Appendix: Subword Encoding in Lattice LSTM for Chinese Word Segmentation

Jie Yang[♠♣], **Yue Zhang**[♦], **Shuailong Liang**[▲]
[♠]Brigham and Women's Hospital. Boston, USA
[♣]Harvard Medical School, Harvard University. Boston, USA
[♦]School of Engineering, Westlake University. Hangzhou, China
[▲]Singapore University of Technology and Design. Singapore
jieynlp@gmail.com
yue.zhang@wias.org.cn
shuailong_liang@mymail.sutd.edu.sg

1 Standard LSTM

Equation 1 shows the calculation of $\overrightarrow{\mathbf{h}}_i$ which is the forward LSTM representation of character c_i .

$$\begin{aligned}\begin{bmatrix}\mathbf{o}_i\\ \mathbf{f}_i\\ \widetilde{\boldsymbol{c}}_i\end{bmatrix} &= \begin{bmatrix}\sigma\\ \sigma\\ tanh\end{bmatrix}\Big(\mathbf{W}^\top\begin{bmatrix}\mathbf{x}_i\\ \overrightarrow{\mathbf{h}}_{i-1}\end{bmatrix}+\mathbf{b}\Big)\\ \mathbf{i}_i &= \mathbf{1}-\mathbf{f}_i\\ \mathbf{c}_i &= \mathbf{f}_i\odot\mathbf{c}_{i-1}+\mathbf{i}_i\odot\widetilde{\boldsymbol{c}}_i\\ \overrightarrow{\mathbf{h}}_i &= \mathbf{o}_i\odot tanh(\mathbf{c}_i)\end{aligned} \tag{1}$$

where $\mathbf{i}_i, \mathbf{f}_i$ and $\mathbf{o}_i$ denote a set of input, forget and output gates, respectively. We choose the coupled LSTM structure (Greff et al., 2017) which sets the input gate $\mathbf{i}_i = \mathbf{1} - \mathbf{f}_i$. $\mathbf{c}_i$ is the memory cell of character c_i. $\mathbf{W}^\top$ and $\mathbf{b}$ are model parameters. $\sigma()$ represents the sigmoid function. The backward LSTM has the symmetrical equations.

2 Lattice LSTM

Equation 2 shows the structure of LSTMcell:

$$\begin{aligned}\begin{bmatrix}\mathbf{i}_{b,e}\\ \mathbf{f}_{b,e}\\ \widetilde{\boldsymbol{c}}_{b,e}\end{bmatrix} &= \begin{bmatrix}\sigma\\ \sigma\\ tanh\end{bmatrix}\Big(\mathbf{W}_s^\top\begin{bmatrix}\mathbf{e}_{w_{b,e}}\\ \overrightarrow{\mathbf{h}}_b\end{bmatrix}+\mathbf{b}_s\Big)\\ \mathbf{c}_{b,e} &= \mathbf{f}_{b,e}\odot\mathbf{c}_b^c+\mathbf{i}_{b,e}\odot\widetilde{\boldsymbol{c}}_{b,e}\end{aligned} \tag{2}$$

where $\mathbf{c}_{b,e}$ is the memory cell of the shortcut path starting from character c_b to character c_e. $\mathbf{W}_s^\top$ and $\mathbf{b}_s$ are model parameters of the shortcut path LSTM.

The subsequence output memory vector $\mathbf{c}_{b,i}$ links to the end character c_i as the input to calculate the hidden vector $\overrightarrow{\mathbf{h}}_i$ of c_i. For character c_i with multiple subsequence memory cell inputs[1], we define the input set as $\mathbb{C}_i = \{\mathbf{c}_{b,i} | b \in \{b' | w_{b',i} \in \mathbb{D}\}\}$, we assign a unique gate for each subsequence input to control its contribution:

$$\mathbf{i}_{b,i} = \sigma\big(\mathbf{W}^{g\top}\begin{bmatrix}\mathbf{x}_i\\ \mathbf{c}_{b,i}\end{bmatrix}+\mathbf{b}^g\big) \tag{3}$$

where $\mathbf{W}^{g\top}$ and $\mathbf{b}^g$ are model parameters for the gate.

