Attention is All you need

- Transformer -

NLP 2조
김유진, 문예진, 송경민, 이상민, 한유경

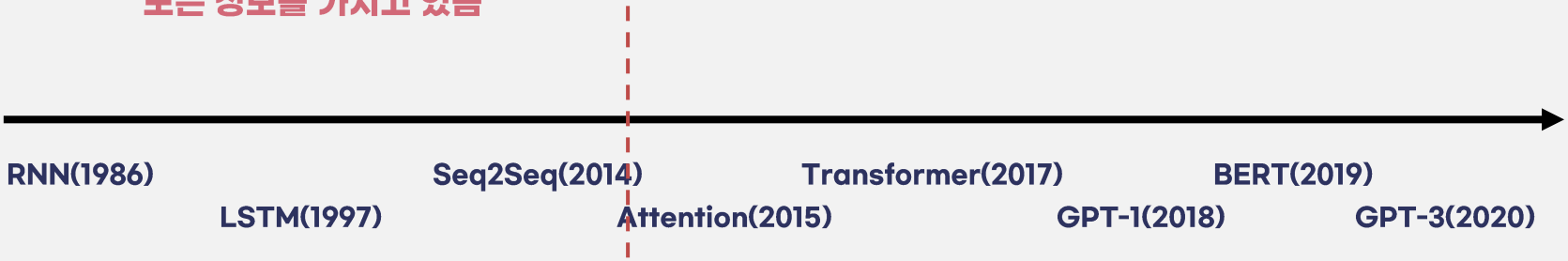
NLP 발전과정

RNN 기반

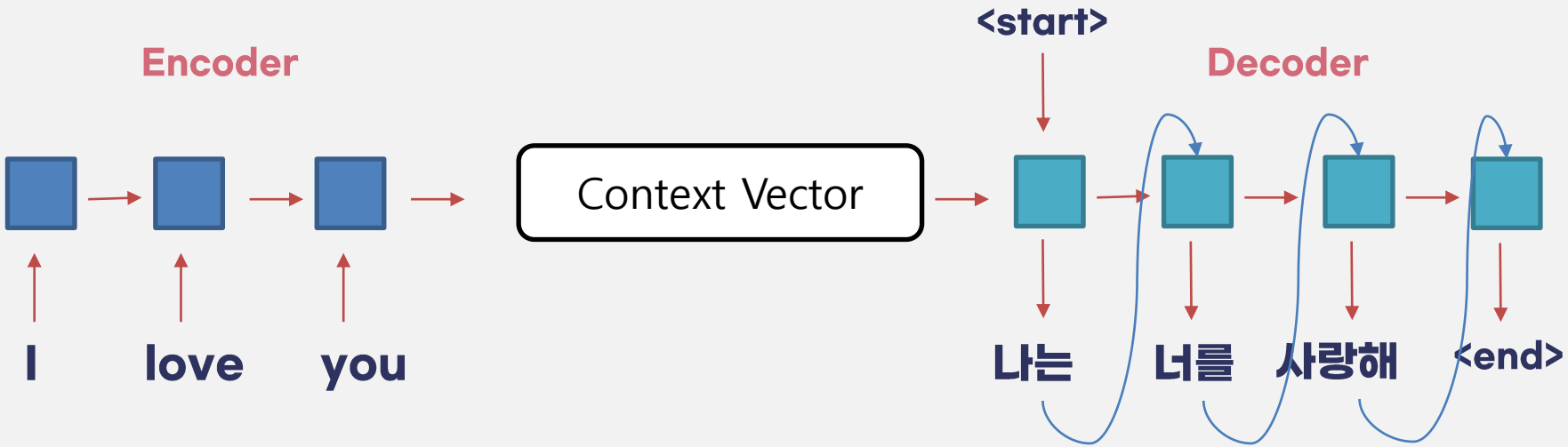
하나의 문맥 벡터가 소스 문장의
모든 정보를 가지고 있음

Attention 기반

