

Context-Aware Self-Attention Network

서상우

Context-Aware Self-Attention Networks

- AAAI 2019
- Self-attention의 변형 모델
- Neural Machine Translation Task
- WMT (Conference on Machine Translation)
 - WMT14 English⇒German
 - WMT17 Chinese⇒English

Baosong Yang¹ Jian Li² Derek F. Wong¹ Lidia S. Chao¹ Xing Wang³ Zhaopeng Tu^{3*}

¹NLP²CT Lab, Department of Computer and Information Science, University of Macau

nlp2ct.baosong@gmail.com, {derekfw, lidiasc}@umac.mo


²The Chinese University of Hong Kong

jianli@cse.cuhk.edu.hk

³Tencent AI Lab

{brightxwang, zptu}@tencent.com

Context-Aware Self-Attention Networks

<div></div> <div><div>Baosong Yang</div><div>Ph.D. candidate, University of Macau 在 umac.mo 的电子邮件经过验证 - 首页 Machine Learning Natural Language Processing Machine Translation</div></div> <div>✉ 关注</div>		
标题	引用次数	年份
Multi-Head Attention with Disagreement Regularization J Li, Z Tu, B Yang, MR Lyu, T Zhang EMNLP 2018	7	2018
Modeling Localness for Self-Attention Networks B Yang, Z Tu, DF Wong, F Meng, LS Chao, T Zhang EMNLP 2018	6	2018
Towards Bidirectional Hierarchical Representations for Attention-based Neural Machine Translation B Yang, DF Wong, T Xiao, LS Chao, J Zhu EMNLP 2017	6	2017
Convolutional Self-Attention Networks B Yang, L Wang, DF Wong, LS Chao, Z Tu NAACL 2019	2	2019
Context-Aware Self-Attention Networks B Yang, J Li, D Wong, LS Chao, X Wang, Z Tu AAAI 2019	2	2019
Information Aggregation for Multi-Head Attention with Routing-by-Agreement J Li, B Yang, ZY Dou, X Wang, MR Lyu, Z Tu NAACL 2019		2019
Modeling Recurrence for Transformer J Hao, X Wang, B Yang, L Wang, J Zhang, Z Tu NAACL 2019		2019

