Neural Machine Translation with Source Dependency Representation

Kehai Chen¹, Rui Wang², Masao Utiyama², Lemao Liu³, Akihiro Tamura⁴, Eiichiro Sumita² and Tiejun Zhao¹

¹Harbin Institute of Technology, China
²National Institute of Information and Communications Technology, Japan
³Tencent AI Lab, China
⁴Ehime University, Japan
Overview

- Traditional NMT Model

\[ \text{Src: } x_1, x_2, x_3, x_4, x_5, x_6, x_7 \]

\[ \text{Trg: } y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8 \]

Standard NMT model
Overview

Our proposed NMT model

Inspired by the syntax knowledge in SMT, we want to explicitly integrate source dependency information into NMT
Related Work

- NMT with source syntax information
  
  - Tree2seq (Eriguchi et al., 2016; Li et al., 2017; +other)
    Tree-based neural network is used to encode source phrase structures
  
  - Extending source inputs with syntax labels (Sennrich et al., 2016; Chen et al., 2017; +other)
    Dependency labels are concatenated to source word
Related Work

• NMT with source syntax information
  - Tree2seq (Eriguchi et al., 2016; Li et al., 2017; +other)
    Tree-based neural network is used to encode source phrase structures
  - Extending source inputs with syntax labels (Sennrich et al., 2016; Chen et al., 2017; +other)
    Dependency labels are concatenated to source word

• Our work
  - A compromise between the two kinds of works
  - A novel double context approach to utilizing source dependency constraints
Source Dependency Representation (SDR)

- Extracting a dependency unit for each source word to capture source long-distance dependency constraints:

\[ U_j = \langle PA_{x_j}, SI_{x_j}, CH_{x_j} \rangle \]
Source Dependency Representation (SDR)

Extracting a dependency unit for each source word to capture source long-distance dependency constraints:

\[ U_j = \langle PA_{x_j}, SI_{x_j}, CH_{x_j} \rangle \]

Where \( PA_{x_j}, SI_{x_j}, \) and \( CH_{x_j} \) denote the parent, siblings and children words of source word \( x_j \) in a dependency structure.

Take \( x_2 \) as an example:

\[ PA_{x_2} = \langle x_3 \rangle, \quad SI_{x_2} = \langle x_1, x_4, x_7 \rangle, \quad CH_{x_2} = \langle \epsilon \rangle, \]

then,

\[ U_2 = \langle x_3, x_1, x_4, x_7, \epsilon \rangle \]
Source Dependency Representation (SDR)

- Learn semantic representation of each dependency unit

  Take $x_2$ as an example: $PA_{x_2} = \langle x_3 \rangle$, then, $U_2 = \langle x_3, x_1, x_4, x_7, \epsilon \rangle$

  $SI_{x_2} = \langle x_1, x_4, x_7 \rangle$, 

  $CH_{x_2} = \langle \epsilon \rangle$, 

\[ \begin{array}{c}
\text{Input layer} \\
\begin{array}{c}
\text{x}_3 \\
\text{x}_1 \\
\text{x}_4 \\
\text{x}_7 \\
\text{\epsilon} \\
\end{array}
\end{array} \text{ Convolution layer 1 } \\
\begin{array}{c}
\begin{array}{c}
\text{3}\times d \text{ kernel} \\
\text{3}\times d \text{ kernel} \\
\text{3}\times d \text{ kernel} \\
\text{3}\times d \text{ kernel} \\
\text{3}\times d \text{ kernel} \\
\end{array}
\end{array} \\
\begin{array}{c}
\text{Max-pooling layer 1} \\
\begin{array}{c}
\text{10}\times d \\
\text{8}\times d \\
\text{4}\times d \\
\text{2}\times d \\
\text{1}\times d \\
\end{array}
\end{array} \text{ Convolution layer 2 } \\
\begin{array}{c}
\begin{array}{c}
\text{Max-pooling layer 2} \\
\text{Output layer} \\
\end{array}
\end{array} \text{ V}_{U2} \]
Neural Machine Translation with SDR

