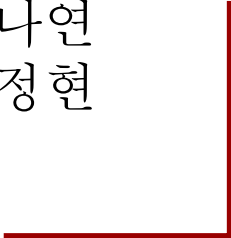




Sequence-to- Sequence model

임형준 강유정 김나연
박진우 김근호 김정현



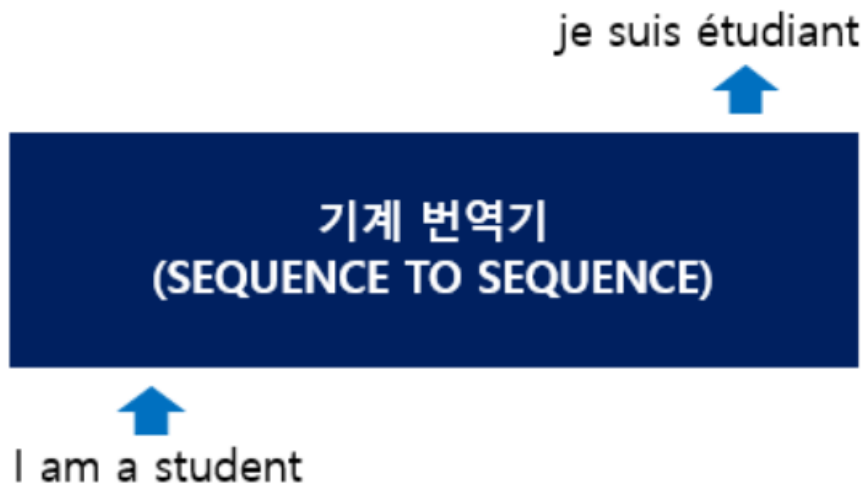


1. Sequence-to-Sequence

2. Transformer

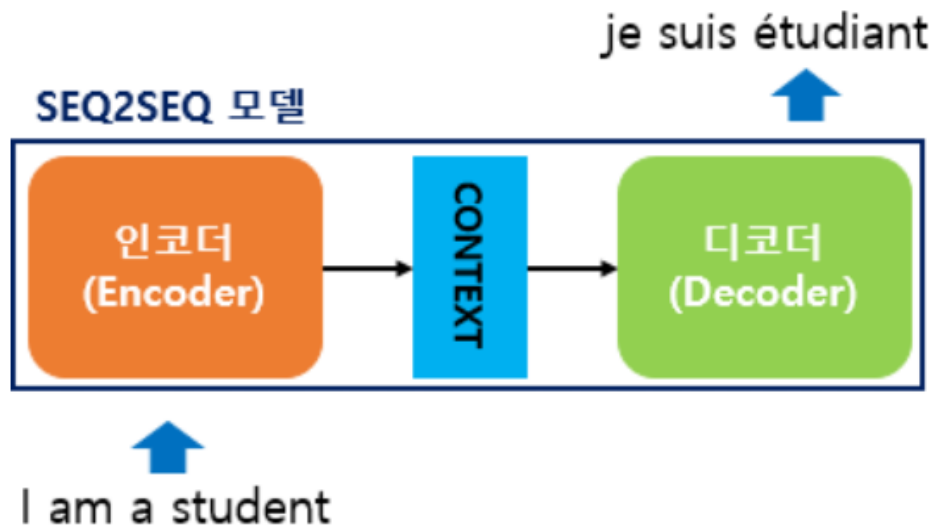
3. 항후계획

Sequence-to-Sequence



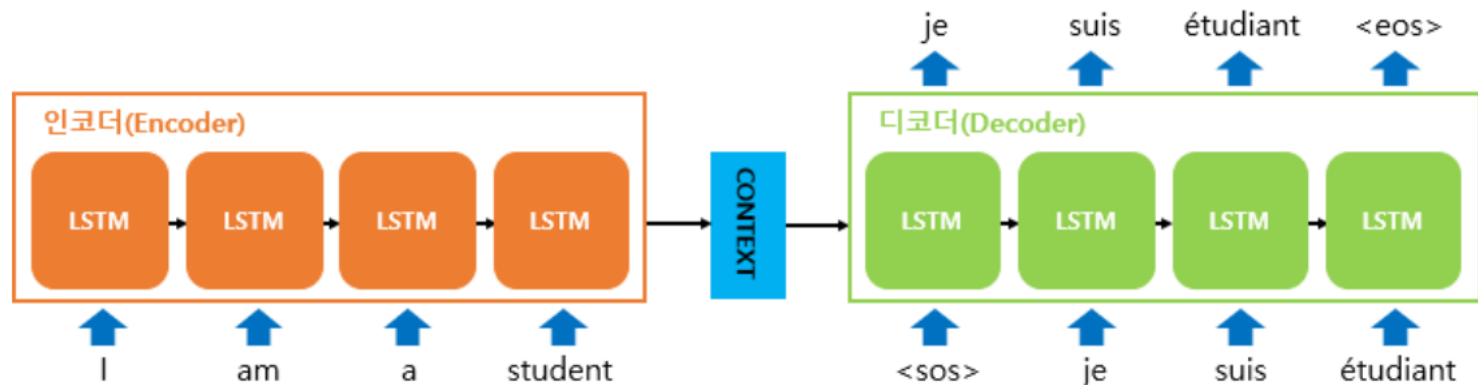
- 시퀀스-투-시퀀스(sequence-to-sequence)에서 시퀀스란 연관된 연속의 데이터를 의미
- 하나의 텍스트 문장이 입력으로 들어오면 하나의 텍스트 문장을 출력하는 구조
- 주로 번역기나 챗봇에 활용됨

