

Sentence-level Sentiment Classification with RNN

袁无为 计预0

Abstract

在这次作业中，我使用了 RNN + Self-Attention 来对句子进行情感分类。实验证明 RNN 等模型能够较好的完成这个任务。

1. Introduction

句子的情感分类问题指的是给出一个句子，要求判断这个句子表达的情感是正面的、负面的、中性的。因为句子的长度不固定，因此我使用了 RNN 这种可以处理不定长输入的结构。另外，我还加入了 Self-Attention 机制，并且与 CNN 进行了对比。

2. Related Work

TextCNN(Yoon Kim, 2014): 对于一个输入 $x_{1..n}$ ，添加多个不同的卷积核 w 和偏置 b ，在该通道上的输出为 $c = [c_1, c_2, \dots, c_{n-h+1}]$ ，其中 $c_i = f(w \cdot x_{i..i+h-1} + b)$ ，然后取 $k_max(c_i)$ 作为该通道上的输出（即每一个通道上最大的 k 个数）。这里 h 为 window 的长度。我们可以添加不同长度的多个卷积核。最后再把全部通道上的输出连接一个全连接层进行分类。

3. Approach

3.1 Basic Structure

Basic RNN

对于一组输入 (x_1, x_2, \dots, x_n) ，中间状态为 $y'_i = h_i = f(W_x x_i + W_h h_{i-1} + b)$ ，输出为 $y = Softmax(f(W(\frac{1}{n} \sum_{i=1}^n y'_i) + b))$ 。

这里对 x 和 y' 进行了 Dropout，下同。

LSTM

中间状态为

$$\begin{aligned} i_i &= Sigmoid(W_1 x_i + W_2 h_{i-1} + b_1) \\ o_i &= Sigmoid(W_3 x_i + W_4 h_{i-1} + b_2) \\ u_i &= Sigmoid(W_5 x_i + W_6 h_{i-1} + b_3) \\ f_i &= tanh(W_7 x_i + W_8 h_{i-1} + b_4) \\ c_i &= f_i \otimes c_{i-1} + i_i \otimes u_i \\ h_i &= o_i \otimes tanh(c_i) \\ y'_i &= h_i \end{aligned}$$

GRU

中间状态为

$$\begin{aligned}
r_i &= \text{Sigmoid}(W_1 x_i + W_2 h_{i-1} + b_1) \\
u_i &= \text{Sigmoid}(W_3 x_i + W_4 h_{i-1} + b_2) \\
c_i &= \tanh(W_5 x_i + W_6 (r_i \otimes h_{i-1}) + b_3) \\
h_i &= (1 - u_i) \otimes h_{i-1} + u_i \otimes c_i \\
y'_i &= h_i
\end{aligned}$$

3.2 Self Attention

对于一组 RNN 的输出 $Y' = \{y'_1, y'_2, \dots, y'_n\}$, 计算

$$\begin{aligned}
A &= \text{Softmax}(W_2 \tanh(W_1 Y'^T)) \\
M &= AY'
\end{aligned}$$

