Attentional Neural Machine Translation

Wei Fang

February 27, 2017

Model

In the assignment, the typical attentional neural machine translation model is used to perform machine translation from German to English. First, there is a bi-directional LSTM to encode the source sentence, for each time step, we have:

$$h_j^{(f)} = \left[\overrightarrow{h}_j^{(f)}; \overleftarrow{h}_j^{(f)}\right]$$

where $\overrightarrow{h}_j^{(f)}$ and $\overleftarrow{h}_j^{(f)}$ are hidden state in the j time step from left and right respectively. In the decode phase, for each time step, because I want to give more attention to those important source words and get a better context representation for the source sentence, I compute the attention score for each source word according to the hidden outputs of the current decode state and all encode states:

$$att_{t,j} = \text{attend}(h_t^{(e)}, h_j^{(f)})$$

where attend(...) is a multiple layer perceptron and $h_t^{(e)} = \text{enc}([\text{embed}(e_{t-1}); c_{t-1}], h_{t-1}^{e})$. Then I can summarize the source sentence into a context vector through a weighted sum:

$$c_t = \left[h_0^{(f)}\ h_1^{(f)}\ ...\ h_n^{(f)}\right] att_t$$

And now, I can compute the output distribution over the vocabulary of the target language:

$$p_t^{(e)} = \text{softmax}\left(W\left[h_t^{(e)}; c_t\right] + b\right)$$

Finally, I use the greedy strategy and pick the word with the largest value.

Experiment and Result

2.1 Baseline Version

In the training phase, I replace the infrequent words (appear less than 3 times) with the unknown work marker <unk>. I use SimpleSGDTrainer to train the model and mini-batch is not included.

2.2 Improved Version

In the baseline version, there are many unknown word marker in the generated result. So I first locate the most matched word in the source sentence according to the attention weights and then pick most popular target word based on the selected source word. I use IBM model 1 to extract the word alignment from the training data.

2.3 Result

The result is shown in the table 2.1 and this is the result after 6 epoches. I evaluate the BLEU score with the tool mentioned in the lecture material and evaluate the PPL with my own code based on bigram written in the first lecture. According to this result, currently I cannot catch up with the performance given by the TAs. I will continue to dig into the result and try to find the reason.

Table 2.1: Data Statistics

	BLEU	PPL
Baseline	14.75	178.34
Improved	15.64	133.32

Conclusion

In this assignment, I implement the attentional neural translation model and train the model on a German-English bilingual dataset. The result is still worse than the baseline performance and I will continue to see which part is not going well.