Encoder-Decoder Model in Dependency Parsing

编码解码模型在依存句法分析中的应用

Tao Ji (纪焘)

November 24, 2017

taoji.cs@gmail.com
AntNLP Group,
Department of Computer Science and Technology,
East China Normal University

Table of contents

Dependency Parsing

Dependency Parsing

Dependency Syntax: Syntactic structure consists of lexical items, linked by binary asymmetric relations called dependencies. [Tesnière, 1959]

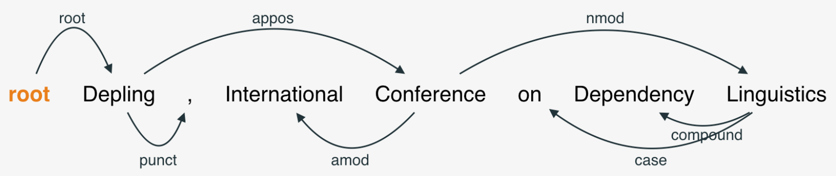


Figure 1: Dependency Tree

Intuitions Connectedness Acyclicity Single-Head

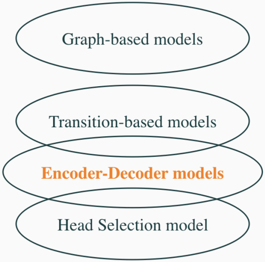
Dependency Parsing

Dependency parsing problem

- Input: Sentence $x = w_0, w_1, \cdots, w_n$ with $w_0 =$ root
- Output: Dependency Tree $T = (V, A)$ for x where:
  - $V = 0, 1, \cdots, n$ is the vertex set,
  - A is the arc set, i.e., $(i, j, k) \in A$ represents a dependency from w_i to w_j with label $l_k \in L$

Two main approaches

- ~~Grammar-based parsing~~
- Data-driven parsing
  - Graph-based models
  - Transition-based models
  - Head Selection model



References

Transition-based models

- **[CL08]** Algorithms for Deterministic Incremental Dependency Parsing. (Joakim Nivre)

Head Selection model

- **[EACL17]** Dependency Parsing as Head Selection. (Lapata et al.)

Transition-based Models

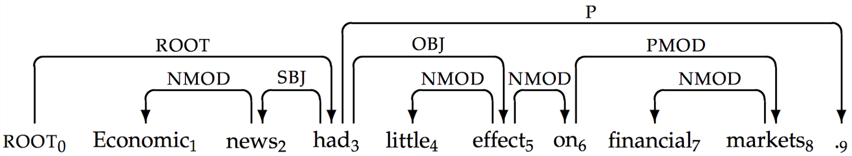


Figure 2: A dependency tree from the Penn Treebank. [Nivre, 2008]

Transition-based Models

Transition	Configuration			
	($[0]$,	$[1,\ldots,9]$,	$\emptyset$	)
$\text{SHIFT} \Longrightarrow$	($[0,1]$,	$[2,\ldots,9]$,	$\emptyset$	)
$\text{LEFT-ARC}_{\text{NMOD}} \Longrightarrow$	($[0]$,	$[2,\ldots,9]$,	$A_1 = \{(2, \text{NMOD}, 1)\}$	)
$\text{SHIFT} \Longrightarrow$	($[0,2]$,	$[3,\ldots,9]$,	A_1	)
$\text{LEFT-ARC}_{\text{SBJ}} \Longrightarrow$	($[0]$,	$[3,\ldots,9]$,	$A_2 = A_1 \cup \{(3, \text{SBJ}, 2)\}$	)
$\text{SHIFT} \Longrightarrow$	($[0,3]$,	$[4,\ldots,9]$,	A_2	)
$\text{SHIFT} \Longrightarrow$	($[0,3,4]$,	$[5,\ldots,9]$,	A_2	)
$\text{LEFT-ARC}_{\text{NMOD}} \Longrightarrow$	($[0,3]$,	$[5,\ldots,9]$,	$A_3 = A_2 \cup \{(5, \text{NMOD}, 4)\}$	)
$\text{SHIFT} \Longrightarrow$	($[0,3,5]$,	$[6,\ldots,9]$,	A_3	)
$\text{SHIFT} \Longrightarrow$	($[0,\ldots 6]$,	$[7,8,9]$,	A_3	)
$\text{SHIFT} \Longrightarrow$	($[0,\ldots,7]$,	$[8,9]$,	A_3	)
$\text{LEFT-ARC}_{\text{NMOD}} \Longrightarrow$	($[0,\ldots 6]$,	$[8,9]$,	$A_4 = A_3 \cup \{(8, \text{NMOD}, 7)\}$	)
$\text{RIGHT-ARC}^{s}_{\text{PMOD}} \Longrightarrow$	($[0,3,5]$,	$[6,9]$,	$A_5 = A_4 \cup \{(6, \text{PMOD}, 8)\}$	)
$\text{RIGHT-ARC}^{s}_{\text{NMOD}} \Longrightarrow$	($[0,3]$,	$[5,9]$,	$A_6 = A_5 \cup \{(5, \text{NMOD}, 6)\}$	)
$\text{RIGHT-ARC}^{s}_{\text{OBJ}} \Longrightarrow$	($[0]$,	$[3,9]$,	$A_7 = A_6 \cup \{(3, \text{OBJ}, 5)\}$	)
$\text{SHIFT} \Longrightarrow$	($[0,3]$,	$[9]$,	A_7	)
$\text{RIGHT-ARC}^{s}_{\text{P}} \Longrightarrow$	($[0]$,	$[3]$,	$A_8 = A_7 \cup \{(3, \text{P}, 9)\}$	)
$\text{RIGHT-ARC}^{s}_{\text{ROOT}} \Longrightarrow$	($[\,]$,	$[0]$,	$A_9 = A_8 \cup \{(0, \text{ROOT}, 3)\}$	)
$\text{SHIFT} \Longrightarrow$	($[0]$,	$[\,]$,	A_9	)

