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

Zhirui Zhang, Shujie Liu, Mu Li, Ming Zhou, Enhong Chen

USTC & Microsoft -- EMNLP17

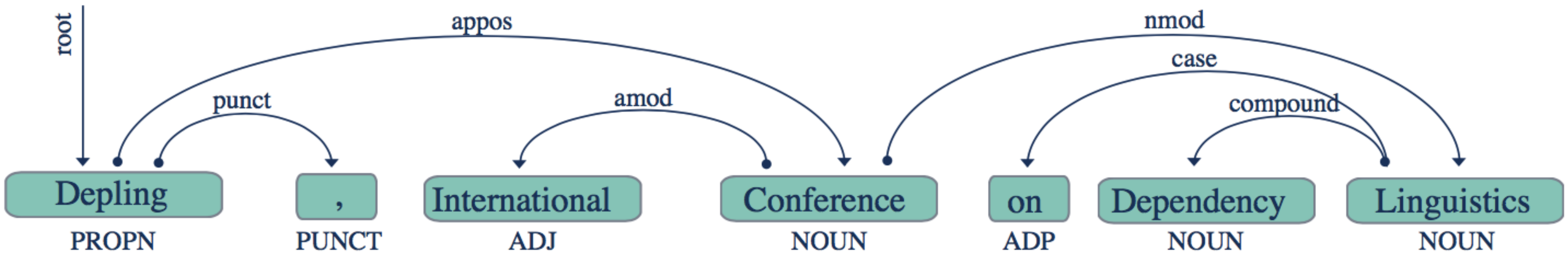
AntNLP -- Tao Ji

taoji.cs@gmail.com

Outline

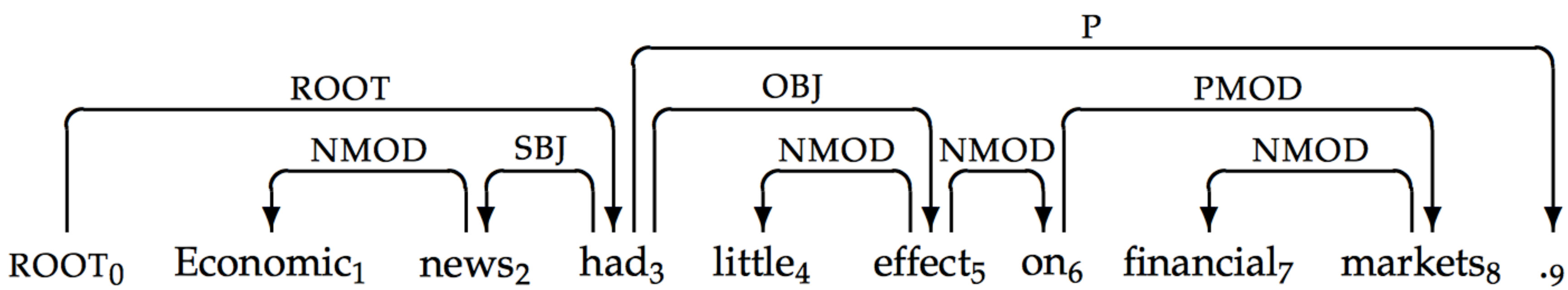
- Transition-based Dependency Parsing
- Seq2Seq Parsing Model
- Motivation
- Architecture of seq2seq parsing model
- Multi-layer Attention
- Experiments

Dependency Parsing



- 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 $l_k \in L$
- Approaches:
  - Transition-based models
  - Graph-based models

Arc-standard Example



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

Transition-based Models

- Transition system: Arc-standard, Arc-eager, Arc-hybrid, ...

