AI Lab
Seminar

SYNTHESIZER: Rethinking Self-Attention in Transformer Models

석사 3기 조충현

Introduction

Intro

SYNTHESIZER: Rethinking Self-Attention in Transformer Models

Transformer

- Transformer 모델은 여러 task에서 다른 딥러닝 모델들의 성능을 능가하는 모델로 현재 대부분의 자연어 처리 task에 사용됨
- 이러한 Transformer 성능의 중요한 요소는 query-key-value의 dot product attention(self-attention)
- Self-attention 메커니즘을 사용하면 긴 범위의 dependency 정보를 찾기때문에 성능이 좋다
- 그러나 이 논문의 저자는 '**과연 이 self-attention이 정말로 중요한지 의구심이 든다**' 고 말함
- 계산량이 많은 dot product 연산을 하는 self-attention이 그 만큼 중요한가?

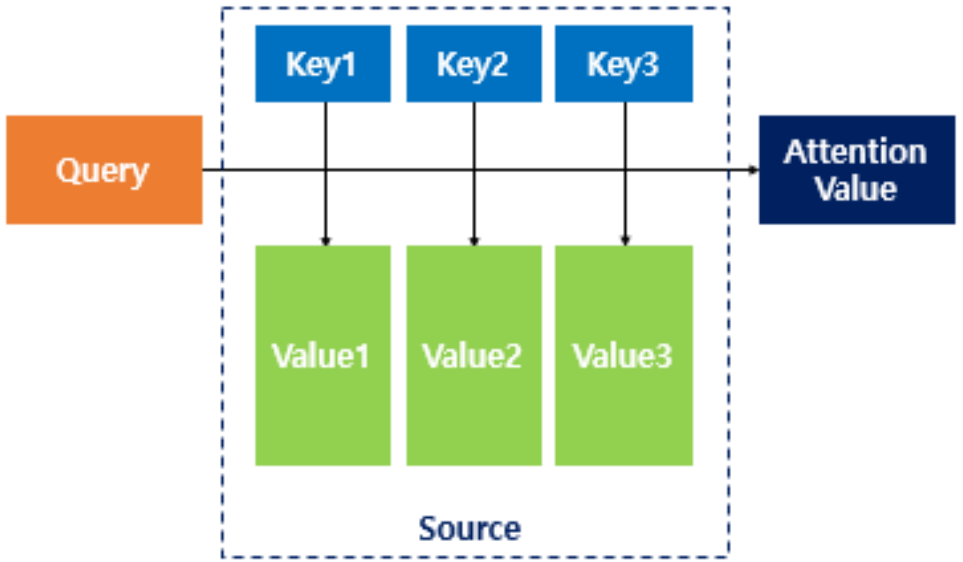
- **이제 그 비밀에 대하여 파헤쳐 보자!**

Related Work

SYNTHESIZER: Rethinking Self-Attention in Transformer Models

Attention

- 기본 아이디어: 해당 시점에서 예측 해야 할 단어와 연관이 있는 입력 단어 부분을 좀 더 집중해서 보자



Q : t 시점의 디코더 셀에서의 은닉 상태

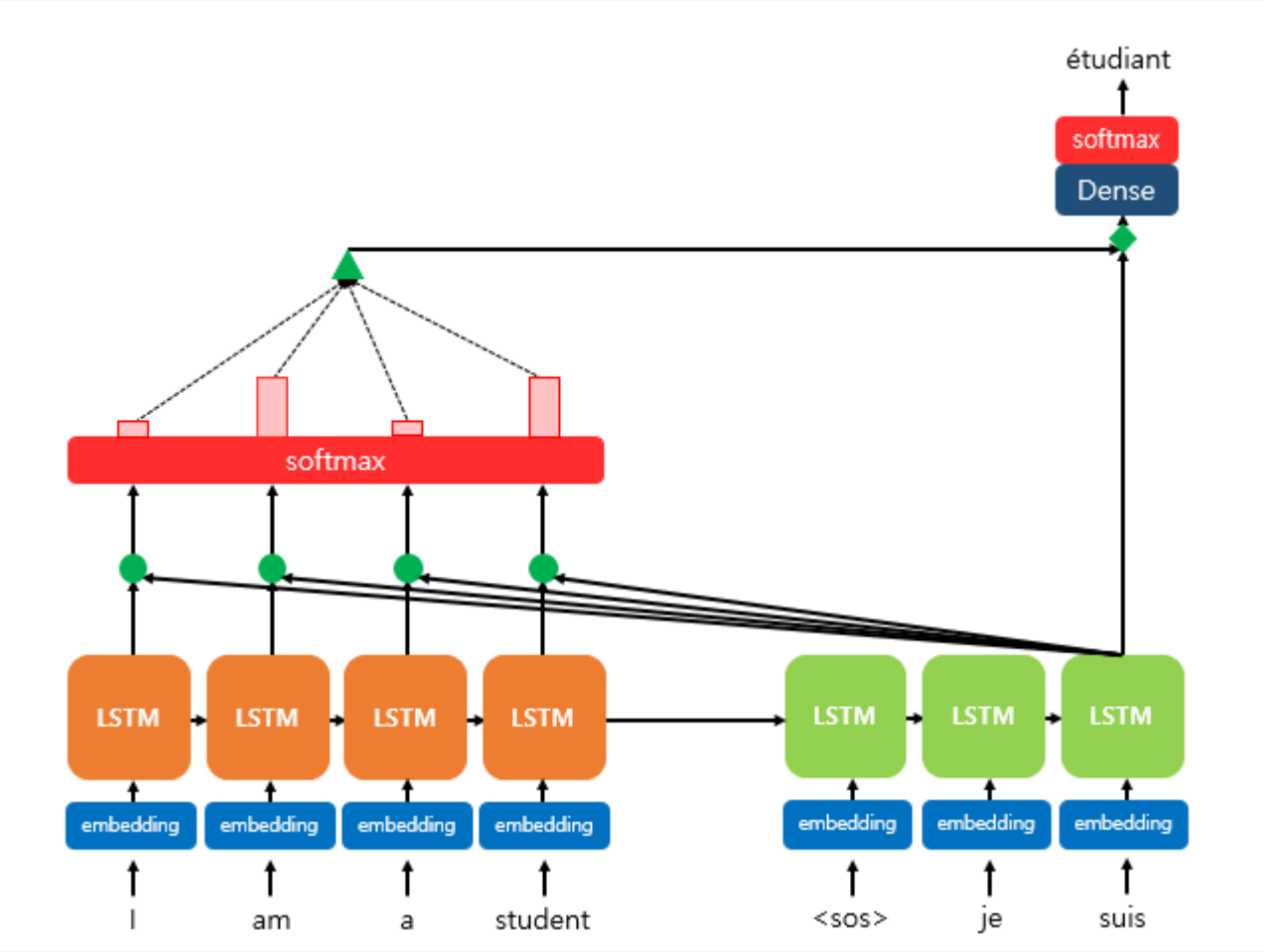
K : 모든 시점의 인코더 셀의 은닉 상태들

V : 모든 시점의 인코더 셀의 은닉 상태들

Attention(Q, K, V) = Attention Value

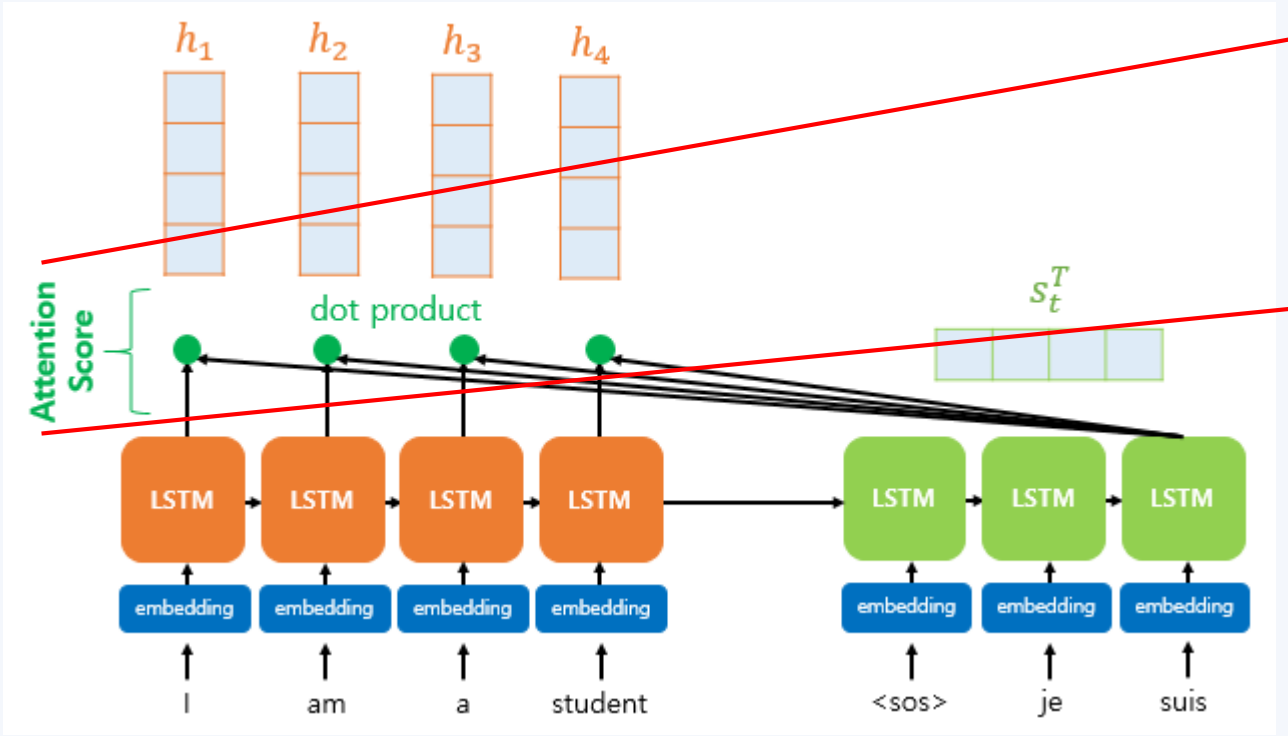
SYNTHESIZER: Rethinking Self-Attention in Transformer Models

