Stack-Pointer Networks for Dependency Parsing

Xuezhe Ma, Zecong Hu, Jingzhou Liu, Graham Neubig, ...

CMU -- ACL18

AntNLP -- Tao Ji

taoji.cs@gmail.com

Outline

- Motivation
- Architecture
  - Pointer Networks & Biaffine
  - Encoder & Decoder
- Experiments

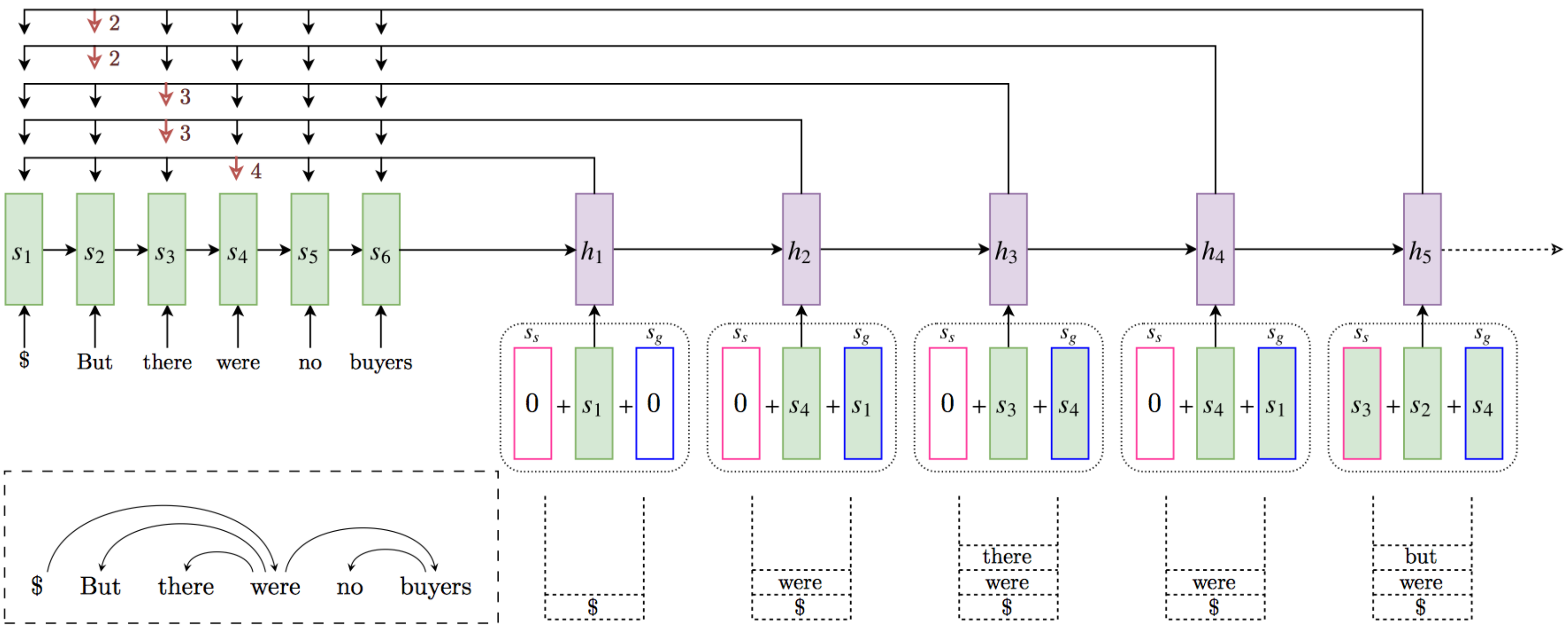
Motivation

- They introduce a novel architecture: stack-pointer networks.
- Combining pointer networks with an internal stack.