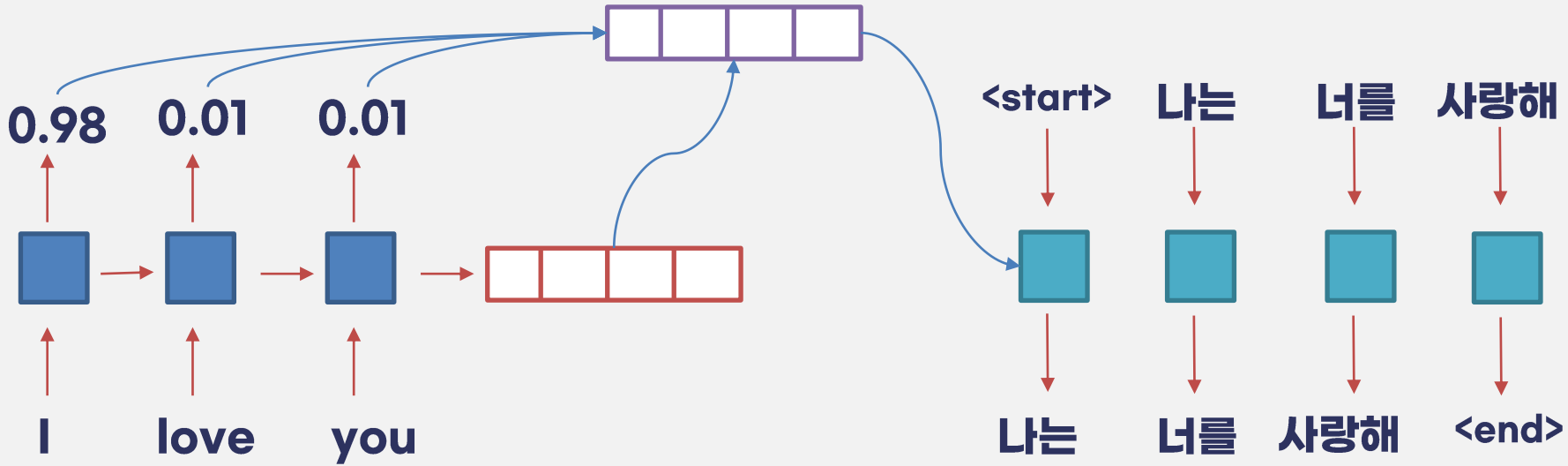
입력 시퀀스 전체에서 정보를 추출하는 방향



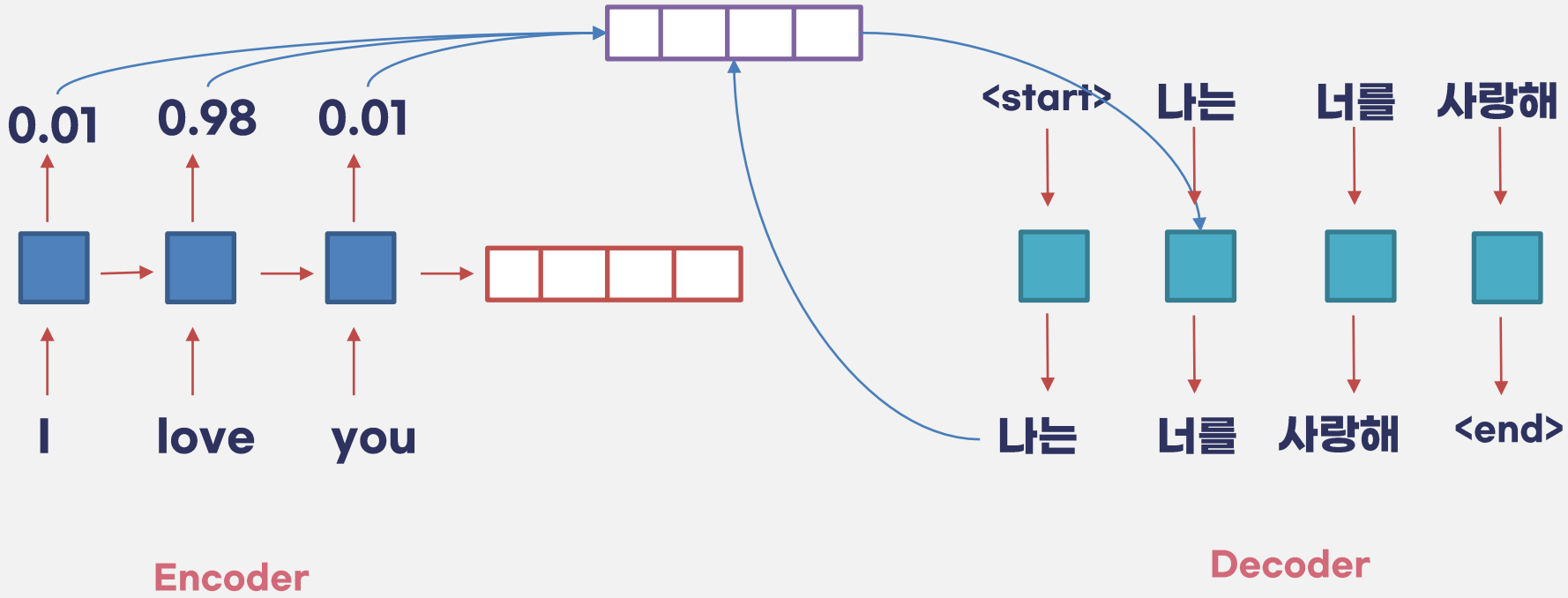
RNN based encoder decoder



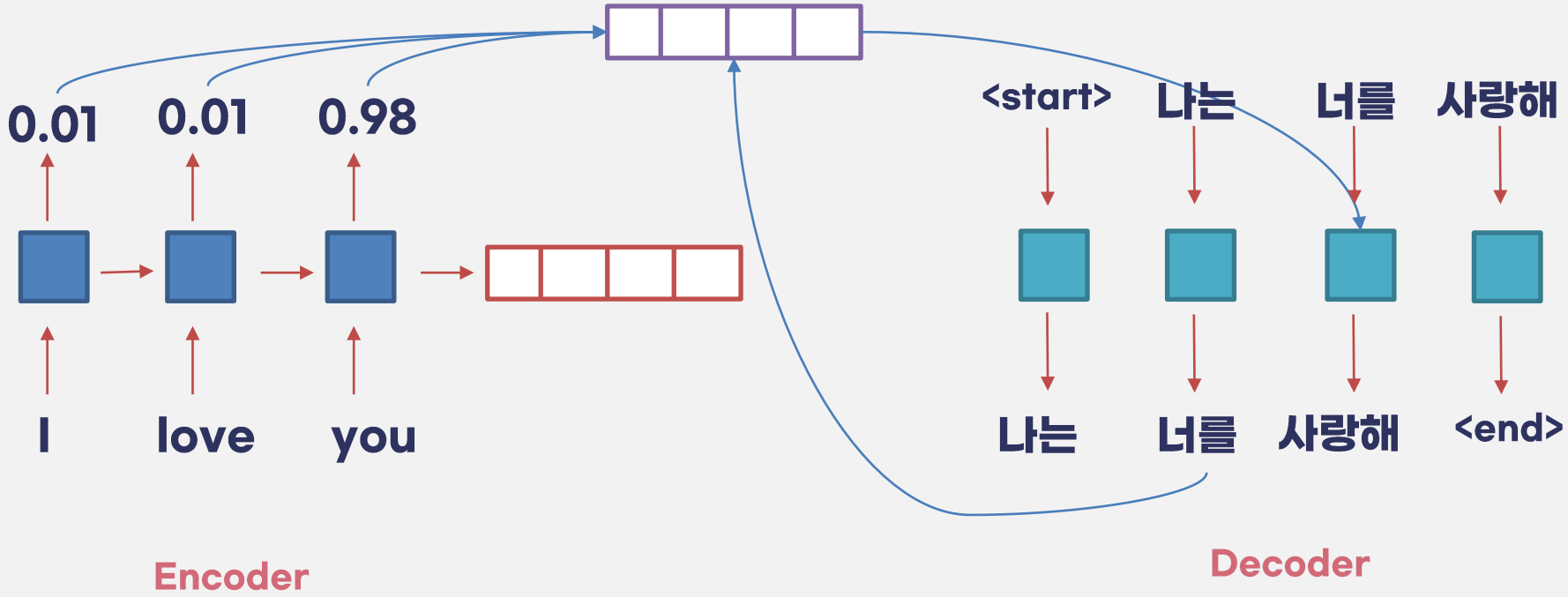
RNN based encoder decoder with attention