Figure 2: Arc-standard transition system. [Nivre, 2008]

Transition-based parsing problem

- **Input:** Sentence $x = w_0, w_1, \cdots, w_n$ with $w_0 =$root
- **Transition system:** Arc-standard, Arc-eager, Arc-hybrid, ...
- **Output:** Transition sequence $y = t_1, t_2, \cdots, t_m$ for x where:
  - $t_i \in T$, T is the transition set.
  - $2n \leq m \leq 3n$ (Arc-standard)

Transitions		**Preconditions**	
LEFT-ARC$_l$	$(\sigma\|i,j\|\beta,A) \Rightarrow (\sigma,j\|\beta,A\cup\{(j,l,i)\})$	LEFT-ARC$_l$	$\neg[i=0]$
RIGHT-ARC^s_l	$(\sigma\|i,j\|\beta,A) \Rightarrow (\sigma,i\|\beta,A\cup\{(i,l,j)\})$		$\neg\exists k\exists l'[(k,l',i)\in A]$
SHIFT	$(\sigma,i\|\beta,A) \Rightarrow (\sigma\|i,\beta,A)$	RIGHT-ARC^s_l	$\neg\exists k\exists l'[(k,l',j)\in A]$

Head Selection model

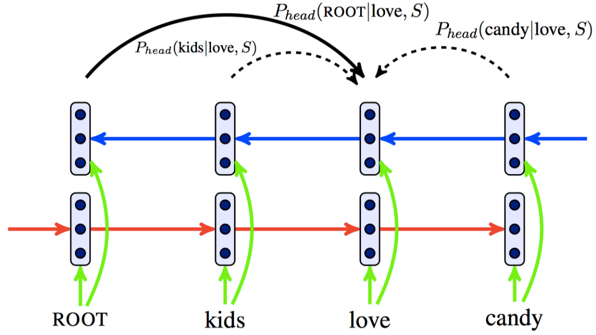


Figure 3: Head slection architecture. [Lapata et al., 2017]

$$P_{head}(w_j|w_i, S) = \frac{\exp(g(\boldsymbol{a}_j, \boldsymbol{a}_i))}{\sum_{k=0}^{N} \exp(g(\boldsymbol{a}_k, \boldsymbol{a}_i))} \tag{1}$$

$$g(\boldsymbol{a}_j, \boldsymbol{a}_i) = \boldsymbol{v}^{\top} \cdot \tanh(\boldsymbol{U} \cdot \boldsymbol{a}_j + \boldsymbol{W} \cdot \boldsymbol{a}_i) \tag{2}$$

