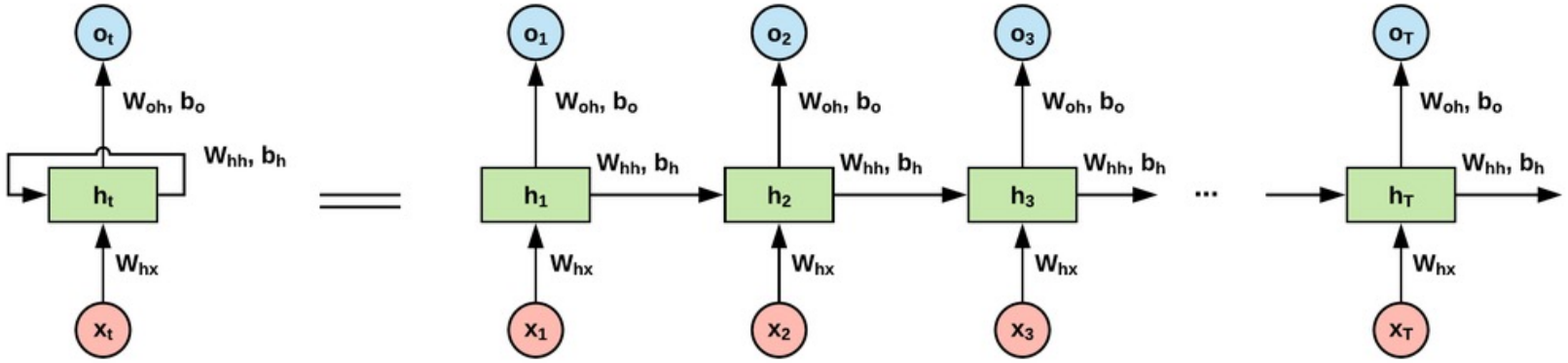
Attention Is All You Need

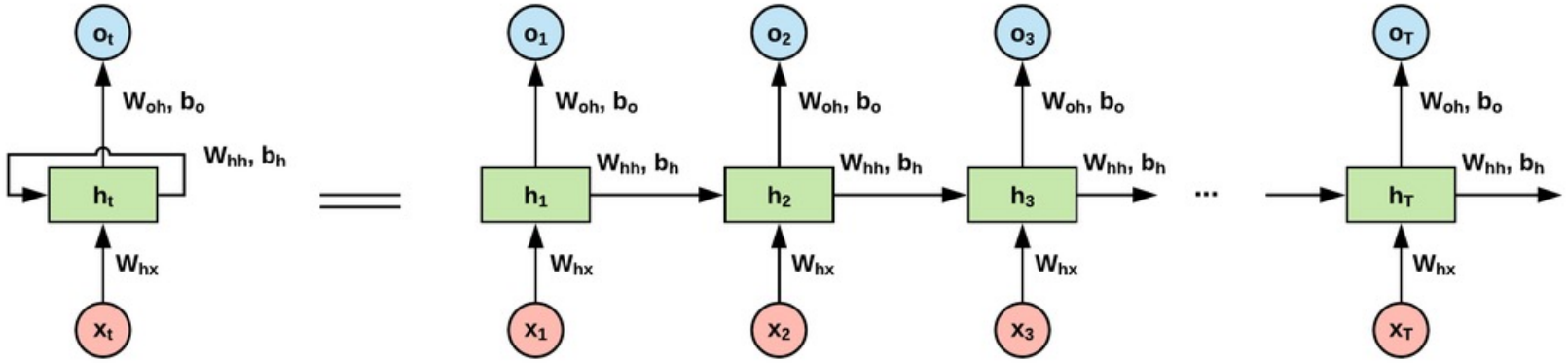
4기 물리학과 이윤영

RNN



Sequential data 처리에 용이한 neural network

RNN

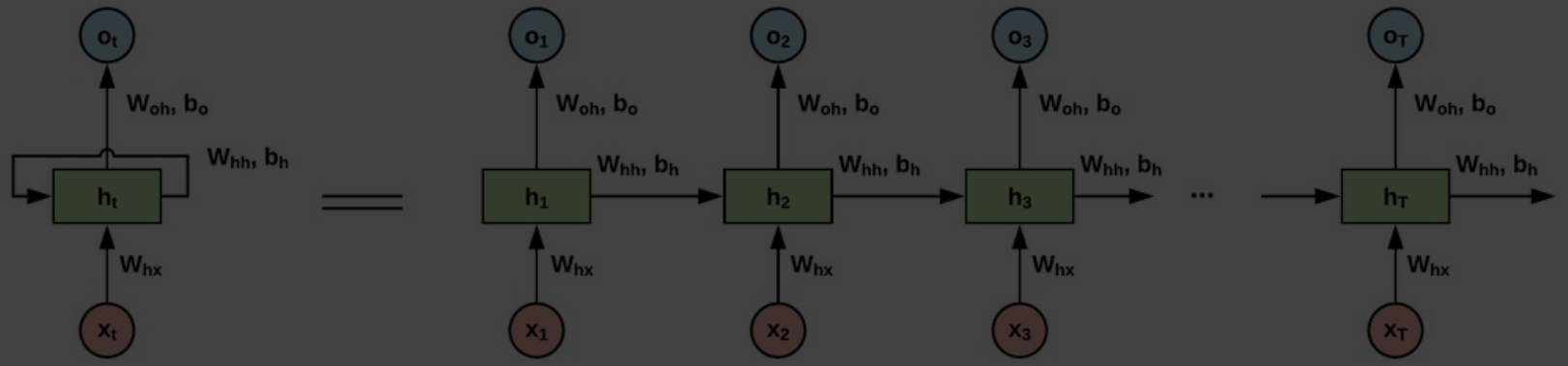


Sequential data 처리에 용이한 neural network

↓

Time step 별로 data를 처리!

RNN



Sequential data 처리에 용이한 neural network

↓

Time step 별로 data를 처리!

Data의 병렬처리 어려움 -> 긴 sequence에서 효과적이지 않음!

TRANSFORMER!

TRANSFORMER!

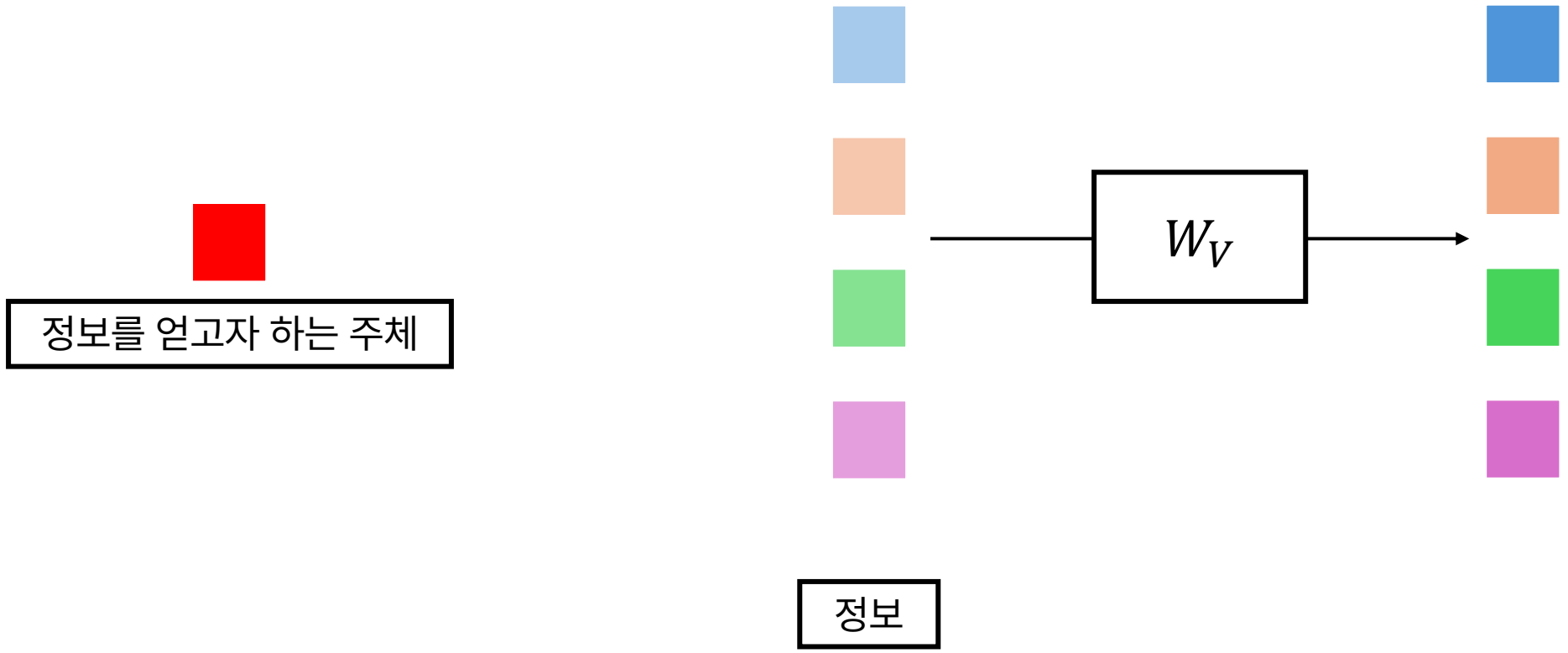
Attention Is All You Need

TRANSFORMER!

Attention Is All You Need

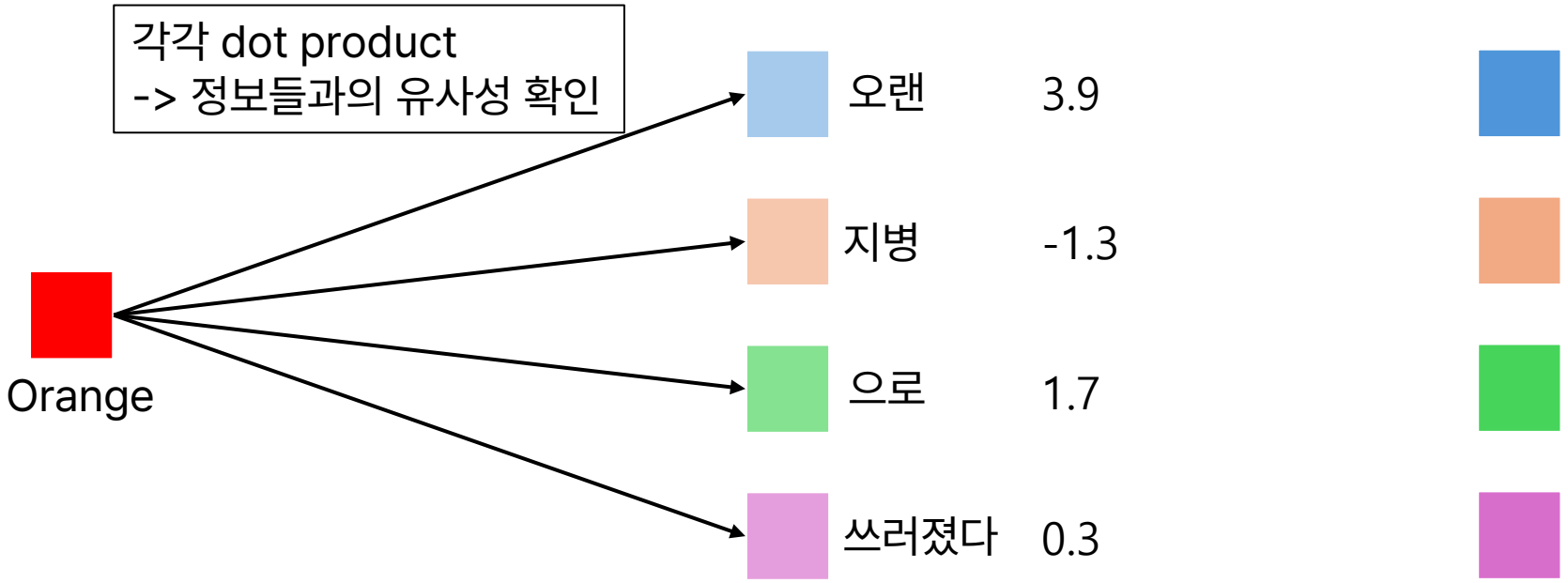
Successfully reduce RNN, CNN

What is attention?



What is attention?

What is attention?

