# CRFs and NER

## Named Entity Recognition

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

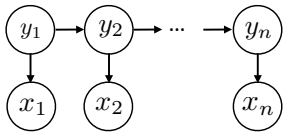
*Barack Obama will travel to Hangzhou today for the G20 meeting .*

PERSON LOC ORG

- Frame as a sequence problem with a BIO tagset: begin, inside, outside
- Why might an HMM not do so well here?
  - Lots of O's, so tags aren't as informative about context
  - Want to use context features (*to Hangzhou* => *Hangzhou* is a LOC)
- Conditional random fields (CRFs) can help solve these problems

## HMMs

- Big advantage: transitions, scoring pairs of adjacent y's




- Big downside: not able to incorporate useful word context information
- Solution: switch from generative to discriminative model (conditional random fields) so we can condition on the *entire input*.
- Conditional random fields: logistic regression + features on pairs of y's

## Tagging with Logistic Regression

- Logistic regression over each tag individually:

"different features" approach to features for a single tag

$$P(y_i = y|\mathbf{x}, i) = \frac{\exp(\mathbf{w}^\top \mathbf{f}(y, i, \mathbf{x}))}{\sum_{y' \in \mathcal{Y}} \exp(\mathbf{w}^\top \mathbf{f}(y', i, \mathbf{x}))}$$

- Over all tags:

$$P(\mathbf{y} = \tilde{\mathbf{y}}|\mathbf{x}) = \prod_{i=1}^{n} P(y_i = \tilde{y}_i|\mathbf{x}, i) = \frac{1}{Z} \exp\left(\sum_{i=1}^{n} \mathbf{w}^\top \mathbf{f}(\tilde{y}_i, i, \mathbf{x})\right)$$

- Score of a prediction: sum of weights dot features over each individual predicted tag (this is a simple CRF but not the general form)
- Set *Z* equal to the product of denominators; we'll discuss this in a few slides

## Example

B-PER I-PER O O
*Barack Obama will travel*
feats = $\mathbf{f}_e$(B-PER, i=1, **x**) + $\mathbf{f}_e$(I-PER, i=2, **x**) + $\mathbf{f}_e$(O, i=3, **x**) + $\mathbf{f}_e$(O, i=4, **x**)

[CurrWord=*Obama* & label=I-PER, PrevWord=*Barack* & label=I-PER, CurrWordIsCapitalized & label=I-PER, …]

B-PER B-PER O O
*Barack Obama will travel*
feats = $\mathbf{f}_e$(B-PER, i=1, **x**) + $\mathbf{f}_e$(B-PER, i=2, **x**) + $\mathbf{f}_e$(O, i=3, **x**) + $\mathbf{f}_e$(O, i=4, **x**)

## Adding Structure

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

- We want to be able to learn that some tags don't follow other tags — want to have features on tag *pairs*

$$P(\mathbf{y} = \tilde{\mathbf{y}}|\mathbf{x}) = \frac{1}{Z} \exp\left(\sum_{i=1}^{n} \mathbf{w}^\top \mathbf{f}_e(\tilde{y}_i, i, \mathbf{x}) + \sum_{i=1}^{n} \mathbf{w}^\top \mathbf{f}_t(\tilde{y}_i, \tilde{y}_{i+1}, i, \mathbf{x})\right)$$

- Score: sum of weights dot $\mathbf{f}_e$ features over each predicted tag ("emissions") plus sum of weights dot $\mathbf{f}_t$ features over tag pairs ("transitions")
- This is a sequential CRF

## Example

B-PER I-PER O O
*Barack Obama will travel*
feats = $\mathbf{f}_e$(B-PER, i=1, **x**) + $\mathbf{f}_e$(I-PER, i=2, **x**) + $\mathbf{f}_e$(O, i=3, **x**) + $\mathbf{f}_e$(O, i=4, **x**)
+ $\mathbf{f}_t$(B-PER, I-PER, i=1, **x**) + $\mathbf{f}_t$(I-PER, O, i=2, **x**) + $\mathbf{f}_t$(O, O, i=3, **x**)