Sequence-to-Sequence



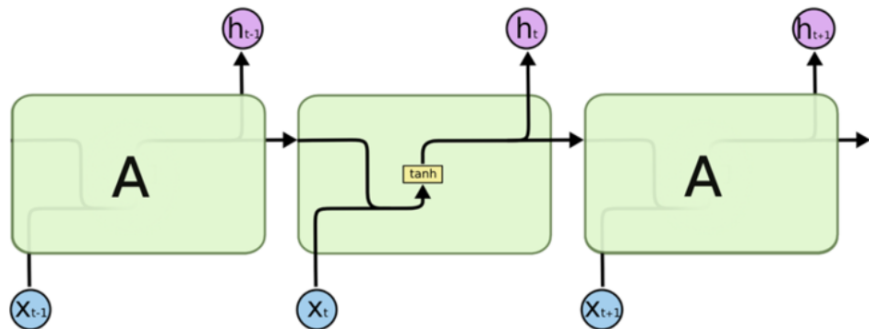
- seq2seq는 크게 두 개의 아키텍처로 구성
- 인코더와 디코더, context vector 존재
- 인코더 부분에서 입력 값을 받아 입력 값의 정보를 담은 벡터를 만들고 (context vector) 디코더에서 이 벡터를 활용해 재귀적으로 출력 값을 만드는 구조

Sequence-to-Sequence



- 인코더와 디코더 역시 내부는 RNN기반 → 성능 문제로 인해 실제로는 LSTM 셀들로 구성
- LSTM은 RNN이 입력 데이터와 참고 데이터의 위치가 멀어지면 문맥을 연결하기 힘들어지는 문제를 보완하기 위해 사용(RNN의 한 종류)