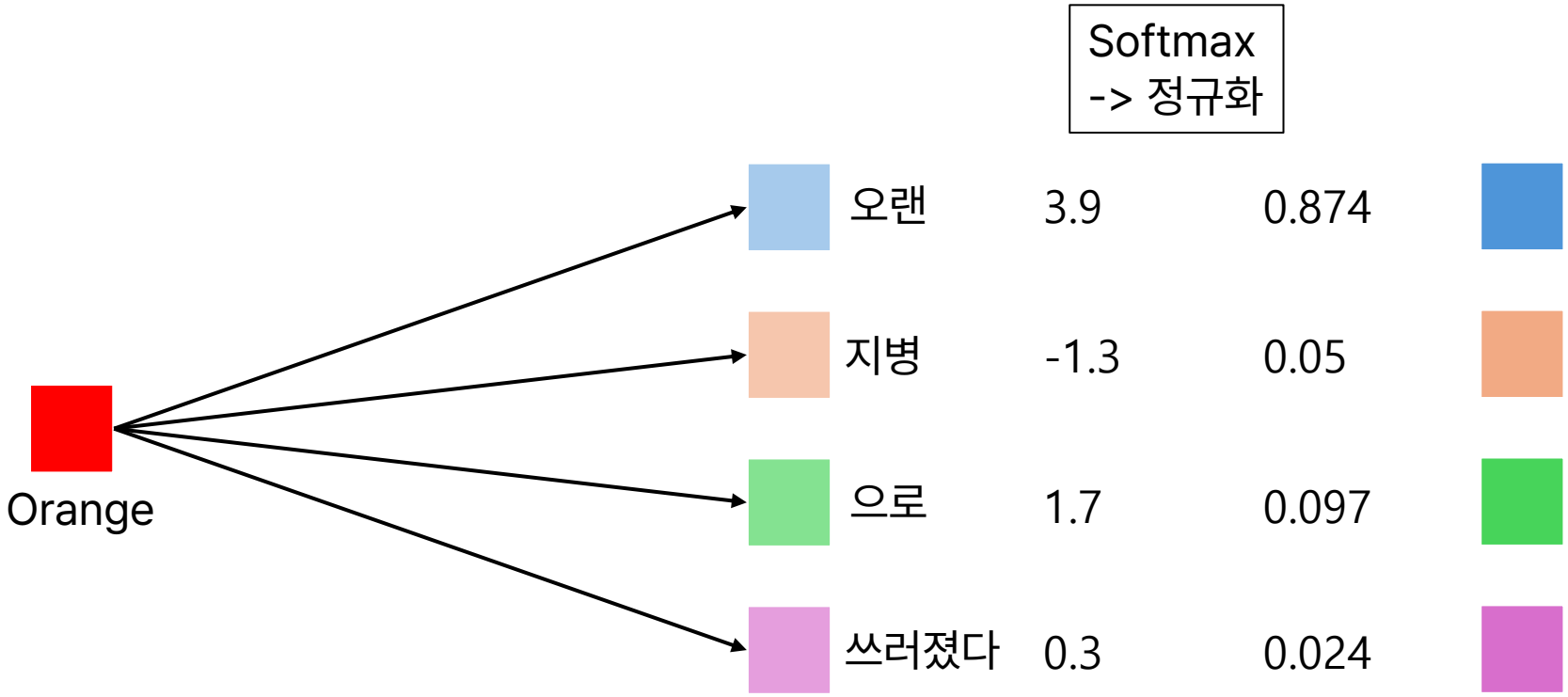


Query

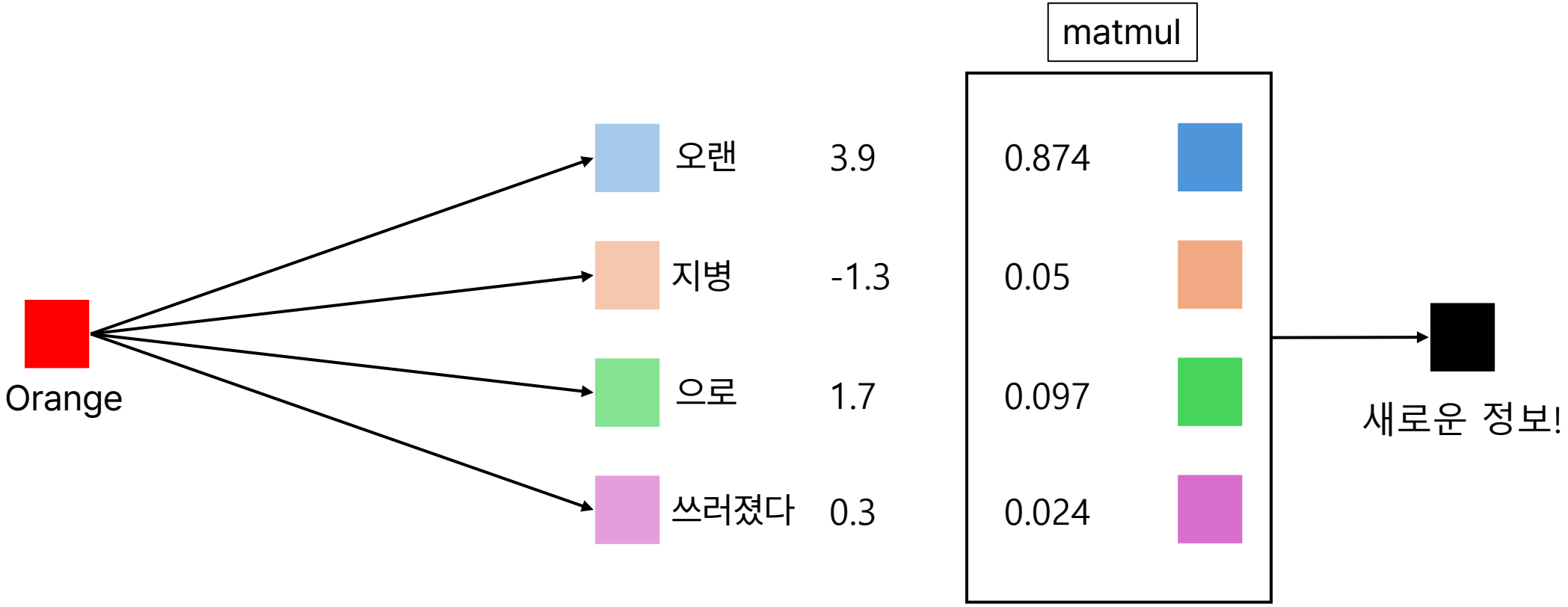
Key

Value

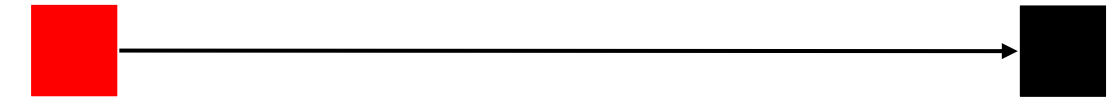
What is attention?



What is attention?



What is attention?



각 key값들의 정보가 포함된 새로운 정보

What is attention?

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

What is attention?

Query, Keys, Values

Scale dot product

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

Sqrt of keys vector dimension
-> To prevent increasing dot product

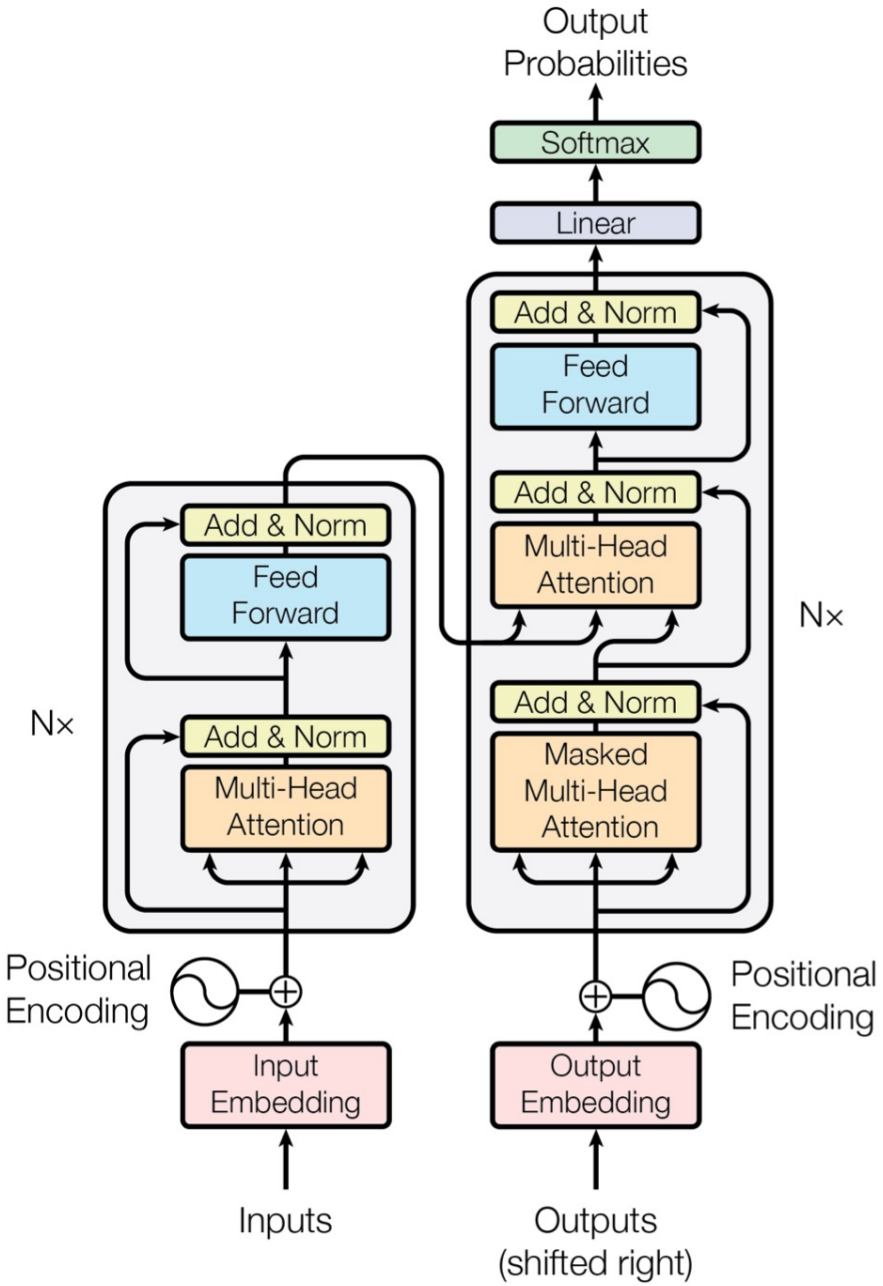


Figure 1: The Transformer - model architecture.

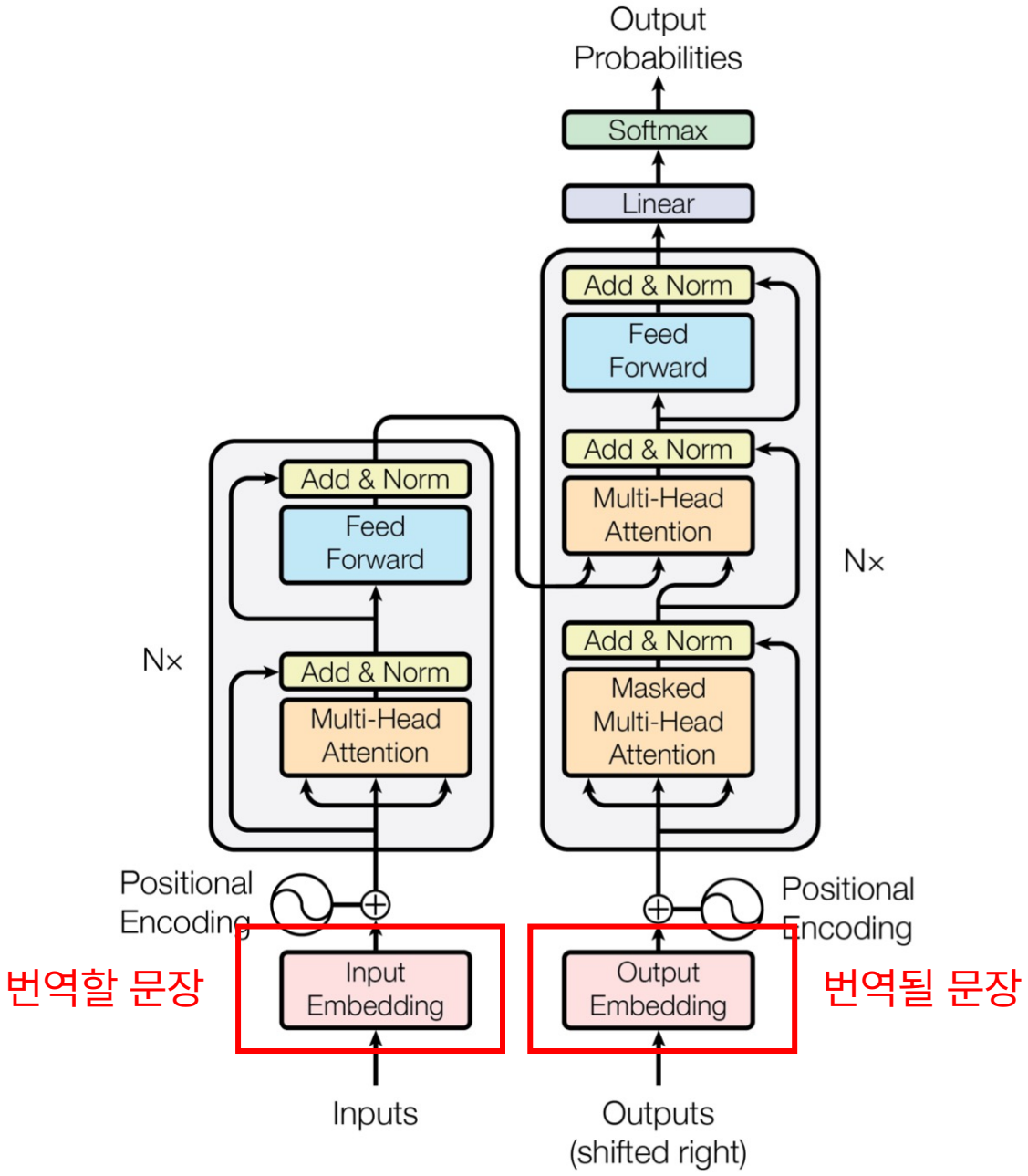


Figure 1: The Transformer - model architecture.

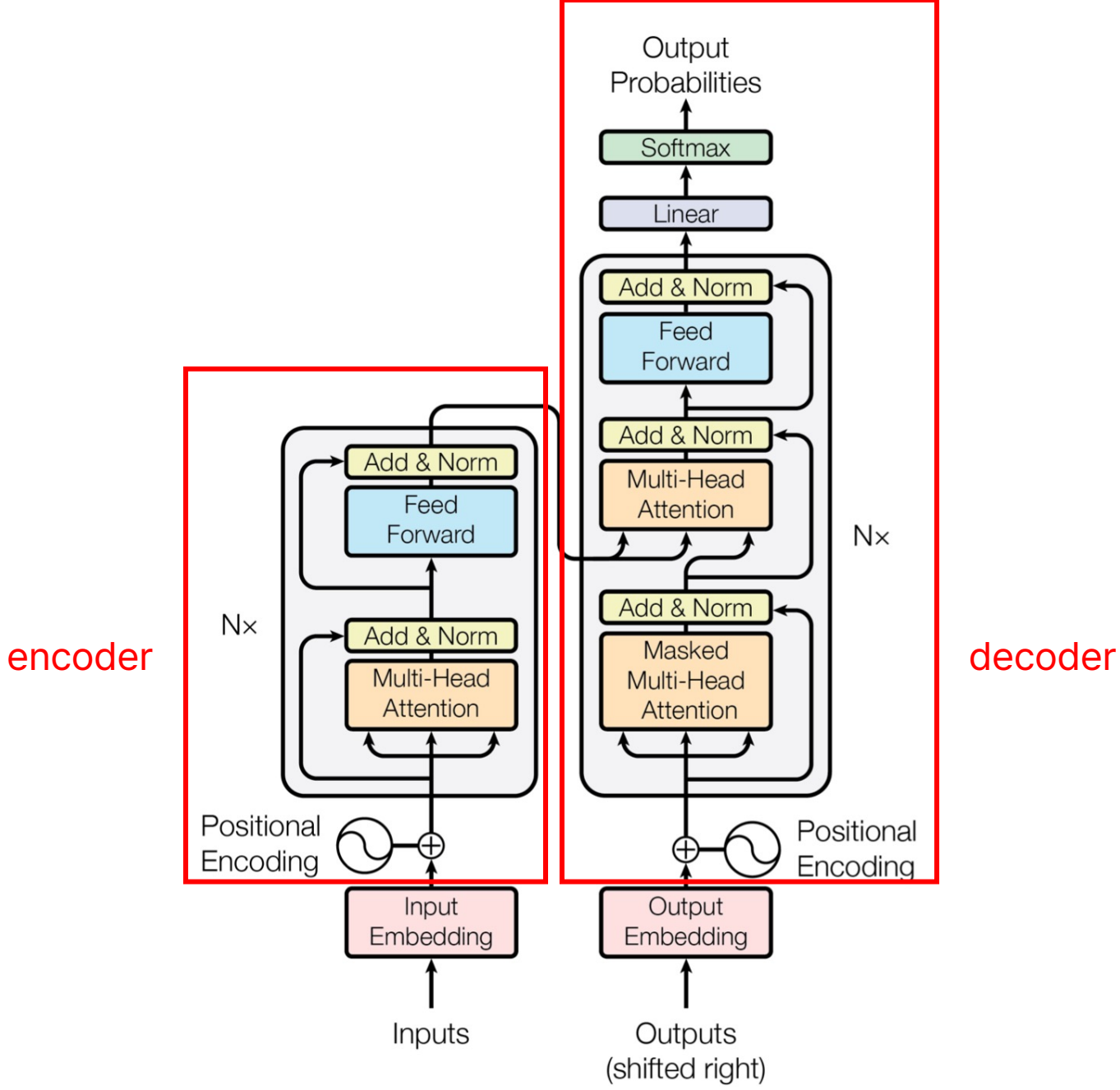


Figure 1: The Transformer - model architecture.

