Neural Constituency Parsing

Dan Klein
CS 288

Syntactic Parsing

She enjoys playing tennis.

Syntactic Parsing

Historical Trends

[Slide from Slav Petrov]
Output Correlations

She enjoys playing tennis.

Grammars

S \rightarrow \text{NP} \text{ VP}

Input-Output Correlations

Span-Based Parsing

NP^S \rightarrow she

NP  \rightarrow \text{she}

VP  \rightarrow \text{enjoys}

NP  \rightarrow \text{playing}

She  \enjoys  \text{ playing}  \text{ tennis}.
Parsing as Span Classification

Routing with LSTMs

VP

She enjoys playing tennis.

Routing with LSTMs

Pronoun to the left

She enjoys playing tennis.

Routing with LSTMs

Verb at the start

She enjoys playing tennis.
Routing with LSTMs

She enjoys playing tennis.

Span Classification

She enjoys playing tennis.

She enjoys playing tennis.

She enjoys playing tennis.
span classification

She enjoys playing tennis.

span classification

She enjoys playing tennis.

span classification

She enjoys playing tennis.

non-constituents

∅

She enjoys playing tennis.
… But Will We Get a Tree Out?

Reconciliation

Does It Work?

What’s Going on in There?
**What’s Going on in There?**

### Why don’t we need a grammar?

- Adjacent tree labels are redundant with LSTM features
- If we can predict surrounding tree labels from our LSTM representation of the input, then this information doesn't need to be provided explicitly by grammar production rules
- We find that for 92.3% of spans, the label of the span’s parent can predicted from the neural representation of the span

---

### Do we need tree constraints?

- Not for F1
- Many neural parsers no longer model output correlations with grammar rules, but still use output correlations from tree constraints
- Predicting span brackets independently gives nearly identical performance on PTB development set F1 and produces valid trees for 94.5% of sentences

---

### Is distant context important?

- Yes!
- Almost a full point of F1 is lost by truncating context 5 words away from span endpoints and half a point with 10 words

---

### What word representations do we need?

<table>
<thead>
<tr>
<th>Representation</th>
<th>F1 Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word Only</td>
<td>91.44</td>
</tr>
<tr>
<td>Word and Tag</td>
<td>92.09</td>
</tr>
<tr>
<td>Character LSTM Only</td>
<td>92.24</td>
</tr>
<tr>
<td>Character LSTM and Word</td>
<td>92.22</td>
</tr>
<tr>
<td>Character LSTM, Word, and Tag</td>
<td>92.24</td>
</tr>
</tbody>
</table>
What’s Going on in There?

What about lexicon features?
The character LSTM captures the same information.
Heavily engineered lexicons used to be critical to good performance, but neural models typically don’t use them.
Word features from the Berkeley Parser (Petrov and Klein 2007) can be predicted with over 99.7% accuracy from the character LSTM representation.

Do LSTMs introduce useful inductive bias compared to feedforward networks?
Yes!
We compare a truncated LSTM with feedforward architectures that are given the same inputs.
The LSTM outperformed the best feedforward by 6.5 F1.

Routing with Transformers

Query: verb
She enjoys playing tennis.

Query: verb, verb (VBD), verb (VBG), noun, punctuation
She enjoys playing tennis.
Routing with Transformers

She enjoys playing tennis.

Query: verb

The verb is: enjoys

Routing with Transformers

She enjoys playing tennis.

Query: verb

The verb is: enjoys

Routing with Transformers

She enjoys playing tennis.

Query: word=She

verb=enjoys
**What Helps?**

<table>
<thead>
<tr>
<th>Method</th>
<th>F1 (English, dev)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM</td>
<td>92</td>
</tr>
<tr>
<td>Self-Attentive</td>
<td>93</td>
</tr>
<tr>
<td>+Factored</td>
<td>94</td>
</tr>
</tbody>
</table>