其中 W_1 是一个 $d_a \times u$ 的矩阵, W_2 是一个 $r \times d_a$ 的矩阵, u 为中间状态的长度, r 为特征数。

最后把 M 这个矩阵连上全连接层进行分类。

另外, 我们还会把 $10^{-3} \times \|(AA^T - I)\|_F^2$ 加到 Loss 上以训练 W_1 和 W_2 。

4. Experiments

4.1 Datasets

使用了给出的数据集和词向量。

4.2 Implementation Details

使用了 GD 优化方法。学习率为 0.005。

mini batch size 为 16。

CNN 的 filter 包含长度为 $[2, 3, 4, 5]$ 的各 1000 个 filter。Drop Rate 为 0.5。 $k = 5$ 。

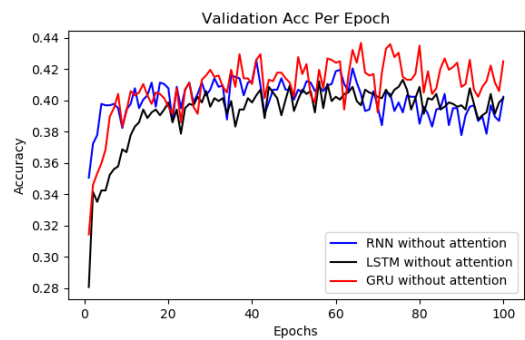
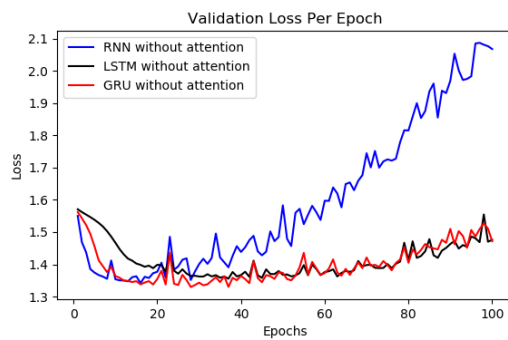
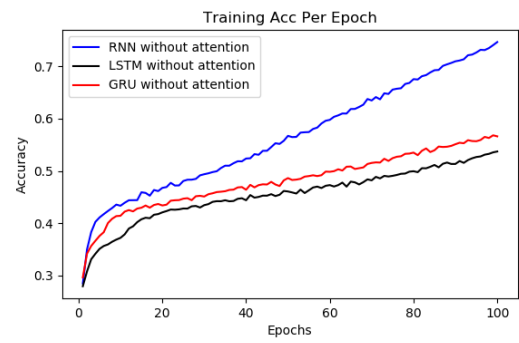
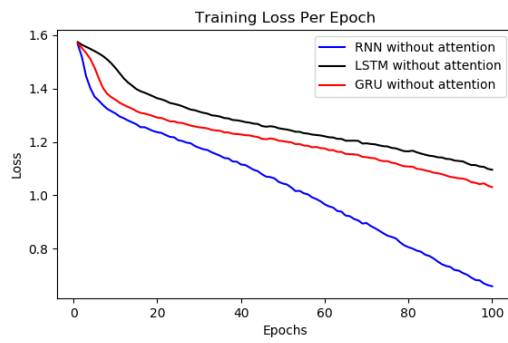
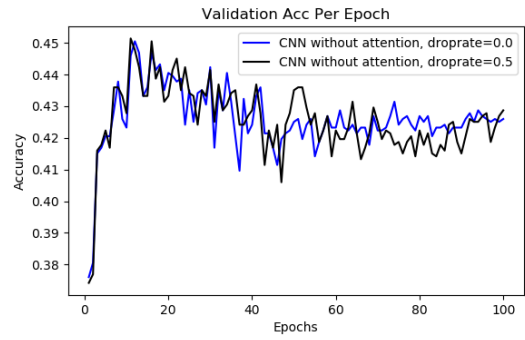
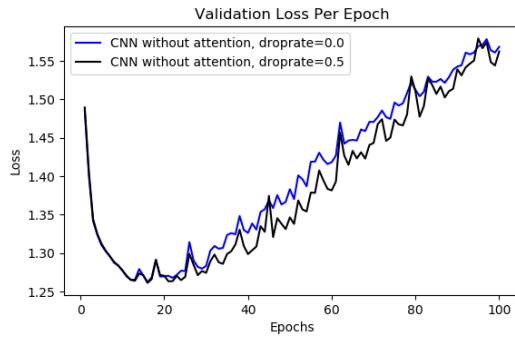
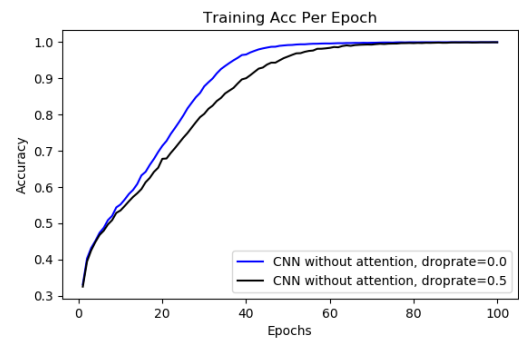
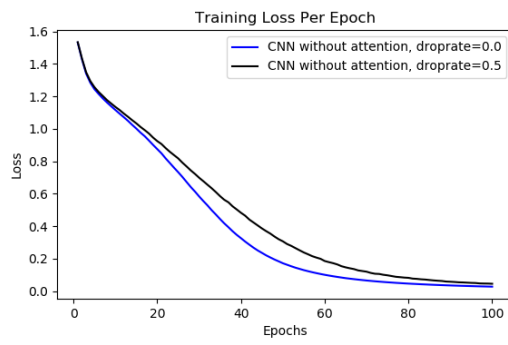
RNN cell 内部的隐藏状态大小为 512。还应用了梯度裁剪, 最大梯度为 5。Drop Rate 为 0.3。

