Deep Learning and Lexical, Syntactic and Semantic Analysis

Wanxiang Che (HIT)

Yue Zhang (SUTD)

Part 4: Dynamic Programming Decoding

Part 4.1: Dynamic Programming Decoding for Tagging

Word-Level Log-Likelihood

Approach	POS (PWA)	Chunking (F1)	NER (F1)	SRL (F1)
Benchmark Systems	97.24	94.29	89.31	77.92
NN+WLL	96.31	89.13	79.53	55.40

- WLL: Word-Level Log-Likelihood
  - Each word in a sentence is considered independently

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural Language Processing (Almost) from Scratch. J. Mach. Learn. Res. 12 (November 2011), 2493-2537.

Sentence-Level Log-Likelihood

- Considering dependencies between tags in a sentence
- Conditional likelihood by normalizing all possible paths (CRF)
- Sentence score for one tag path

$$\log p([y]_1^T \mid [\boldsymbol{x}]_1^T, \tilde{\boldsymbol{\theta}}) = s([\boldsymbol{x}]_1^T, [y]_1^T, \tilde{\boldsymbol{\theta}}) - \underset{\forall [j]_1^T}{\operatorname{logadd}}\, s([\boldsymbol{x}]_1^T, [j]_1^T, \tilde{\boldsymbol{\theta}})$$

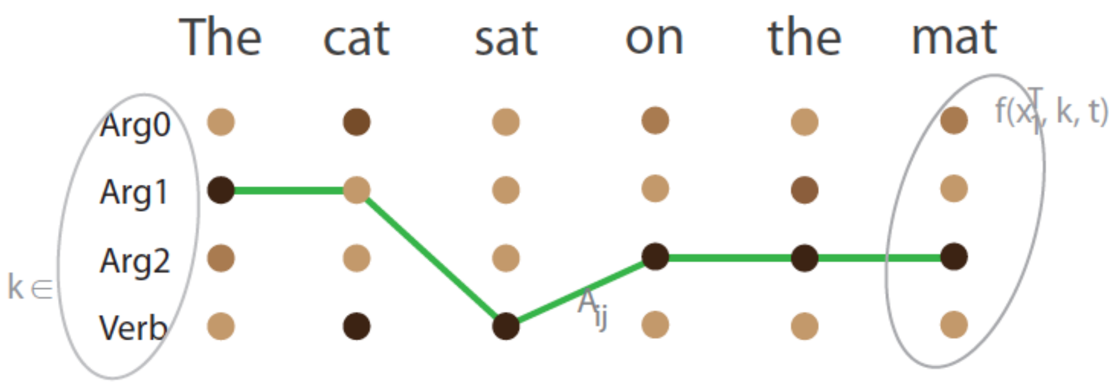
– where $A_{[i][j]}$ is a transition score for jumping from tag *i* to *j*

$$s([\boldsymbol{x}]_1^T, [i]_1^T, \tilde{\boldsymbol{\theta}}) = \sum_{t=1}^{T} \left(A_{[i]_{t-1}[i]_t} + f([\boldsymbol{x}]_1^T, [i]_t, t, \boldsymbol{\theta}) \right)$$

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural Language Processing (Almost) from Scratch. J. Mach. Learn. Res. 12 (November 2011), 2493-2537.

Sentence-Level Log-Likelihood

- Decoding: finding the max scored path
  - Viterbi algorithm



Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural Language Processing (Almost) from Scratch. J. Mach. Learn. Res. 12 (November 2011), 2493-2537.

Results

Approach	POS (PWA)	Chunking (F1)	NER (F1)	SRL (F1)
Benchmark Systems	97.24	94.29	89.31	77.92
NN+WLL	96.31	89.13	79.53	55.40
NN+SLL	96.37	90.33	81.47	70.99

- SLL helps, but fair performance for POS

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural Language Processing (Almost) from Scratch. J. Mach. Learn. Res. 12 (November 2011), 2493-2537.

Improvements

- Supervised word embeddings

Approach	POS (PWA)	CHUNK (F1)	NER (F1)	SRL (F1)
Benchmark Systems	97.24	94.29	89.31	77.92
NN+WLL	96.31	89.13	79.53	55.40
NN+SLL	96.37	90.33	81.47	70.99
NN+WLL+LM1	97.05	91.91	85.68	58.18
NN+SLL+LM1	97.10	93.65	87.58	73.84
NN+WLL+LM2	97.14	92.04	86.96	58.34
NN+SLL+LM2	97.20	93.63	88.67	74.15

- More (embedding) features

Approach	POS (PWA)	CHUNK (F1)	NER (F1)	SRL
Benchmark Systems	97.24	94.29	89.31	77.92
NN+SLL+LM2	97.20	93.63	88.67	74.15
NN+SLL+LM2+Suffix2	97.29	–	–	–
NN+SLL+LM2+Gazetteer	–	–	89.59	–
NN+SLL+LM2+POS	–	94.32	88.67	–
NN+SLL+LM2+CHUNK	–	–	–	74.72

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural Language Processing (Almost) from Scratch. J. Mach. Learn. Res. 12 (November 2011), 2493-2537.

Speed

System	RAM (Mb)	Time (s)
Toutanova, 2003	1100	1065
Shen, 2007	2200	833
SENNA	32	4

(a) POS

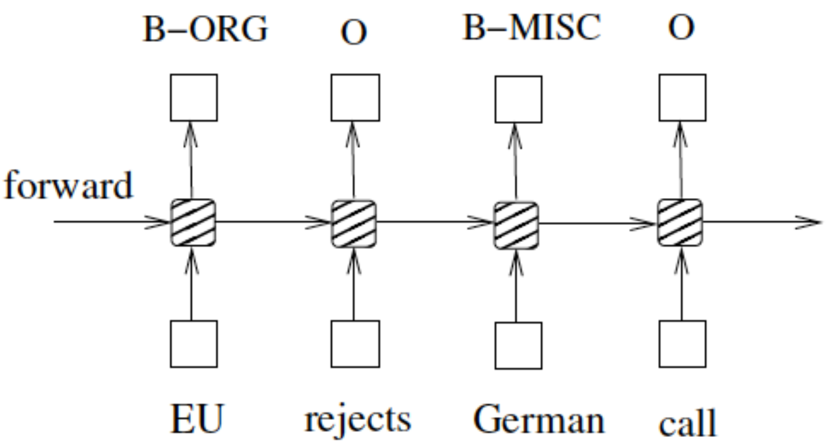
System	RAM (Mb)	Time (s)
Koomen, 2005	3400	6253
SENNA	124	52

(b) SRL

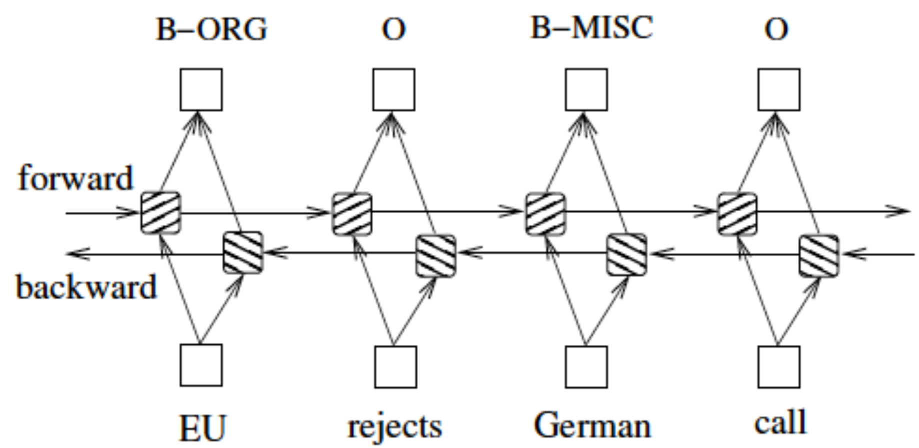
Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural Language Processing (Almost) from Scratch. J. Mach. Learn. Res. 12 (November 2011), 2493-2537.