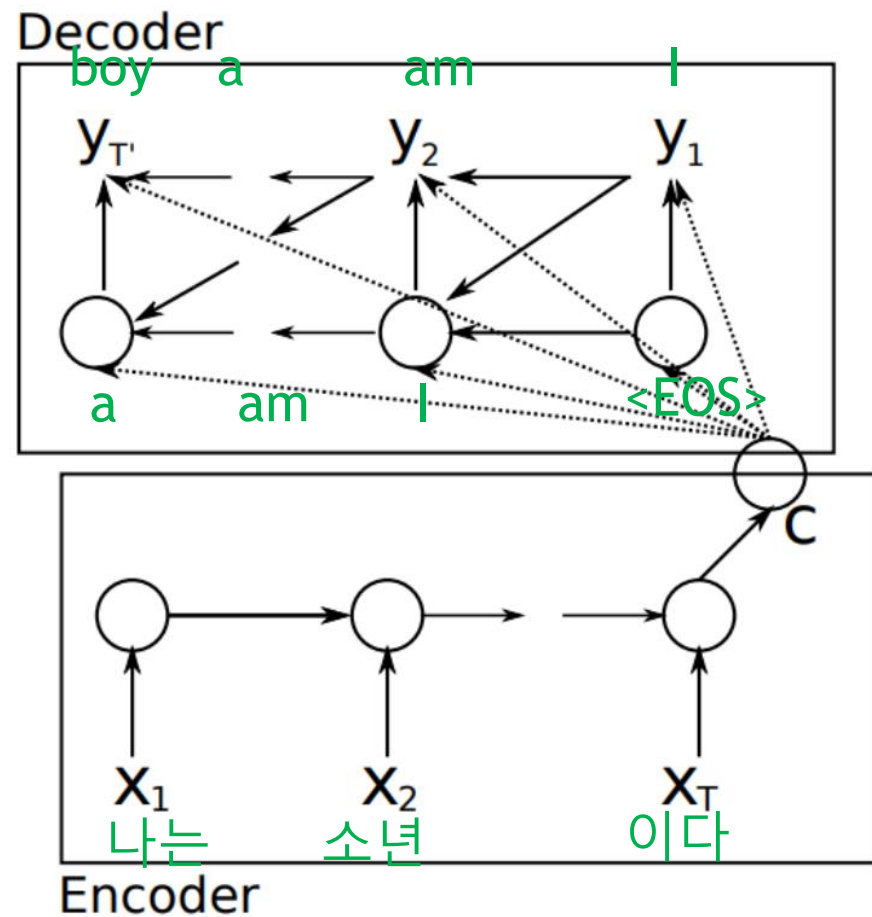
Seq2Seq

- 가장 기본적인 NMT task 모델
- Encoder Input으로 Source Sentence,
- Decoder Output으로 Target Sentence, Input은 shifted right 한 거

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

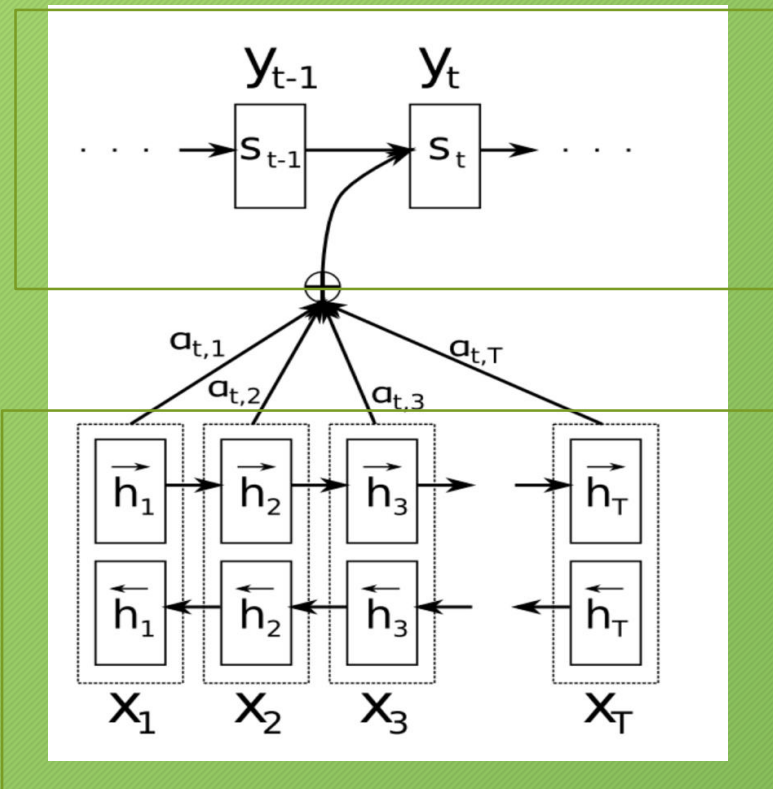
$$\mathbf{h}_{\langle t \rangle} = f(\mathbf{h}_{\langle t-1 \rangle}, y_{t-1}, \mathbf{c}),$$

$$P(y_t | y_{t-1}, y_{t-2}, \dots, y_1, \mathbf{c}) = g(\mathbf{h}_{\langle t \rangle}, y_{t-1}, \mathbf{c}).$$



Attention mechanism

decoder



encoder

$$s_i = f(s_{i-1}, y_{i-1}, c_i).$$

$$p(y_i | y_1, \dots, y_{i-1}, \mathbf{x}) = g(y_{i-1}, s_i, c_i),$$

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j.$$

최종 Encoder output c 대신
Input 전체를 의미하는 x

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})},$$

$$e_{ij} = a(s_{i-1}, h_j)$$

$$e_{ij} = s_{i-1} h_j^T$$

(dot product attention)

What is self attention?