Results

Datasets: Penn Treebank (PTB) with Stanford Dependencies

UAS/LAS: Unlabeled/Labeled Attachment Score

	Dev		Test	
Parser	UAS	LAS	UAS	LAS
Bohnet10	—	—	92.88	90.71
Martins13	—	—	92.89	90.55
Z&M14	—	—	93.22	91.02
Z&N11	—	—	93.00	90.95
C&M14	92.00	89.70	91.80	89.60
Dyer15	93.20	90.90	93.10	90.90
Weiss15	—	—	93.99	92.05
Andor16	—	—	**94.61**	**92.79**
K&G16 *graph*	—	—	93.10	91.00
K&G16 *trans*	—	—	93.90	91.90
DENSE-Pei	90.77	88.35	90.39	88.05
DENSE-Pei+E	91.39	88.94	91.00	88.61
DENSE	94.17	91.82	94.02	91.84
DENSE+E	**94.30**	**91.95**	94.10	91.90

Figure 4: Head slection results. [Lapata et al., 2017]

Encoder-Decoder Models

References

Seq2Seq + Attention

- **[NIPS15]** Grammar as a Foreign Language. (Vinyals et al.)
- **[ICLR15]** Neural Machine Translation by Jointly Learning to Align and Translate. (Bahdanau et al.)

Pointer Networks

- **[NIPS15]** Pointer Networks. (Vinyals et al.)

Seq2Tree

- **[ACL16]** Language to Logical Form with Neural Attention. (Dong and Lapata)

Seq2Seq + Attention

Paper: Grammar as a Foreign Language

Task: Syntactic constituency parsing

Model: Seq2seq model (NMT [Sutskever et al., 2014])

Results: 90.5 F1 scores on WSJ dataset.

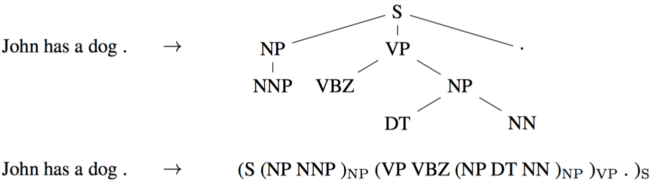


Figure 5: Example parsing task and its linearization. [Vinyals et al., 2015b]

Encoder/Decoder: {Deep-, Bi-} RNN, LSTM, GRU, ...

Classify:

$$P(\mathrm{Y}|\mathrm{X}) = \prod_{t=1}^{T_y} P(y_t|\mathrm{X}, y_{<t}) = \prod_{t=1}^{T_y} \mathrm{softmax}(\boldsymbol{W} \cdot s_t)[y_t]$$

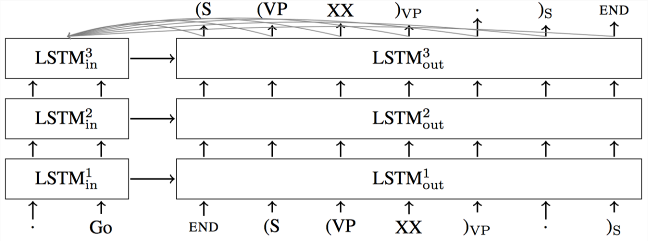


Figure 6: DeepLSTM+A seq2seq architecture. [Vinyals et al., 2015b]

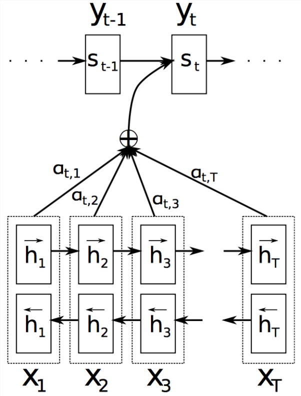


Figure 7: Attention architecture. [Bahdanau et al., 2014]

- Encoder vector:

$$h_i = [\vec{h}_i \circ \overleftarrow{h}_i]$$

- Attention score:

$$u_{t,i} = v^\top \cdot \tanh(\mathbf{W}'_1 \cdot h_i + \mathbf{W}'_2 \cdot s_{t-1})$$

