Bayesian learning of a tree substitution grammar

Matt Post and Daniel Gildea • University of Rochester • Rochester, NY, USA

TSGs
- like CFGs, but nonterminals rewrite as subtrees of arbitrary size
- rule application is still context-free
- subsume context free grammars

<table>
<thead>
<tr>
<th></th>
<th>+</th>
<th>-</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFG</td>
<td>have annotated</td>
<td>small domain of</td>
</tr>
<tr>
<td></td>
<td>data, easy to</td>
<td>locality, too</td>
</tr>
<tr>
<td></td>
<td>train</td>
<td>much independence</td>
</tr>
<tr>
<td>TSG</td>
<td>larger rules</td>
<td>no annotated</td>
</tr>
<tr>
<td></td>
<td>capture long-</td>
<td>data, not clear</td>
</tr>
<tr>
<td></td>
<td>distance deps</td>
<td>how to learn</td>
</tr>
</tbody>
</table>

Learning a TSG
- we don’t have a manually annotated training corpus (like a Treebank with TSG derivations) from which we could just count rules
- heuristics produce large grammars with the wrong shape (see figure in right column)
- EM overfits and requires us to maintain explicit counts of an exponential number of subtrees
- collapsed Gibbs sampling with a nonparametric prior
- avoids overfitting: Dirichlet Process (DP) prior discourages larger subtrees unless the data warrants it
- avoids counting: sample derivations instead of maintaining explicit subtree counts

Derivations as segmentations
- annotate the nodes of a parse tree with flags marking rule boundaries; this induces a set of rules (and a derivation)
- DP helps overfitting: for each nonterminal X,
  \[ g_X \sim D P(G_X, \alpha) \]
  \[ G_X(t) = \prod_{r_i \in t} P_{SMLG}(r_i) \]
  where
  - \( \alpha \) is a hyperparameter roughly controlling the variance
  - \( P_{SMLG} \) is a geometric distribution on the number of PCFG rules in the subtree
  - \( P_{SMLG} \) is the Treebank PCFG prob of each rule in the subtree
- we use collapsed Gibbs sampling with a DP prior: sample a grammar \( g \) from the posterior \( P(g | I) \) based on the Chinese Restaurant Process view of the DP
- this is done by iteratively considering each node of each tree in the training data and randomly joining or splitting the subtree rooted at that node from the subtree its parent is part of
- subtree probability \( \theta \) is given as
  \[ \theta(t) = \frac{\text{count}_z(t) + \alpha \text{count}_{\text{root}(t)}(t)}{|z(t)| + \alpha} \]
  where \( z(t) \) is the set of rewrites of root(t) in the current state of the rest of the corpus
- successfully used for segmentation tasks (Goldwater et al. 2009, DeNero et al. 2008)
- similar models were developed independently by Cohn et al. (2009) and O’Donnell et al. (2009)

Baseline grammars
- Treebank PCFG
- Bod (2001) “minimal subset”: all rules of height 1, plus 400K subtrees sampled at each height \( 2 \leq h \leq 14 \), minus unlexicalized subtrees of \( h \geq 6 \), minus lexicalized subtrees with more than 12 words
- spinal extraction heuristic: extract as one subtree the sequence of CFG rules from leaf upward sharing a head

Results: accuracy
We approximated the most probable parse with the most probable derivation.

<table>
<thead>
<tr>
<th>grammar</th>
<th>size</th>
<th>LP</th>
<th>LR</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCFG</td>
<td>46K</td>
<td>75.37</td>
<td>70.05</td>
<td>72.61</td>
</tr>
<tr>
<td>minimal subset</td>
<td>190K</td>
<td>80.30</td>
<td>78.10</td>
<td>79.18</td>
</tr>
<tr>
<td>(100, 0.7, 500)</td>
<td>2.55M</td>
<td>76.40</td>
<td>78.29</td>
<td>77.33</td>
</tr>
<tr>
<td>(100, 0.8, 500)</td>
<td>61K</td>
<td>81.95</td>
<td>81.76</td>
<td>81.85</td>
</tr>
<tr>
<td>(100, 0.9, 500)</td>
<td>59K</td>
<td>82.73</td>
<td>82.21</td>
<td>82.46</td>
</tr>
<tr>
<td>(100, 0.0, 500)</td>
<td>2.00M</td>
<td>82.81</td>
<td>82.01</td>
<td>82.40</td>
</tr>
<tr>
<td>(100, 0.0, 500)</td>
<td>1.13M</td>
<td>83.06</td>
<td>82.10</td>
<td>82.57</td>
</tr>
<tr>
<td>(100, 0.0, 500)</td>
<td>528K</td>
<td>83.17</td>
<td>82.91</td>
<td>82.53</td>
</tr>
</tbody>
</table>

Results: sampled rules
(S (VP (TO to) VP)) 0.106
(S (VP (TO to) (VP VBD NP))) 0.035
(S (NP (VP (VBD said) SBAR) .)) 0.029
(SBAR (IN that) S) 0.137
(SBAR S) 0.036
(SBAR (WHNP (WDT that)) S) 0.035
(SINV “S,” (VP (VBZ says) NP .)) 0.181
(SINV “S,” (VP (VBD said)) NP .) 0.176

Conclusion
We can efficiently learn compact TSGs that outperform heuristic approaches, and which take the expected shape in terms of histograms of subtree sizes.