- Attention is All You Need (NIPS 2017)에서 제안
- Transformer 블록
 - Self attention
 - Layer normalization
 - Residual network
- CNN, RNN 없이 NMT task의 SOTA
- Seq2Seq 모델의 단점 해결
 - 병렬처리가 불가능 속도 느림
 - Long Term Dependency를 잡을 수 없음 (CNN도 마찬가지)

Transformer
block

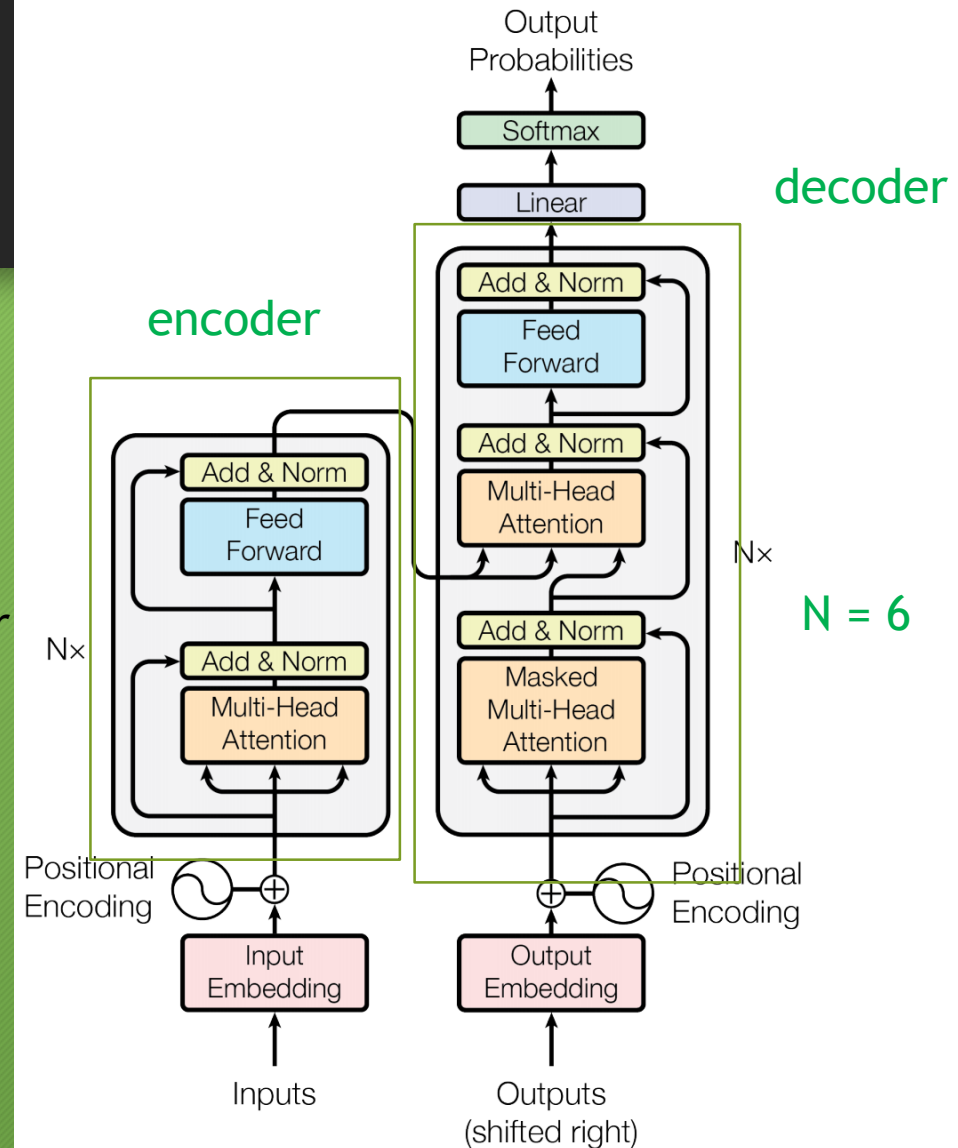
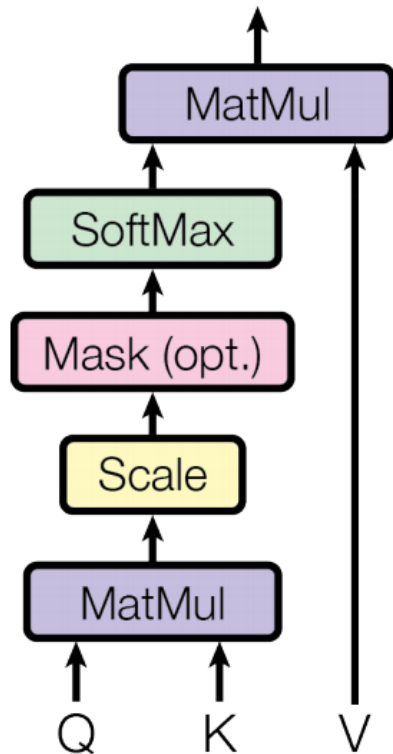


