Exploring Segment Representations for Neural Segmentation Models

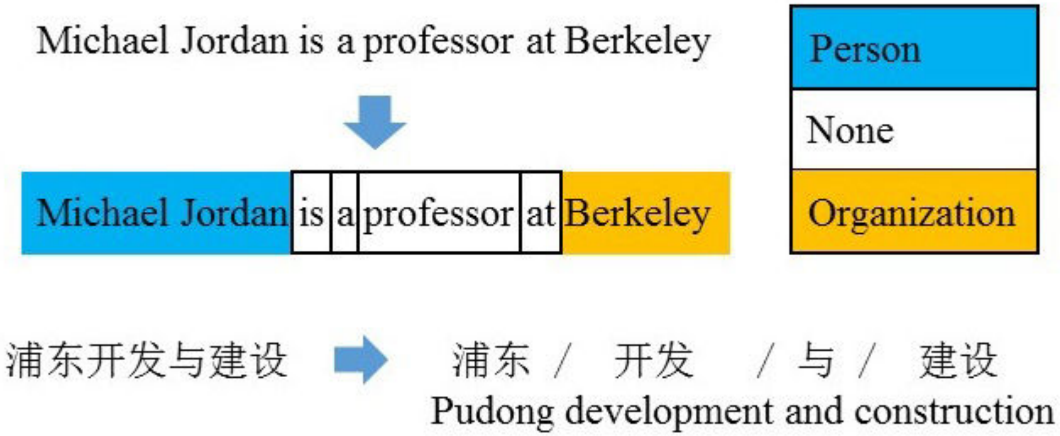
Yijia Liu, Wanxiang Che, Jiang Guo, Bing Qin, and Ting Liu

Research Center for Social Computing and Information Retrieval

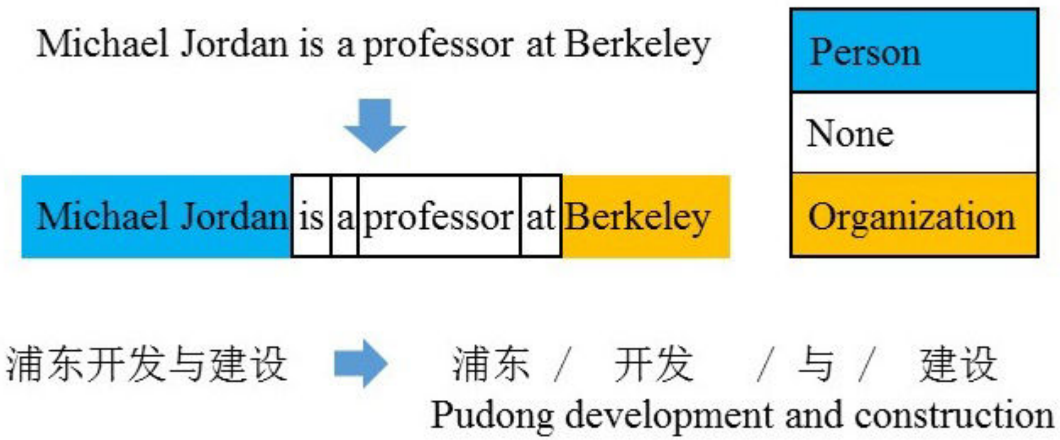
Harbin Institute of Technology

Problem:

Natural Language Segmentation Problem



Problem:
Natural Language Segmentation Problem



- ***input*** *is* a sequence of elements
- ***segmentation*** is a sequence of segment $\mathbf{S} = (s_1, s_2, \dots, s_p)$
- a ***segment*** is a tuple $s = (u, v, y)$
  - u: the beginning position
  - v: the ending position
  - y: the label associated with the segment (optional)
- constrained on $v_i + 1 = u_{i+1}$

Motivating:
Can we use word embedding in CWS?

浦东开发与建设 → 浦东 / 开发 / 与 / 建设

Pudong development and construction

Motivating:
Can we use word embedding in CWS?

- To achieve this gold, we need
  - to access the segment (the potential word) during inference
  - to represent the segment

Motivating:
Can we use word embedding in CWS?

