earlier editions of this class (ANLP), given by Tatjana Scheffler and Alexander Koller

and their use is gratefully acknowledged. I try to make any modifications obvious, but if there are errors on a slide, assume that I added them.