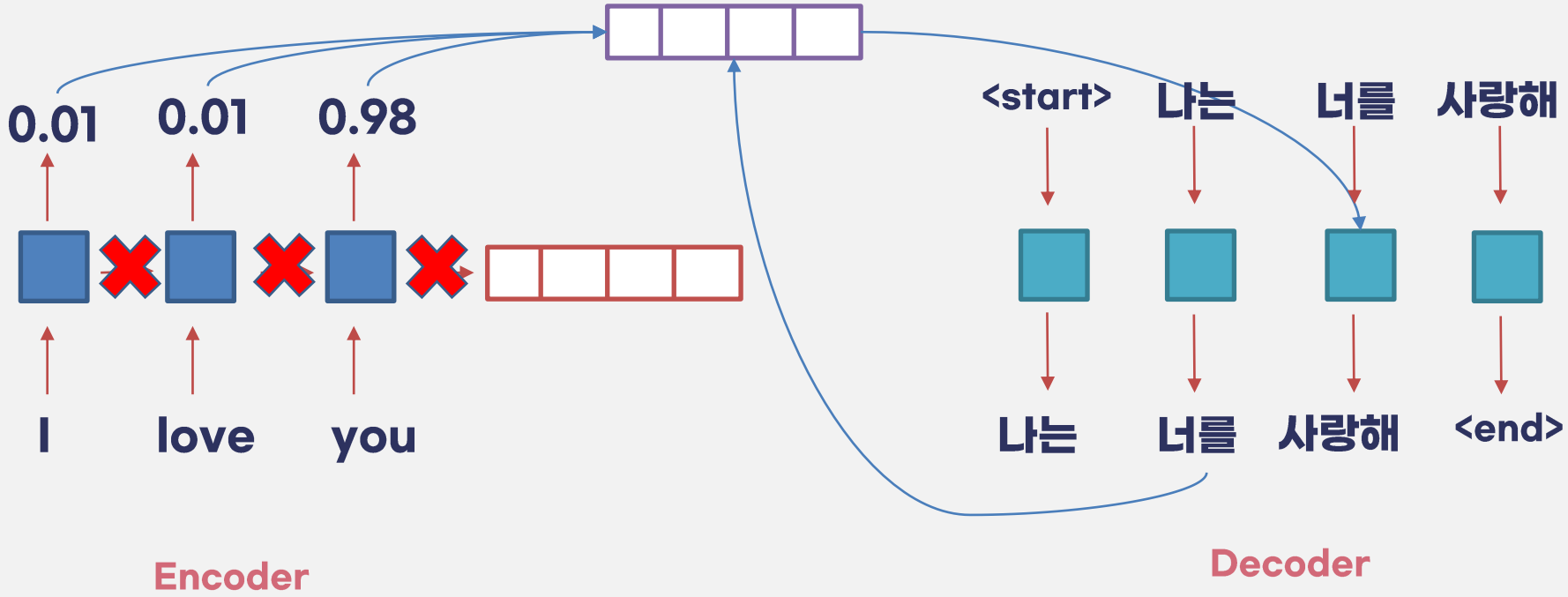
RNN based encoder decoder with attention



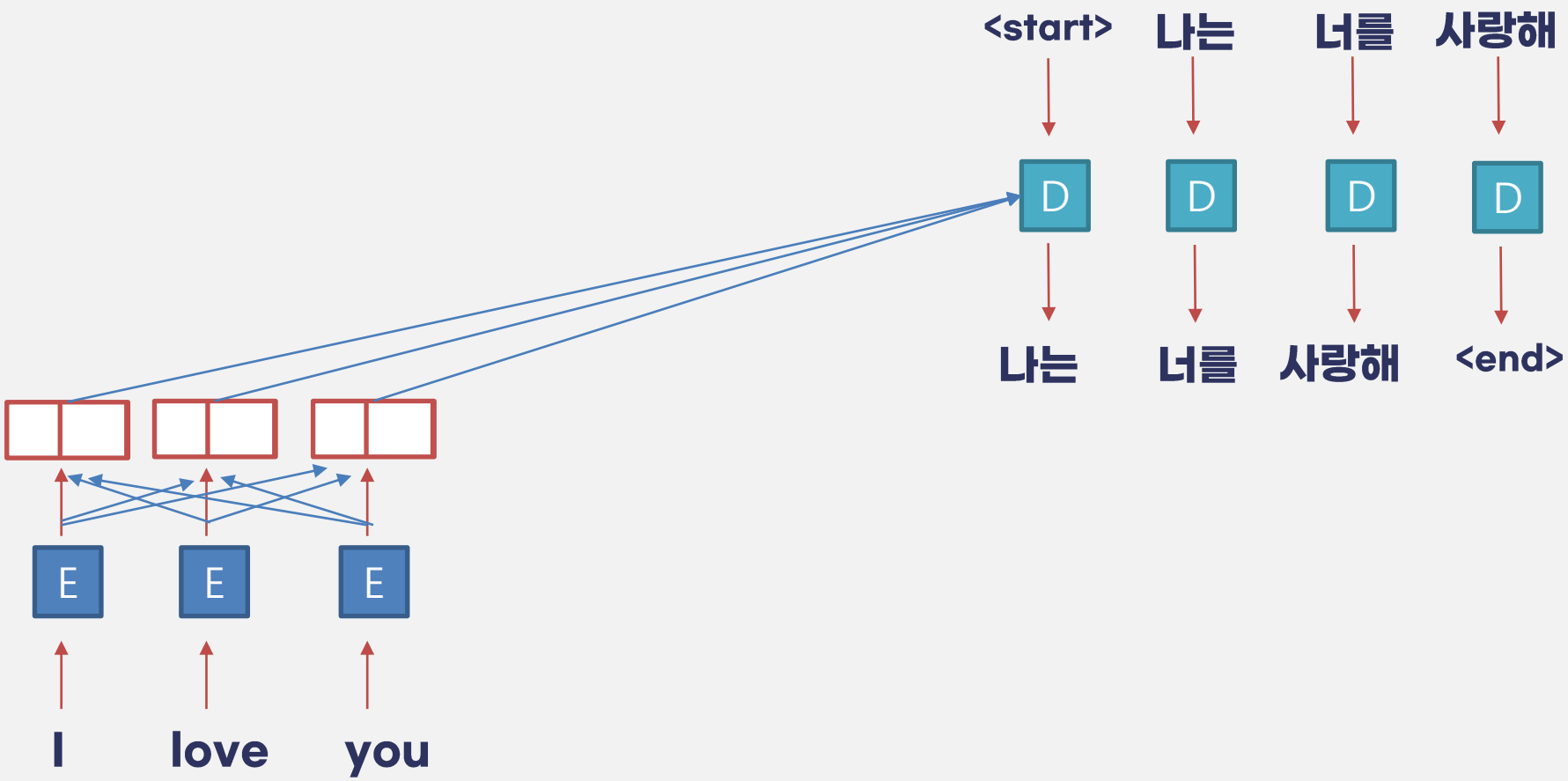
RNN based encoder decoder with attention



