How (not) to train a dependency parser: The curious case of jackknifing part-of-speech taggers

Željko Agić <zeag@itu.dk> and Natalie Schluter <natschluter@itu.dk>
Department of Computer Science, IT University of Copenhagen, Denmark

“The traditions in literature”

Tenfold jackknifing (n=10) of treebank part-of-speech tags is used indiscriminately as makeshift adaptation in dependency parsing.

Linear and n-fold jackknifing

Linear jackknifing permits exploring training sets with less than 50% of treebank data:

n=2 equals p=50%
p<50% is inaccessible to n

Adaptation thus suffers especially in low-resource scenarios.

Experiment setup

Data. 26 languages, overlap between UD v1.2 and WTC parallel corpus
Tagger and parsers. TnT tagger, Mate: graph-based, and Yara: transition-based
Tagging quality. GOLD, PRED: direct supervision (94.1%), PROJ: cross-lingual (71.7%)
Other. Results averaged over 5 randomized runs for all experiments

Monolingual parsing

Observations

Median \( p_{\text{max}} \) = 75% ; 5%
Median \( n_{\text{max}} \) = 11 ; 2
Fixing n=10 is suboptimal, -0.2 and -4.6 UAS to \( p_{\text{max}} \).

GOLD training provides the best parser for 0 languages.

PROJ\( \rightarrow \)PROJ is by far the best low-resource option.

Delexicalized transfer

Best choice for # of 26 languages

<table>
<thead>
<tr>
<th></th>
<th>PRED</th>
<th>PROJ</th>
</tr>
</thead>
<tbody>
<tr>
<td>GOLD</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( n_{\text{max}} )</td>
<td>18</td>
<td>0</td>
</tr>
<tr>
<td>( p_{\text{max}} )</td>
<td>21</td>
<td>0</td>
</tr>
</tbody>
</table>

Observations

The unadapted GOLD\( \rightarrow \)PROJ parsers perform better.

\(+1.1\); +0.1 UAS with Mate
\(+2.1\); +0.7 UAS with Yara

Best choice for 14-17/26 langs.