Toward Tree Substitution Grammars with Latent Annotations

Francis Ferraro, Benjamin Van Durme and Matt Post
Motivation

• Treebanks are coarse in multiple dimensions
• Complementary solutions have been proposed
  • Latent annotations (symbol refinements)
  • Tree substitution grammars
• Can we combine these approaches?
Learning Latent Annotations
Learning Latent Annotations

PRP

it  her
he   we
me  we
they ...

Learning Latent Annotations

PRP

it her
he
me we
they ...

PRP

nominative

he we they

accusative

it her me
...

PRP

PRP$_2$

PRP$_5$
Learning Latent Annotations

• Can we compensate for PCFG permissiveness?
• Johnson (1998): parent annotation
• Klein and Manning (2003): linguistically motivated annotations
Learning Latent Annotations

- Can we learn them automatically?
- Split all categories equally and learn weights via EM (Matsuzaki et al., 2005)
- Iteratively improve refinements: “split-merge” framework (Petrov et al., 2006)
- Intuitive lexical clusterings
Tree Substitution Grammars

VBD  NP  NP
    give  PRP  DT  NN

him  the  report
Tree Substitution Grammars

- Non-terminals rewrite as tree fragments
Tree Substitution Grammars

• Non-terminals rewrite as tree fragments
Tree Substitution Grammars

- Non-terminals rewrite as tree fragments
- Extended domain of locality
- Capture long-range grammatical dependencies
Tree Substitution Grammars

- Non-terminals rewrite as tree fragments
- Extended domain of locality
- Capture long-range grammatical dependencies
Tree Substitution Grammars

- Learning TSGs is not straightforward
- DOP: All fragments (Bod, 1993)
  - $\times$ Even with heuristic selection, get large, over-fit grammars (Bod, 2001)
Learning TSGs is not straightforward

DOP: All fragments (Bod, 1993)

Even with heuristic selection, get large, over-fit grammars (Bod, 2001)

Non-parametric: DP Prior (Cohn et al., 2009; Post et al., 2009)

Compact fragments

Non-deterministic

Complex to implement.
Framework

• Build within the Berkeley parser codebase

• Make the root of every internal depth-one subtree unique

• Place the entirety of the TSG weight on the root depth-one rule.
Framework

- Build within the Berkeley parser codebase
- Make the root of every internal depth-one subtree unique
- Place the entirety of the TSG weight on the root depth-one rule.
Framework

- Build within the Berkeley parser codebase
- Make the root of every internal depth-one subtree unique
- Place the entirety of the TSG weight on the root depth-one rule.

![Diagram of grammatical structures]
Framework

- Build within the Berkeley parser codebase
- Make the root of every internal depth-one subtree unique
- Place the entirety of the TSG weight on the root depth-one rule.
Framework

- Build within the Berkeley parser codebase
- Make the root of every internal depth-one subtree unique
- Place the entirety of the TSG weight on the root depth-one rule.
Algorithm Overview

Split symbols in two

Run EM

Merge back some refinements

Run EM

Petrov et al., 2006
Algorithm Overview

**Split** symbols in two

Run **EM**

**Merge** back some refinements

Run **EM**

**Couple** existing fragments

Run **EM**

Petrov et al., 2006
Control Exponential Growth

• Use binary trees

• Forbid multiple frontier nodes from simultaneously becoming internal nodes ("chained" couplings)

• Allow couplings only if permitted by a constraint set \( \mathcal{C} \).
Control Exponential Growth

- Use binary trees
- Forbid multiple frontier nodes from simultaneously becoming internal nodes ("chained" couplings)
- Allow couplings only if permitted by a constraint set $\mathbf{c}$. 
Coupling
Coupling

Given a grammar $\mathfrak{g}$, constraint set $\mathfrak{c}$:
Coupling

Given a grammar $G$, constraint set $c$: 

![Diagram of a grammar structure]
Coupling

Given a grammar $G$, constraint set $C$:
Coupling

Given a grammar \( G \), constraint set \( C \):

1. Construct a grammar \( G' \) from \( G \) and allowed couplings from \( C \)

\[
G' = G \cup \{X \circ Y \in C \mid X \in G\}
\]
Coupling

Given a grammar $G$, constraint set $C$:

1. Construct a grammar $G'$ from $G$ and allowed couplings from $C$

$$G' = G \cup \{X \circ Y \in C \mid X \in G\}$$
Coupling

Given a grammar $G$, constraint set $C$:

1. Construct a grammar $G'$ from $G$ and allowed couplings from $C$

$G' = G \cup \{X \circ Y \in C \mid X \in G\}$
Given a grammar $G$, constraint set $C$:

1. Construct a grammar $G'$ from $G$ and allowed couplings from $C$

$$G' = G \cup \{X \circ Y \in C \mid X \in G\}$$
Coupling

Given a grammar $G$, constraint set $C$:

I. Construct a grammar $G'$ from $G$ and allowed couplings from $C$

$$G' = G \cup \{X \circ Y \in C \mid X \in G\}$$
Given a grammar $G$, constraint set $C$:

1. Construct a grammar $G'$ from $G$ and allowed couplings from $C$

$$G' = G \cup \{ X \circ Y \in C \mid X \in G \}$$
Coupling

Given a grammar $G$, constraint set $C$:

