Named Entity Recognition for Chinese Social Media with Jointly Trained Embeddings

Nanyun Peng, Mark Dredze
Human Language Technology Center of Excellence
Center for Language and Speech Processing
Johns Hopkins University, Baltimore, Maryland USA

Named Entity Recognition
- Detecting boundaries and classifying types of text chunks that correspond to entities:
  - persons, organizations, locations

Challenges of Chinese
- No word boundaries
- Logograms -- lack NER cues such as capitalization and punctuation marks

Challenges of Social Media
- Many new words (OOV)
- Different dialects, jargons, writing systems mixed together
- Foreign names
- Spelling errors, typos, etc.

Dataset
1890 Weibo messages annotated by Mechanical Turk
- Data from Nov 2013 - Dec 2014
- Data split: 5/7 train, 1/7 dev, 1/7 test.
- 2,259,434 unlabeled weibos for training embeddings.

Code and data available at: https://github.com/hltcoe/goldenhorse

Embeddings
- Represent each word in a continuous low dimensional space
- Encodes lexical semantics:
  - similar words have similar representations

Previous work showed embedding features help NER.
- Embeddings indicate whether words are likely names, especially helpful for OOVs
  - eg. Flowers as person names (both English and Chinese):
    - Lily, Rose, Violet, Daisy, Jasmine......
    - 百合, 玫瑰, 菊, 茉莉......
  - Learn “flowers can be a person name” from NER training data
  - Propagate the information through unlabeled data.

Joint training schema
Jointly train embeddings and the traditional CRF objective:

$$\mathcal{L}_1(\lambda, e_w) = \frac{1}{K} \sum_{k=1}^{K} \left[ \log \frac{1}{Z(x^k)} + \sum_{i=1}^{l} \lambda_i f_i(y^k_i, x^k, e_w) \right]$$

A log-bilinear model

The skip-gram word embedding objective:

$$\mathcal{L}_2(e_w) = \frac{1}{J} \sum_{l=1}^{L} \sum_{j=1}^{l} \log p(w_l|w_j) - \frac{\exp(e_w^T e_{w_j})}{\sum \exp(e_w^T e_{w_j})}$$

Combine them:
\[
\arg \max_{e_w} = \mathcal{L}_1(\lambda, e_w) + C \mathcal{L}_2(e_w)
\]

Embeddings for Chinese
- Chinese does not have word boundaries; learning word embeddings is a challenge
- The state-of-the-art Chinese NER systems are character-based
- Explored three types of embeddings:
  - Character embeddings
  - Word embeddings
  - Char-position embeddings


<table>
<thead>
<tr>
<th>Method</th>
<th>Without Fine Tuning</th>
<th>With Fine Tuning</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
</tr>
<tr>
<td>Stanford</td>
<td>63.51</td>
<td>23.27</td>
<td>34.06</td>
</tr>
<tr>
<td>Baseline Features</td>
<td>63.51</td>
<td>27.17</td>
<td>38.06</td>
</tr>
<tr>
<td>+ word</td>
<td>60.87</td>
<td>32.37</td>
<td>42.26</td>
</tr>
<tr>
<td>+ character</td>
<td>65.71</td>
<td>26.59</td>
<td>37.86</td>
</tr>
<tr>
<td>+ character-position Joint (cp)</td>
<td>72.39</td>
<td>31.80</td>
<td>44.19</td>
</tr>
<tr>
<td>+ word</td>
<td>69.66</td>
<td>33.55</td>
<td>45.29</td>
</tr>
<tr>
<td>+ character</td>
<td>73.43</td>
<td>34.88</td>
<td>47.30</td>
</tr>
<tr>
<td>+ character-position Joint (cp)</td>
<td>72.55</td>
<td>36.88</td>
<td>48.90</td>
</tr>
</tbody>
</table>

NER results for named mentions (top) and name + nominal mentions (bottom) on weibo data.