Efficient Third-Order Dependency Parsers

Terry Koo and Michael Collins
MIT CSAIL
Dependency Parsing

- Syntax represented by head-modifier dependencies
- Parsing is a search for the highest-scoring tree

\[ y^*(x) = \text{argmax}_{y \in \mathcal{Y}(x)} \text{SCORE}(x, y) \]
Factored (Graph-based) Parsing

Decompose \( \text{Score}(x, y) = \sum_{p \in y} \text{Score}(x, p) \)

John ate ice cream with sprinkles.
Higher-Order Factorizations


Vertical Context

Horizontal Context

dependency

h
m
Higher-Order Factorizations


Vertical Context

Horizontal Context
Higher-Order Factorizations

Vertical Context

Carreras (2007) Second-Order

Horizontal Context
Higher-Order Factorizations

Vertical Context

H. J. M. Grandchild
g h m

dependency
h m

Horizontal Context

<table>
<thead>
<tr>
<th>Factorization</th>
<th>Accuracy</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dep</td>
<td>90.9</td>
<td>$O(n^3)$</td>
</tr>
<tr>
<td>Dep+Sib</td>
<td>91.5</td>
<td>$O(n^3)$</td>
</tr>
<tr>
<td>Dep+Sib+Grand</td>
<td>92.0</td>
<td>$O(n^4)$</td>
</tr>
</tbody>
</table>
Higher-Order Factorizations

Vertical Context

Grandchild:
\[ g \rightarrow h \rightarrow m \]

Dependency:
\[ h \rightarrow m \]

Grand-sibling:
\[ g \rightarrow h \rightarrow s \rightarrow m \]

Sibling:
\[ h \rightarrow s \rightarrow m \]

Horizontal Context

Third-Order Model 1
Higher-Order Factorizations

Vertical Context

- Grandchild: \((g, h, m)\)
- Dependency: \((h, m)\)

Horizontal Context

- Grand-sibling: \((g, h, s, m)\)
- Sibling: \((h, s, m)\)
- Tri-sibling: \((h, t, s, m)\)

Third-Order Model 2
Higher-Order Factorizations

### Vertical Context
- grandchild: $g \rightarrow h \rightarrow m$
- dependency: $h \rightarrow m$

### Horizontal Context
- sibling: $h \rightarrow s \rightarrow m$
- tri-sibling: $h \rightarrow t \rightarrow s \rightarrow m$

<table>
<thead>
<tr>
<th>Factorization</th>
<th>Accuracy</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dep</td>
<td>90.9</td>
<td>$O(n^3)$</td>
</tr>
<tr>
<td>Dep+Sib</td>
<td>91.5</td>
<td>$O(n^3)$</td>
</tr>
<tr>
<td>Dep+Sib+Grand</td>
<td>92.0</td>
<td>$O(n^4)$</td>
</tr>
<tr>
<td>Model 1</td>
<td>93.0</td>
<td>$O(n^4)$</td>
</tr>
<tr>
<td>Model 2</td>
<td>92.9</td>
<td>$O(n^4)$</td>
</tr>
</tbody>
</table>

$O(n^3)$ and $O(n^4)$ represent the computational complexity of the factorization methods.
**First-Order Parser**

- Eisner (2000) algorithm: $O(n^3)$

**Complete Span**
- A “half-constituent”

**Incomplete Span**
- A dependency
Eisner (2000) algorithm: $O(n^3)$

Derivation of complete and incomplete spans:
First-Order Parsing Example

- Eisner (2000) algorithm: $O(n^3)$
First-Order Parsing Example

- Eisner (2000) algorithm: $O(n^3)$
First-Order Parsing Example

- Eisner (2000) algorithm: $O(n^3)$
First-Order Parsing Example

- Eisner (2000) algorithm: $O(n^3)$
First-Order Parsing Example

- Eisner (2000) algorithm: $O(n^3)$
Eisner (2000) algorithm: $O(n^3)$
Second-Order Sibling Parser

- McDonald (2006) and Eisner (1996): $O(n^3)$
- Introduce a third type of span:

**Sibling Span**
A pair of adjacent modifiers

$s \quad m$
Second-Order Sibling Parser

- McDonald (2006) and Eisner (1996): $O(n^3)$

- Scores sibling interactions
Second-Order Sibling Parser

- McDonald (2006) and Eisner (1996): $O(n^3)$

Scores sibling interactions
Second-Order Sibling Parser

McDonald (2006) and Eisner (1996): $O(n^3)$
Model 0

Model 0, all grandparents: $O(n^4)$

Complete G-Span
A “half-constituent” with its grandparent

Incomplete G-Span
A dependency with its grandparent

Superficially similar to parent annotation in CFGs
Model 0: Derivations

Model 0, all grandparents: \( O(n^4) \)

Grandparent indices propagated to smaller g-spans

4 active indices, runtime \( O(n^4) \)
Model 1

- Model 1, grand-siblings: $O(n^4)$
- Introduce a third type of span:

**Sibling G-Span**
A pair of adjacent modifiers with their shared head

$h\quad s\quad m$
Model 1: Grand-Sibling Scores

Model 1, grand-siblings: $O(n^4)$

Scores grand-sibling interactions
Model 1: Grand-Sibling Scores

Model 1, grand-siblings: $O(n^4)$

Scores grand-sibling interactions
Model 1: Derivations

**Model 1, grand-siblings:** $O(n^4)$
Model 2

- Model 2, grand-siblings and tri-siblings: $O(n^4)$
- Introduce a fourth type of span:

\textit{Incomplete S-Span}

A dependency with its next-inner modifier

\[ h \quad s \quad m \]
Model 2, grand-siblings and tri-siblings: $O(n^4)$

Scores tri-sibling interactions
Model 2: Tri-Sibling Scores

Model 2, grand-siblings and tri-siblings: $O(n^4)$

Scores tri-sibling interactions
Model 2: Grand-Sibling Scores

Model 2, grand-siblings and tri-siblings: $O(n^4)$

Scores grand-sibling interactions
Model 2: Grand-Sibling Scores

Model 2, grand-siblings and tri-siblings: $O(n^4)$

Scores grand-sibling interactions
Model 2: Derivations

Model 2, grand-siblings and tri-siblings: $O(n^4)$
Summary of Parsing Algorithms

- Model 0 parses an all-grandchildren factorization
- Model 1 parses an all-grand-siblings factorization
- Model 2 parses all-tri-siblings and some grand-siblings

All parsers require $O(n^4)$ time and $O(n^3)$ space

- Identical to Carreras (2007) second-order
- Models 1 and 2 are asymptotically fast:
  - Number of third-order parts is $\Omega(n^4)$
Parsing Experiments

- English Penn Treebank (Penn2Malt conversion)
- Czech Prague Dependency Treebank
- Averaged perceptron training
- Features based on words and POS tags
- Coarse-to-fine pruning (Carreras et al., 2008)
### English and Czech Parsing

<table>
<thead>
<tr>
<th>Parser</th>
<th>English</th>
<th>Czech</th>
</tr>
</thead>
<tbody>
<tr>
<td>McDonald and Pereira (2006)</td>
<td>91.5</td>
<td>85.2</td>
</tr>
<tr>
<td>Koo, Carreras and Collins (2008), Normal</td>
<td>92.0</td>
<td>86.1</td>
</tr>
<tr>
<td>Model 1</td>
<td>93.0</td>
<td>87.4</td>
</tr>
<tr>
<td>Model 2</td>
<td>92.9</td>
<td>87.4</td>
</tr>
<tr>
<td>Koo, Carreras and Collins (2008), Semisup</td>
<td>93.2</td>
<td>87.1</td>
</tr>
</tbody>
</table>

- Unlabeled attachment score on the test sets
- Third-order is comparable to semi-supervised features
Final Remarks

- Third-order factorizations can be parsed in $O(n^4)$
- Third-parsers work well in practice
- Possible extensions:
  - Recovering word senses or dependency labels
  - Full head automata: e.g., TAG-style parsing (Carreras et al., 2008)
  - Increasing context to fourth-order or more