- Attention weight:

$$a_{t,i} = \mathrm{softmax}(u_{t,i})$$

- Attention vector:

$$c_t = \sum_{i=1}^{T_x} a_{t,i} \cdot h_i$$

Seq2Seq + Attention

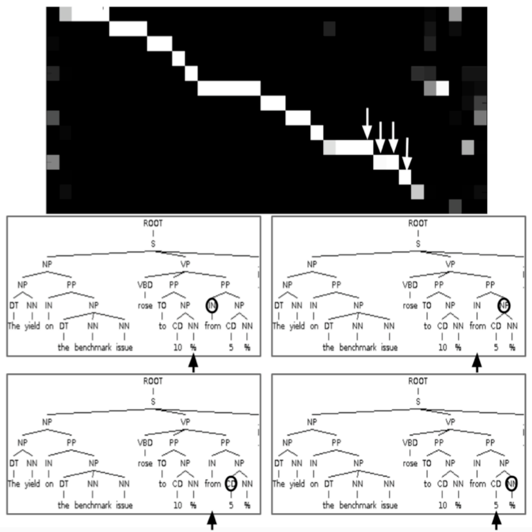


Figure 8: Attention matrix. [Vinyals et al., 2015b]

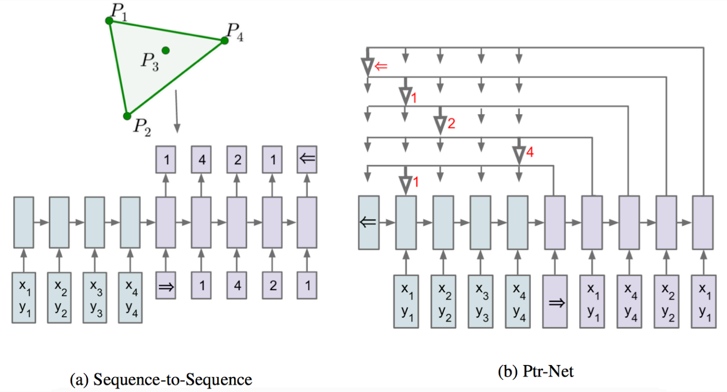


Figure 9: Pointer Network architecture. [Vinyals et al., 2015a]

Attention & Ptr-Net

- Attention score:

$$u_{t,i} = v^\top \cdot \tanh(\mathbf{W}'_1 \cdot h_i + \mathbf{W}'_2 \cdot s_{t-1})$$

- Attention weight:

$$a_{t,i} = \mathrm{softmax}(u_{t,i})$$

- Attention vector:

$$c_t = \sum_{i=1}^{T_x} a_{t,i} \cdot h_i$$

- Pointer score:

$$u_{t,i} = v^\top \cdot \tanh(\mathbf{W}'_1 \cdot h_i + \mathbf{W}'_2 \cdot s_{t-1})$$

- Pointer probability:

$$p_{t,i} = \mathrm{softmax}(u_{t,i})$$

Seq2Tree

Paper: Language to Logical Form with Neural Attention

Task: Semantic parsing

Model: Sequence-to-sequence/tree model

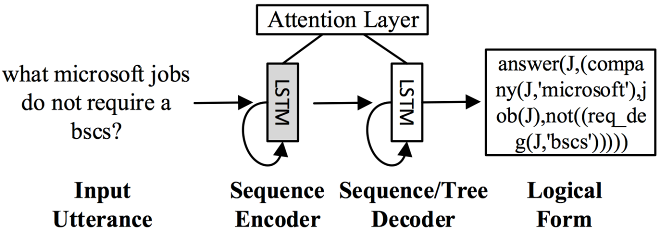


Figure 10: Semantic parsing architecture. [Dong and Lapata, 2016]

Seq2Tree

Paper: Language to Logical Form with Neural Attention

Task: Semantic parsing

Model: Sequence-to-sequence/tree model

Dataset	Length	Example
JOBS	9.80	*what microsoft jobs do not require a bscs?*
	22.90	answer(company(J,’microsoft’),job(J),not((req_deg(J,’bscs’))))
GEO	7.60	*what is the population of the state with the largest area?*
	19.10	(population:i (argmax $0 (state:t $0) (area:i $0)))
ATIS	11.10	*dallas to san francisco leaving after 4 in the afternoon please*
	28.10	(lambda $0 e (and (>(departure_time $0) 1600:ti) (from $0 dallas:ci) (to $0 san_francisco:ci)))
IFTTT	6.95	*Turn on heater when temperature drops below 58 degree*
	21.80	TRIGGER: Weather - Current_temperature_drops_below - ((Temperature (58)) (Degrees_in (f))) ACTION: WeMo_Insight_Switch - Turn_on - ((Which_switch? (””)))

Figure 10: Examples of datasets. [Dong and Lapata, 2016]

