Bert

BERT : **B**idirectional **E**ncoder **R**epresentations from **T**ransformers

Transformer 기반, 그러나 인코더만 사용함

차이점1. Input Embeddings

- Positional Encoding을 사용하지 않고 대신 Position Embeddings를 사용

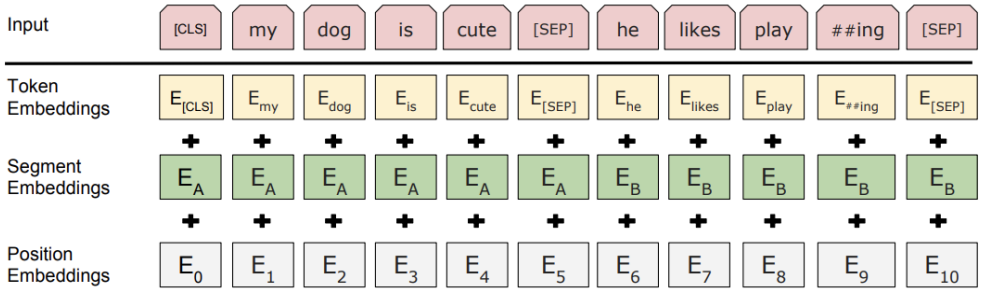
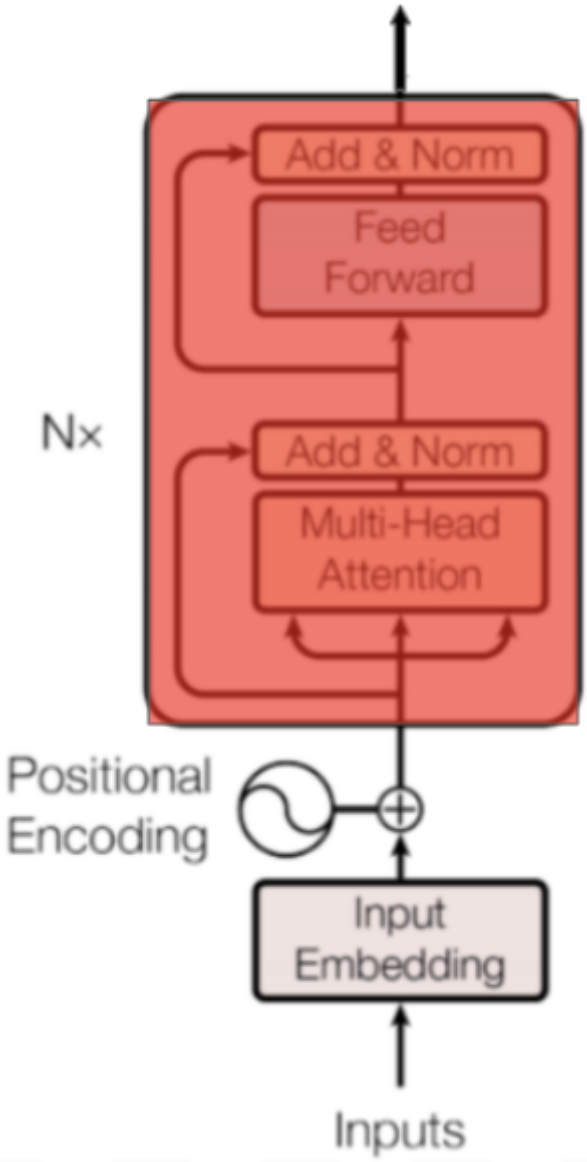


Figure 2: BERT input representation. The input embeddings is the sum of the token embeddings, the segmentation embeddings and the position embeddings.

1. Token Embeddings : 그냥 토큰 임베딩, 기존의 word2vec 같은 느낌
2. Segment Embeddings : 문서에 문장이 여러개 들어갈 경우, 한 문서는 [CLS]로, 각 문서 내 문장들은

   [SEP]라는 토큰으로 구분한다, [CLS]는 나중에 Q/A나 NLI에 쓰인다
3. Position Embeddings : 해당 단어들의 위치만을 임베딩