Until now, we have calculated the subsequence memory inputs $\mathbb{C}_i$ and their control gates $\mathbb{I}_i = \{\mathbf{i}_{b,i} | b \in \{b' | w_{b',i} \in \mathbb{D}\}\}$. Following the idea of coupled LSTM (Greff et al., 2017) which keeps the sum of input and forget gate as $\mathbf{1}$, we normorize all the subsequence gates $\mathbb{I}_i$ with the standard LSTM input gate $\mathbf{i}_i$ to ensure their sum equals to $\mathbf{1}$ (Eq. 4).

$$\begin{aligned}\boldsymbol{\alpha}_{b,i} &= \frac{exp(\mathbf{i}_{b,i})}{exp(\mathbf{i}_i)+\sum_{\mathbf{i}_{b',i}\in\mathbb{I}_i} exp(\mathbf{i}_{b',i})}\\ \boldsymbol{\alpha}_i &= \frac{exp(\mathbf{i}_i)}{exp(\mathbf{i}_i)+\sum_{\mathbf{i}_{b',i}\in\mathbb{I}_i} exp(\mathbf{i}_{b',i})}\end{aligned} \tag{4}$$

$\boldsymbol{\alpha}_{b,i}$ and $\boldsymbol{\alpha}_i$ are the subsequence memory gate and the standard LSTM input gate after the normalization, respectively. The final forward lattice LSTM representation $\overrightarrow{\mathbf{h}}_i$ of character c_i is calculated as:

$$\begin{aligned}\begin{bmatrix}\mathbf{o}_i\\ \mathbf{f}_i\\ \widetilde{\boldsymbol{c}}_i\end{bmatrix} &= \begin{bmatrix}\sigma\\ \sigma\\ tanh\end{bmatrix}\Big(\mathbf{W}^\top\begin{bmatrix}\mathbf{x}_i\\ \overrightarrow{\mathbf{h}}_{i-1}\end{bmatrix}+\mathbf{b}\Big)\\ \mathbf{i}_i &= \mathbf{1}-\mathbf{f}_i\\ \mathbf{c}_i &= \sum_{\mathbf{c}_{b,i}\in\mathbb{C}_i}\boldsymbol{\alpha}_{b,i}\odot\mathbf{c}_{b,i}+\boldsymbol{\alpha}_i\odot\widetilde{\boldsymbol{c}}_i\\ \overrightarrow{\mathbf{h}}_i &= \mathbf{o}_i\odot tanh(\mathbf{c}_i)\end{aligned} \tag{5}$$

[1]e.g. The first "院(College)" in Figure **??** takes two subsequence memory vectors of both "学院(Academy)" and "科学院(Academy of Sciences)" as input.

where $\mathbf{W}^\top$ and $\mathbf{b}$ are the model parameters which are the same with the standard LSTM in Eq. 2. Compare with Eq. 2, Eq. 5 has a more complex memory calculation step which integrates both the standard character LSTM memory $\widetilde{\mathbf{c}}_i$ and all the matched subsequence memory inputs $\mathbb{C}_i$. In this respect, we can regard the lattice LSTM as an extension of the standard LSTM with the ability of taking multiple inputs.

The backward lattice LSTM $\overleftarrow{\mathbf{h}}_i$ has a symmetrical calculation process. The final hidden vector $\mathbf{h}_i$ is the concatenation of the hidden vectors on both directions.

3 CRF

A standard CRF layer is used. The probability of a label sequence $y = l_1, l_2, \ldots, l_m$ is

$$P(y|s) = \frac{exp(\sum_{i=1}^{m}(F(l_i) + L(l_{i-1}, l_i)))}{\sum_{y' \in \mathbb{C}(s)} exp(\sum_{i=1}^{m}(F(l'_i) + L(l'_{i-1}, l'_i))}, \quad (6)$$

where $\mathbb{C}(s)$ is the set of all possible label sequences on sentence s and y' is an arbitary label sequence. $F(l_i) = \mathbf{W}^{l_i}\mathbf{h}_i + b^{l_i}$ is the emission score from hidden vector $\mathbf{h}_i$ to label l_i. $L(l_{i-1}, l_i)$ is the transition score from l_{i-1} to l_i. $\mathbf{W}^{l_i}$ and b^{l_i} are model parameters specific to label l_i.

4 Statistics of datasets

The statistics of four examined datasets are shown in Table 1

Dataset	Type	Train	Dev	Test
CTB6	Sentence	23.4k	2.08k	2.80k
	Word	641k	59.9k	81.6k
	Char	1.06m	100k	134k
PKU	Sentence	17.2k	1.91k	1.95k
	Word	1.01m	99.9k	104k
	Char	1.66m	164k	173k
MSR	Sentence	78.2k	8.69k	3.99k
	Word	2.12m	247k	107k
	Char	3.63m	417k	184k
Weibo	Sentence	20.1k	2.05k	8.59k
	Word	421k	43.7k	188k
	Char	689k	73.2k	316k

Table 1: Statistics of datasets.

References

Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. 2017. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10):2222–2232.