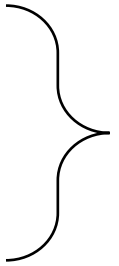
# Introduction to Transformer

# Contents

seq2seq
- encoder-decoder structure

attention mechanism
- dot product attention, scaled dot product attention

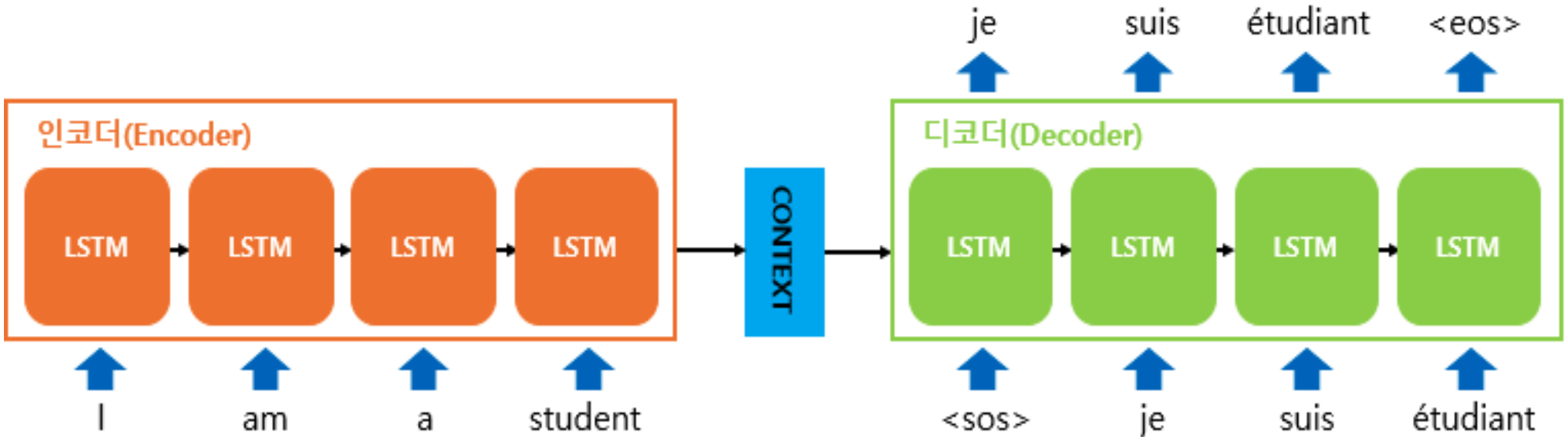
Attention is all you need (transformer)
- positional encoding
- multi head attention
- self attention
- masked attention
- transformer architecture



homework

# seq2seq

Introduced by Sutskever et al. in *Sequence to Sequence Learning with Neural Networks*




Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, *27*.