Figure 1: The Transformer - model architecture.

What is self attention?

Scaled Dot-Product Attention



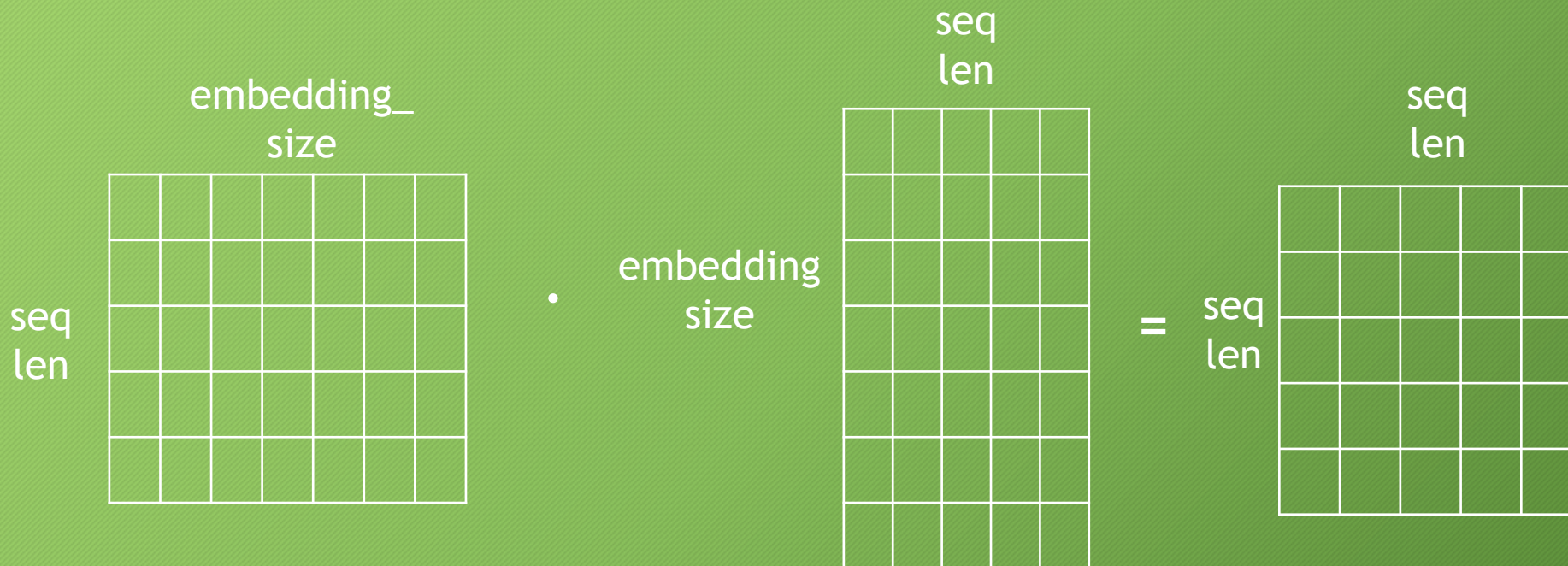
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

- encoder-decoder attention의 경우,
Q : 디코더의 이전 레이어 hidden state
K : 인코더의 output state
V : 인코더의 output state
- self-attention의 경우,
Q=K=V : 인코더의 output state (입력 Sentence)

What is self attention?

