Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation

1 Citation

Cho, Kyunghyun, et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." arXiv preprint arXiv:1406.1078 (2014).

`https://arxiv.org/pdf/1406.1078.pdf`

2 Abstract

In translation, you must turn an input sequence in one language into an output sequence in another language. To do this, we've created a neural network where an Encoder Recurrent Neural Network (RNN) turns the input sequence into a fixed length context and then we feed that context into a Decoder RNN that produces the output sequence.

3 Introduction

Statistical Machine Translation (SMT) is a system that can translate from one language into another (we consider English to French). Part of the system involves scoring phrase pairs. We use an Encoder-Decoder neural network to do the scoring. We also have a sophisticated hidden unit (based on LSTM) that we use in our network. The context learned by the Encoder-Decoder preserves semantic and syntactic structure of the phrase and leads to improved SMT performance.

4 RNN Encoder Decoder

Given input words $\mathbf{x} = [x_1, x_2, ..., x_T]$, we compute hidden state

$$\mathbf{h}_t = f(\mathbf{h}_{t-1}, x_t)$$

We then set $\mathbf{c} = h_T$ as our context. Now, the decoder network produces and output sequence $\mathbf{y} = [y_1, ..., y_{T'}]$ where:

$$\mathbf{h}_t = f(\mathbf{h}_{t-1}, y_{t-1}, \mathbf{c})$$

$$p(y_t|y_{t-1}, ..., y_1, \mathbf{c}) = g(\mathbf{h}_t, y_{t-1}, \mathbf{c})$$

where $\mathbf{h}$ and f are different than in the encoder RNN.

We can then train this network end-to-end to maximize log likelihood:

$$\max_{\theta} \frac{1}{N} \sum_{n=1}^{N} \log p_\theta(\mathbf{y}_n, \mathbf{x}_n)$$

For f in the equations above, we use the following hidden unit, which is based on the Long Short-Term Memory (LSTM) unit. We start with a reset gate:

$$r_j = \sigma([\mathbf{W}_r \mathbf{x}]_j + [\mathbf{U}_r \mathbf{h}_{t-1}]_j)$$

Then we compute an update gate:

$$z_j = \sigma([\mathbf{W}_z \mathbf{x}]_j + [\mathbf{U}_z \mathbf{h}_{t-1}]_j)$$

Now we compute the activation of the hidden unit:

$$\tilde{h}_j^t = \phi([\mathbf{W}\mathbf{x}]_j + [\mathbf{U}(\mathbf{r} \odot \mathbf{h}_{t-1})]_j)$$

$$h_j^t = z_j h_j^{t-1} + (1 - z_j)\tilde{h}_j^t$$

We also tried a tanh unit, but did not get gains.

5 Statistical Machine Translation

An SMT system figures out the most probable translation $\mathbf{f}$ of an input sequence $\mathbf{e}$ using:

$$p(\mathbf{f}|\mathbf{e}) = \sum_{n=1}^{N} w_n f_n(\mathbf{f}, \mathbf{e}) + \log Z(e)$$

where w_n denotes weights and f_n denotes features. Our metric is BLEU score. In the SMT system, we need to assign scores to phrase pairs. We use our neural network for this purpose.

When training, we sample from a uniform distribution over phrase pairs. We don't sample by phrase frequency to avoid training the network to just remember the most common phrases. In theory, you can replace the generation of the phrase table with our neural network, but we don't do that in this paper.

Other papers have used feedforward neural nets to score phrases, but they require fixed size phrases. One paper embeds words in both languages into the same space and then maps one phrase to another by finding the closest phrase in the other language. There are other RNN-based approaches.

6 Experiments

We do English to French translation. We use multiple datasets, but use a data selection technique to pick a 418M word corpus from a 2B one. Historically, training on the entire corpus doesn't give best performance. Our vocabulary has 15K words and the rest are treated as "unknown" symbols.

Our neural net has 1000 hidden units and matrix multiplication is approximated with two low-rank (100) matrices instead of one big matrix. The activation function in $\tilde{h}$ is tanh. The output of the decoder goes through a layer with 500 maxout units (each pooling 2 inputs). Non-recurrent weights are sampled from

Gaussians, but recurrent weights come from a white Gaussian's left singular vectors matrix. We train with Adadelta with a batch size of 64 for three days.

We also tried scoring phrase pairs with a CLSM language model. We find that we get best performance when we ensemble the CLSM and our neural network, indicating that they are complementary. If you must pick one, our neural network is better.

Looking at some phrase pairs, our neural network tends to go for literal translations, prefers shorter phrases, and doesn't just memorize the most frequent phrases. When you sample from the network, you get phrases that are not in the phrase table, indicating this might a good way to generate the phrase table as well.

If we look at the phrases in the learned embedding space, we find that our neural network learns semantic and syntactic similarities.

7 Conclusion

Our Encoder-Decoder structure turns one variable length sequence into another. We also have a new hidden unit, based on the LSTM, to use with this network. Using our network to score phrases increases BLEU score. It also seems to generate good phrases, so maybe you can use it to generate the phrase table.