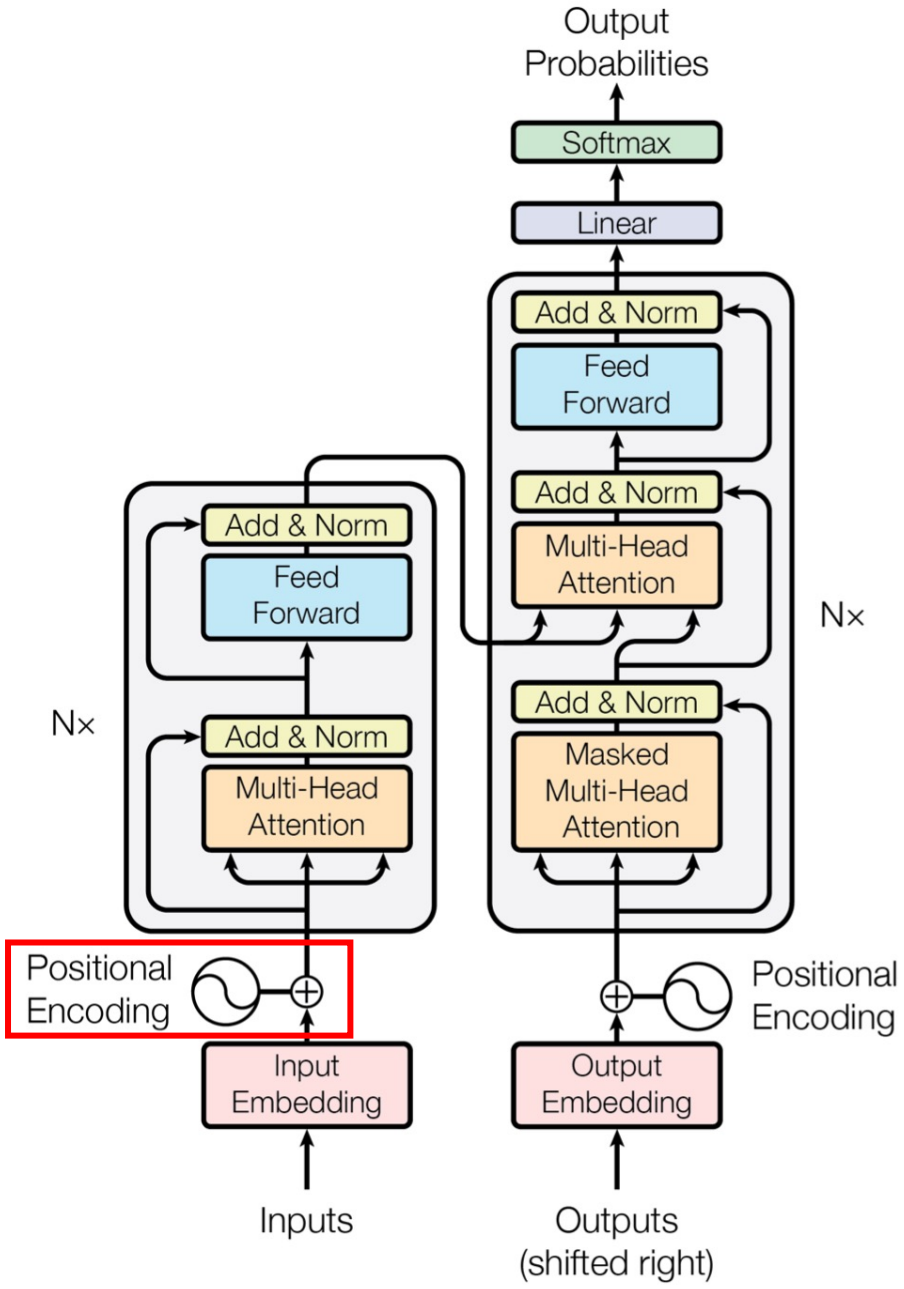
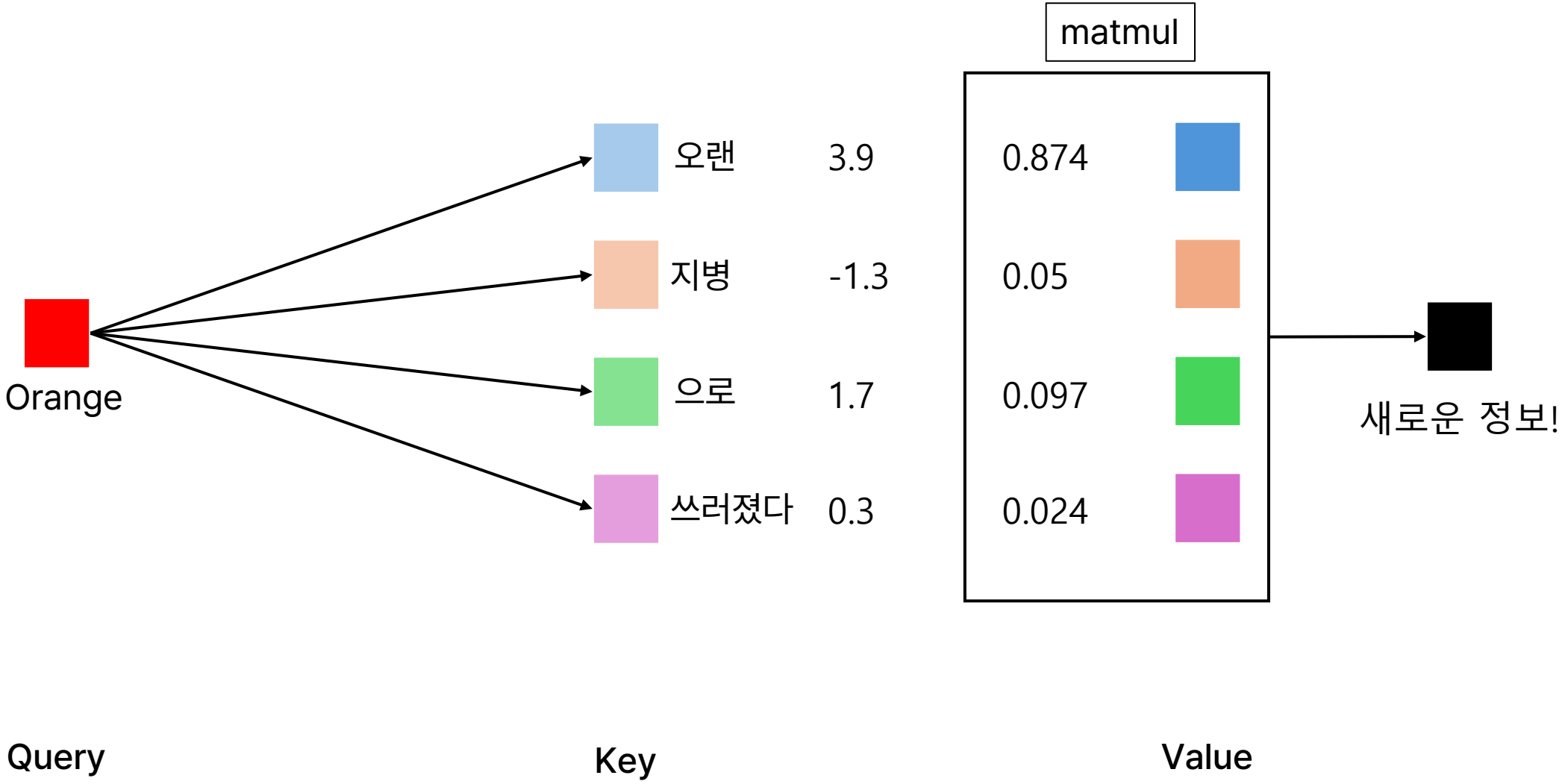
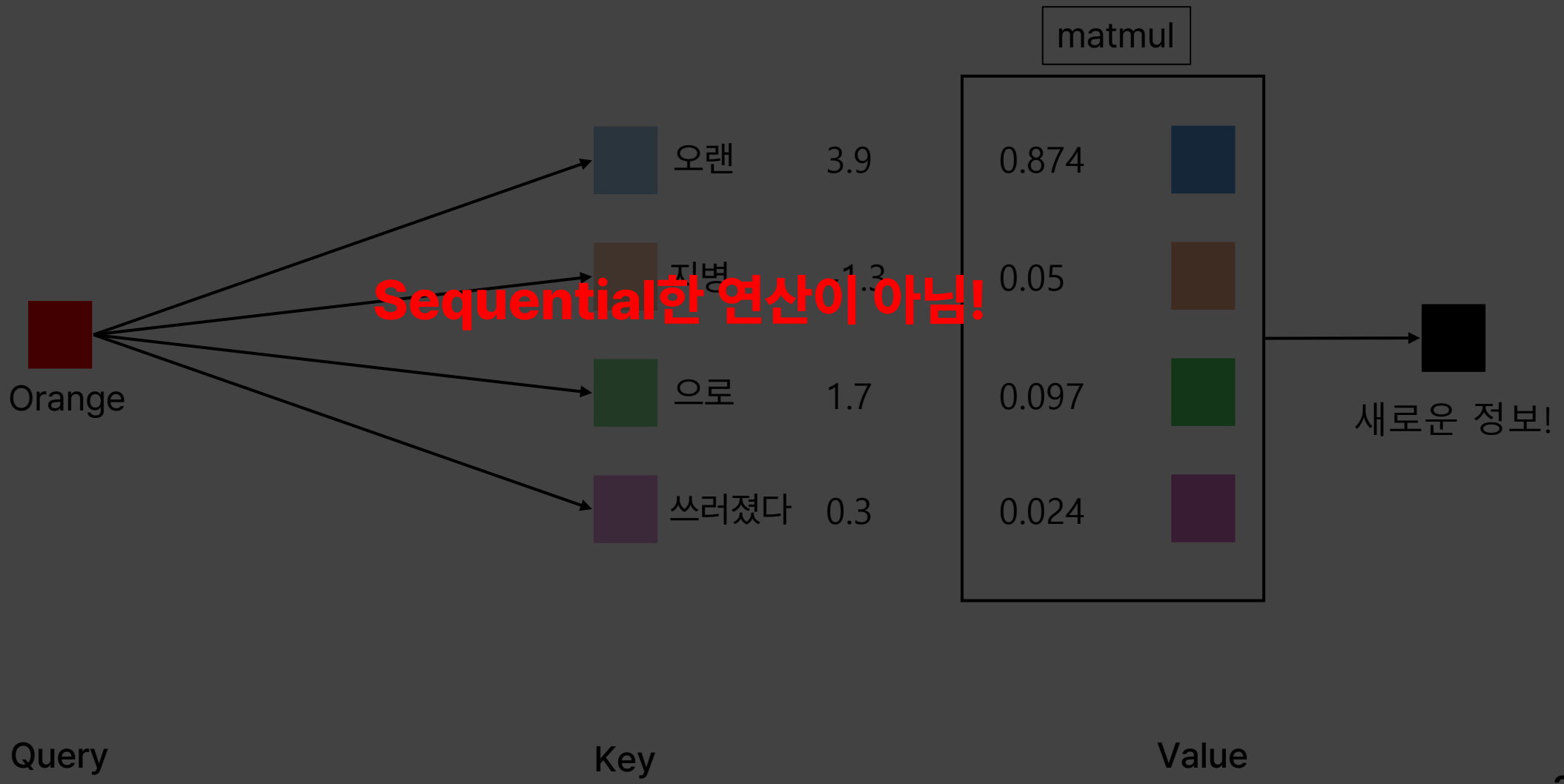


Figure 1: The Transformer - model architecture.

