Attention Is All You Need

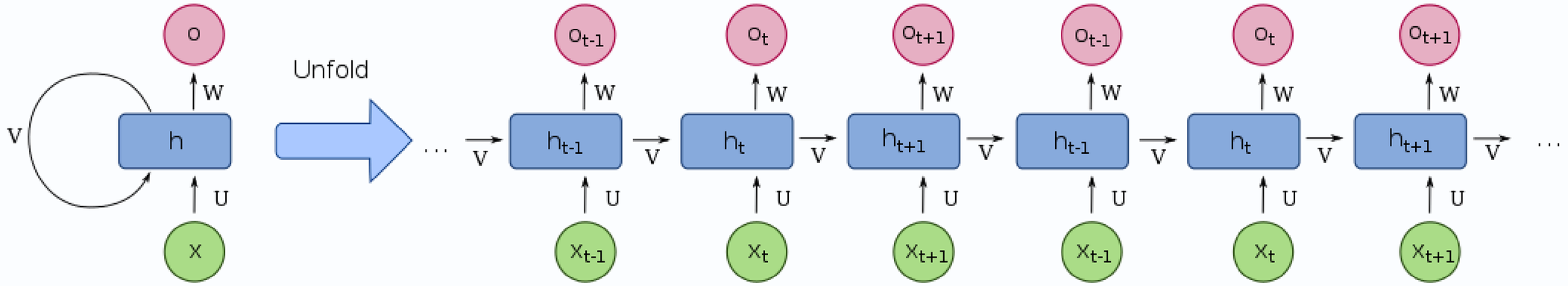
Ashish Vaswani et al., NIPS 2017

발표자 : 남경탁

발표일 : 2025-06-19

기존 자연어 처리

기존 모델 : RNN 계열 or CNN 계열 인코더-디코더(encoder-decoder) 구조

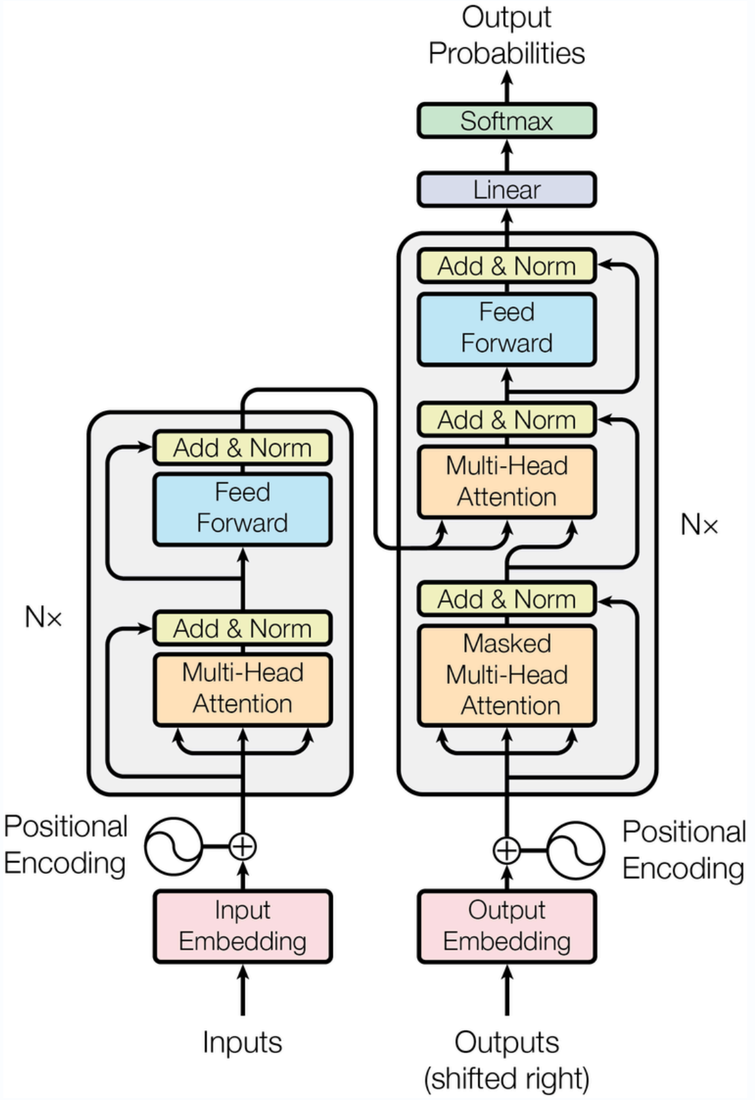


1. 병렬화 어려움(RNNs).
2. 장기 의존성 문제
3. Input 길이의 한계 RNNs : context vector 길이 한계
   CNNs : 연산량 폭발

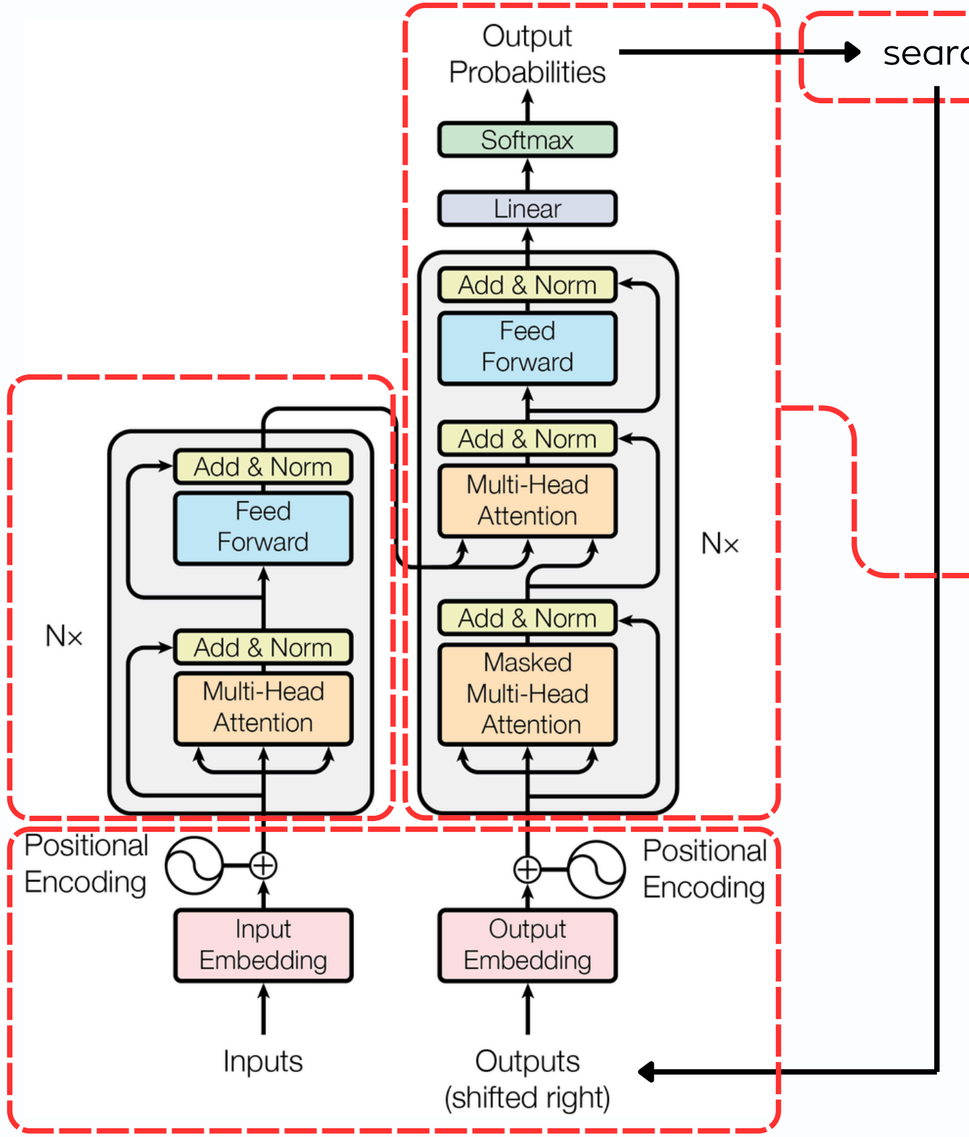
그래서 transformer는...