Contribution

- They propose a neural network architecture for dependency parsing that is simple, effective, and efficient.
- Empirical evaluations on benchmark datasets over 20 languages show that their method achieves state-of-the-art performance on 21 different treebanks.
- Comprehensive error analysis is conducted to compare the proposed method to a strong graph-based baseline using biaffine attention (Dozat and Manning, 2017).

Architecture



Pointer Networks

- $e_i^t = score(s_t \to h_i)$
- $p^t = softmax(e^t)$
- $score(\cdot \to \cdot)$ is the *attention scoring function*

Biaffine Attention Mechanism

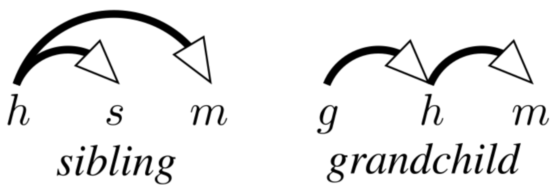
- $e_i^t = s_t^\top \mathbf{W} h_i + \mathbf{U}^\top s_t + \mathbf{V}^\top h_i + \mathbf{b}$

Encoder

- BLSTM-CNNs architecture

Decoder

- Order: `left-to-right` or `inside-out`
  - `inside-out`: utilize second-order sibling information.
- Higher-order Information:



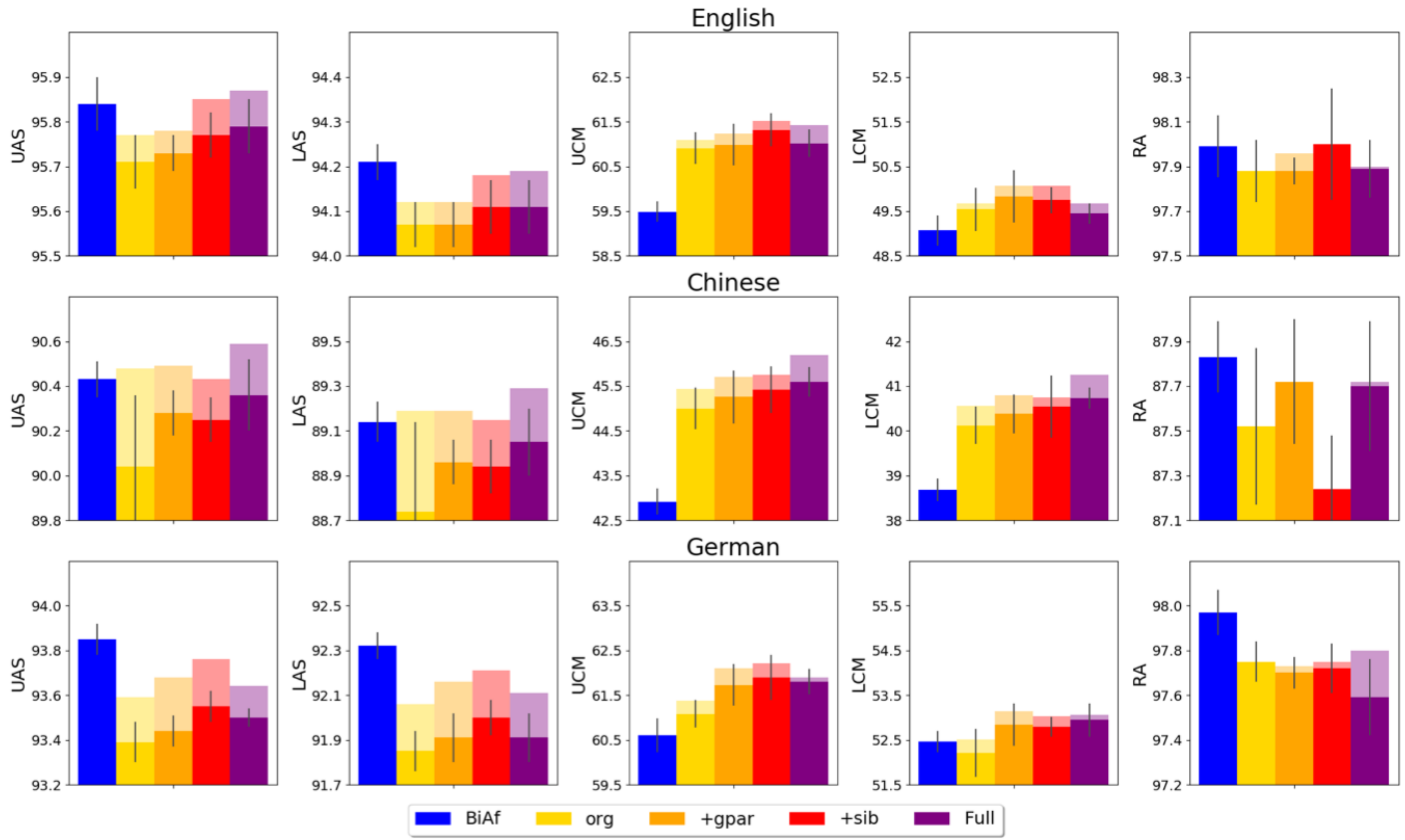
- Input: $\beta_t = \boldsymbol{s}_h + \boldsymbol{s}_g + \boldsymbol{s}_s$