RNNs for Tagging

- LSTM

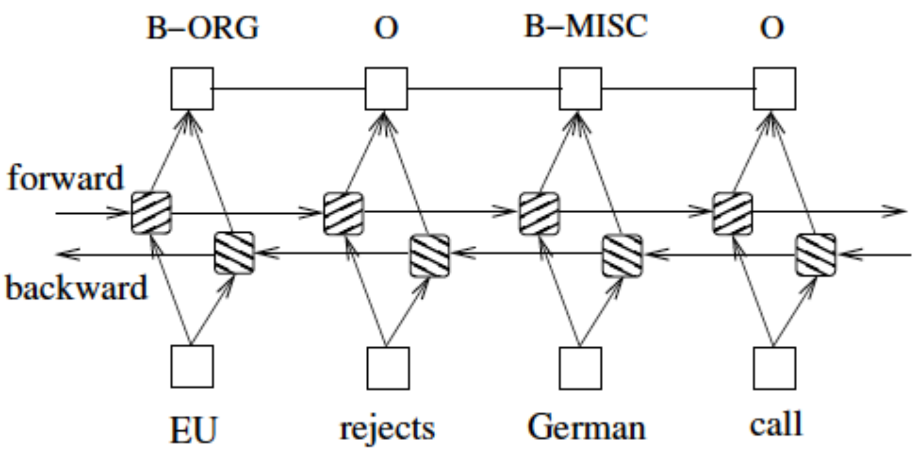


- Bi-LSTM



Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional LSTM-CRF models for sequence tagging. CoRR, abs/1508.01991, 2015.

Bi-LSTM-CRF



Algorithm 1 Bidirectional LSTM CRF model training procedure

```
for each epoch do
    for each batch do
        1) bidirectional LSTM-CRF model forward pass:
            forward pass for forward state LSTM
            forward pass for backward state LSTM
        2) CRF layer forward and backward pass
        3) bidirectional LSTM-CRF model backward pass:

            backward pass for forward state LSTM
            backward pass for backward state LSTM
        4) update parameters
    end for
end for
```

Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional LSTM-CRF models for sequence tagging. CoRR, abs/1508.01991, 2015.

Results

		POS	CoNLL2000	CoNLL2003
Random	Conv-CRF (Collobert et al., 2011)	96.37	90.33	81.47
	LSTM	97.10	92.88	79.82
	BI-LSTM	97.30	93.64	81.11
	CRF	97.30	93.69	83.02
	LSTM-CRF	**97.45**	93.80	84.10
	BI-LSTM-CRF	97.43	**94.13**	**84.26**
Senna	Conv-CRF (Collobert et al., 2011)	97.29	94.32	88.67 (89.59)
	LSTM	97.29	92.99	83.74
	BI-LSTM	97.40	93.92	85.17
	CRF	97.45	93.83	86.13
	LSTM-CRF	97.54	94.27	88.36
	BI-LSTM-CRF	**97.55**	**94.46**	**88.83 (90.10)**

Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional LSTM-CRF models for sequence tagging. CoRR, abs/1508.01991, 2015.

BI-LSTM-CRF for SRL

- End-to-end tagging model
  - 8 layer bi-directional LSTM
  - No parsing features
- Features
  - Argument
  - Predicate
  - Predicate-context
  - Region-mark
- Achieving new SOTA

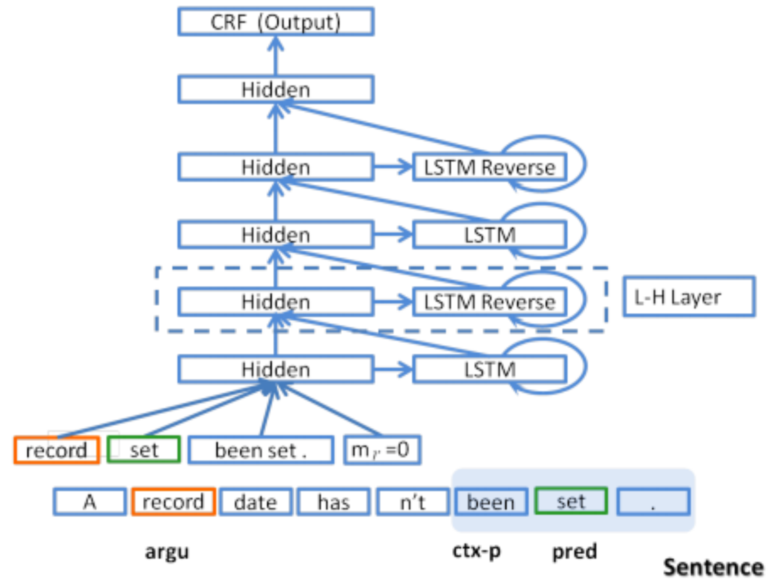
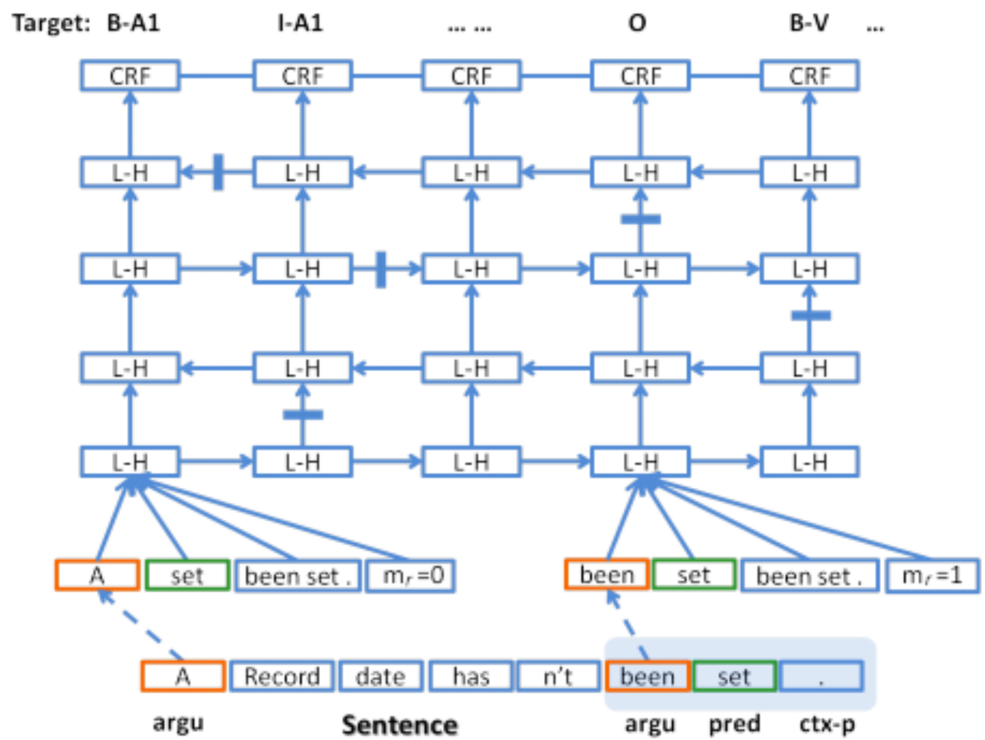


Figure 2: DB-LSTM network.Shadow part denote the predicate context within length 1.