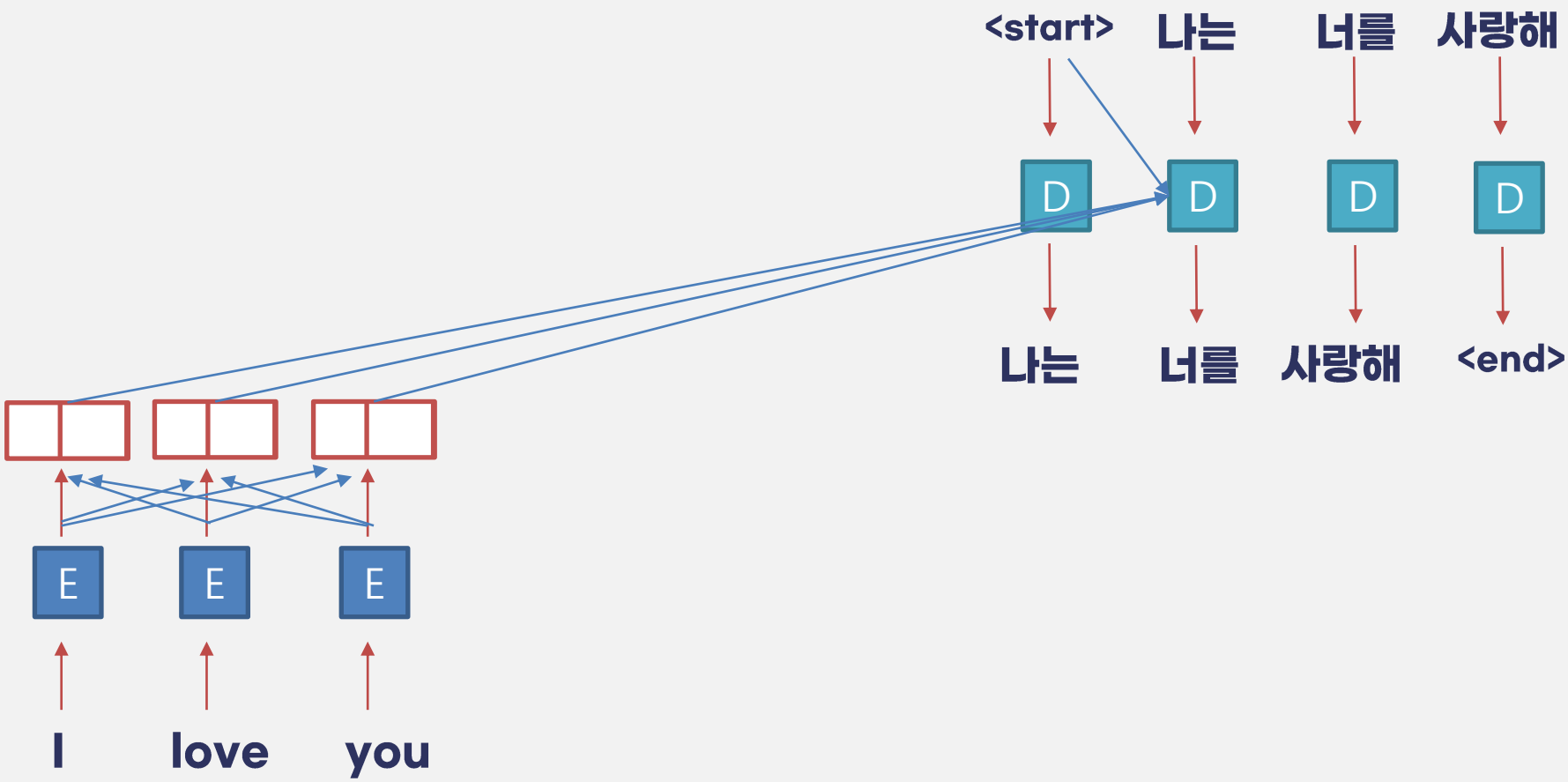
RNN based encoder decoder with attention



