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

Zhirui Zhang[1], Shujie Liu[2], Mu Li[2], Ming Zhou[2], Enhong Chen[1]
University of Science and Technology of China[1] and Microsoft Research Aisa[2]

Abstract

Although sequence-to-sequence (seq2seq) network has achieved significant success in many NLP tasks such as machine translation and text summarization, simply applying this approach to transition-based dependency parsing cannot yield a comparable performance gain as in other state-of-the-art methods, such as stack-LSTM and head selection. In this paper, we propose a stack-based multi-layer attention model for seq2seq learning to better leverage structural linguistics information. In our method, two binary vectors are used to track the decoding stack in transition-based parsing, and multi-layer attention is introduced to capture multiple word dependencies in partial trees. We conduct experiments on PTB and CTB datasets, and the results show that our proposed model achieves state-of-the-art accuracy and significant improvement in labeled precision with respect to the baseline seq2seq model.

Background

Sequence-to-sequence Learning:
- Follow the attention-based encoder-decoder architecture

Encoder
- The encoder reads in the source sentence $\boldsymbol{X} = (\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_T})$ and transforms it into a sequence of hidden states $\boldsymbol{h} = (\boldsymbol{h_1}, \boldsymbol{h_2}, \dots, \boldsymbol{h_T})$ using a bi-directional RNN

Decoder
- The decoder uses another RNN to generate a corresponding target sequence $\boldsymbol{Y} = (\boldsymbol{y_1}, \boldsymbol{y_2}, \dots, \boldsymbol{y_{T'}})$ based on hidden states $\boldsymbol{h} = (\boldsymbol{h_1}, \boldsymbol{h_2}, \dots, \boldsymbol{h_T})$
- At each time i, the conditional probability of target symbol y_i is computed by

$$z_i = \mathrm{RNN}([\mathrm{emb}(y_{i-1}); c_i], z_{i-1})$$
$$p(y_i|y_{<i}, h) = \mathrm{softmax}(g(\mathrm{emb}(y_{i-1}), z_i, c_i))$$

Where z_i is the hidden state of the decoder and c_i is the source context vector

Attention Mechanism
- In attention-based seq2seq model, the context vector c_i is a weighted sum of the hidden states $\boldsymbol{h} = (\boldsymbol{h_1}, \boldsymbol{h_2}, \dots, \boldsymbol{h_T})$ with the coefficients $\alpha_{i,1}, \alpha_{i,2}, \dots, \alpha_{i,T}$ computed by

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

Our Approach

Architecture
- Transition-based dependency parsing conceptualizes the process of transforming a sentence into a dependency tree as a sequence of actions (SHIFT(SH), LEFT-ARC(LR(d)), RIGHT-ARC(RR(d))). The whole architecture of our approach →

Encoder
- Each word w_i is additionally represented by x_i, the concatenation of two vectors corresponding to w_i's lexical and POS tag t_i embedding

Decoder
- Add some tree constraints to make sure predictions can generate a dependency tree