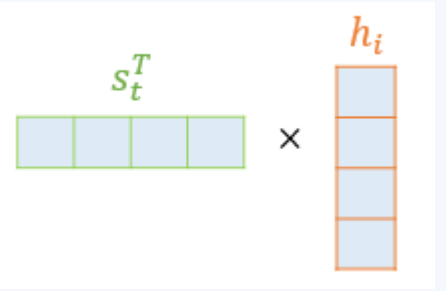
Dot-Product Attention



SYNTHESIZER: Rethinking Self-Attention in Transformer Models

Dot-Product Attention



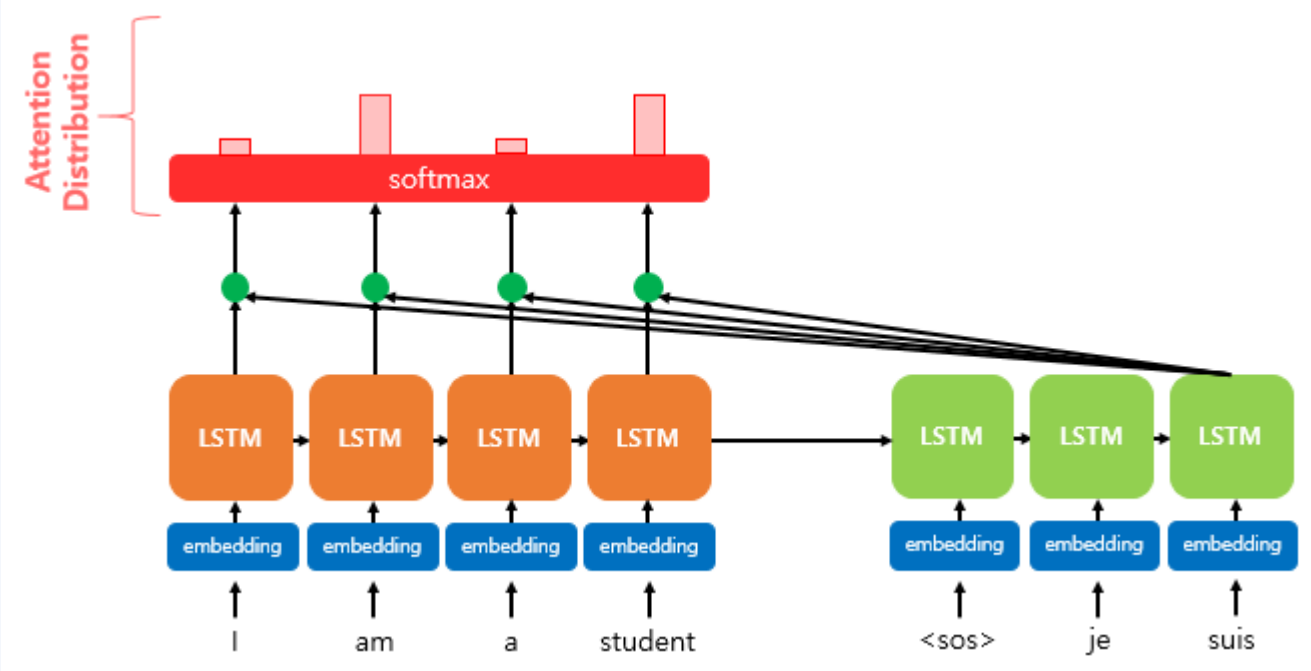


$$score(s_t, h_i) = s_t^T h_i$$

$$e^t = [s_t^T h_1, \ldots, s_t^T h_N]$$

SYNTHESIZER: Rethinking Self-Attention in Transformer Models

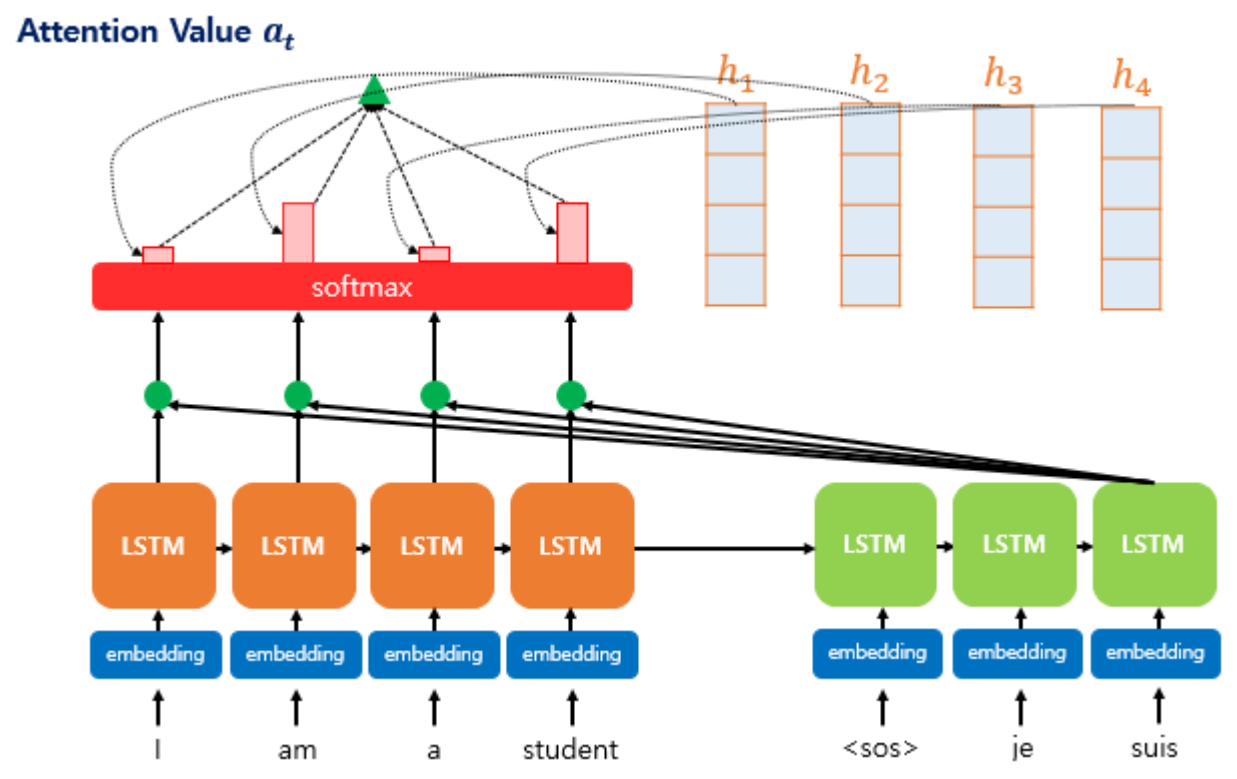
Dot-Product Attention



$$\alpha^t = softmax(e^t)$$

