

# Structured Learning for Taxonomy Induction with Belief Propagation

Mohit Bansal  
TTI-Chicago



David Burkett  
Twitter Inc.



Gerard de Melo  
Tsinghua U.



Dan Klein  
UC Berkeley

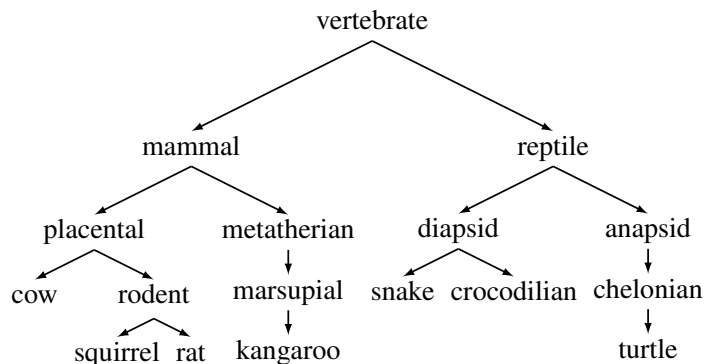




# A Lexical Taxonomy



- ▶ Captures types and categories via hypernymy



- ▶ Current resources incomplete, unavailable, time-intensive

**WordNet**



- ▶ Automatically build taxonomy trees

Widdows (2003), Snow et al. (2006), Yang and Callan (2009), Poon and Domnigos (2010), Fountain and Lapata (2012), Kozareva and Hovy (2010), Navigli et al. (2011)



# Outline



- ▶ Structured inference (during both learning and decoding) and learned semantic features on links and siblings
- ▶ Supervised learning: train on one part of WordNet (e.g., food) and test on a new part (e.g., animals)

$$\text{Train} \cap \text{Test} = \emptyset$$

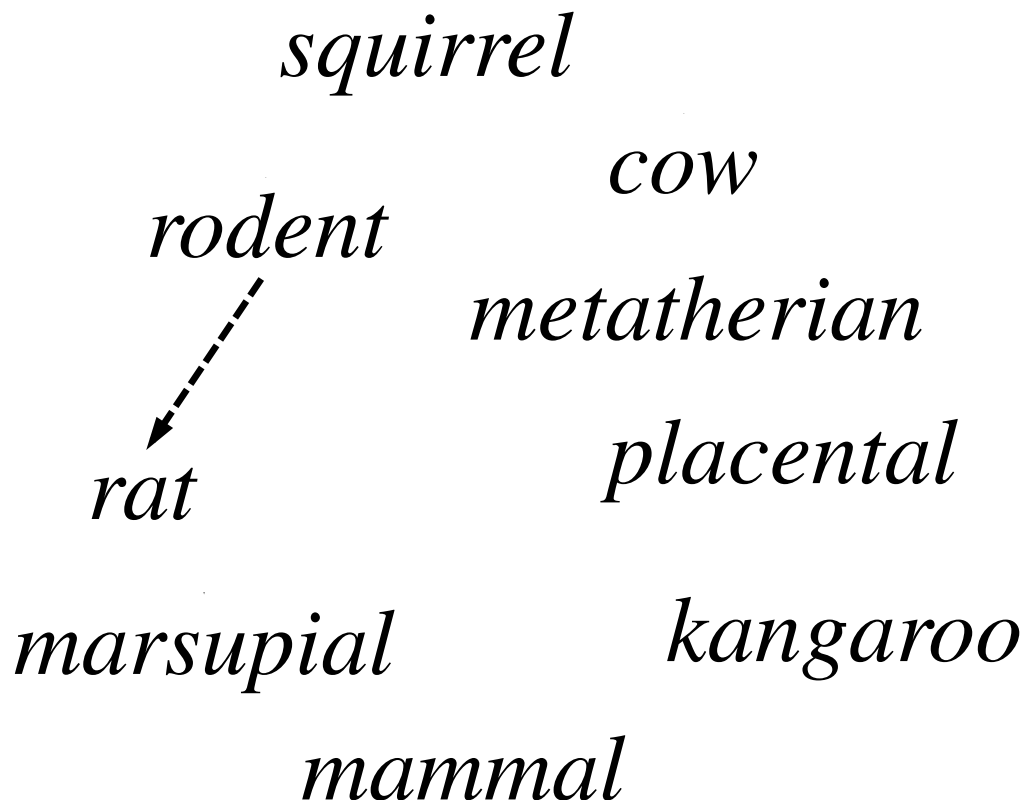
- ▶ No repeated words!!! → Cannot use lexicalized features; need surface and external Web features



# Taxonomy Induction



- For a particular set of terms  $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$





# Taxonomy Induction



Hearst, 1992

- ▶ Need features for terms that we have never seen before!

Web Ngrams

*rats and other rodents* →  $x$

*squirrel*

*rodent*

*cow*

*metatherian*

*placental*

*rat*

*marsupial*

*kangaroo*

*mammal*



# Taxonomy Induction



Hearst, 1992

- ▶ Need features for terms that we have never seen before!

Web Ngrams

*C and other P* → *x*

*squirrel*

*cow*

*rodent*

*metatherian*

*placental*

*rat*

*marsupial*

*kangaroo*

*mammal*



# Taxonomy Induction



Hearst, 1992

- ▶ Need features for terms that we have never seen before!

Web Ngrams

*rodents such as rats* → *x*

*squirrel*

*rodent*

*cow*

*metatherian*

*placental*

*rat*

*marsupial*

*kangaroo*

*mammal*



# Taxonomy Induction



Hearst, 1992

- ▶ Need features for terms that we have never seen before!

Web Ngrams

$P$  such as  $C \rightarrow x$

*squirrel*

*cow*

*rodent*

*metatherian*

*placental*

*rat*

*marsupial*

*kangaroo*

*mammal*



# Taxonomy Induction



- ▶ Need features for terms that we have never seen before!

Web Ngrams

*rodent \* \* \* rat* → *x*

*squirrel*

*rodent*

*rat*

*marsupial*

*mammal*

*cow*

*metatherian*

*placental*

*kangaroo*



# Taxonomy Induction



- ▶ Need features for terms that we have never seen before!

