Semantic Dependency Graph Parsing
Using Tree Approximations

Željko Agić♣♥ Alexander Koller♥ Stephan Oepen♣♥

♠ Center for Language Technology, University of Copenhagen
♥ Department of Linguistics, University of Potsdam
♣ Department of Informatics, University of Oslo

IWCS 2015, London, 2015-04-17
Dependency tree parsing is not a contradiction.
Dependency tree parsing

Very high accuracy and fast **dependency parsing** is not a contradiction

B. Bohnet - Proceedings of the 23rd International Conference on ..., 2010 - dl.acm.org

Abstract In addition to high accuracy, short **parsing** and training times are the most important properties of a **parser**. However, **parsing** and training times are still relatively long. To determine why, we analyzed the time usage of a **dependency parser**. We illustrate that ...
Dependency *tree* parsing

- it is also a big success story in NLP
  - robust and efficient
  - high accuracy across domains and languages
  - enables cross-lingual approaches
Dependency tree parsing

- it is also a big success story in NLP
  - robust and efficient
  - high accuracy across domains and languages
  - enables cross-lingual approaches

- and it is simple
The simplicity

He walks and talks.
The simplicity

He walks and talks.
The simplicity

He walks and talks.
He walks and talks.
The simplicity

*With great speed and accuracy, come great constraints.*

- tree constraints
  - single root, single head
  - spanning, connectedness, acyclicity
  - sometimes even projectivity
- there’s been a lot of work beyond that
  - plenty of lexical resources
  - successful semantic role labeling shared tasks
  - algorithms for DAG parsing
- but?
  - it’s apparently *balkanized*, i.e.,
    the representations are not as uniform as in depparsing
Recent efforts

- Banarescu et al. (2013):
  
  *We hope that a sembank of simple, whole-sentence semantic structures will spur new work in statistical natural language understanding and generation, like the Penn Treebank encouraged work on statistical parsing.*

- Oepen et al. (2014):
  
  SemEval semantic dependency parsing (SDP) shared task
  
  - WSJ PTB text
  - three DAG annotation layers: DM, PAS, PCEDT
  - bilexical dependencies between words
  - disconnected nodes allowed
A similar technique is almost impossible to apply to other crops, such as cotton, soybeans and rice.

(b) DELPH-IN Minimal Recursion Semantics–derived bi-lexical dependencies (DM).

A similar technique is almost impossible to apply to other crops, such as cotton, soybeans and rice.

(c) Enju Predicate–Argument Structures (PAS).

A similar technique is almost impossible to apply to other crops, such as cotton, soybeans and rice.

(d) Parts of the textogrammatical layer of the Prague Czech-English Dependency Treebank (PCEDT).
SDP 2014 shared task

- uniform, but not the same
- PCEDT seems to be somewhat more distinct
- key ingredients of non-trees
  - singletons
  - reentrancies: indegree > 1
Reentrancies

node indegree

indegree of sources in reentrancy

DM  PAS  PCEDT

low-hanging fruit?
Reentrancies

Node indegree

Indegree of sources in reentrancy
- DM
- PAS
- PCEDT

Low-hanging fruit?
Hey, these DAGs are very tree-like. Let's convert them to trees and use standard depparsers!
Parsing with tree approximations

**GOLD STANDARD**

- will
- that
- be
- winner
- is
- highly
- uncertain
- who

**LOCAL**

- will
- be
- winner
- that
- is
- uncertain
- highly
- who

**DFS**

- be
- is
- winner
- that
- highly
- uncertain

- gold standard edge
- flipped
- not preserved
Parsing with tree approximations

- flip the flippable, baseline-delete the rest
- train on trees, parse for trees, flip back in post-processing

GOLD STANDARD

LOCAL

DFS

- gold standard edge
- flipped
- not preserved
Parsing with tree approximations

- flip the flippable, baseline-delete the rest
- train on trees, parse for trees, flip back in post-processing

- works OK...ish
  - average labeled $F_1$ in the high 70s
  - task winner votes between tree approximations
Where do all the lost edges go?