B-PER B-PER O O
*Barack Obama will travel*
feats = $\mathbf{f}_e$(B-PER, i=1, **x**) + $\mathbf{f}_e$(B-PER, i=2, **x**) + $\mathbf{f}_e$(O, i=3, **x**) + $\mathbf{f}_e$(O, i=4, **x**)
+ $\mathbf{f}_t$(B-PER, B-PER, i=1, **x**) + $\mathbf{f}_t$(B-PER, O, i=2, **x**) + $\mathbf{f}_t$(O, O, i=3, **x**)

- *Obama* can start a new named entity (emission feats look okay), but we're not likely to have two PER entities in a row (transition feats)

## Features for NER

$$P(\mathbf{y} = \tilde{\mathbf{y}}|\mathbf{x}) = \frac{1}{Z} \exp\left(\sum_{i=1}^{n} \mathbf{w}^\top \mathbf{f}_e(\tilde{y}_i, i, \mathbf{x}) + \sum_{i=1}^{n} \mathbf{w}^\top \mathbf{f}_t(\tilde{y}_i, \tilde{y}_{i+1}, i, \mathbf{x})\right)$$

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

Transitions: $\mathbf{f}_t(\text{O}, \text{B-LOC}, i = 5, \mathbf{x})$ = Indicator[O — B-LOC]

Emissions: $\mathbf{f}_e(\text{B-LOC}, i = 6, \mathbf{x})$ = Indicator[B-LOC & Curr word = *Hangzhou*]
Indicator[B-LOC & Prev word = *to*]

- We couldn't use a "previous word" feature in the HMM at all!

## 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:

$$P(\mathbf{y} = \tilde{\mathbf{y}}|\mathbf{x}) = \frac{1}{Z}\exp\left(\sum_{i=1}^{n}\mathbf{w}^\top\mathbf{f}_e(\tilde{y}_i, i, \mathbf{x}) + \sum_{i=1}^{n}\mathbf{w}^\top\mathbf{f}_t(\tilde{y}_i, \tilde{y}_{i+1}, i, \mathbf{x})\right)$$

normalizer *Z*: must make this a probability distribution over all possible seqs

$$Z = \sum_{\mathbf{y}' \in \mathcal{Y}^n}\exp\left(\sum_{i=1}^{n}\mathbf{w}^\top\mathbf{f}_e(y'_i, i, \mathbf{x}) + \sum_{i=1}^{n}\mathbf{w}^\top\mathbf{f}_t(y'_i, y'_{i+1}, i, \mathbf{x})\right)$$

- CRFs in general: replace weights dot features with so-called "potential functions" over *y*'s

## Inference and Learning

$$P(\mathbf{y} = \tilde{\mathbf{y}}|\mathbf{x}) = \frac{1}{Z}\exp\left(\sum_{i=1}^{n}\mathbf{w}^\top\mathbf{f}_e(\tilde{y}_i, i, \mathbf{x}) + \sum_{i=1}^{n}\mathbf{w}^\top\mathbf{f}_t(\tilde{y}_i, \tilde{y}_{i+1}, i, \mathbf{x})\right)$$

- Inference: Can use the Viterbi algorithm to find the highest scoring path. Replace HMM log probs with "scores" from weights dot features

$$\log P(x_i|y_i) \rightarrow \mathbf{w}^\top\mathbf{f}_e(y_i, i, \mathbf{x})$$
$$\log P(y_i|y_{i-1}) \rightarrow \mathbf{w}^\top\mathbf{f}_t(y_{i-1}, y_i, i, \mathbf{x})$$

(initial distribution is removed)

- Learning: requires running *forward-backward* (like Viterbi but with summing instead of maxing over *y*'s) to compute *Z*, then doing some tricky math to compute gradients [outside scope of the course/not on midterm]

## Takeaways

- CRFs provide a way to build structured feature-based models: logistic regression over structured objects like sequences
- Inference and learning can still be done efficiently but require dynamic programming
- CRFs don't have to be linear models; can use scores derived from neural networks ("neural CRFs")