Sequence-to-Sequence

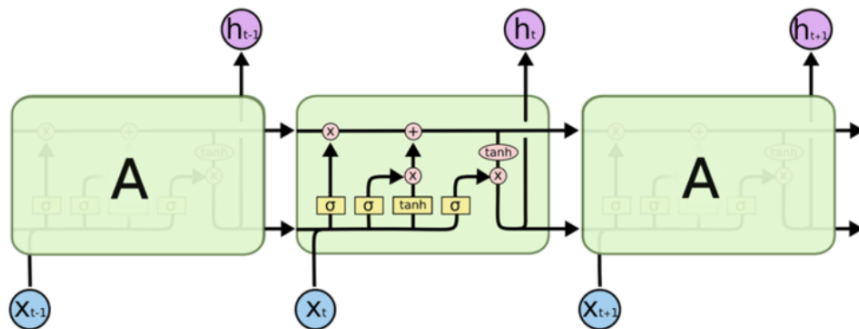


RNN

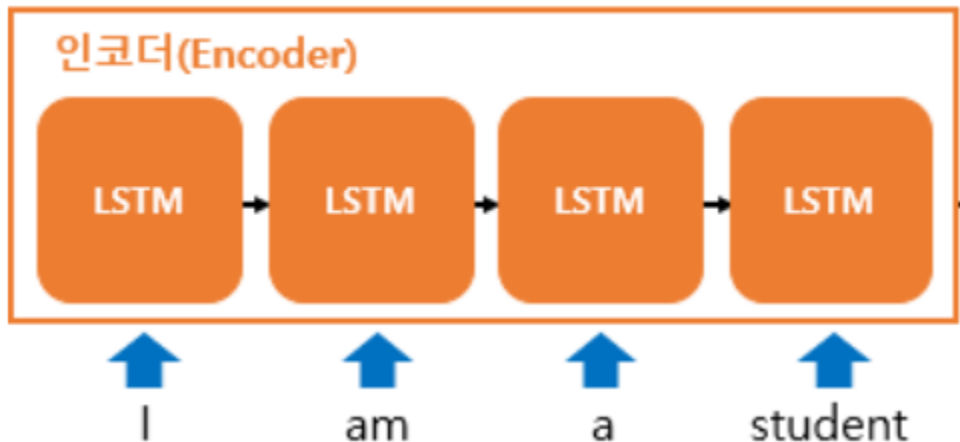
싱글레이어를
가지고 반복됨

LSTM

상호작용하는
4개의 레이어가
반복되는 구조

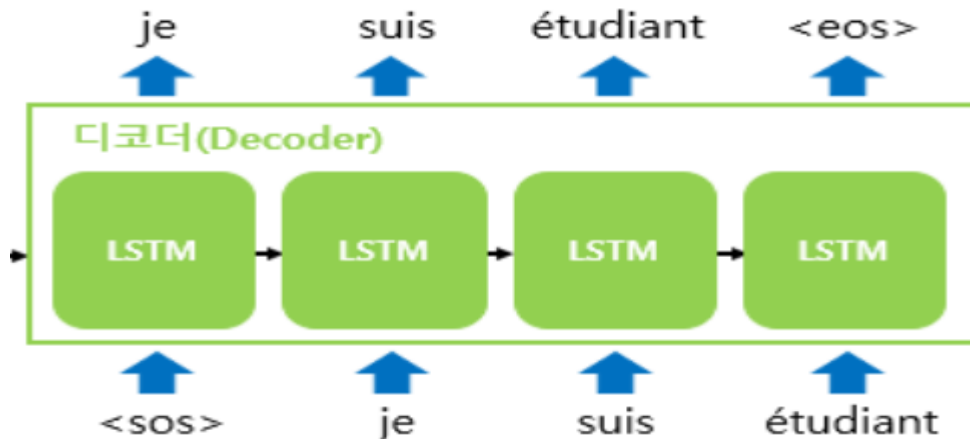


Sequence-to-Sequence



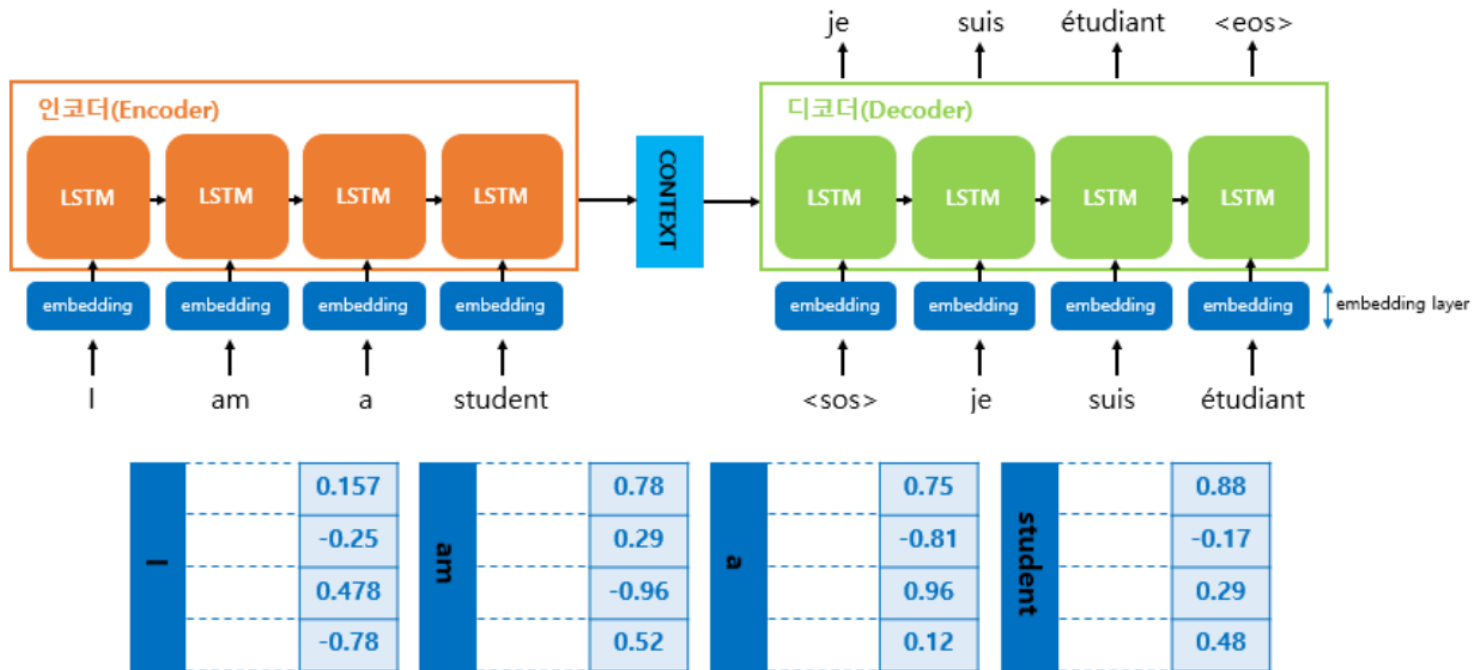
- 단어 토큰화 통해 단어 단위로 쪼개짐
- 각각의 단어 토큰이 셀의 입력
- 모든 단어를 입력 받은 후 마지막 셀의 은닉 상태를 디코더 셀로 넘겨줌 (컨텍스트 벡터)
- 컨텍스트 벡터가 디코더의 첫 번째 셀의 은닉상태로 활용

Sequence-to-Sequence

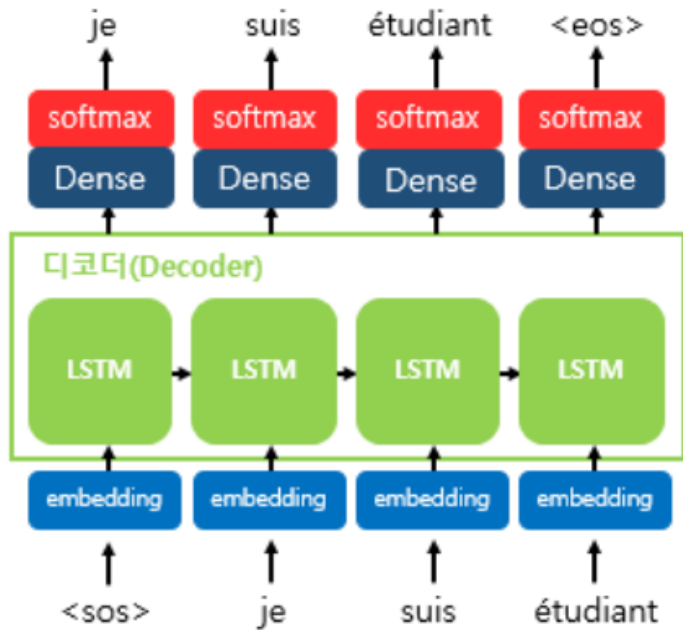


- 초기 입력은 문장의 시작을 나타내는 'sos'
- 'sos' 입력되면 디코더는 단어를 예측
- 순차적으로 단어를 예측해서 문장의 끝을 의미하는 'eos' 등장할 때까지 반복