▶ the deleted edges cannot be recovered
▶ upper bound recall
  ▶ graph-tree-graph conversion with no parsing in-between
  ▶ measure the lossiness

<table>
<thead>
<tr>
<th></th>
<th>DM</th>
<th>PCEDT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td>OFFICIAL</td>
<td>100.00</td>
<td>55.28</td>
</tr>
<tr>
<td>LOCAL</td>
<td>100.00</td>
<td>87.50</td>
</tr>
<tr>
<td>DFS</td>
<td>100.00</td>
<td>97.30</td>
</tr>
</tbody>
</table>

▶ new agenda
  ▶ inspect the lost edges
  ▶ build a better tree approximation on top
Where do all the lost edges go?
Where do all the lost edges go?

- there are *undirected cycles* in the graphs
  - interesting structural properties?
  - discriminate specific phenomena they encode?
Undirected cycles

- we mostly ignore PAS from now on
- DM: 3-word cycles dominate *(triangles)*
- PCEDT: 4-word cycles *(squares)*
- sentences with more than one cycle not very frequent
Undirected cycles

<table>
<thead>
<tr>
<th></th>
<th>DM</th>
<th></th>
<th></th>
<th>PAS</th>
<th></th>
<th></th>
<th>PCEDT</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#</td>
<td>%</td>
<td></td>
<td>#</td>
<td>%</td>
<td></td>
<td>#</td>
<td>%</td>
</tr>
<tr>
<td>N V V</td>
<td>3843</td>
<td>29.63</td>
<td></td>
<td>N V V</td>
<td>15541</td>
<td>34.44</td>
<td>CC N N V</td>
<td>4789</td>
</tr>
<tr>
<td>PRP V V</td>
<td>1208</td>
<td>9.31</td>
<td></td>
<td>MD N V</td>
<td>5005</td>
<td>11.09</td>
<td>CC N N N</td>
<td>3418</td>
</tr>
<tr>
<td>N TO V V</td>
<td>1203</td>
<td>9.28</td>
<td></td>
<td>PRP V V</td>
<td>4012</td>
<td>8.89</td>
<td>N N V</td>
<td>2512</td>
</tr>
<tr>
<td>J N V</td>
<td>1059</td>
<td>8.16</td>
<td></td>
<td>J N V</td>
<td>3544</td>
<td>7.85</td>
<td>CC V V V</td>
<td>1633</td>
</tr>
<tr>
<td>IN N V</td>
<td>962</td>
<td>7.42</td>
<td></td>
<td>CC N V V</td>
<td>2155</td>
<td>4.78</td>
<td>CC N V V</td>
<td>1614</td>
</tr>
<tr>
<td>J J N</td>
<td>506</td>
<td>3.90</td>
<td></td>
<td>MD PRP V</td>
<td>1622</td>
<td>3.59</td>
<td>N N N V</td>
<td>805</td>
</tr>
<tr>
<td>CD CD N</td>
<td>324</td>
<td>2.50</td>
<td></td>
<td>IN N V</td>
<td>1087</td>
<td>2.41</td>
<td>N N V V</td>
<td>752</td>
</tr>
<tr>
<td>PRP TO V V</td>
<td>277</td>
<td>2.14</td>
<td></td>
<td>J PRP V</td>
<td>877</td>
<td>1.94</td>
<td>N V V V</td>
<td>665</td>
</tr>
<tr>
<td>J PRP V</td>
<td>228</td>
<td>1.76</td>
<td></td>
<td>CC N N N</td>
<td>676</td>
<td>1.50</td>
<td>, N N N</td>
<td>495</td>
</tr>
<tr>
<td>N N V</td>
<td>202</td>
<td>1.56</td>
<td></td>
<td>CC V V</td>
<td>561</td>
<td>1.24</td>
<td>CC J J N</td>
<td>447</td>
</tr>
</tbody>
</table>

- DM, PAS: mostly control and coordination
- PCEDT: almost exclusively coordination
- supported also by the edge label tuples, and the lemmas
Back to tree approximations

