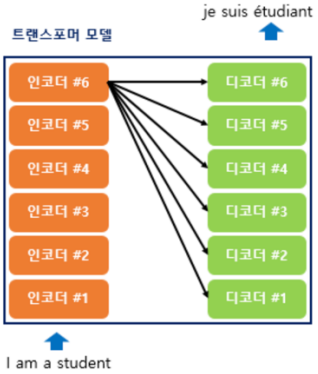
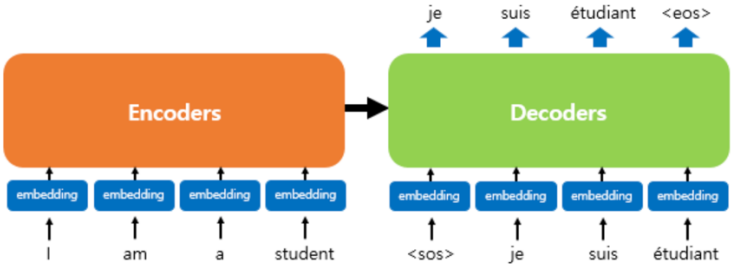
- Transformer





1. Encoder – Decoder 구조

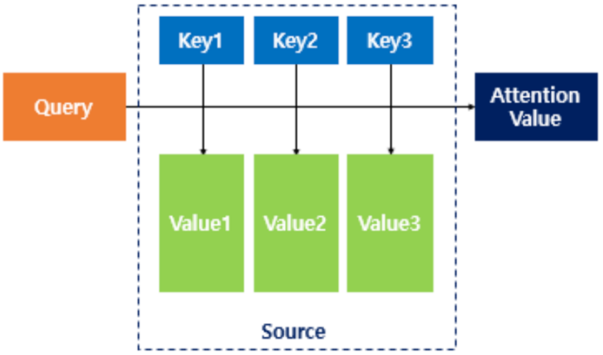
: Encoder 는 입력을 Decoder 는 출력하는 구조

→ Encoder 만 사용하면 분류모델(ex. BERT), Decoder 만 사용하면 생성모델(ex. GPT)로도 구분할 수 있음

2. 〈SOS〉 문장 〈EOS〉로 출력됨 → Seq2Seq 구조

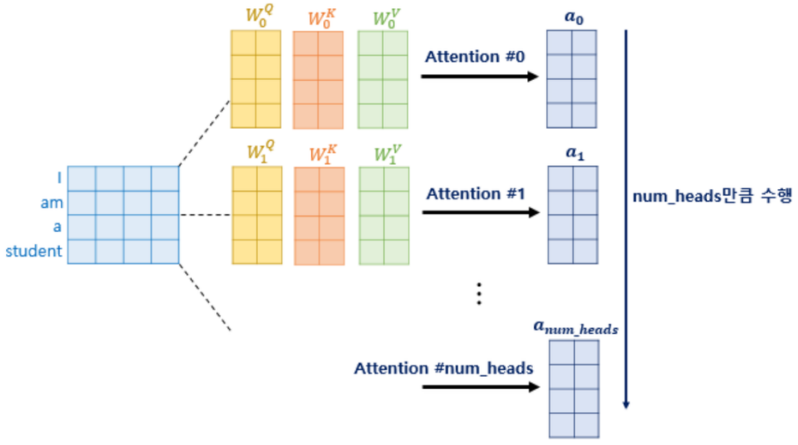
3. 여러 레이어(층)으로 구성

- Encoder 의 Self-Attention



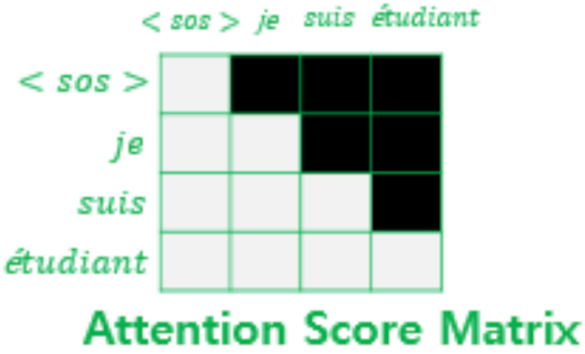
: Encoder 내부에서 각 토큰은 Query-Key-Value 기반 Self-Attention 을 통해 문맥 정보 통합

- Encoder 의 Multi-Head Attention



: 여러 Attention 을 병렬 수행하여 다양한 문맥 관계를 동시에 학습 (num_heads 만큼 병렬 수행)

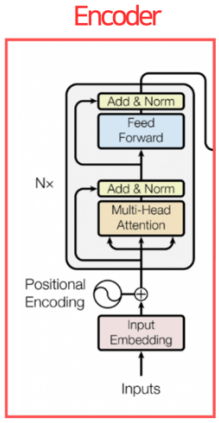
- Decoder 의 Masked Multi-Head Attention



: 이후 토큰을 가려 이전 토큰만을 참조하여 다음 토큰을 예측하도록 함

→ Attention 으로 토큰 간 관계를 학습하고, FFN 으로 표현력을 강화하며, Residual Connection 과 Layer Normalization 으로 깊은 구조의 학습을 안정화함

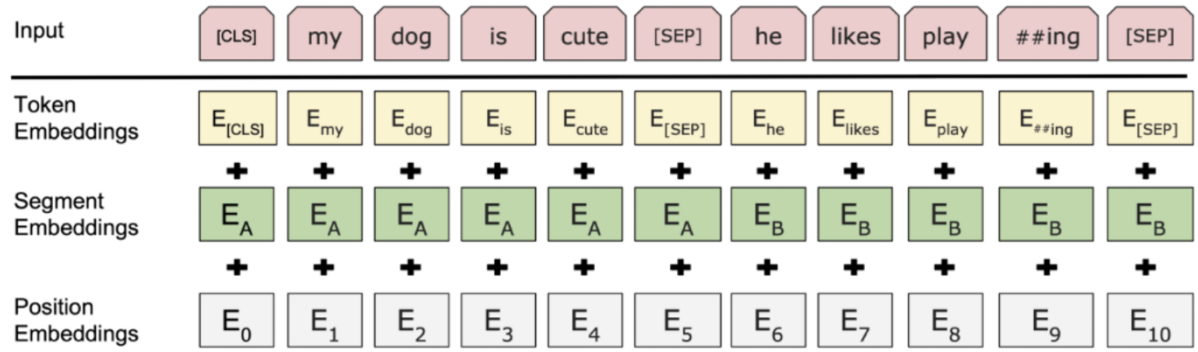
- BERT



: Transformer 의 Encoder 만 사용, Self-Attention 을 통해 문맥 동시 반영

- BERT 의 Input Embedding

$$E_i = E_i^{token} + E_i^{segment} + E_i^{position}$$



: Token (WordPiece 기반 서브워드), Segment(문장 구분), Position(토큰 순서, 상대적 위치)

- MLM

: 좌우 문맥을 동시에 활용하여 전체 토큰 중 15%만을 선택하여 예측

- NSP

: 두 문장이 연속된 문장인지에 대한 이진 분류(NotNext - 0, IsNext - 1)

→ Transformer 의 Encoder 구조를 기반으로 MLM, NSP 를 활용해 문장을 양방향으로 이해하는 문맥 표현을 사전학습한 모델