SYNTHESIZER: Rethinking Self-Attention in Transformer Models

Dot-Product Attention

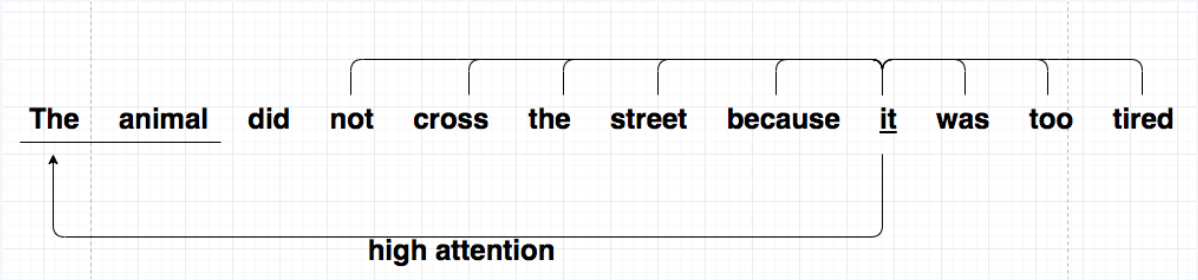


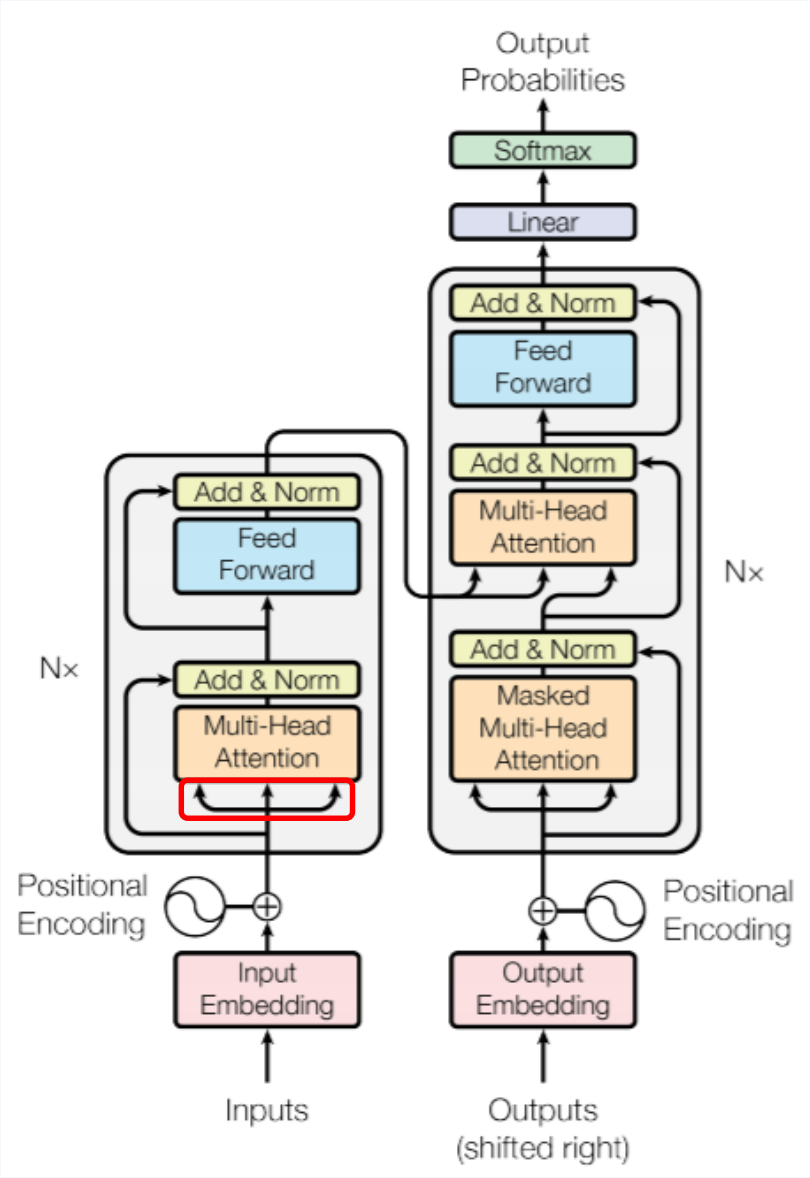
$$a_t = \sum_{i=1}^{N} \alpha_i^t h_i$$

Attention value

SYNTHESIZER: Rethinking Self-Attention in Transformer Models

Self-attention in Transformer





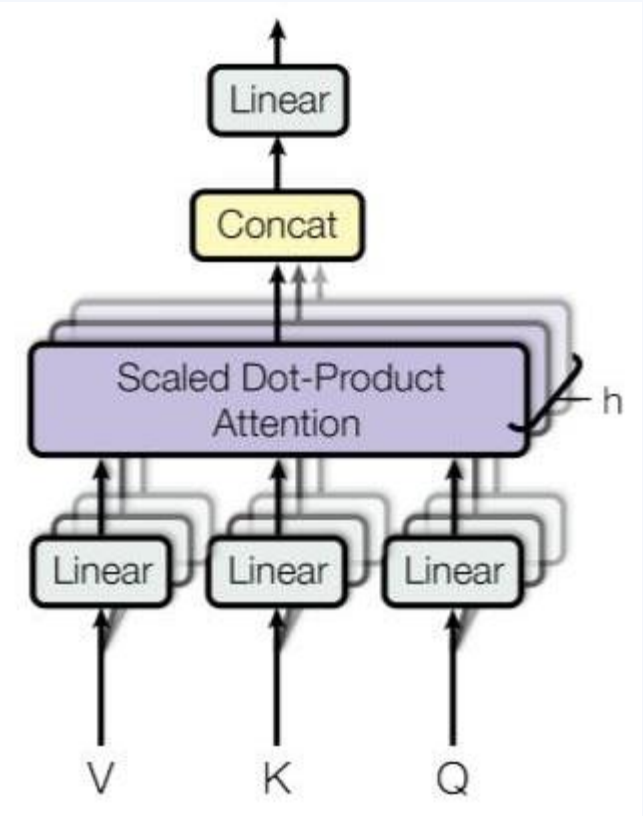
SYNTHESIZER: Rethinking Self-Attention in Transformer Models