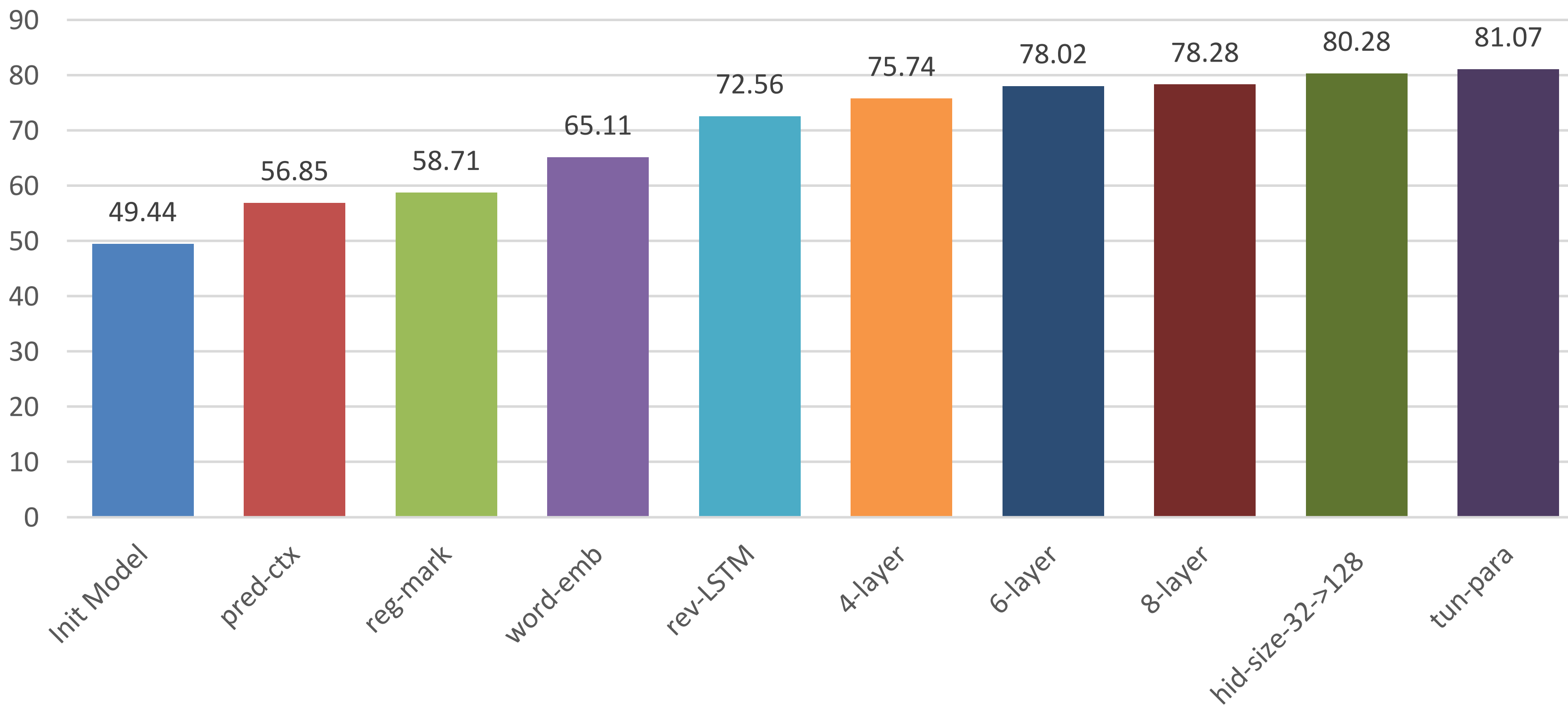
Jie Zhou and Wei Xu. (2015). End-to-end learning of semantic role labeling using recurrent neural networks. ACL.

Temporal Expanded



Jie Zhou and Wei Xu. (2015). End-to-end learning of semantic role labeling using recurrent neural networks. ACL.

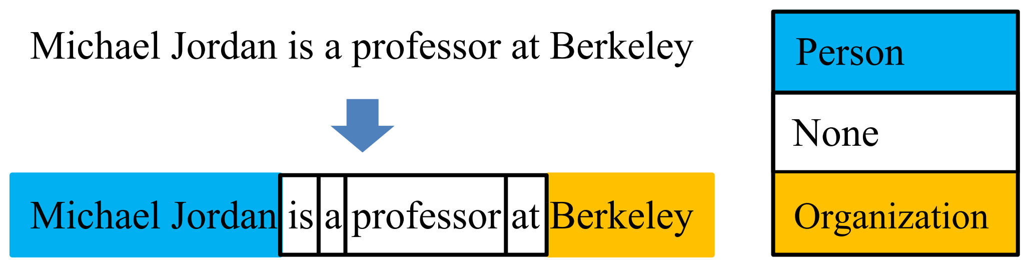
Results



Jie Zhou and Wei Xu. (2015). End-to-end learning of semantic role labeling using recurrent neural networks. ACL.

Segmentation Models

- Tagging models cannot extract segment information
  - E.g. the length of a segment
- Some tagging problems can be naturally modeled into segmentation task
  - E.g. word segmentation, named entity recognition



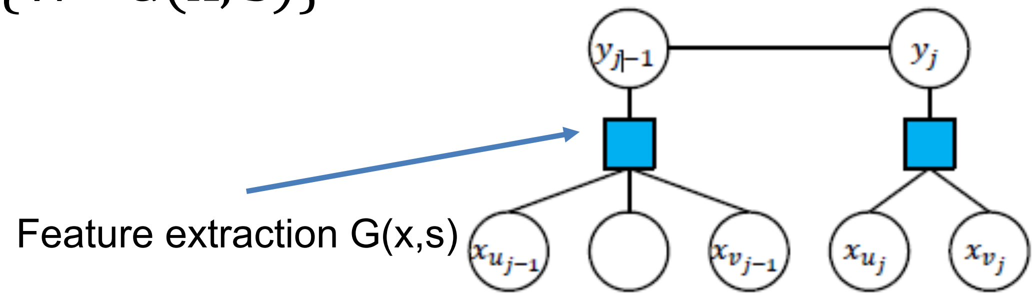
浦东开发与建设 → 浦东 / 开发 / 与 / 建设

Pudong development and construction

Yijia Liu, Wanxiang Che, Jiang Guo, Bing Qin, Ting Liu. (2016). Exploring Segment Representations for Neural Segmentation Models. IJCAI.

Semi-CRF

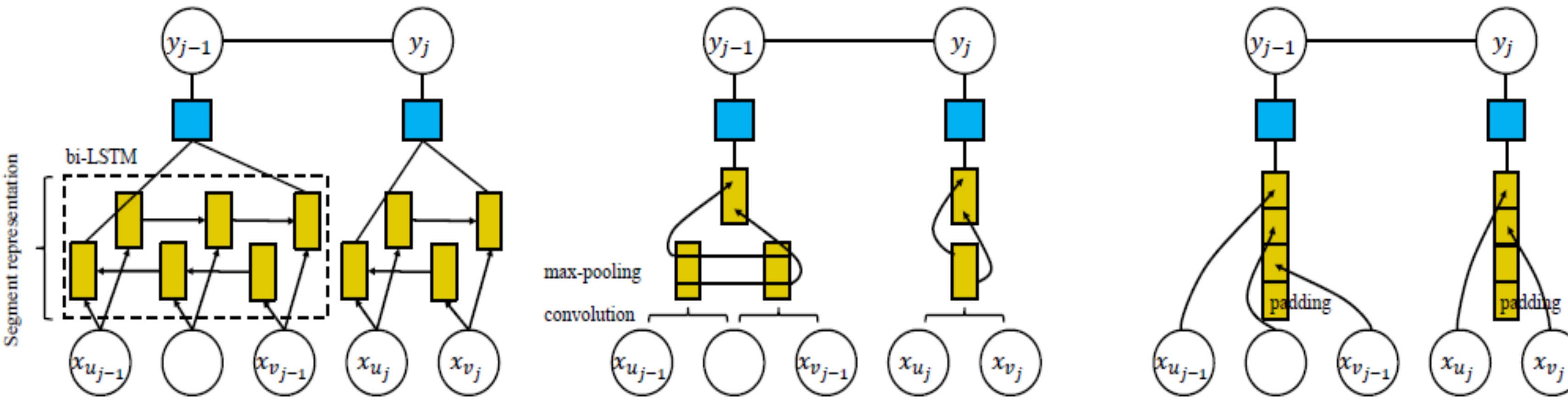
- A solution
  - Semi-Markov CRF [Sarawagi and Cohen, 2004]
  - Modeling segments directly
  - $p(\mathbf{s}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\{W \cdot G(\mathbf{x}, \mathbf{s})\}$



Can we represent segments with vectors?