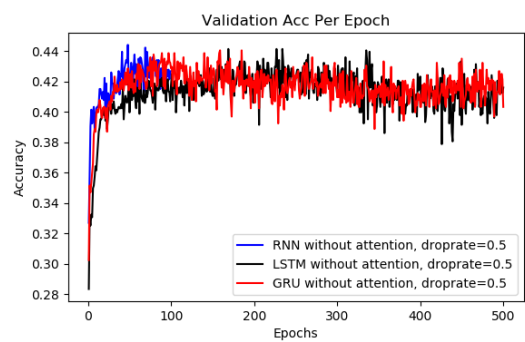
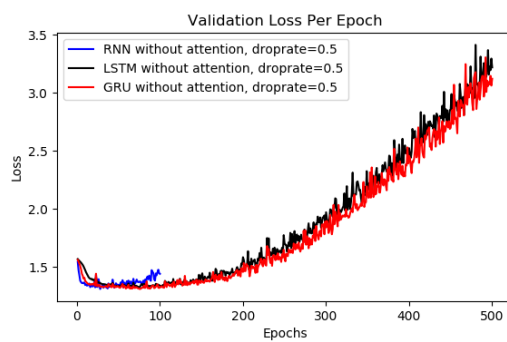
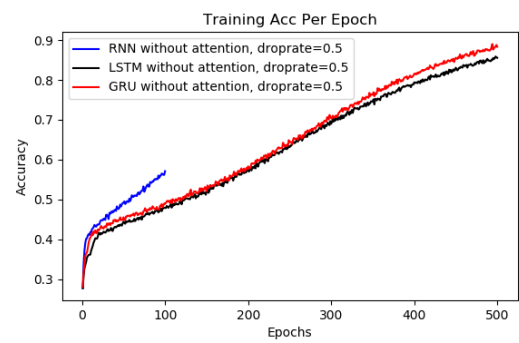
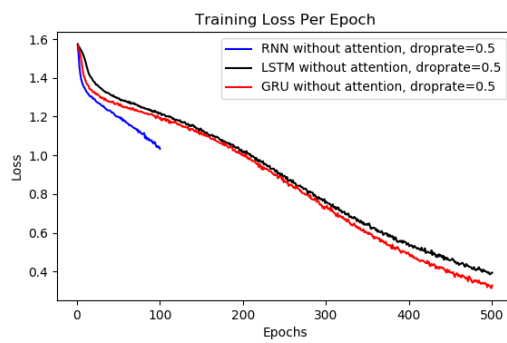
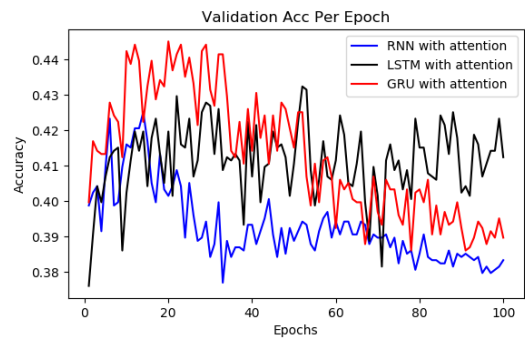
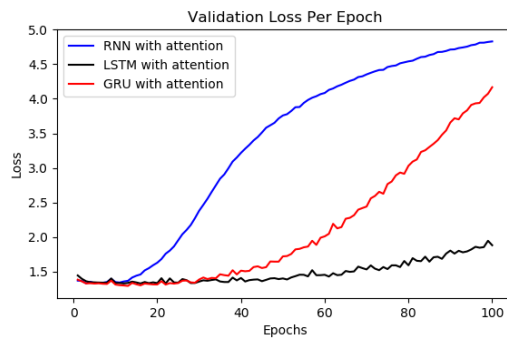
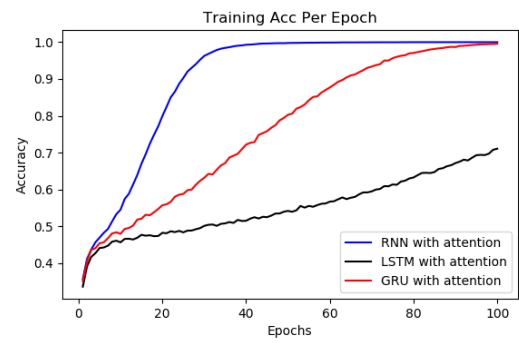
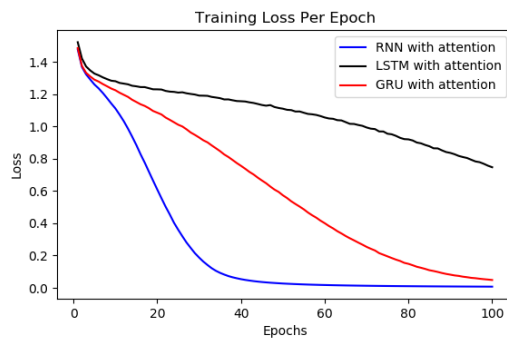
LSTM cell 的 b_4 和 GRU cell 的 b_1, b_2 初始化为全 1。其他超参数与 RNN 相同。