Web Ngrams



*squirrel*  
*cow*  
*rodent*  
*metatherian*  
*placental*  
*rat*  
*marsupial*  
*kangaroo*  
*mammal*



# Surface Features



- ▶ Capitalization:  $(\text{ISCAPS}(x_j), \text{ISCAPS}(x_i))$

E.g.,

<i>singer</i>		<i>actor</i>		<i>tiger</i>
↓	,	↓	,	↓
<i>Madonna</i>		<i>Tom Hanks</i>		<i>Bengal tiger</i>

- ▶ Ends-with:  $\text{ENDS WITH}(x_j, x_i)$

E.g.,

<i>nut</i>		<i>bee</i>		<i>salad</i>
↓	,	↓	,	↓
<i>chestnut</i>		<i>honeybee</i>		<i>potato salad</i>

- ▶ Contains, LCS, Suffix-match, Length-difference



# Semantic Features



## ► Web $n$ -gram Patterns and Counts

$P = \text{rodent}$



$C = \text{rat}$

Top 100 strings

### Web Ngrams

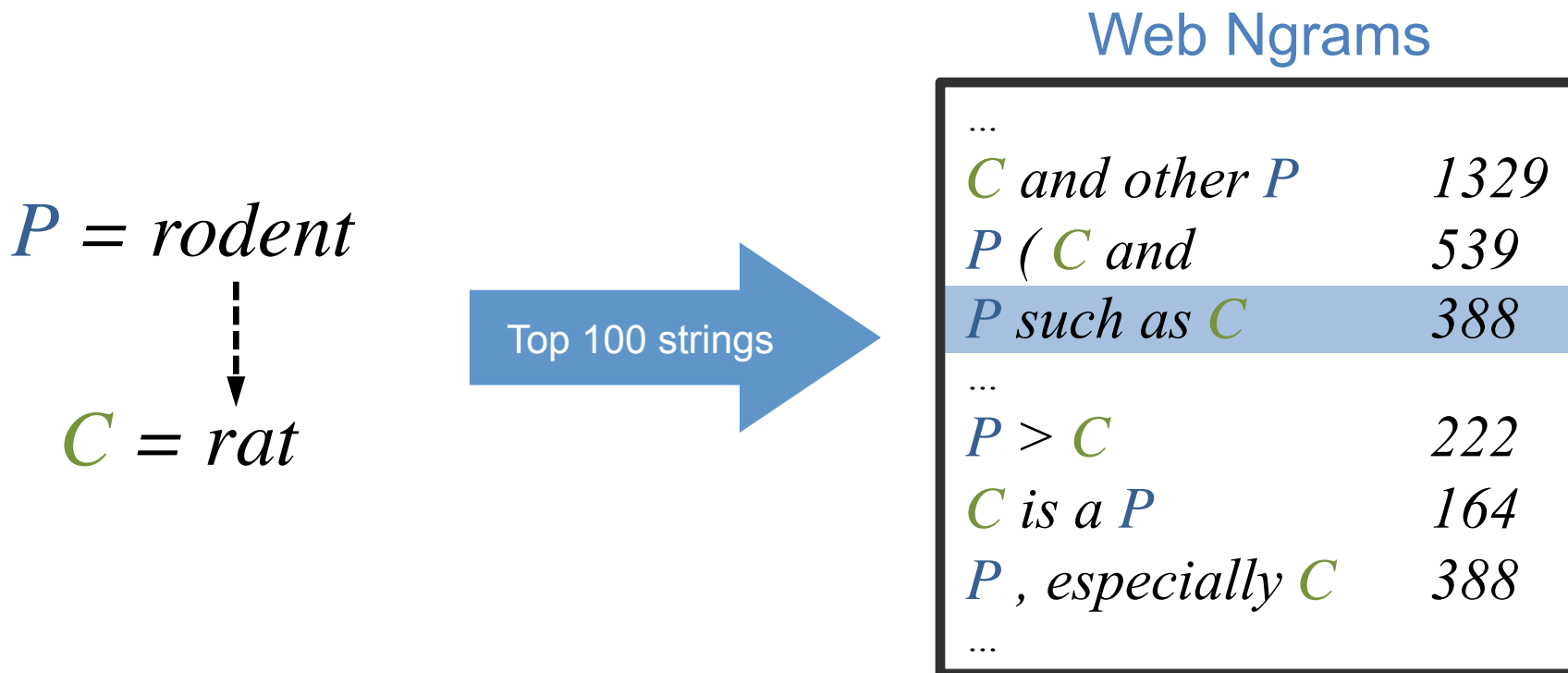
...	
$P w_1 w_2 w_3 C$	$x$
...	
$w_1 P w_2 w_3 C$	$x$
...	
$P w_1 w_2 C w_3$	$x$
...	
$P w_1 w_2 C$	$x$
...	
...	



# Semantic Features



## ► Web $n$ -gram Patterns and Counts



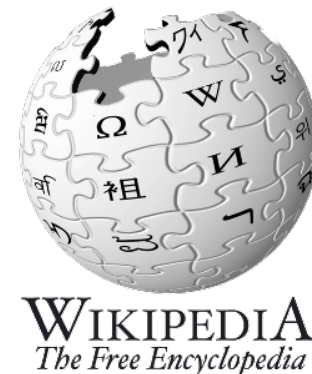
## ► Individual count, Unary patterns, Pattern order



# Semantic Features



- ▶ Wikipedia abstracts (for longer terms)



The **Rhode Island Red** is a breed of **chicken** (*Gallus gallus domesticus*). They are ...

... **Department of Justice** (DOJ), ... is the U.S. federal **executive department** ...

The **Gulf Stream**, together with its northern ... swift Atlantic **ocean current** that ...