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

단어 간의 유사도
matrix가 나온다



What is self attention?

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

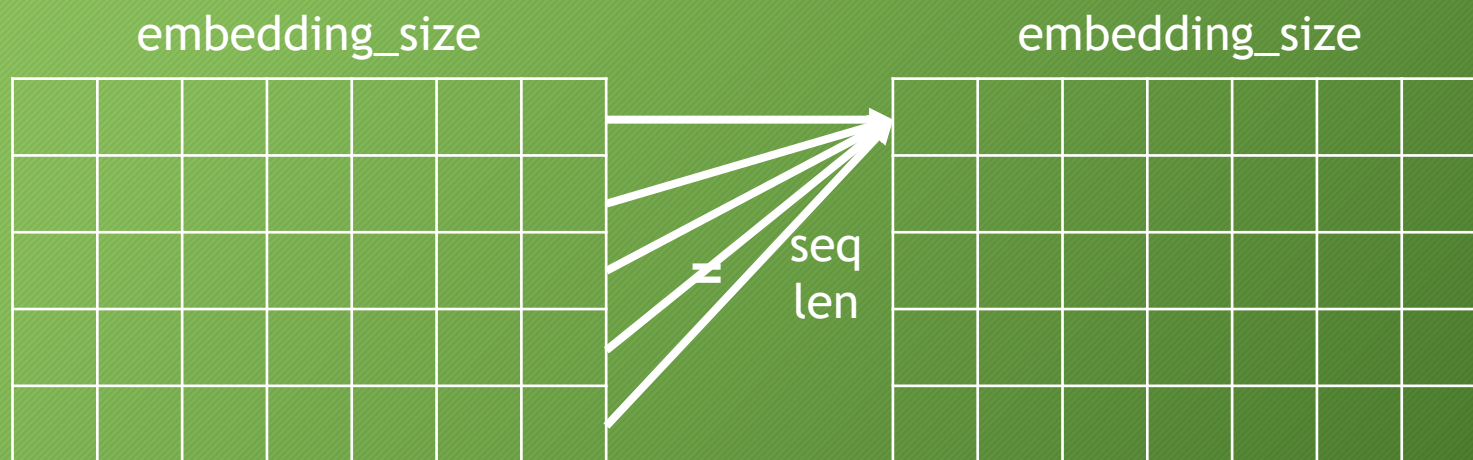
Dot product하면 값이 상당히 커진다.

Dot product한 값과 V 값의 스케일을 맞추어 주기 위해 $\sqrt{d_k}$ 를 나누어 준다.

	i	am	sangwoo	...	thanks
i	0.5	0.1	0.3	...	0.1
am	0.1	0.5	0.2	...	0.1
sangwoo	0.3	0.2	0.4	...	
...					
thanks					

.

seq
len



What is self attention?

- Multi-Head Attention은 self-attention을 여러 개 사용
- 문장 내의 여러 의미를 파악하기 위해 self-attention을 여러 개를 준다.
- 헤드에는 서로 다른 Linear가 곁해져서 들어가기 때문에 여러 의미를 파악 가능

