Neural Network Techniques in Dependency Parsing

Joakim Nivre

Uppsala University
Linguistics and Philology
Overall Plan

1. Basic notions of dependency grammar and dependency parsing
2. Graph-based and transition-based dependency parsing
3. Advanced graph-based parsing techniques
4. Advanced transition-based parsing techniques
5. Neural network techniques in dependency parsing
6. Multilingual parsing from raw text to universal dependencies
**Taking Stock**

**Graph-based Parsers**
- Global Inference
- Global Learning
- Local Feature Scope

**Transition-based Parsers**
- Local Inference
- Local Learning
- Global Feature Scope

2007

LAS: 83.8 v. 83.6
[McDonald & Nivre 2007]

- higher-order chart parsing
- pruning
- ILP
- dual decomp
- mildly non-projective
- etc.

2014

LAS: 85.8 v. 85.5
[Zhang et al. 2013]

- beam search
- perceptron
- dynamic oracles
- dynamic programming
- more features
- etc.

**Evaluated on overlapping 9 languages in studies**
Neural Network Techniques

- Empirical results have improved substantially since 2014
- Neural networks techniques yield more effective features:
  - Features are learned (not hand-crafted)
  - Features are continuous and dense (not discrete and sparse)
  - Features can be tuned to (multiple) specific tasks
  - Features can capture unbounded dependencies
- Parsing architectures remain essentially the same
Learning Features [Titov and Henderson 2007]

- Incremental Sigmoid Belief Network (ISBN)
- Learns complex features using binary latent variables
- Captures dependencies at arbitrarily long distances
- First generative model for transition-based parsing
Learning Dense Features [Chen and Manning 2014]

- MaltParser with MLP instead of SVM (greedy, local)
- But 2 percentage points better LAS on PTB/CTB!?
Traditional Features [Chen and Manning 2014]

- Sparse – but lexical features and interaction features crucial
- Incomplete – unavoidable with hand-crafted feature templates
- Expensive – accounts for 95% of computing time
Dense Features [Chen and Manning 2014]

- Sparse – dense features capture similarities (words, pos, dep)
- Incomplete – neural network learns interaction features
- Expensive – matrix multiplication with low dimensionality

Dense dim = 200

0.1 0.9 -0.2 0.3 ... -0.1 -0.5
PoS Embeddings [Chen and Manning 2014]
Dep Embeddings [Chen and Manning 2014]
The Power of Embeddings

**One-Hot** (discrete, sparse)

- 1-of-N encoding

**Embedding** (continuous, dense)

- Inherently much more expressive ($\mathcal{R} \times D$ vs. 1)
- Can capture similarities between items (sparsity)
- Can be pre-trained on large unlabeled corpora (OOV)
- Can be learned/tuned specifically for the parsing task
Neural Dependency Parsing

- Dominated by transition-based approaches
- Two main lines of work:
  - More powerful (recurrent) neural networks
    [Dyer et al. 2015, Kiperwasser and Goldberg 2016]
  - Global optimization and beam search
    [Weiss et al. 2015, Andor et al. 2016]
- Additional themes:
  - Character-based models for morphologically rich languages
    [Ballesteros et al. 2015]
  - Cross-lingual embeddings and typological features
    [Ammar et al. 2016]
Stack-LSTM [Dyer et al. 2015]

- LSTM encoding of parser configurations (S, B, A)
- Stack elements recursively composed of word representations
Bi-LSTM [Kiperwasser and Goldberg 2016]

- Bi-LSTM encodes global context in word representations
- Exploration with dynamic oracles prevent error propagation
Global Normalization [Andor et al. 2016]

Local

\[ p_L(d_{1:n}) = \prod_{j=1}^{n} p(d_j|d_{1:j-1}; \theta) \]

\[ p(d_j|d_{1:j-1}; \theta) = \frac{\exp \rho(d_{1:j-1}, d_j; \theta)}{Z_L(d_{1:j-1}; \theta)} \]

\[ Z_L(d_{1:j-1}; \theta) = \sum_{d' \in \mathcal{A}(d_{1:j-1})} \exp \rho(d_{1:j-1}, d'; \theta) \]

Global

\[ p_G(d_{1:n}) = \frac{\exp \sum_{j=1}^{n} \rho(d_{1:j-1}, d_j; \theta)}{Z_G(\theta)} \]

\[ Z_G(\theta) = \sum_{d'_{1:n} \in \mathcal{D}_n} \exp \sum_{j=1}^{n} \rho(d'_{1:j-1}, d'_j; \theta) \]

- Global normalization \(\rightarrow\) sum over all transition sequences
- Approximation using beam search and early update
## Evaluation

<table>
<thead>
<tr>
<th>System</th>
<th>UAS</th>
<th>LAS</th>
<th>Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhang and Nivre (2011)</td>
<td>93.5</td>
<td>91.9</td>
<td>Transition, struct perc, beam</td>
</tr>
<tr>
<td>Martins et al. (2013)</td>
<td>92.9</td>
<td>90.6</td>
<td>Graph, 3rd-order, dual decomp</td>
</tr>
<tr>
<td>Zhang and McDonald (2014)</td>
<td>92.9</td>
<td>90.6</td>
<td>Graph, 3rd-order, cube pruning</td>
</tr>
<tr>
<td>Dyer et al. (2015)</td>
<td>93.1</td>
<td>90.9</td>
<td>Transition, LSTM, greedy</td>
</tr>
<tr>
<td>Kiperwasser et al. (2016)</td>
<td>93.9</td>
<td>91.9</td>
<td>Transition, LSTM/MLP, greedy</td>
</tr>
<tr>
<td>Weiss et al. (2015)</td>
<td>94.0</td>
<td>92.0</td>
<td>Transition, MLP, beam</td>
</tr>
<tr>
<td>Andor et al. (2016)</td>
<td>94.6</td>
<td>92.8</td>
<td>Transition, MLP global, beam</td>
</tr>
</tbody>
</table>

Evaluation on WSJ with Stanford Dependencies
Taking Stock Again

- Traditional architectures persist
  - When will we see a new dependency parsing algorithm?
  - Do we even need parsing algorithms?
- Transition-based parsers dominate
  - Rich features trump global learning/inference?
  - Or will the empire strike back?
- Predicting the future is hard . . .
Coming Up Next

1. Basic notions of dependency grammar and dependency parsing
2. Graph-based and transition-based dependency parsing
3. Advanced graph-based parsing techniques
4. Advanced transition-based parsing techniques
5. Neural network techniques in dependency parsing
6. Multilingual parsing from raw text to universal dependencies
References and Further Reading