- ▶ Features on Presence, Min-distance, and Patterns

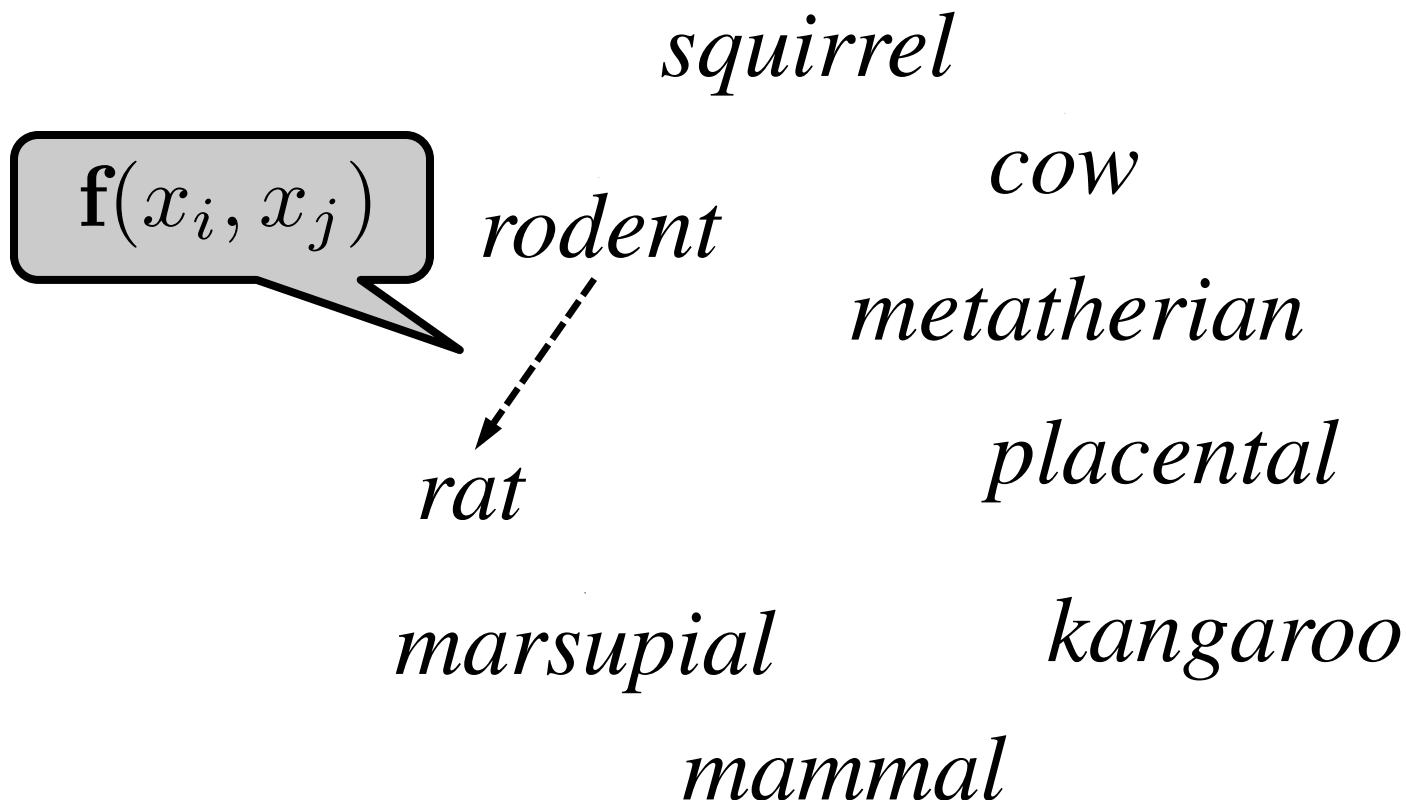


# Structured Taxonomy Induction



Hearst, 1992

- ▶ Each edge fires features with score  $s(y_{ij}) = \mathbf{w} \cdot \mathbf{f}(x_i, x_j)$

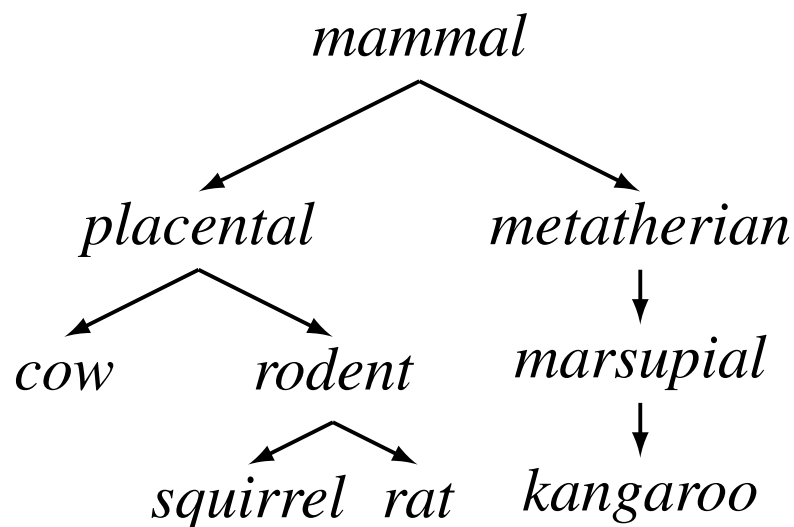




# Edge-factorization



- Chu-Liu-Edmonds: MST  $\hat{\mathbf{y}} = \operatorname{argmax}_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} \left\{ \sum_{y_{ij} \in \mathbf{y}} s(y_{ij}) \right\}$



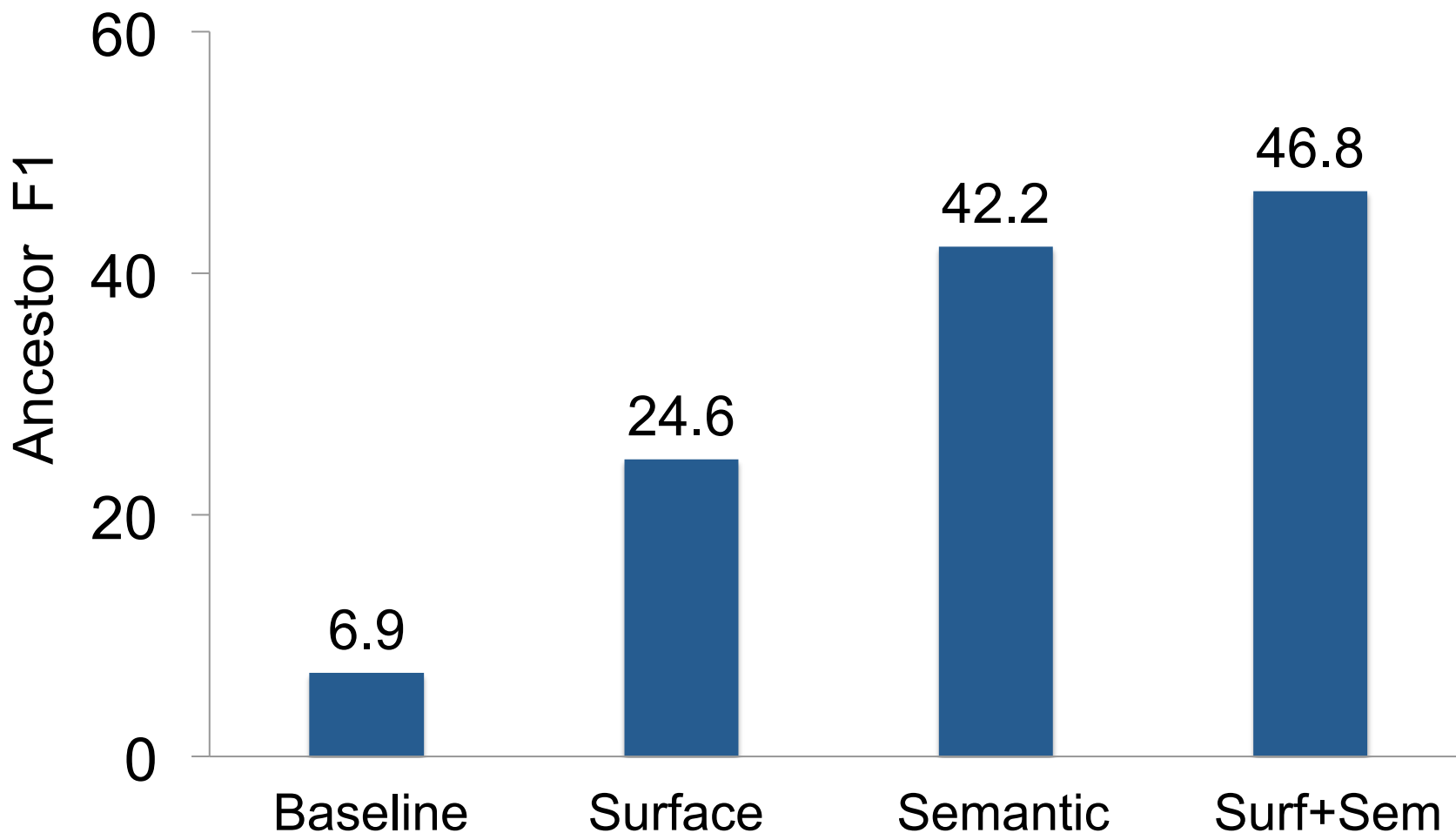
- Weights learned using standard gradient descent



