Strongly Incremental Constituency Parsing with Graph Neural Networks

Kaiyu Yang, Jia Deng
Department of Computer Science, Princeton University

1. Introduction

- Parsing: a core task in NLP

- Shift-reduce parsing
  1. State: a stack of subtrees + the remaining sentence
  2. Actions: SHIFT and REDUCE

Unlike shift-reduce parsers, human parsing appears to be strongly incremental [1-3]

1. Exactly ONE token at each step: no more, no less
2. The state is a single connected tree for the partial sentence

- Contributions
  1. Attach-juxtapose: a novel transition system for strongly incremental parsing
  2. Theoretical results characterizing its capability and connections with existing shift-reduce systems
  3. A parser that generates actions using GNNs. State-of-the-art performance on Penn Treebank and Chinese Treebank

2. Attach-juxtapose Transition System

- Enabling strongly incremental parsing
  1. State: partial tree + the next word
  2. Actions: determine where and how to integrate the next word into the partial tree

  - Where: the rightmost chain
  - How: ATTACH and JUXTAPOSE

Oracle actions
1. For any parse tree without unary chains, there are oracle actions for constructing it in the attach-juxtapose system
2. The sequence of oracle actions is unique

3. Action Generation with GNNs

- Encoder: Following prior work [5, 6], we use BERT/XLNet + self-attention layers to encode the sentence as a sequence of vectors
- GNNs
  1. Initialize node features in the partial tree using the encoder’s output
  2. Apply GNNs to calculate the features on the rightmost chain
- Action decoder: Generate an action based on features on the rightmost chain

5. Experiments

- Training: Maximizing the likelihood of oracle action sequences

- Competitive Performance on Penn Treebank

<table>
<thead>
<tr>
<th>Model</th>
<th>EM</th>
<th>F1</th>
<th>LP</th>
<th>LR</th>
<th>#Params</th>
</tr>
</thead>
<tbody>
<tr>
<td>Liu and Zhang [25]</td>
<td>37.05</td>
<td>91.71</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Liu and Zhang [25] (BERT)</td>
<td>49.31</td>
<td>93.55</td>
<td>93.80</td>
<td>93.70</td>
<td>26M</td>
</tr>
<tr>
<td>Kkuev and Klein [21]</td>
<td>53.06</td>
<td>93.93</td>
<td>94.45</td>
<td>94.73</td>
<td>34M</td>
</tr>
<tr>
<td>Zhao and Zhou [50] (XLM-R)</td>
<td>53.84</td>
<td>93.94</td>
<td>93.94</td>
<td>93.94</td>
<td>34M</td>
</tr>
<tr>
<td>Zhao and Zhou [50] (XLM)</td>
<td>58.33</td>
<td>96.33</td>
<td>96.31</td>
<td>96.33</td>
<td>37M</td>
</tr>
<tr>
<td>Mehta et al. [28] (XLM)</td>
<td>58.41</td>
<td>96.38</td>
<td>96.36</td>
<td>96.34</td>
<td>40M</td>
</tr>
<tr>
<td>Ours (BERT)</td>
<td>57.29</td>
<td>95.79</td>
<td>0.05</td>
<td>95.55</td>
<td>0.06</td>
</tr>
<tr>
<td>Ours (XLM)</td>
<td>58.27</td>
<td>96.34</td>
<td>0.03</td>
<td>96.35</td>
<td>0.02</td>
</tr>
</tbody>
</table>

- New state of the art on Chinese Treebank

- Parsing speed: Time for parsing the Penn Treebank testing set

<table>
<thead>
<tr>
<th>Model</th>
<th>EM</th>
<th>F1</th>
<th>LP</th>
<th>LR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kkuev et al. [21]</td>
<td>-</td>
<td>91.75</td>
<td>91.96</td>
<td>91.55</td>
</tr>
<tr>
<td>Zhao and Zhou [50]</td>
<td>44.42</td>
<td>92.14</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Mehta et al. [28] (BERT)</td>
<td>-</td>
<td>92.18</td>
<td>92.32</td>
<td>92.03</td>
</tr>
<tr>
<td>Ours (BERT)</td>
<td>49.72</td>
<td>93.59</td>
<td>0.29</td>
<td>93.49</td>
</tr>
<tr>
<td>Ours (XLM)</td>
<td>59.48</td>
<td>95.79</td>
<td>0.29</td>
<td>95.80</td>
</tr>
</tbody>
</table>

- Parser: Using GCN-GNNs