Adaptive Supertagging [Clark & Curran, 2007]

Start with an initial prob. cutoff $\beta$

He reads the book

$\begin{array}{llll}
\text{He} & \text{reads} & \text{the} & \text{book} \\
\text{NP} & (S[pss]\backslash NP)/NP & NP/N & N
\end{array}$
Adaptive Supertagging [Clark & Curran, 2007]

Prune a category, if its probability is below $\beta$ times the prob. of the best category

He reads the book

\[
\begin{array}{cccc}
\text{NP} & \text{reads} & \text{the} & \text{book} \\
(S[pss] \backslash NP)/NP & NP/N & N
\end{array}
\]
Decrease $\beta$ if no spanning analysis
Adaptive Supertagging [Clark & Curran, 2007]

Decrease $\beta$ if no spanning analysis

<table>
<thead>
<tr>
<th>He</th>
<th>reads</th>
<th>the</th>
<th>book</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP</td>
<td>$(S[pss]\backslash NP)/NP$</td>
<td>NP/N</td>
<td>N</td>
</tr>
<tr>
<td>N</td>
<td>$(S\backslash NP)/NP$</td>
<td>NP/NP</td>
<td>$(S\backslash NP)/NP$</td>
</tr>
<tr>
<td>N/N</td>
<td>$S/\backslash NP$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP/NP</td>
<td>$(S[pt]\backslash NP)/NP$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$(S[dcl]\backslash NP)/NP$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Recurrent neural networks (RNN)
Recurrent neural networks

• Use the same computational function and parameters across different time steps of the sequence

• Each time step: takes the input entry and the previous hidden state to compute the output entry

• Loss: typically computed every time step
Recurrent neural networks

Figure from *Deep Learning*, by Goodfellow, Bengio and Courville
Math formula:

\[
\begin{align*}
    a^{(t)} &= b + W s^{(t-1)} + U x^{(t)} \\
    s^{(t)} &= \tanh(a^{(t)}) \\
    o^{(t)} &= c + V s^{(t)} \\
    \hat{y}^{(t)} &= \text{softmax}(o^{(t)})
\end{align*}
\]
Advantage

• Hidden state: a lossy summary of the past
• Shared functions and parameters: greatly reduce the capacity and good for generalization in learning
• Explicitly use the prior knowledge that the sequential data can be processed by in the same way at different time step (e.g., NLP)
Advantage

• Hidden state: a lossy summary of the past
• Shared functions and parameters: greatly reduce the capacity and good for generalization in learning
• Explicitly use the prior knowledge that the sequential data can be processed by in the same way at different time step (e.g., NLP)

• Yet still powerful (actually universal): any function computable by a Turing machine can be computed by such a recurrent network of a finite size (see, e.g., Siegelmann and Sontag (1995))
Figure 4: A simple recurrent network.
Supertagging with a RNN

• Using only dense features
  – word embedding
  – suffix embedding
  – capitalization

• The input layer is a concatenation of all embeddings of all words in a context window
Supertagging with a RNN

... bought some books and ...

Diagram showing a sequence of labeled nodes, with arrows indicating the flow of information.
Supertagging with a RNN

... bought some books and ...

... bought some books and ...
Supertagging with a RNN

... bought some books and ...
Supertagging with a RNN

... bought some books and ...
Supertagging with a RNN

... bought some books and ...
1-best Supertagging Results: dev

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>C&amp;C (gold POS)</td>
<td>92.60</td>
<td>-</td>
</tr>
<tr>
<td>C&amp;C (auto POS)</td>
<td>91.50</td>
<td>0.57</td>
</tr>
<tr>
<td>NN</td>
<td>91.10</td>
<td>21.00</td>
</tr>
<tr>
<td>RNN</td>
<td>92.63</td>
<td>-</td>
</tr>
<tr>
<td>RNN+dropout</td>
<td>93.07</td>
<td>2.02</td>
</tr>
</tbody>
</table>

Table 1: 1-best tagging accuracy and speed comparison on CCGBank Section 00 with a single CPU core (1,913 sentences), tagging time in secs.
1-best Supertagging Results: test

<table>
<thead>
<tr>
<th>Model</th>
<th>Section 23</th>
<th>Wiki</th>
<th>Bio</th>
</tr>
</thead>
<tbody>
<tr>
<td>C&amp;C (gold POS)</td>
<td>93.32</td>
<td>88.80</td>
<td>91.85</td>
</tr>
<tr>
<td>C&amp;C (auto POS)</td>
<td>92.02</td>
<td>88.80</td>
<td>89.08</td>
</tr>
<tr>
<td>NN</td>
<td>91.57</td>
<td>89.00</td>
<td>88.16</td>
</tr>
<tr>
<td>RNN</td>
<td>93.00</td>
<td>90.00</td>
<td>88.27</td>
</tr>
</tbody>
</table>

Table 2: 1-best tagging accuracy comparison on CCGBank Section 23 (2,407 sentences), Wikipedia (200 sentences) and Bio-GENIA (1,000 sentences).
Multi-tagging Results: dev

![Graph showing multi-tagging accuracy vs ambiguity level for different models: RNN + dropout (red), RNN (green), NN (blue), C&C (black)].
Multi-tagging Results: test

multi-tagging accuracy vs ambiguity level

RNN + dropout
NN
C&C
Table 3: Parsing test results (auto POS). We evaluate on all sentences (100% coverage) as well as on only those sentences that returned spanning analyses (% cov.). RNN and NN both have 100% coverage on the Wikipedia data.
Training & Experiments

- Mini-batched BPTT [Rumelhart et al., 1988; Mikolov, 2012]
- A context window-size of 7, a BPTT step size of 9, avg ambig 1.4
- 50-dim scaled Turian embeddings [Turian et al., 2010]
- Other two look-up tables randomly initialized
- Embedding fine-tuning during training
- Dropout regularization
- Parsing experiments: use the same supertagger prob. cutoff values as C&C
<table>
<thead>
<tr>
<th>Model</th>
<th>Development set</th>
<th>Test set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Acc</td>
<td>OOV</td>
</tr>
<tr>
<td>Clark et al. (2018)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CVT</td>
<td>—</td>
<td>0</td>
</tr>
<tr>
<td>ELMo-based</td>
<td>—</td>
<td>0</td>
</tr>
<tr>
<td>BiLSTM</td>
<td>96.15</td>
<td>0</td>
</tr>
<tr>
<td>PRIMDecoder</td>
<td><strong>96.27</strong></td>
<td><strong>11</strong></td>
</tr>
</tbody>
</table>

Table 1: Our model's word accuracy, OOV category word accuracy, parser F₁, and parser coverage on CCGbank, compared to the bidirectional LSTM classifier baseline and comparable results from the most recent previous work. All accuracies are averaged over 20 runs with different random seeds. Standard deviations range around 0.05% for word accuracy, 0.1% for F₁, 0.4% for BiLSTM coverage, and 0.2% for PRIMDecoder coverage. All improvements are statistically significant with $p \ll 0.001$. 