# Results: 1<sup>st</sup> Order



- ▶ Setup: Train on a WordNet portion and reproduce the rest

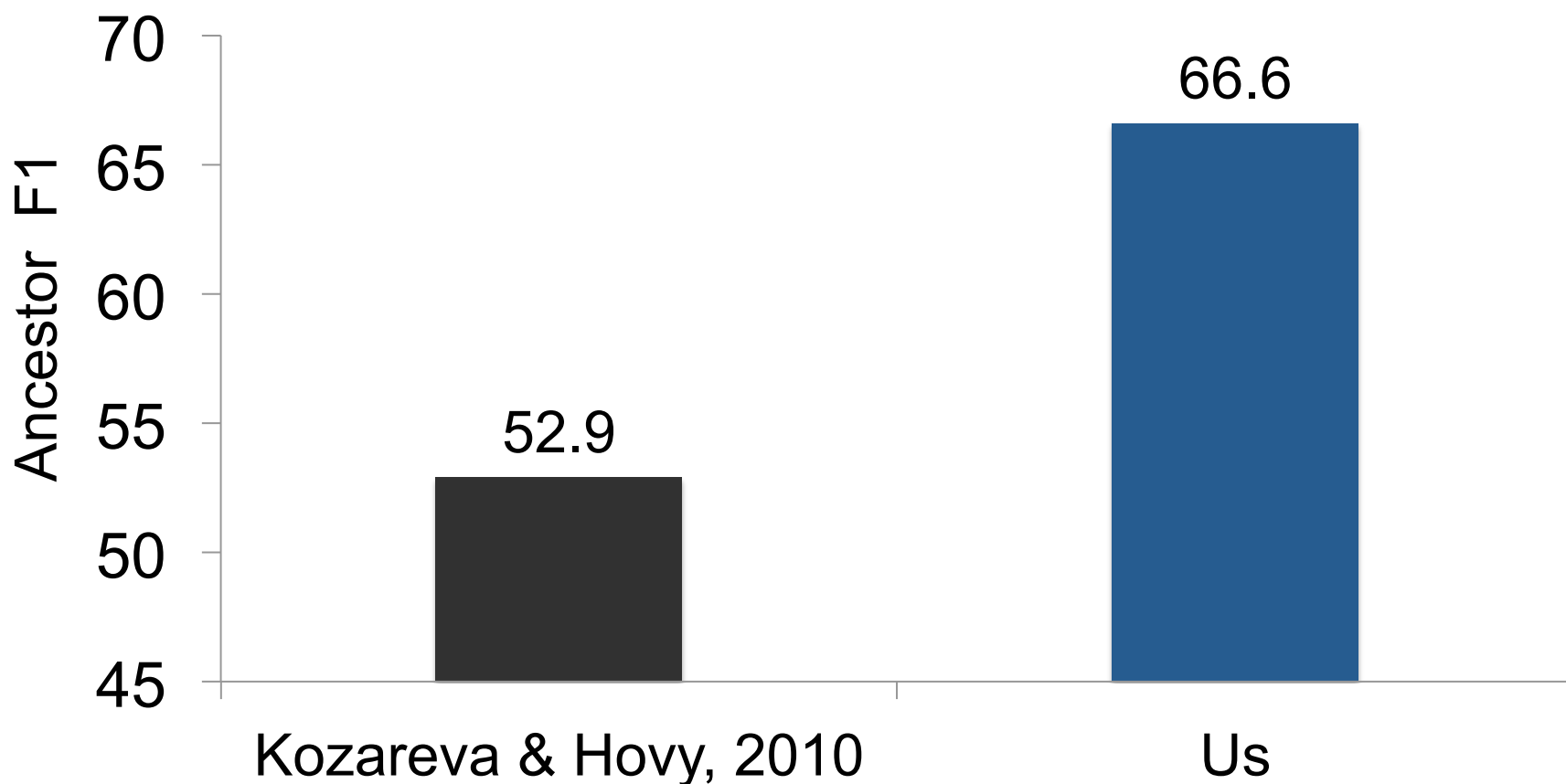




# Comparison Results



- ▶ Setup: Train on a WordNet portion and reproduce the rest





# Analysis: Learned Edge Features



Hearst, 1992

## ► High-weight edge pattern examples

<i>C and other P</i>	<i>&gt; P &gt; C</i>
<i>C , P of</i>	<i>C is a P</i>
<i>C , a P</i>	<i>P , including C</i>
<i>C or other P</i>	<i>P ( C</i>
<i>C : a P</i>	<i>C , american P</i>
<i>C - like P</i>	<i>C , the P</i>

*rats and other rodents*



# Analysis: Learned Edge Features



Hearst, 1992

## ► High-weight edge pattern examples

<i>C and other P</i>	<i>&gt; P &gt; C</i>
<i>C , P of</i>	<i>C is a P</i>
<i>C , a P</i>	<i>P , including C</i>
<i>C or other P</i>	<i>P ( C</i>
<i>C : a P</i>	<i>C , american P</i>
<i>C - like P</i>	<i>C , the P</i>