**SDRNMT-1:**

```
| Src | x_1 | x_2 | x_3 | x_4 | x_5 | x_6 | x_7 |
|-----|-----|-----|-----|-----|-----|-----|-----|
```

**Dep Tuples**

```
| Encoder |
|---------|
| V_{x_1} | V_{x_2} | ... | V_{x_J} |
| h_1     | h_2     | ... | h_J     |
```

**Decoder**

```
| s_i     |
|---------|
| c_i     |
| s_{i-1} |
| y_{i-1} |
| y_i     |
```

---

9
Neural Machine Translation with SDR

**SDRNMT-1:**

Where the $v_{x_j}$ is 360-dim and the learned $v_{U_j}$ is 260-dim.
Neural Machine Translation with SDR

SDRNMT-2:

\[
\begin{align*}
\text{Encoder} & : \quad V_{x_1} \xrightarrow{\text{CNN}} V_{U_1} \xrightarrow{} h_1 \\
\text{Dep Tuples} & : \quad U_1 = \langle x_1, x_2, x_3, x_4, x_7, \varepsilon \rangle \\
\text{Src dep} & : \quad x_1, x_2, x_3, x_4, x_5, x_6, x_7
\end{align*}
\]
Neural Machine Translation with SDR

**SDRNMT-2:**

Encoder: 
\[ h_j = f_{enc}(V_{x_j}, h_{j-1}), \]
\[ d_j = f_{enc}(V_{U_j}, d_{j-1}) \]
Neural Machine Translation with SDR

SDRNMT-2:

Encoder: $h_j = f_{enc}(V_{x_j}, h_{j-1})$

$d_j = f_{enc}(V_{U_j}, d_{j-1})$

Attention: $e_{i,j}^s = f(s_{i-1}^s + h_j)$,

$e_{i,j}^d = f(s_{i-1}^d + d_j)$.

$$\alpha_{i,j} = \frac{\exp(\lambda e_{i,j}^s + (1-\lambda)e_{i,j}^d)}{\sum_{j=1}^{J} \exp(\lambda e_{i,j}^s + (1-\lambda)e_{i,j}^d)}$$
**SDRNMT-2:**

**Encoder:**

\[ h_j = f_{enc}(V_{x_j}, h_{j-1}), \]

\[ d_j = f_{enc}(V_{U_j}, d_{j-1}) \]

**Attention:**

\[ e_{i,j}^s = f(s_{i-1}^s + h_j), \]

\[ e_{i,j}^d = f(s_{i-1}^d + d_j). \]

\[ \alpha_{i,j} = \frac{\exp(\lambda e_{i,j}^s + (1-\lambda)e_{i,j}^d)}{\sum_{j=1}^{J} \exp(\lambda e_{i,j}^s + (1-\lambda)e_{i,j}^d)} \]

**Decoder:**

\[ c_{i,j}^s = \sum_{j=1}^{J} \alpha_{i,j} h_j, c_{i,j}^d = \sum_{j=1}^{J} \alpha_{i,j} d_j \]

\[ s_{i,j}^s = \varphi(s_{i-1}^s, y_{i-1}, c_{i,j}^s), \]

\[ s_{i,j}^d = \varphi(s_{i-1}^d, y_{i-1}, c_{i,j}^d). \]
Neural Machine Translation with SDR

SDRNMT-2:

Encoder: \[ h_j = f_{enc}(V_{x_j}, h_{j-1}), \]
\[ d_j = f_{enc}(V_{U_j}, d_{j-1}) \]

Attention: \[ e_{i,j}^s = f(s_{i-1}^s + h_j), \]
\[ e_{i,j}^d = f(s_{i-1}^d + d_j). \]

\[ \alpha_{i,j} = \frac{\exp(\lambda e_{i,j}^s + (1-\lambda)e_{i,j}^d)}{\sum_{j=1}^J \exp(\lambda e_{i,j}^s + (1-\lambda)e_{i,j}^d)} \]