1. attention을 활용해 순환에서 해방.
   → input 토큰을 한번에 병렬로 처리가능

2. 입력 간 거리를 상수화
   → 단어간의 dependencies를 matrix 형태로 활용
   → 단어 위치는 Positional Encoding
   → 장기 의존성문제 해결



Transformer의 구조



search

4. 확률을 보고 정답 단어를 유추

2. encoder(입력 문장의 해석)

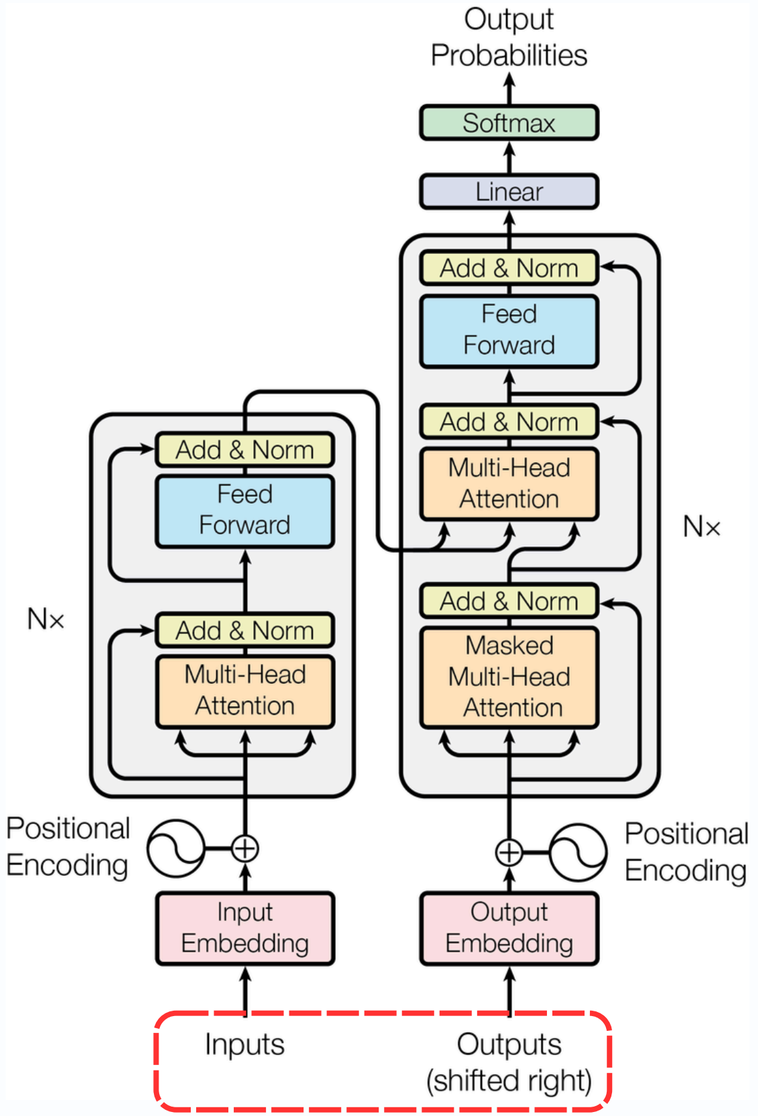
3. Decoder(실질적 문장 생성)

1. embeding(입력 문장 벡터화)

Input

- encoder : (예시 “We study papers every Thursday”)
  - {x1, x2, x3, x4, x5, <PAD>, <PAD>, </s>}
  - 기존 bag of words에 존재하지 않은 단어 등장시
    → {x1, x2, <UNK>, x4, x5, <PAD>, <PAD>, </s>}

- decoder : (예시 “우리는 매주 목요일에 논문을 공부한다”)
  - {<s>, x1, x2, x3, x4, x5, <PAD>, </s>}



Token Embedding & Positional Encoding

	Embedding (4D)			
"We"	0.2	0.1	0.3	0.4
"study"	0.5	0.1	0.4	0.3
"papers"	0.3	0.4	0.2	0.5
"every"	0.4	0.3	0.3	0.2
"Thursday"	0.6	0.2	0.1	0.3

+

	Positional (4D)			
"We"	0	1	0	1
"study"	0.8	0.5	0	1
"papers"	0.9	-0.4	0	1
"every"	0.1	-1	0	1
"Thursday"	-0.8	-0.7	0	1

=

	Sum			
"We"	0.2	1.1	0.3	1.4
"study"	1.3	0.6	0.4	1.3
"papers"	1.2	0	0.2	1.5
"every"	0.5	-0.7	0.3	1.2
"Thursday"	-0.2	-0.5	0.1	1.3