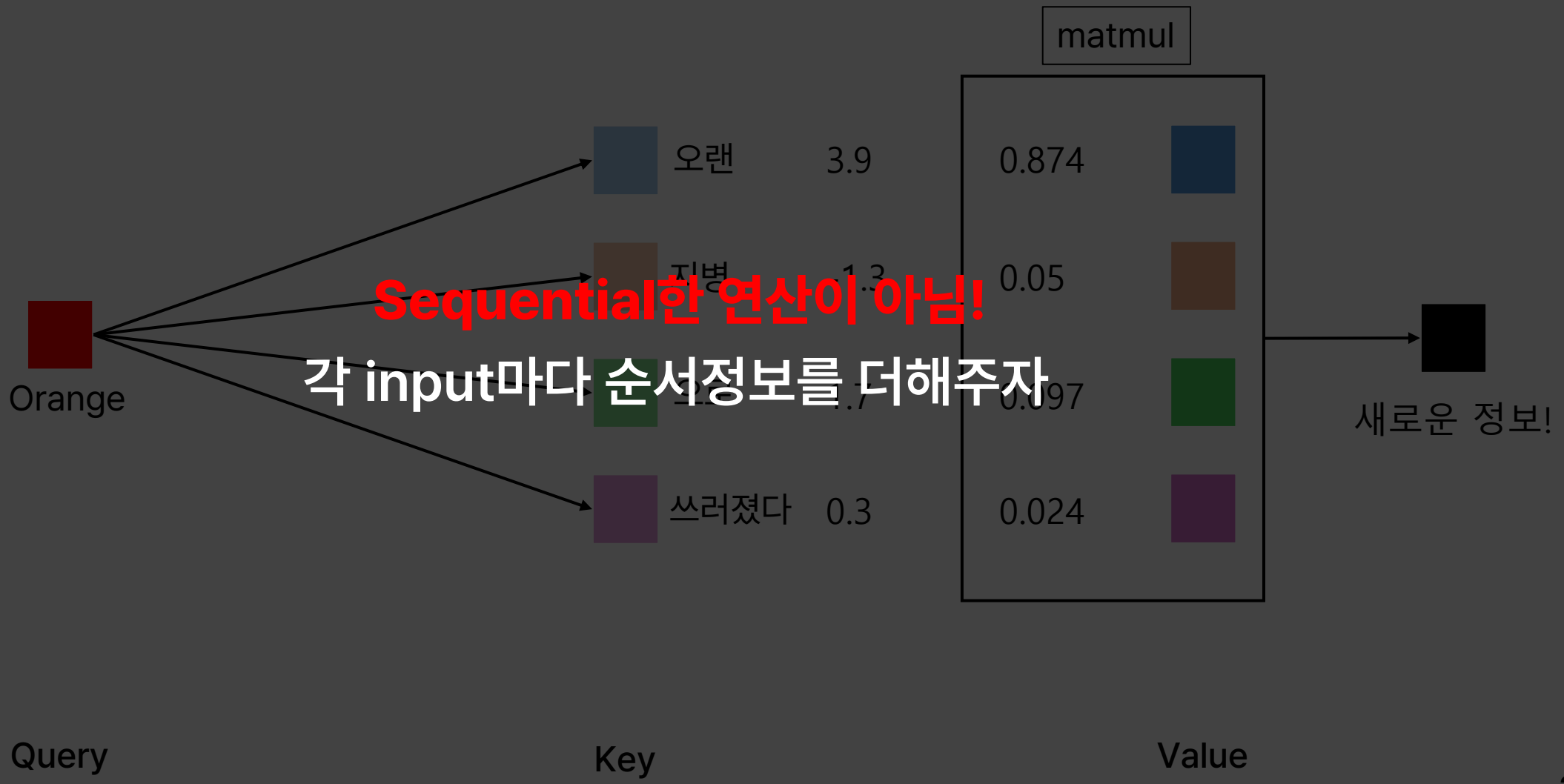
1. Positional Encoding



1. Positional Encoding



1. Positional Encoding



1. Positional Encoding

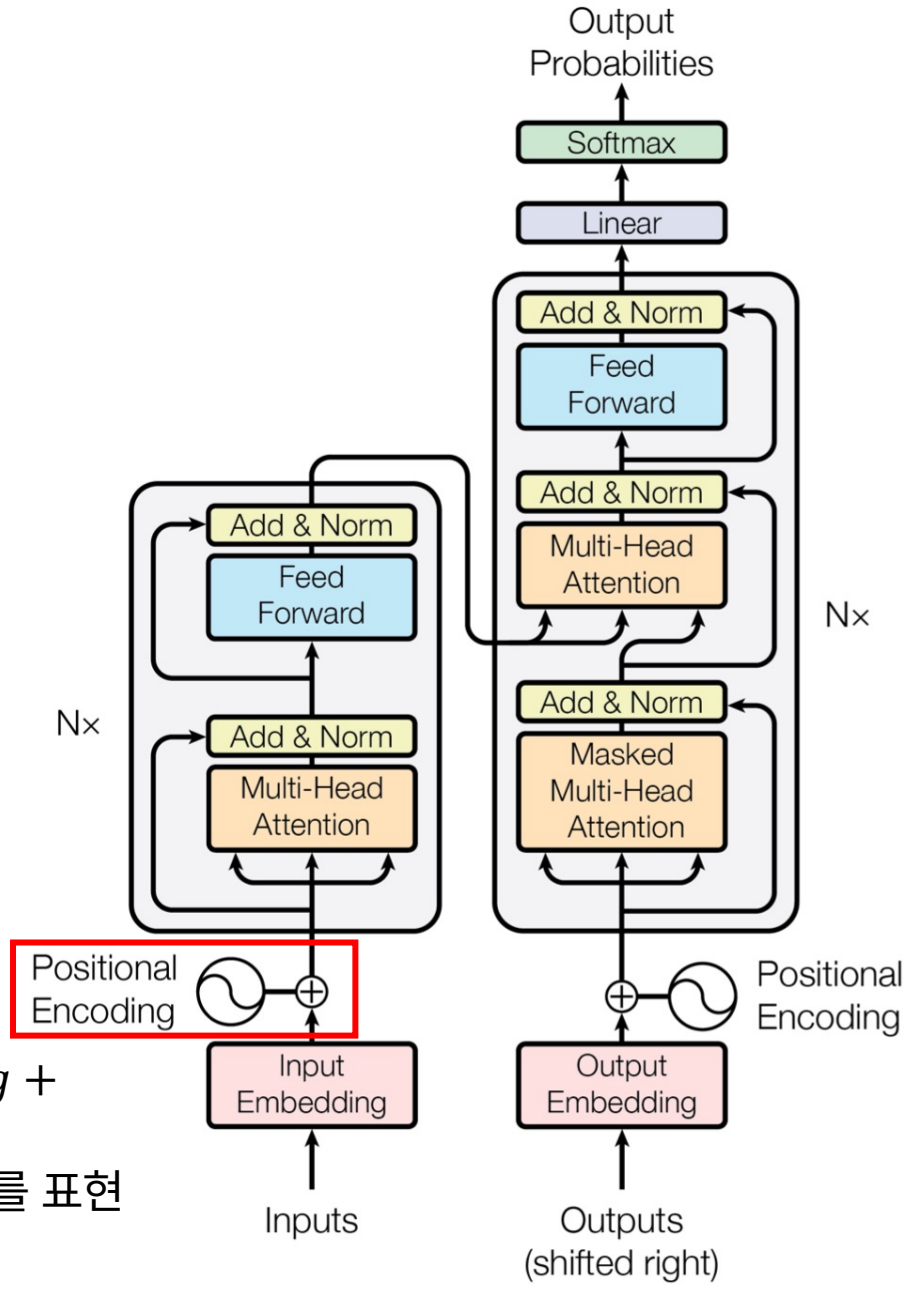
pos : input 순서
i : vector 내의 차원 index

Model dimension

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

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

1. 주기함수의 주기성
   - pos의 차이가 커질 수록 PE차이가 커짐
   - pos의 상대적인 차이가 같다면 PE차이가 같음
   -> 순서정보에 적합!
2. 10000?
   - Input sequence의 현실적인 최대길이?
   - 적절한 분해능?
     - 10000보다 작으면 : 서로 다른 위치가 유사한 값을 가질수도
     - 10000보다 크면 : 인코딩간 차이가 너무 작아져 분별력이 떨어질수도
     - 아마 실험적인 결과가 아닐까



1. $Inputs = Word\ Embadding + Positional\ Encoding$
2. PE는 sinsoid함수로 순서정보를 표현

Figure 1: The Transformer - model architecture.

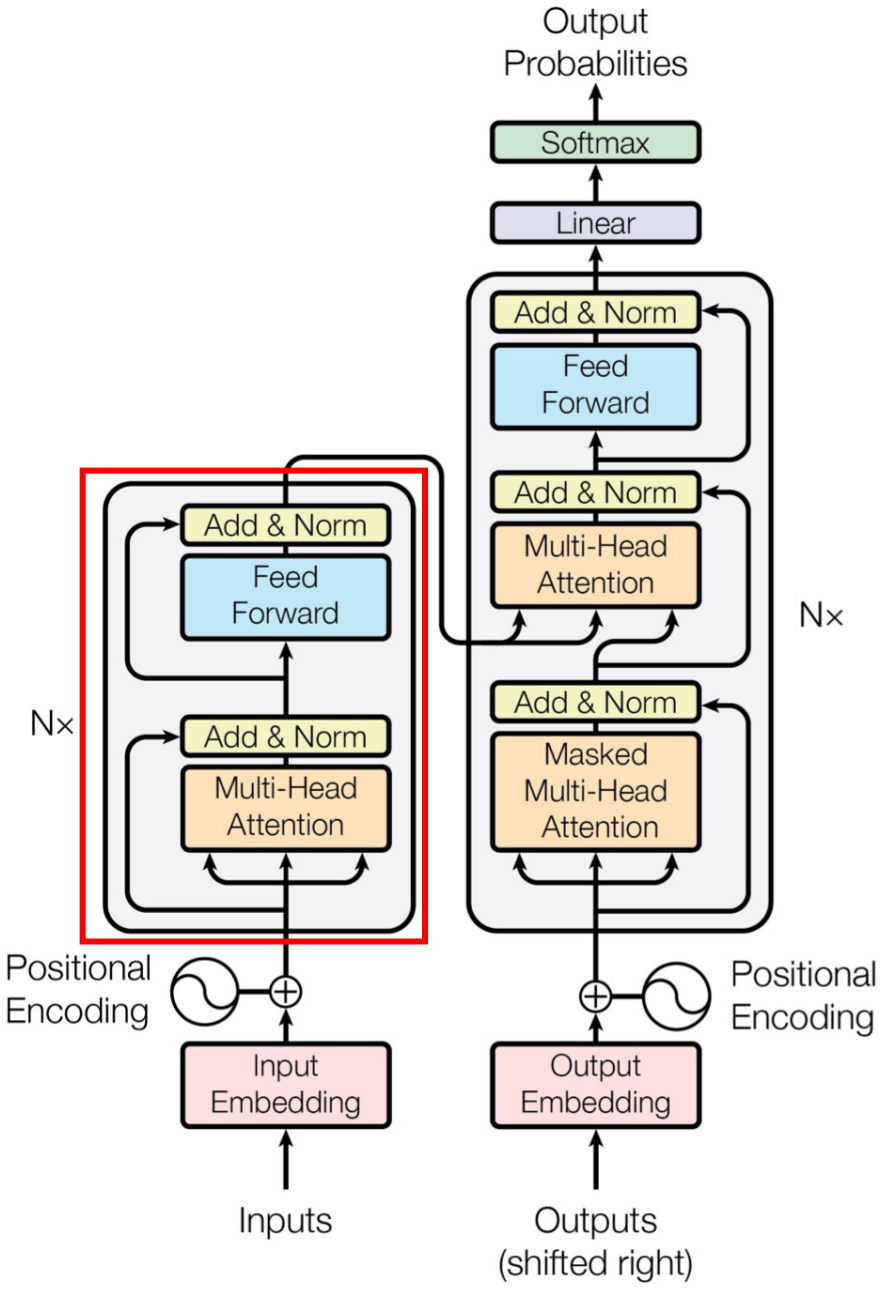


Figure 1: The Transformer - model architecture.

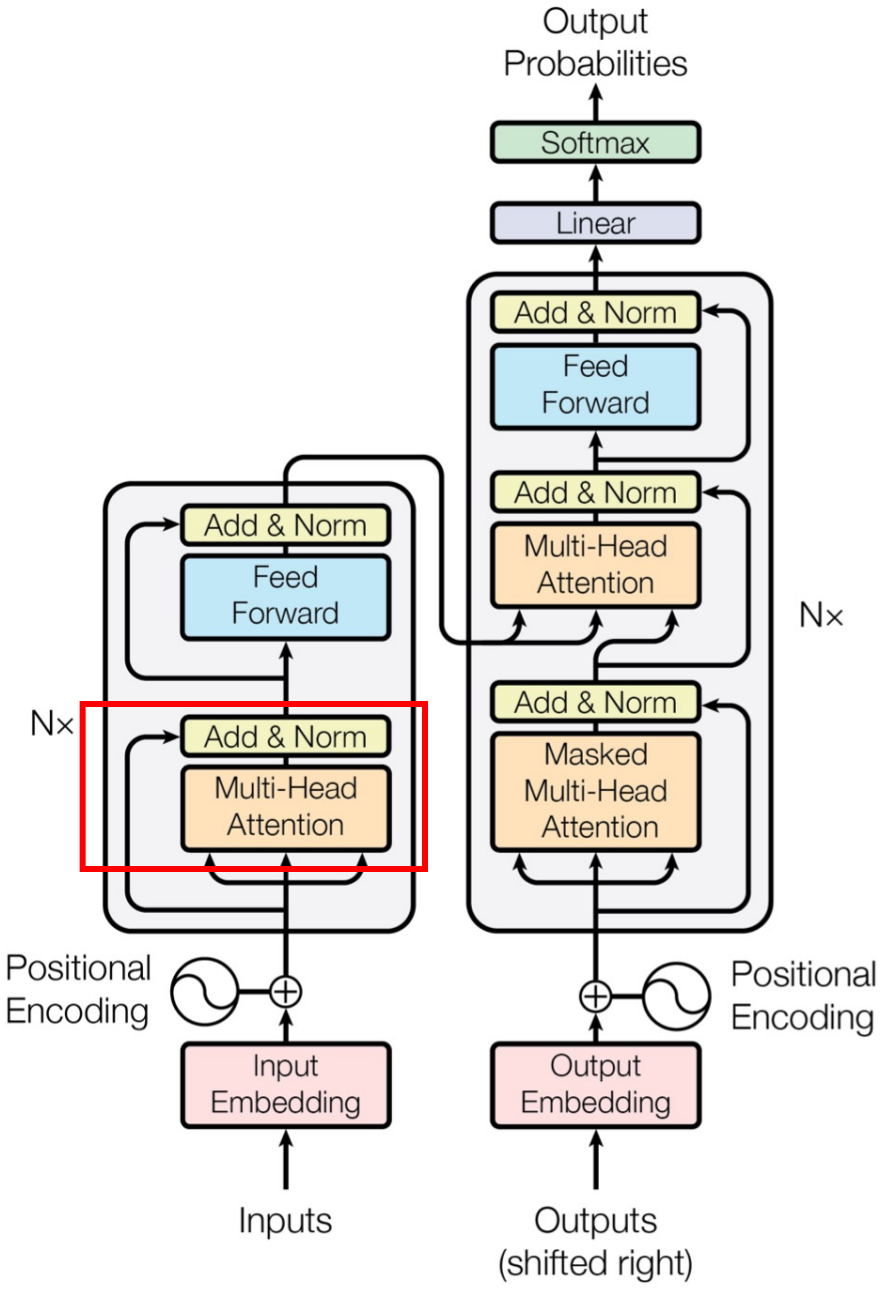


Figure 1: The Transformer - model architecture.

2. Multi-head attention

인도 음식점에 가던 중 주행 금지 표지판을 보았다

인도 주행 금지 표지판을 음식점에 가던 중 보았다

2. Multi-head attention

인도 음식점에 가던 중 주행 금지 표지판을 보았다

인도 주행 금지 표지판을 음식점에 가던 중 보았다

동음이의어
But, 둘 다 첫번째 순서인데?