**Results: Multilingual**

<table>
<thead>
<tr>
<th>Language</th>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arabic</td>
<td>Ours</td>
<td>85.5</td>
</tr>
<tr>
<td>German</td>
<td>Coavoux and Craíbe (2017)</td>
<td>84.0</td>
</tr>
<tr>
<td>Hebrew</td>
<td>Cross and Huang (2016)</td>
<td>84.5</td>
</tr>
<tr>
<td>Korean</td>
<td>Björkund et al. (2014)</td>
<td>83.25</td>
</tr>
<tr>
<td>Polish</td>
<td>Ours</td>
<td>84.75</td>
</tr>
<tr>
<td>Swedish</td>
<td>Ours</td>
<td>85.5</td>
</tr>
<tr>
<td>Hungarian</td>
<td>Ours</td>
<td>86.25</td>
</tr>
</tbody>
</table>

**Data Hunger**

Problem: Input has more variation than output

Need to handle:
- Rare words not seen during training
- Word forms in morphologically rich languages

**Historical Trends**

[Graph showing historical trends]
Knowledge Modularity

- Knowledge modularity: Learn domain-general knowledge from one data source and use it to solve specific problems elsewhere.

Context Embeddings and Pretraining

**Key Idea**: Embed contexts, not words. Use these embeddings for other tasks.

Example: BERT (Devlin et al., 2019)—bidirectional Transformer trained on masked language modeling and next-sentence prediction.

Recent Explosion of Pretraining Work

GLUE ScTI (ICLR 2020)

BERT

GLUE Baseline (ICLR 2019)

Parsing as Span Classification
Pretraining

She enjoys playing tennis.

Architecture

Encoder Architectures

<table>
<thead>
<tr>
<th>Encoder Architectures</th>
<th>F1 Score (English)</th>
<th>Number of Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>No pre-training</td>
<td>92.08 F1</td>
<td>ELMo 26M</td>
</tr>
<tr>
<td>Pre-training</td>
<td>95.13 F1 (with ELMo)</td>
<td>BERT-base 107M</td>
</tr>
<tr>
<td></td>
<td>95.60 F1 (with BERT)</td>
<td>XLNet-large 343M</td>
</tr>
</tbody>
</table>
Results: Multilingual

<table>
<thead>
<tr>
<th>Language</th>
<th>Arabic</th>
<th>French</th>
<th>German</th>
<th>Hebrew</th>
</tr>
</thead>
<tbody>
<tr>
<td>Björkellund et al. (2014)</td>
<td>82.5</td>
<td>82.5</td>
<td>83.3</td>
<td>84.1</td>
</tr>
<tr>
<td>Kräu and Klein (2018)</td>
<td>87</td>
<td>87</td>
<td>89.5</td>
<td>88.5</td>
</tr>
<tr>
<td>Cross and Crabbé (2017)</td>
<td>82.9</td>
<td>84.0</td>
<td>84.5</td>
<td>89</td>
</tr>
<tr>
<td>This work (one model per language)</td>
<td>85.5</td>
<td>84.0</td>
<td>84.5</td>
<td>89</td>
</tr>
<tr>
<td>Cross and Huang (2016)</td>
<td>87</td>
<td>87</td>
<td>89.5</td>
<td>88.5</td>
</tr>
<tr>
<td>This work (joint multilingual model)</td>
<td>85.5</td>
<td>84.0</td>
<td>84.5</td>
<td>89</td>
</tr>
</tbody>
</table>

Does Structure Help?

- Out of Domain Parsing

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Berkeley</th>
<th>BLLIP</th>
<th>In-Order</th>
<th>Chart</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSJ Test</td>
<td>90.06</td>
<td>+0.0%</td>
<td>91.48</td>
<td>+0.0%</td>
</tr>
<tr>
<td>Brown All</td>
<td>84.64</td>
<td>+54.3%</td>
<td>85.89</td>
<td>+65.6%</td>
</tr>
<tr>
<td>EWT All</td>
<td>77.38</td>
<td>+127.6%</td>
<td>79.91</td>
<td>+135.8%</td>
</tr>
</tbody>
</table>

Neural parsers improve out-of-domain numbers, but not more than in-domain numbers

- Other Neural Constituency Parsers

- Back to at least Henderson 1998!
- Recent directions:
  - Shift-Reduce, eg Cross and Huang 2016
  - SR/Generative, eg Dyer et al 2016 (RNNG)
  - In-Order Generative, eg Liu and Zhang 2017
Thank You!

Berkeley

nlp.cs.berkeley.edu