Neural Machine Translation by Jointly Learning to Align and Translate

D. Bahdanau, K. Cho, Y. Bengio (ICLR 2015)

KAMRAN ALIPOUR
APRIL 2019

OUTLINE

- PROBLEM:
  - Fixed-length vector representation is a bottleneck
  - Difficult to cope with long sentences, especially when longer than the sentences in the training corpus.
Previous work shows performance of a basic encoder–decoder deteriorates rapidly as the length of an input sentence increases.

**SOLUTION:**
- An extension which learns to **align** and **translate** jointly.
- Does NOT encode a whole input sentence into a single fixed-length vector.
OUTLINE

● SOLUTION:
  ○ Encode the input into a sequence of vectors and choose a subset of them adaptively while decoding.

OUTLINE

● SOLUTION:
  ○ For each translation, it (soft-)searches for a set of positions in a source sentence where the most relevant information is concentrated.
**OUTLINE**

- **SOLUTION:**
  - The model predicts a target word based on the context vectors associated with source positions and all the previously generated target words.

**OUTLINE**

- **SOLUTION:**
  - The model does not squash all the information of a source sentence, regardless of its length, into a fixed-length vector.
OUTLINE

- SOLUTION:
  - State-of-the-art phrase-based system on the task of English-to-French translation.

PREVIOUS WORK

Phrase-based translation system
- Tuning sub-components separately (e.g. Koehn et al. 2003)

Neural machine translation
- Proposed by: Kalchbrenner and Blunsom (2013)
  Cho et al. (2014)
  Sutskever et al. (2014)
- Mostly encoder-decoder architectures
- Encoders and decoders for each language
- Encoder-decoder jointly trained
BACKGROUND

NEURAL MACHINE TRANSLATION

\[ \arg \max_y p(y|x) \]

*Cho et al. (2014)* & *Sutskever et al. (2014)*

RNN Encoder-Decoder

\[ p(y_1, y_2, \ldots, y_{T_y} | x_1, x_2, \ldots, x_{T_x}) \]

RNN ENCODER-DECODER

Introduced by: *Cho et al. (2014)*

\[ x = (x_1, \ldots, x_{T_x}) \]

\[ h_t = f(x_t, h_{t-1}) \quad h_t \in \mathbb{R}^n \]

\[ c = q(\{h_1, \ldots, h_{T_x}\}) \]
RNN ENCODER-DECODER

Encoder:

\[ h_t = f(x_t, h_{t-1}) \]

\[ X_1 \rightarrow h_1 \]
\[ X_2 \rightarrow h_2 \]
\[ X_T \rightarrow h_T \]

\[ c = q(\{h_1, ..., h_{T_x}\}) \]

Sutskever et al. (2014):

LSTM as \( f(x, h) \)

\[ q(\{h_1, ..., h_T\}) = h_T \]

RNN ENCODER-DECODER

Decoder:

\[ h_t = f(h_{t-1}, y_{t-1}, c) \]

\[ h_1 \rightarrow Y_1 \]
\[ h_2 \rightarrow Y_2 \]
\[ h_T \rightarrow Y_T \]

\[ Y = (y_1, ..., y_{T_y}) \]

\[ p(y) = \prod_{t=1}^{T} p(y_t|\{y_1, ..., y_{t-1}\}, c) \]

\[ p(y_t|\{y_1, ..., y_{t-1}\}, c) = g(h_t, y_{t-1}, c) \]
**RNN Encoder-Decoder**

Encoder-decoder jointly trained to maximize the conditional log-likelihood:

\[
\max_{\theta} \frac{1}{N} \sum_{n=1}^{N} \log p_{\theta}(y_n | x_n)
\]

- Better captures the linguistic regularities in the phrase table
- Indirectly explains the quantitative improvements in the overall translation performance
- Learns a continuous space representation of a phrase that preserves both the semantic and syntactic structure of the phrase.
NEW ARCHITECTURE

Encoder:
- Bidirectional (BiRNN, Schuster and Paliwal, 1997)
- Annotation of each word summarizes not only the preceding words, but also the following words.

