Neural CRF Parsing

AUTHORS: GREG DURRETT AND DAN KLEIN
PRESENTER: YUNDI FEI
Overview

- Based on the baseline CRF model by Hall et al. (2014)
- What is CRF (Conditional Random Field)?
  - A class of statistical modeling method in the sequence modeling family
  - Often used for labeling or parsing of sequential data, such as natural language processing
Overview

- What is CRF (Conditional Random Field)?
  - Defines posterior probability of a label sequence given an input observation sequence
  - Conditional probability is $P(\text{label sequence } Y \mid \text{observation sequence } X)$ instead of $P(Y \mid X)$
  - The probability of a transition between the labels may depend on past and future observations
Overview

- This work: a CRF constituency parser that individual anchored rule productions are scored based on nonlinear features computed with a feedforward neural network in addition to linear functions of sparse indicator features like a standard CRF
Prior work

- Compared to conventional CRFs
  - Scores can be thought of as **nonlinear potentials analogous** to linear potentials in conventional CRFs
  - Computations factor along the same substructures as in standard CRFs
Prior work

- Compared to prior neural network models
  - Prior: sidestepped the problem of structured inference by making sequential decisions or by doing reranking
  - This framework: permits exact inference via CKY, since the model’s structured interactions are purely discrete and do not involve continuous hidden state
Model

- Decomposes over anchored rules, and it scores each to these with a potential function
  - Nonlinear functions of word embedding
  - Linear functions of sparse indicator features like a standard CRF
Model – Anchored Rule

- An anchored rule: a tuple \((r, s)\)
  - \(r\) = an indicator of the rule’s identity
  - \(s = (i, j, k)\) indicator of the span \((i, k)\) and split point \(j\) of the rule
  - For unary rules, specify a null value for the split point
Model – Anchored Rule

- A tree $T$ = a collection of anchored rules subject to the constraint that those rules form a tree

- All of the parsing models are CRFs that decompose over anchored rule productions and place a probability distribution over trees conditioned on a sentence $w$ as

$$P(T|w) \propto \exp \left( \sum_{(r,s) \in T} \phi(w, r, s) \right)$$
Model – Scoring Anchored Rule

- $\Phi$ is a scoring function that considers the input sentence and the anchored rule
- Can be a neural net, a linear function of surface features, or combination of the two
Model – Scoring Anchored Rule

- Baseline sparse scoring function
  - \( f_o(r) \in \{0,1\}^{n_o} \) a sparse vector of features expressing properties of \( r \) (such as the rule’s identity or its parent label)
  - \( f_s(w, s) \in \{0,1\}^{n_s} \) a sparse vector of surface features associated with the words in the sentence and the anchoring
  - \( W \) a \( n_s \times n_o \) matrix of weights

\[
\phi_{\text{sparse}}(w, r, s; W) = f_s(w, s)^T W f_o(r)
\]
\[ W_{ij} = \text{ght}([f_s, i \quad f_o, j]) \]
Model – Scoring Anchored Rule

- Neural scoring function
  - $f_w(w, s) \in \mathbb{N}^{n_w}$ a function that produces a fixed-length sequence of word indicators based on the input sentence and the anchoring
  - $v : \mathbb{N} \to \mathbb{R}^{n_e}$ embedding function
    - the dense representations of the words are subsequently concatenated to form a vector we denote by $v(f_w)$
  - $H \in \mathbb{R}^{n_h \times (n_wn_e)}$ real valued parameters
  - $g$ elementwise nonlinearity: rectified linear units $g(x) = \max(x, 0)$

$$h(w, s; H) = g(Hv(f_w(w, s)))$$

$$\phi_{\text{neural}}(w, r, s; H, W) = h(w, s; H)^{\top} W f_o(r)$$
Model – Scoring Anchored Rule

- Two models combined:

\[ \phi(w, r, s; W_1, H, W_2) = \phi_{\text{sparse}}(w, r, s; W_1) + \phi_{\text{neural}}(w, r, s; H, W_2) \]
Features

- **Sparse**: $f_s$
  - At *pretermimal* layer
    - Prefixes and suffixes up to length 5 of the current word and neighboring words as well as the words’ identities
  - For *nonterminal* productions, fire indicators on
    - The words before and after the start, end, and split point of the anchored rule
  - Span properties: span length + span shape
    - Span shape: an indicator of where capitalized words, numbers, and punctuation occur in the span
Features

- **Neural**: $f_w$
  - Words surrounding the beginning and end of a span and the split point
  - Look two words in either direction around each point of interest

- **Neural**: $v$
  - Use pre-trained word vectors from Bansal et al. (2014)
  - Contrary to standard practice, do not update these vectors during training
Learning

- To learn weights for neural model, maximize the conditional log likelihood of $D$ training trees $T^*$

$$
\mathcal{L}(H, W) = \sum_{i=1}^{D} \log P(T_i^* | w_i; H, W)
$$
Learning

- The gradient of $W$ takes the standard form of log-linear models:

$$\frac{\partial \mathcal{L}}{\partial W} = \left( \sum_{(r,s) \in \mathcal{T}^*} h(w, s; H) f_o(r)^\top \right) - \left( \sum_T P(T \mid w; H, W) \sum_{(r,s) \in \mathcal{T}} h(w, s; H) f_o(r)^\top \right)$$
Learning

- **To update** $H$, use standard backpropagation
  - First, compute
    \[
    \frac{\partial L}{\partial h} = \left( \sum_{(r,s) \in T^*} W f_o(r) \right) - \left( \sum_{T} P(T|w; H, W) \sum_{(r,s) \in T} W f_o(r) \right)
    \]
  - Because $h$ is the output of the neural network, then apply the chain rule to compute gradients for $H$ and any other parameters in the neural network
Learning