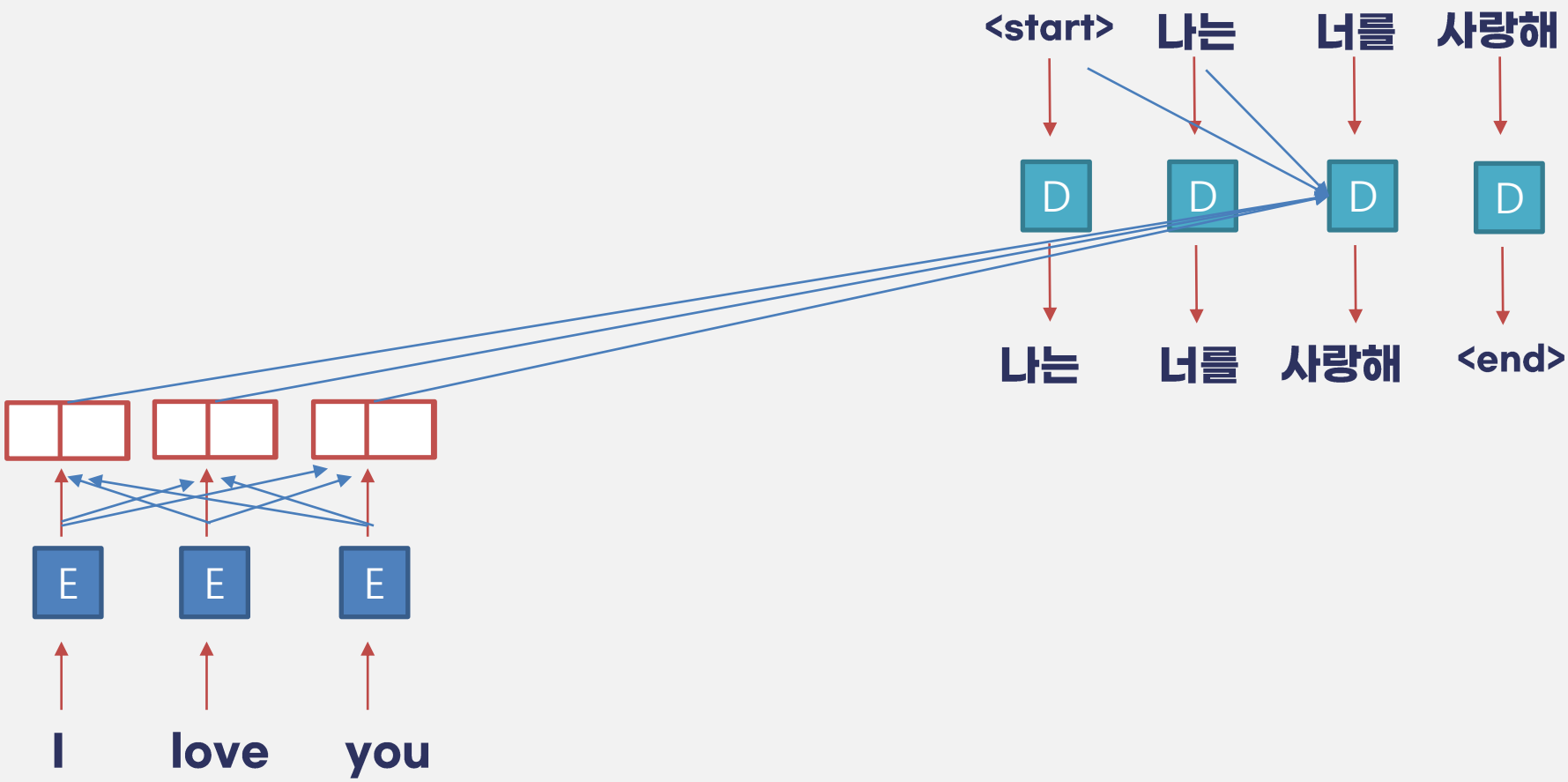
encoder decoder with attention



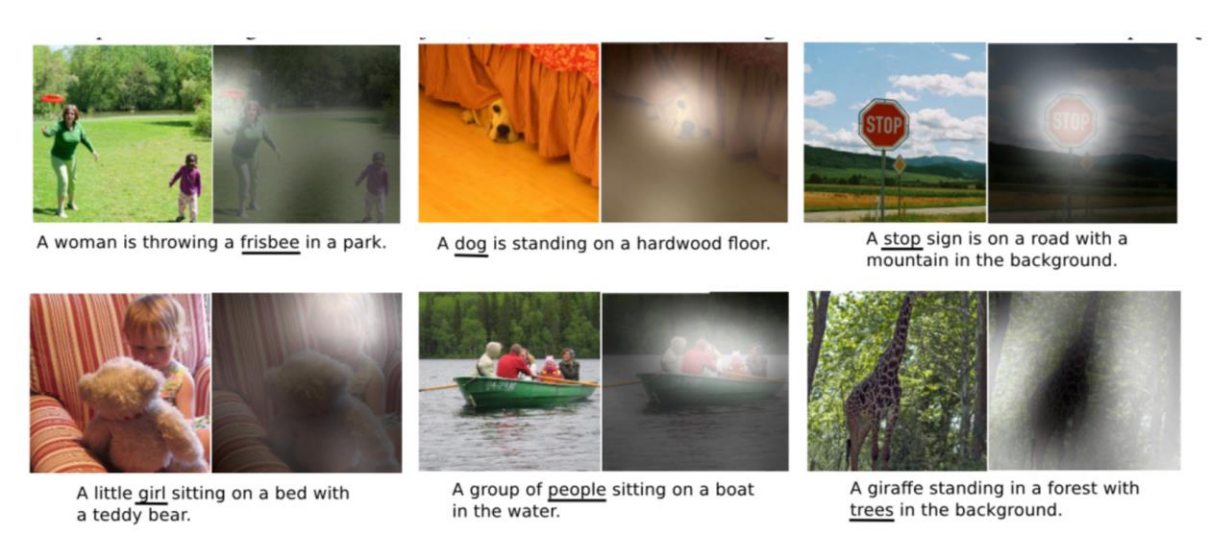
encoder decoder with attention

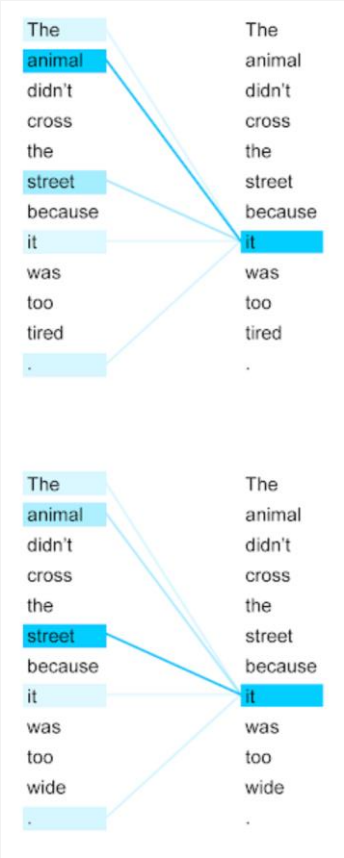


encoder decoder with attention



Attention의 직관적인 이해

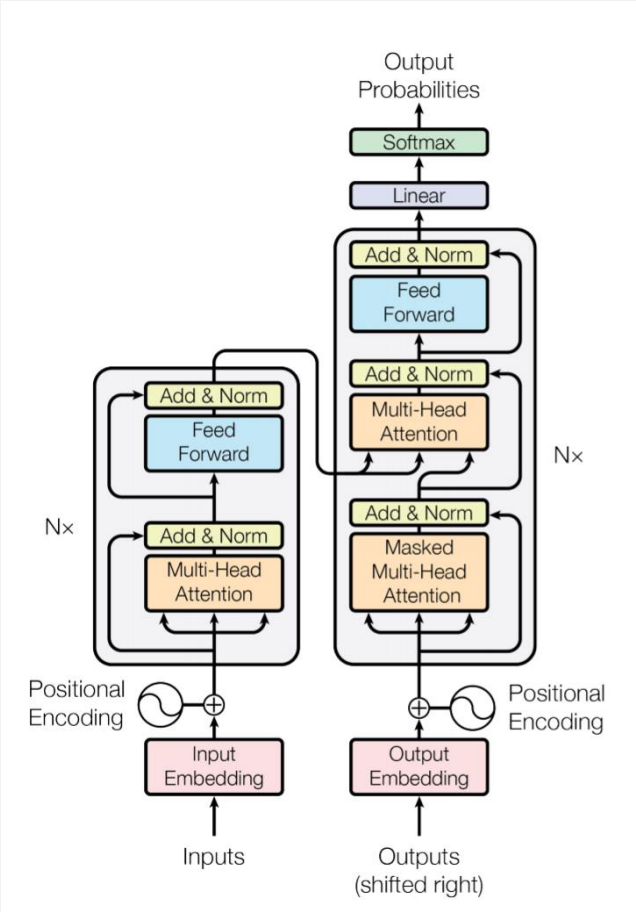




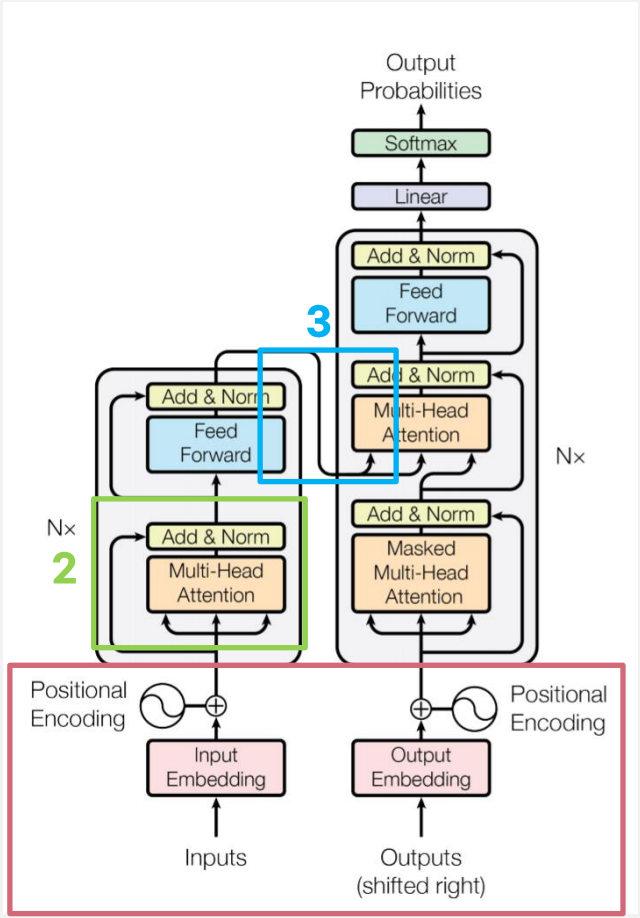
Transformer 구조

RNN X

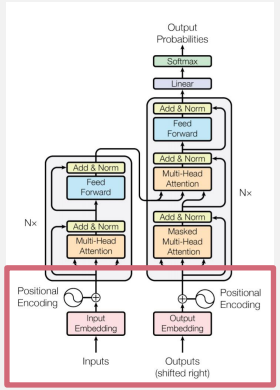
Only Attention



Transformer 구조



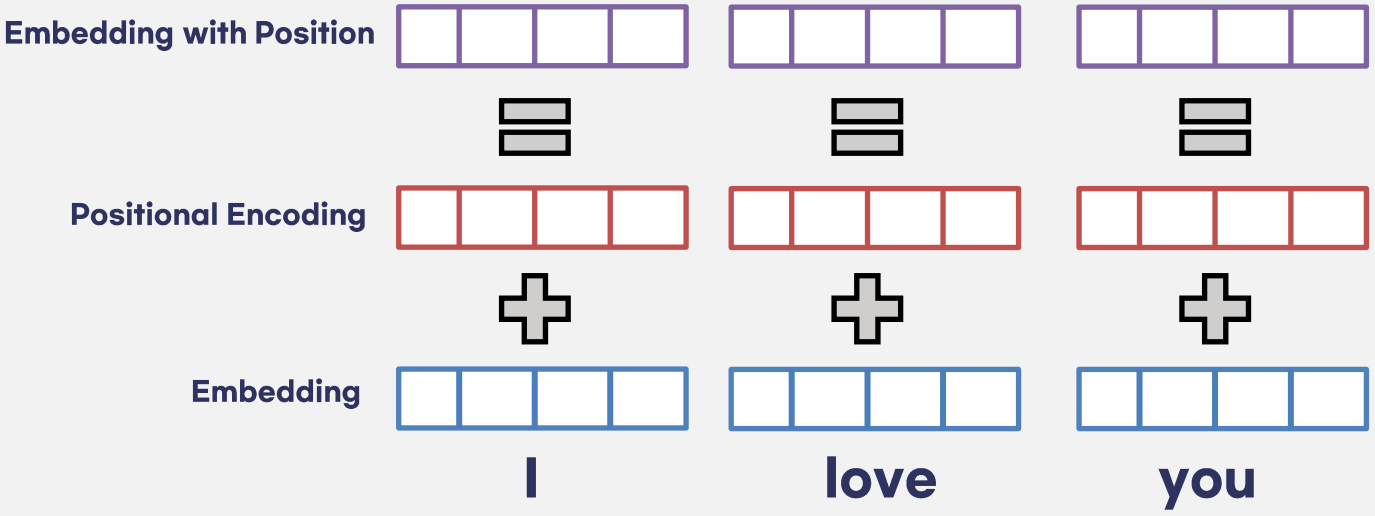
Transformer 구조 1 - Positional Encoding



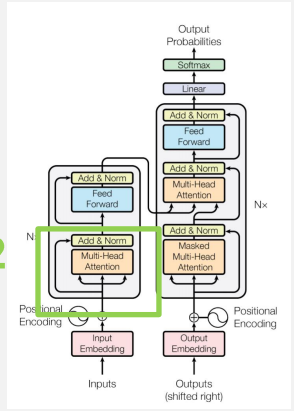
1

