Student ID : 20194293

Name : Go, Kyeong Ryeol

[AI 502] Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation

1. Paper Summary

Neural network is enlarging its applicability to various tasks in natural language processing (NLP). Especially, many experimental trials have done in the field of the statistical machine translation (SMT) to improve the performance of phrase-based method. To struggle this problem, this paper comes up with a novel neural network architecture motivated by Recurrent Neural Network (RNN) structure that can be used as a part of the conventional methods.

Comparing to other neural network architectures, RNN has a strength for handling sequence of data as the hidden units $\mathrm{h}_{<\mathrm{t}>} = f(h_{<t-1>}, x_t)$ are updated along the steps and it can learn a probability distribution $\mathrm{p(x)} = \prod_{t=1}^{T} p(x_t | x_{t-1}, \ldots, x_1)$ over the sequence which can be utilized in predicting the next symbol by sampling. In case of SMT, this sampling procedure has to be done after all the source sequence is read. So, the whole network should be comprised of 2 steps which are the encoding parts and the decoding parts. Therefore, the author proposed 'RNN-Encoder-Decoder' that maps a variable length source sentence x to a fixed length vector c by encoder RNN and maps the vector representation c back to a variable length target sequence y by decoder RNN. Here, unlike the vanilla RNN, both the hidden units $\mathrm{h}_{<\mathrm{t}>} = f(h_{<t-1>}, y_{t-1}, c)$ and outputs $\mathrm{p(y)} = \prod_{t=1}^{T} \mathrm{p}(\mathrm{y}_\mathrm{t} | y_{t-1}, \ldots, y_1, c)$ are also conditioned on previous step's output and the summary of the source sentence. Then, the two components are jointly trained to maximize the conditional log-likelihood $\frac{1}{\mathrm{N}} \sum_{n=1}^{N} \log p_\theta (y_n | x_n)$.

Furthermore, a sophisticated hidden unit that named as 'GRU' was introduced for memory efficient and easy training. Comparing to LSTM, it has an advantage as it is simpler to compute and implement due to the smaller number of gating units. The reset gate $\mathrm{r}_\mathrm{j} = \sigma([W_r x]_j + [U_r h_{<t-1>}]_j$ and the update gate $\mathrm{z}_\mathrm{j} = \sigma([W_z x]_j + [U_z h_{<t-1>}]_j$ are the only gating units in GRU that handle the short-term and the long-term dependency respectively. Then the actual activation of hidden unit can be computed as the follows; $\mathrm{h}_\mathrm{j}^{<t>} = z_j h_j^{<\mathrm{t}-1>} + (1 - z_j) \widetilde{h_t^{<t>}}\ where\ \widetilde{h_t^{<t>}} = \emptyset([Wx]_j + [U(r \odot h_{<t-1>})]_j)$. This enables each hidden unit to capture dependencies over different time scales as the different reset/update gates are assigned to each hidden unit.

The validity of the model is verified through the task of translating from English to French with WMT'14 dataset. The quantitative analysis compared Bilingual Evaluation Understudy (BLEU) score with baseline model from Moses. The best model was CSLM+RNN which achieves 31.48 BLEU score that significantly outperforms that of baseline. In the qualitative analysis for overall translation performance, it was shown that the proposed model is better at capturing the linguistic regularities in the phrase table than the conventional model. Finally, the embedded words were projected into 2D space and revealed that the semantic and syntactic structure of the phrase is learned as a continuous space representation.

2. Discussion

Here, I want to offer two discussion points. To begin with, how can the computational inefficiency in RNN structure be further alleviated? According to Figure 1 in the paper, a lot of parameters are added to incorporate the summary of source sentence in the target output. This will be more critical when the length of the sentence gets longer and longer. One suggestion is to devise a nice nonlinear transformation from the fixed length summary to the first hidden unit of the decoder so that the summary no more needs to be connected to the hidden units and the outputs at the further steps. Next, how can the parameters of the decoder be correlated to that of encoders? Even if they are dealing with different languages, there are somethings in common between them as some components are essentially required when writing a sentence. If we can correlate the parameters of the two separate network, it will significantly reduce the number of parameters or at least it can be used as an initialization.