Lexical Normalization for Neural Network Parsing

Rob van der Goot, Gertjan van Noord
University of Groningen
r.van.der.goot@rug.nl

26-01-2018
Last Year (CLIN27)

kheb da gzien

orig → kheb
lookup → da
w2v → gzie
aspell → gezien

ik heb dat gezien

+ N-grams

classifier
Use normalization to adapt neural network dependency parsers
Evaluate the effect of normalization versus externally trained word embeddings and character level models
See if we can exploit top-n candidates
New treebank to evaluate domain adaptation
New Treebank

Why?

- Manually corrected train data
- Gold normalization available
- Data should be non-canonical
- UD format
New Treebank

- Pre-filtered to contain non-standard words
- Data from Li and Liu (2015): Owoputi and LexNorm
- 600 Tweets / 10,000 words
- UD2.1 format
I feel so bad...
Not so sure about wrk tommara
New Treebank

Experimental setup:
- Train: English Web Treebank
- Dev: Owoputi
- Test: Lexnorm
Neural Network parser

Configuration:

<table>
<thead>
<tr>
<th>s2</th>
<th>s1</th>
<th>s0</th>
<th>b0</th>
<th>b1</th>
<th>b2</th>
<th>b3</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>jumped</td>
<td>over</td>
<td>the</td>
<td>lazy</td>
<td>dog</td>
<td>ROOT</td>
</tr>
</tbody>
</table>

Scoring:

\[(\text{ScoreLeftArc}, \text{ScoreRightArc}, \text{ScoreShift})\]

Taken from Kiperwasser and Goldberg (2016)
Neural Network parser

UUparser (de Lhoneux et al., 2017)
- Performs well
- Relatively easy to adapt
- No POS tags
- Characters + external embeddings
Neural Network parser

\[ \vec{t}_1, \vec{c}_1, \vec{e}_1 \]

\[ \vec{t}_2, \vec{c}_2, \vec{e}_2 \]

\[ \vec{t}_3, \vec{c}_3, \vec{e}_3 \]
Use Normalization as Pre-processing
Use Normalization as Pre-processing

```
root

amod  obj  obl
new pix  coming  tomorrow
new pix  comming  tomarrooe
```
Use Normalization as Pre-processing
Use Normalization as Pre-processing
Use Normalization as Pre-processing
Use Normalization as Pre-processing
Use Normalization as Pre-processing
Integrate Normalization

new pix comming tomorro
# Integrate Normalization

<table>
<thead>
<tr>
<th>new</th>
<th>pix</th>
<th>comming</th>
<th>tomoroe</th>
</tr>
</thead>
<tbody>
<tr>
<td>new</td>
<td>0.9466</td>
<td>0.7944</td>
<td>0.5684</td>
</tr>
<tr>
<td>news</td>
<td>0.0315</td>
<td>0.0882</td>
<td>0.4314</td>
</tr>
<tr>
<td>knew</td>
<td>0.0111</td>
<td>0.0559</td>
<td>0.0002</td>
</tr>
<tr>
<td>now</td>
<td>0.0063</td>
<td>0.0449</td>
<td>&lt;0.0001</td>
</tr>
<tr>
<td>newt</td>
<td>0.0045</td>
<td>0.0165</td>
<td>&lt;0.0001</td>
</tr>
</tbody>
</table>
Integrate Normalization

\[ \vec{t}_1 \vec{c}_1 \vec{e}_1 \]
\[ \vec{t}_2 \vec{c}_2 \vec{e}_2 \]
\[ \vec{t}_3 \vec{c}_3 \vec{e}_3 \]

LSTM $^f$

LSTM $^b$

word1

word2

word3
Integrate Normalization

\[ \vec{w}_i = \sum_{j=0}^{n} P_{ij} \ast \vec{n}_{ij} \]
Integrate Normalization

\[ \vec{w}_1 = (\text{new} \times 0.9466) + (\text{news} \times 0.0315) + (\text{knew} \times 0.0111) + (\text{now} \times 0.0063) + (\text{newt} \times 0.0045) \]
Integrate Normalization

![Graph showing integration of normalization methods: base, +char, +text, +char+text, with raw, norm., integr., and gold labels. The graph compares different normalization techniques against a baseline (base) and shows improvements with the addition of character (+char) and text (+text) features, reaching highest improvements with character and text combined (+char+text).]
Integrate Normalization

But what about in-domain performance?
Integrate Normalization

![Bar chart showing 'base' and '+norm' categories]

- 'base': 84
- '+norm': 80

25 / 31
Integrate Normalization

+8.42e1

base
+norm
Integrate Normalization

Test data:

<table>
<thead>
<tr>
<th>Model</th>
<th>UAS</th>
<th>LAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td>70.47</td>
<td>60.16</td>
</tr>
<tr>
<td>normalization-direct</td>
<td>71.03*</td>
<td>61.83*</td>
</tr>
<tr>
<td>integrated</td>
<td>71.15</td>
<td>62.30*</td>
</tr>
<tr>
<td>gold</td>
<td>71.45</td>
<td>63.16*</td>
</tr>
</tbody>
</table>

Table: *indicates statistical significance compared to previous entry.
Conclusions:

- Normalization is still helpful on top of character and external embeddings
- Integrating normalization leads to a small but consistent/significant improvement
- Performance +60% from using gold normalization
- New dataset will be made available, provides a nice benchmark for domain adaptation
Effect of different categories of normalization replacements
Get closer to gold normalization


Integrate Normalization

- Foster: not noisy, constituency
- Denoised Web Treebank: no train
- Tweebank: no train
- Foreebank: not noisy