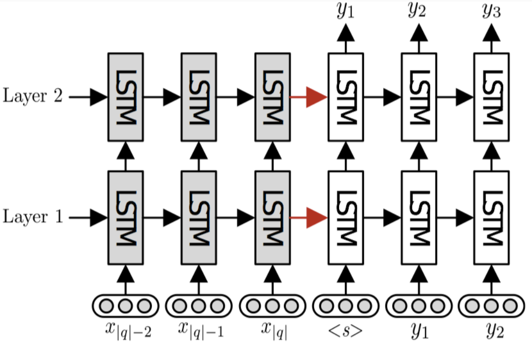


Figure 11: Seq2Seq semantic parsing. [Dong and Lapata, 2016]

Seq2Seq vs Seq2Tree

"lambda \$0 e (and (>(departure time \$0) 1600:ti) (from \$0 dallas:ci))"

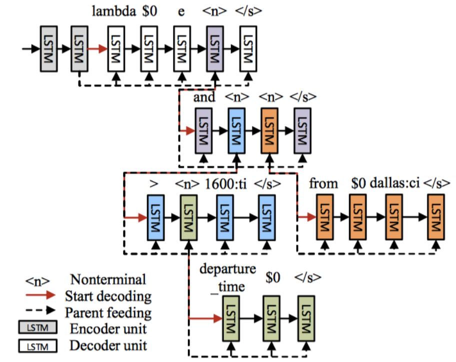


Figure 11: Seq2Tree semantic parsing. [Dong and Lapata, 2016]

Seq2Tree

$$\boldsymbol{h}_t^{att} = \tanh(\boldsymbol{W}_1 \boldsymbol{h}_t + \boldsymbol{W}_2 \boldsymbol{c}_t)$$

$$p(y_t | y_{<t}, x) = softmax(\boldsymbol{W}_o \boldsymbol{h}_t^{att})^\top \boldsymbol{e}(y_t)$$

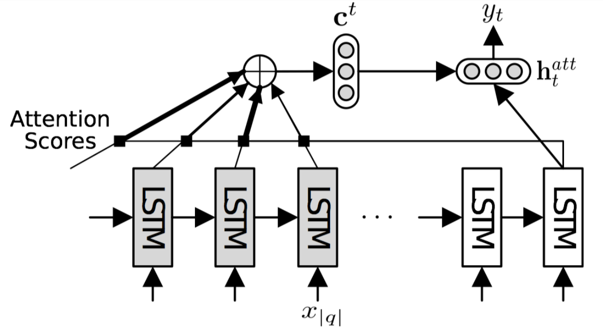


Figure 12: Classification. [Dong and Lapata, 2016]

Results

Method	Accuracy
COCKTAIL (Tang and Mooney, 2001)	79.4
PRECISE (Popescu et al., 2003)	88.0
ZC05 (Zettlemoyer and Collins, 2005)	79.3
DCS+L (Liang et al., 2013)	90.7
TISP (Zhao and Huang, 2015)	85.0
SEQ2SEQ	87.1
− attention	77.9
− argument	70.7
SEQ2TREE	90.0
− attention	83.6

Figure 13: Results on JOBS. [Dong and Lapata, 2016]

Results

Method	Accuracy
SCISSOR (Ge and Mooney, 2005)	72.3
KRISP (Kate and Mooney, 2006)	71.7
WASP (Wong and Mooney, 2006)	74.8
λ-WASP (Wong and Mooney, 2007)	86.6
LNLZ08 (Lu et al., 2008)	81.8
ZC05 (Zettlemoyer and Collins, 2005)	79.3
ZC07 (Zettlemoyer and Collins, 2007)	86.1
UBL (Kwiatkowski et al., 2010)	87.9
FUBL (Kwiatkowski et al., 2011)	88.6
KCAZ13 (Kwiatkowski et al., 2013)	89.0
DCS+L (Liang et al., 2013)	87.9
TISP (Zhao and Huang, 2015)	88.9
SEQ2SEQ	84.6
− attention	72.9
− argument	68.6
SEQ2TREE	87.1
− attention	76.8

Figure 14: Results on GEO. [Dong and Lapata, 2016]

Results

Method	Accuracy
ZC07 (Zettlemoyer and Collins, 2007)	84.6
UBL (Kwiatkowski et al., 2010)	71.4
FUBL (Kwiatkowski et al., 2011)	82.8
GUSP-FULL (Poon, 2013)	74.8
GUSP++ (Poon, 2013)	83.5
TISP (Zhao and Huang, 2015)	84.2
SEQ2SEQ	84.2
— attention	75.7
— argument	72.3
SEQ2TREE	84.6
— attention	77.5

Figure 15: Results on ATIS. [Dong and Lapata, 2016]

Conclusions

- Seq2Seq+Attention is very useful in many NLP tasks.
- Seq2Tree is better for some tree output tasks.
- We need Pointer Network at some tasks.

