Deliberation Networks: Sequence Generation Beyond One-Pass Decoding

Yingce Xia, Fei Tian, Lijun Wu, Jianxin Lin, Tao Qin, Nenghai Yu, Tie-Yan Liu

Lisa Kuhn

Recent Advances In Sequence-To-Sequence Learning
Introduction

- Deliberation: process of polishing (e.g. an essay) while looking at how a local element fits in its global environment
- Output of a neural network depends partly on what has been output already, not what will be output
- Extended encoder-decoder model with second decoder to do the deliberation process
Encoder

- Encodes input sequence $x$ to hidden states $H = \{h_1, h_2, ..., h_{T_x}\}$

$$h_i = \text{RNN}(x_i, h_{i-1})$$
First-pass Decoder

- Generates hidden states $\hat{s}$, first-pass sequence $\hat{y}$
- Attention Model:
  \[
  ctx_e = \sum_{i=1}^{T_x} \alpha_i h_i
  \]
  \[
  \alpha_i \propto \exp(v^T \alpha \tanh(W^c_{att,h} h_i + W^c_{att,\hat{s}} \hat{s}_{j-1}))
  \]
- Hidden state calculation:
  \[
  \hat{s}_j = \text{RNN}([\hat{y}_{j-1}; ctx_e], \hat{s}_{j-1})
  \]
- Apply affine transformation on $[\hat{s}_j; ctx_e; \hat{y}_{j-1}]$
- Softmax layer $\rightarrow$ sample out $\hat{y}_j$ from multinomial distribution
Second-pass Decoder

- Takes:
  - Previous hidden state $s_{t-1}$
  - Previously decoded word $y_{t-1}$
  - Source contextual information $ctx'_e$
  - First-pass contextual information $ctx_c$

- $ctx'_e$ computed similarly to $ctx_e$, last hidden state of second-pass decoder is used ($s_{t-1}$ instead of $\hat{s}_{j-1}$)
Second-pass Decoder

- Attention Model:
  
  \[ \text{ctx}_c = \sum_{j=1}^{T_y} \beta_j [\hat{s}_j; \hat{y}_j] \]

  \[ \beta_j \propto \exp(v_\beta^T \tanh(W_{att, s\hat{y}} [\hat{s}_j; \hat{y}_j] + W_{att, s_{t-1}})) \]

- Hidden state calculation:
  
  \[ s_t = \text{RNN}([y_{t-1}; \text{ctx}'_e; \text{ctx}_c], s_{t-1}) \]

- To generate \( y_t \) further transform \([s_t; \text{ctx}'_e; \text{ctx}_c; y_{t-1}]\) similar to sampling of \( \hat{y}_j \)
Training
For this setting, data log likelihood specialized to

\[
\frac{1}{n} \sum_{(x, y) \in \mathcal{D}_{XY}} J(x, y; \theta_e, \theta_1, \theta_2)
\]

where

\[
J(x, y; \theta_e, \theta_1, \theta_2) = \log \sum_{y' \in \mathcal{Y}} P(y | y', E(x; \theta_e); \theta_2) P(y' | E(x; \theta_e); \theta_1)
\]

\( \mathcal{Y} \) is the collection of all possible target sentences
• Derivation of $\mathcal{J}$ w.r.t $\theta_1$:

$$\nabla_{\theta_1} \mathcal{J}(x, y; \theta_e, \theta_1, \theta_2) = \sum_{y' \in \mathcal{Y}} P(y'|E(x; \theta_e); \theta_1) \frac{\nabla_{\theta_1} P(y'|E(x; \theta_e); \theta_2)}{\sum_{y'' \in \mathcal{Y}} P(y''|E(x; \theta_e); \theta_2) P(y'|E(x; \theta_e); \theta_1)}$$

• Computationally not feasible because of $\mathcal{Y}$

• Solution: Monte Carlo based method to optimize lower bound of $\mathcal{J}$ function $\rightarrow \tilde{\mathcal{J}}$

$$\tilde{\mathcal{J}}(x, y; \theta_e, \theta_1, \theta_2) = \sum_{y' \in \mathcal{Y}} P(y'|E(x; \theta_e); \theta_1) \log P(y'|E(x; \theta_e); \theta_2)$$
Algorithm 1: Algorithm to train the deliberation network

**Input:** Training data corpus $D_{XY}$; minibatch size $m$; optimizer $Opt(\cdots)$ with gradients as input;

**while models not converged do**

- Randomly sample a mini-batch of $m$ sequence pairs $\{x^{(i)}, y^{(i)}\}$ $\forall i \in [m]$ from $D_{XY}$;
- For any $x^{(i)}$ where $i \in [m]$, sample $y'^{(i)}$ according to distribution $P(\cdot | E(x^{(i)}; \theta_e); \theta_1)$;
- Perform parameter update: $\Theta \leftarrow \Theta + Opt(\frac{1}{m} \sum_{i=1}^{m} G(x^{(i)}, y^{(i)}, y'^{(i)}; \Theta))$. 