2. Multi-head attention

인도 음식점에 가던 중 주행 금지 표지판을 보았다

인도 주행 금지 표지판을 음식점에 가던 중 보았다

동음이의어

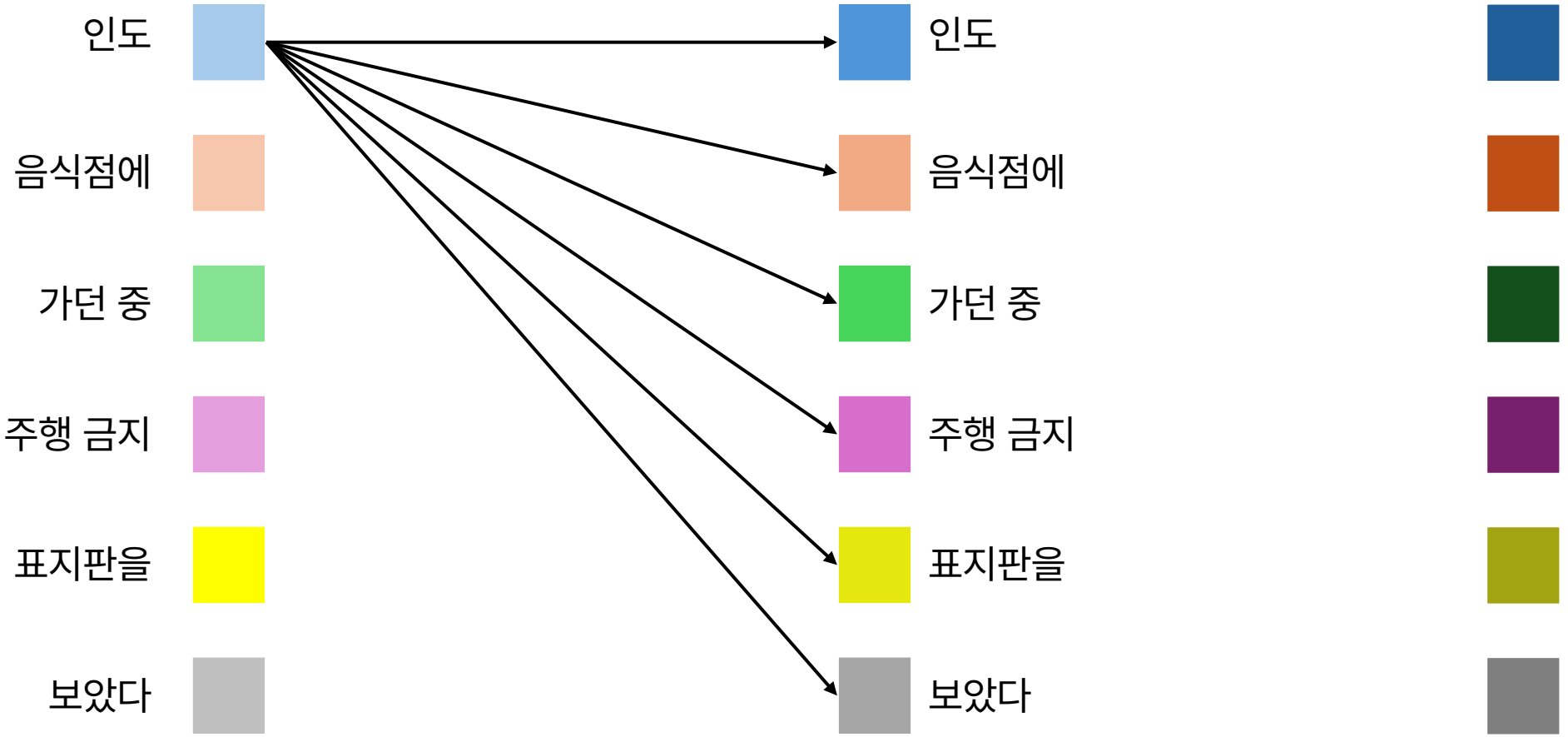
But, 둘 다 첫번째 순서인데?

주변 단어들을 통해 맥락을 확인하자

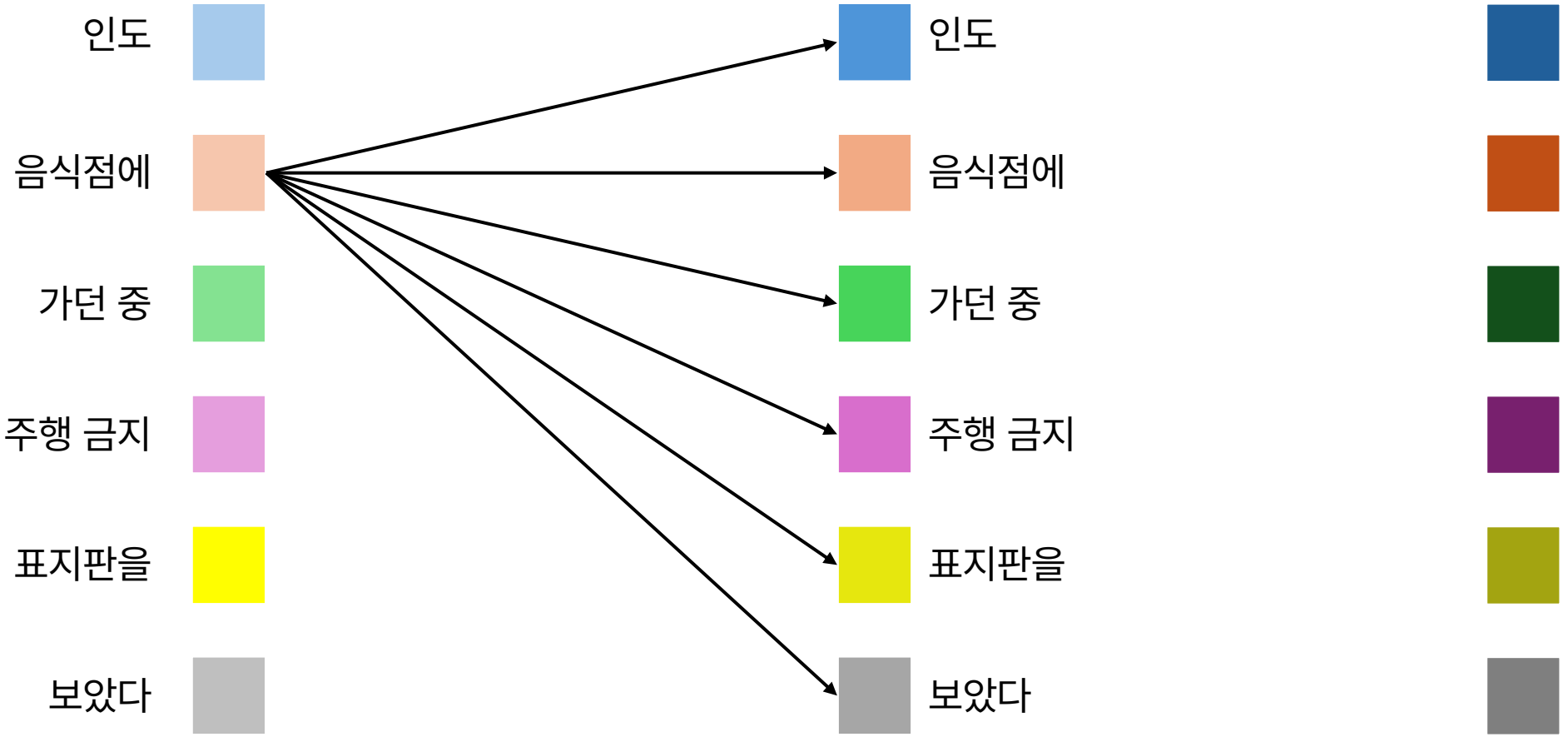
2. Multi-head attention

Self Attention

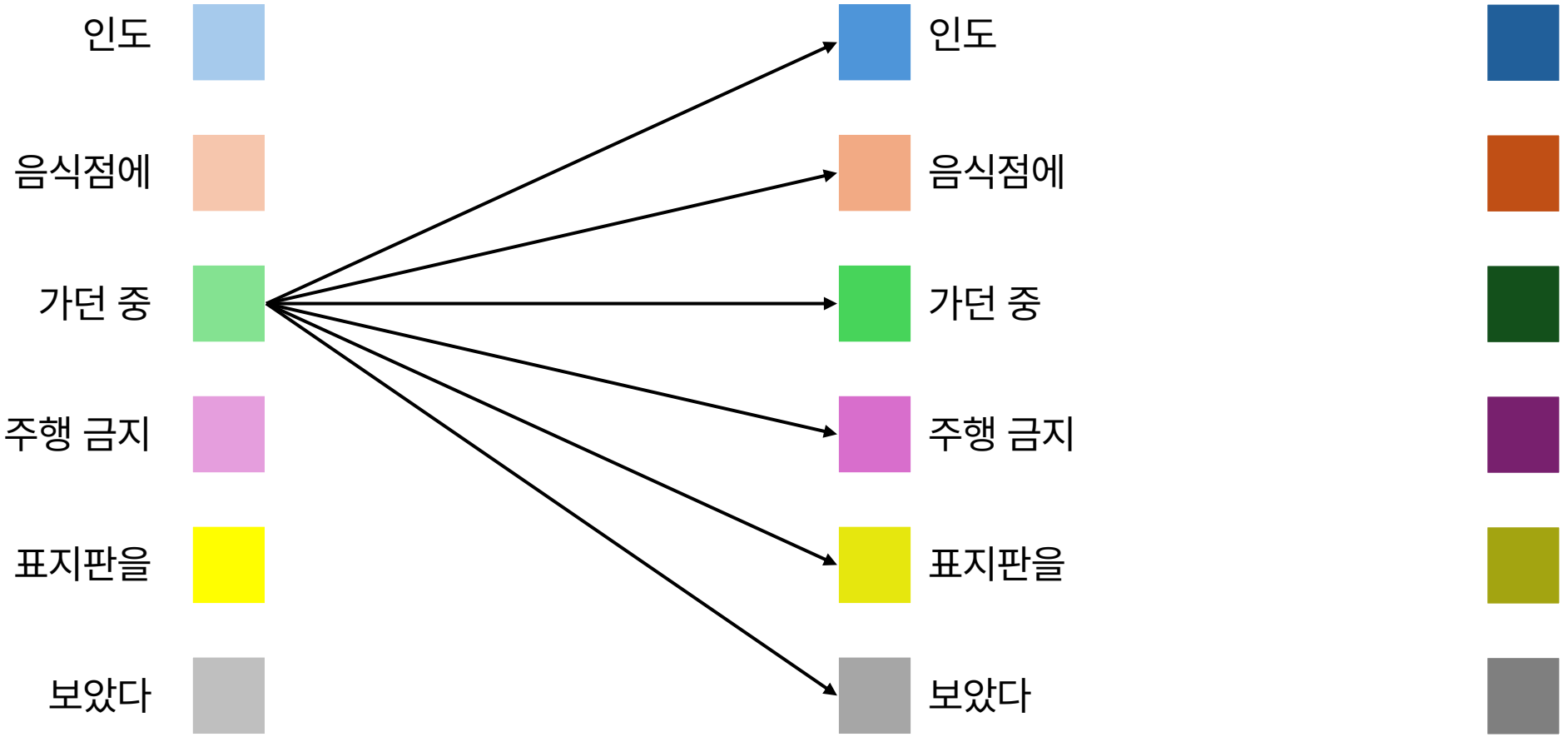
2. Multi-head attention – self attention