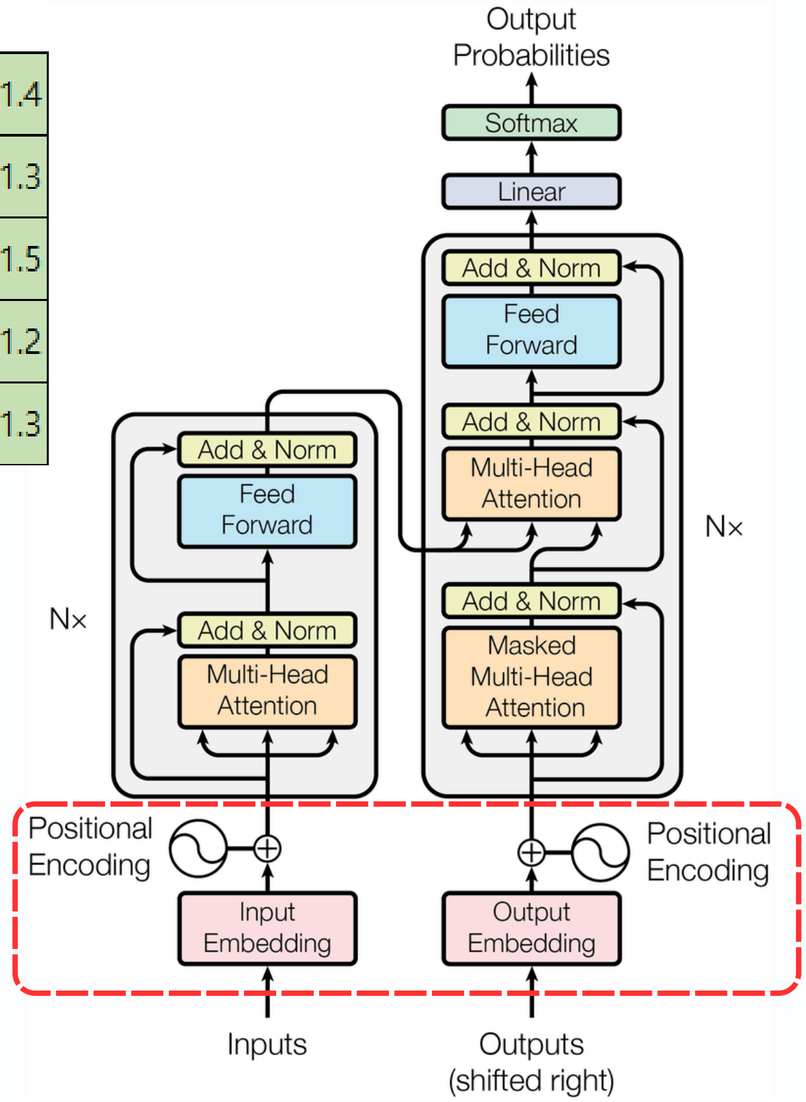
look-up table에서
단어에 맞는 임베딩 벡터 불러옴

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{\text{model}}}}}\right)$$

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

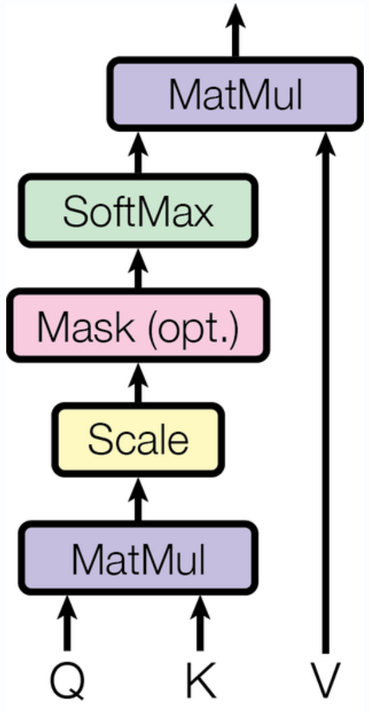
위치 0 (첫 번째 단어: `"We"`)

$$PE(0) = \left[\sin\left(\frac{0}{10000^{0/4}}\right), \cos\left(\frac{0}{10000^{0/4}}\right), \sin\left(\frac{0}{10000^{2/4}}\right), \cos\left(\frac{0}{10000^{2/4}}\right)\right] = [0, 1, 0, 1]$$

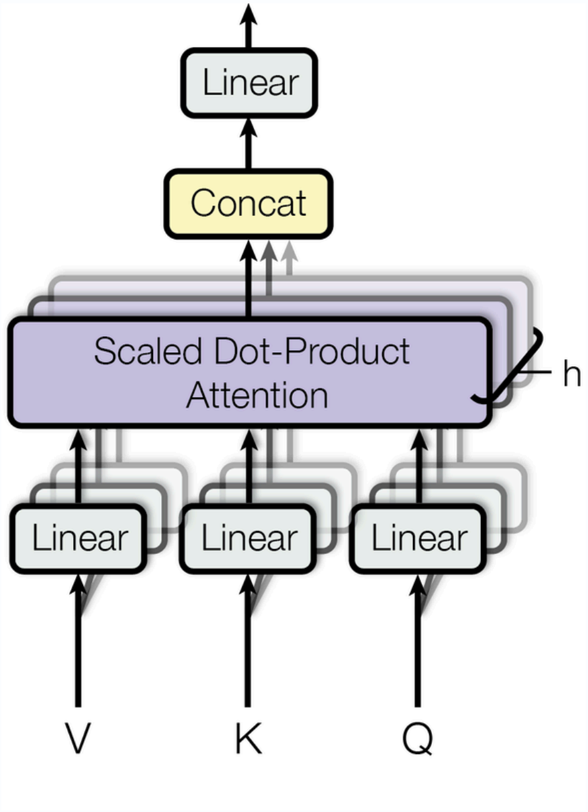


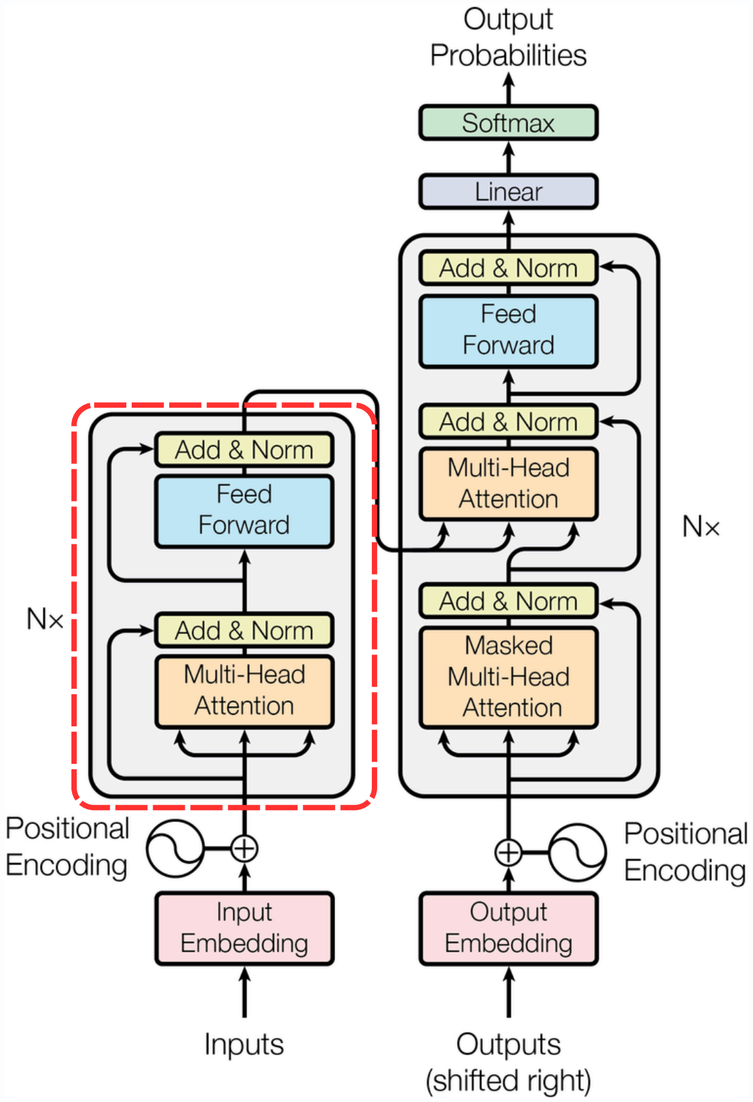
Encoder

Scaled Dot-Product Attention



Multi-Head Attention





Scaled Dot-Product Attention

$$\mathbf{Q} = \mathbf{X} \cdot \mathbf{W}_Q, \quad \mathbf{K} = \mathbf{X} \cdot \mathbf{W}_K, \quad \mathbf{V} = \mathbf{X} \cdot \mathbf{W}_V$$