# seq2seq

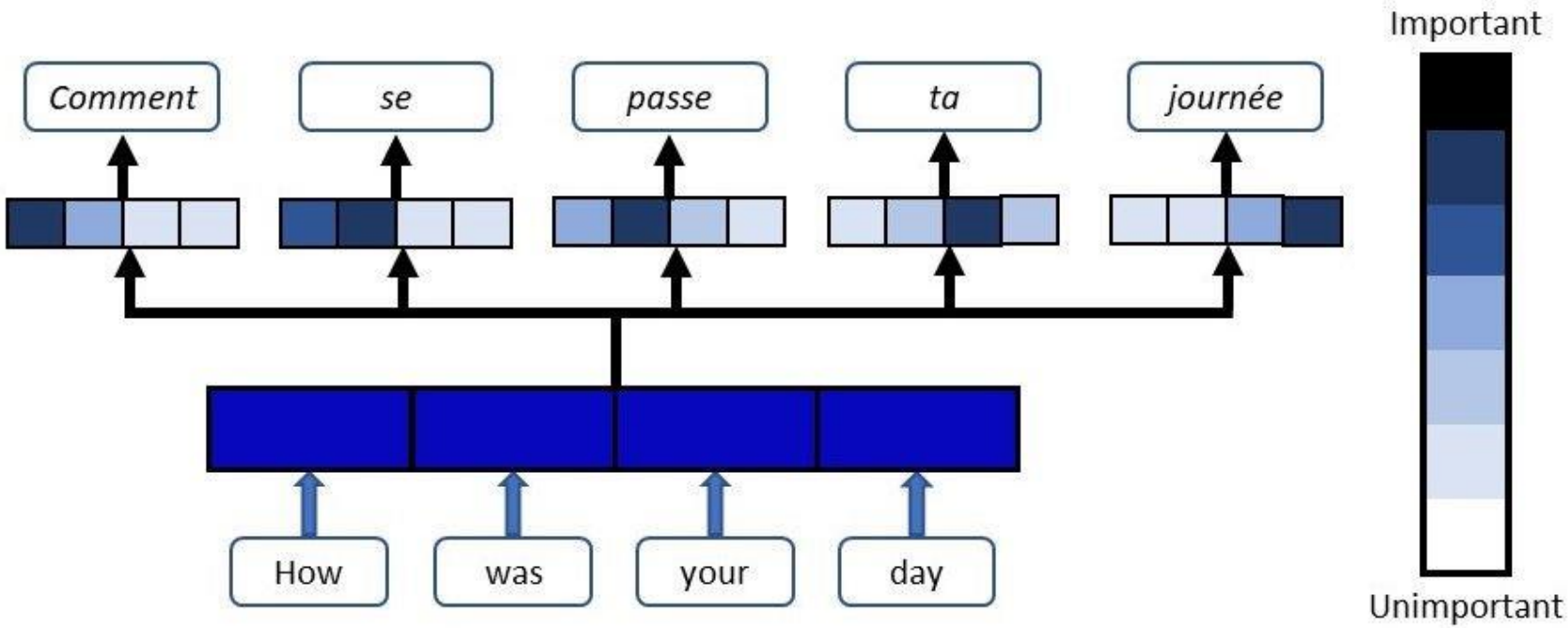
seq2seq의 문제점

input sequence 전체를 고정된 크기의 context vector로 표현함
- 입력 시퀀스가 길어질수록 정보의 손실이 커짐

RNN 구조 기반
- gradient vanishing / exploding이 발생함
- Recurrent 구조이므로 입력 시퀀스가 길어질수록 계산량이 많아짐

# attention

특정 토큰에 대해 다른 토큰들과의 상관관계를 모델링



# attention

각 토큰은 key-value 구조로 구성된 정보를 가짐

key-value 구조의 예시) python의 dictionary 자료형

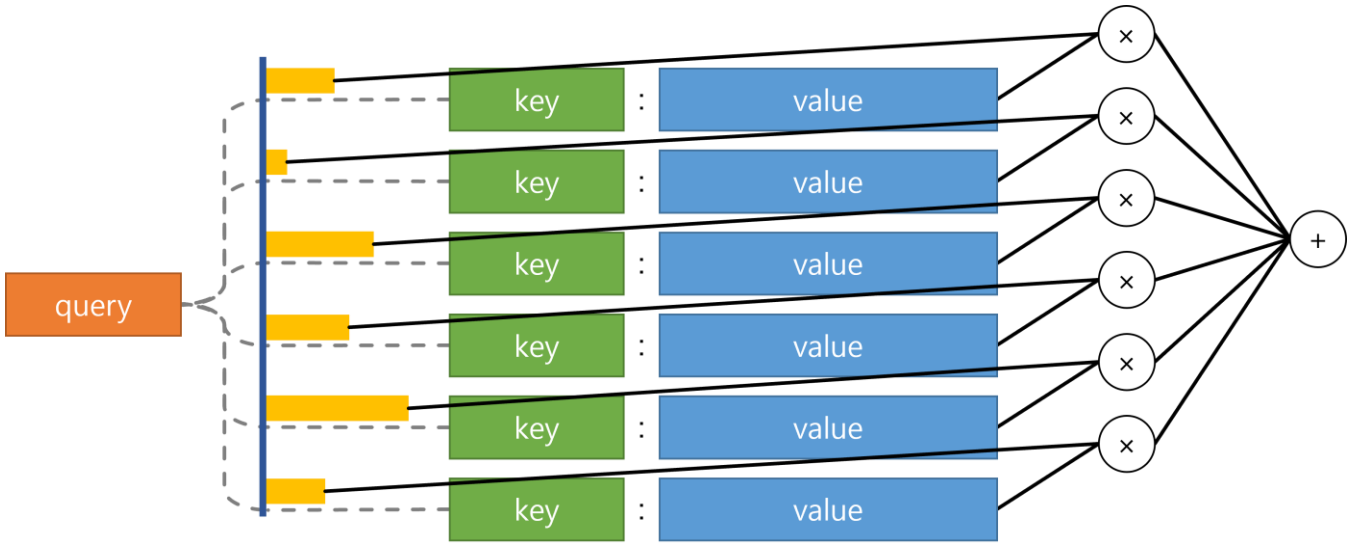
```python
data = {
  'alpha': 2023,
  'beta': 2017,
  'gamma': 2015,
  'delta': 2010
}

print(data['alpha'])  # 2023
print(data['delta'])  # 2010
```

# attention

Attention(Q, K, V) = attention value

주어진 Query(Q)에 대해
모든 Key(K)와의 유사도를 구함
유사도에 따라 각 Key에 mapping된 Value(V) 값을 가중합하여 리턴
(soft mapping)