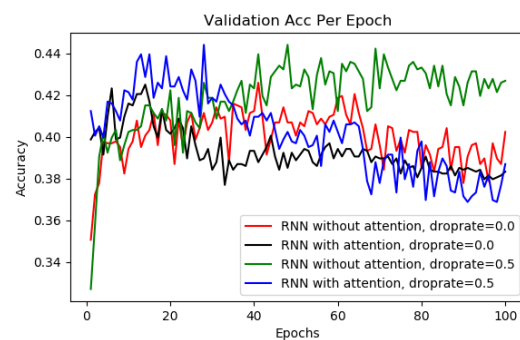
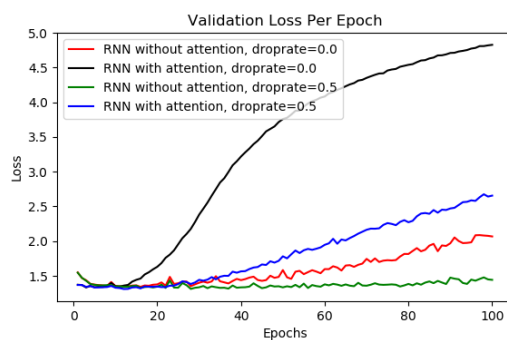
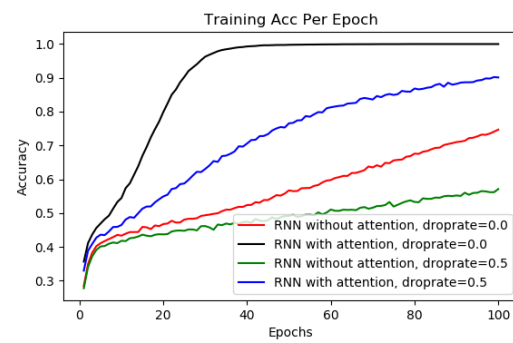
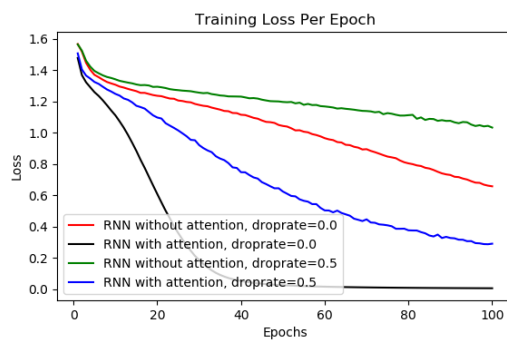
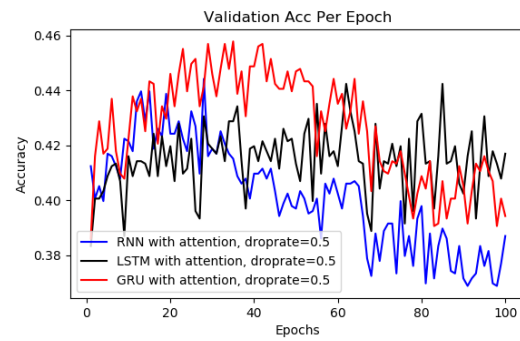
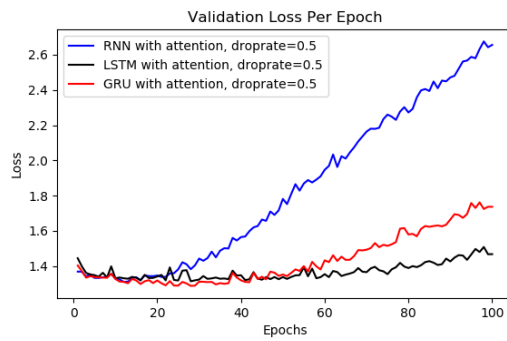
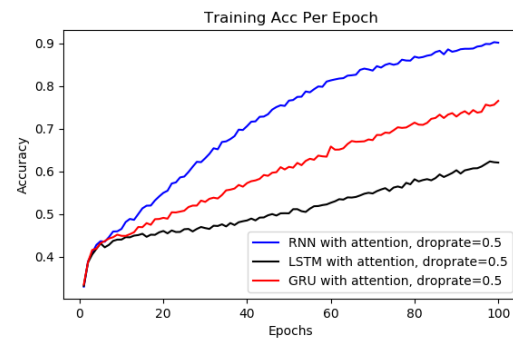
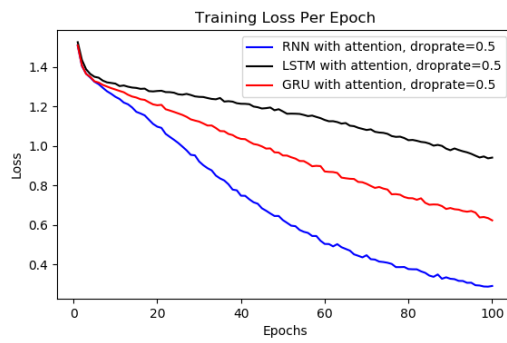
Self-Attention 的中间节点个数为 150, 特征个数为 10。