X				
"We"	0.2	1.1	0.3	1.4
"study"	1.3	0.6	0.4	1.3
"papers"	1.2	0	0.2	1.5
"every"	0.5	-0.7	0.3	1.2
"Thursday"	-0.2	-0.5	0.1	1.3

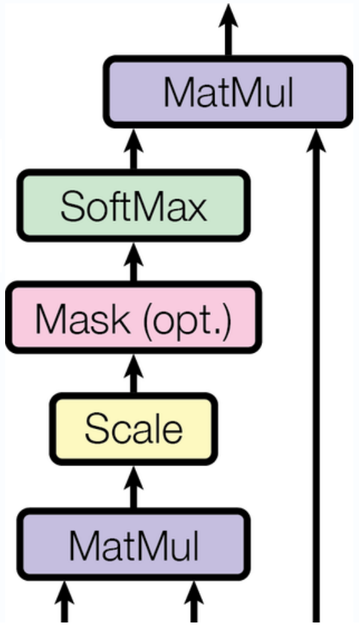
•

W_Q		
0.1	0.2	0.3
0.3	0.4	0.2
0.5	0.6	0.7
0.7	0.8	0.3

=

Q			
"We"	1.5	1.8	0.9
"study"	1.4	1.8	1.2
"papers"	1.3	1.6	1
"every"	0.8	1	0.6
"Thursday"	0.8	0.9	0.3

$$\alpha_{ij} = \frac{\exp(\text{score}_{ij})}{\sum_{j'} \exp(\text{score}_{ij'})}$$



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

Query			
"We"	1.5	1.8	0.9
"study"	1.4	1.8	1.2
"papers"	1.3	1.6	1
"every"	0.8	1	0.6
"Thursday"	0.8	0.9	0.3

•

Key.T					
	"We"	"study"	"papers"	"every"	"Thursda
	2.3	2.7	2.4	1.2	0.9
	1.3	1.2	0.9	0.5	0.4
	1.4	1.5	0.8	-0.2	-0.5

=

Attention Score					
	"We"	"study"	"papers"	"every"	"Thursday"
"We"	7	7.6	5.9	2.5	1.6
"study"	7.3	7.8	5.9	2.4	1.4
"papers"	6.3	6.8	5.2	2.1	1.3
"every"	4	4.3	3.3	1.4	0.9
"Thursday"	3.4	3.7	2.9	1.3	0.9

→

Attention Prob					
	"We"	"study"	"papers"	"every"	"Thursday"
"We"	0.9	0.5	0.6	0.9	0.2
"study"	0.7	0.7	0.9	0.2	0.3
"papers"	0.7	0.1	0.5	0.8	0.9
"every"	0.7	0.1	0.2	0.9	0.8
"Thursday"	0.6	0.9	0.1	0.3	0.4