Encoder-Decoder Dependency Parsing

References

Encoder-Decoder Dependency Parsing

- **[IWPT17]** Encoder-Decoder Shift-Reduce Syntactic Parsing. (Liu and Zhang)
- **[EMNLP17]** Stack-based Multi-layer Attention for Transition-based Dependency Parsing. (Zhirui Zhang et al.)

Paper: Encoder-Decoder Shift-Reduce Syntactic Parsing

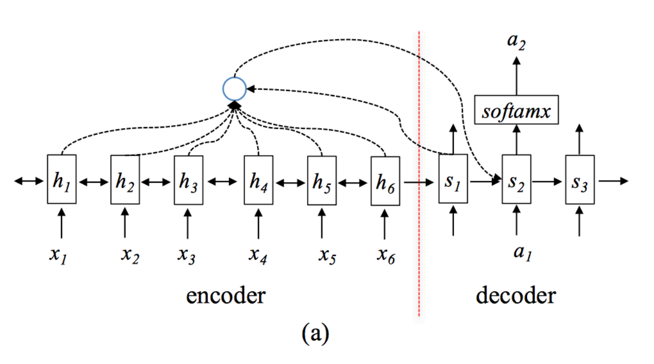


Figure 16: Vanilla decoder. [Liu and Zhang, 2017]

Encoder-Decoder Dependency Parsing

Paper: Encoder-Decoder Shift-Reduce Syntactic Parsing

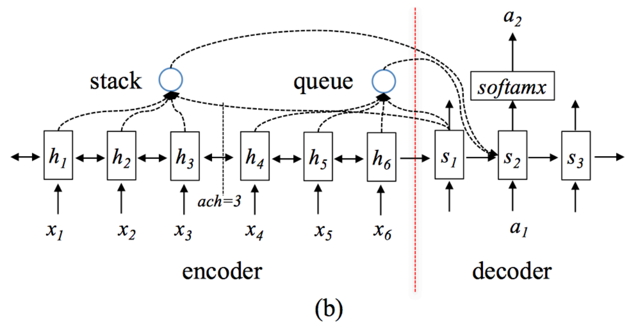


Figure 16: Stack-queue decoder. [Liu and Zhang, 2017]

Encoder-Decoder Dependency Parsing

Attention:

$$h_{l_{att_j}} = attention(1, t) = \sum_{i=1}^{t} \alpha_i h_i$$

$$h_{r_{att_j}} = attention(t + 1, n) = \sum_{i=t+1}^{n} \alpha_i h_i$$

$$s_j = g(W_{dec}[s_{j1}; e_{a_{j1}}; h_{l_{att_j}}; h_{r_{att_j}}] + b_d ec)$$

Encoder-Decoder Dependency Parsing

Attention:

$$h_{l_{att_j}} = attention(1, t; \theta_l) = \sum_{i=1}^{t} \alpha_i h_i$$

$$h_{r_{att_j}} = attention(t+1, n; \theta_r) = \sum_{i=t+1}^{n} \alpha_i h_i$$

Model	UAS (%)
Dyer et al. (2015)	92.3
Vanilla decoder	88.5
SQ decoder + average pooling	91.9
SQ decoder + attention	92.4
SQ decoder + treeLSTM	92.4

Figure 17: Results. [Liu and Zhang, 2017]

Model	UAS (%)	LAS (%)
Graph-based		
Kiperwasser and Goldberg (2016)	93.0	90.9
Dozat and Manning (2017)	95.7	94.1
Transition-based		
Chen and Manning (2014)	91.8	89.6
Dyer et al. (2015)	93.1	90.9
Kiperwasser and Goldberg (2016)†	93.9	91.9
Andor et al. (2016)	92.9	91.0
Andor et al. (2016)*	94.6	92.8
SQ decoder + attention	93.1	90.1

Figure 17: Results. [Liu and Zhang, 2017]

Encoder-Decoder Dependency Parsing

Paper: Stack-based Multi-layer Attention for Transition-based Dependency Parsing

Motivation:

- Seq2seq transition-based dependency parsing is not good.
- Two binary vectors are used to track the decoding stack.
- Multi-layer attention is introduced to capture multiple word dependencies.
- Outperform the basic seq2seq model with 1.87 UAS (en) and 1.61 UAS (zh).