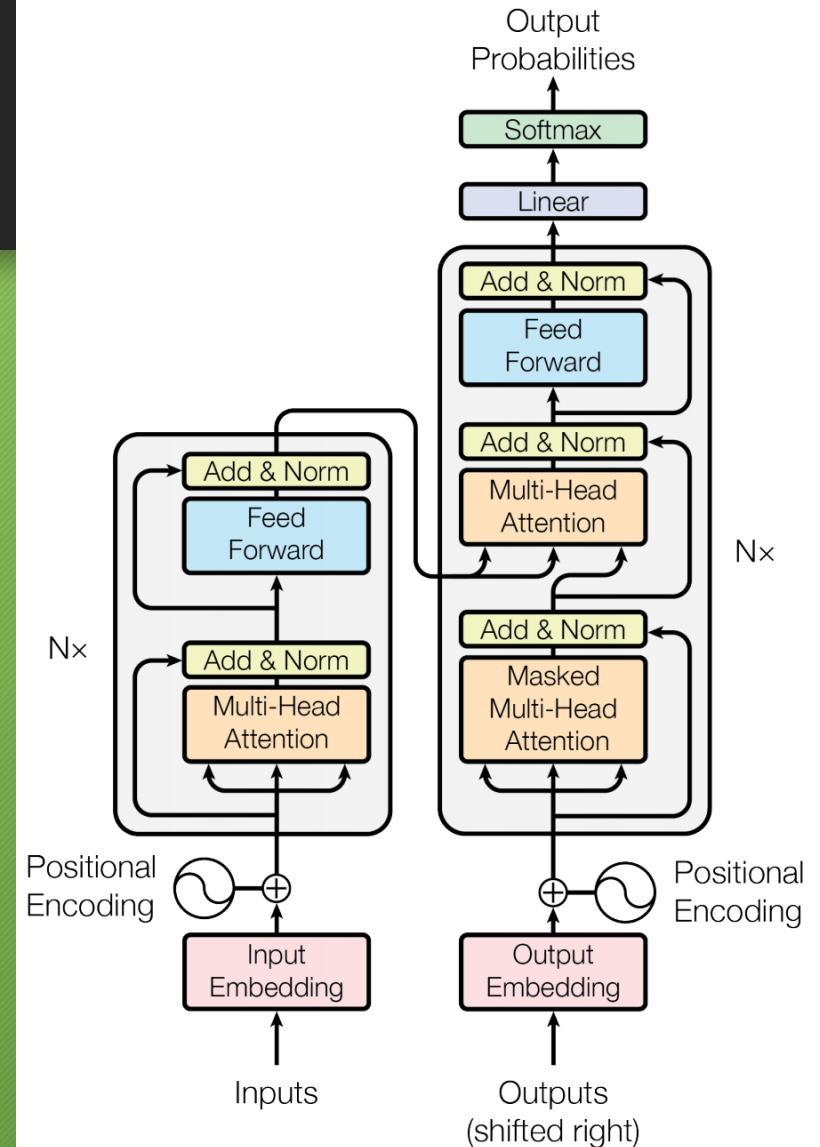
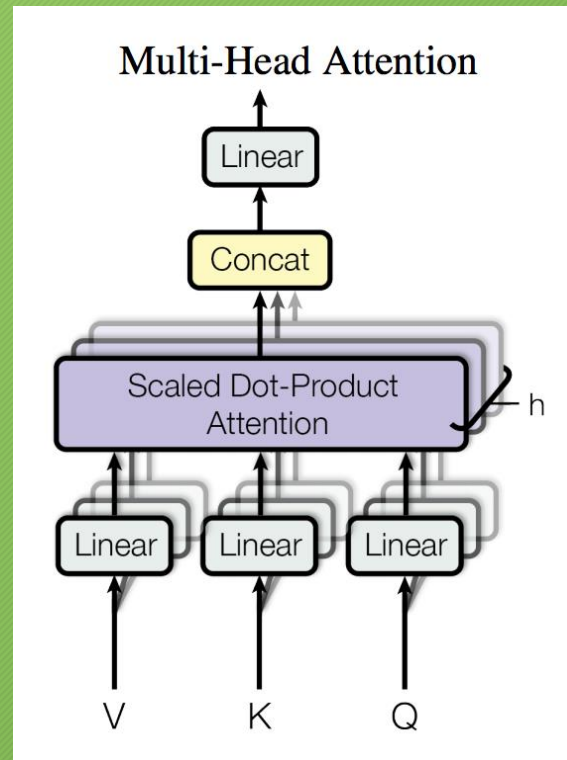


Figure 1: The Transformer - model architecture.