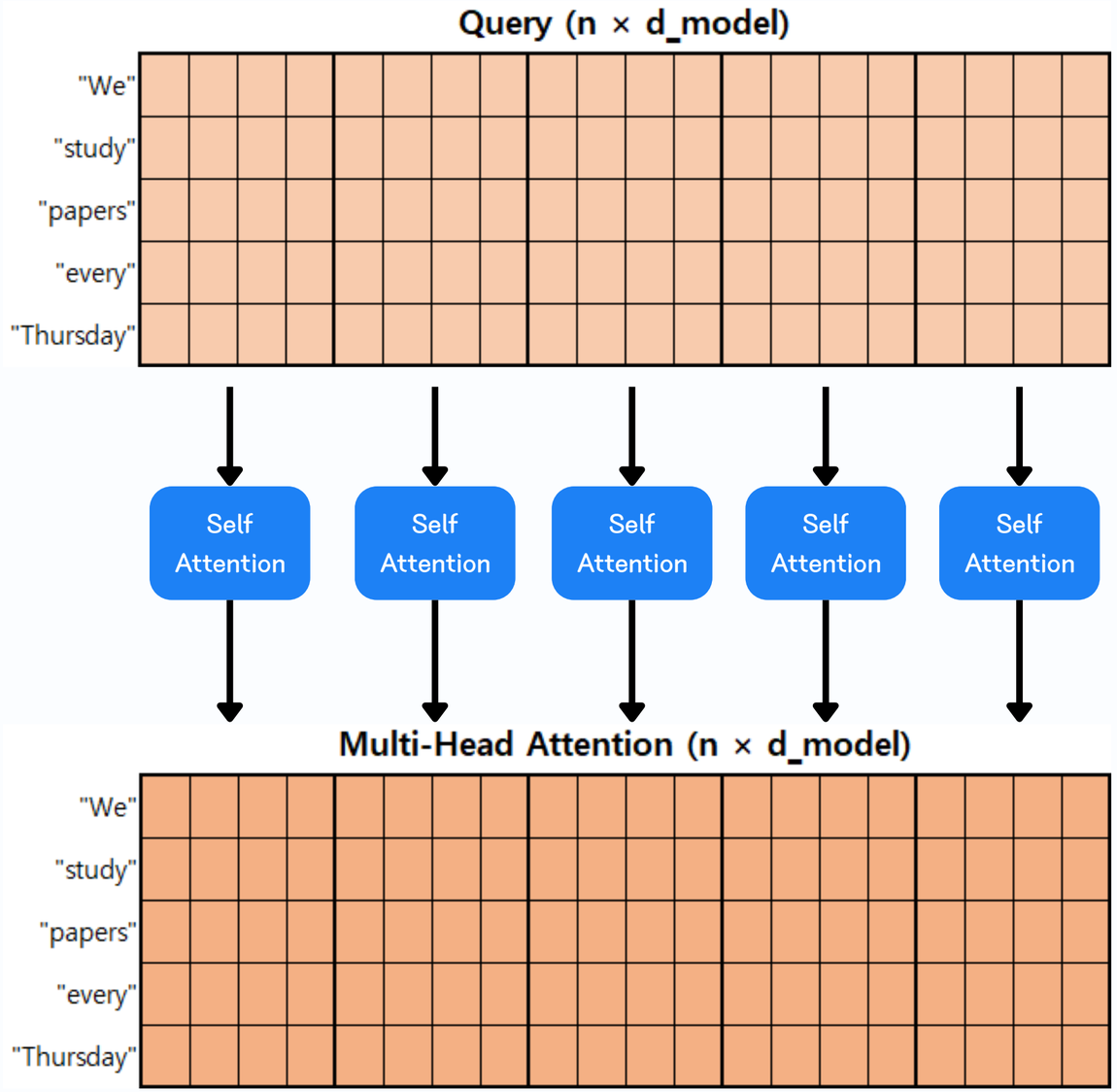
•

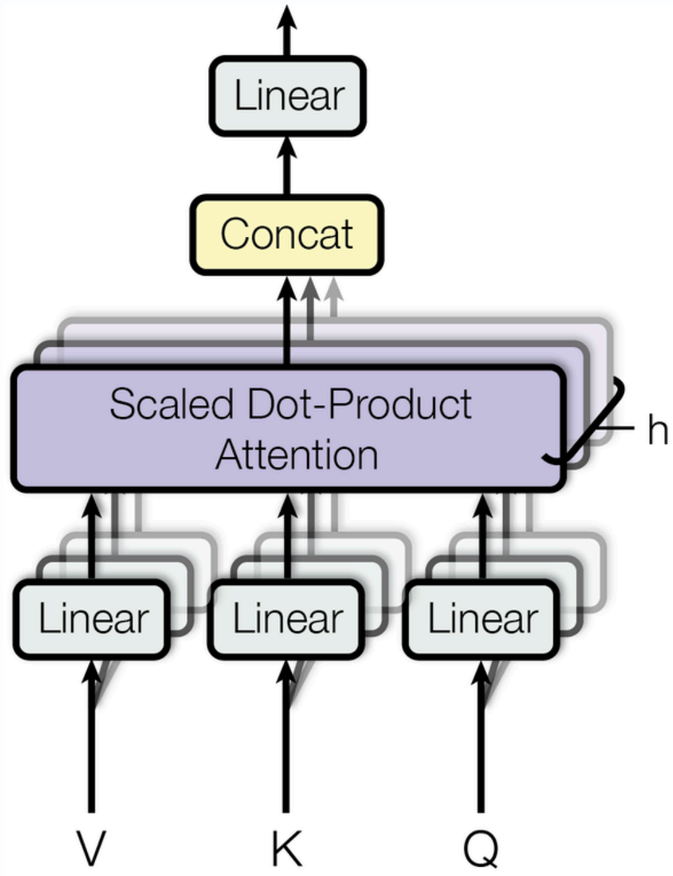
Value			
"We"	0.8	2.3	1.9
"study"	0.6	2.6	2.4
"papers"	0.2	2.3	1.7
"every"	-0.2	1.4	0.4
"Thursday"	-0.2	1	-0.1

=

Attention			
"We"	1	6.4	4.4
"study"	1.1	6.1	4.7
"papers"	0.4	4.9	2.6
"every"	0.3	4.4	2.3
"Thursday"	0.9	4.8	3.5

Multi-Head Attention

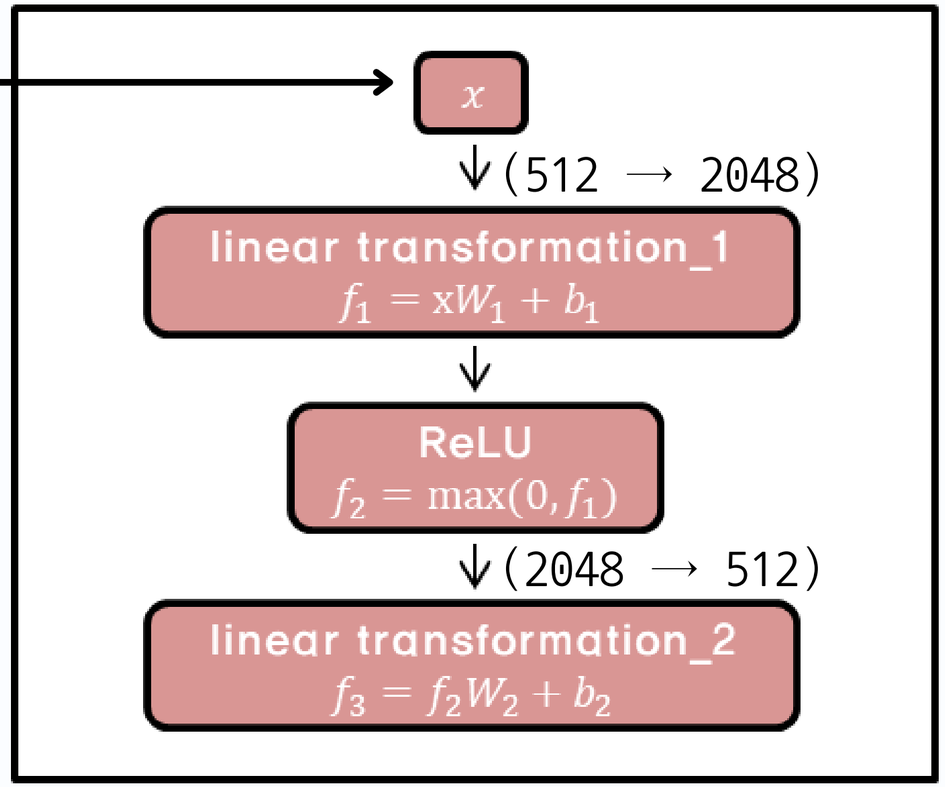




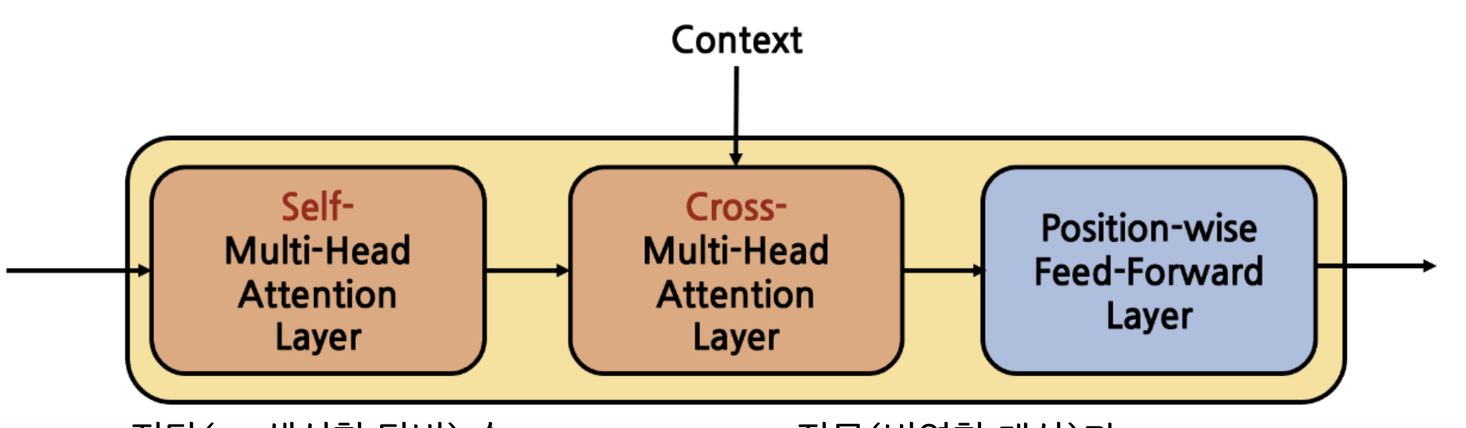
Position-wise Feed-Forward Networks

	Q		
"We"	1	6.4	4.4
"study"	1.1	6.1	4.7
"papers"	0.4	4.9	2.6
"every"	0.3	4.4	2.3
"Thursday"	0.9	4.8	3.5