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

Pos : 토근의 위치 인덱스, I : 벡터의 차원 인덱스



Transformer 구조 2 - Encoding (Self-Attention)



2

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

Query : 현재 구하려는 단어
Key : 문장의 모든 단어
Value : 값

Query * Key^T | Score | Softmax | Value | Softmax * value

I — I * I — = 80 — 0.8

$\sqrt{d_k}$,

I * love — = 5 — 0.05

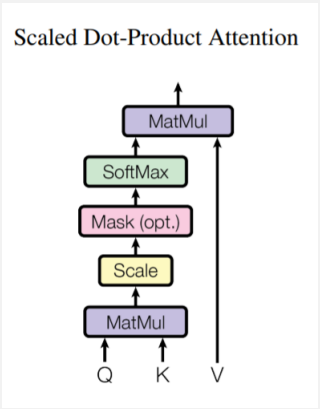
I * you — = 10 — 0.1

Mask(opt)

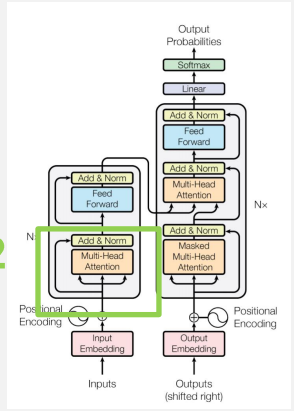
음수 무한의 값을 넣어 특정 값 무시

love

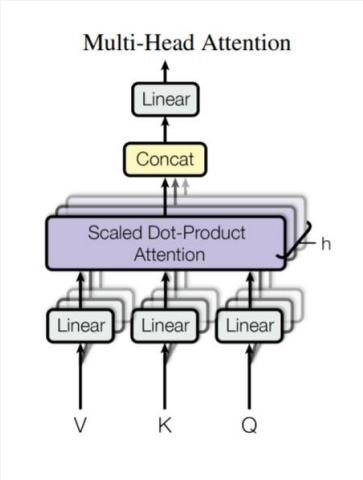
you