Decoder: \[ c_{i,j}^s = \sum_{j=1}^J \alpha_{i,j} h_j, c_{i,j}^d = \sum_{j=1}^J \alpha_{i,j} d_j \]
\[ s_{i,j}^s = \varphi(s_{i-1}^s, y_{i-1}, c_{i,j}^s), \]
\[ s_{i,j}^d = \varphi(s_{i-1}^d, y_{i-1}, c_{i,j}^d). \]

\[ p(y_i | y_{i-1}, x, T) = g(y_{i-1}, s_{i,j}^s, s_{i,j}^d, c_{i,j}, c_{i,j}^d) \]
Neural Machine Translation with SDR

**SDRNMT-2:**

**Encoder:**
\[ h_j = f_{\text{enc}}(V_{x_j}, h_{j-1}), \]
\[ d_j = f_{\text{enc}}(V_{U_j}, d_{j-1}) \]

**Attention:**
\[ e_{i,j}^s = f(s_{i-1}^s + h_j), \]
\[ e_{i,j}^d = f(s_{i-1}^d + d_j). \]

**Decoder:**
\[ c_{i}^s = \sum_{j=1}^{J} \alpha_{i,j} h_j, c_{i}^d = \sum_{j=1}^{J} \alpha_{i,j} d_j \]
\[ s_{i}^s = \varphi(s_{i-1}^s, y_{i-1}, c_{i}^s), \]
\[ s_{i}^d = \varphi(s_{i-1}^d, y_{i-1}, c_{i}^d). \]

**Double Context NMT**
Experimental

- Experiments on Chinese-to-English translation task, 1.42M LDC corpus
- Parse source sentences of training data by Stanford Parser (Chang et al., 2009)
- For the SDRNMT-1 and SDRNMT-2, the dimension of $V_{xj}$ is 360 and the dimension of $V_{uj}$ is 260, and input embedding of the baseline is 620
- The baselines include Phrase-Based Statistical Machine Translation (PBSMT) (Koehn et al., 2007), standard Attentional NMT (AttNMT) (Bahdanau et al., 2014), NMT with dependency labels (Sennrich and Haddow, 2016)
### Experimental

| System          | Dev(NIST02) | NIST03 | NIST04 | NIST05 | NIST06 | NIST08 | AVG  |
|-----------------|-------------|--------|--------|--------|--------|--------|------|
| PBSMT           | 33.15       | 31.02  | 33.78  | 30.33  | 29.62  | 23.53  | 29.66|
| AttNMT          | 36.31       | 34.02  | 37.11  | 32.86  | 32.54  | 25.44  | 32.40|
| Sennrich-deponly| 36.68       | 34.51  | 38.09  | 33.37  | 32.96  | 26.96  | 32.98|
| SDRNMT-1        | 36.88       | 34.98* | 38.14  | 34.61**| 33.58* | 27.06  | 33.32|
| SDRNMT-2        | **37.34**   | **35.91** | **38.73** | **34.18** | **33.76** | **27.64** | **34.04** |

“*” indicates statistically significant better than “Sennrich-deponly” at $p$-value < 0.05 and “**” at $p$-value < 0.01 by bootstrap resampling (Koehn, 2004)
Experimental Results

- Translation qualities for different sentence lengths

![Graph showing BLEU scores for different sentence lengths with comparisons between PBSMT, AttNMT, Sennrich-deponly, SDSNMT-1, SDSNMT-2.](image-url)
Conclusion

- Source dependency unit to capture source long-distance dependency constraint
- The proposed $SDRNMT-1$ and $SDRNMT-2$ consist of NMT and CNN, which are jointly trained to learn SDR and translation instead of separately trained
- Double-Context approach to further utilize source dependency representation