*electronics > office electronics > shredders*



# Analysis: Learned Edge Features



Hearst, 1992

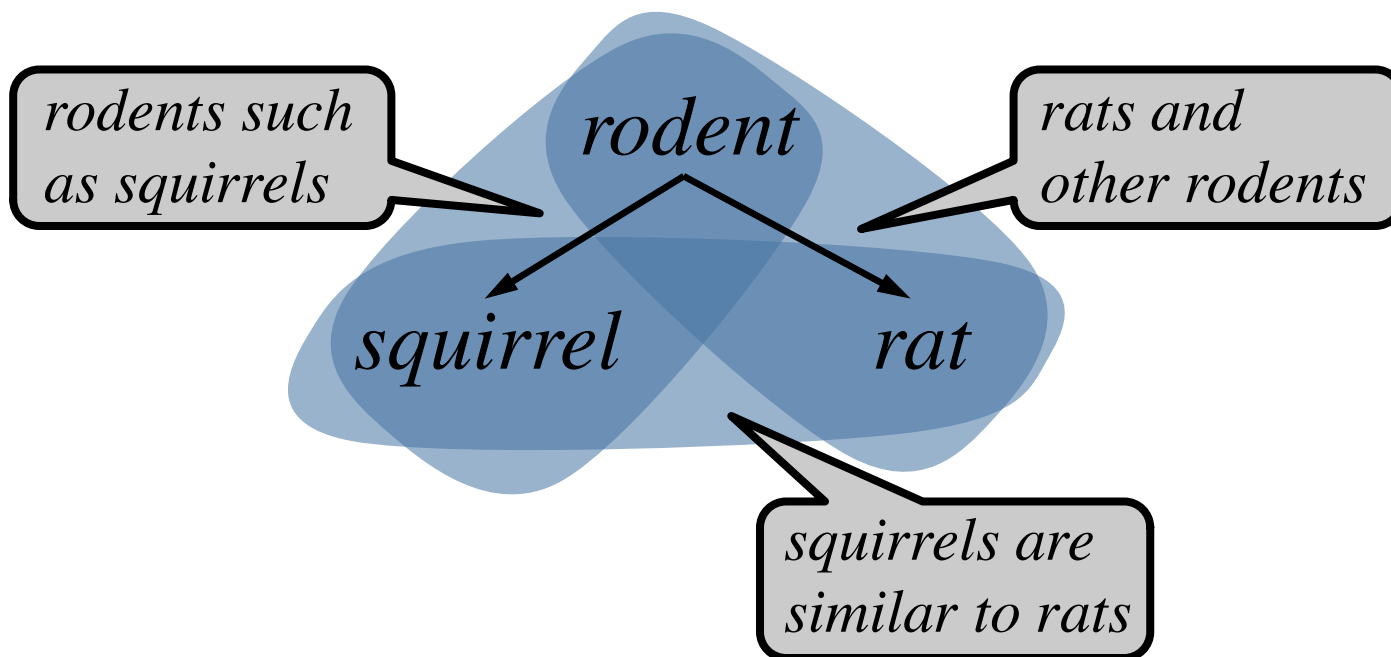
## ► High-weight edge pattern examples

<i>C and other P</i>	<i>&gt; P &gt; C</i>
<i>C , P of</i>	<i>C is a P</i>
<i>C , a P</i>	<i>P , including C</i>
<i>C or other P</i>	<i>P ( C</i>
<i>C : a P</i>	<i>C , american P</i>
<i>C - like P</i>	<i>C , the P</i>

*Michael Jackson, American singer*

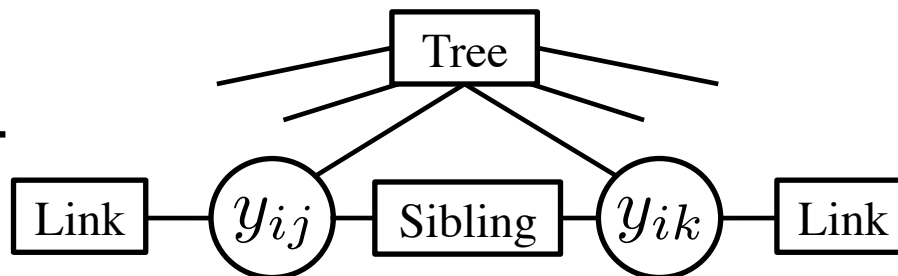


# Higher Order (Siblinghood)



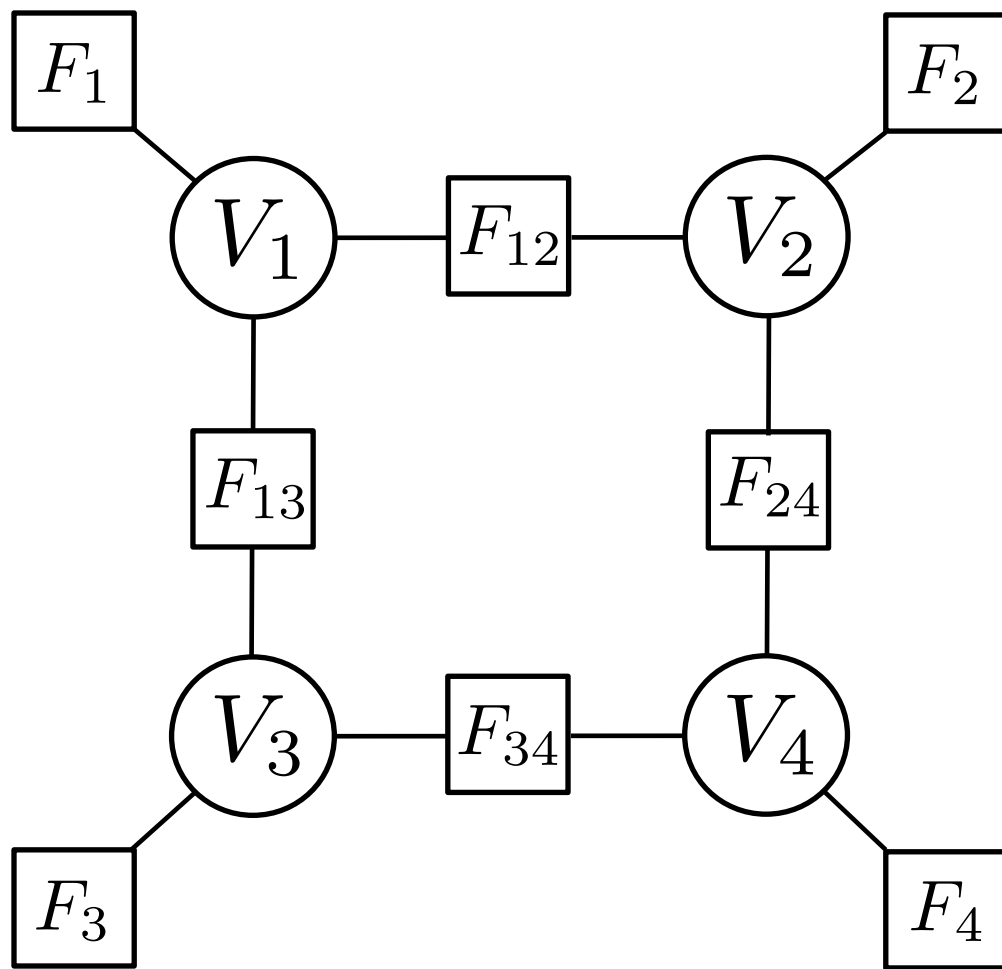
NP-hard!!