Self-attention in Transformer

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O$$
$$\text{where head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

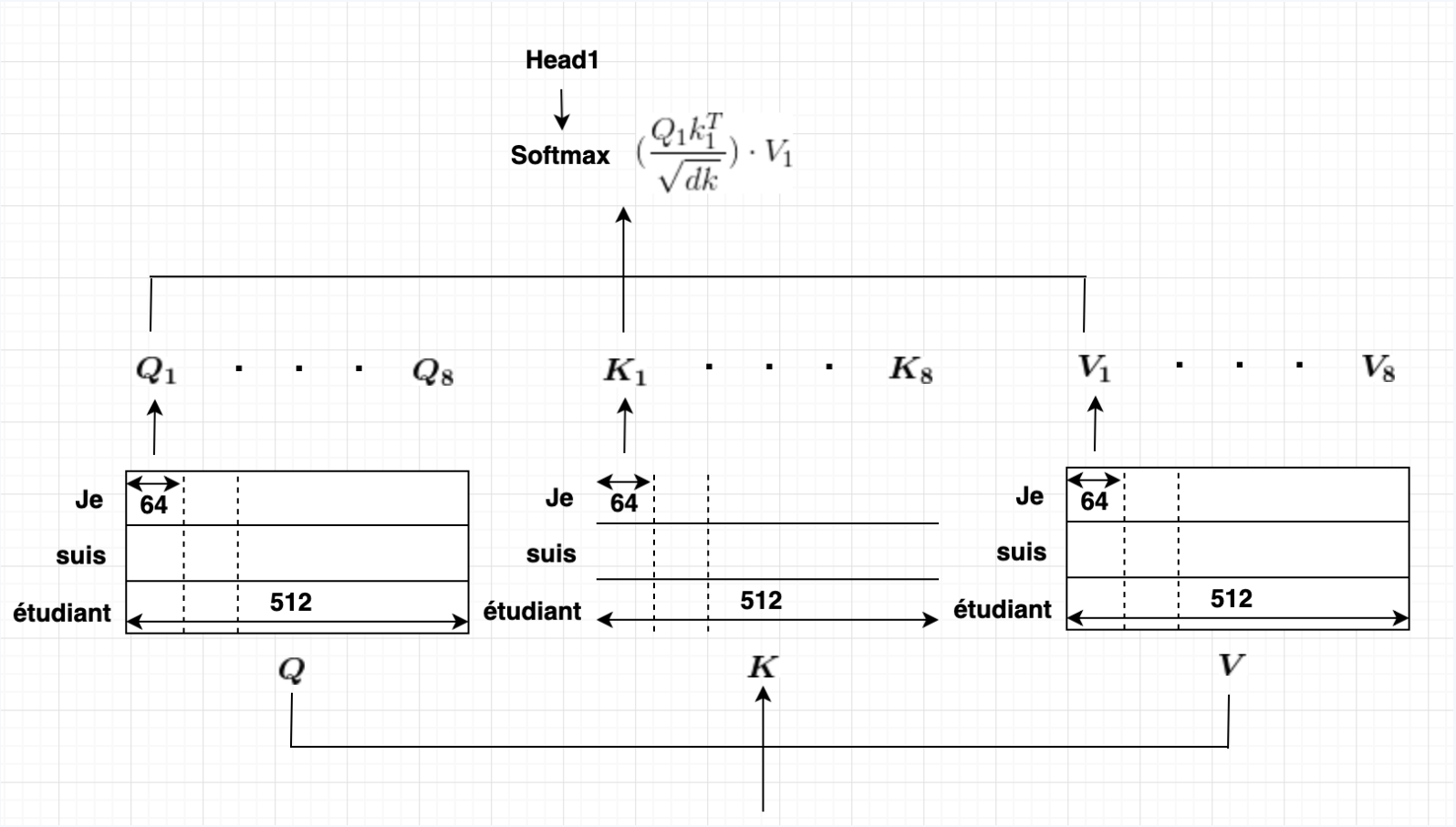
$$W_i^Q,\ W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}, W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_k}, W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$$