Sequence-to-Sequence



Sequence-to-Sequence



-출력 단어로 나올 수 있는 여러 단어들 중 seq2seq 모델이 하나 선택해서 예측

-예측할 때 쓰는 함수가 softmax, 이 함수를 통해 각 확률 값을 반환하고 디코더가 출력 단어 결정



Transformer Model



Sequence-to-Sequence 의 한계

RNN에 기반한 seq2seq 모델

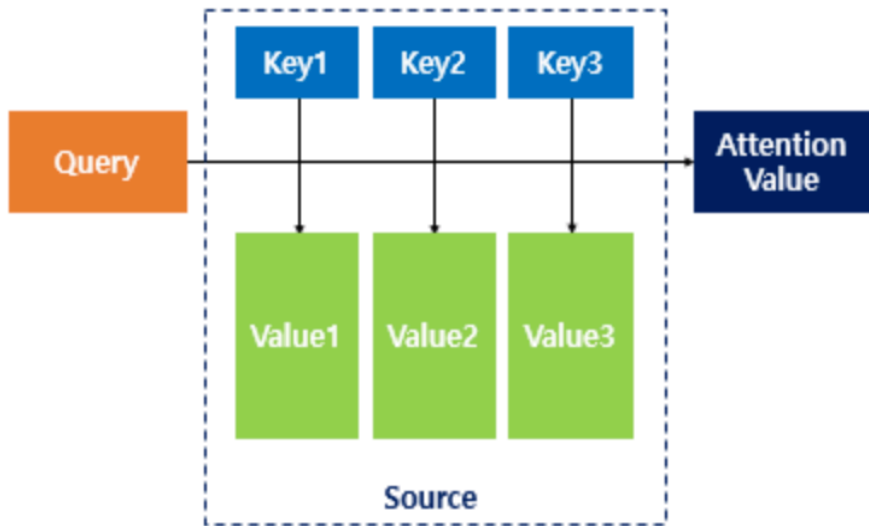
1. 하나의 고정된 크기의 벡터에 모든 정보를 압축

-> 정보 손실

2. RNN의 고질적인 기울기 소실(Vanishing Gradient) 문제

-> 입력 문장이 길면 번역 품질이 떨어지는 현상

Attention Mechanism

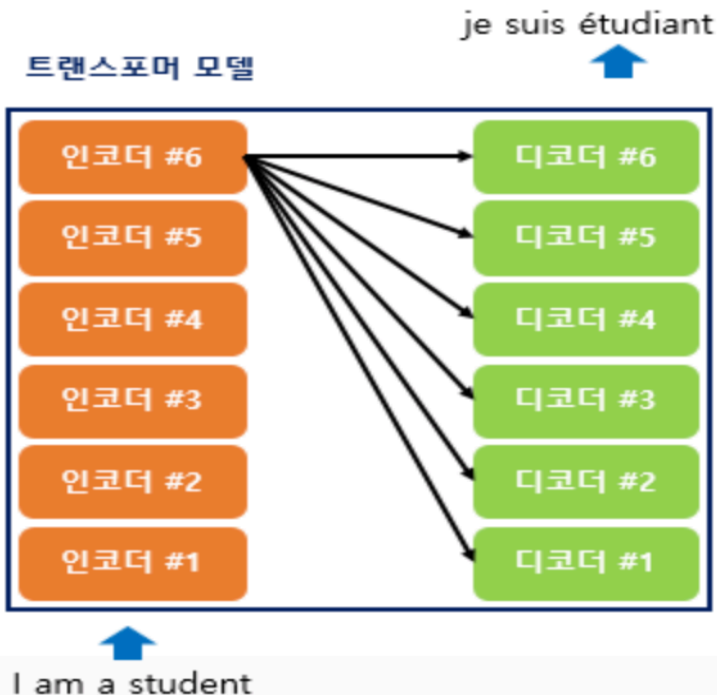


- 디코더에서 출력 단어를 예측하는 매 시점마다, 인코더에서의 전체 입력 문장을 다시 한 번 참고
- 쿼리(Query)에 대한 모든 키(Key)와의 유사도
- 유사도를 가중치로 해서 Value에 반영