# attention

상관관계를 수치화하는 방법
– euclidean distance, cosine similarity, dot similarity 등…

| 이름 | 식 | 출처 |
| --- | --- | --- |
| content-based attention | $f(s, h) = \frac{s^T h}{\|\|s\|\| \cdot \|\|h\|\|}$ | Graves, 2014 |
| additive attention (Bahdanau attention) | $f(s, h) = V^T \tanh(W_1 s + W_2 h)$[4] | Bahdanau, 2015 |
| dot-product attention (Loung attention) | $f(s, h) = s^T h$ | Luong, 2015 |
| scaled dot-product attention | $f(s, h) = \frac{s^T h}{\sqrt{n}}$[5] | Vaswani, 2017 |

In Transformer – dot product (scaled dot product) attention 주로 사용

# (scaled) dot product attention

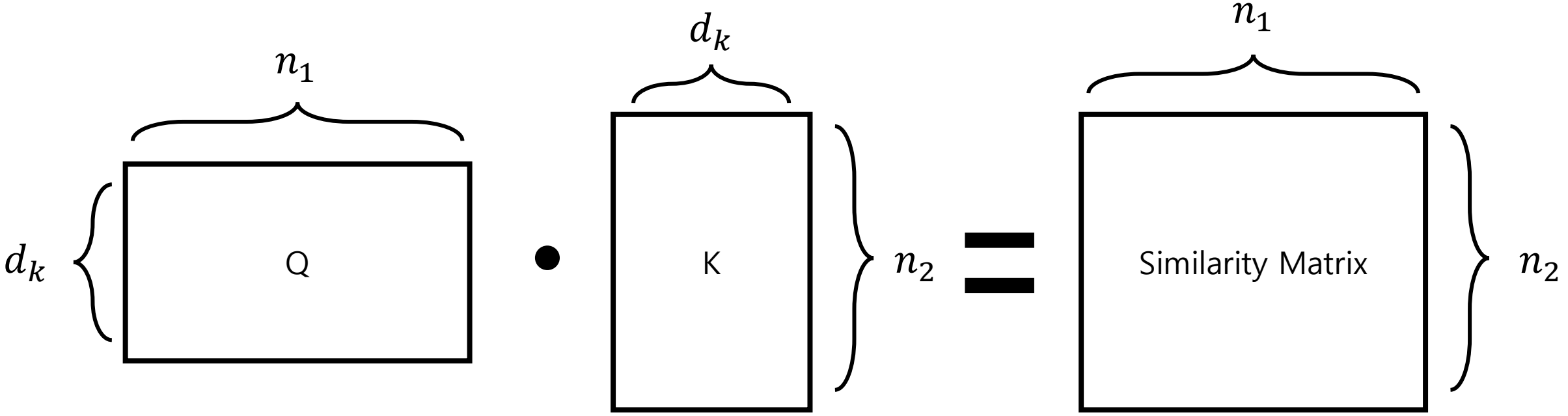
$$Q = W_Q x_1 + b_Q$$
$$K = W_K x_2 + b_K$$
$$V = W_V x_2 + b_V$$

$x_1$ : Querying token (or sequence)
$x_2$ : Queried token (or sequence)