## CRFs vs. Classifiers [Poll]

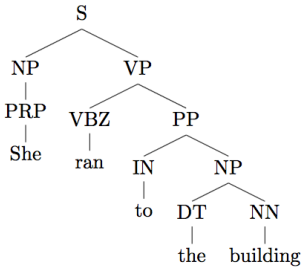
# Constituency Parsing

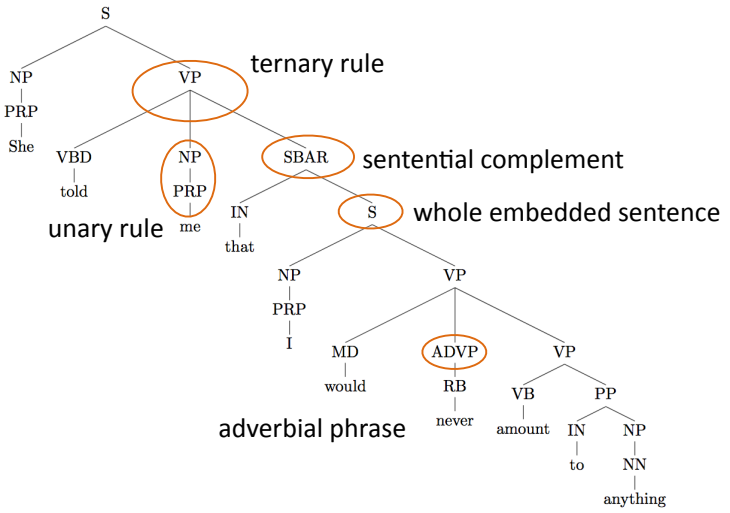
## Syntax

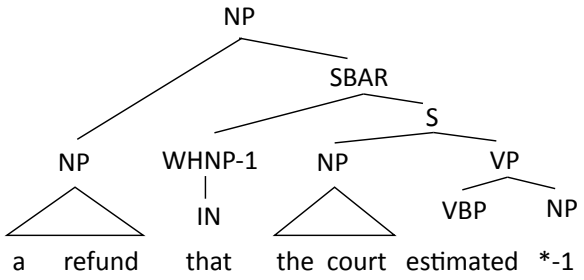
- Study of word order and how words form sentences
- Why do we care about syntax?
  - Multiple interpretations of words (noun or verb? *Fed raises*… example)
  - Recognize verb-argument structures (who is doing what to whom?)
  - Higher level of abstraction beyond words: some languages are SVO, some are VSO, some are SOV, parsing can canonicalize

## Constituency Parsing

- Tree-structured syntactic analyses of sentences
- *Constituents*: (S)entence, (N)oun (P)hrases, (V)erb (P)hrases, (P)repositional (P)hrases, and more
- Bottom layer is POS tags
- Examples will be in English. Constituency makes sense for a lot of languages but not all

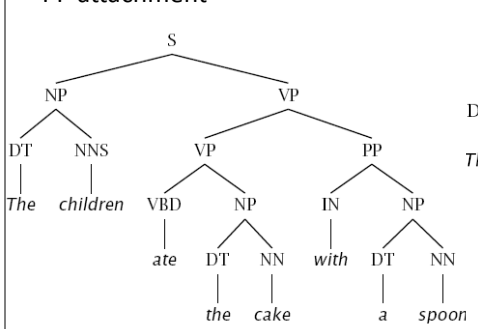


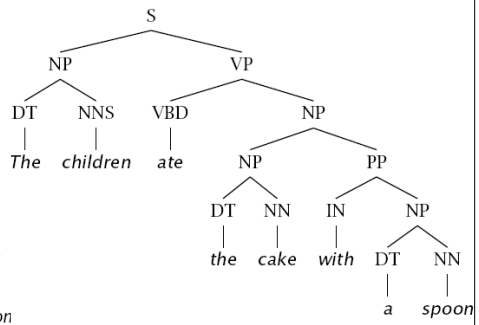






# Challenges

PP attachment




same parse as "the cake with some icing"

# Challenges

Modifier scope:





Complement structure:

*The students complained to the professor that they didn't understand*

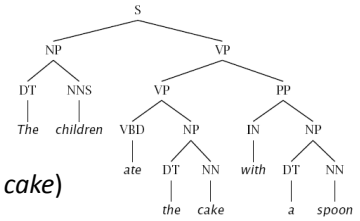
Coordination scope:

*The man picked up his hammer and saw*
compare: *The man picked up his hammer and swung*

[Eisenstein book]

# Constituency

- How do we know what the constituents are?
- Constituency tests:
  - Substitution by *proform* (e.g., pronoun)
  - Clefting (*It was with a spoon that…*)
  - Answer ellipsis (What did they eat? *the cake*)
    (How? *with a spoon*)




- Sometimes constituency is not clear, e.g., coordination: *she went to and bought food at the store*