Yijia Liu, Wanxiang Che, Jiang Guo, Bing Qin, Ting Liu. (2016). Exploring Segment Representations for Neural Segmentation Models. IJCAI.

Compositional Segment Representation



(b) SRNN (c) SCNN (d) SCONCATE

Yijia Liu, Wanxiang Che, Jiang Guo, Bing Qin, Ting Liu. (2016). Exploring Segment Representations for Neural Segmentation Models. IJCAI.

Decoding Algorithm

Input: a sequence $X = (x_0, \dots, x_{n-1})$ of n units, the maximum length of the segment L
Output: the highest scored segmentation $S = (s_0, \dots, s_{m-1})$, where $s = (u, v, y)$ is a segment and u represents the starting position, v represents the ending position, and an optional tag y associate with the segment.
Defining $V(i, y)$ which represents the best sub-segmentation that ends with x_i (not included) and $V(i, y)$ can be calculated as:

$$V(i,y) = \begin{cases} \max\limits_{y', d=1\dots L} V(i-d, y') + score(i-d, i, y), if\ i > 0 \\ 0, if\ i = 0 \\ -\infty, if\ i < 0 \end{cases}$$

for $i \leftarrow 1 \dots n$
    for $y \in \mathcal{Y}$:
        for $d \leftarrow 1 \dots L$
        if $i - d = 0$:
            $V(i, y) \leftarrow score(i-d, i, y)$
        else:
            $best_{i-d} \leftarrow \max\limits_{y'} V(i-d, y')$
            $V(i, y) \leftarrow \max\big(V(i, y), best_{i-d} + score(i-d, i, y)\big)$

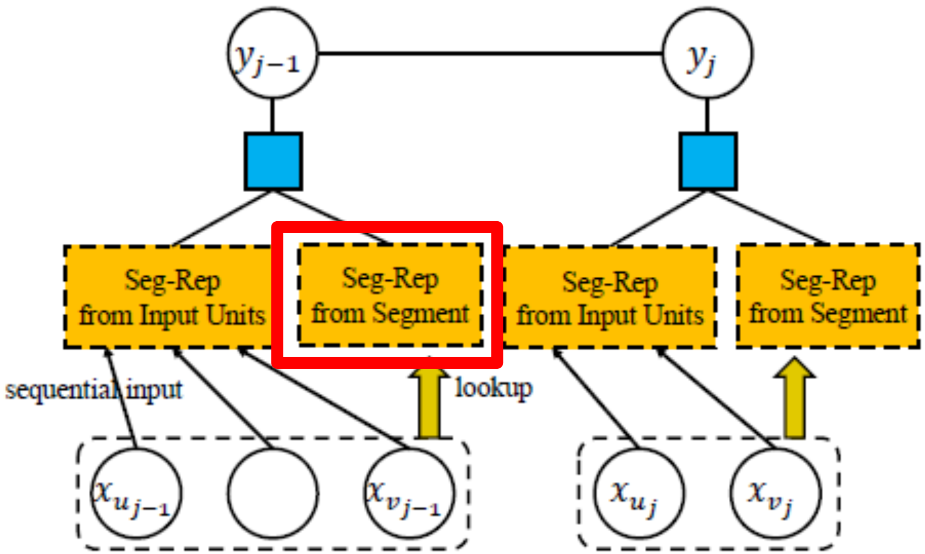
Yijia Liu, Wanxiang Che, Jiang Guo, Bing Qin, Ting Liu. (2016). Exploring Segment Representations for Neural Segmentation Models. IJCAI.

Results

	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

Yijia Liu, Wanxiang Che, Jiang Guo, Bing Qin, Ting Liu. (2016). Exploring Segment Representations for Neural Segmentation Models. IJCAI.

Segment-level Representation



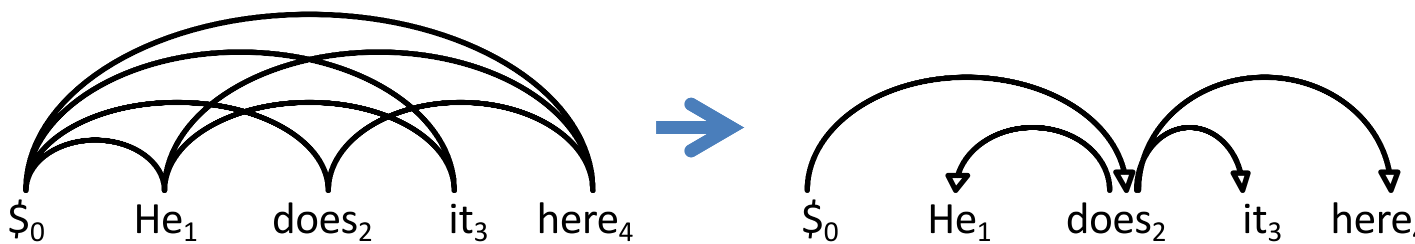
model	CoNLL03	CTB6	PKU	MSR
NN-LABELER	88.62	93.06	92.99	93.79
NN-CRF	89.08	93.65	93.28	94.17
SPARSE-CRF	83.43	95.08	95.06	96.54
SRNN	88.63	94.06	93.91	95.21
+SEMB-HETERO	89.59	**95.48**	95.60	97.39
	+0.96	*+1.42*	*+1.69*	*+2.18*
SCONCATE	89.07	93.96	93.57	94.53
+SEMB-HETERO	**89.77**	95.42	**95.67**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

Yijia Liu, Wanxiang Che, Jiang Guo, Bing Qin, Ting Liu. (2016). Exploring Segment Representations for Neural Segmentation Models. IJCAI.

Part 4.2: Dynamic Programming Decoding for Parsing

Graph-based Dependency Parsing

- Find the highest scoring tree from a complete graph
- Dynamic Programming Decoding
  - E.g. Eisner Algorithm



$$Y^* = \arg\max_{Y \in \Phi(X)} score(X, Y)$$

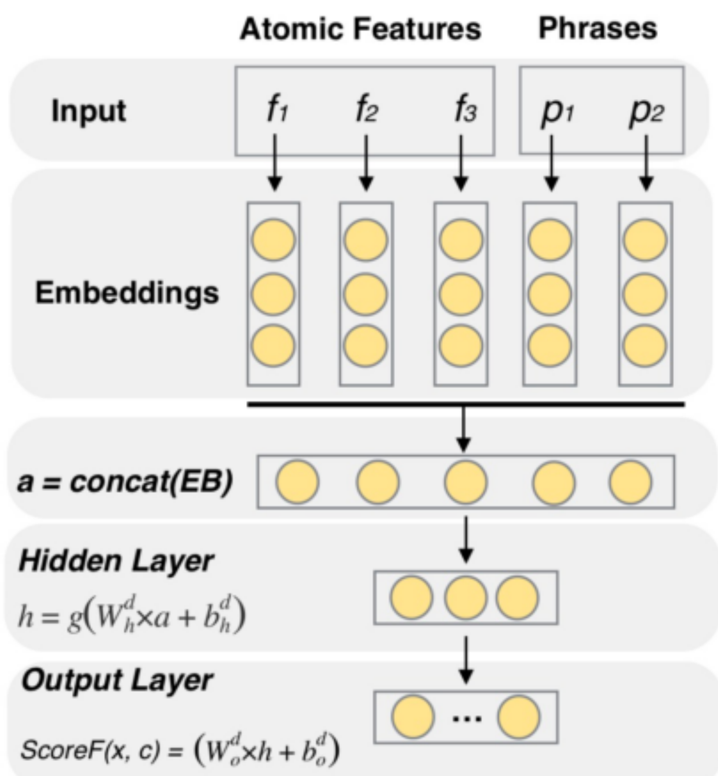
How to Score an Arc?