- To achieve this gold, we need
  - to access the segment (the potential word) during inference
  - to represent the segment

in “浦东开发与建设”
“浦东” is a potential word

structure prediction

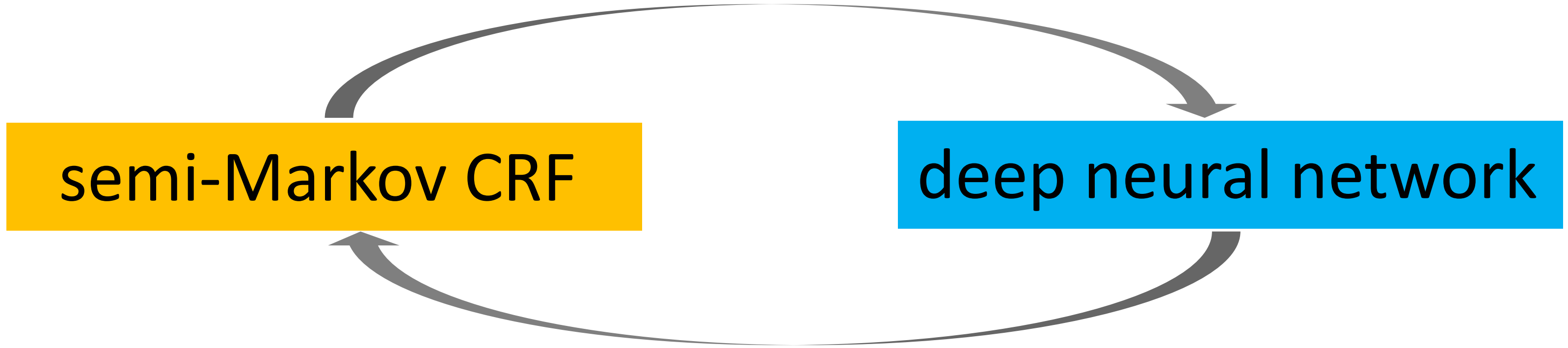
segment representation

“浦东”: [0.5, 0.3, 0.6, …]
“虹桥”: [0.5, 0.2, 0.5, …]
they have similar syntactic/semantic function.

Motivating:
Can we use word embedding in CWS?

- To achieve this gold, we need
  - to access the segment (the potential word) during inference
  - to represent the segment

in “浦东开发与建设”
“浦东” is a potential word



“浦东”: [0.5, 0.3, 0.6, …]
“虹桥”: [0.5, 0.2, 0.5, …]
they have similar syntactic/semantic function.

Refresh on semi-CRF

- semi-CRF model the conditional probability of S as
  - $p(S|X) = \frac{1}{Z} \exp W\Phi(S, X)$
- by restricting features within one certain segment, $\Phi(S, X)$ can be decomposed as $\sum_i^p \phi(s_i, X)$
- core problem in achieving good segment performance

Representing $\phi(s_i, X)$

(Old-school) $\phi(s_i, X)$ representation

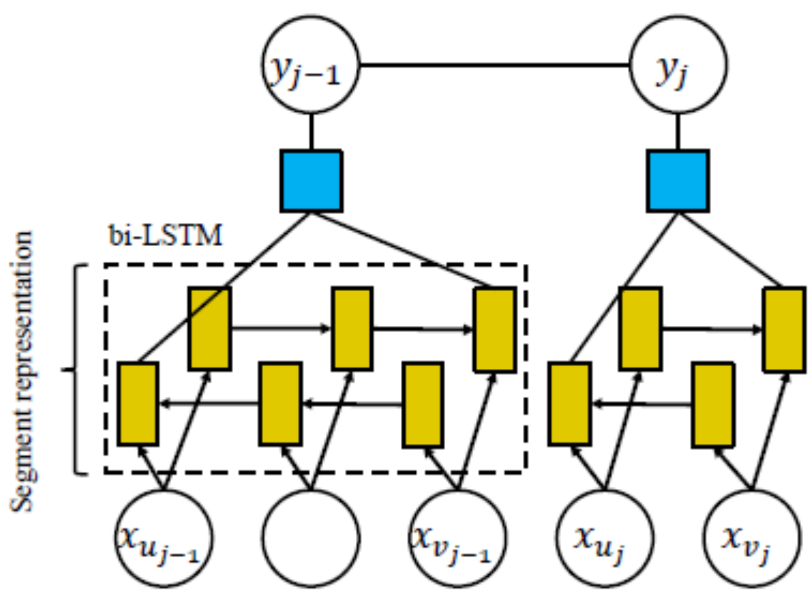
- *crf styled features*:
  - input unit level information
  - e.g.: character
- *semi-crf styled features*:
  - segment-level information
  - e.g.: length of the segment
- require carefully designed features and do not generalize well