- **Momentum term** $\rho = 0.95$ as suggested by Zeiler (2012)
- **Use minibatch size of 200 trees**
  - For each treebank, train for either 10 passes through the treebank or 1000 minibatches, whichever is shorter
- **Initialize the output weight matrix** $W$ to zero
- **Initialize the lower level neural network parameters** $H$ with each entry being independently sampled from a Gaussian with mean 0 and variance 0.01
  - Better than uniform initialization but variance not important
Improvements

- Follow Hall et al. (2014) and prune according to an X-bar grammar with head-outward binarization
- Ruling out any constituent whose max marginal probability is less than $e^{-9}$
- Reduce the number of spans and split points
Improvements

- Note that the same word will appear in the same position in a large number of span/split point combinations, and cache the contributions to the hidden layer caused by that word (Chen and Manning, 2014)
# Test results

- **Section 23 of the English Penn Treebank (PTB)**

<table>
<thead>
<tr>
<th></th>
<th>F₁ all</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single model, PTB only</td>
<td></td>
</tr>
<tr>
<td>Hall et al. (2014)</td>
<td>89.2</td>
</tr>
<tr>
<td>Berkeley</td>
<td>90.1</td>
</tr>
<tr>
<td>Carreras et al. (2008)</td>
<td>91.1</td>
</tr>
<tr>
<td>Shindo et al. (2012)</td>
<td>91.1</td>
</tr>
<tr>
<td>Single model, PTB + vectors/clusters</td>
<td></td>
</tr>
<tr>
<td>Zhu et al. (2013)</td>
<td>91.3</td>
</tr>
<tr>
<td>This work*</td>
<td>91.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>F₁ all</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extended conditions</td>
<td></td>
</tr>
<tr>
<td>Charniak and Johnson (2005)</td>
<td>91.5</td>
</tr>
<tr>
<td>Socher et al. (2013)</td>
<td>90.4</td>
</tr>
<tr>
<td>Vinyals et al. (2014)</td>
<td>90.5</td>
</tr>
<tr>
<td>Vinyals et al. (2014) ensemble</td>
<td>91.6</td>
</tr>
<tr>
<td>Shindo et al. (2012)</td>
<td>92.4</td>
</tr>
</tbody>
</table>
**Test results**

- on the **nine languages** used in the SPMRL 2013 and 2014 shared tasks

<table>
<thead>
<tr>
<th></th>
<th>Arabic</th>
<th>Basque</th>
<th>French</th>
<th>German</th>
<th>Hebrew</th>
<th>Hungarian</th>
<th>Korean</th>
<th>Polish</th>
<th>Swedish</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dev, all lengths</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hall et al. (2014)</td>
<td>78.89</td>
<td>83.74</td>
<td>79.40</td>
<td>83.28</td>
<td>88.06</td>
<td>87.44</td>
<td>81.85</td>
<td>91.10</td>
<td>75.95</td>
<td>83.30</td>
</tr>
<tr>
<td>This work*</td>
<td><strong>80.68</strong></td>
<td><strong>84.37</strong></td>
<td><strong>80.65</strong></td>
<td><strong>85.25</strong></td>
<td><strong>89.37</strong></td>
<td><strong>89.46</strong></td>
<td><strong>82.35</strong></td>
<td><strong>92.10</strong></td>
<td><strong>77.93</strong></td>
<td><strong>84.68</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Test, all lengths</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Berkeley</td>
<td>79.19</td>
<td>70.50</td>
<td>80.38</td>
<td>78.30</td>
<td>86.96</td>
<td>81.62</td>
<td>71.42</td>
<td>79.23</td>
<td>79.18</td>
<td>78.53</td>
</tr>
<tr>
<td>Berkeley-Tags</td>
<td>78.66</td>
<td>74.74</td>
<td>79.76</td>
<td>78.28</td>
<td>85.42</td>
<td>85.22</td>
<td>78.56</td>
<td>86.75</td>
<td>80.64</td>
<td>80.89</td>
</tr>
<tr>
<td>Crabbé and Seddah (2014)</td>
<td>77.66</td>
<td>85.35</td>
<td>79.68</td>
<td>77.15</td>
<td>86.19</td>
<td>87.51</td>
<td>79.35</td>
<td>91.60</td>
<td>82.72</td>
<td>83.02</td>
</tr>
<tr>
<td>Hall et al. (2014)</td>
<td>78.75</td>
<td>83.39</td>
<td>79.70</td>
<td>78.43</td>
<td>87.18</td>
<td>88.25</td>
<td>80.18</td>
<td>90.66</td>
<td>82.00</td>
<td>83.17</td>
</tr>
<tr>
<td>This work*</td>
<td><strong>80.24</strong></td>
<td><strong>85.41</strong></td>
<td><strong>81.25</strong></td>
<td><strong>80.95</strong></td>
<td><strong>88.61</strong></td>
<td><strong>90.66</strong></td>
<td><strong>82.23</strong></td>
<td><strong>92.97</strong></td>
<td><strong>83.45</strong></td>
<td><strong>85.08</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Reranked ensemble</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2014 Best</td>
<td><strong>81.32</strong></td>
<td><strong>88.24</strong></td>
<td><strong>82.53</strong></td>
<td><strong>81.66</strong></td>
<td><strong>89.80</strong></td>
<td><strong>91.72</strong></td>
<td><strong>83.81</strong></td>
<td><strong>90.50</strong></td>
<td><strong>85.50</strong></td>
<td><strong>86.12</strong></td>
</tr>
</tbody>
</table>
Conclusion

- Decomposes over anchored rules, and it scores each with a potential function.
- Add scoring based on nonlinear features computed with a feedforward neural network to the baseline CRF model.
- Improvements for both English and 9 other languages.
Thank you!