Appendix - Structured Self-Attention Weights Encode Semantics in Sentiment Analysis

Zhengxuan Wu[1], **Thanh-Son Nguyen**[2], **Desmond C. Ong**[2,3]
[1]Symbolic Systems Program, Stanford University
[2]Institute of High Performance Computing, Agency for Science, Technology and Research, Singapore
[3]Department of Information Systems and Analytics, National University of Singapore
wuzhengx@stanford.edu, Nguyen_Thanh_Son@ihpc.a-star.edu.sg,
dco@comp.nus.edu.sg

1 Evaluation Metrics

Concordance Correlation Coefficient (CCC (Lin, 1989)): The CCC of vectors X and Y is:

$$\mathrm{CCC}(X,Y) \equiv \frac{2\mathrm{Corr}\,(X,Y)\,\sigma_X\sigma_Y}{\sigma_X^2+\sigma_Y^2+\left(\mu_X-\mu_Y\right)^2} \quad (1)$$

where $\mathrm{Corr}\,(X,Y) \equiv \mathrm{cov}(X,Y)/(\sigma_X\sigma_Y)$ is the Pearson correlation coefficient, and μ and σ denotes the mean and standard deviation respectively.

2 Experiment Setup

Computing Infrastructure: To train our models, we use a single Standard NV6 instance on Microsoft Azure. The instance is equipped with a single NVIDIA Tesla M60 GPU.

Average Runtime: With the computing infrastructure, it takes about 1.5 hrs to train both models, where each model is trained with 200 epochs. Both model reaches maximum performances in about 1.5 hrs at about 100 epochs.

Number of Trainable Parameters: The model trained on SST-5 that uses a LSTM decoder has 3,993,222 parameters. Additionally, the model trained on SEND that uses a MLP decoder has 4,715,362 parameters.

3 Task-specific Decoders

Long-Short Term Memory Network (LSTM): For the time-series task, we use a LSTM layer (Hochreiter and Schmidhuber, 1997) to decode the context vector c_i from our encoder for each window i to output a hidden vector h_i. Then, the hidden vector passes through a MLP to make the valence prediction:

$$h_i = \mathrm{LSTM}(h_{i-1}, c_i) \quad (2)$$
$$\hat{r}_i = \mathrm{MLP}(h_i) \quad (3)$$

Multilayer Perceptron (MLP): Our MLP contains 3 consecutive linear layers with a single ReLU activation in between layers. For the classification task, we feed in the context vector c from our encoder to MLP to make the sentiment prediction:

$$f_1(c) = \mathrm{ReLU}(\mathbf{W}_1 c + \mathbf{b}_1) \quad (4)$$
$$\hat{r}_i = \mathbf{W}_3\left[\mathrm{ReLU}(\mathbf{W}_2 f_1(c) + \mathbf{b}_2)\right] + \mathbf{b}_3 \quad (5)$$

where $\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3, \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$ are learnable parameters for linear layers. For the time-series task, the hidden vector h_t is the input instead.

References

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Lawrence I-Kuei Lin. 1989. A concordance correlation coefficient to evaluate reproducibility. *Biometrics*, pages 255–268.