2. Multi-head attention – self attention



2. Multi-head attention – self attention



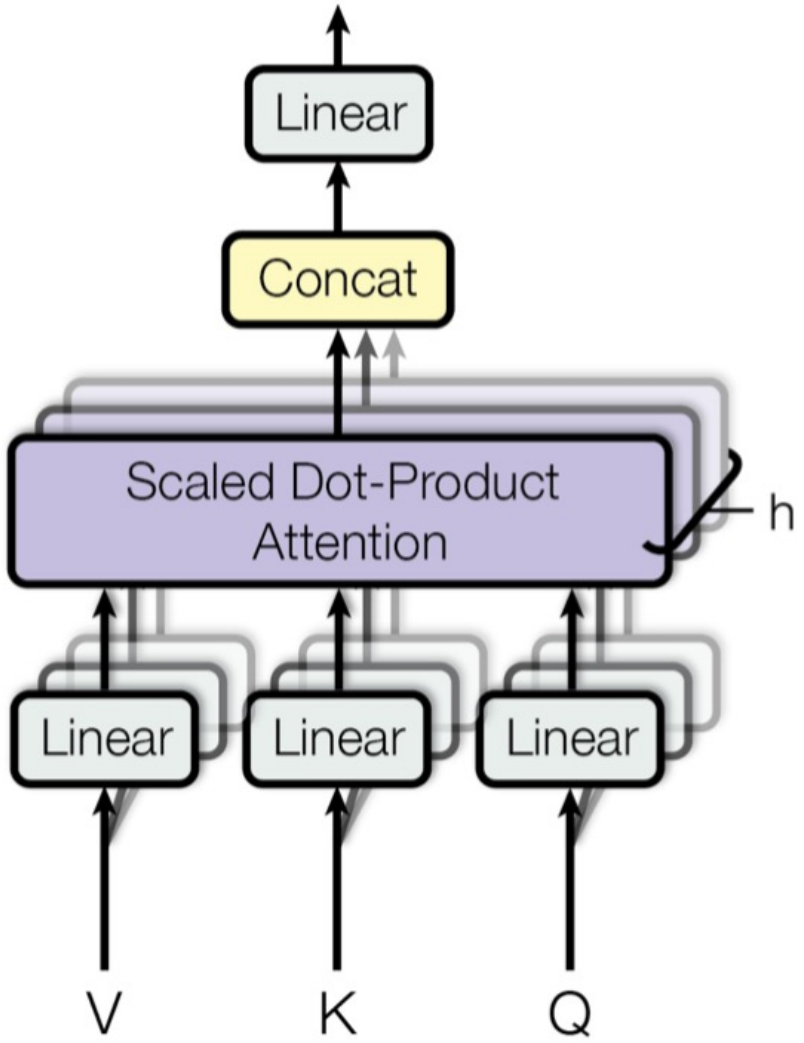
2. Multi-head attention

Self Attention : 자기 자신과 attention 연산을 하는 것

2. Multi-head attention

Multi-head Attention : self attention을 한 번에 병렬처리 하여 연산하는 것

2. Multi-head attention



h=8

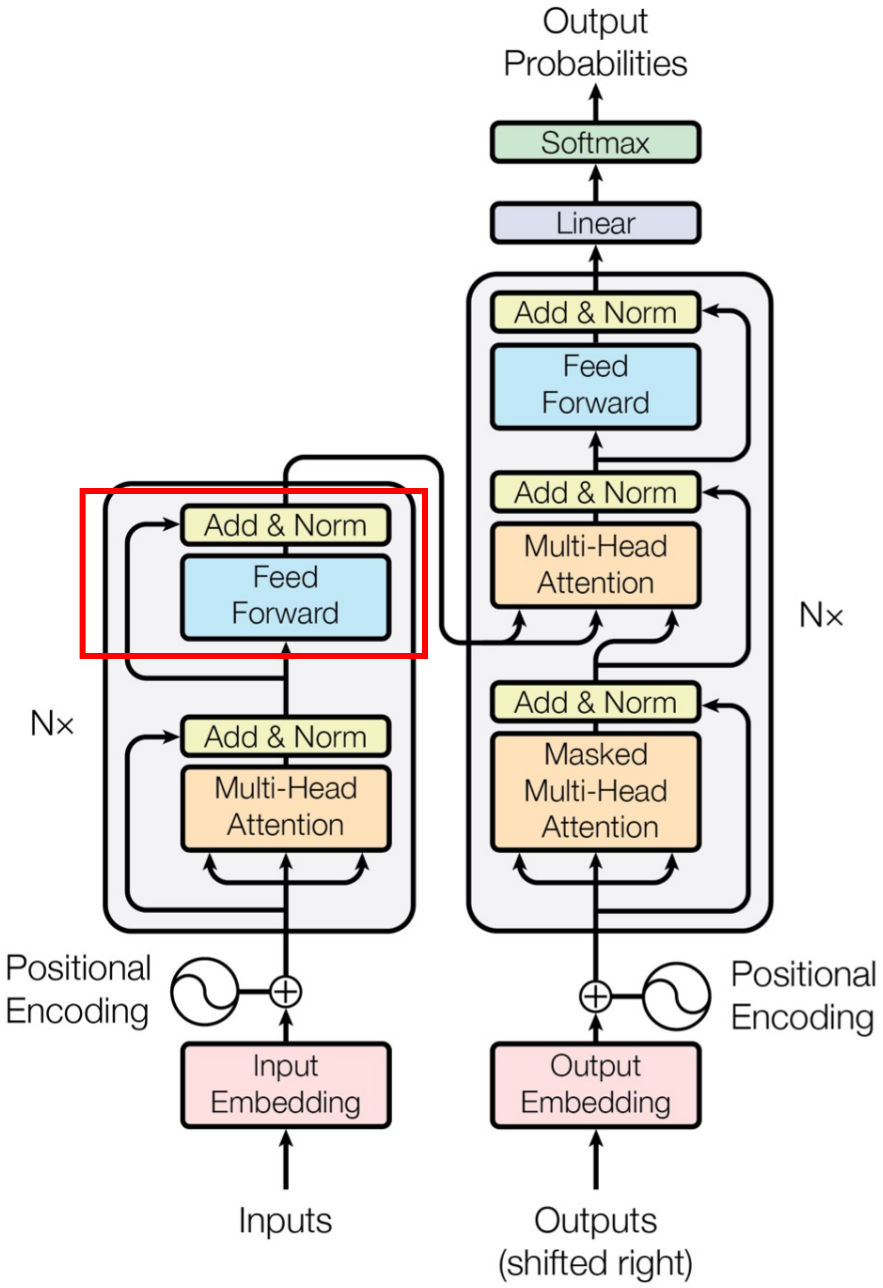


Figure 1: The Transformer - model architecture.

3. Feed Forward

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

2. ReLU

1. fully connected layer

3. fully connected layer

-> 비선형성 부여!

N=6

Encoder의 input, output의 차원은 같다

input 문장으로 맥락이 포함된 단어 정보를 만듦

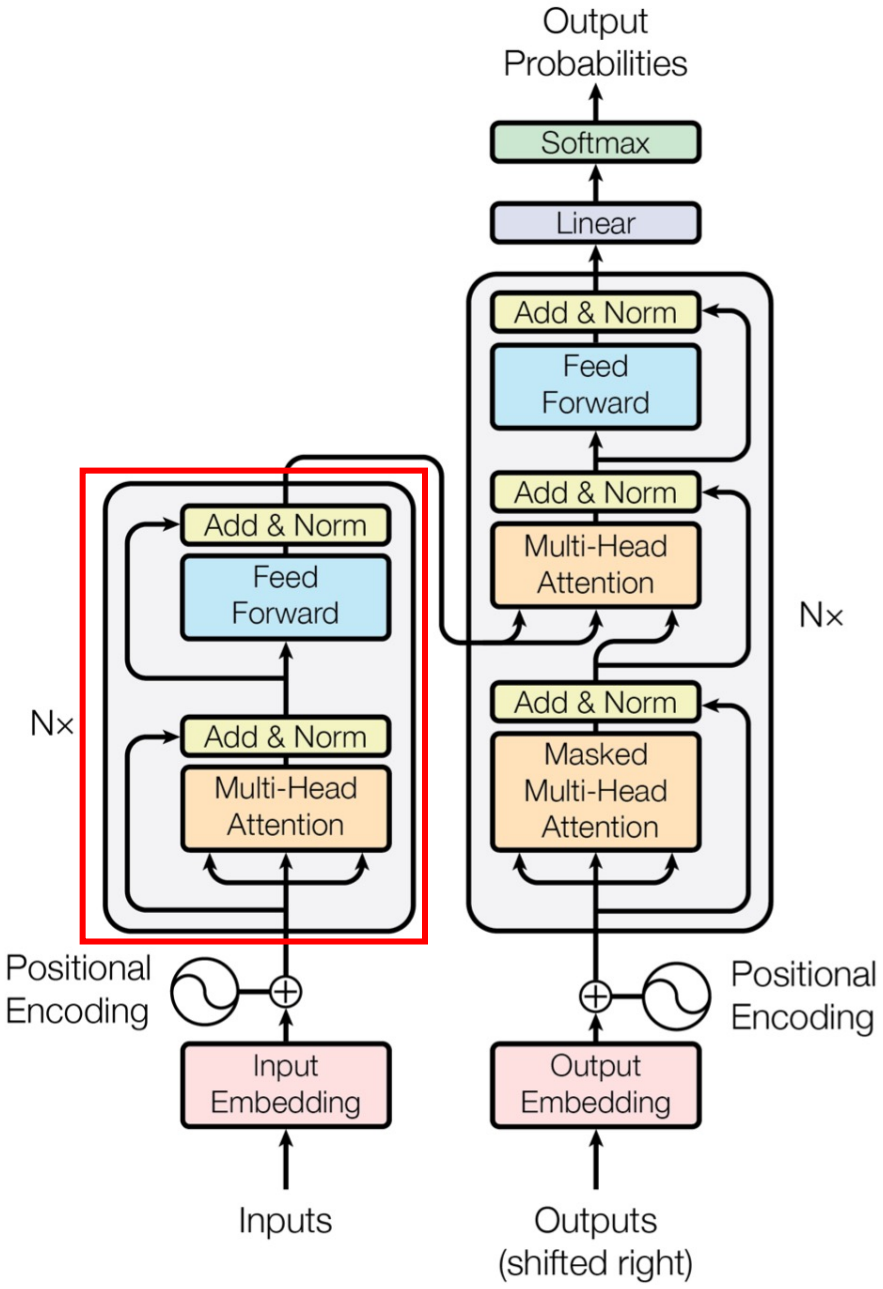


Figure 1: The Transformer - model architecture.

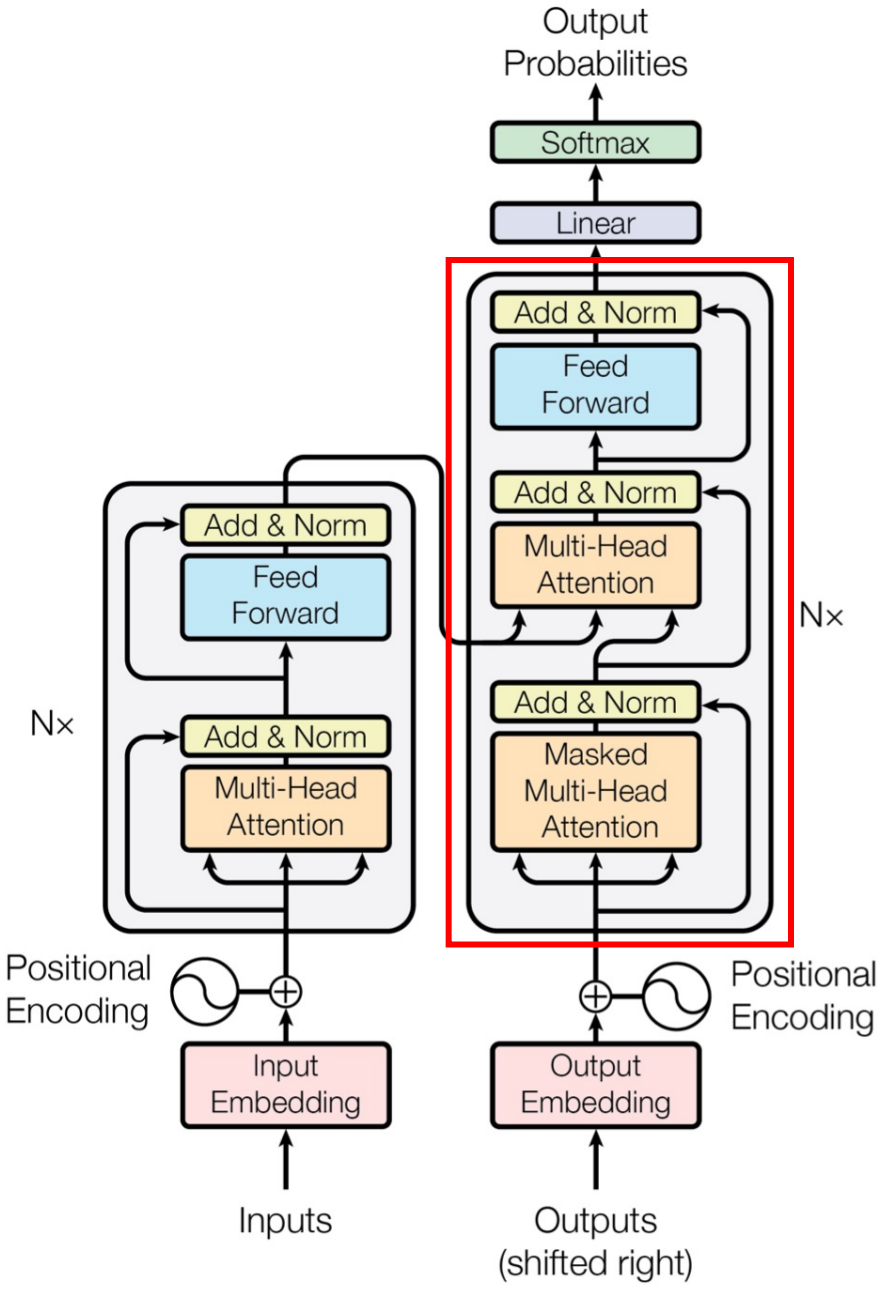


Figure 1: The Transformer - model architecture.

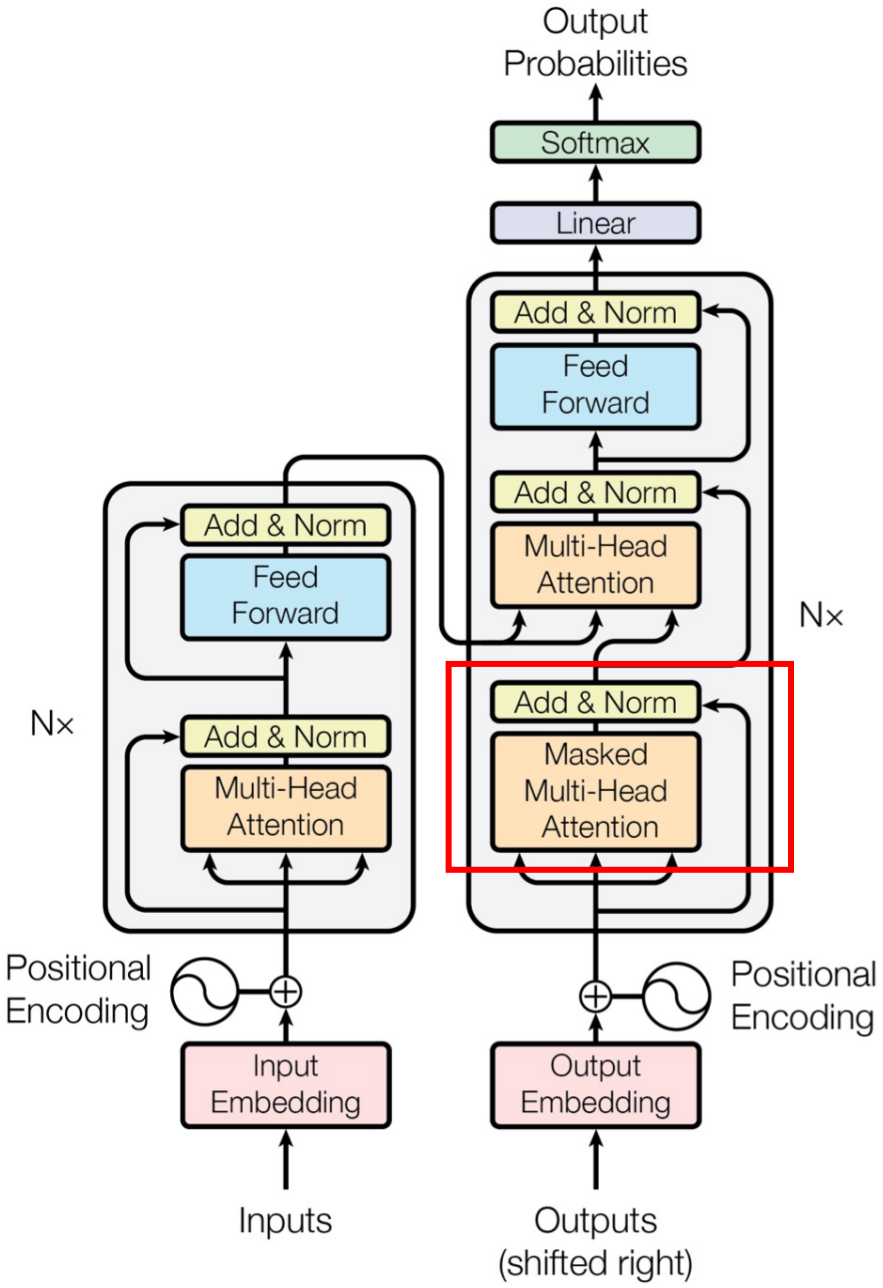


Figure 1: The Transformer - model architecture.