Context-Aware Self-Attention Networks

Self attention review

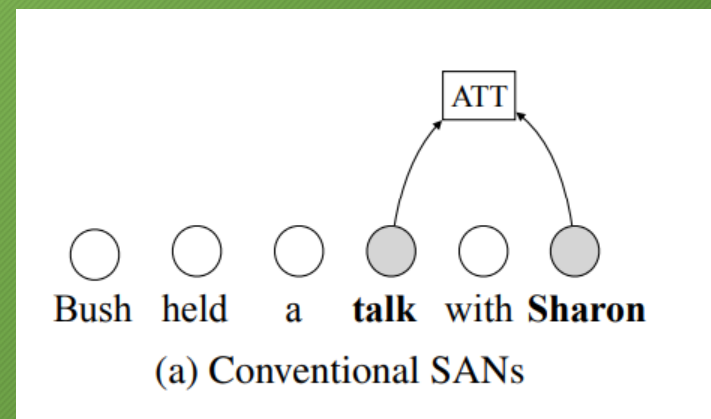
$$\begin{bmatrix} \mathbf{Q} \\ \mathbf{K} \\ \mathbf{V} \end{bmatrix} = \mathbf{H} \begin{bmatrix} \mathbf{W}_Q \\ \mathbf{W}_K \\ \mathbf{W}_V \end{bmatrix},$$

$$\mathbf{O} = \text{ATT}(\mathbf{Q}, \mathbf{K}) \mathbf{V},$$

$$\text{ATT}(\mathbf{Q}, \mathbf{K}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right),$$

$$\mathbf{Q}\mathbf{K}^T = (\mathbf{H}\mathbf{W}_Q)(\mathbf{H}\mathbf{W}_K)^T = \mathbf{H}(\mathbf{W}_Q\mathbf{W}_K^T)\mathbf{H}^T,$$

해당 식에는 두 단어 간의 유사도만 구할 뿐 Context를 반영하지 않는다 => Self attention의 문제점



Context-Aware Self-Attention Networks

기존의 Q,K 에 C라는 context vectore 더한다!

C와 Q,K 간의 크기를 맞추기 위해 weight matrix U_Q, U_K 를 곱한다!

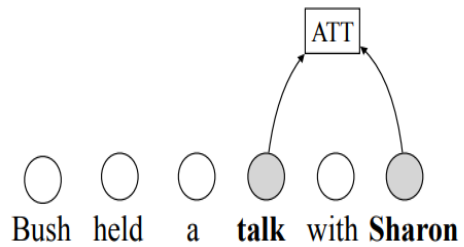
λ_Q, λ_K 는 Q,K와 C 중 어떠한 것을 더 사용할 지를 결정하는 gating parameter

λ_Q, λ_K 또한 trainable하게 주어지는 파라미터로 뉴럴 네트워크가 contex와 원래의 값 중에서 어디를 더 치중할 지를 결정

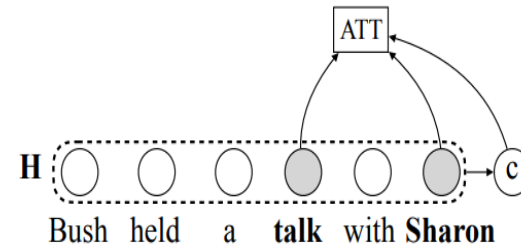
$$\begin{bmatrix} \hat{\mathbf{Q}} \\ \hat{\mathbf{K}} \end{bmatrix} = (1 - \begin{bmatrix} \lambda_Q \\ \lambda_K \end{bmatrix}) \begin{bmatrix} \mathbf{Q} \\ \mathbf{K} \end{bmatrix} + \begin{bmatrix} \lambda_Q \\ \lambda_K \end{bmatrix} (\mathbf{C} \begin{bmatrix} \mathbf{U}_Q \\ \mathbf{U}_K \end{bmatrix})$$

$$\begin{bmatrix} \lambda_Q \\ \lambda_K \end{bmatrix} = \sigma \left(\begin{bmatrix} \mathbf{Q} \\ \mathbf{K} \end{bmatrix} \begin{bmatrix} \mathbf{V}_Q^H \\ \mathbf{V}_K^H \end{bmatrix} + \mathbf{C} \begin{bmatrix} \mathbf{U}_Q \\ \mathbf{U}_K \end{bmatrix} \begin{bmatrix} \mathbf{V}_Q^C \\ \mathbf{V}_K^C \end{bmatrix} \right),$$