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

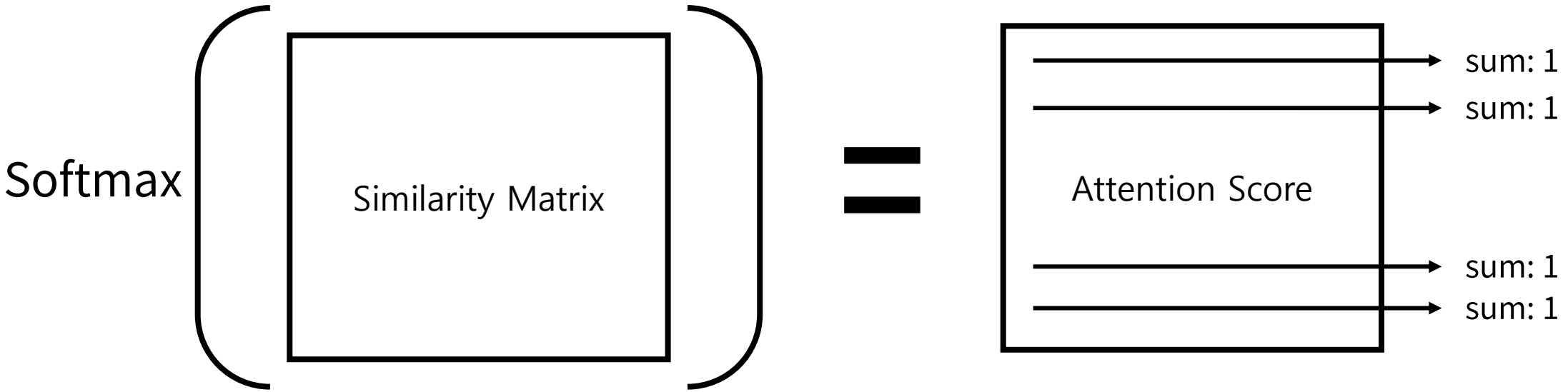
식을 뜯어 봅시다

# (scaled) dot product attention

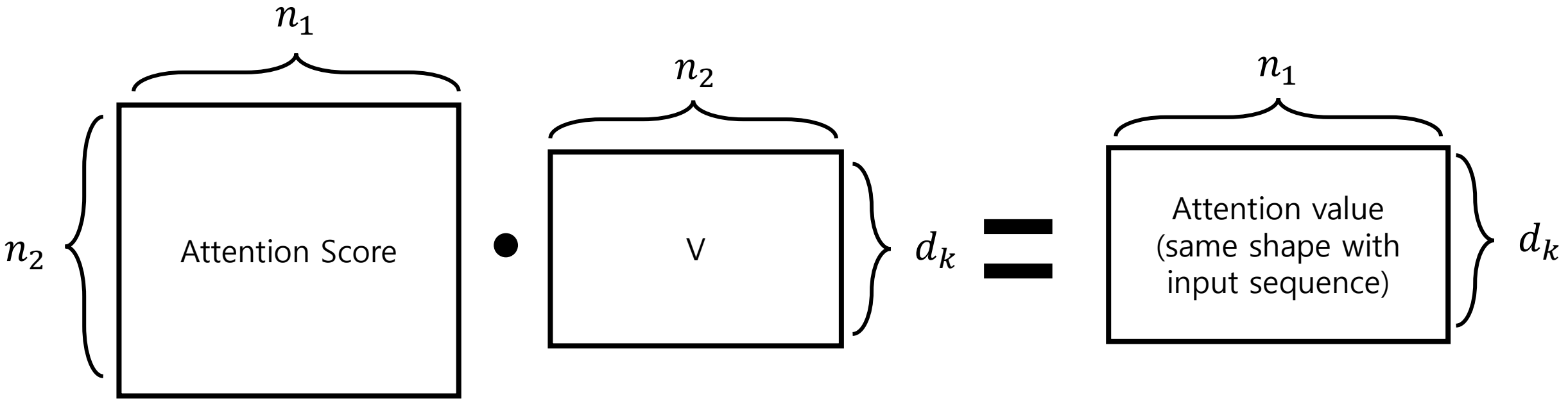



$d_k$ : Q, K, V 벡터의 차원
$n_1$ : Querying sequence의 길이
$n_2$ : Queried sequence의 길이

# (scaled) dot product attention



# (scaled) dot product attention



# seq2seq with attention

seq2seq의 문제점

input sequence 전체를 고정된 크기의 context vector로 표현함
- 입력 시퀀스가 길어질수록 정보의 손실이 커짐

→ decoder에서 encoder의 상태를 attention을 통해 반영하여 해결

# transformer

seq2seq의 문제점

RNN 구조 기반
- gradient vanishing / exploding이 발생함
- Recurrent 구조이므로 입력 시퀀스가 길어질수록 계산량이 많아짐

→ RNN을 제거하고 Attention만 사용 (attention is all you need)

# transformer

Introduced by Vaswani et al. in *Sequence to Sequence Learning with Neural Networks*

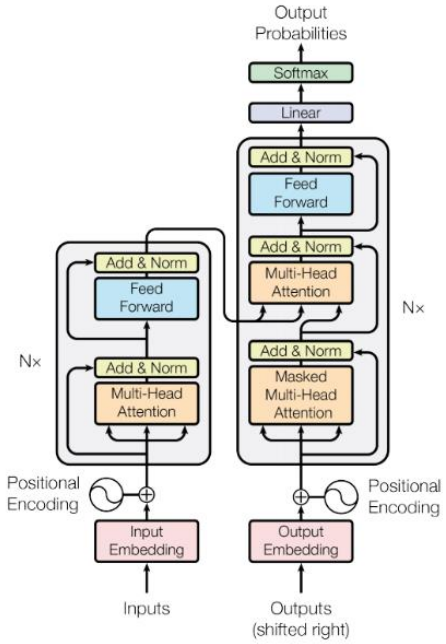



Figure 1: The Transformer - model architecture.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

# transformer

seq2seq의 encoder-decode구조를 차용

lstm을 mlp 레이어로 대체
- layer를 여러층으로 쌓을 수 있음
- RNN 기반 모델에서 발생하는 문제 해결

lstm을 제거할 경우 생기는 문제점
- token들 간의 위치 정보를 반영하지 못함

# 다음 회합 전까지

transformer에서 사용된
- positional encoding
- self attention
- multi-head attention
- masked attention
- layer normalization
- residual connection

- 위 키워드를 기반으로 transformer architecture 공부해오기
  + 노션 정리