$$score(6,1) = \mathbf{w} \cdot \mathbf{f}(6,1)$$

* As McGwire neared , fans went wild

[went]	[VBD]	[As]	[ADP]	[went]
[VERB]	[As]	[IN]	[went, VBD]	[As, ADP]
[went, As]	[VBD, ADP]	[went, VERB]	[As, IN]	[went, As]
[VERB, IN]	[VBD, As, ADP]	[went, As, ADP]	[went, VBD, ADP]	[went, VBD, As]
[ADJ, *, ADP]	[VBD, *, ADP]	[VBD, ADJ, ADP]	[VBD, ADJ, *]	[NNS, *, ADP]
[NNS, VBD, ADP]	[NNS, VBD, *]	[ADJ, ADP, NNP]	[VBD, ADP, NNP]	[VBD, ADJ, NNP]
[NNS, ADP, NNP]	[NNS, VBD, NNP]	[went, left, 5]	[VBD, left, 5]	[As, left, 5]
[ADP, left, 5]	[VERB, As, IN]	[went, As, IN]	[went, VERB, IN]	[went, VERB, As]
[JJ, *, IN]	[VERB, *, IN]	[VERB, JJ, IN]	[VERB, JJ, *]	[NOUN, *, IN]
[NOUN, VERB, IN]	[NOUN, VERB, *]	[JJ, IN, NOUN]	[VERB, IN, NOUN]	[VERB, JJ, NOUN]
[NOUN, IN, NOUN]	[NOUN, VERB, NOUN]	[went, left, 5]	[VERB, left, 5]	[As, left, 5]
[IN, left, 5]	[went, VBD, As, ADP]	[VBD, ADJ, *, ADP]	[NNS, VBD, *, ADP]	[VBD, ADJ, ADP, NNP]
[NNS, VBD, ADP, NNP]	[went, VBD, left, 5]	[As, ADP, left, 5]	[went, As, left, 5]	[VBD, ADP, left, 5]
[went, VERB, As, IN]	[VERB, JJ, *, IN]	[NOUN, VERB, *, IN]	[VERB, JJ, IN, NOUN]	[NOUN, VERB, IN, NOUN]
[went, VERB, left, 5]	[As, IN, left, 5]	[went, As, left, 5]	[VERB, IN, left, 5]	[VBD, As, ADP, left, 5]
[went, As, ADP, left, 5]	[went, VBD, ADP, left, 5]	[went, VBD, As, left, 5]	[ADJ, *, ADP, left, 5]	[VBD, *, ADP, left, 5]
[VBD, ADJ, ADP, left, 5]	[VBD, ADJ, *, left, 5]	[NNS, *, ADP, left, 5]	[NNS, VBD, ADP, left, 5]	[NNS, VBD, *, left, 5]
[ADJ, ADP, NNP, left, 5]	[VBD, ADP, NNP, left, 5]	[VBD, ADJ, NNP, left, 5]	[NNS, ADP, NNP, left, 5]	[NNS, VBD, NNP, left, 5]
[VERB, As, IN, left, 5]	[went, As, IN, left, 5]	[went, VERB, IN, left, 5]	[went, VERB, As, left, 5]	[JJ, *, IN, left, 5]
[VERB, *, IN, left, 5]	[VERB, JJ, IN, left, 5]	[VERB, JJ, *, left, 5]	[NOUN, *, IN, left, 5]	[NOUN, VERB, IN, left, 5]

NN for Graph-based Parsing



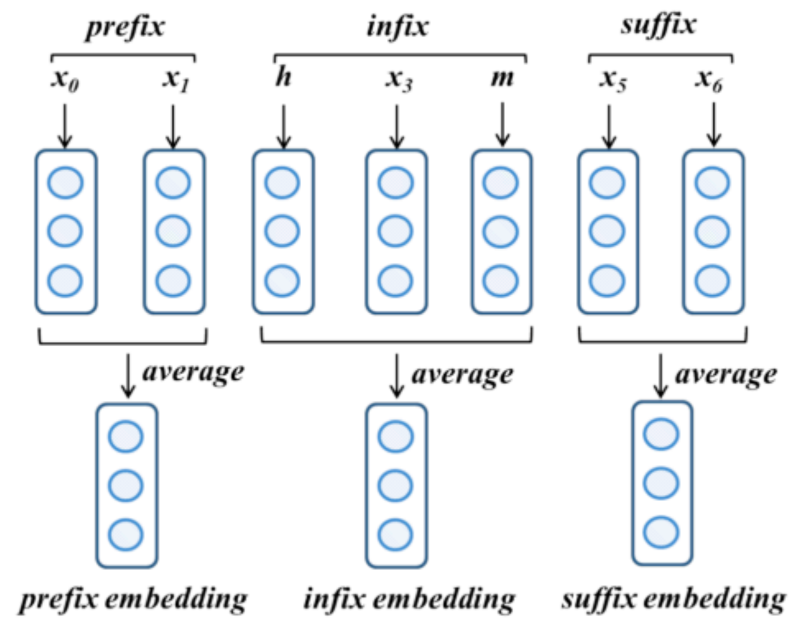


Figure 3: Illustration for phrase embeddings. h, m and x_0 to x_6 are words in the sentence.

Pei, W., Ge, T., & Chang, B. (2015). An Effective Neural Network Model for Graph-based Dependency Parsing. ACL.

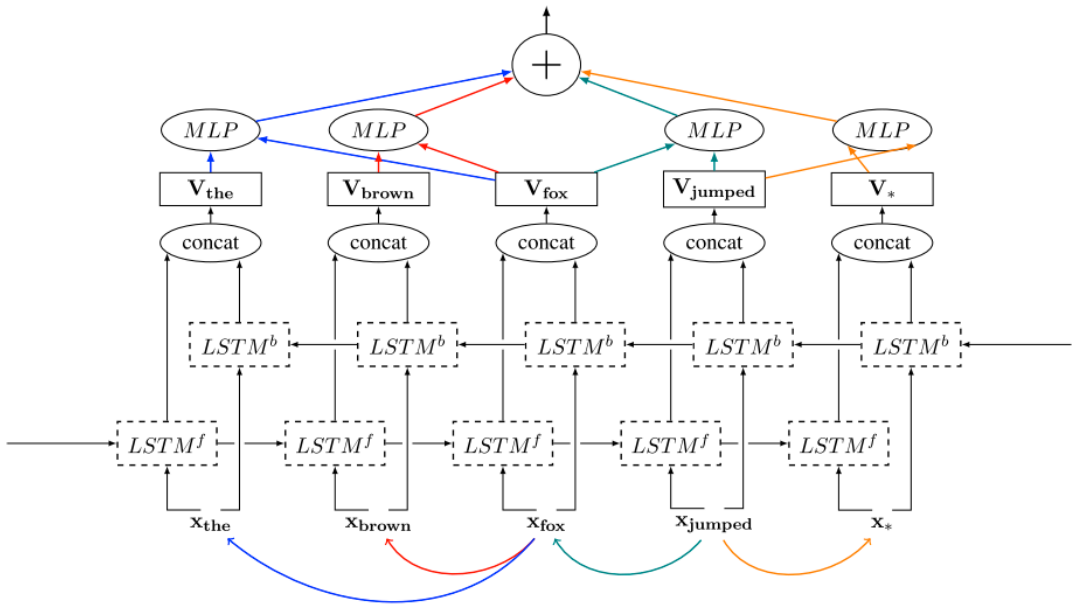
Results

	Models	Dev		Test		Speed (sent/s)
		UAS	LAS	UAS	LAS	
First-order	MSTParser-1-order	92.01	90.77	91.60	90.39	20
	1-order-atomic-rand	92.00	90.71	91.62	90.41	**55**
	1-order-atomic	92.19	90.94	92.14	90.92	**55**
	1-order-phrase-rand	92.47	91.19	92.25	91.05	26
	1-order-phrase	**92.82**	**91.48**	**92.59**	**91.37**	26
Second-order	MSTParser-2-order	92.70	91.48	92.30	91.06	14
	2-order-phrase-rand	93.39	92.10	92.99	91.79	10
	2-order-phrase	**93.57**	**92.29**	**93.29**	**92.13**	10
Third-order	(Koo and Collins, 2010)	93.49	N/A	93.04	N/A	N/A

Pei, W., Ge, T., & Chang, B. (2015). An Effective Neural Network Model for Graph-based Dependency Parsing. ACL.