Neuralized $\phi(s_i, X)$ representing

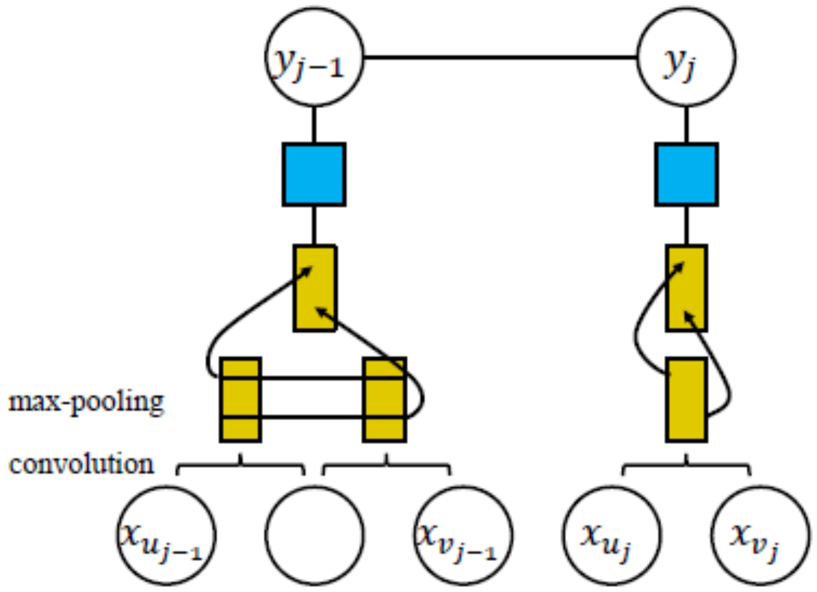
- *neural crf styled features*:
  - composing the representation of input units into a vector
  - handling variable length nature
- *neural semi-crf styled features:*
  - embed the entire segment
  - learning from labeled/unlabeled data

Composing Input Units

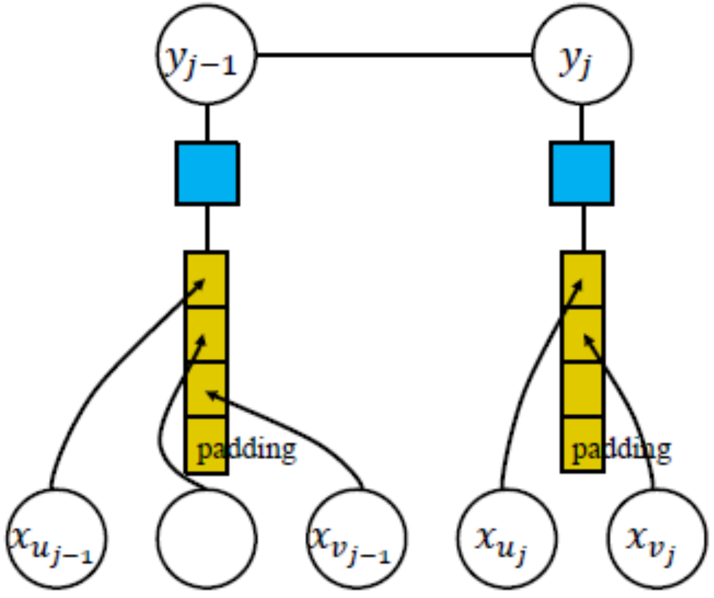
- *neural crf styled features*: $\boldsymbol{SCOMP_i = Net(x_u, x_{u+1}, \dots, x_v)}$
  - composing the representation of input units into a vector
  - handling variable length nature