- edge operations up to now
  - *flipping* – comes with implicit *overloading*
  - *deletion* – edges are permanently lost
Back to tree approximations

- edge operations up to now
  - *flipping* – comes with implicit *overloading*
  - *deletion* – edges are permanently lost

- new proposal
  - detect an undirected cycle
  - select and disconnect an appropriate edge
    - *radical*: overload an appropriate label for reconstruction, or
    - *conservative*: trim only a subset of edges using lemma-POS cues
  - in post-processing, reconnect the edge
    - by reading the reconstruction off of the overloaded label, or
    - by detecting the lemma-POS trigger

- we call these operations *trimming* and *untrimming*
Trimming and untrimming
## Upper bounds

<table>
<thead>
<tr>
<th></th>
<th>DM</th>
<th>PCEDT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$P$</td>
<td>$R$</td>
</tr>
<tr>
<td><strong>OFFICIAL</strong></td>
<td>100.00</td>
<td>55.28</td>
</tr>
<tr>
<td><strong>LOCAL</strong></td>
<td>100.00</td>
<td>87.50</td>
</tr>
<tr>
<td><strong>DFS</strong></td>
<td>100.00</td>
<td>97.30</td>
</tr>
<tr>
<td><strong>radical trimming</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\triangledown$ + LOCAL</td>
<td>100.00</td>
<td>88.33</td>
</tr>
<tr>
<td>$\triangledown$ + DFS</td>
<td>100.00</td>
<td>98.89</td>
</tr>
<tr>
<td>$\square$ + LOCAL</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>$\square$ + DFS</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td><strong>conservative trimming</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\triangledown$ + LOCAL</td>
<td>98.98</td>
<td>87.93</td>
</tr>
<tr>
<td>$\triangledown$ + DFS</td>
<td>99.12</td>
<td>98.07</td>
</tr>
<tr>
<td>$\square$ + LOCAL</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>$\square$ + DFS</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td><strong>radical – DFS</strong></td>
<td>0.00</td>
<td>+1.59</td>
</tr>
<tr>
<td><strong>conservative – DFS</strong></td>
<td>-0.88</td>
<td>+0.77</td>
</tr>
</tbody>
</table>
Parsing

- preprocessing: trimming + DFS + baseline = training trees
- training and parsing
  - mate-tools graph-based depparser
  - CRF++ for top node detection
  - SDP companion data and Brown clusters as additional features
- postprocessing: removing baseline artifacts + reflipping +
  + untrimming = output graphs
Results

<table>
<thead>
<tr>
<th></th>
<th>closed track</th>
<th>open track</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DM</td>
<td>PCEDT</td>
</tr>
<tr>
<td></td>
<td>LF</td>
<td>LM</td>
</tr>
<tr>
<td><strong>DFS</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>radical</strong></td>
<td>79.35</td>
<td>9.05</td>
</tr>
<tr>
<td>(\nabla + \text{DFS})</td>
<td>77.73</td>
<td>12.15</td>
</tr>
<tr>
<td>(\square + \text{DFS})</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td><strong>conservative</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\nabla + \text{DFS})</td>
<td>80.05</td>
<td>18.91</td>
</tr>
<tr>
<td>(\square + \text{DFS})</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td><strong>radical – DFS</strong></td>
<td>-1.62</td>
<td>3.10</td>
</tr>
<tr>
<td><strong>conservative – DFS</strong></td>
<td>0.70</td>
<td>9.86</td>
</tr>
</tbody>
</table>

- lower upper bounds, higher parsing scores
- nice increase in \(LM\)
- best overall score for any tree approximation-based system
Conclusions

➤ our contributions
  ➤ put SDP DAGs under the lens
  ➤ uncovered the link between non-trees and control, coordination
  ➤ used this to implement a state-of-the-art system based on tree approximations

➤ future work
  ➤ did some more experiments
    ➤ answer set programming for better tree approximations
    ➤ did not see improvements

➤ go for real graph parsing
Thank you for your attention. 😊