결국, 전체 입력 문장을 동일한 비율로 참고 X
해당 시점에서 예측해야 할 단어와
연관이 있는 입력 단어 부분에 좀 더 집중

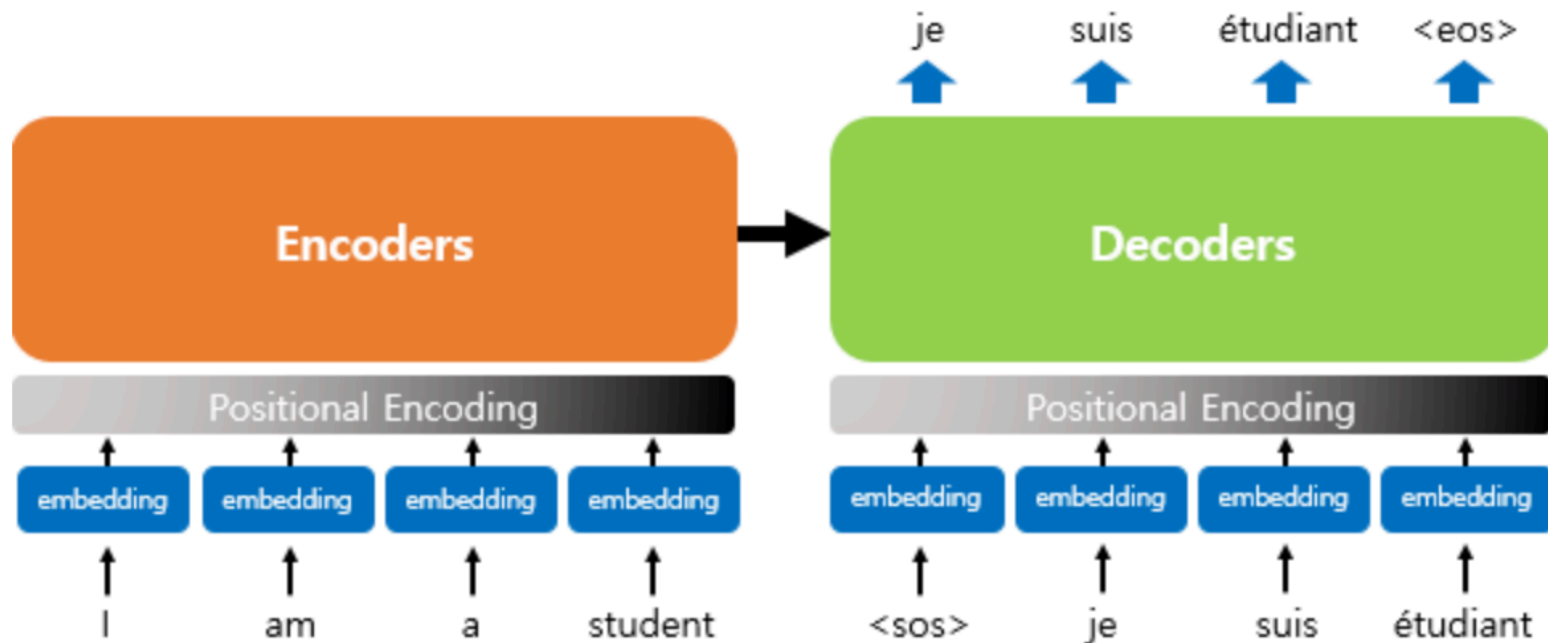
- $\text{Attention}(Q, K, V) = \text{Attention Value}$

Transformer

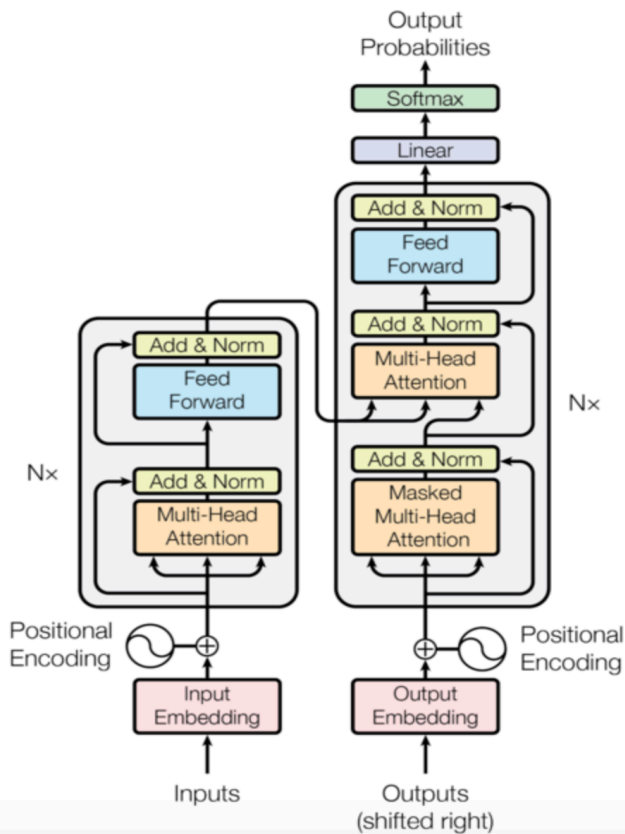


- 논문 "Attention is all you need " 의 모델
- seq2seq의 인코더-디코더 구조
- +
RNN대신 어텐션으로 구현한 모델
- 학습 속도가 빠르고 RNN보다 성능 우수
- N개의 인코더, 디코더 구조