Experiments

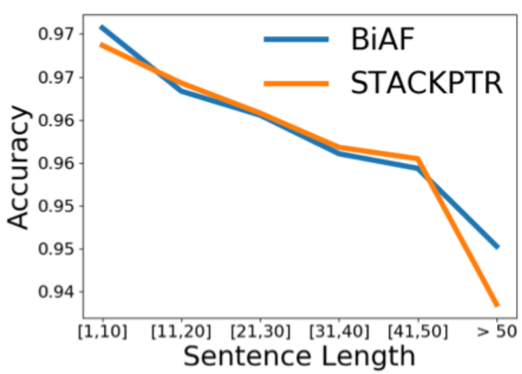
Main Results

System		English UAS	English LAS	Chinese UAS	Chinese LAS	German UAS	German LAS
Chen and Manning (2014)	T	91.8	89.6	83.9	82.4	–	–
Ballesteros et al. (2015)	T	91.63	89.44	85.30	83.72	88.83	86.10
Dyer et al. (2015)	T	93.1	90.9	87.2	85.7	–	–
Bohnet and Nivre (2012)	T	93.33	91.22	87.3	85.9	91.4	89.4
Ballesteros et al. (2016)	T	93.56	91.42	87.65	86.21	–	–
Kiperwasser and Goldberg (2016)	T	93.9	91.9	87.6	86.1	–	–
Weiss et al. (2015)	T	94.26	92.41	–	–	–	–
Andor et al. (2016)	T	94.61	92.79	–	–	90.91	89.15
Kiperwasser and Goldberg (2016)	G	93.1	91.0	86.6	85.1	–	–
Wang and Chang (2016)	G	94.08	91.82	87.55	86.23	–	–
Cheng et al. (2016)	G	94.10	91.49	88.1	85.7	–	–
Kuncoro et al. (2016)	G	94.26	92.06	88.87	87.30	91.60	89.24
Ma and Hovy (2017)	G	94.88	92.98	89.05	87.74	92.58	90.54
BIAF: Dozat and Manning (2017)	G	95.74	94.08	89.30	88.23	93.46	91.44
BIAF: re-impl	G	95.84	**94.21**	90.43	89.14	**93.85**	**92.32**
STACKPTR: Org	T	95.77	94.12	90.48	89.19	93.59	92.06
STACKPTR: +gpar	T	95.78	94.12	90.49	89.19	93.65	92.12
STACKPTR: +sib	T	95.85	94.18	90.43	89.15	93.76	92.21
STACKPTR: Full	T	**95.87**	94.19	**90.59**	**89.29**	93.65	92.11

