Comparing Non-projective Strategies for Labeled Graph-based Dependency Parsing

Anders Björkelund  Jonas Kuhn
Institut für Maschinelle Sprachverarbeitung, University of Stuttgart
{anders, jonas}@ims.uni-stuttgart.de

Non-projective Parsing

Avoid non-projectivity by applying transformations in pre- and post-processing steps (Nivre and Nilsson, 2005)
- Postprocess the training data by projectivizing all trees (by lifting) and encoding the transformations in the edge labels
- Preprocess the parser output by applying an inverse transformation that recovers non-projective edges

Inversion schemes
- **Head**: Mark the lifted edge with the label of the non-projective head
- **Path**: Mark all edges along the path where the edge was lifted, as well as the lifted edge
- **HeadPath**: A combination of Head and Path

Explosion of label set
- Breadth-first search from pseudo-projective head
- Halts depending on encoding scheme
  - Head = First matching label
  - Path = Deepest marked with ↓
  - HeadPath = Combination of above, though Head is used as fallback

Non-projective Approximation Algorithm

- Non-projective approximation (NPA) (McDonald and Pereira, 2006)
- Greedy approximate search
  - Start from best projective tree from chart-parser
  - Move edges iteratively, one at a time, until score does not increase
  - Potentially exponential, however fast in practice

Pseudo-projective Parsing

- Avoid non-projectivity by applying transformations in pre- and post-processing steps (Nivre and Nilsson, 2005)
- Postprocess the training data by projectivizing all trees (by lifting) and encoding the transformations in the edge labels
- Preprocess the parser output by applying an inverse transformation that recovers non-projective edges

Inversion schemes
- **Head**: Mark the lifted edge with the label of the non-projective head
- **Path**: Mark all edges along the path where the edge was lifted, as well as the lifted edge
- **HeadPath**: A combination of Head and Path

Explosion of label set
- The encoding schemes lead to an increased set of labels
  - Increase depends on encoding scheme (at worst 2n+1 for HeadPath), and many labels are very infrequent
  - Johansson and Nugues (2008) proposed to cap the number of newly introduced labels to the m most frequent

Conclusions at a Glance

- Handling non-projective edges is important, especially for languages with freer word order (Czech and German)
- The non-projective approximation algorithm generally outperforms pseudo-projective parsing
- Capping the number of new labels with pseudo-projective parsing leads to slight performance degradations but considerable speed improvements

Chart-based Dependency Parsing

Data and Evaluation

- Data from CoNLL 2009 shared task
  - Cross-validation on training set
    - Baseline is the projective chart-based algorithm
  - Evaluation metrics
    - Labeled Attachment Score (LAS): Percentage of tokens that received the correct head and correct edge label
    - Exact Match (LEM): Percentage of fully correct sentences
    - Non-projective Recall: Percentage of edges that are non-projective in gold that received correct head and label
    - Non-projective Precision: Percentage of edges that are non-projective in prediction that received the correct head and label

Results

<table>
<thead>
<tr>
<th>Czech</th>
<th>English</th>
<th>German</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAS</td>
<td>LEM</td>
<td>P R</td>
</tr>
<tr>
<td>Baseline</td>
<td>81.10</td>
<td>29.50</td>
</tr>
<tr>
<td>Path</td>
<td>81.75</td>
<td>27.28</td>
</tr>
<tr>
<td>Head</td>
<td>81.86</td>
<td>27.67</td>
</tr>
<tr>
<td>HeadPath</td>
<td>81.74</td>
<td>27.67</td>
</tr>
<tr>
<td>Path</td>
<td>81.76</td>
<td>27.38</td>
</tr>
<tr>
<td>Head</td>
<td>81.94</td>
<td>27.87</td>
</tr>
<tr>
<td>HeadPath</td>
<td>81.94</td>
<td>27.94</td>
</tr>
<tr>
<td>NPA</td>
<td>81.12</td>
<td>28.40</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Value</th>
<th>Value</th>
<th>Value</th>
<th>Value</th>
<th>Value</th>
<th>Value</th>
<th>Value</th>
<th>Value</th>
<th>Value</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>36.05</td>
<td>13.86</td>
<td>0.95</td>
<td>6.746</td>
<td>86.01</td>
<td>30.94</td>
<td>4.74</td>
<td>0</td>
<td>86.50</td>
<td>33.44</td>
<td>7.26</td>
</tr>
<tr>
<td>89.74</td>
<td>29.01</td>
<td>76.42</td>
<td>23.38</td>
<td>8.84</td>
<td>86.74</td>
<td>33.77</td>
<td>62.45</td>
<td>40.12</td>
<td>87.41</td>
<td>33.53</td>
</tr>
<tr>
<td>89.80</td>
<td>29.37</td>
<td>61.47</td>
<td>39.02</td>
<td>1.983</td>
<td>86.64</td>
<td>33.53</td>
<td>64.32</td>
<td>40.24</td>
<td>84.46</td>
<td>33.52</td>
</tr>
<tr>
<td>89.77</td>
<td>29.41</td>
<td>76.85</td>
<td>23.85</td>
<td>8.24</td>
<td>86.81</td>
<td>33.52</td>
<td>69.78</td>
<td>40.53</td>
<td>83.85</td>
<td>33.52</td>
</tr>
<tr>
<td>89.83</td>
<td>29.44</td>
<td>61.33</td>
<td>39.98</td>
<td>2.061</td>
<td>86.79</td>
<td>33.74</td>
<td>60.27</td>
<td>41.65</td>
<td>9.424</td>
<td>33.66</td>
</tr>
<tr>
<td>89.82</td>
<td>29.41</td>
<td>60.55</td>
<td>40.44</td>
<td>2.066</td>
<td>86.75</td>
<td>33.66</td>
<td>60.78</td>
<td>42.14</td>
<td>9.359</td>
<td>33.66</td>
</tr>
<tr>
<td>89.90</td>
<td>29.32</td>
<td>59.16</td>
<td>43.77</td>
<td>1.747</td>
<td>87.05</td>
<td>33.99</td>
<td>60.37</td>
<td>50.47</td>
<td>14.265</td>
<td>33.99</td>
</tr>
</tbody>
</table>

All parsers except English Path are significantly better than the baseline (using a paired t-test)
- For Czech and German, non-projective approximation is significantly better than all the pseudo-projective parsers
- Also considerable improvements in exact match on Czech and German
- For English, only small differences and only NPA vs Path is significant
- Runtime of non-capped pseudo-projective parsers increased roughly 50% compared to NPA