Head-Lexicalized Bidirectional Tree LSTMs

WeiYang

weiyang@godweiyang.com

www.godweiyang.com

East China Normal University
Department of Computer Science and Technology

2018.03.08

Outline

Comparision

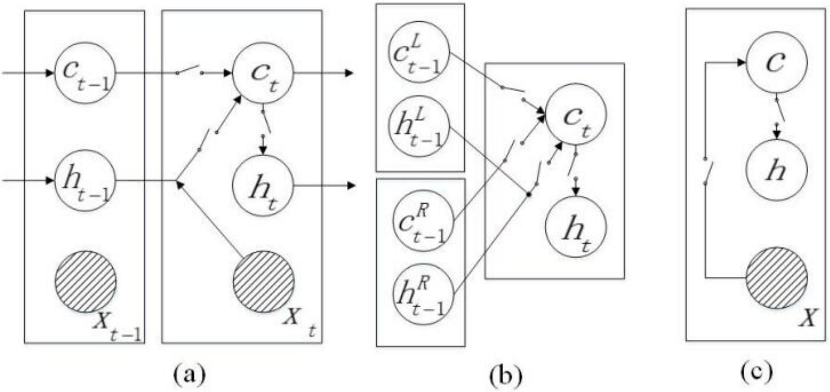

Figure: Topology of sequential and tree LSTMs.

Comparision

- Traditional tree LSTMs have no direct association between non-leaf constituent nodes and input words.
- However, each node in a constituent tree structure is governed by a head word.
- So the head lexical information of each constituent word can be added as the input node x.
- Use neural attention mechanism instead of specific rules which are language- and formalism-dependent.
- Add bidirectional extension of the tree structured LSTM.

Head-Lexicalized Constituent Tree

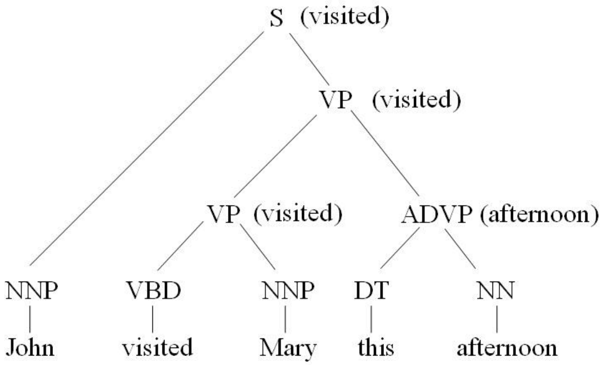

Figure: Head-Lexicalized Constituent Tree.

Model

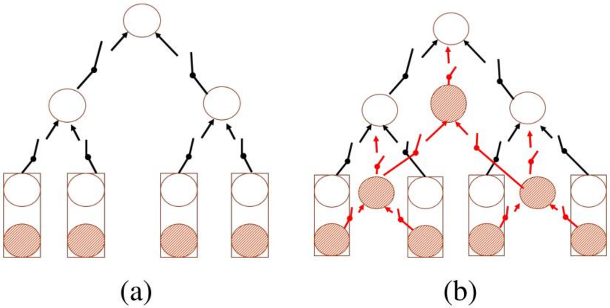

Figure: Contrast between Zhu et al. (2015) (a) and this paper (b).

Head Lexicon

$$x_t = z_t \otimes x_{t-1}^L + (1 - z_t) \otimes x_{t-1}^R$$
$$z_t = \sigma(W_{zx}^L x_{t-1}^L + W_{zx}^R x_{t-1}^R + b_z)$$

Lexicalized Tree LSTM

$$
\begin{aligned}
i_t &= \sigma\Big(\mathbf{W_{xi}x_t} + \\
&\sum_{N\in\{L,R\}} (W_{hi}^N h_{t-1}^N + W_{ci}^N c_{t-1}^N) + b_i\Big) \\
f_t^L &= \sigma\Big(\mathbf{W_{xf}x_t} + \\
&\sum_{N\in\{L,R\}} (W_{hf_l}^N h_{t-1}^N + W_{cf_l}^N c_{t-1}^N) + b_{f_l}\Big) \\
f_t^R &= \sigma\Big(\mathbf{W_{xf}x_t} + \\
&\sum_{N\in\{L,R\}} (W_{hf_r}^N h_{t-1}^N + W_{cf_r}^N c_{t-1}^N) + b_{f_r}\Big) \\
o_t &= \sigma\Big(\mathbf{W_{xo}x_t} + \\
&\sum_{N\in\{L,R\}} W_{ho}^N h_{t-1}^N + W_{co} c_t + b_o\Big)
\end{aligned}
$$

$$
g_t = \tanh\Big(\mathbf{W_{xg}x_t} + \sum_{N\in\{L,R\}} W_{hg}^N h_{t-1}^N + b_g\Big)
$$

Bidirectional Extensions

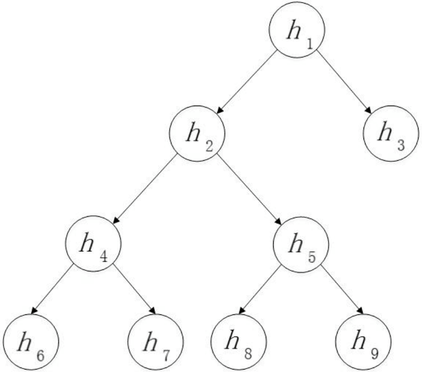

Figure: Top-down tree LSTM.