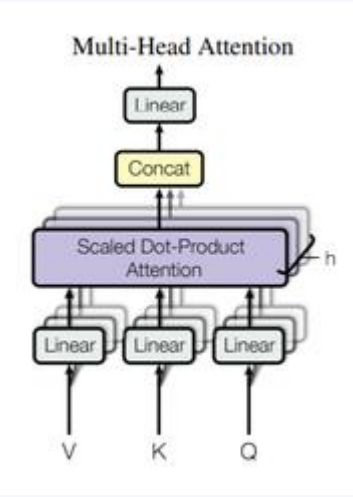
$$d_k = d_v = d_{\text{model}}/h = 64$$



SYNTHESIZER: Rethinking Self-Attention in Transformer Models

Self-attention in Transformer





Self-attention 단점

- Single token과 모든 다른 tokens 과의 상대적 정보를 학습하는 것

-> 이러한 특징은 일관된 global context가 없기 때문에 다른 token에 의해 자유롭게 변동 될 수 있다

- Dot product 연산량이 상당히 많다

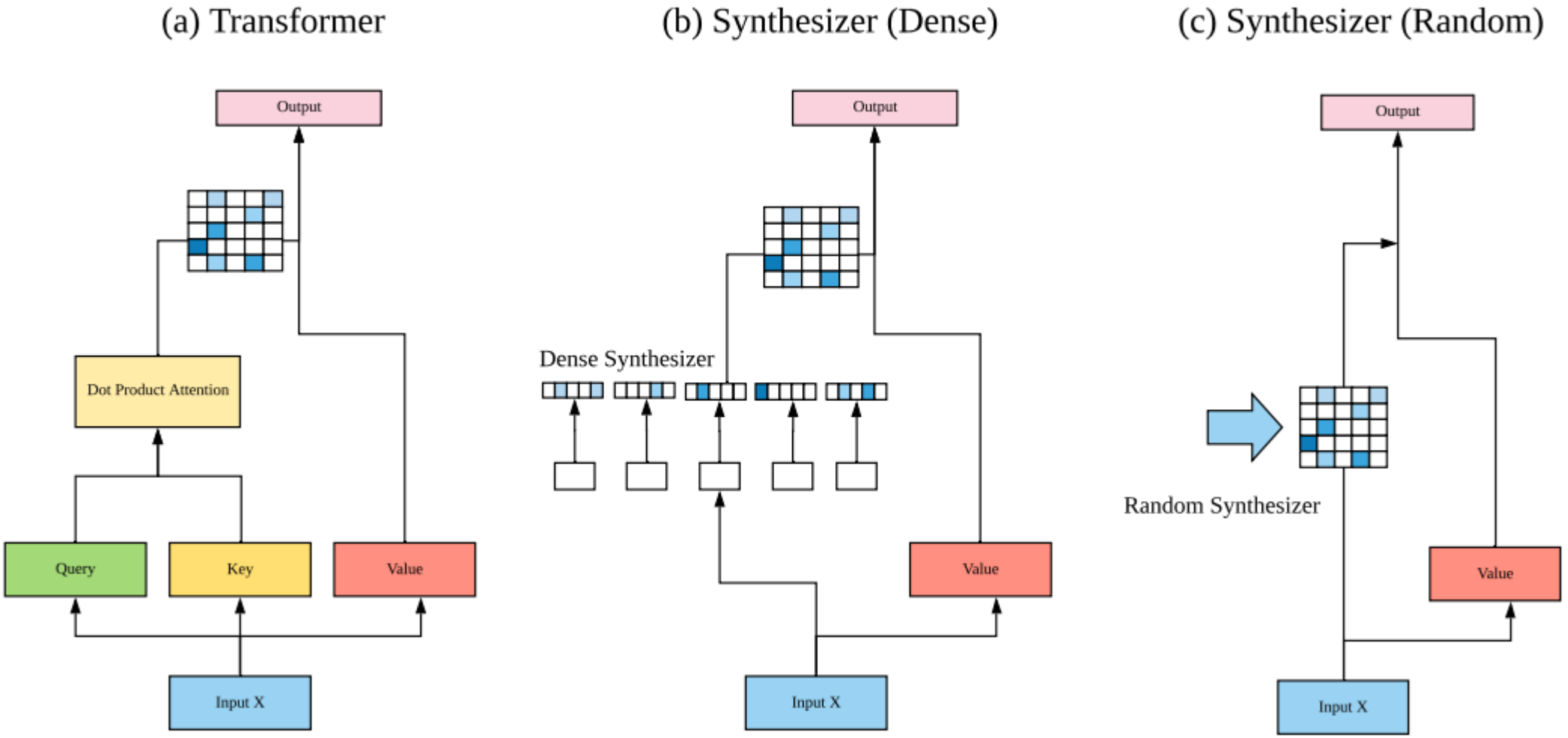
그래서 우리는 self-attention을 대체하는 SYNTHESIZER 모델을 제안한다!