$$\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

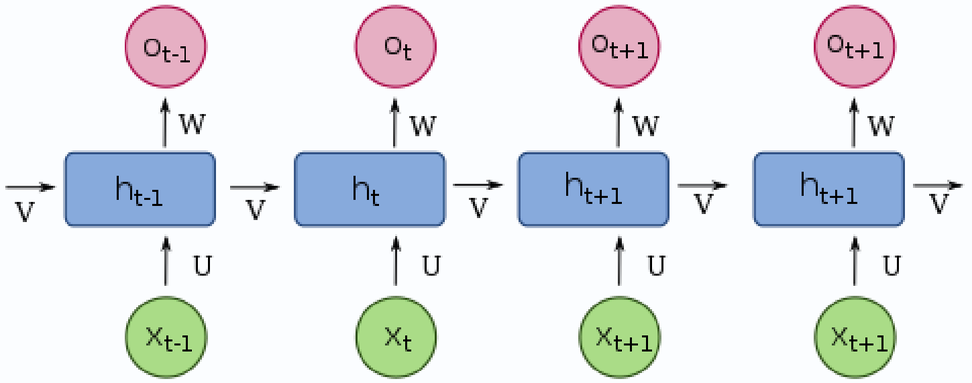


Decoder : Self-Multi-Head Attention Layer

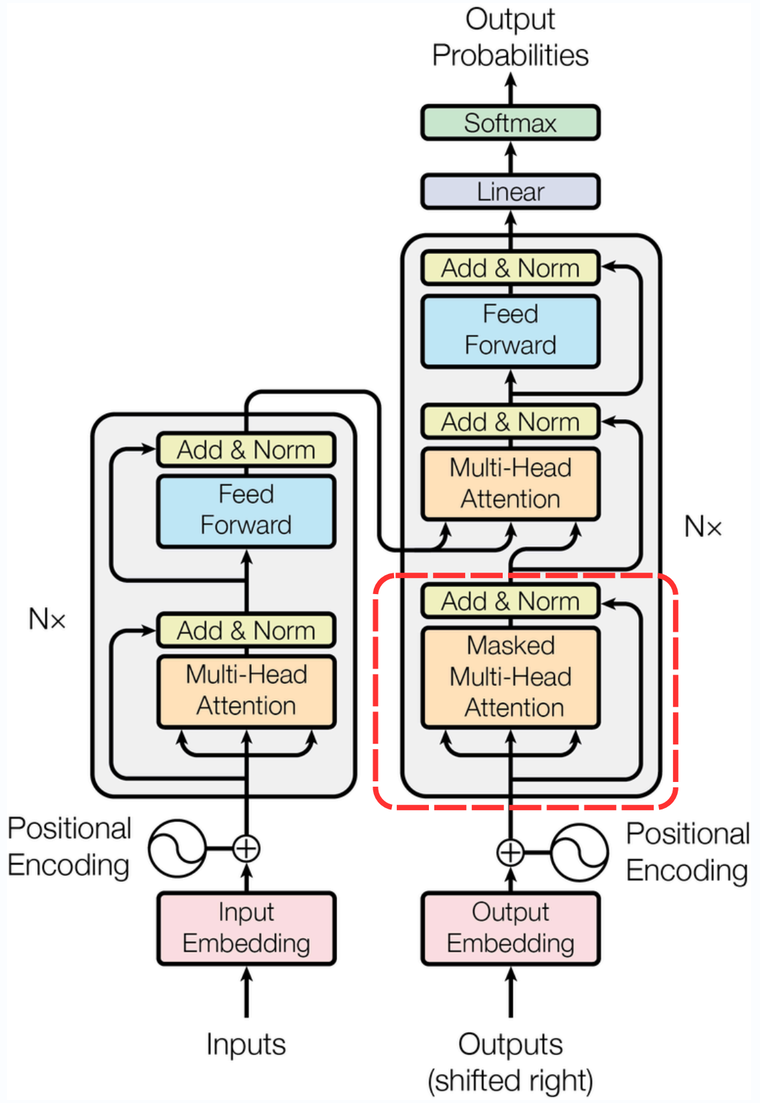


정답(or 생성한 답변) 속
단어들의 관계성 확인

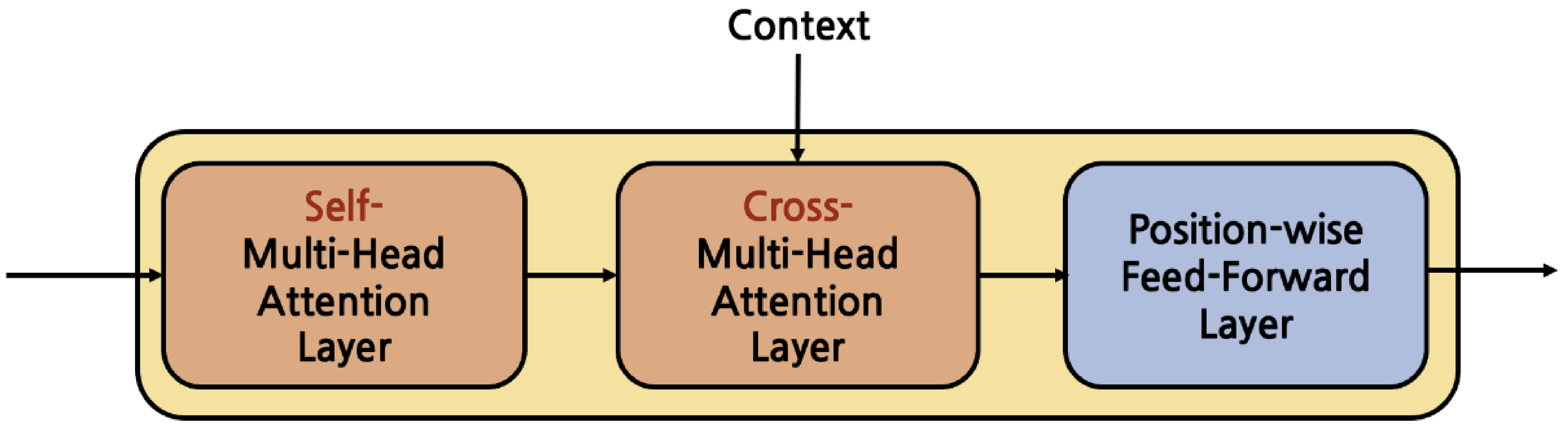
질문(번역할 대상)과
앞선 답변을 통해 다음 단어 확률 표 생성



	<s>	나는	매주	목요일에	논문을	공부한다
"<s>"	0.2	0	0	0	0	0
"우리는"	0.2	0.4	0	0	0	0
"매주"	0.4	0.3	0.6	0	0	0
"목요일에"	0.7	0.2	0.7	0.4	0	0
"논문을"	0.6	0.1	0.2	0.6	0.1	0
"공부한다"	0.2	0.5	0.3	0.5	0.6	0.4