BI-LSTM for Graph-based Parsing-I

- Each dependency arc in a sentence is scored using MLP that is fed the BI-LSMT encoding of the words at the arc's end points



Kiperwasser, E., & Goldberg, Y. (2016). Simple and Accurate Dependency Parsing Using Bidirectional LSTM Feature Representations. TACL.

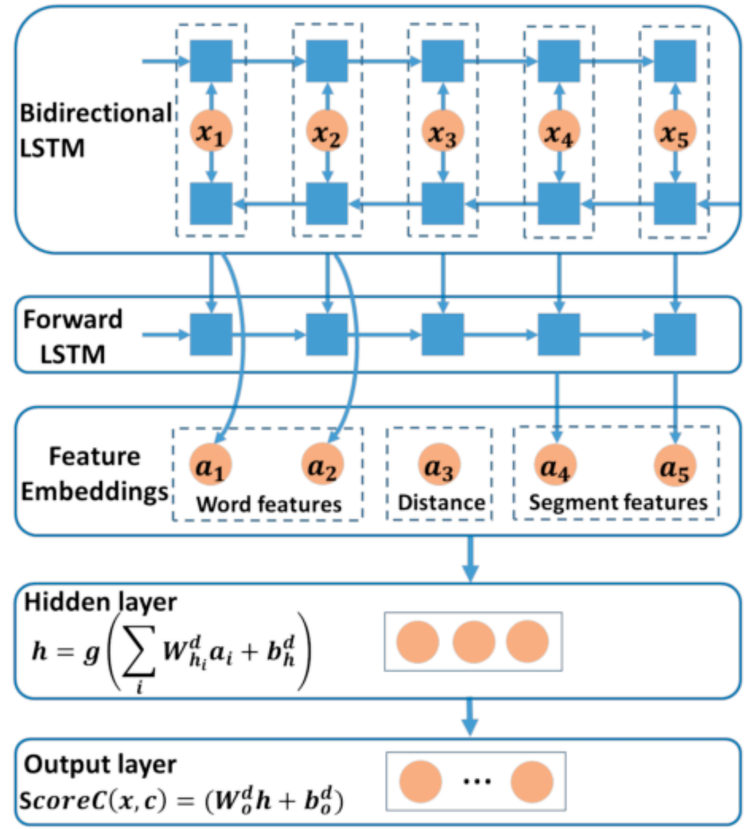
Results

System	Method	Representation	Emb	PTB-YM	PTB-SD		CTB	
				UAS	UAS	LAS	UAS	LAS
This work	graph, 1st order	2 BiLSTM vectors	–	–	93.1	91.0	**86.6**	**85.1**
This work	transition (greedy, dyn-oracle)	4 BiLSTM vectors	–	–	93.1	91.0	86.2	85.0
This work	transition (greedy, dyn-oracle)	11 BiLSTM vectors	–	–	**93.2**	**91.2**	86.5	84.9
ZhangNivre11	transition (beam)	large feature set (sparse)	–	92.9	–	–	86.0	84.4
Martins13 (TurboParser)	graph, 3rd order+	large feature set (sparse)	–	92.8	93.1	–	–	–
Pei15	graph, 2nd order	large feature set (dense)	–	93.0	–	–	–	–
Dyer15	transition (greedy)	Stack-LSTM + composition	–	–	92.4	90.0	85.7	84.1
Ballesteros16	transition (greedy, dyn-oracle)	Stack-LSTM + composition	–	–	92.7	90.6	86.1	84.5
This work	graph, 1st order	2 BiLSTM vectors	YES	–	93.0	90.9	86.5	84.9
This work	transition (greedy, dyn-oracle)	4 BiLSTM vectors	YES	–	93.6	91.5	87.4	85.9
This work	transition (greedy, dyn-oracle)	11 BiLSTM vectors	YES	–	93.9	91.9	**87.6**	86.1
Weiss15	transition (greedy)	large feature set (dense)	YES	–	93.2	91.2	–	–
Weiss15	transition (beam)	large feature set (dense)	YES	–	**94.0**	**92.0**	–	–
Pei15	graph, 2nd order	large feature set (dense)	YES	93.3	–	–	–	–
Dyer15	transition (greedy)	Stack-LSTM + composition	YES	–	93.1	90.9	87.1	85.5
Ballesteros16	transition (greedy, dyn-oracle)	Stack-LSTM + composition	YES	–	93.6	91.4	**87.6**	**86.2**
LeZuidema14	reranking /blend	inside-outside recursive net	YES	93.1	93.8	91.5	–	–
Zhu15	reranking /blend	recursive conv-net	YES	93.8	–	–	85.7	–

Kiperwasser, E., & Goldberg, Y. (2016). Simple and Accurate Dependency Parsing Using Bidirectional LSTM Feature Representations. TACL.

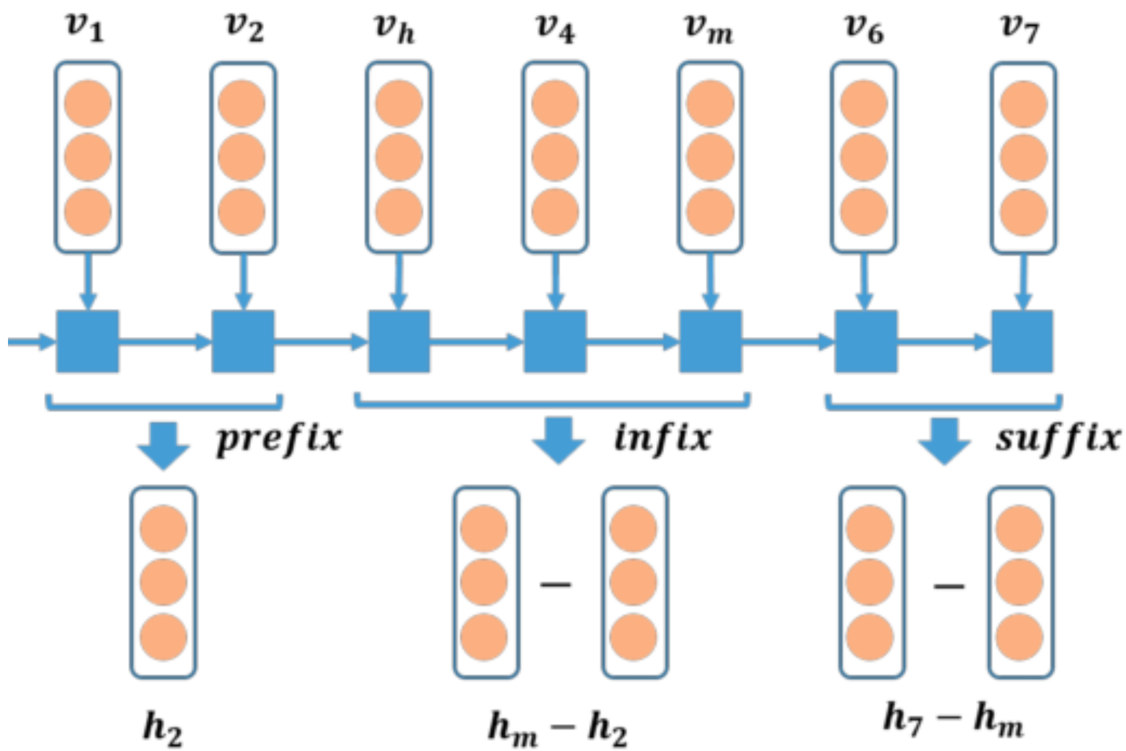
BI-LSTM for Graph-based Parsing-II

- Besides the word vectors, they used sentence segment (phrase) embeddings



Wang, W., & Chang, B. (2016). Graph-based Dependency Parsing with Bidirectional LSTM. ACL.

Learning Segment Embeddings



Wang, W., & Chang, B. (2016). Graph-based Dependency Parsing with Bidirectional LSTM. ACL.

