Using the Parser

CSCI-GA.2590

Ralph Grishman
Ever Faster

• Change from CKY and graph-based parsers to transition-based parsers has led to large speed-ups
  – with little loss of performance
  – making full-sentence parsing viable for large corpora
Dependency Parsers

<table>
<thead>
<tr>
<th>Parser</th>
<th>LAS</th>
<th>UAS</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yara (arc eager – beam) (2015)</td>
<td>92.3</td>
<td></td>
<td>45 sent/sec</td>
</tr>
<tr>
<td>Stanford (NN) (2014)</td>
<td>90.7</td>
<td>92</td>
<td>1000 sent/sec</td>
</tr>
<tr>
<td>Yara (arc eager) (2015)</td>
<td>88</td>
<td>89.3</td>
<td>4000 sent / sec</td>
</tr>
</tbody>
</table>
Constituent Parsers

Methods developed to speed dependency parsers also applied to constituent parsers.

<table>
<thead>
<tr>
<th>Parser</th>
<th>F</th>
<th>speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Charniak (2000)</td>
<td>89.5</td>
<td>5.7 sent / sec</td>
</tr>
<tr>
<td>Northeastern (China)(2013) (shift reduce)</td>
<td>91.3</td>
<td>100 sent / sec</td>
</tr>
</tbody>
</table>
Using the Parses

• partial parse approach: match sequence of chunks

• full parse approach: match parse subtree

sell (subj person) (dobj object) (to person) \(\rightarrow\) ...

\[
\text{Mark reportedly marshmallows yesterday \(\rightarrow\) Mabel}
\]
Benefits of Parsing

- parsing captures head-argument-modifier relations directly
  - with a partial parser, must explicitly account for and skip over all irrelevant modifiers
  - Mary Smith, who last week announced the discovery of a new species of salamander, was named head of research.
Downside of Parsing

- Parsing errors may lead to missed extractions

Fred sold the plane ticket to Mary.

Parse obtained will not match a rule of the form
sell (subj person) (dobj object) (to person)
Dual Approach

• To recover from some parsing errors, combine results of partial-parse and full-parse systems.