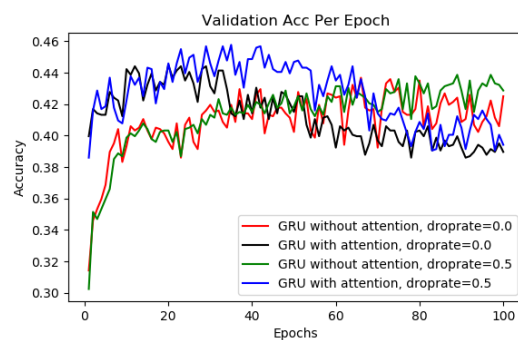
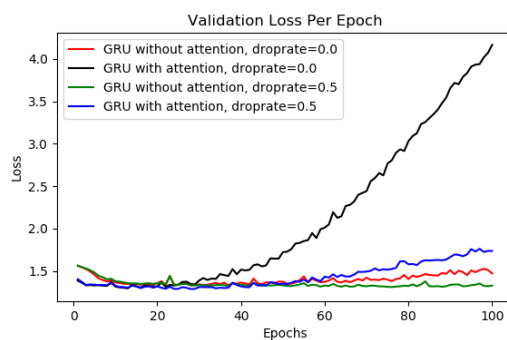
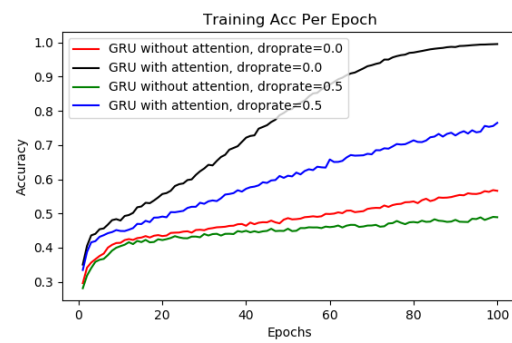
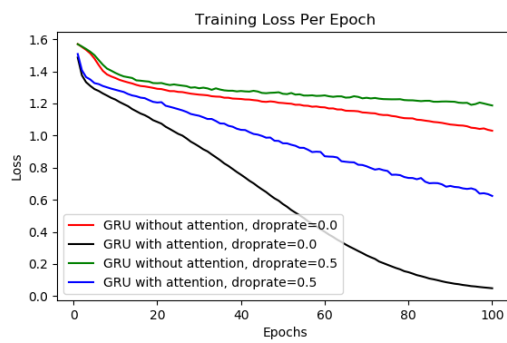
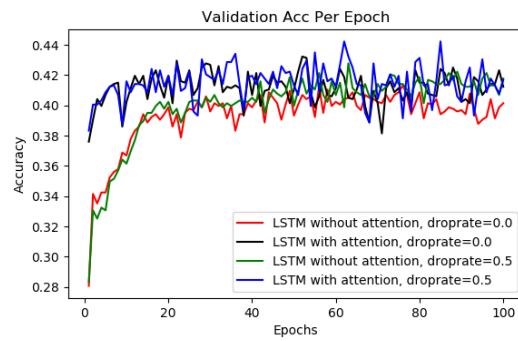
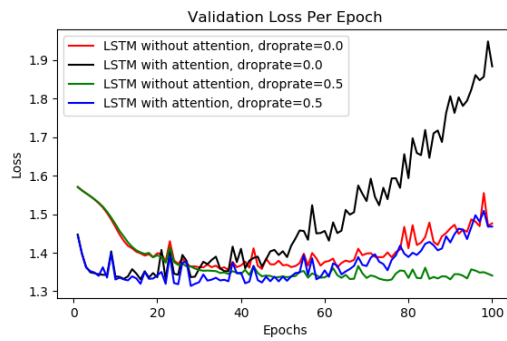
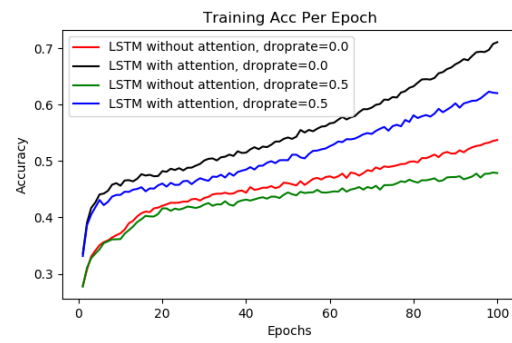
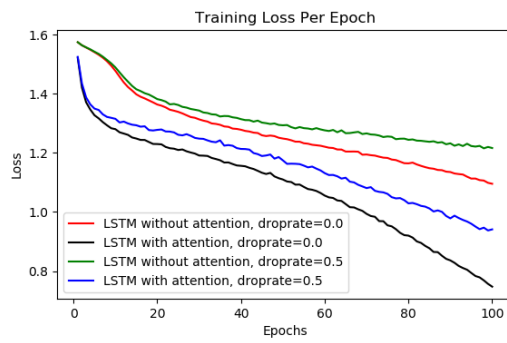
将所有模型训练到收敛, 取最大的 validation accuracy 作为评价指标。