Use factor graphs and  
loopy belief propagation...





# Factor Graph Formulation



$$P(V) \propto \prod_F F(V_F)$$



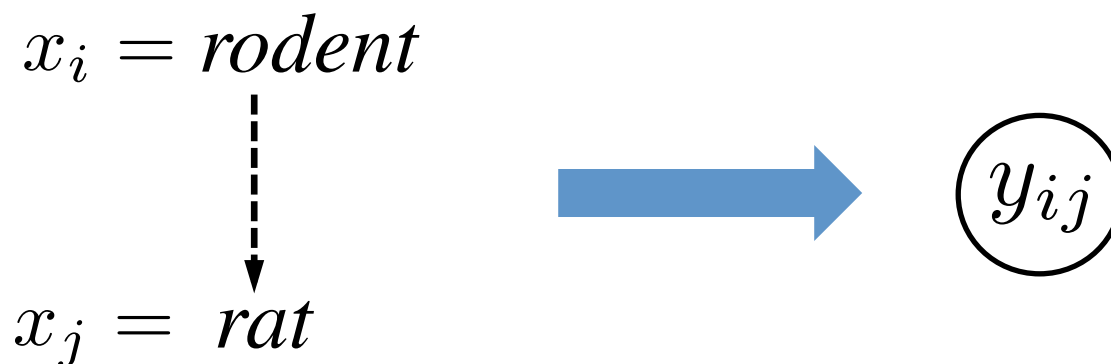
# Factor Graph Formulation



- ▶ Given the input term set  $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$ , we want

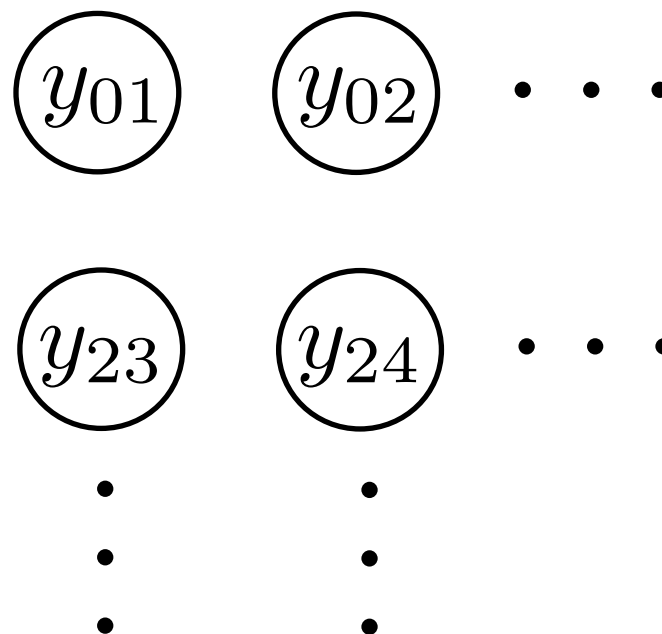
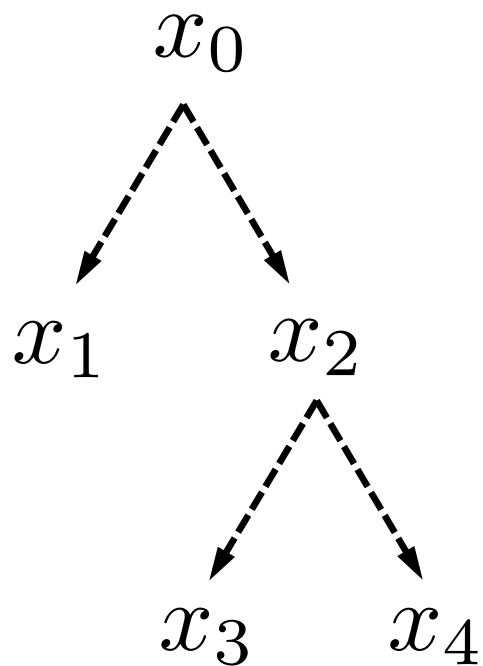
$$P(\mathbf{y}|\mathbf{x}) \propto \prod_F \phi_F(\mathbf{y})$$

- ▶ Each potential taxonomy edge  $x_i \rightarrow x_j$  is a variable  $y_{ij}$



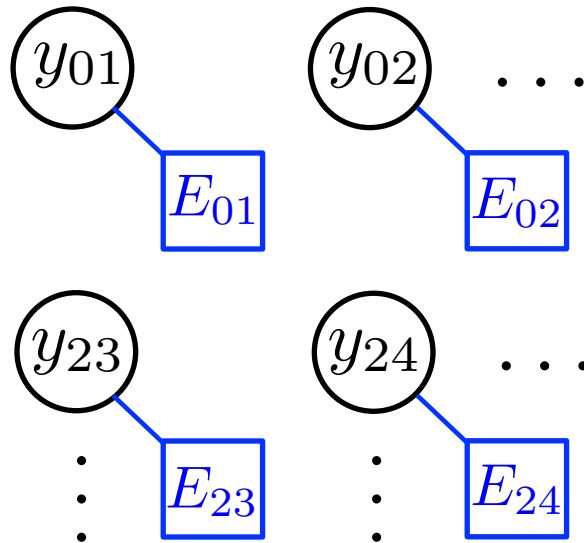


# Variables





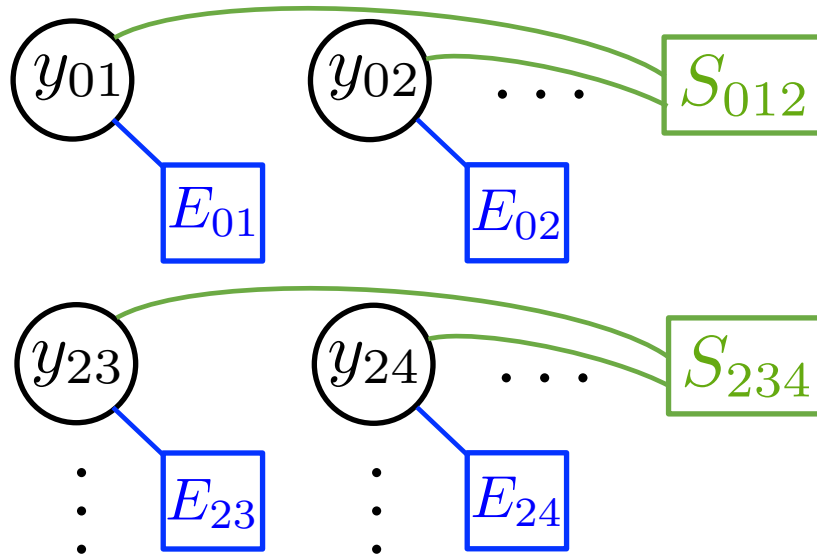
# Edge Factors



$$\phi_{E_{ij}}(y_{ij}) = \begin{cases} \exp(\mathbf{w} \cdot \mathbf{f}(x_i, x_j)) & y_{ij} = \text{ON} \\ \exp(0) = 1 & y_{ij} = \text{OFF} \end{cases}$$