Bidirectional Extensions

$$h_t = o_t \otimes \tanh(c_{t-1})$$
$$c_t = f_t \otimes c_{t-1} + i_t \otimes g_t$$
$$g_t = \tanh(W^N_{xg\downarrow} x_{t-1} + W^N_{hg\downarrow} h_{t-1} + b^N_{g\downarrow})$$

$$i_t = \sigma(W^N_{xi\downarrow} x_t + W^N_{hi\downarrow} h_{t-1} + W^N_{ci\downarrow} c_{t-1} + b^N_{i\downarrow})$$
$$f_t = \sigma(W^N_{xf\downarrow} x_t + W^N_{hf\downarrow} h_{t-1} + W^N_{cf\downarrow} c_{t-1} + b^N_{f\downarrow})$$
$$o_t = \sigma(W^N_{xo\downarrow} x_t + W^N_{ho\downarrow} h_{t-1} + W^N_{co\downarrow} c_t + b^N_{o\downarrow})$$

Hint: The top-down tree LSTM must be built after bottom-up tree LSTM.

Sentiment Classification

$$h = \tilde{h}_{ROOT\uparrow} \oplus \tilde{h}_{ROOT\downarrow} \oplus \frac{1}{n}\sum_{i=1}^{n} \tilde{h}'_i$$

$$h_l = \mathrm{ReLU}(W_{hl}h + b_{hl})$$

$$P = \mathrm{softmax}(W_{lp}h_l + b_{lp})$$

$$p_j = P[j]$$

Loss:

$$L(\Theta) = -\sum_{i=1}^{|D|} \log p_{y_i} + \frac{\lambda}{2}\|\Theta\|^2$$

Sentiment Classification

Model	5-class		binary	
	Root	Phrase	Root	Phrase
RNTN(Socher et al., 2013b)	45.7	80.7	85.4	87.6
BiLSTM(Li et al., 2015)	49.8	83.3	86.7	-
DepTree(Tai et al., 2015)	48.4	-	85.7	-
ConTree(Le and Zuidema, 2015)	49.9	-	88.0	-
ConTree(Zhu et al., 2015)	50.1	-	-	-
ConTree(Li et al., 2015)	50.4	83.4	86.7	-
ConTree(Tai et al., 2015)	51.0	-	88.0	-
BiLSTM (Our implementation)	49.9	82.7	87.6	91.8
ConTree (Our implementation)	51.2	83.0	88.5	92.5
Top-down ConTree	51.0	82.9	87.8	92.1
ConTree + Lex	52.8	83.2	89.2	92.3
BiConTree	**53.5**	**83.5**	**90.3**	**92.8**

Figure: Test set accuracies for sentiment classification tasks.

Question Type Classification

- ENTY, HUM, LOC, DESC, NUM and ABBR.
- Different from Sentiment Classification, only root node has one label.

Model	Accuracy
Baseline BiLSTM	93.8
Baseline BottomUp ConTree LSTM	93.4
SVM (Silva et al., 2011)	**95.0**
Bidirectional ConTree LSTM	94.8

Figure: TREC question type classification results.

Syntactic Parsing

$$\hat{y} = \arg\max_{y \in Y(x)} \{f(x, y; \Theta)\}$$

$$f(x, y; \Theta) = \sum_{r \in node(x,y)} Score(r; \Theta)$$

$$o_A^{BC} = \mathrm{ReLU}(W_s^L n_B + W_s^R n_C + W_s^H h_A + b^s)$$

$$Score_A^{BC} = \log(\mathrm{softmax}(o_A^{BC}))[A]$$

$$L(\Theta) = \frac{1}{|D|} \sum_{i=1}^{|D|} r_i(\Theta) + \frac{\lambda}{2} ||\Theta||^2$$

$$r_i(\Theta) = \max_{\hat{y}_i \in Y(x_i)} (0, f(x_i, \hat{y}_i; \Theta) + \Delta(y_i, \hat{y}_i) - f(x_i, y_i; \Theta))$$

$$\Delta(y_i, \hat{y}_i) = \sum_{node \in \hat{y}_i} \kappa 1\{node \notin y_i\}$$

Syntactic Parsing

Model	F_1
Baseline (Charniak (2000))	89.7
ConTree	90.6
ConTree+Lex	90.9
Our 10-best Oracle	94.8

Figure: Reranking results on WSJ test set.

Syntactic Parsing

Parser	dev (all)	test$\leq$ 40	test (all)
Stanford PCFG	85.8	86.2	85.5
Stanford Factored	87.4	87.2	86.6
Factored PCFGs	89.7	90.1	89.4
Collins			87.7
SSN (Henderson)			89.4
Berkeley Parser			90.1
CVG (RNN)	85.7	85.1	85.0
CVG (SU-RNN)	91.2	91.1	90.4
Charniak-SelfTrain			91.0
Charniak-RS			92.1

Figure: Reranking results of Socher et al. (2013a).

Other Applications

This model can be used for all tasks that require representation learning for sentences, given their constituent syntax.

- Language Model
- Relation Extraction

Conclusions

- Head-lexicalization.
- Top-down extension.
- Bidirectional constituent Tree LSTM.