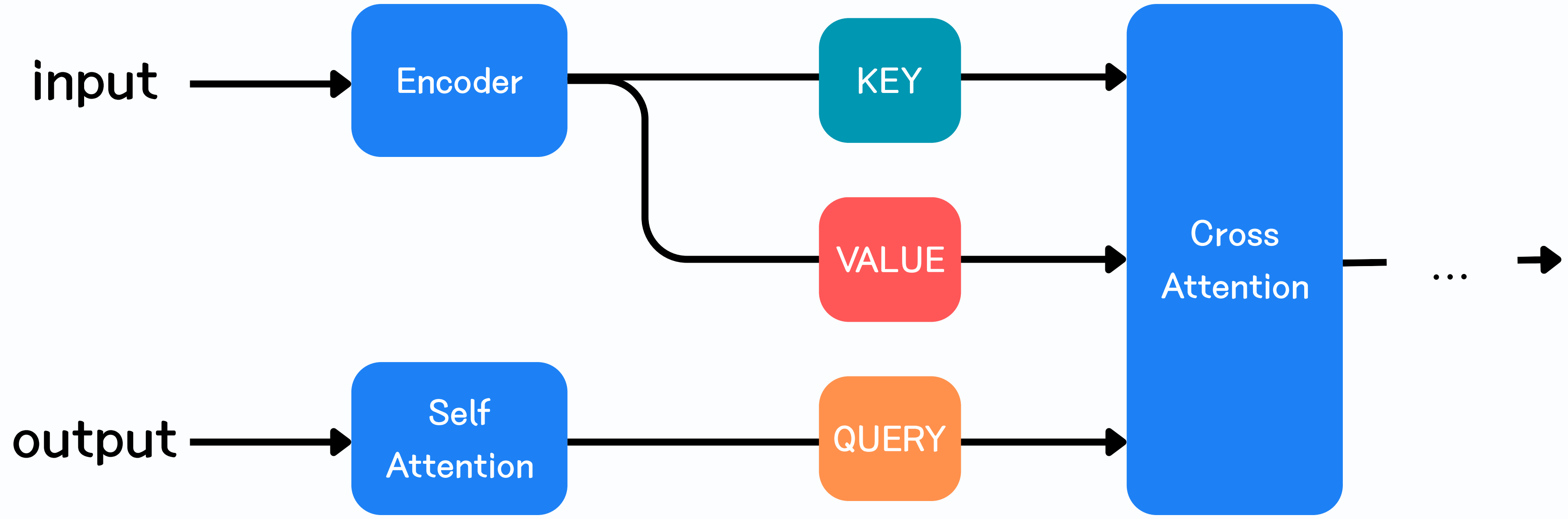


Decoder : Cross-Multi-Head Attention Layer



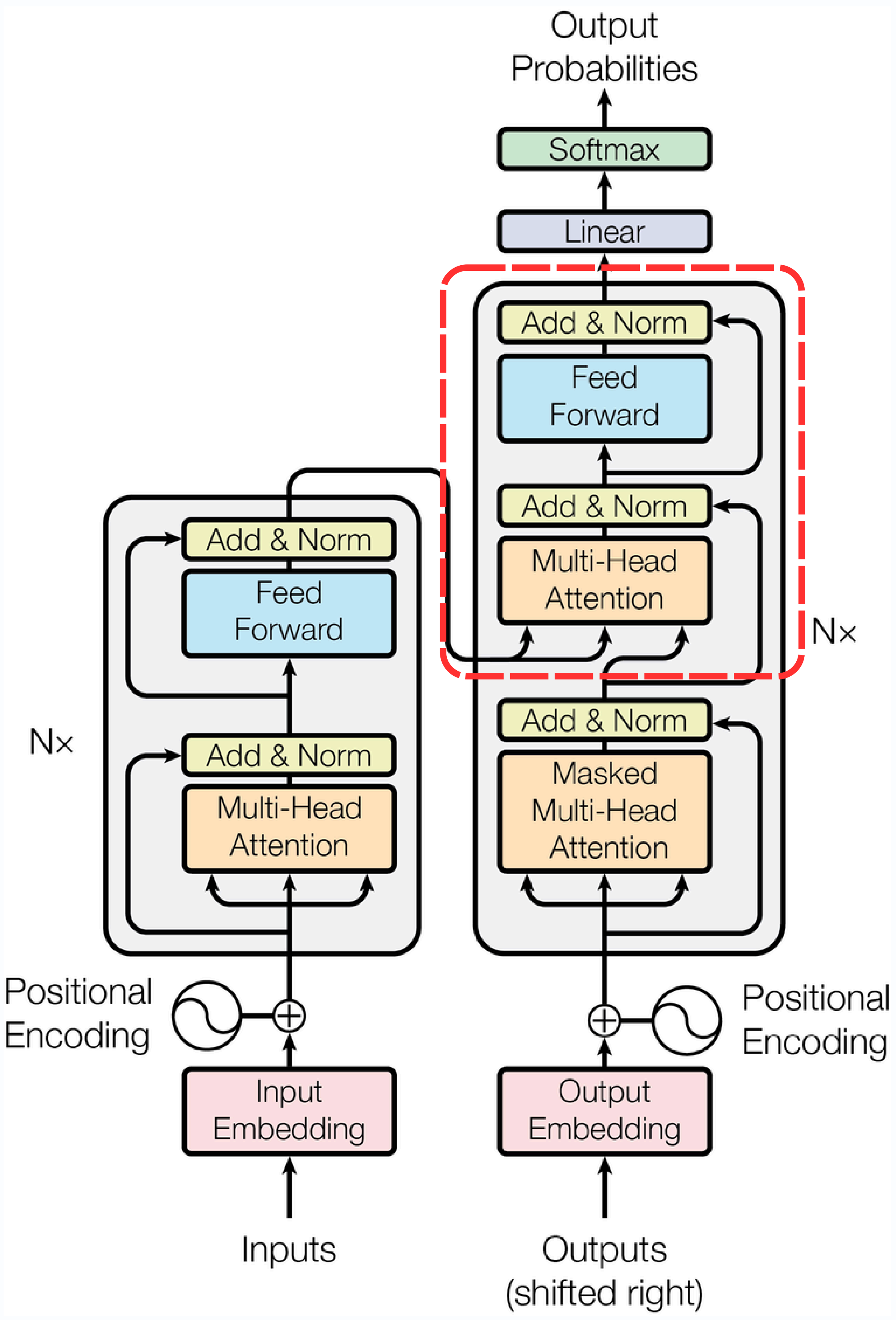
정답(or 생성한 답변) 속
단어들의 관계성 확인

질문(번역할 대상)과
앞선 답변을 통해 다음 단어 확률 표 생성



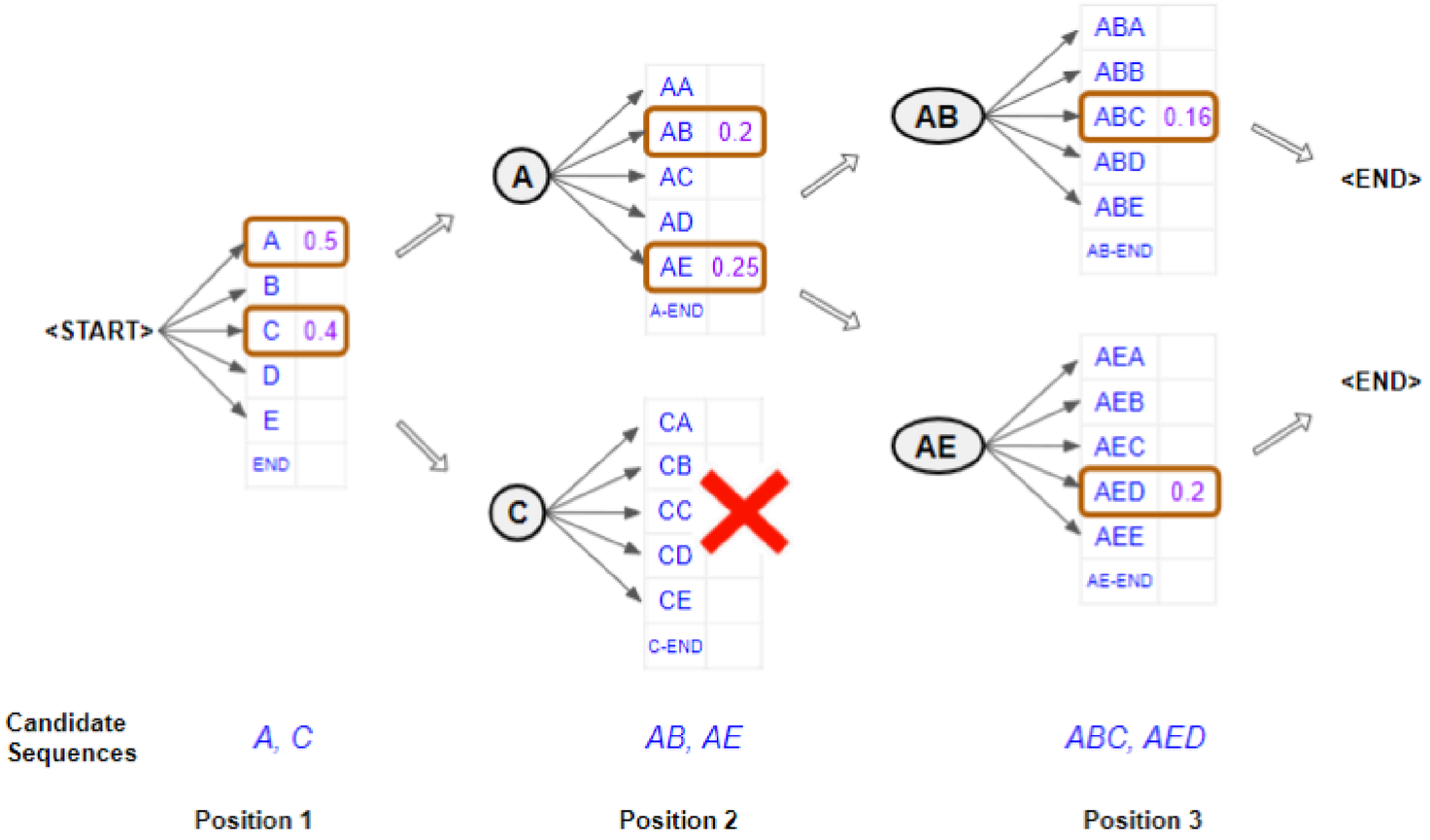
	"우리는"	"논문을"	"공부한다"	"먹는다"	"좋아한다"
t=1	0.6	0.1	0.15	0.1	0.1
t=2	0.2	0.6	0.05	0.1	0.1
t=3	0.1	0.1	0.8	0.1	0.1
⋮	⋮	⋮	⋮	⋮	⋮

t는 문장 안에서의 위치(시점)



Beam Search (문장 생성)

- greedy decoding : 확률 분포에서 가장 높은 확률을 가진 단어를 단순히 선택하여 다음 단어로 결정하는 방식
- Beam Search : 가장 가능성 있는 여러 개의 부분 가설(partial hypotheses)을 동시에 유지하며 탐색



성능 비교

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

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**	$2.3 \cdot 10^{19}$	

	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

Parser	Training	WSJ 23 F1
Vinyals & Kaiser el al. (2014) [37]	WSJ only, discriminative	88.3
Petrov et al. (2006) [29]	WSJ only, discriminative	90.4
Zhu et al. (2013) [40]	WSJ only, discriminative	90.4