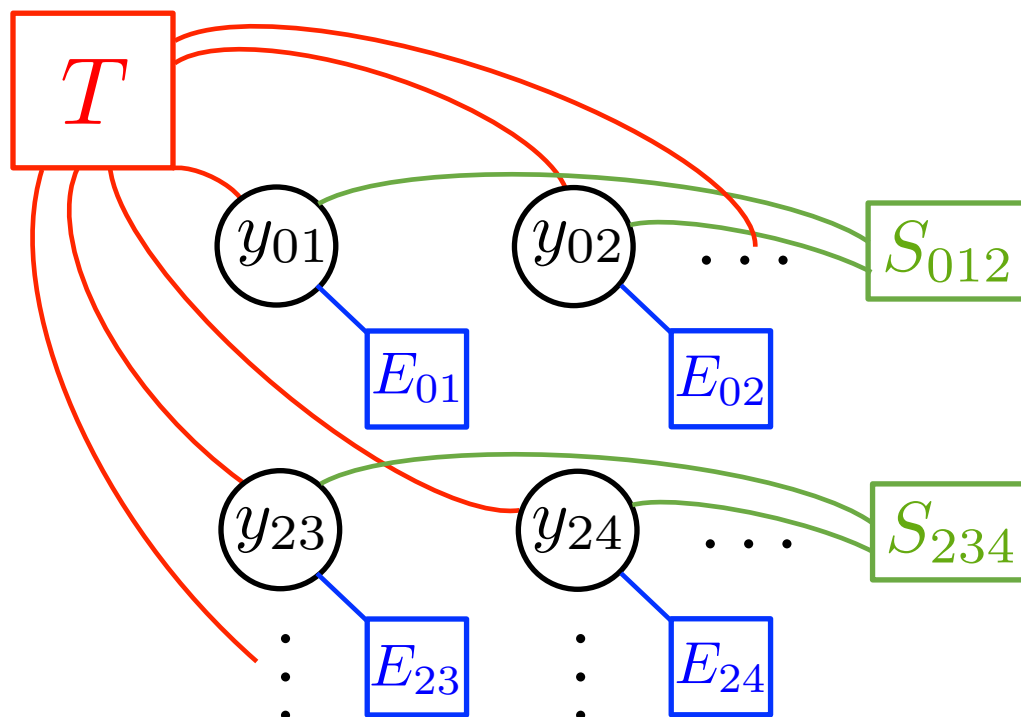
# Sibling Factors



$$\phi_{S_{ijk}}(y_{ij}, y_{ik}) = \begin{cases} \exp(\mathbf{w} \cdot \mathbf{f}(x_i, x_j, x_k)) & y_{ij} = y_{ik} = \text{ON} \\ 1 & \text{otherwise} \end{cases}$$



# Tree Factor



$$\phi_T(\mathbf{y}) = \begin{cases} 1 & \mathbf{y} \text{ forms a legal taxonomy tree} \\ 0 & \text{otherwise} \end{cases}$$



# Model Score



$$P(\mathbf{y}|\mathbf{x}) \propto \prod_F \phi_F(\mathbf{y}) \propto \begin{cases} \exp(\mathbf{w} \cdot \mathbf{f}(\mathbf{y})) & \mathbf{y} \text{ is a tree} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{f}(\mathbf{y}) = \underbrace{\sum_{\substack{i,j \\ y_{ij}=\text{ON}}} \mathbf{f}(x_i, x_j)}_{\text{Edge features}} + \underbrace{\sum_{\substack{i,j,k \\ y_{ij}=y_{ik}=\text{ON}}} \mathbf{f}(x_i, x_j, x_k)}_{\text{Sibling features}}$$



# Inference



- ▶ 2 main inference tasks:
  - ▶ learn  $\mathbf{w}$  (expected feature counts)
  - ▶ decode (select a taxonomy tree)
- ▶ Each needs marginals of edges and triples being ON
- ▶ One natural way to compute marginals in factor graph:  
Belief Propagation (MacKay, 2003)



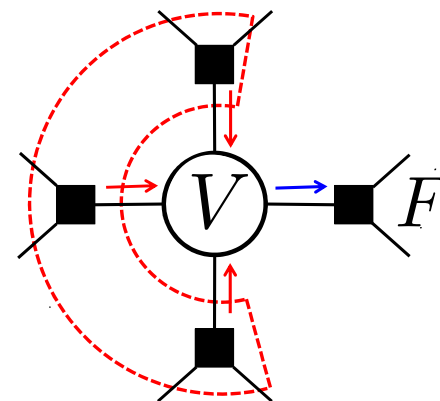
# Inference: Belief Propagation



Smith and Eisner, 2008; Burkett and Klein, 2012 (tutorial); Gormley and Eisner, 2014 (tutorial)

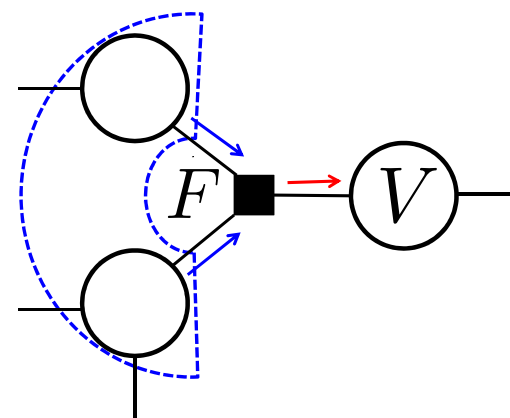
## ► Message from variables to factors:

$$m_{V \rightarrow F}(v) \propto \prod_{F' \in N(V) \setminus \{F\}} m_{F' \rightarrow V}(v)$$



## ► Message from factors to variables:

$$m_{F \rightarrow V}(v) \propto \sum_{\mathcal{X}_F, \mathcal{X}_F[V]=v} \phi_F(\mathcal{X}_F) \prod_{V' \in N(F) \setminus V} m_{V' \rightarrow F}(\mathcal{X}_F[V'])$$





# Inference: Belief Propagation



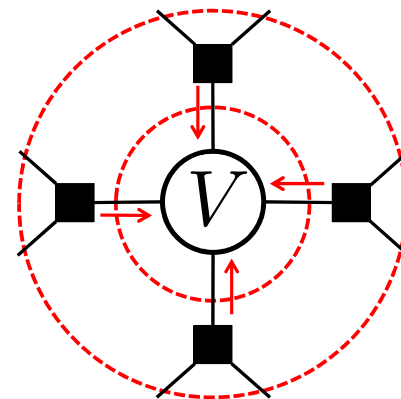
Smith and Eisner, 2008; Burkett and Klein, 2012 (tutorial); Gormley and Eisner, 2014 (tutorial)

- ▶ Messages from tree factor exponentially slow!