3. Masked Multi-head attention

아 자퇴하고 싶다

3. Masked Multi-head attention

아 (나는) 자퇴하고 싶다 ⟶ Oh, I want to quit my school

3. Masked Multi-head attention

아 (나는) 자퇴하고 싶다 ⟶ Oh, I want to quit my school

3. Masked Multi-head attention

아 (나는) 자퇴하고 싶다 ⟶ Oh, I want to quit my school

3. Masked Multi-head attention

아 (나는) 자퇴하고 싶다 ⟶ Oh, I want to quit my school

3. Masked Multi-head attention

아 (나는) 자퇴하고 싶다 ⟶ Oh, I want to **quit** my school

3. Masked Multi-head attention

아 (나는) 자퇴하고 싶다 ⟶ Oh, I want to quit my school

3. Masked Multi-head attention

아 (나는) 자퇴하고 싶다 → Oh, I want to quit my school

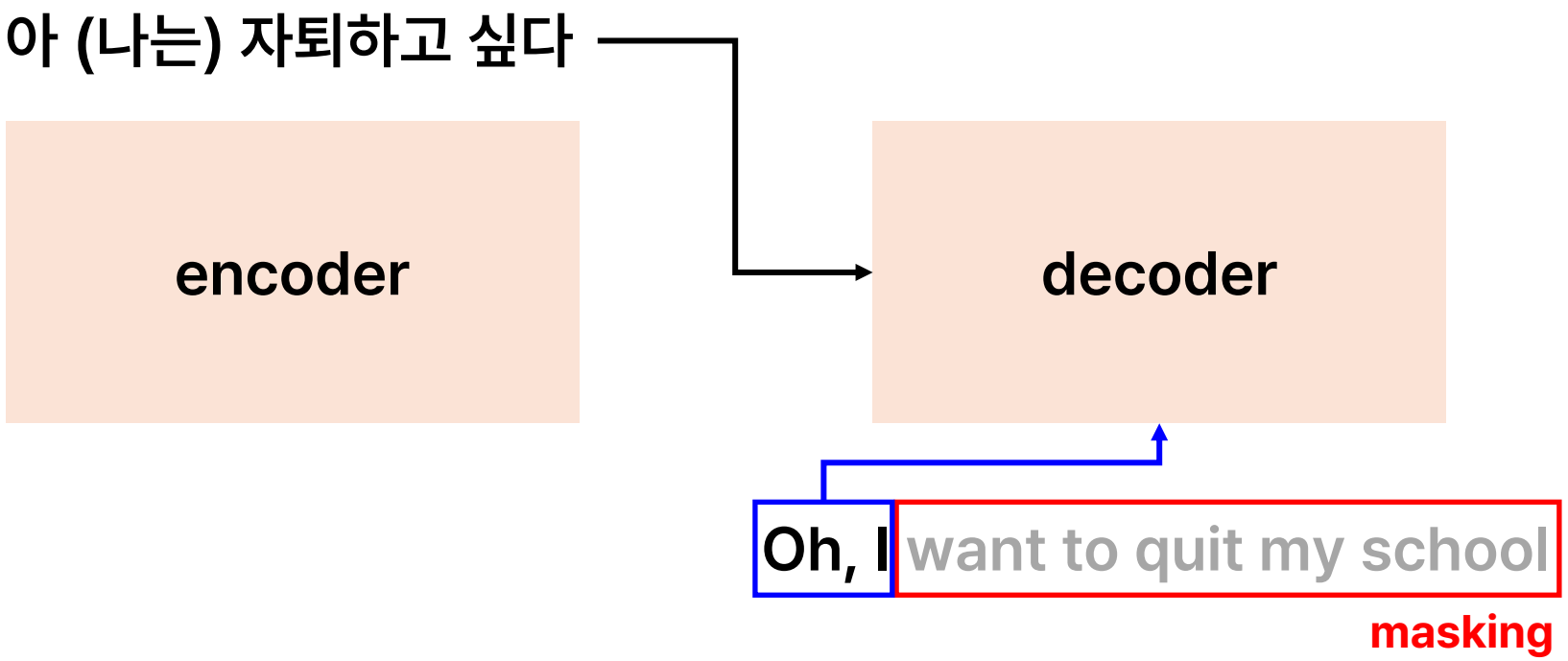
want to를 출력하고 싶은데
school을 이미 학습한 상태라면?
-> 학습에 혼동

3. Masked Multi-head attention

아 (나는) 자퇴하고 싶다 ⟶ Oh, I want to quit my school

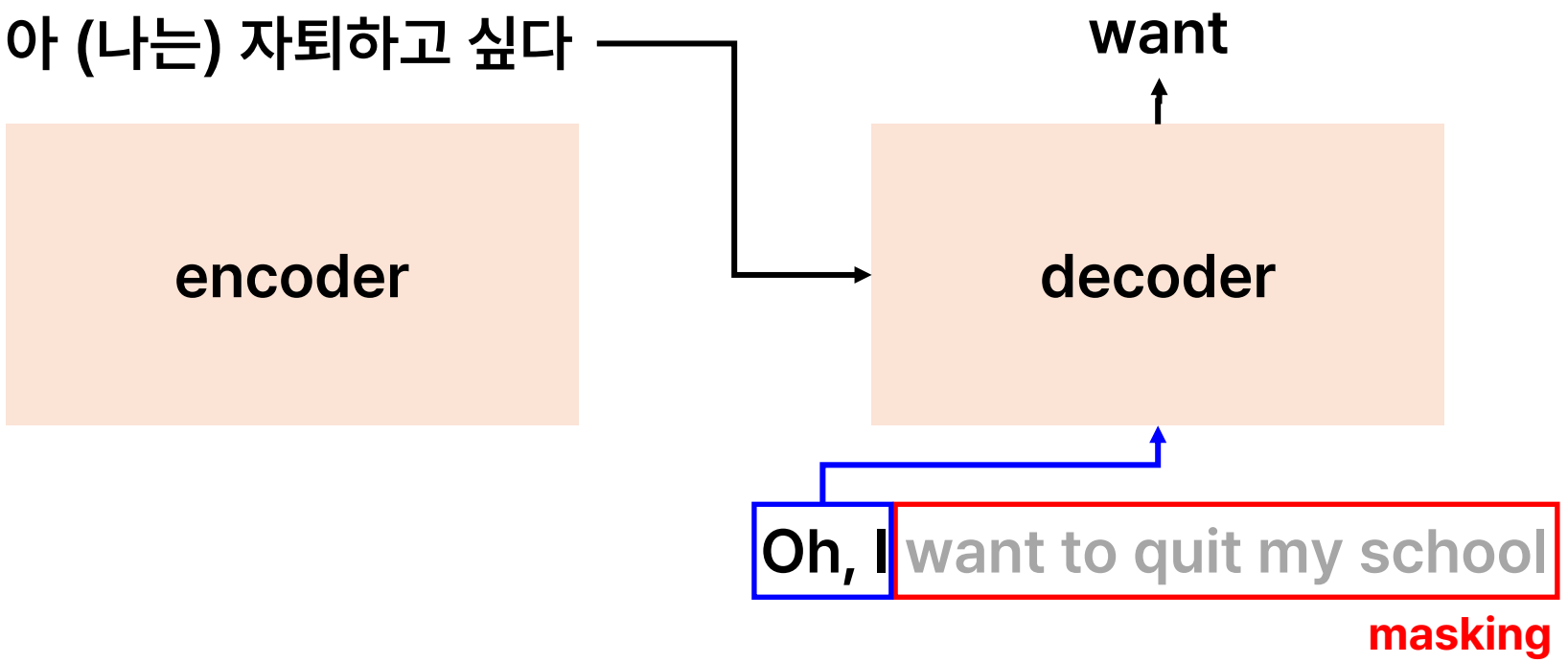
want to를 출력하고 싶은데
school을 이미 학습한 상태라면?
-> 학습에 혼동
-> 그럼 뒤에 단어는 지우고 학습해~