Transformer 구조 2 - Encoding (Multi Head Attention)



2



I

love

you

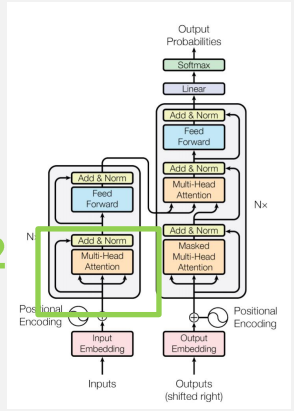
Head 1 (Attention)

Head 2 (Attention)

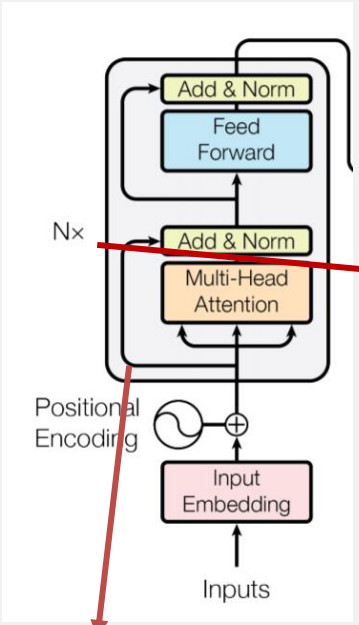
Head i (Attention)

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$
$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

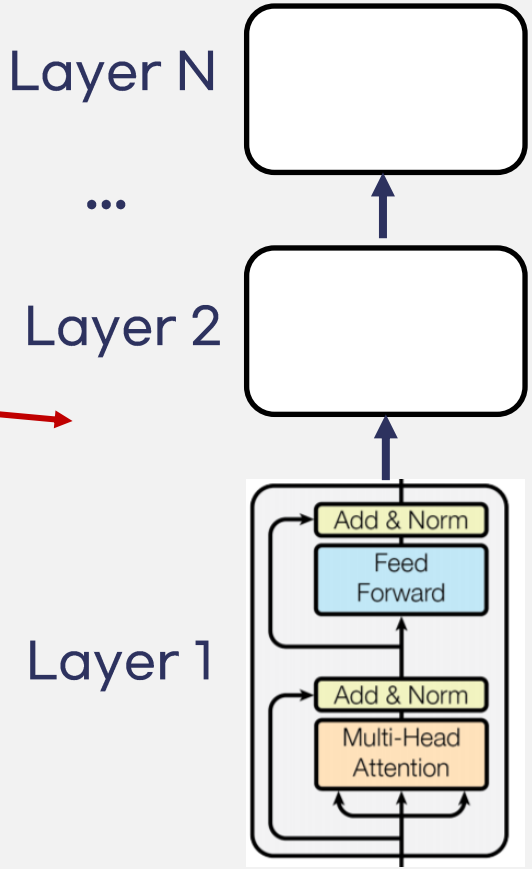
Transformer 구조 2 - Encoding (Residual Connection)



2

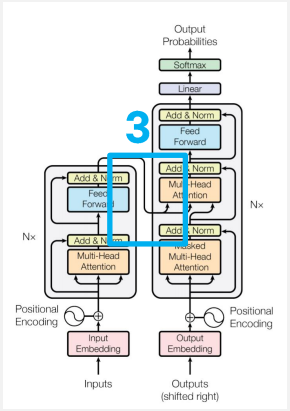


Residual



각 레이어는 서로 다른 파라미터를 가짐

Transformer 구조 3 - Encoding -> Decoding (Encoder - Decoder Attention)



Layer N

...