Results

	Models	UAS	LAS	Speed(sent/s)
First-order	MSTParser	91.60	90.39	20
	1st-order atomic (Pei et al., 2015)	92.14	90.92	55
	1st-order phrase (Pei et al., 2015)	92.59	91.37	26
	Our basic model	93.09	92.03	**61**
	Our basic model + segment	**93.51**	**92.45**	26
Second-order	MSTParser	92.30	91.06	14
	2nd-order phrase (Pei et al., 2015)	93.29	92.13	10
Third-order	(Koo and Collins, 2010)	93.04	N/A	N/A
Fourth-order	(Ma and Zhao, 2012)	93.4	N/A	N/A
Unlimited-order	(Zhang and McDonald, 2012)	93.06	91.86	N/A
	(Zhang et al., 2013)	93.50	92.41	N/A
	(Zhang and McDonald, 2014)	**93.57**	**92.48**	N/A

Wang, W., & Chang, B. (2016). Graph-based Dependency Parsing with Bidirectional LSTM. ACL.

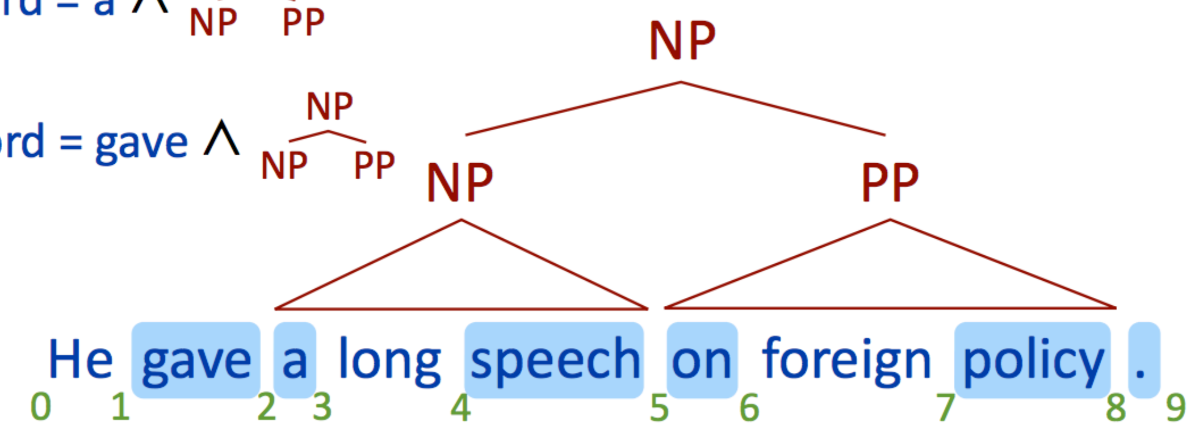
Neural CRF for Constituency Parsing

- CRF Parsing with CKY decoding

$$P(T|x) \propto \prod_{r \in T} \exp\left(\mathrm{score}(r)\right) \qquad \mathrm{score}\left(\overset{\mathrm{NP}}{{}_2\,\mathrm{NP}\,{}_5\,\mathrm{PP}\,{}_8}\right) = w^\top f\left(\overset{\mathrm{NP}}{{}_2\,\mathrm{NP}\,{}_5\,\mathrm{PP}\,{}_8}\right)$$

FirstWord = a $\wedge$ $\overset{\mathrm{NP}}{\mathrm{NP}\quad\mathrm{PP}}$

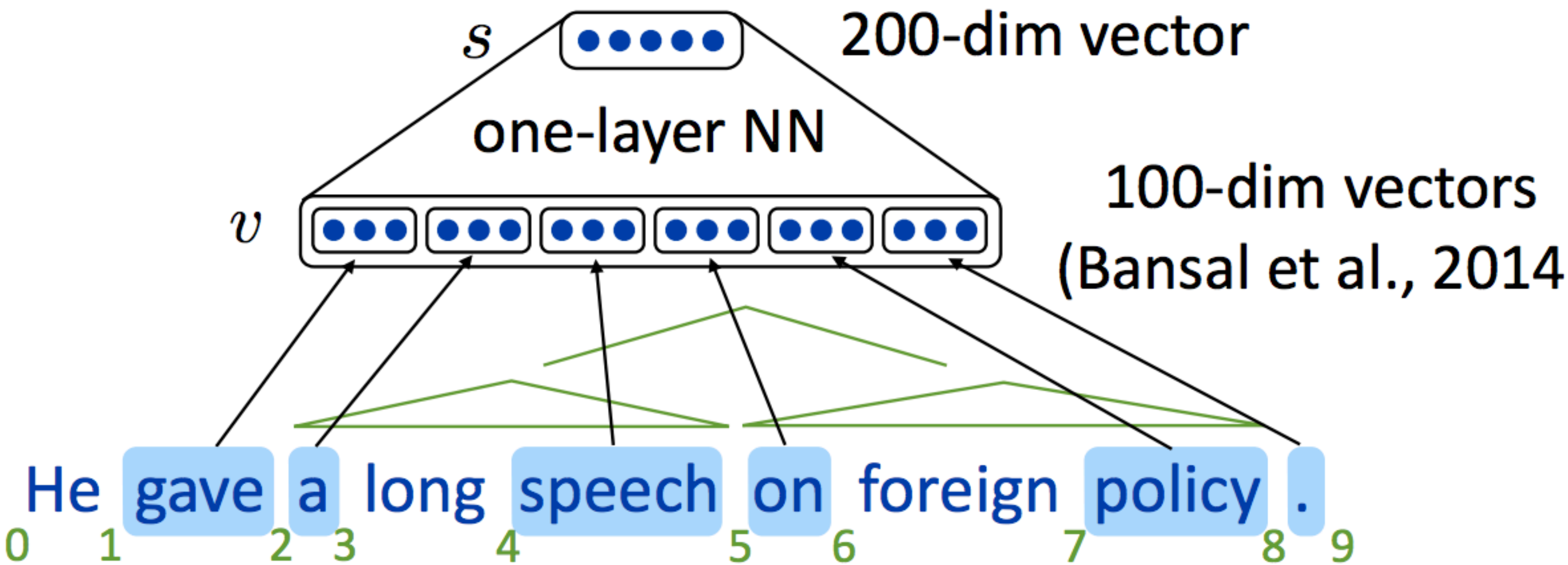
PrevWord = gave $\wedge$ $\overset{\mathrm{NP}}{\mathrm{NP}\quad\mathrm{PP}}$



Durrett, G., & Klein, D. (2015). Neural CRF Parsing. ACL.