SRNN



SCNN

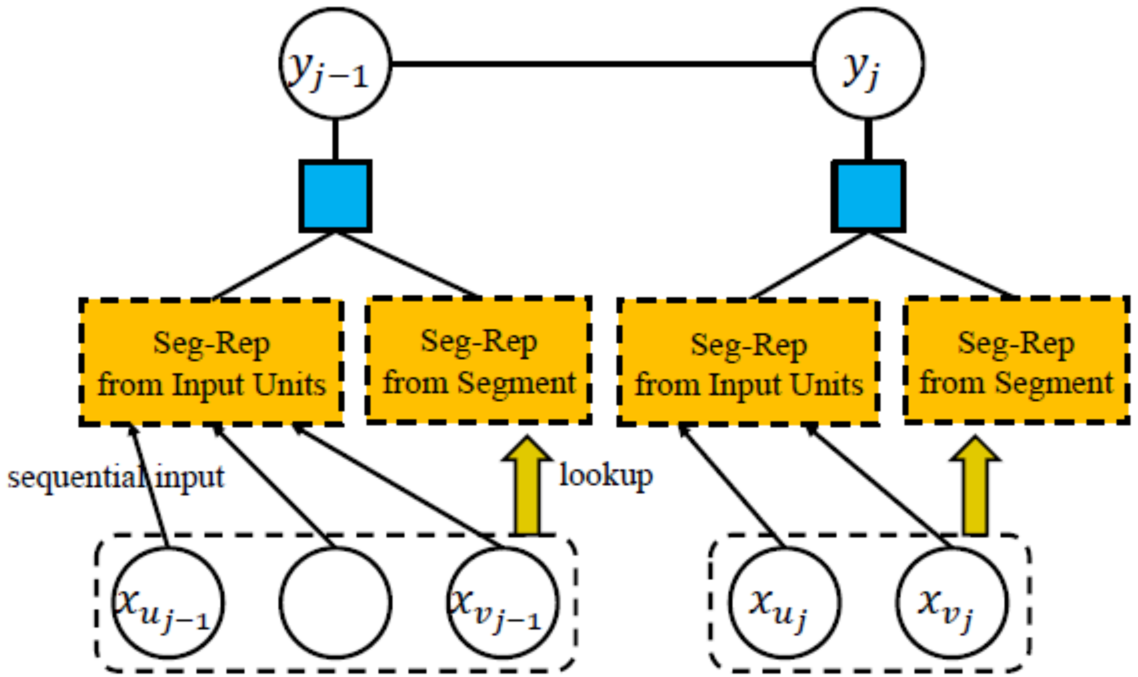


SCONCATE

Embedding Entire Segment

- *neural semi-crf styled features:* $\boldsymbol{SEMB_i = lookup(x_u x_{u+1} \ldots x_v)}$
  - embed the entire segment
  - learning from raw text
- Problem: where did the embedding come from?
- Answer: learning from unlabeled but auto-segmented data

Final Model


y_{j-1}
y_j
Seg-Rep from Input Units
Seg-Rep from Segment
Seg-Rep from Input Units
Seg-Rep from Segment
sequential input
lookup
$x_{u_{j-1}}$
$x_{v_{j-1}}$
x_{u_j}
x_{v_j}

Experiments

- Two typical NLP segmentation tasks: NER and CWS
  - NER: CoNLL2003
  - CWS: Sighan2005
- Baselines:
  - sparse feature CRF
  - neural sequence labeling
  - neural CRF

w/ Input Units Composition only

	model	NER CoNLL03 dev	NER CoNLL03 test	CWS CTB6 dev	CWS CTB6 test	CWS PKU dev	CWS PKU test	CWS MSR dev	CWS MSR test	spd
baseline	NN-LABELER	93.03	88.62	93.70	93.06	93.57	92.99	93.22	93.79	**3.30**
	NN-CRF	**93.06**	**89.08**	94.33	93.65	94.09	93.28	93.81	94.17	2.72
	SPARSE-CRF	88.87	83.43	**95.68**	**95.08**	**95.85**	**95.06**	**96.09**	**96.54**	
neural semi-CRF	SRNN	92.97	88.63	94.56	94.06	94.86	93.91	94.38	95.21	0.62
	SCONCATE	92.96	89.07	94.34	93.96	94.41	93.57	94.05	94.53	1.08
	SCNN	91.53	87.68	87.82	87.51	79.64	80.75	85.04	85.79	1.46

- structure predication models outperform classification
- but difference is not significant within structure models

w/ Input Units Composition + segment embedding

- Using segment-level representation greatly improve the performance

model	CoNLL03	CTB6	PKU	MSR
NN-LABELER	88.62	93.06	92.99	93.79
NN-CRF	89.08	93.65	93.28	94.17
SPARSE-CRF	83.43	95.08	95.06	96.54
SRNN	88.63	94.06	93.91	95.21
+SEMB-HETERO	89.59	**95.48**	95.60	97.39
	+0.96	*+1.42*	*+1.69*	*+2.18*
SCONCATE	89.07	93.96	93.57	94.53
+SEMB-HETERO	**89.77**	95.42	**95.67**	**97.58**
	+0.70	*+1.43*	*+2.10*	*+3.05*

Final Result (compare w/ NER SOTA)

- achieve comparable performance without domain-specific knowledge

genre	*model*	CoNLL03
NN	[Collobert *et al.*, 2011]	89.59
	[Huang *et al.*, 2015]	90.10
non-NN	[Ando and Zhang, 2005]	89.31
	[Guo *et al.*, 2014]	88.58
	[Passos *et al.*, 2014]	**90.90**
	our best	89.77

Final Result (compare w/ CWS SOTA)

- achieve SOTA on two datasets

genre	*model*	CTB6	PKU	MSR
NN	[Zheng *et al.*, 2013]	-	92.4	93.3
	[Pei *et al.*, 2014]		94.0	94.9
	[Pei *et al.*, 2014] w/bigram	-	95.2	97.2
	[Kong *et al.*, 2015]		90.6	90.7
non-NN	[Tseng, 2005]	-	95.0	96.4
	[Zhang and Clark, 2007]	-	95.1	97.2
	[Sun *et al.*, 2009]	-	95.2	97.3
	[Wang *et al.*, 2011]	**95.7**	-	-
	our best	95.48	**95.67**	**97.58**

Conclusion

- We thoroughly study representing the segment in neural semi-CRF
- SCONCATE is comparable with SRNN
- Segment embedding greatly improve the performance
- Our code can be found at: https://github.com/ExpResults/segrep-for-nn-semicrf
- Talk to me at the poster for more details

Thanks and Questions!