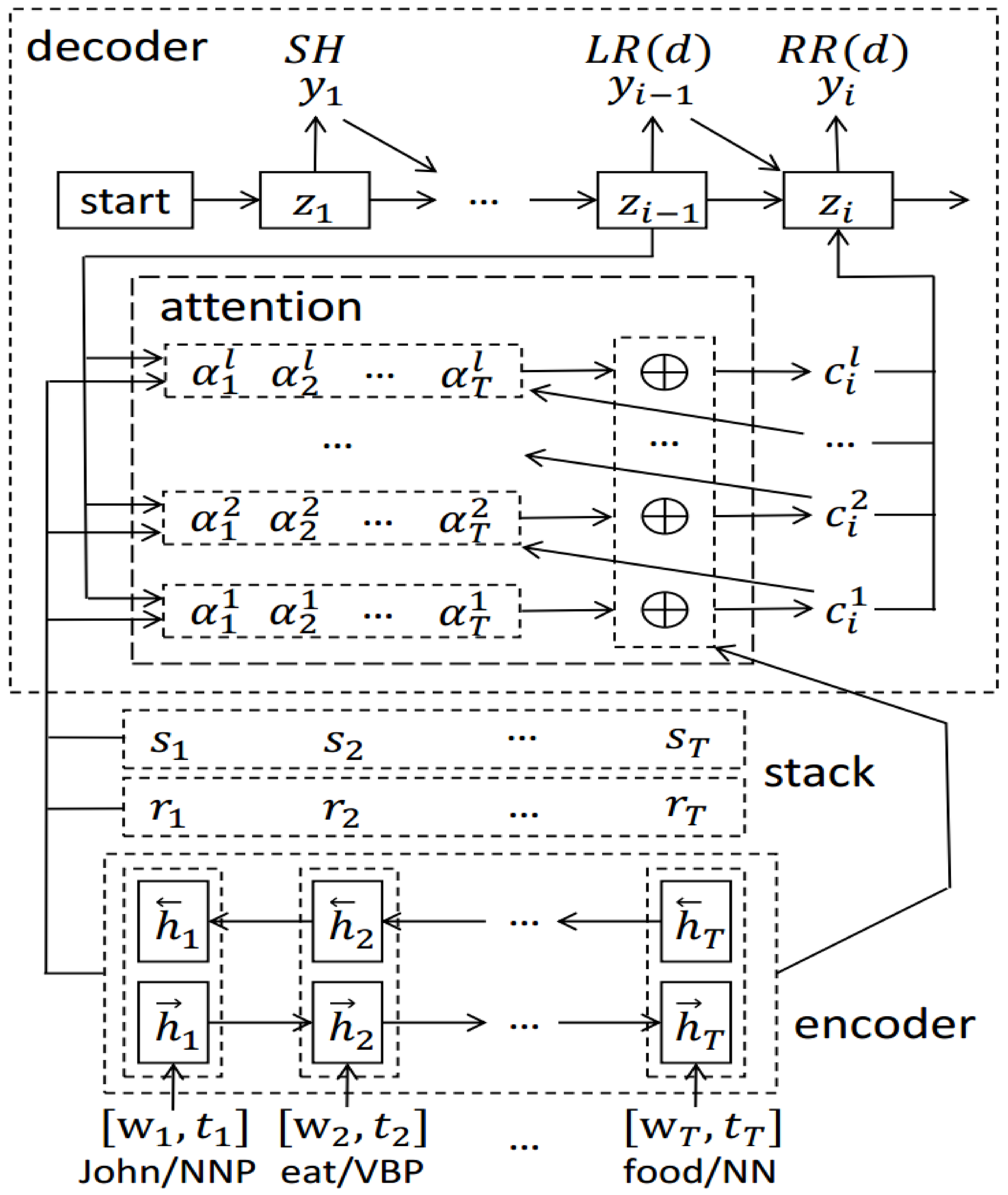
$$p(y_i|y_{<i}, h) = \frac{\exp(g_i) * I(y_i)}{\sum_k \exp(g_k) * I(y_k)} \qquad I(y_i) = \begin{cases} 0 & y_i = \mathrm{SH}, W_c \le 0 \\ 0 & y_i = \mathrm{LR}(d) \text{ } or \text{ } \mathrm{RR}(d), S_c < 2 \\ 1 & otherwise \end{cases}$$

Attention Mechanism
- Introduce stack information to guide the attention model to focus more on words in the stack. The coefficients $\alpha_{i,1}, \alpha_{i,2}, \dots, \alpha_{i,T}$ computed by

$$\alpha_{i,t} = \frac{\exp(e_{i,t}) * (1 - r_t)}{\sum_k \exp(e_{i,k}) * (1 - r_k)} \qquad e_{i,t} = v_a^T \tanh(W_a z_{i-1} + U_a h_t + S_a s_t)$$

- Leverage multi-layer attention to capture multiple word dependencies in partial trees

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



Experiments

Dataset
- Stanford Dependencies conversion of Penn Treebank (PTB-SD) and Chinese Treebank 5.1 (CTB)

System Settings
- 3-layer GRU is used for encoder and decoder (hidden size: 500)
- leverage 300-dimensional pre-trained GloVe vectors to initialize embedding matrix

Evaluation metric
- Unlabeled attachment scores (UAS) and Labeled attachment scores (LAS)

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**

Table 1: Results of various state-of-the-art parsing systems on English dataset (PTB with Stanford Dependencies) and Chinese dataset (CTB). The numbers reported from different systems are taken from: Z&N11 (Zhang and Nivre, 2011); C&M14 (Chen and Manning, 2014); ConBSO (Wiseman and Rush, 2016); Dyer15 (Dyer et al., 2015); Weiss15 (Weiss et al., 2015); K&G16 (Kiperwasser and Goldberg, 2016); DENSE (Zhang et al., 2017).

Note that Dozat and Manning(2016) achieve 95.74 UAS and 89.30 UAS on PTB-SD and CTB datasets respectively. For ensemble, we train 4 models using the same network with different random initialization.

Analysis

	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

Table 2: Impact of l on English PTB dataset.

	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

Table 3: Impact of the different components on English PTB dataset.