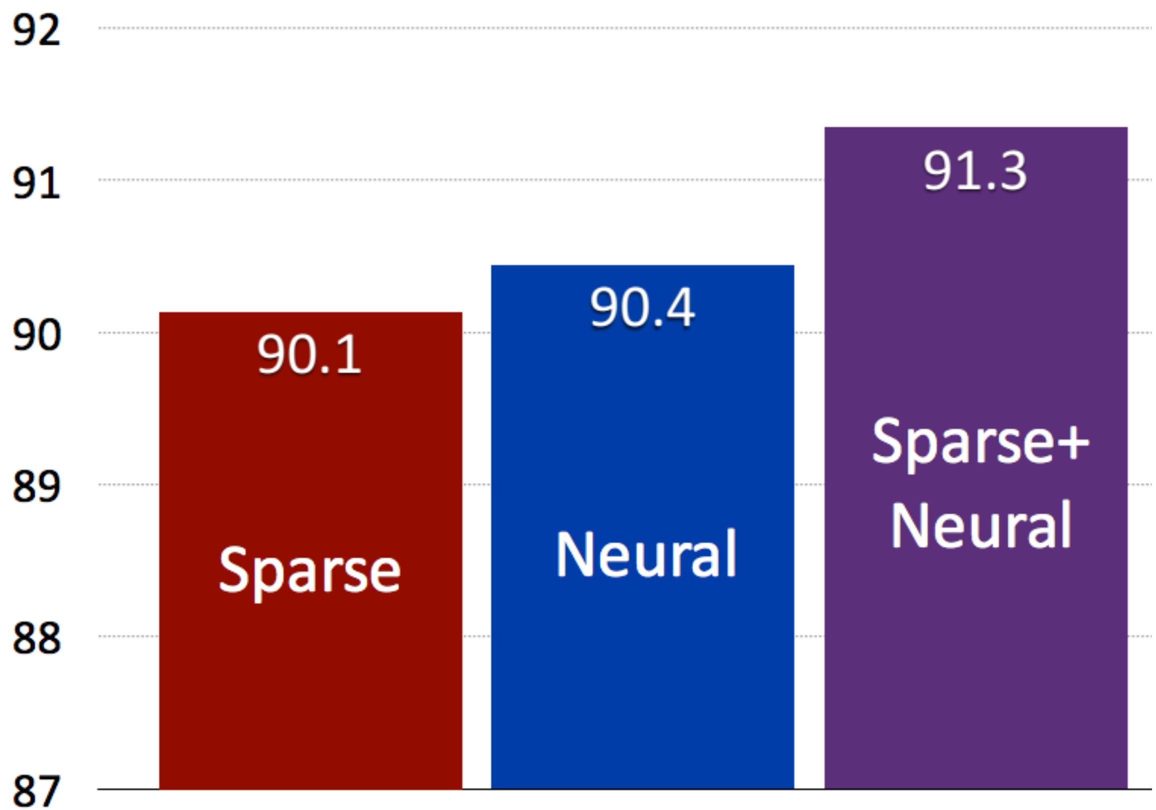
Neural CRF for Constituency Parsing

- Neural CRF Parsing



Durrett, G., & Klein, D. (2015). Neural CRF Parsing. ACL.

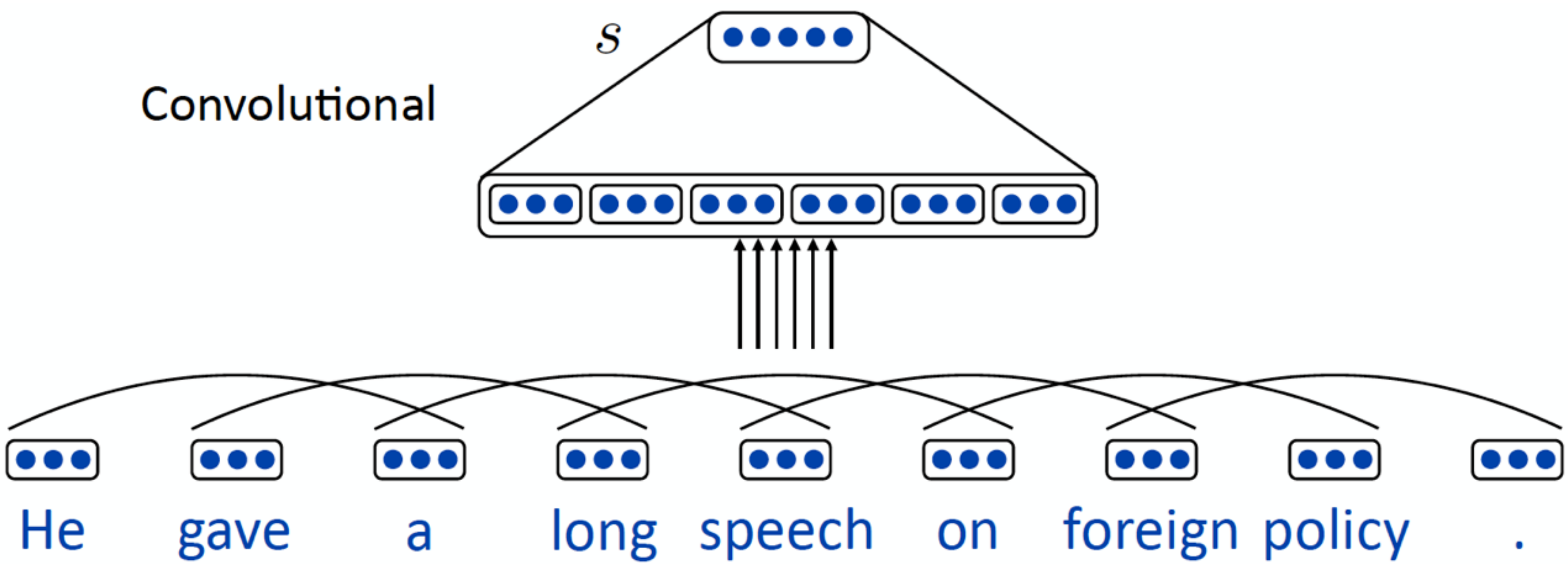
Results



Durrett, G., & Klein, D. (2015). Neural CRF Parsing. ACL.

Neural CRF for Constituency Parsing

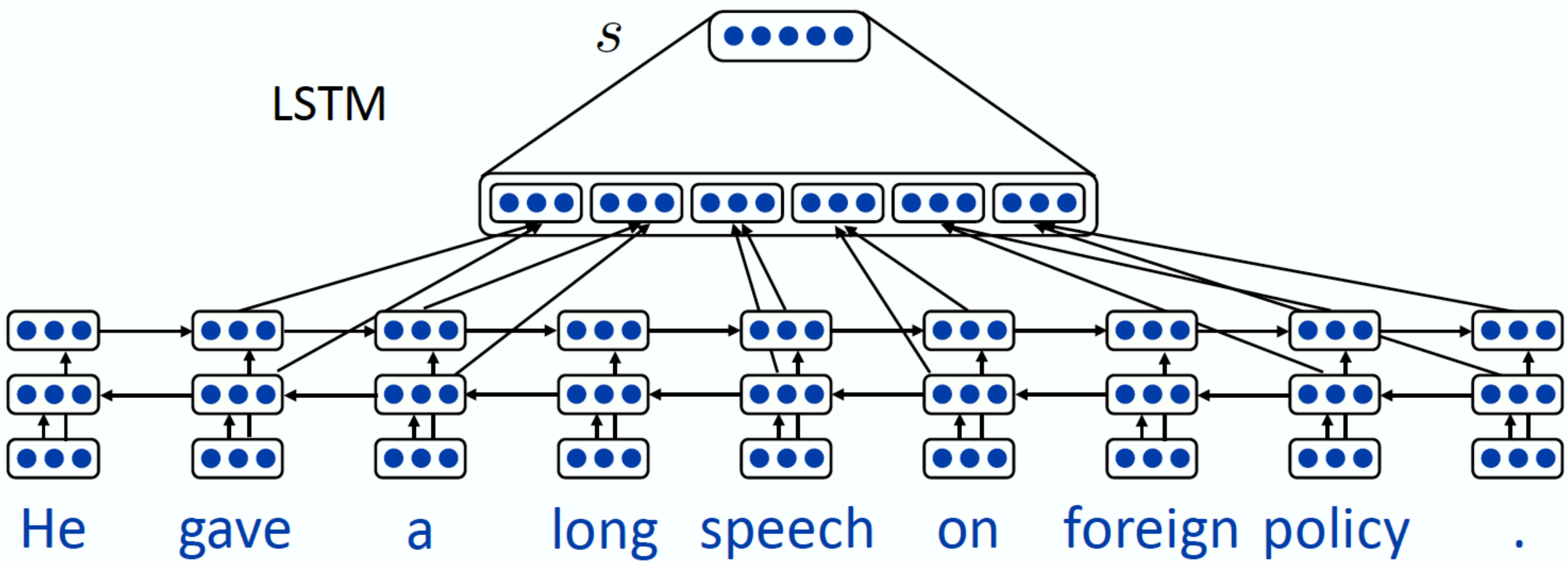
- More neural networks



Durrett, G., & Klein, D. (2015). Neural CRF Parsing. ACL.

Neural CRF for Constituency Parsing

- More neural networks



Durrett, G., & Klein, D. (2015). Neural CRF Parsing. ACL.

Conclusion

- Neural nets can provide continuous features in discrete structured models
- Inference and learning are almost unchanged from the purely discrete model