Layer 2

Layer 1

Encoding

Key, Value

Query

Layer N

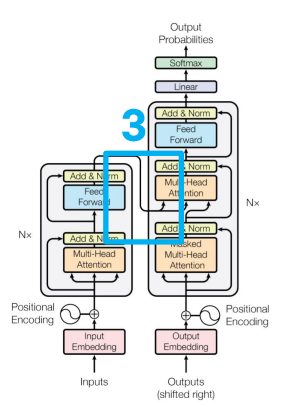
...

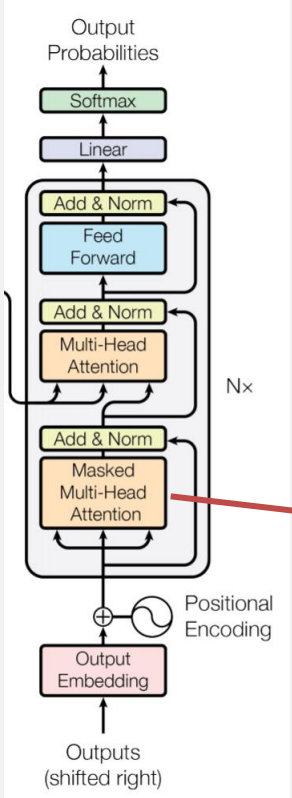
Layer 2

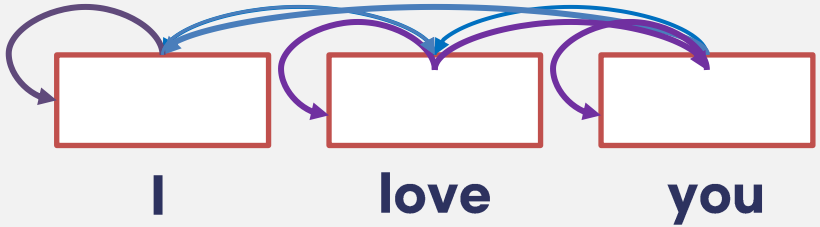
Layer 1

Decoding

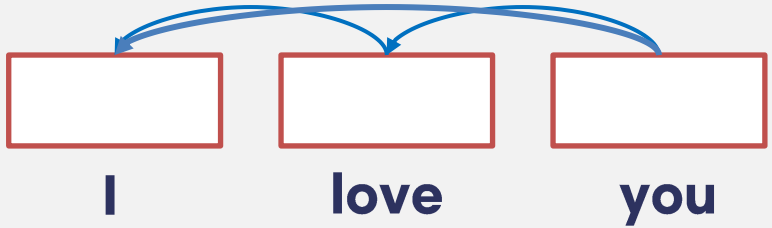
Transformer 구조 3 - Decoder (Mask Decoder Self-Attention)





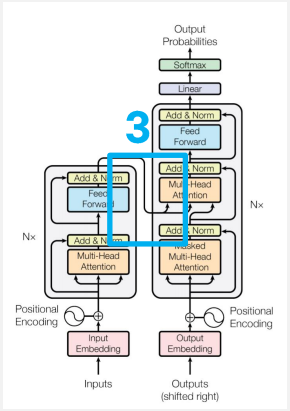


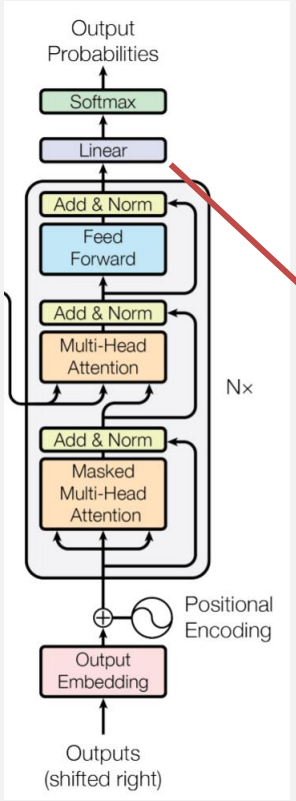
Encoder Self-Attention



Mask Decoder Self-Attention

Transformer 구조 3 - Decoding (Output & Label Smoothing)

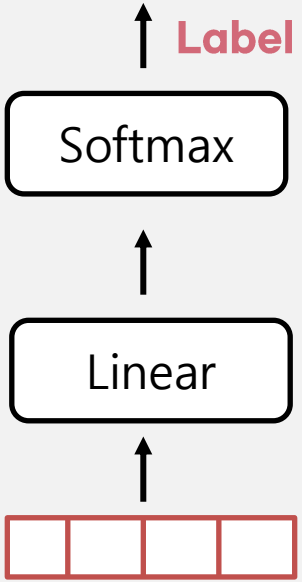




0.01 0.98 0.01 0 1 0

Label Smoothing

Label Smoothing X
(Only One hot Encoding)



Evaluation - 성능 개선

Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	**$3.3 \cdot 10^{18}$**	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	

그 당시 최고 성능

Evaluation - 최고의 파라미터

	N	d_{model}	d_{ff}	h	d_k	d_v	P_{drop}	ϵ_{ls}	train steps	PPL (dev)	BLEU (dev)	params $\times 10^6$
base	6	512	2048	8	64	64	0.1	0.1	100K	4.92	25.8	65
(A)				1	512	512				5.29	24.9	
				4	128	128				5.00	25.5	
				16	32	32				4.91	25.8	
				32	16	16				5.01	25.4	
(B)					16					5.16	25.1	58
					32					5.01	25.4	60
(C)	2									6.11	23.7	36
	4									5.19	25.3	50
	8									4.88	25.5	80
		256			32	32				5.75	24.5	28
		1024			128	128				4.66	26.0	168
			1024							5.12	25.4	53
			4096							4.75	26.2	90
(D)							0.0			5.77	24.6	
							0.2			4.95	25.5	
								0.0		4.67	25.3	
								0.2		5.47	25.7	
(E)	positional embedding instead of sinusoids									4.92	25.7	
big	6	1024	4096	16			0.3		300K	**4.33**	**26.4**	213

THANK YOU