  Transitions

LEFT-ARC_l	$(\sigma\|i,j\|\beta,A) \Rightarrow (\sigma,j\|\beta,A\cup\{(j,l,i)\})$
RIGHT-ARC_l^s	$(\sigma\|i,j\|\beta,A) \Rightarrow (\sigma,i\|\beta,A\cup\{(i,l,j)\})$
SHIFT	$(\sigma,i\|\beta,A) \Rightarrow (\sigma\|i,\beta,A)$

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

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

Encoder:

- Input x_i: $x_i = [W_e * e(w_i); W_t * e(t_i)]$
- $X = (x_1, x_2, \cdots, x_T) \to h = (h_1, h_2, \cdots, h_T)$
- *Deep-BiGRU* or *Deep-BiLSTM*

Vanilla Attention Mechanism:

- $e_{i,t} = v_a^\top \tanh(W_a z_{i-1} + U_a h_t)$
- $\alpha_{i,t} = \frac{\exp(e_{i,t})}{\sum_k \exp(e_{i,k})}$
- $\{v_a, W_a, U_a\} \in \theta$

Attention Mechanism:

- Two binary vectors $s = (s_1, \cdots, s_T)$ and $r = (r_1, \cdots, r_T)$
- $e_{i,t} = v_a^\top \tanh(W_a z_{i-1} + U_a h_t + S_a s_t)$
- $\alpha_{i,t} = \frac{\exp(e_{i,t}) * (1 - r_t)}{\sum_k \exp(e_{i,k}) * (1 - r_t)}$
- $c_i = \sum_t \alpha_{i,t} h_t$

Multi-layer (m>1)

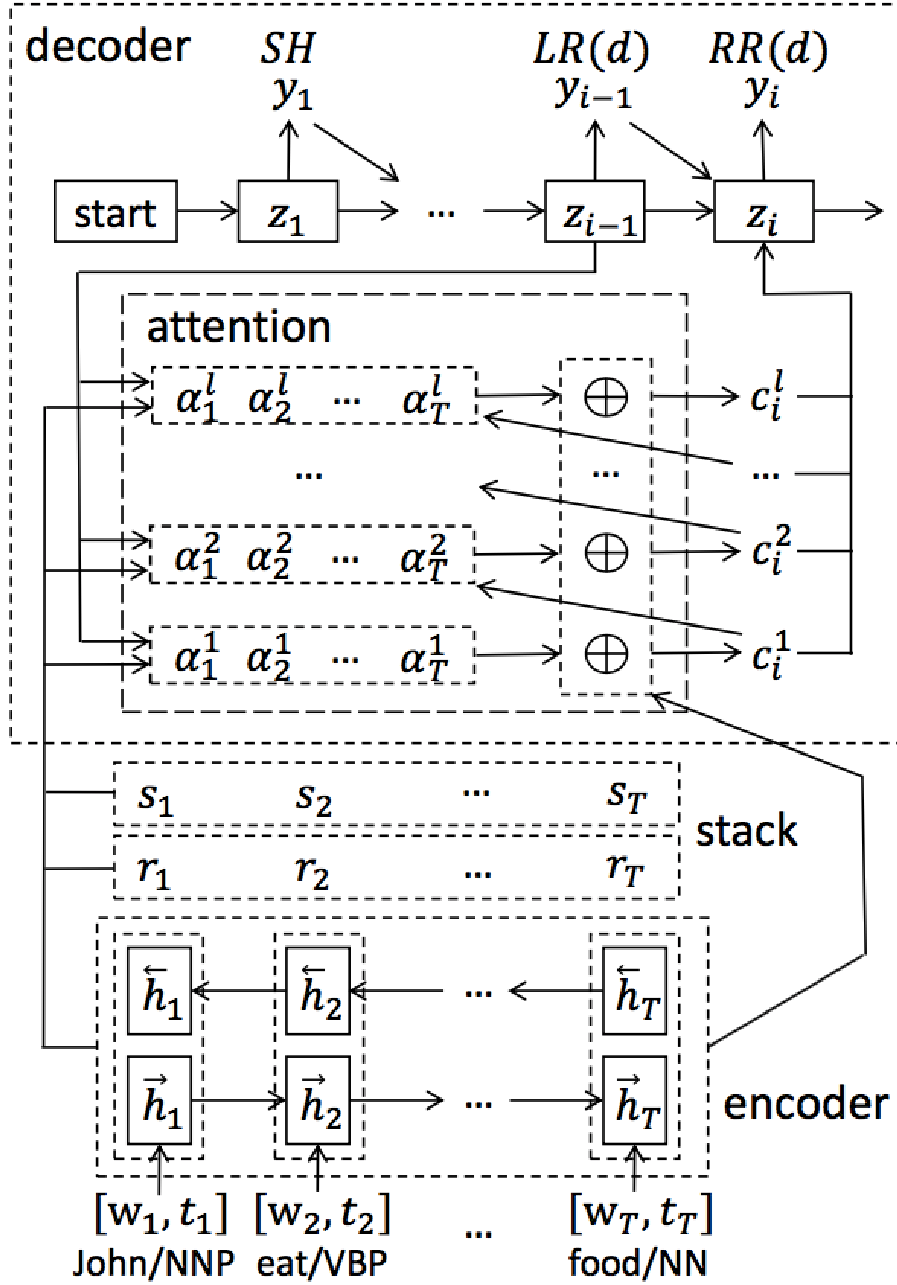
- $e_{i,t}^m = v_a^\top \tanh(W_a^m [z_{i-1}; c_i^{m-1}] + U_a h_t + S_a s_t)$
- $c_i' = [c_i^1; \cdots; c_i^M]$

