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

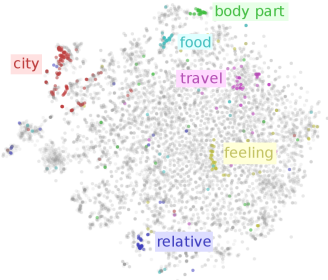
- **Detecting** boundaries and **classifying** types of text chunks that correspond to entities:
  - *persons, organizations, locations*

成都(GPE.NAM)电信(ORG.NAM)到底有没的时间观念哦，一托再托，日妈(PER.NOM)我们时间就不是时间哇，等了你两天啥子速度。
Chengdu(GPE.NAM) Telecom(ORG.NAM) do you have no concept of time, delay again and again, mother(PER.NOM) fxxxer our time is not time, waited for you for two days what a speed..

- **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
  - Foreigh names
  - Spelling errors, typos, etc.

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, 堇, 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:

$$\mathcal{L}_s(\boldsymbol{\lambda}, \boldsymbol{e_w}) = \frac{1}{K}\sum_k \left[\log \frac{1}{Z(x)^k} + \sum_j \lambda_j F_j(\boldsymbol{y}^k, \boldsymbol{x}^k, \boldsymbol{e_w})\right]$$

A log-bilinear model

The skip-gram word embedding objective:

$$\mathcal{L}_u(\boldsymbol{e_w}) = \frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c, j\neq 0} \log p(w_{t+j}|w_t) \text{ where } p(w_i|w_j) = \frac{\exp\left(e_{w_i}^T e_{w_j}\right)}{\sum_{i'}\exp\left(e_{w_{i'}}^T e_{w_j}\right)}$$

Combine them:

$$\arg\max_{e_w} = \mathcal{L}_s(\boldsymbol{\lambda}, \boldsymbol{e_w}) + C\mathcal{L}_u(\boldsymbol{e_w})$$

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 有 没 有
  - Word embeddings 有 没有
  - Char-position embeddings 有$_0$ 没$_0$ 有$_1$

Dataset

1890 Weibo messages annotated by Mechanical Turk

Entity Type	Mentions: Name	Nominal	Total
Geo-political	243	0	243
Location	88	38	126
Organization	224	31	255
Person	721	636	1,357

- 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

Method	Dev: Without Fine Tuning P	R	F1	Dev: With Fine Tuning P	R	F1	Test: Without Fine Tuning P	R	F1	Test: With Fine Tuning P	R	F1
Stanford	63.51	23.27	34.06				55.70	22.86	33.06			
Baseline Features	63.51	27.17	38.06				56.98	25.26	35.00			
+ word	65.71	26.59	37.86	70.97	25.43	37.45	56.82	25.77	35.46	64.94	25.77	36.90
+ character	53.54	30.64	38.97	58.76	32.95	42.22	56.48	31.44	40.40	57.89	34.02	42.86
+ character+position	60.87	32.37	42.26	61.76	36.42	**45.82**	61.90	33.51	43.48	57.26	34.53	43.09
Joint (cp)				57.41	35.84	44.13				57.98	35.57	**44.09**
Stanford	72.39	31.80	44.19				63.96	22.19	32.95			
Baseline Features	71.94	33.22	45.45				60.16	23.87	34.18			
+ word	69.66	33.55	45.29	70.67	35.22	47.01	59.40	25.48	35.67	60.68	22.90	33.26
+ character	58.76	32.95	42.22	66.88	35.55	46.42	58.28	28.39	38.18	55.15	29.35	38.32
+ character+position	73.43	34.88	47.30	69.38	36.88	48.16	65.91	28.06	39.37	62.33	29.35	39.91
Joint (cp)				72.55	36.88	**48.90**				63.84	29.45	**40.38**

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