Neural Machine Translation by Jointly Learning to Align and Translate

Dzmitry Bahdanau, KyungHyun Cho, Yoshua Bengio

Presented by Xinyuan Zhang

April 17, 2017
Introduction

- Neural machine translation models often encode a source sentence into a fixed-length vector from which a decoder generates a translation. However, they suffer on long sentences.
- Instead of having to encode all information in the source sentence into a fixed-length vector, the model is extended by allowing it to automatically (soft-)search for parts of a source sentence that are relevant to predicting a target word.
RNN Encoder-Decoder

- **RNN Encoder**: Map the input sentence $\mathbf{x} = (x_1, \ldots, x_{T_x})$ into a fixed-length context vector $c$.

  \[ h_t = f_e(x_t, h_{t-1}) \]  
  \[ c = q(\{h_1, \ldots, h_{T_x}\}) \]  

  where $h_t \in \mathbb{R}^n$ is the annotation of $x_t$.

- **RNN Decoder**: Predict the output sentence $\mathbf{y} = (y_1, \ldots, y_{T_y})$ by maximizing the probability.

  \[ p(\mathbf{y}) = \prod_{t=1}^{T_y} p(y_t | \{y_1, \ldots, y_{t-1}\}, \mathbf{x}) \]  
  \[ p(y_t | \{y_1, \ldots, y_{t-1}\}, \mathbf{x}) = g(y_{t-1}, s_t, c) \]  

  where $s_t$ is the hidden state of the RNN.

  \[ s_t = f_d(s_{i-1}, y_{i-1}) \]
Model

$p(y_i|y_1, ..., y_{i-1}, X) = g(y_{i-1}, s_i, c_i)$

$s_i = f(s_{i-1}, y_{i-1}, c_i)$

$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$

$\alpha_{i,j} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}$

$e_{ij} = a(s_{i-1}, h_j)$

where $a$ is a feed forward neural network.

$h_j = [\rightarrow h_j; \leftarrow h_j]_{concat}$
Gated hidden unit (Cho et al. 2014) is used as the activation function $f$ of RNN. The forward states of BiRNN are computed by

\[
\mathbf{h}_i = f(x_i, h_{i-1}) = (1 - \mathbf{z}_i) \odot \mathbf{h}_{i-1} + \mathbf{z}_i \odot \mathbf{h}_i \tag{6}
\]

\[
\mathbf{\tilde{h}}_i = \tanh(\mathbf{W}_E x_i + \mathbf{U} [\mathbf{r}_i \odot \mathbf{h}_{i-1}]) \tag{7}
\]

\[
\mathbf{z}_i = \sigma(\mathbf{W}_z E x_i + \mathbf{U}_z \mathbf{h}_{i-1}) \tag{8}
\]

\[
\mathbf{r}_i = \sigma(\mathbf{W}_r E x_i + \mathbf{U}_r \mathbf{h}_{i-1}) \tag{9}
\]

where $\odot$ is an element-wise multiplication, and $E$ is the word embedding matrix. The backward states $(\mathbf{h}_1, ..., \mathbf{h}_{T_x})$ are computed similarly. By concatenating $\mathbf{h}_i$ and $\mathbf{\tilde{h}}_i$

\[
\mathbf{h}_i = \left[ \begin{array}{c} \mathbf{h}_i \\ \mathbf{\tilde{h}}_i \end{array} \right]. \tag{10}
\]
The alignment model $a$ is designed as

$$e_{ij} = a(s_{i-1}, h_j) = \nu_a^\top \tanh(W_a s_{i-1} + U_a h_j) \quad (11)$$

The RNN hidden states $s_i$ are computed by gated hidden unit

$$s_i = f(s_{i-1}, y_{i-1}, c_i) = (1 - z_i) \circ s_{i-1} + z_i \circ \tilde{s}_i \quad (12)$$

$$\tilde{s}_i = \tanh(W E_{y_{i-1}} + U s_{i-1} + C c_i) \quad (13)$$

$$z_i = \sigma(W_z E_{y_{i-1}} + U_z s_{i-1} + C_z c_i) \quad (14)$$

$$r_i = \sigma(W_r E_{y_{i-1}} + U_r s_{i-1} + C_r c_i) \quad (15)$$

The probability of a target word $y_i$ is defined as

$$p(y_i|s_i, y_{i-1}, c_i) \propto \exp(y_i^\top W_o t_i) \quad (16)$$

$$t_i = [\max\{\tilde{t}_{i,2j-1}, \tilde{t}_{i,2j}\}]_{j=1,...,l}^\top \quad (17)$$

where $\tilde{t}_{i,k}$ is the $k$-th element of a vector $\tilde{t}_i$

$$\tilde{t}_i = U_o s_i + V_o E_{y_{i-1}} + C_o c_i \quad (18)$$
### Table: BLEU scores of the trained models computed on the test set.

RNNsearch-50* was trained much longer until the performance on the development set stopped improving. Moses is a conventional phrase-based translation method.

<table>
<thead>
<tr>
<th>Model</th>
<th>All</th>
<th>No UNK</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNNencdec-30</td>
<td>13.93</td>
<td>24.19</td>
</tr>
<tr>
<td>RNNsearch-30</td>
<td>21.50</td>
<td>31.44</td>
</tr>
<tr>
<td>RNNencdec-50</td>
<td>17.82</td>
<td>26.71</td>
</tr>
<tr>
<td>RNNsearch-50</td>
<td>26.75</td>
<td>34.16</td>
</tr>
<tr>
<td>RNNsearch-50*</td>
<td>28.45</td>
<td>36.15</td>
</tr>
<tr>
<td>Moses</td>
<td>33.30</td>
<td>35.63</td>
</tr>
</tbody>
</table>
Figure: The BLEU scores of the generated translations on the test set with respect to the lengths of the sentences. The results are on the full test set which includes sentences having unknown words to the models.