Method

AI Lab Seminar

Method

SYNTHESIZER: Rethinking Self-Attention in Transformer Models



SYNTHESIZER: Rethinking Self-Attention in Transformer Models

Dense Synthesizer

$$X \in \mathbb{R}^{l \times d} \rightarrow Y \in \mathbb{R}^{l \times d}$$

l : sequence length

d : dimension of the model

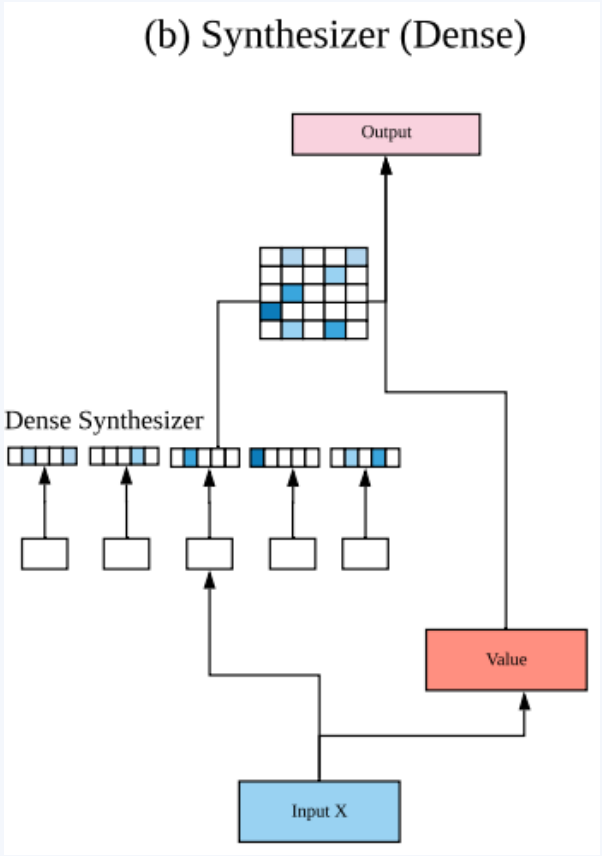
$$B_i = F(X_i)$$

$$F(X) = W(\sigma_R(W(X) + b)) + b$$

σ_R 은 ReLU

$$Y = \text{Softmax}(B)G(X)$$

G(X) 은 Transformer model의 V와 동일



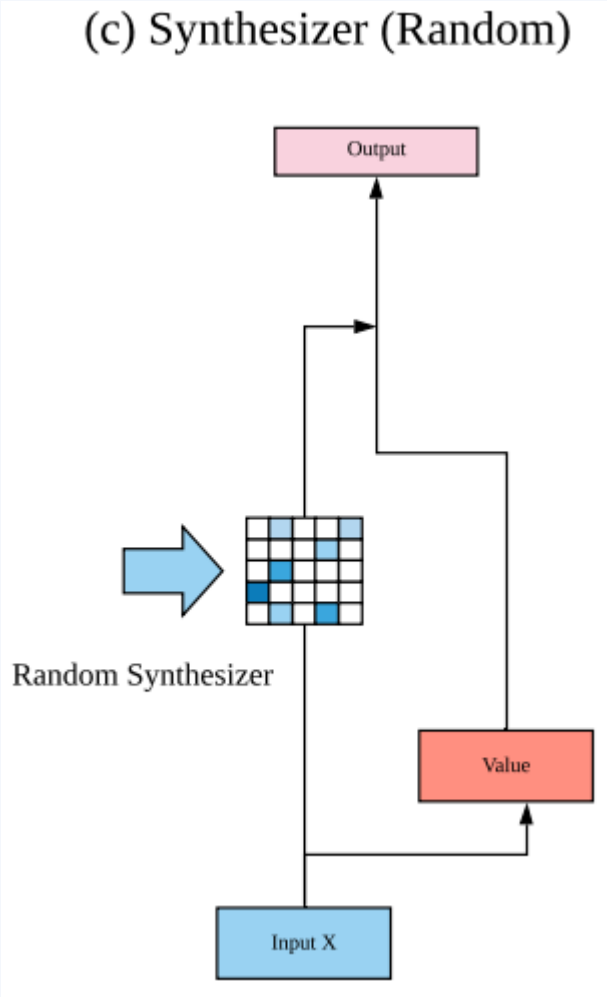
SYNTHESIZER: Rethinking Self-Attention in Transformer Models

Random Synthesizer

$$Y = \text{Softmax}(R)G(X)$$

R: randomly initialized matrix

$$R \in \mathbb{R}^{l \times l}$$



이렇게 random으로 하게 되면 token의 상호작용이나 token 각각의 정보에 의존하지 않고 globally 한 정보를 가지는 task-specific 배열을 학습한다