Architecture

Attention Mechanism:

$e_{i,t} = v_a^\top \tanh(W_a z_{i-1} + U_a h_t + S_a s_t)$

$\alpha_{i,t} = \frac{\exp(e_{i,t})*(1-r_t)}{\sum_k \exp(e_{i,k})*(1-r_t)}$

$c_i = \sum_t \alpha_{i,t} h_t$

Multi-layer (m>1)

$e_{i,t}^m =$

$v_a^\top \tanh(W_a^m [z_{i-1}; c_i^{m-1}] + U_a h_t + S_a s_t)$

$c_i' = [c_i^1; \cdots; c_i^M]$

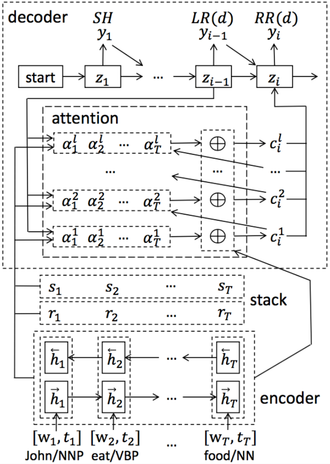


Figure 18: Parsing Architecture. [Zhang et al., 2017]

Results

Single-word features (9)
$s_1.w$; $s_1.t$; $s_1.wt$; $s_2.w$; $s_2.t$;
$s_2.wt$; $b_1.w$; $b_1.t$; $b_1.wt$
Word-pair features (8)
$s_1.wt \circ s_2.wt$; $s_1.wt \circ s_2.w$; $s_1.wts_2.t$;
$s_1.w \circ s_2.wt$; $s_1.t \circ s_2.wt$; $s_1.w \circ s_2.w$
$s_1.t \circ s_2.t$; $s_1.t \circ b_1.t$
Three-word feaures (8)
$s_2.t \circ s_1.t \circ b_1.t$; $s_2.t \circ s_1.t \circ lc_1(s_1).t$;
$s_2.t \circ s_1.t \circ rc_1(s_1).t$; $s_2.t \circ s_1.t \circ lc_1(s_2).t$;
$s_2.t \circ s_1.t \circ rc_1(s_2).t$; $s_2.t \circ s_1.w \circ rc_1(s_2).t$;
$s_2.t \circ s_1.w \circ lc_1(s_1).t$; $s_2.t \circ s_1.w \circ b_1.t$

Figure 19: Impact of attention layers. [Chen and Manning, 2014]

Results

	Dev		Test	
	UAS	LAS	UAS	LAS
seq2seq	92.02	89.10	91.84	88.84
$l = 1$	92.85	90.44	92.70	90.40
$l = 2$	93.30	91.13	93.21	90.98
$l = 3$	**93.65**	**91.52**	**93.71**	**91.60**
$l = 4$	93.49	91.29	93.42	91.24

Figure 19: Impact of attention layers. [Zhang et al., 2017]

Results

	Dev		Test	
	UAS	LAS	UAS	LAS
Our model	93.65	91.52	93.71	91.60
–pretraining	93.19	90.92	93.22	91.11
–POS	92.73	89.86	92.57	90.05
–s vector	93.18	90.68	93.02	90.89
–r vector	93.16	90.90	93.27	91.02

Figure 20: Impact of different components. [Zhang et al., 2017]

Results

Parser	PTB-SD				CTB			
	Dev		Test		Dev		Test	
	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
Z&N11	-	-	93.00	90.95	-	-	86.00	84.40
C&M14	92.20	89.70	91.80	89.60	84.00	82.40	83.90	82.40
ConBSO	-	-	91.57	87.26	-	-	-	-
Dyer15	93.20	90.90	93.10	90.90	87.20	85.90	87.20	85.70
Weiss15	-	-	93.99	92.05	-	-	-	-
K&G16	-	-	93.99	91.90	-	-	87.60	86.10
DENSE	**94.30**	91.95	94.10	91.90	87.35	85.85	87.84	86.15
seq2seq	92.02	89.10	91.84	88.84	86.21	83.80	85.80	83.53
Our model	93.65	91.52	93.71	91.60	87.28	85.30	87.41	85.40
Ensemble	94.24	**92.01**	**94.16**	**92.13**	**88.06**	**86.30**	**87.97**	**86.18**

Figure 20: Results. [Zhang et al., 2017]

Conclusions

- Vanilla seq2seq parsing model lack structural information.
- Multi-layer Attention is effective.
- Encoder-Decoder parsing model is not good enough.