Dyer et al. (2016) [8]	WSJ only, discriminative	91.7
Transformer (4 layers)	WSJ only, discriminative	91.3
Zhu et al. (2013) [40]	semi-supervised	91.3
Huang & Harper (2009) [14]	semi-supervised	91.3
McClosky et al. (2006) [26]	semi-supervised	92.1
Vinyals & Kaiser el al. (2014) [37]	semi-supervised	92.1
Transformer (4 layers)	semi-supervised	92.7
Luong et al. (2015) [23]	multi-task	93.0
Dyer et al. (2016) [8]	generative	93.3

정리...

핵심 요약

Transformer는 RNN, CNN 없이 Attention만으로 시퀀스를 처리
병렬화 가능 + 긴 거리 의존성 학습 효과적
학습 속도, 성능 향상
해석 가능한 구조 (attention weights 시각화 가능)

한계 및 단점

시간복잡도 $O(N^2)$: 입력 길이가 길어질수록 연산량 폭발
Position Encoding이 학습 기반이 아님 (정적인 sin/cos → 학습형으로 바뀌는 추세임)
Context 제한: 아주 긴 문장에서는 여전히 부족함

후속 연구

BERT : 양방향 Self-Attention + pretraining → fine-tuning 구조
GPT : Decoder-only 구조 + Language Modeling에 특화
Longformer, Reformer: 긴 시퀀스 효율적 처리
Multimodal (e.g. ViT, Flamingo, LLaVA): 이미지, 음성 등에도 확장