ENCODER: BIDIRECTIONAL RNN FOR ANNOTATING SEQUENCES

BiRNN:
- Introduced by Schuster and Paliwal, 1997
- Also successfully used for speech recognition (Graves et al., 2013)
- Consists of a forward and a backward RNN
**ALIGN AND TRANSLATE**

\[ a_j = [a_j^T; \bar{a}_j^T]^T \]

\[ h_i = f(h_{i-1}, y_{i-1}, c_i) \]

\[ c_i = \sum_{j=1}^{T_i} \alpha_{ij} a_j \]

**Annotations:**

\[ (a_1, ..., a_{T_x}) \]

\[ c_i = \sum_{j=1}^{T_x} \alpha_{ij} a_j \]

\[ h_i = f(h_{i-1}, y_{i-1}, c_i) \]

\[ p(y_i|y_1, ..., y_{i-1}, X) = g(y_{i-1}, h_i, c_i) \]
Align and Translate

Decoder:
- Searching through source sentence while decoding a translation

Alignment Model

\[ \alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} \]

Alignment model:
\[ e_{ij} = a(h_{i-1}, a_j) = \nu_a^T \tanh(W_a h_{i-1} + U_a a_j) \]

\[ W_a \in \mathbb{R}^{n \times n}, \quad U_a \in \mathbb{R}^{n \times 2n}, \quad \nu_a \in \mathbb{R}^n \]
Alignment Model

Notes about alignment model:

- Parameterized as a feedforward neural network
- Trained jointly with all the other components of the proposed model
- Alignment is NOT a latent variable
- Directly computes a soft alignment
- The gradient of the cost function can backpropagate through it

ALIGNMENT MODEL

\[ C_i = \sum_{j=1}^{T_x} \alpha_{ij} a_j \]

"Expected Annotation"

\( \alpha_{ij} \) is the probability that the target word \( y_i \) is aligned to, or translated from, a source word \( x_j \)
ALIGNMENT MODEL

Attention based decoder
  ● Decoder decides which parts of sentence to pay attention to.
  ● Encoder relieved from encoding all the information in a fixed-length vector.

EXPERIMENT SETTINGS

Task: English to French translation

Bilingual, parallel corpora provided by ACL WMT ’14

Results compared with original RNN Encoder-Decoder by Cho et al. ’14
**Experiment Settings**

Dataset: WMT '14

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>Europarl</td>
<td>61M</td>
</tr>
<tr>
<td>News commentary</td>
<td>5.5M</td>
</tr>
<tr>
<td>UN</td>
<td>421M</td>
</tr>
<tr>
<td>Crawled corpora</td>
<td>90M + 272.5M</td>
</tr>
<tr>
<td>Total</td>
<td>850M</td>
</tr>
</tbody>
</table>

Selected Data: 384M words

Using data selection method by Axelrod et al. 2011
## EXPERIMENT SETTINGS

### Training

<table>
<thead>
<tr>
<th>Validation</th>
<th>News-test-2012 + news-test-2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Set</td>
<td>News-test-2014 from WMT '14</td>
</tr>
</tbody>
</table>

### Training Models:
- RNN Encoder-Decoder
- RNNsearch

Each model trained twice:

<table>
<thead>
<tr>
<th>Model</th>
<th>Type</th>
<th>Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNNencdec-30</td>
<td>RNN Encoder-Decoder</td>
<td>30</td>
</tr>
<tr>
<td>RNNencdec-50</td>
<td>RNN Encoder-Decoder</td>
<td>50</td>
</tr>
<tr>
<td>RNNsearch-30</td>
<td>RNNsearch</td>
<td>30</td>
</tr>
<tr>
<td>RNNsearch-50</td>
<td>RNNsearch</td>
<td>50</td>
</tr>
</tbody>
</table>
**Experiment Settings**

- 1000 hidden units in both models
- A multilayer network with a single maxout hidden layer (Goodfellow et al. '13)
- Using a mini-batch stochastic gradient descent (SGD) algorithm together with Adadelta (Zeiler 2012)
- Training for each model: 5 days

**Results**

<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>
RESULTS

Strength of sof-alignments:

- Understands change of order
- Looks at close words for better translation
- Naturally deals with different lengths of source and target

\[ c_i = \sum_{j=1}^{T_i} \alpha_{ij} \alpha_{ji} \]
Conclusion

- Use of a fixed-length context vector is problematic for translating long sentences
- Novel architecture that models a (soft-)search for a set of input words, or their annotations computed by an encoder, when generating each target word
- Better results on long sentences
- All components (including annotations) are jointly trained
- Performance comparable to the existing phrase-based statistical machine translation