Our Work

linearization

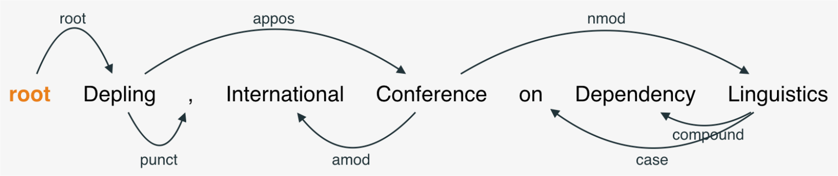


Figure 21: Dependency Tree

Seq2Seq+Attention+Ptr-Net

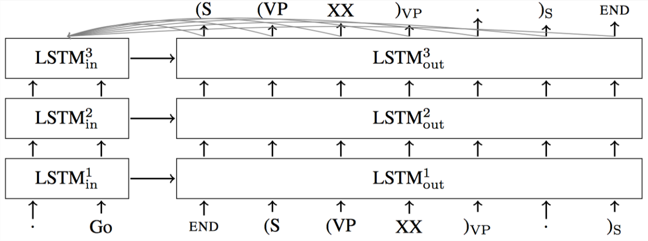


Figure 22: DeepLSTM+A seq2seq architecture. [Vinyals et al., 2015b]

- Pointer score: $u_{t,i} = \boldsymbol{V}^\top \cdot \tanh(\mathbf{W}'_1 \cdot h_i + \mathbf{W}'_2 \cdot s_{t-1})$
- Pointer probability: $p_{t,i} = \mathrm{softmax}(u_{t,i})$

Encoder-Decoder Head/Son Selection Parsing Models

- Seq2Seq (no pre-trained): 92.61% UAS, 90.68% LAS
  ([Zhang et al., 2017] 91.84% UAS, 88.84% LAS)
- $\rightarrow$ Seq2Tree model
- +pre-trained word embedding
- greedy search $\rightarrow$ beam search
- +early update
- +multi-layer attention (Neural Network structure)...
- joint Seq2Seq and Seq2Tree
- ...

Conclusion

Questions?

Paper I

Bahdanau, D., Cho, K., and Bengio, Y. (2014).
Neural machine translation by jointly learning to align and translate.
CoRR, abs/1409.0473.

Chen, D. and Manning, C. D. (2014).
A fast and accurate dependency parser using neural networks.
In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 740–750.

Paper II

Dong, L. and Lapata, M. (2016).
Language to logical form with neural attention.
In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Lapata, M., Zhang, X., and Cheng, J. (2017).
Dependency parsing as head selection.
In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*, pages 665–676.

Paper III

- Liu, J. and Zhang, Y. (2017).
  Encoder-decoder shift-reduce syntactic parsing.
  In *Proceedings of the 15th International Conference on Parsing Technologies, IWPT 2017, Pisa, Italy, September 20-22, 2017*, pages 105–114.
- Nivre, J. (2008).
  Algorithms for deterministic incremental dependency parsing.
  Computational Linguistics, 34(4):513–553.
- Sutskever, I., Vinyals, O., and Le, Q. V. (2014).
  Sequence to sequence learning with neural networks.
  In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Tesnière, L. (1959).
***Elements de syntaxe structurale*.**
Editions Klincksieck.

Vinyals, O., Fortunato, M., and Jaitly, N. (2015a).
Pointer networks.
In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2692–2700.

Vinyals, O., Kaiser, L., Koo, T., Petrov, S., Sutskever, I., and Hinton, G. E. (2015b).
Grammar as a foreign language.
In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2773–2781.

Paper V

Zhang, Z., Liu, S., Li, M., Zhou, M., and Chen, E. (2017).
Stack-based multi-layer attention for transition-based dependency parsing.
In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 1678–1683.

Thank you!