Project Proposal – Exploring Complex Reordering in Neural MT

Before the introduction of attention model, sequence to sequence models (Sutskever et al., 2014; Cho et al., 2014) encode the whole input sentence into a sequence of vectors and use the last one as the context vector to decode all the target words. A potential issue for this approach is the difficulty of compressing the whole sentence into a single fixed vector, especially for long sentences. The attention model (Bahdanau et al., 2015), on the other hand, keeps the whole sequence of vectors and calculates the weighted sum of the sequence adaptively (i.e. according to the previous target-side vector) to decode. The weight matrix is trained by a single layer perceptron which aims to give high scores for the relevant source-side vectors given the previous target-side vector. The intuition for this approach is to let the decoder "focus" on a subset of the source-side vectors rather than the last one.

1. Training Using Known Alignment

Since the idea of choosing a subset of source-side vectors is similar to alignment, I want to investigate the effect of training the neural net using the known alignment. Specifically, in the equation $c_i = \sum_{j=1}^{T_x} a_{ij} h_j$ (Bahdanau et al., 2015), each a_{ij} can be an output which equals to 1 if word i is aligned to word j in the known alignment. Thus, the gradients will not only back propagate from the target word, but also from the alignment. This alignment model can be obtained using SMT system or other software.

2. Using GRU for the Alignment Score

In the equation $e_{ij} = {v_a}^T \tanh(W_a s_{i-1} + U_a h_j)$ (Bahdanau et al., 2015), the alignment score only takes input of the previous encoded vector s_{i-1} and the source-side vector h_j. However, the previous alignment scores might also be useful, especially in a long sentence (i.e. more than 50 words). Therefore, I propose to use a GRU model for the alignment score, which can potentially capture more complex ordering. The equations will then be:

$$e_{ij} = \tanh(v_{i,j})$$
$$\overline{v}_{i,j} = \tanh(W s_{i-1} + U(r_i \circ v_{i-1,j}) + Q h_j)$$
$$z_{i,j} = \sigma(W_z s_{i-1} + U_z v_{i-1,j} + Q_z h_j)$$
$$r_{i,j} = \sigma(W_r s_{i-1} + U_r v_{i-1,j} + Q_r h_j)$$

3. Other Thoughts

I will first do the above 2 steps. If I finish early, I also want to investigate the possibility of using bidirectional RNN in the decoder and the effect of removing the GRU nodes h while doing step 2.

REFERENCES

Cho, K., van Merrienboer, B., Gulcehre, C., Bougares, F., Schwenk, H., and Bengio, Y. (2014a). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*. to appear.

Sutskever, I., Vinyals, O., and Le, Q. (2014). Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems (NIPS 2014)*.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations (ICLR 2015)*.