Overview

- Using a feature combination of:
  - local context information and
  - corpus-wide information
- State-of-the-art POS tagging accuracies
- PTB-WSJ: 97.51% (ours) vs. 97.50% (Søgaard, 2011)
- CoNLL2009: 98.02% (ours) vs. 97.84% (Bohnet and Nivre, 2012)

Four types of corpus-wide features

- Word embeddings (w2v and glv)
  - word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014)
- POS tag distribution (pos)
  - Pr(pos | wj), Pr(pos | affixj), Pr(pos | spellingj)
- Supertag distribution (stag)
  - Pr(stag | wj): Supertags are dependency labels and directions of parent/children, e.g. “nn/L” (Ouchi et al., 2014)
- Context word distribution (cw)
  - Pr(wt+1 | wj); Pr(wt+1 | wj); (Schnabel and Schütze, 2014)

Activation Functions

- Let \( v \) be a linear filter: \( v = \theta^T x \)
- Rectified Linear Units (ReLUs)
  \[ h = \max(v, 0) \]
- Maxout networks (MAXOUT)
  \[ h = \max(v_1, v_2, \ldots, v_n) \]
- Normalized \( L_p \) pooling (\( L_p \))
  \[ h = \left( \frac{1}{G} \sum_{j=1}^{G} |v_j|^p \right)^{1/p} \]

Results on Penn Treebank (PTB-WSJ)

- Evaluation of the hybrid model

<table>
<thead>
<tr>
<th>Neural Network Settings</th>
<th>#Hidden</th>
<th>Group size (G)</th>
<th>Development Set</th>
<th>Test Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear model</td>
<td>384</td>
<td></td>
<td>All / Unk.</td>
<td>All / Unk.</td>
</tr>
<tr>
<td>( L_p (p = 2) )</td>
<td>48</td>
<td>8</td>
<td>97.45 / 90.67</td>
<td>97.42 / 91.04</td>
</tr>
<tr>
<td>( L_p (p = 3) )</td>
<td>32</td>
<td>8</td>
<td>97.52 / 90.91</td>
<td>97.51 / 91.64</td>
</tr>
<tr>
<td>MAXOUT</td>
<td>48</td>
<td>8</td>
<td>97.51 / 90.91</td>
<td>97.51 / 91.53</td>
</tr>
<tr>
<td>( L_p (p = 2) ) (w/o linear part)</td>
<td>48</td>
<td>8</td>
<td>97.50 / 90.89</td>
<td>97.50 / 91.67</td>
</tr>
</tbody>
</table>

Why neural net. for continuous features?

- The non-linearity of discrete features has been exploited by the simple conjunction of the discrete features.
- In contrast, the non-linear feature design of continuous features is not intuitive.

Online learning of a left-to-right tagger

- Deterministically predicts each tag using prediction history (Choi and Palmer, 2012)
  - Binary features: N-grams, affix, spelling types, etc.
- A variant of the on-the-fly example generation algorithm (Goldberg and Nivre, 2012)
  - Using the prediction of the previously learned model as prediction history to overcome error propagation.
- FTRLProximal algorithm (McMahan, 2011) with Adagrad (Duchi et al., 2010)
  - Multi-class hinge loss + L1/L2 regularization terms
- Random hyper-parameter searches (Bergstra and Bengio, 2012)
  - Initial weights, initial weight range, momentum, learning rate, regularization, epoch to start the regularizations, etc. (256 initial weights are tried!)

Learned representations

Scatter plots of verbs for all combinations between the first 4 principal components of the raw features and the activation of hidden variables.

PCA of raw feature

- PCA of hidden activations
  - Known token accuracy (\%), Unknown token accuracy (\%)

<table>
<thead>
<tr>
<th>Feature engineering using linear model</th>
<th>Evaluation results of corpus-wide features on dev. set.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary feature only</td>
<td>+w2v: 97.36 / 88.96</td>
</tr>
<tr>
<td>97.15 / 86.81</td>
<td>+w2v+glv: 97.40 / 90.44</td>
</tr>
<tr>
<td>+glv: 97.34 / 89.55</td>
<td>+pos: 97.44 / 90.17</td>
</tr>
<tr>
<td>+stag(w=1): 97.45 / 90.53</td>
<td>+stag(w=3): 97.45 / 90.22</td>
</tr>
</tbody>
</table>

Neural Networks Leverage Corpus-wide Information for Part-of-speech Tagging

Yuta Tsuboi <yutat@jp.ibm.com>

IBM Research – Tokyo