→  $O(n^3)$  Matrix Tree Theorem (Tutte, 1984)

- ▶ Marginal beliefs:

$$b_V(v) \propto \prod_{F \in N(V)} m_{F \rightarrow V}(v)$$



- ▶ Loopy belief propagation (sibling factors introduce cycles)



# Learning



- ▶ Gradient-based maximum likelihood training to learn  $w$
- ▶ Run loopy BP to get approximate marginals
- ▶ Compute expected feature counts and gradients
- ▶ Plug into any gradient optimizer – we use AdaGrad (Duchi et al., 2011)



# Decoding



Smith and Eisner, 2008

- ▶ After learning  $w$ , run BP again to get marginal beliefs
- ▶ Set edge-scores = belief-odds-ratio =  $\frac{b_{Y_{ij}}(\text{on})}{b_{Y_{ij}}(\text{off})}$
- ▶ Run MST algorithm to get minimum Bayes risk tree



# Sibling Features



- ▶ Consider each potential sibling pair  $(x_j, x_k)$  in factor  $S_{ijk}$
- ▶ Fire similar Web  $n$ -gram and Wikipedia features

$C_1$   $\longleftrightarrow$   $C_2$   
(squirrel) (rat)

Top 100 strings

## Web Ngrams

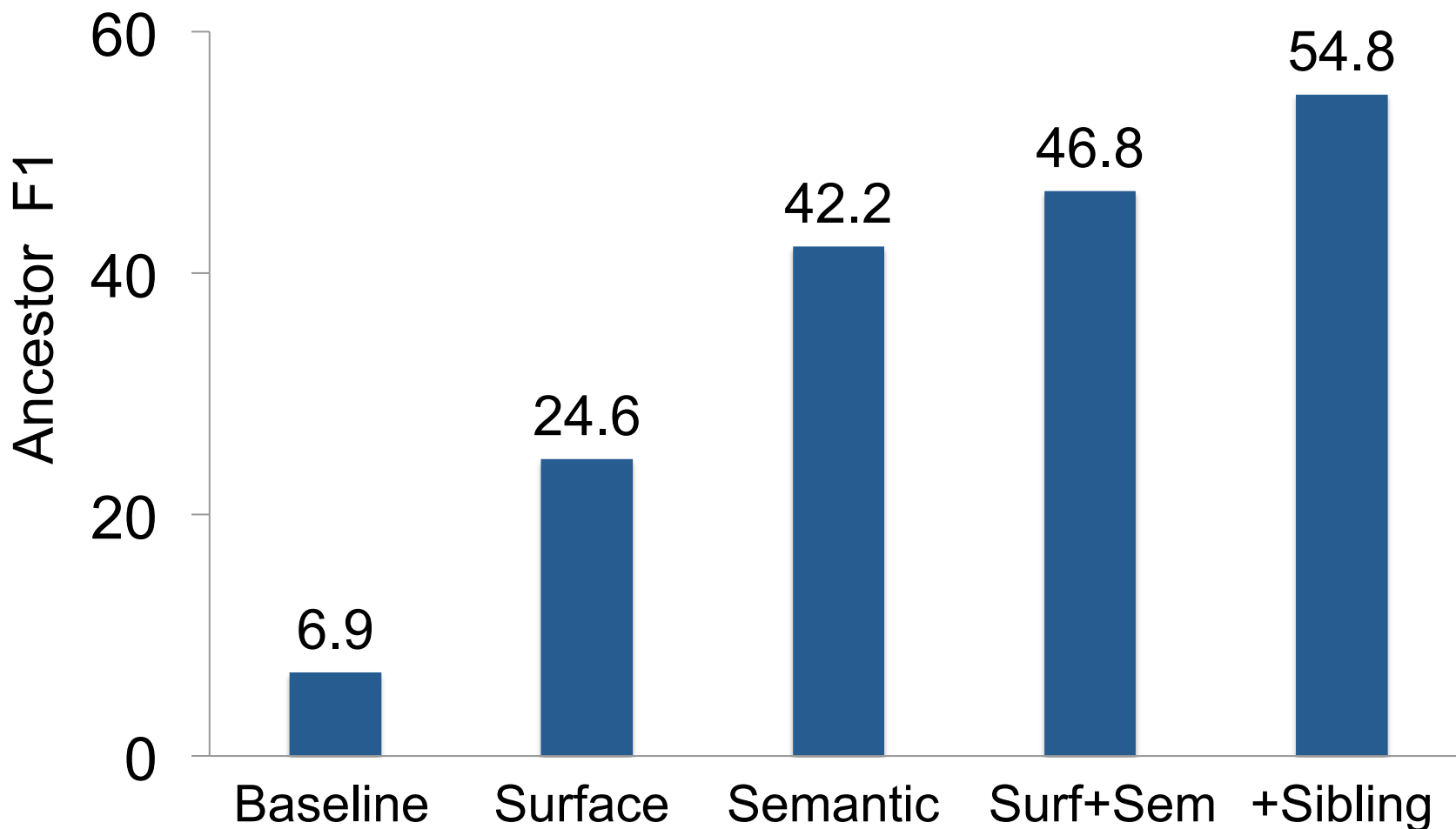
...	
$C_1 w_1 w_2 w_3 C_2$	$x$
...	
$w_1 C_1 w_2 w_3 C_2$	$x$
...	
$C_1 w_1 w_2 C_2 w_3$	$x$
...	
$C_1 w_1 w_2 C_2$	$x$
...	
...	



# Results: Adding Siblings



- ▶ Setup: Train on a WordNet portion and reproduce the rest





# Analysis: Learned Sibling Features



- ▶ High-weight sibling pattern examples

$C_1$  and  $C_2$   
 $C_1$  or  $C_2$  of  
,  $C_1$ ,  $C_2$  and  
the  $C_1 / C_2$

$C_1$ ,  $C_2$  (  
 $C_1$  and / or  $C_2$   
either  $C_1$  or  $C_2$   
<s>  $C_1$  and  $C_2$  </s>



# Conclusion



- ▶ Structured learning for taxonomy induction
- ▶ No lexicalized features possible, so learned external pattern features from Web  $n$ -grams and Wikipedia
- ▶ Incorporated sibling information via 2<sup>nd</sup> order factors and loopy BP
- ▶ Strong improvements on WordNet corpora

# Thank you!



# Questions?