4.3 Quantitative Results









模型	正确率(with Self-Attention)	正确率(without Self-Attention)
CNN	-----	45.05%/45.14%
RNN	42.51%/44.41%	42.60%/44.41%
LSTM	43.23%/44.23%	41.33%/44.14%
GRU	44.50%/45.78%	43.69%/44.05%

(其中每一个中左边的数值是未加 Dropout 的结果，右边的数值是带 Dropout 的结果)

通过对比，可以得到结论：在没有 Self-Attention 时，RNN 的收敛速度较快，LSTM 和 GRU 的收敛速度差不多相同，在没有 Dropout 时，RNN 的正确率最高，有 Dropout 时 GRU 的正确率最高。在有 Self-Attention 时，RNN 的收敛速度最快，GRU 次之，LSTM 最慢。GRU 的正确率最高。无论是否有 Self-Attention，加入 Dropout 会给正确率带来一定的提升。Self-Attention 会加快模型的收敛速度。最终，GRU+Self-Attention+Dropout 能够在正确率上超过 CNN 方法。

5. Conclusion

通过实验和对比，我发现 RNN、LSTM、GRU 三种模型都能比较好的完成这次任务，其中 GRU 的正确率最高并且能超过 CNN 方法。另外 Self-Attention 也能提高正确率和收敛速度。

References

Yoon Kim. Convolutional Neural Networks for Sentence Classification

Zhouhan Lin, Minwei Feng , Cicero Nogueira dos Santos , Mo Yu , Bing Xiang□, Bowen Zhou & Yoshua Bengio. A STRUCTURED SELF-ATTENTIVE SENTENCE EMBEDDING