Positional encoding



Encoder + Decoder

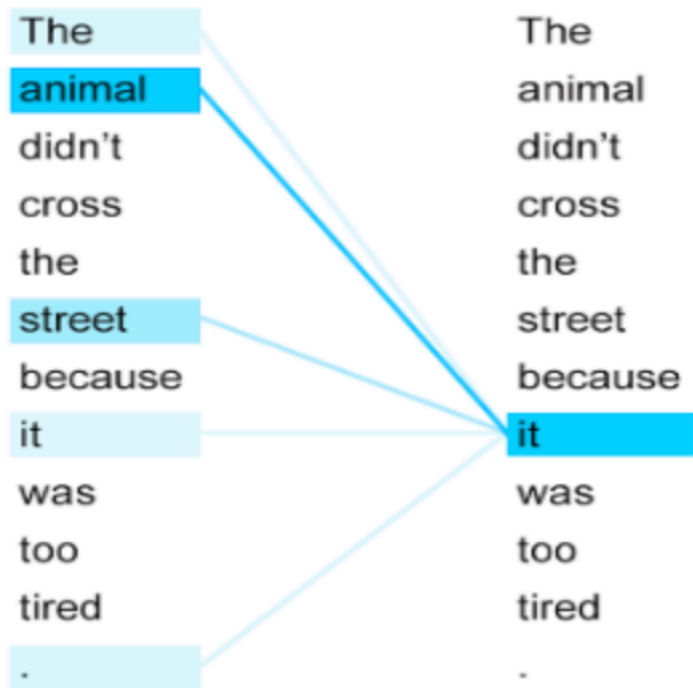


- N개의 인코더, 디코더로 구성
- 서브층: **셀프어텐션**, **피드 포워드 신경망**

(모듈)