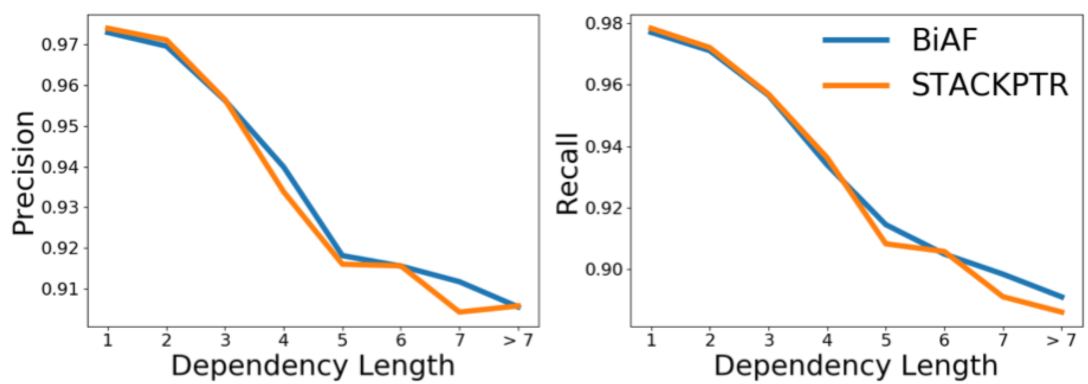
Additional Results



Additional Results

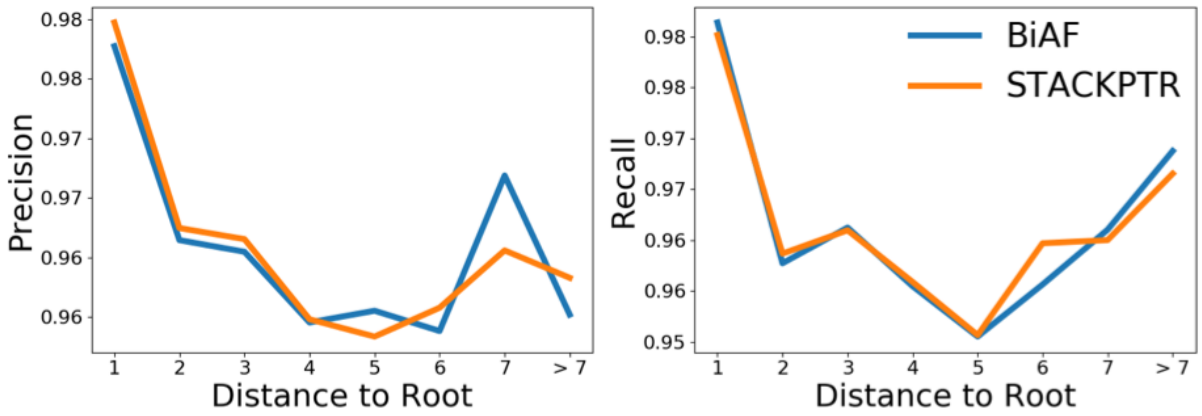


(a)



(b)

Additional Results



(c)

POS	UAS	LAS	UCM	LCM
Gold	96.12±0.03	95.06±0.05	62.22±0.33	55.74±0.44
Pred	95.87±0.04	94.19±0.04	61.43±0.49	49.68±0.47
None	95.90±0.05	94.21±0.04	61.58±0.39	49.87±0.46

Additional Results

Layer	Hyper-parameter	Value
CNN	window size	3
	number of filters	50
LSTM	encoder layers	3
	encoder size	512
	decoder layers	1
	decoder size	512
MLP	arc MLP size	512
	label MLP size	128
Dropout	embeddings	0.33
	LSTM hidden states	0.33
	LSTM layers	0.33
Learning	optimizer	Adam
	initial learning rate	0.001
	(β_1, β_2)	(0.9, 0.9)
	decay rate	0.75
	gradient clipping	5.0

Table 5: Hyper-parameters for all experiments.

Thank you!

Q&A