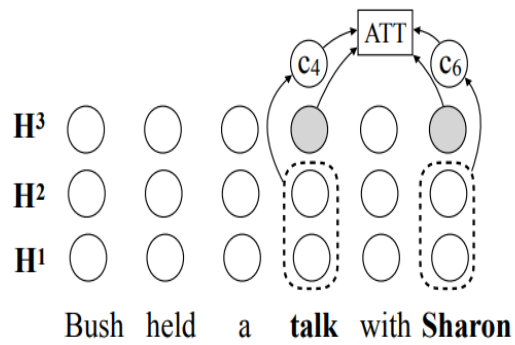
What is context?



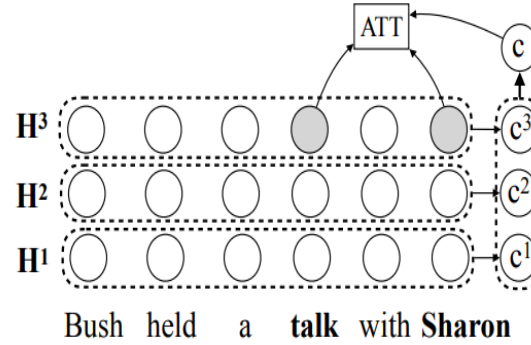
(a) Conventional SAs



(b) Global Context



(c) Deep Context



(d) Deep-Global Context

- Global context
 - Hidden의 평균

$$\mathbf{c} = \overline{\mathbf{H}}$$

- Deep context
 - 모든 이전의 hidden

$$\mathbf{C} = [\mathbf{H}^1, \dots, \mathbf{H}^{l-1}]$$

- Deep-Global context
 - 모든 이전의 context

$$\mathbf{c} = [\mathbf{c}^1, \dots, \mathbf{c}^l]$$

Experiments WMT14 (En \Rightarrow De)

#	Model	Applied to	Context Vectors	# Para.	Train	Decode	BLEU
1	BASE	n/a	n/a	88.0M	1.28	1.52	27.31
2	OURS	encoder	<i>global context</i>	91.0M	1.26	1.50	27.96
3			<i>deep-global context</i>	99.0M	1.25	1.48	28.15
4			<i>deep context</i>	95.9M	1.18	1.38	28.01
5			<i>deep-global context + deep context</i>	106.9M	1.16	1.36	28.26
6		decoder	<i>deep-global context</i>	99.0M	1.23	1.44	27.94
7			<i>deep-global context + deep context</i>	106.9M	1.15	1.35	28.02
8		both	5 + 7	125.8M	1.04	1.20	28.16

Experiments WMT17 (Zh \Rightarrow En)

System	Architecture	Zh⇒En			En⇒De		
		# Para.	Train	BLEU	# Para.	Train	BLEU
Existing NMT systems							
(Vaswani et al. 2017)	TRANSFORMER-BASE	n/a	n/a	n/a	65M	n/a	27.30
	TRANSFORMER-BIG	n/a	n/a	n/a	213M	n/a	28.40
(Hassan et al. 2018)	TRANSFORMER-BIG	n/a	n/a	24.20	n/a	n/a	n/a
Our NMT systems							
this work	TRANSFORMER-BASE	107.9M	1.21	24.13	88.0M	1.28	27.31
	+ Context-Aware SANs	126.8M	1.10	24.67 [↑]	106.9M	1.16	28.26 [↑]
	TRANSFORMER-BIG	303.9M	0.58	24.56	264.1M	0.61	28.58
	+ Context-Aware SANs	379.4M	0.41	25.15 [↑]	339.6M	0.44	28.89

Avg weight Experiments