- 멀티 헤드 어텐션
- 서브시퀀스 마스크 어텐션
- 포지션-와이즈 피드 포워드 네트워크
- 리지듀얼 커넥션

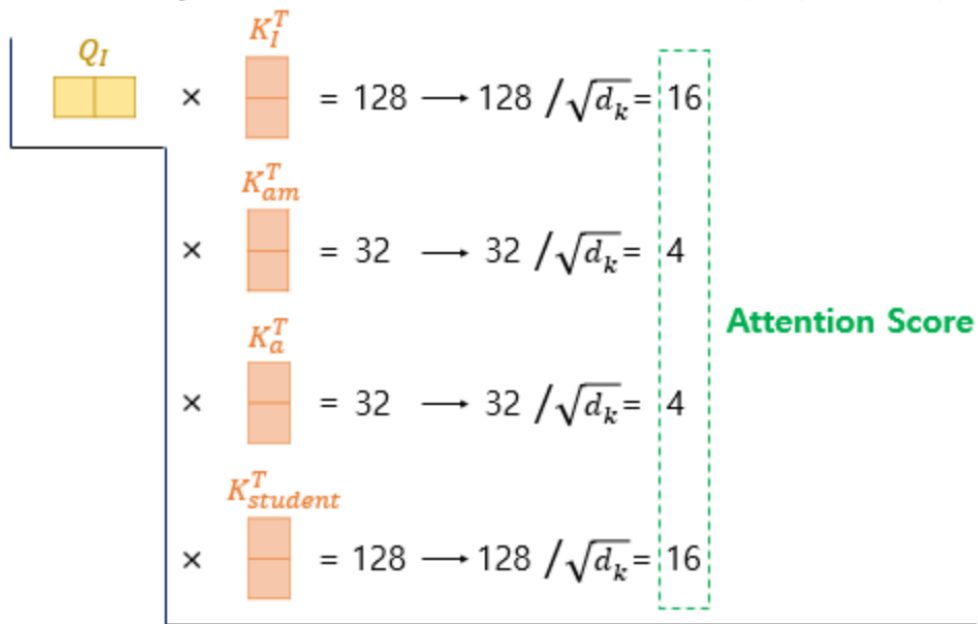
Self-attention



- 문장 안에서 단어들 간의 관계 측정
- 어텐션 스코어:
다른 단어들과의 관계값
- 어텐션 맵:
어텐션 스코어 값을 하나의 테이블로 표현

Multi-head attention / Scaled dot product attention

Scaled dot product Attention : $score\ function(q, k) = q \cdot k / \sqrt{n}$



셀프 어텐션 구조에서 **Scaling 처리**

Query와 value를 이용해 내적을 한 값이 벡터의 차원이 커지면 학습이 잘 안 될 수 있으므로

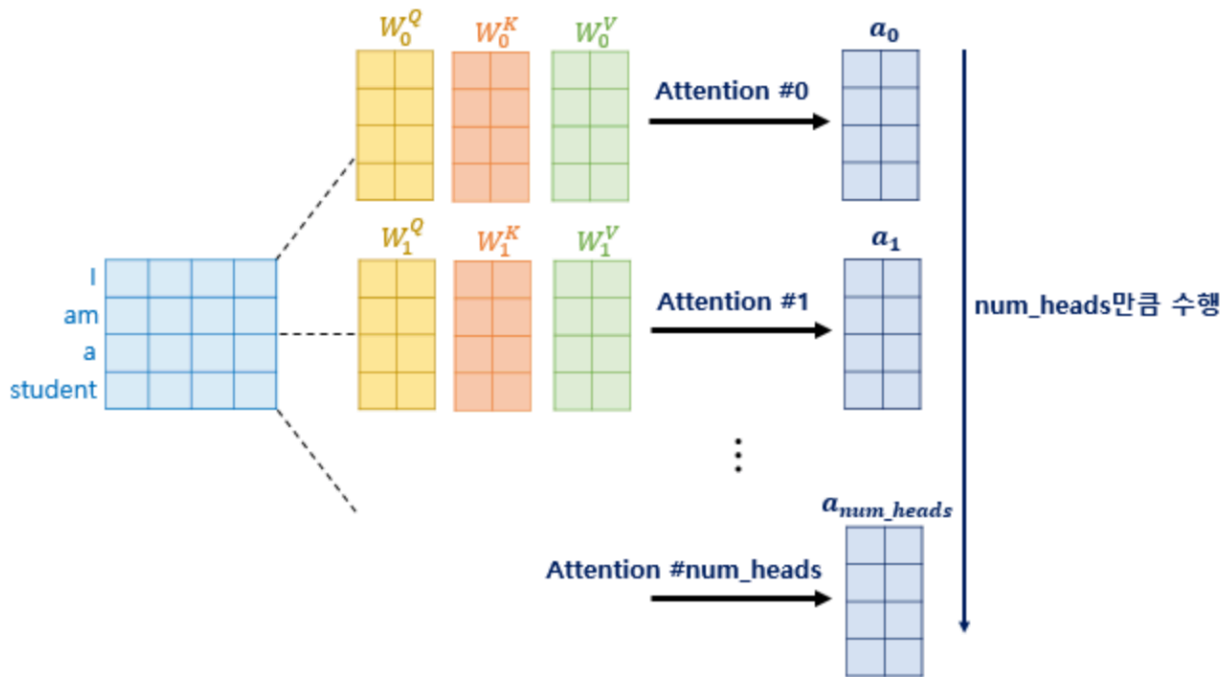
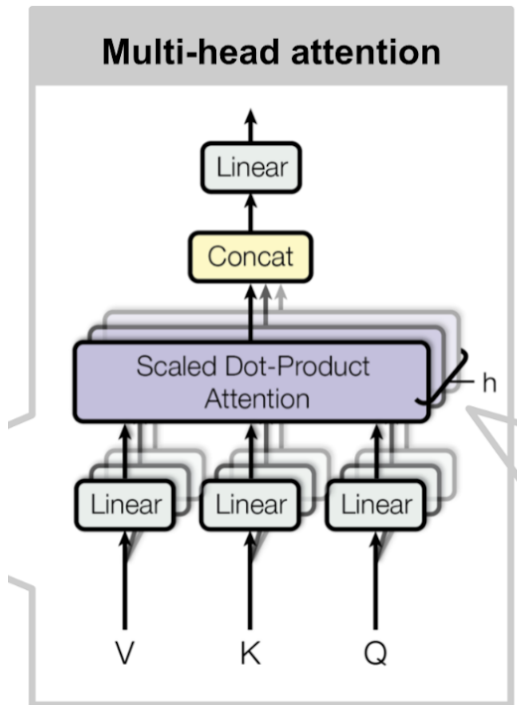
벡터 크기에 반비례하도록 크기 조정