Decoder:

- $I(y_i) = \begin{cases} 0 & y_i = \mathrm{SH}, W_c \leq 0 \\ 0 & y_i = \mathrm{LR}(d) \; or \; \mathrm{RR}(d), S_c < 2 \\ 1 & \text{otherwise} \end{cases}$
- $p(y_i | y_{<i}, h) = \frac{\exp(g_i) * I(y_i)}{\sum_k \exp(g_k) * I(y_k)}$
  where g_i is the ith element of $\mathrm{MLP}(z_i)$
  $z_i = \mathrm{LSTM}(y_{i-1}, z_{i-1}, c_i')$.

Architecture

$$x_i = [E(w_i); E(t_i)]$$
$$a_{i,t}^m = f(z_{i-1}, c_i^{m-1}, h_t, s_t, r_t)$$
$$c_i = \sum_t \alpha_{i,t} h_t$$
$$z_i = \mathrm{LSTM}(y_{i-1}, z_{i-1}, c_i^l)$$

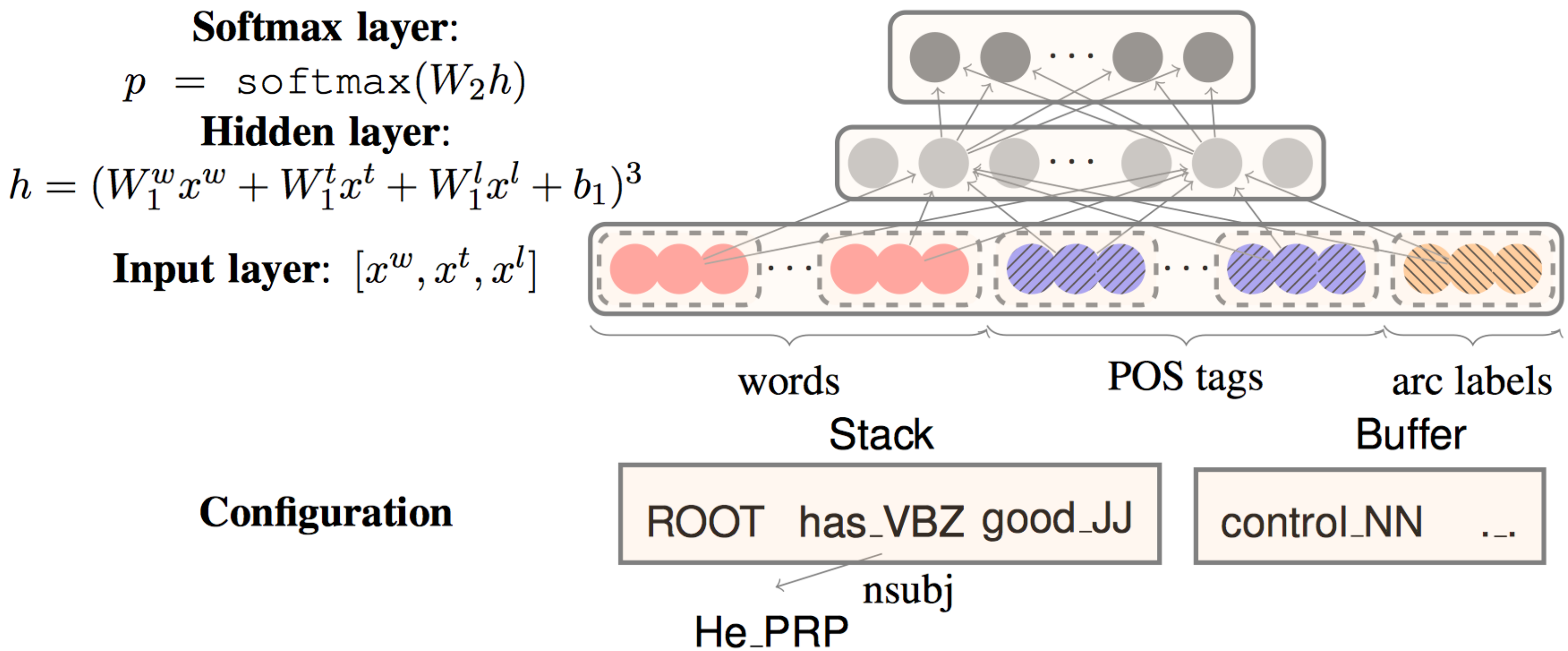


Analysis

[EMNLP14, Chen and Manning] A Fast and Accurate Dependency Parser using Neural Networks

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)
$s_2.t \circ s_1.t \circ b_1.t$; $s_2.t \circ s_1.t \circ lc_1(s_1).t$;
$s_2.t \circ s_1.t \circ rc_1(s_1).t$; $s_2.t \circ s_1.t \circ lc_1(s_2).t$;
$s_2.t \circ s_1.t \circ rc_1(s_2).t$; $s_2.t \circ s_1.w \circ rc_1(s_2).t$;
$s_2.t \circ s_1.w \circ lc_1(s_1).t$; $s_2.t \circ s_1.w \circ b_1.t$

Analysis



$$g(w_1x_1 + \cdots + w_mx_m + b) = \sum_{i,j,k}(w_iw_jw_k)x_ix_jx_k + \sum_{i,j} b(w_iw_j)x_ix_j \cdots$$

Analysis

1-layer

$$c_i^1 = f(z_{i-1}, h_{1,T}) = f(z_{i-1}, [w;t]_{1,T})$$

2-layer

$$e_{i,t}^2 = v_a^\top \tanh(W_a^2[z_{i-1}; c_i^1] + U_a h_t + S_a s_t)$$

$$c_i^2 = g(z_{i-1}, ([w;t]_{1:T}, [w;t]_{1:T}))$$

Experiments

Datasets

English: Penn Treebank (PTB) with Stanford Dependencies
Chinese: Chinese Treebank 5.1 (CTB)

Setup

- 3-layers GRU (encoder and decoder)
- 500-d hidden units
- 300-d word, 32-d POS-tag/action embedding
- 3-layers attention structure
- 0.2 dropout rate
- 8 beam size

Main 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**

Additional 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

Additional 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

Conclusion

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

Thank you!

Q&A