Full Training Process

- Pre-train standard encoder-decoder based NMT models until convergence
- Deliberation network encoder initialized by encoder of standard model
- Both deliberation network decoders are initialized by the decoder of the pre-trained model
- Train deliberation network until convergence
- Use beam search to sample output by first decoder
Results
Neural Machine Translation

- English-to-French translation on WMT’14 and newstest datasets
- Chinese-to-English translation on n LDC corpus and NIST datasets
- Removed sentences with more than 50 words, limit source and target words
- Two models: shallow and deep model
Shallow Model

- Based on single-layer GRU model RNNSearch
- Baselines:
  - Standard NMT model RNNSearch
  - Standard NMT model with two stacked decoding layers
  - Review Network
### Table 1: BLEU scores of En→Fr translation

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$M_{\text{base}}$</th>
<th>$M_{\text{dec} \times 2}$</th>
<th>$M_{\text{reviewer} \times 4}$</th>
<th>$M_{\text{delib}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLEU</td>
<td>29.97</td>
<td>30.40</td>
<td>30.76</td>
<td><strong>31.67</strong></td>
</tr>
</tbody>
</table>

### Table 2: BLEU scores of Zh→En translation

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>NIST04</th>
<th>NIST05</th>
<th>NIST06</th>
<th>NIST08</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_{\text{base}}$</td>
<td>34.96</td>
<td>34.57</td>
<td>32.74</td>
<td>26.21</td>
</tr>
<tr>
<td>$M_{\text{delib}}$</td>
<td>36.90</td>
<td>35.57</td>
<td>33.90</td>
<td>27.13</td>
</tr>
</tbody>
</table>
### Source Sentence
Aiji shuo, zhongdong heping xieyi yuqi jiang you yige xinde jiagou.

### Reference Translation
Egypt says a new framework is expected to come into being for the Middle East peace agreement.

### Translation Base Model
egypt’s middle east peace agreement is expected to have a new framework, he said.

### First-pass decoder output
egypt’s middle east peace agreement is expected to have a new framework, egypt said.

### Second-pass decoder Output
egypt says the middle east peace agreement is expected to have a new framework
Deep Model

- Deep LSTM model
- Only on En-Fr translation task
- Apply BPE techniques: split training sentences in sub-word units
- Restrict source and target sentence lengths within 64 subwords
- Encoder and Decoders are 4-layer LSTMs with residual connections
<table>
<thead>
<tr>
<th>System</th>
<th>Configurations</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>GNMT [31]</td>
<td>Stacked LSTM (8-layer encoder + 8 layer decoder) + RL finetune</td>
<td>39.92</td>
</tr>
<tr>
<td>FairSeq [4]</td>
<td>Convolution (15-layer) encoder and (15-layer) decoder</td>
<td>40.51</td>
</tr>
<tr>
<td>Transformer [26]</td>
<td>Self-Attention + 6-layer encoder + 6-layer decoder</td>
<td>41.0</td>
</tr>
<tr>
<td><em>this work</em></td>
<td>Stack LSTM (4-layer encoder and 4-layer decoder)</td>
<td>39.51</td>
</tr>
<tr>
<td></td>
<td>Stack 4-layer NMT + Dual Learning</td>
<td>40.53</td>
</tr>
<tr>
<td></td>
<td>Stack 4-layer NMT + Dual Learning + Deliberation Network</td>
<td><strong>41.50</strong></td>
</tr>
</tbody>
</table>
### Text Summarization Results

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>ROUGE-1</th>
<th>ROUGE-2</th>
<th>ROUGE-L</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_{\text{base}}$</td>
<td>27.45</td>
<td>10.51</td>
<td>26.07</td>
</tr>
<tr>
<td>$M_{\text{dec \times 2}}$</td>
<td>27.93</td>
<td>11.09</td>
<td>26.50</td>
</tr>
<tr>
<td>$M_{\text{reviewer \times 4}}$</td>
<td>28.26</td>
<td>11.25</td>
<td>27.28</td>
</tr>
<tr>
<td>$M_{\text{delib}}$</td>
<td>30.90</td>
<td>12.21</td>
<td>29.09</td>
</tr>
</tbody>
</table>