I. Construct a grammar $G'$ from $G$ and allowed couplings from $C$

$$G' = G \cup \{X \circ Y \in C \mid X \in G\}$$
Coupling

Given a grammar $G$, constraint set $C$:

1. Construct a grammar $G'$ from $G$ and allowed couplings from $C$

$$G' = G \cup \{X \circ Y \in C \mid X \in G\}$$
Coupling

Given a grammar $G$, constraint set $C$:

1. Construct a grammar $G'$ from $G$ and allowed couplings from $C$

   $$G' = G \cup \{X \circ Y \in C \mid X \in G\}$$

2. Estimate initial $G'$ fragment weights
Coupling

Given a grammar $G$, constraint set $C$:

1. Construct a grammar $G'$ from $G$ and allowed couplings from $C$

$$G' = G \cup \{X \circ Y \in C \mid X \in G\}$$

2. Estimate initial $G'$ fragment weights
   - Uniform redistribution not appropriate
Coupling

Given a grammar $G$, constraint set $C$:

1. Construct a grammar $G'$ from $G$ and allowed couplings from $C$
   \[ G' = G \cup \{X \circ Y \in C \mid X \in G\} \]

2. Estimate initial $G'$ fragment weights
   - Uniform redistribution not appropriate

3. Fit weights of $G'$ via inside/outside
Deriving the Constraint Set

- Deterministically count compact TSG fragments
- Iteratively extract the $K$ most frequent subtrees of size $R$
- $R$, $K$ enforce sparsity and control what can and cannot couple

```
EXTRACTFRAGMENTS(R,K)

S ← ∅
F(1,K) ← top K CFG rules used
for $r = 2$ to $R$ do
    S ← \{F ∈ F(r-1, K), extended by 1 rule\}
    F(r,K) ← top K elements of F(r-1, K) U S
end for
```
Qualitative Evaluation
Qualitative Evaluation

- Korean Treebank v2.0
- Sect. 2-3 of Penn Treebank (WSJ)
Qualitative Evaluation

• Korean Treebank v2.0
• Sect. 2-3 of Penn Treebank (WSJ)
  • Petrov et al. (2011)’s universal tag set
Qualitative Evaluation

- Korean Treebank v2.0
- Sect. 2-3 of Penn Treebank (WSJ)
  - Petrov et al. (2011)’s universal tag set
  - Replace all preterminals with a single symbol, X.
Qualitative Evaluation

• Korean Treebank v2.0
• Sect. 2-3 of Penn Treebank (WSJ)
• Petrov et al. (2011)’s universal tag set
• Replace all preterminals with a single symbol, X.
WSJ, §2-3
WSJ, §2-3

Modals

\[
\begin{array}{c}
\overline{S_2} \\
\overline{S} & \overline{VP} \\
NP_0 & VP_0 & MD & \overline{VP}_0 \\
\text{will} \\
\end{array}
\]
WSJ, §2-3

Modals

Perfectives
WSJ, §2-3

- **Modals**
  - $\overline{S}_2$
  - $\overline{S}$
  - $\overline{VP}_0$
  - $\overline{NP}_0$
  - Will

- **Perfectives**
  - $\overline{VP}_2$
  - $\overline{VBN}$
  - Been
  - $\overline{VP}_0$
  - $\overline{PP}_0$

- **Nominals**
  - $\overline{NP}_0$
  - $\overline{NNP}_0$
  - $\overline{NP}$
  - $\overline{NNP}_3$
  - $\overline{NNP}_1$
Universal Tag Set, WSJ, §2-3

<table>
<thead>
<tr>
<th></th>
<th>PRON</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>its, his, your</td>
</tr>
<tr>
<td>5</td>
<td>it, he, they</td>
</tr>
<tr>
<td>6</td>
<td>it, them, him</td>
</tr>
</tbody>
</table>
Universal Tag Set, WSJ, §2-3

<table>
<thead>
<tr>
<th></th>
<th>PRON</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>its, his, your</td>
</tr>
<tr>
<td>5</td>
<td>it, he, they</td>
</tr>
<tr>
<td>6</td>
<td>it, them, him</td>
</tr>
</tbody>
</table>

\[
\begin{array}{c}
\text{PRON} \\
1 & \text{its, his, your} \\
5 & \text{it, he, they} \\
6 & \text{it, them, him} \\
\end{array}
\]

\[
\begin{array}{c}
\overline{VP}_0 \\
\text{VERB}_0 & \text{NP} \\
\text{PRON}_6 & \\
\end{array}
\]
# Universal Tag Set, WSJ, §2-3

<table>
<thead>
<tr>
<th></th>
<th>PRON</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>its, his, your</td>
</tr>
<tr>
<td>5</td>
<td>it, he, they</td>
</tr>
<tr>
<td>6</td>
<td>it, them, him</td>
</tr>
</tbody>
</table>

```
VP₀
   / \  \
  VP  VERB₀
     /   \   /
    NP    PRON₀₆
```

```
VP₀
   / \  \
  VP  VERB₀
     /   \   /
    ADVP₀ VERB₀ NP₀
```
Thank you!