Key 벡터의 차원수를 제곱근한 값으로 나눈 후 소프트맥스 함수 적용

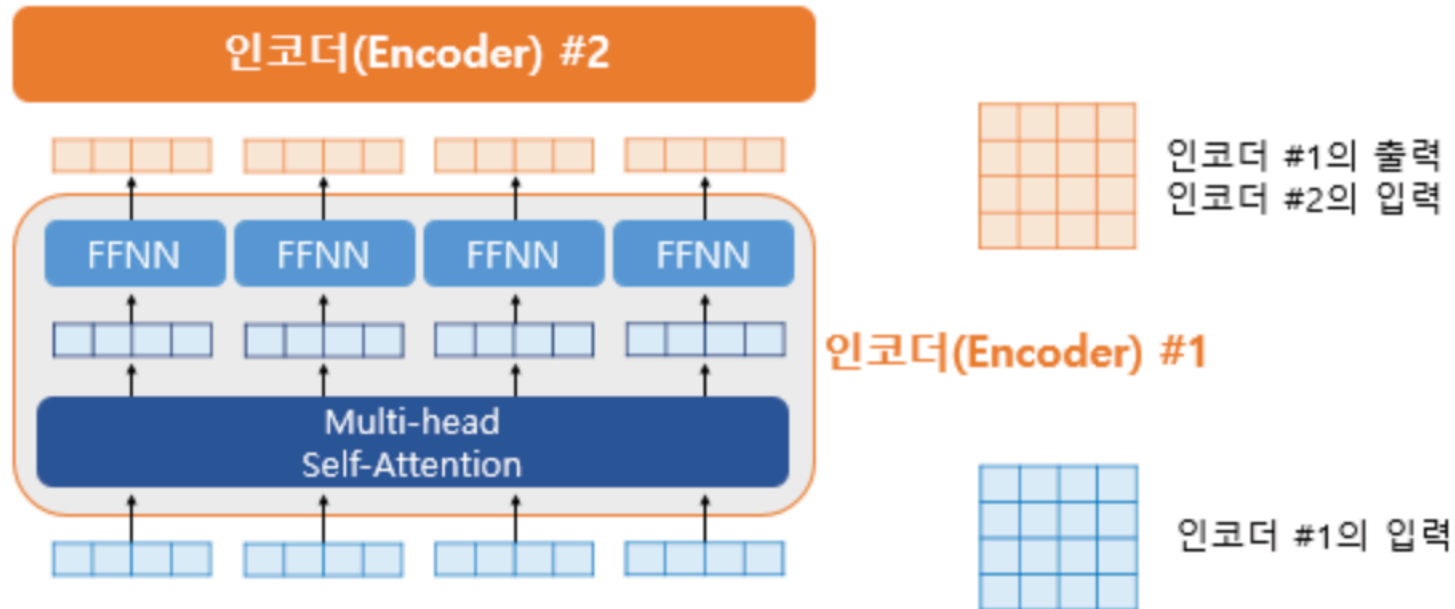
Multi-head attention / 순방향 마스크 어텐션

	자연어	처리	아주	좋아요
자연어				
처리				
아주				
좋아요				

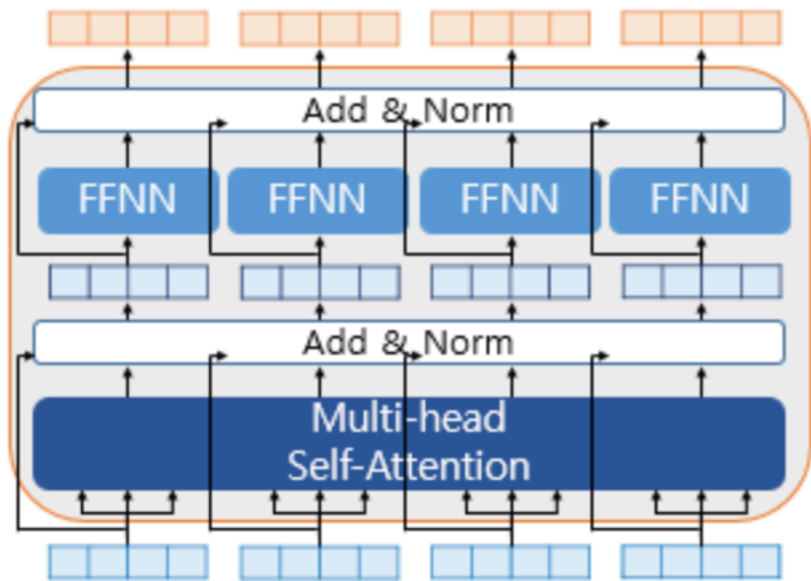
Multi-head attention



포지션-와이즈 피드 포워드 네트워크

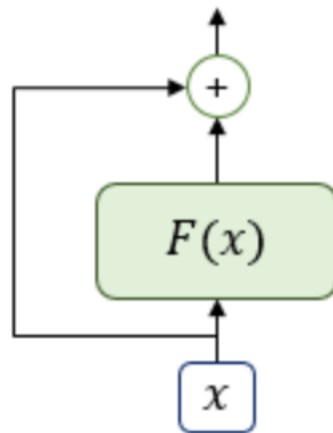


리지듀얼 커넥션, 층 정규화

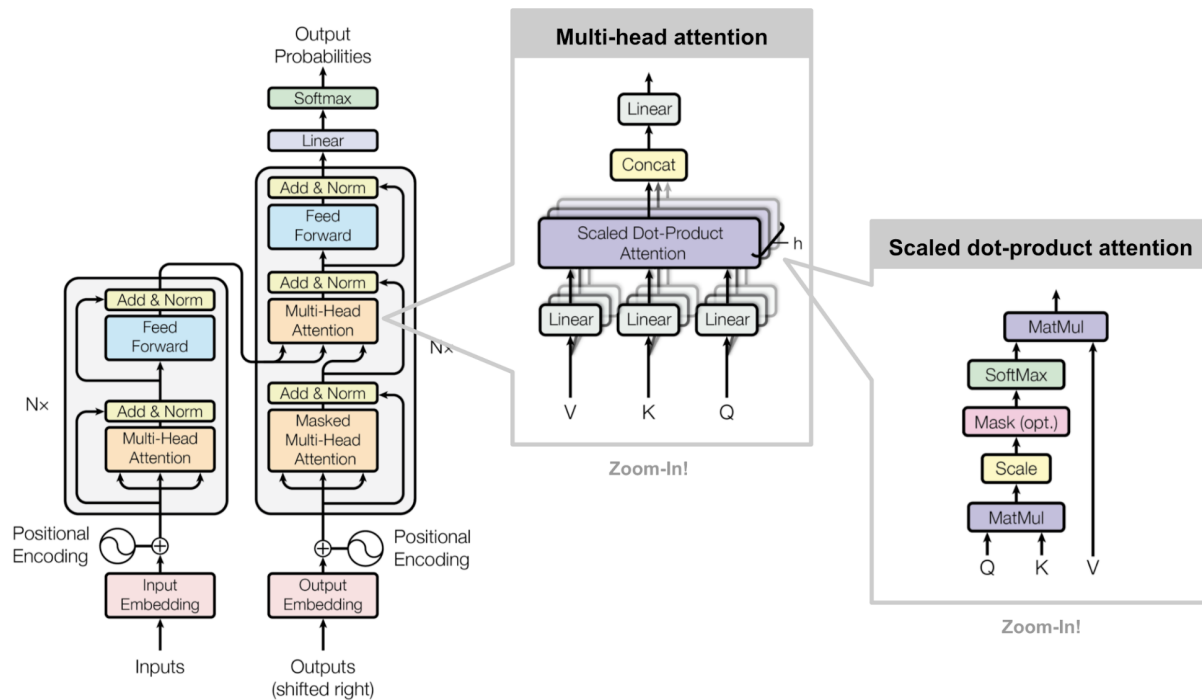


인코더(Encoder) #1

$$H(x) = x + F(x)$$



SUM UP



그래서 대체 무엇을 할 것인가!!!!!!



“보고싶었어 죽을만큼.. 다신 안봐줄거야
약속했었지? 돌아오면 책임져 준다고
이 구준표님이랑 결혼해줘!”

COMING
SOOOOOON~~~~~