3. Masked Multi-head attention

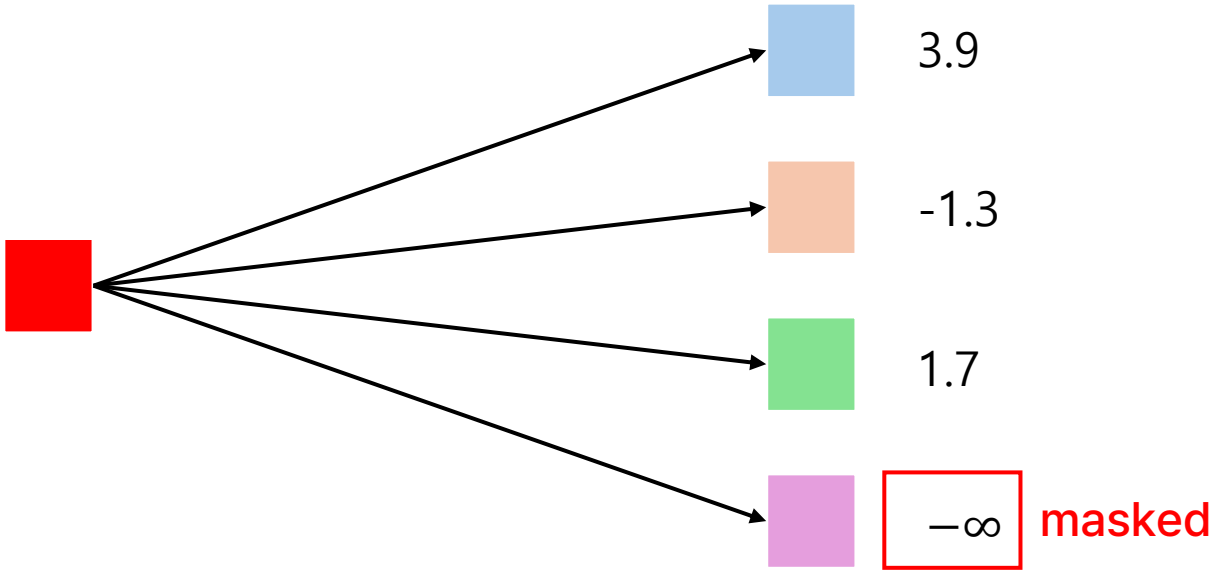


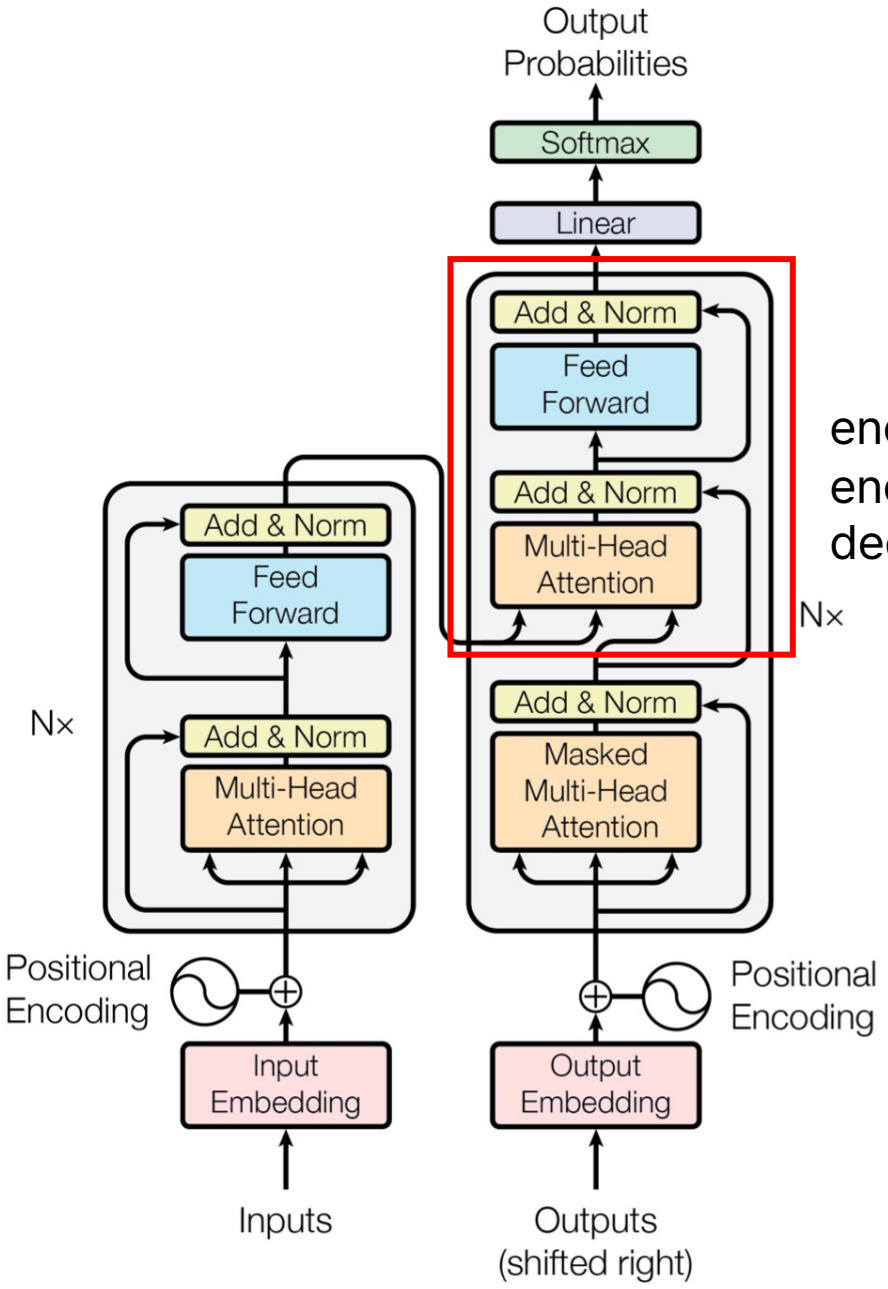
3. Masked Multi-head attention

-> 3번째 단어를 예측하기 위해 2번째 단어까지만 사용
나머지 단어들 masking



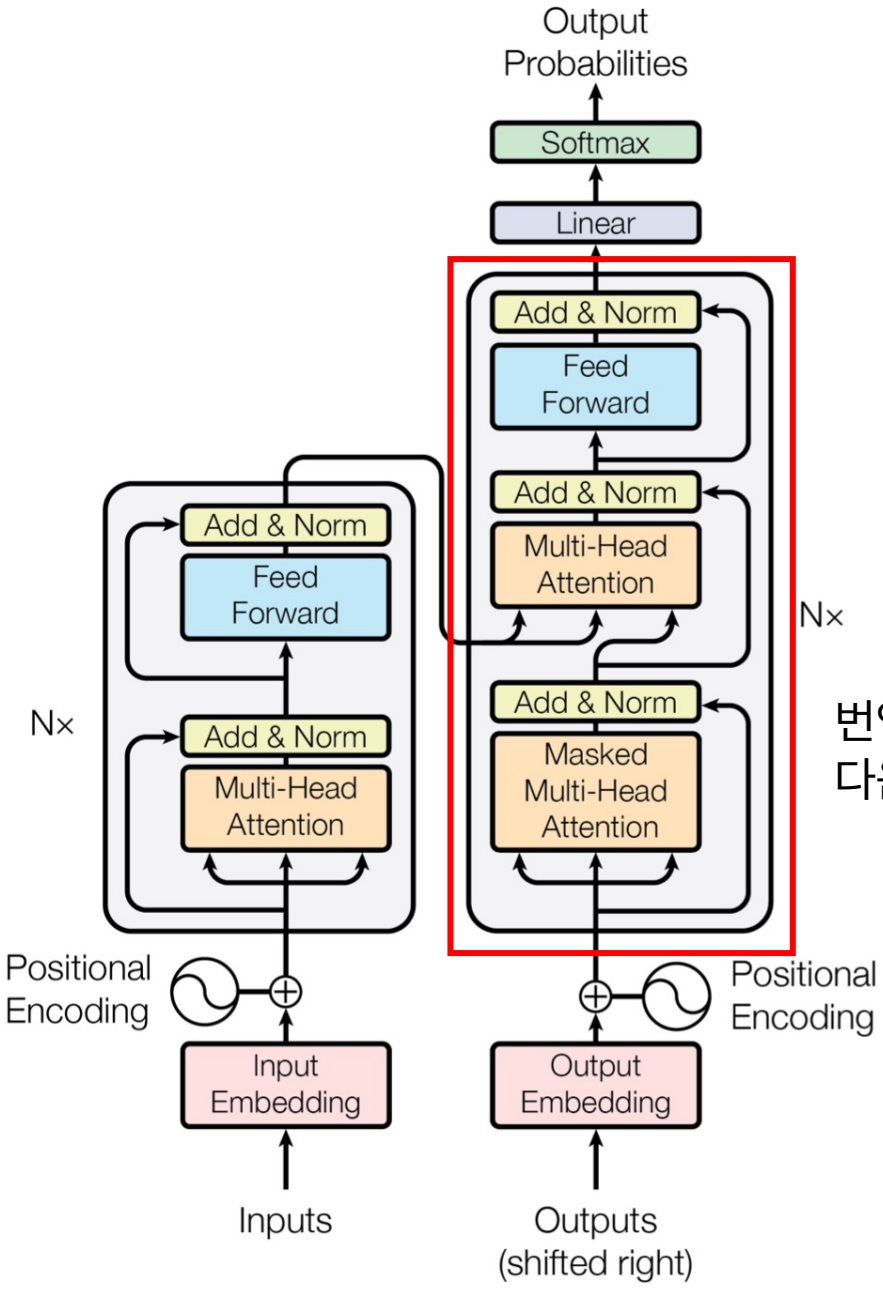
3. Masked Multi-head attention





encoder 구조와 동일
encoder output을
decoder의 input(key, value)으로 사용

Figure 1: The Transformer - model architecture.



번역할 문장과 번역된 문장의 정보를 바탕으로 다음의 올 단어를 예측

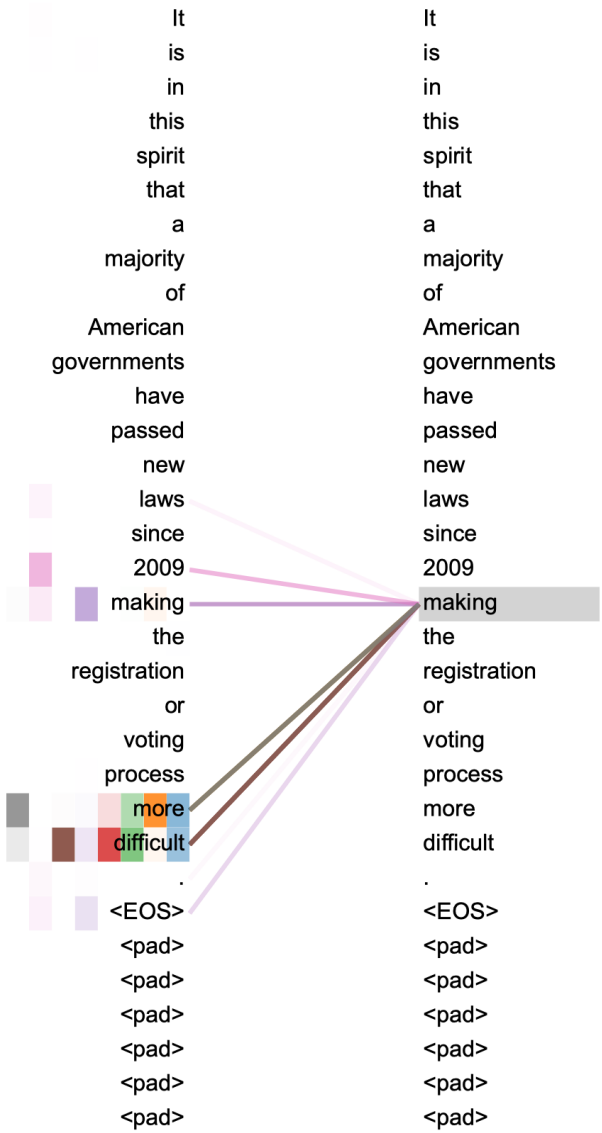
Figure 1: The Transformer - model architecture.

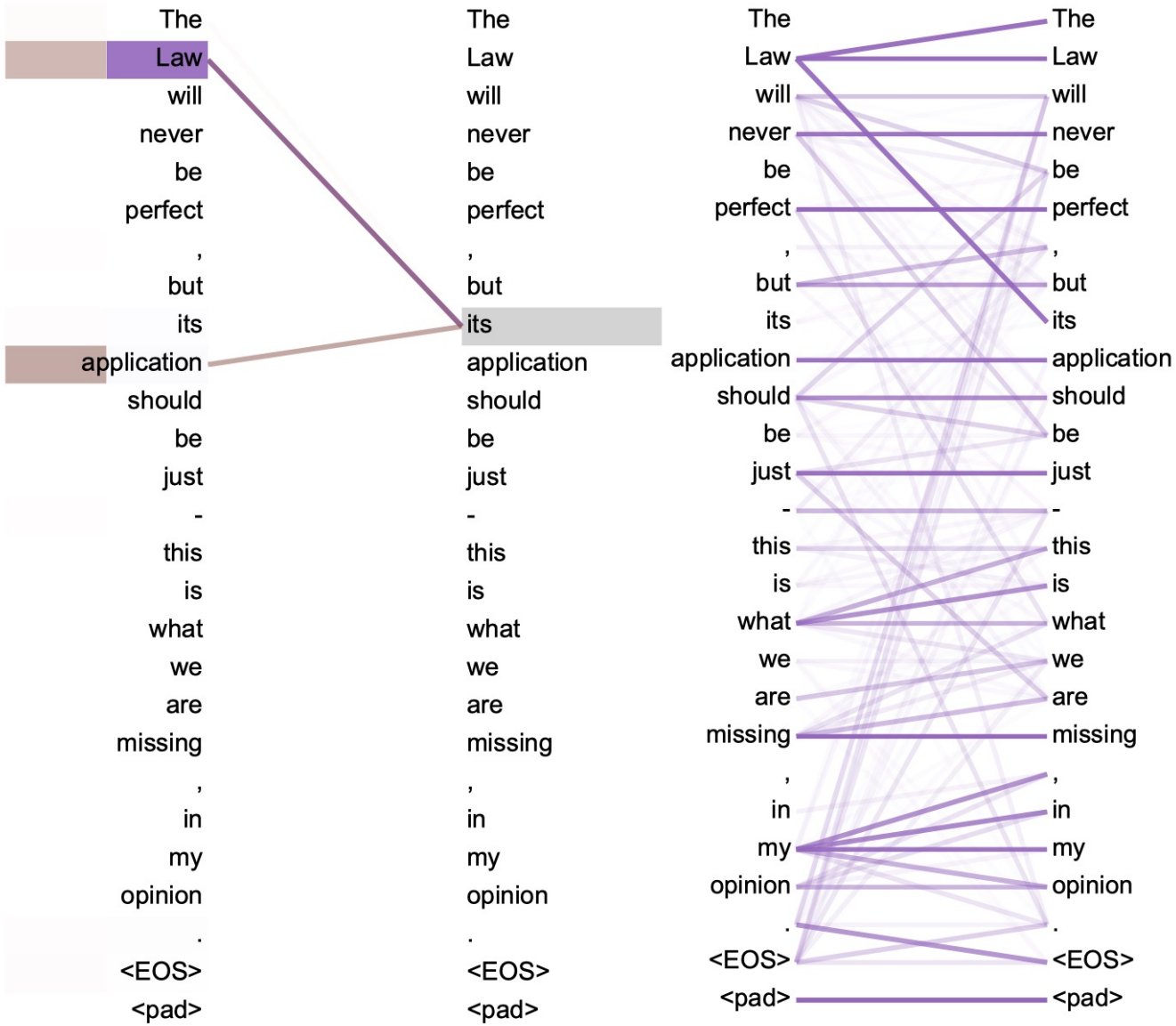
Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

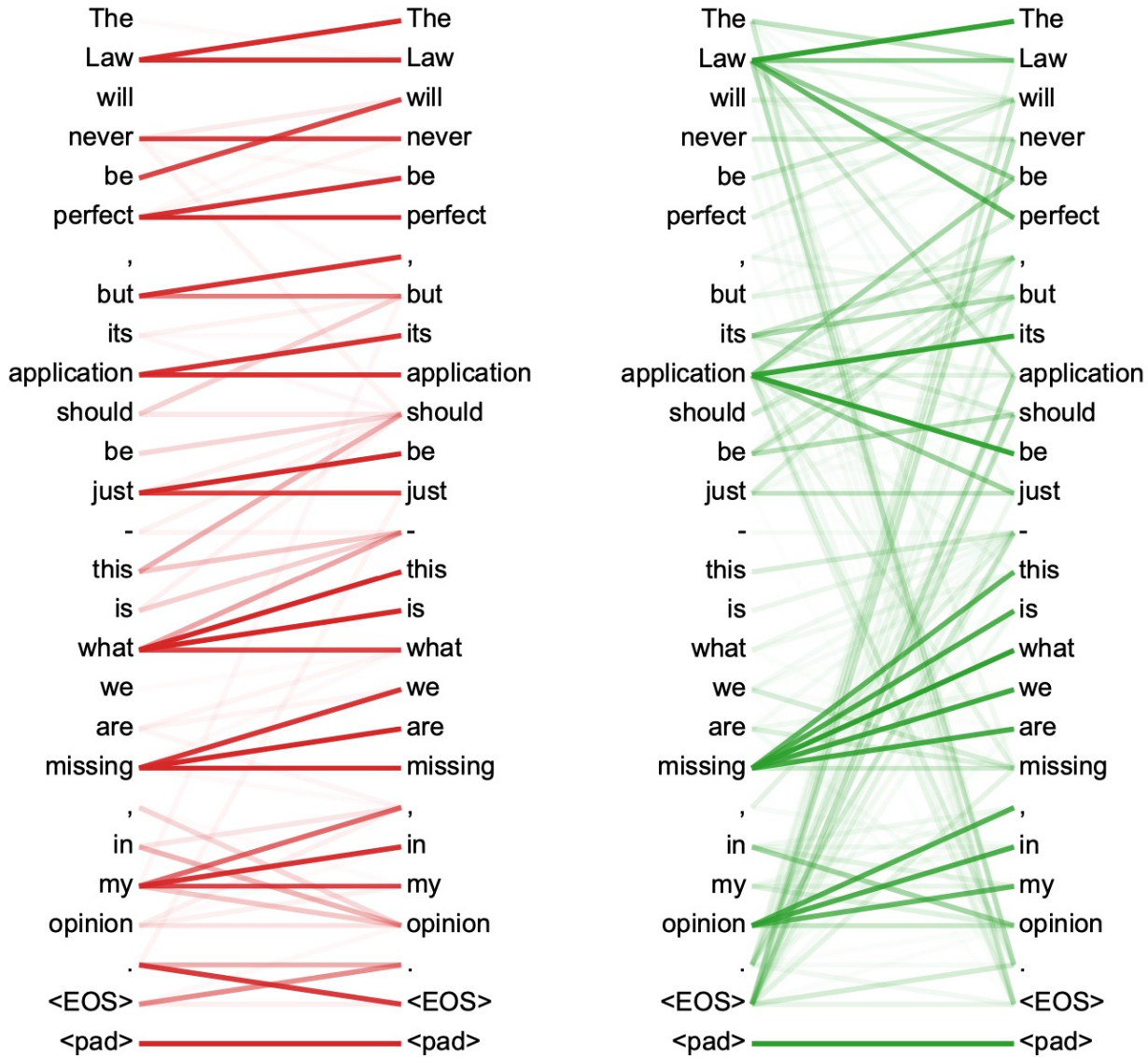
1. Data : standard WMT 2014 English – German dataset, English– French dataset
   - 4.5million, 3.6million개의 문장 pairs
   - 37000, 25000개의 tokens
2. Hardware and Schedule
   - 8대의 NVIDIA P100 GPU 사용
   - base model : 0.4s/step -> 100,000step : 12h 소요
   - big model : 1.0s/step -> 300,000step : 1d 소요
3. Optimizer : Adam
   - $lrate = d_{\text{model}}^{-0.5} \cdot \min(step\_num^{-0.5}, step\_num \cdot warmup\_steps^{-1.5})$
   - $\beta_1 = 0.9, \beta_2 = 0.98, \varepsilon = 10^{-9}$
4. Regularization
   - Dropout : p=0.1
   - Label smoothing(one-hot encoding을 약간 부드럽게 하여 모델이 너무 확산되지 않도록 함)
     - $y_i' = (1-\epsilon)y_i + \frac{\epsilon}{K}$
     - $\varepsilon = 0.1$

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

1. BLEU(번역된 문장의 품질 평가 점수)제일 높음
2. training cost 제일 작음(big model도 작은 축에 속함)


It
is
in
this
spirit
that
a
majority
of
American
governments
have
passed
new
laws
since
2009
making
the
registration
or
voting
process
more
difficult
.
<EOS>
<pad>
<pad>
<pad>
<pad>
<pad>
<pad>
It
is
in
this
spirit
that
a
majority
of
American
governments
have
passed
new
laws
since
2009
making
the
registration
or
voting
process
more
difficult
.
<EOS>
<pad>
<pad>
<pad>
<pad>
<pad>
<pad>


The
Law
will
never
be
perfect
,
but
its
application
should
be
just
-
this
is
what
we
are
missing
,
in
my
opinion
.
<EOS>
<pad>
The
Law
will
never
be
perfect
,
but
its
application
should
be
just
-
this
is
what
we
are
missing
,
in
my
opinion
.
<EOS>
<pad>
The
Law
will
never
be
perfect
,
but
its
application
should
be
just
-
this
is
what
we
are
missing
,
in
my
opinion
.
<EOS>
<pad>
The
Law
will
never
be
perfect
,
but
its
application
should
be
just
-
this
is
what
we
are
missing
,
in
my
opinion
.
<EOS>
<pad>


The
Law
will
never
be
perfect
,
but
its
application
should
be
just
-
this
is
what
we
are
missing
,
in
my
opinion
.
<EOS>
<pad>
The
Law
will
never
be
perfect
,
but
its
application
should
be
just
-
this
is
what
we
are
missing
,
in
my
opinion
.
<EOS>
<pad>

감사합니다