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

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

$squirrel$

$rodent$

$rat$

$metatherian$

$placental$

$marsupial$

$kangaroo$

$mammal$
Need features for terms that we have never seen before!
Taxonomy Induction

Need features for terms that we have never seen before!

Web Ngrams

C and other P → x

squirrel

rodent

cow

metatherian

placental

marsupial

kangaroo

mammal
Taxonomy Induction

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

Web Ngrams:

- rodents such as rats → x

Diagram:

- Squirrel
- Cow
- Metatherian
- Placental
- Kangaroo
- Mammal
- Rodent
- Rat
- Marsupial
Taxonomy Induction

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

Web Ngrams

\( P \text{ such as } C \rightarrow x \)

squirrel

rodent

cow

metatherian

placental

marsupial

kangaroo

mammal
Need features for terms that we have never seen before!
Need features for terms that we have never seen before!

Web Ngrams

\[ P \ast \ast \ast C \rightarrow x \]
Surface Features

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

  E.g.,
  
  singer,   actor,   tiger
  Madonna,  Tom Hanks,  Bengal tiger

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

  E.g.,
  
  nut,   bee,   salad
  chestnut,  honeybee,  potato salad

- **Contains, LCS, Suffix-match, Length-difference**
Semantic Features

- Web $n$-gram Patterns and Counts

$P = \text{rodent}$
$C = \text{rat}$

Web Ngrams

- $P \ w_1 \ w_2 \ w_3 \ C \ x$
- $w_1 \ P \ w_2 \ w_3 \ C \ x$
- $P \ w_1 \ w_2 \ C \ w_3 \ x$
- $w_1 \ P \ w_2 \ C \ x$
- $P \ w_1 \ w_2 \ C \ x$
- $\ldots$
- $\ldots$
Semantic Features

- Web $n$-gram Patterns and Counts

$P = \text{rodent}$
$C = \text{rat}$

Web Ngrams

- $C$ and other $P$: 1329
- $P$ (C and): 539
- $P$ such as $C$: 388

Top 100 strings

- $P > C$: 222
- $C$ is a $P$: 164
- $P$, especially $C$: 388

- 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

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

Hearst, 1992
Edge-factorization

- Chu-Liu-Edmonds: MST \( \hat{y} = \arg\max_{y \in \mathcal{Y}(x)} \left\{ \sum_{y_{ij} \in y} s(y_{ij}) \right\} \)

![Mammal tree diagram]

- Weights learned using standard gradient descent
Results: 1\textsuperscript{st} Order

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

Setup: Train on a WordNet portion and reproduce the rest

| Ancestor F1 | Kozareva & Hovy, 2010 | Us |
|-------------|------------------------|----|
|             | 52.9                   | 66.6 |
## Analysis: Learned Edge Features

- **High-weight edge pattern examples**

| C and other P                                                                 | > P > C                                                                 |
|-------------------------------------------------------------------------------|------------------------------------------------------------------------|
| C, P of                                                                      | C is a P                                                                |
| C, a P                                                                       | P, including C                                                         |
| C or other P                                                                 | P ( C                                                                  |
| C : a P                                                                      | C, american P                                                          |
| C - like P                                                                   | C, the P                                                               |

- *Hearst, 1992*

- *rats and other rodents*
**Analysis: Learned Edge Features**

- **High-weight edge pattern examples**

| C and other P | > P > C |
|---------------|---------|
| C, P of C, a P | C is a P |
| C or other P  | P, including C |
| C : a P       | P ( C |
| C - like P    | C, american P |

| electronics > office electronics > shredders |
### Analysis: Learned Edge Features

#### High-weight edge pattern examples

| C and other P | > P > C |
|---------------|---------|
| C, P of C, a P | C is a P |
| C or other P | P, including C |
| C : a P | P ( C |
| C - like P | C, american P |
|             | C, the P |

*Michael Jackson, American singer*
Higher Order (Siblinghood)

rodents such as squirrels

rat

squirrel

rats and other rodents

squirrels are similar to rats

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 \( x = \{x_1, x_2, \ldots, x_n\} \), we want

\[
P(y|x) \propto \prod_{F} \phi_F(y)
\]

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

\[
x_i = \text{rodent}
\]

\[
x_j = \text{rat}
\]
Variables

\[ \begin{align*}
x_0 & \quad (y_{01}, y_{02}, \ldots) \\
x_1 \quad x_2 & \quad (y_{23}, y_{24}, \ldots) \\
x_3 \quad x_4 & \\
\end{align*} \]
### Edge Factors

\[
\phi_{E_{ij}}(y_{ij}) = \begin{cases} 
\exp(w \cdot f(x_i, x_j)) & y_{ij} = \text{ON} \\
\exp(0) = 1 & y_{ij} = \text{OFF}
\end{cases}
\]
Thus, our model has the form of a standard log-linear scoring function:

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

where \( f(x_i, x_j, x_k) \) is a feature function.
Thus, our model has the form of a standard log-linear model, as shown in Equation 0.

Note that by substituting our model's factor scores into the above equation, we get:

\[ \phi_T(y) = \begin{cases} 
1 & y \text{ forms a legal taxonomy tree} \\
0 & \text{otherwise} 
\end{cases} \]
Model Score

\[
P(y|x) \propto \prod_{F} \phi_F(y) \propto \begin{cases} 
\exp(w \cdot f(y)) & y \text{ is a tree} \\
0 & \text{otherwise}
\end{cases}
\]

\[
f(y) = \sum_{i,j} f(x_i, x_j) + \sum_{i,j,k} f(x_i, x_j, x_k)
\]

Edge features

Sibling features
2 main inference tasks:

- learn \( w \) (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:**

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

- **Message from factors to variables:**

  \[
  m_{F\rightarrow V}(v) \propto \sum_{x_F, x_F[V]=v} \phi_F(x_F) \prod_{V' \in \text{N}(F) \setminus V} m_{V'\rightarrow F}(x_F[V'])
  \]
Inference: Belief Propagation

Messages from tree factor exponentially slow!
\[ \rightarrow O(n^3) \] 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

- 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

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
... 
```

Top 100 strings

(squirrel) \(\rightarrow\) (rat)
Results: Adding Siblings

Setup: Train on a WordNet portion and reproduce the rest

|          | Baseline | Surface | Semantic | Surf+Sem | +Sibling |
|----------|----------|---------|----------|----------|----------|
| Ancestor F1 | 6.9      | 24.6    | 42.2     | 46.8     | 54.8     |
### High-weight sibling pattern examples

| $C_1$ and $C_2$ | $C_1$, $C_2$ (either $C_1$ or $C_2$) |
|-----------------|----------------------------------------|
| $C_1$ or $C_2$ of $C_1$, $C_2$ and the $C_1$/$C_2$ | $<s>$ $C_1$ 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 2nd order factors and loopy BP
- Strong improvements on WordNet corpora
Thank you!

Questions?