SYNTHESIZER: Rethinking Self-Attention in Transformer Models

Factorized Models

- l 이 크면 연산량이 많아지는 단점이 있다. 그러므로 factorized variations을 사용하자

Factorized Dense Synthesizer

$$A, B = F_A(X_i), F_B(X_i)$$

$F_A(.)$는 a dimension으로 project, $F_B(.)$는 b dimension으로 project

$$a \times b = l$$

$$Y = \text{Softmax}(C)G(X)$$

$$C = H_A(A) * H_B(B)$$

H_A, H_B는 tiling function : vector를 k번 복제하는 것

$$H_A(.), H_B(.) \rightarrow \mathbb{R}^{ab}$$

$$C \in \mathbb{R}^{l \times l}$$

Factorized Random Synthesizer

$$Y = \text{Softmax}(R_1 R_2^{\top})G(X)$$

$$R_1, R_2 \in \mathbb{R}^{\ell \times k} \quad k << \ell$$

SYNTHESIZER: Rethinking Self-Attention in Transformer Models

Mixture of Synthesizers

- 제안된 SYNTHESIZER model을 혼합하여서 사용할 수 있다

$$Y = \text{Softmax}(\alpha_1 S_1(X) + \cdots \alpha_N S_N(X))G(X)$$

$S(.)$는 synthesizing function

$\alpha(where\ \sum \alpha = 1)$ 는 learnable weights

$$Y = \text{Softmax}(R_1 R_2^\top + F(X))G(X)$$

Mixing Random Factorized with Dense Synthesizer

SYNTHESIZER: Rethinking Self-Attention in Transformer Models

Transformer도 Synthesizing function의 한 종류

$$S(X) = F_Q(X)F_K(X)^\top$$

Model	$S(X)$	Condition On	Sample	Interact	$\|\theta\|$
Dot Product Attention	$F_Q(X)F_K(X_i)^\top$	$X_j \ \forall j$	Local	Yes	$2d^2$
Random	R	N/A	Global	No	$. \ell^2$
Factorized Random	$R_1 R_2^\top$	N/A	Global	No	$2\ell k$
Dense	$F_1\sigma(F_2(X_i))$	X_i	Local	No	$d^2 + d\ell$
Factorized Dense	$H_A(F_A(X_i))) * H_B(F_B(X_i)))$	X_i	Local	No	$d^2 + d(k_1 + k_2)$

Table 1: Overview of all Synthesizing Functions.

Experiments

SYNTHESIZER: Rethinking Self-Attention in Transformer Models

Experiments

Machine Translation

- WMT'14 English-German(EnDe)
  - 4.5M sentence pairs
- WMT'14 English-French(EnFr)
  - 36M sentence pairs

Language Modeling

- Language Modeling One Billion (LM1B)

Text Generation

- CNN/Dailymail : summarization
- PersonaChat : dialogue generation

SYNTHESIZER: Rethinking Self-Attention in Transformer Models

Model	NMT (BLEU) # Params	EnDe	EnFr	LM (PPL) # Params	LM1B
Transformer [Vaswani et al., 2017]	68M	27.30	38.10	-	-
Transformer (Our run)	68M	27.67	41.57	70M	38.21
Transformer (Control)	73M	27.97	41.83	-	-
Synthesizer (Fixed Random)	61M	23.89	38.31	53M	50.52
Synthesizer (Random)	67M	27.27	41.12	58M	40.60
Synthesizer (Factorized Random)	61M	27.30	41.12	53M	42.40
Synthesizer (Dense)	62M	27.43	41.39	53M	40.88
Synthesizer (Factorized Dense)	61M	27.32	41.57	53M	41.20
Synthesizer (Random + Dense)	67M	27.68	41.21	58M	42.35
Synthesizer (Dense + Vanilla)	74M	27.57	41.38	70M	**37.27**
Synthesizer (Random + Vanilla)	73M	**28.47**	**41.85**	70M	40.05

Table 2: Experimental Results on WMT'14 English-German, WMT'14 English-French Machine Translation tasks and Language Modeling One Billion (LM1B).

SYNTHESIZER: Rethinking Self-Attention in Transformer Models

Model	Summarization			Dialogue				
	Rouge-1	Rouge-2	Rouge-L	Bleu-1/4	Rouge-L	Meteor	CIDr	Emb
Transformer	38.24	**17.10**	35.77	12.03/3.20	13.38	5.89	18.94	83.43
Synthesizer (R)	35.47	14.92	33.10	14.64/2.25	15.00	6.42	19.57	84.50
Synthesizer (D)	36.05	15.26	33.70	**15.58/4.02**	**15.22**	**6.61**	**20.54**	**84.95**
Synthesizer (D+V)	38.57	16.64	**36.02**	14.24/3.57	14.22	6.32	18.87	84.21
Synthesizer (R+V)	**38.57**	16.24	35.95	14.70/2.28	14.79	6.39	19.09	84.54

Table 3: Experimental results on Abstractive Summarization (CNN/Dailymail) and Dialogue Generation (PersonaChat).

Conclusion

Conclusion

- SYNTHESIZER라는 새로운 Transformer model 제안
- Global, local, token-token wise alignment의 utility 연구
- 여러 task에서 synthetic attention이 self-attention과 견줄 만한 성능 기록
- 차후 연구에서 self-attention의 대한 더 깊은 연구가 필요하다는 것 제시

Q&A

Thank you