Encoder Block

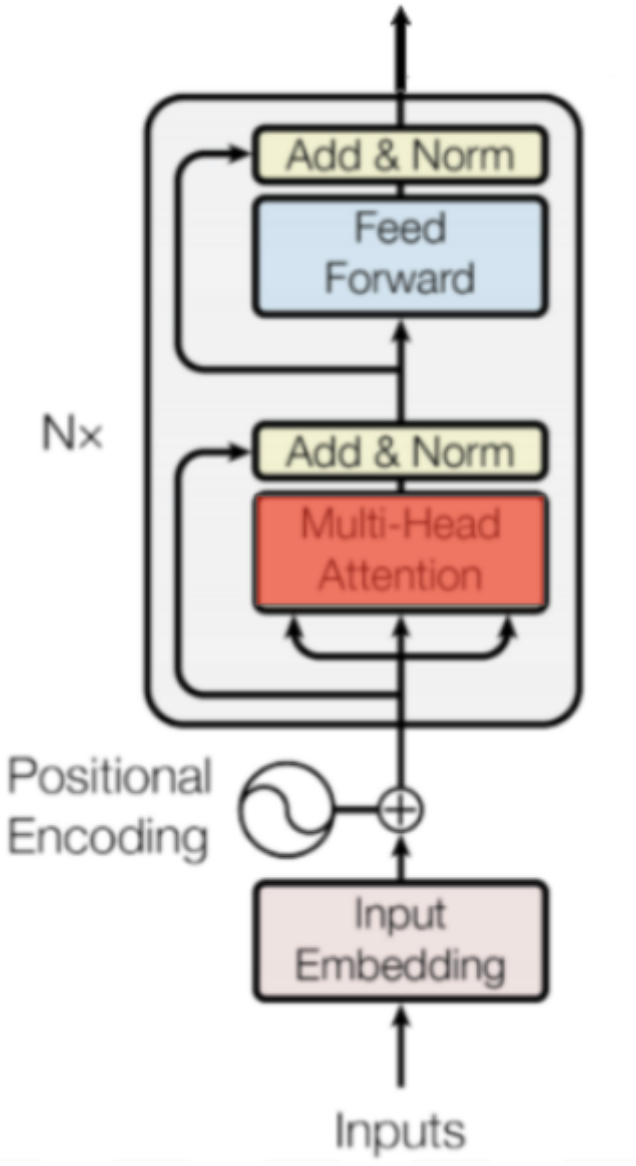


논문 Base모델은 12개, Large모델은 24개로, 시퀀스 전체의 의미를 N(12,24)반복적으로 구축

-> 블록의 수가 많을수록 단어 사이에 복잡한 관계를 잘 포착할 수 있다

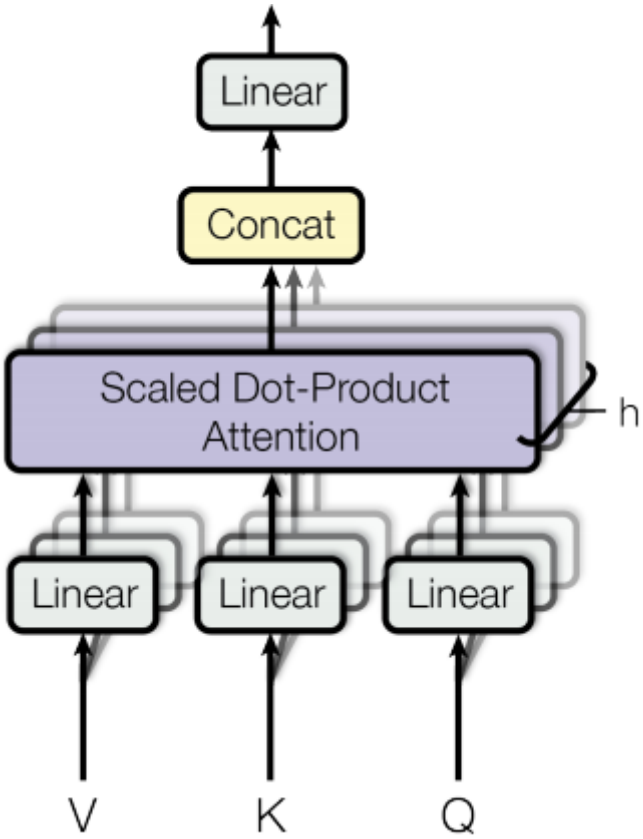
인코더 블록은 병렬처리가 아닌, 이전 출력값을 현재의 입력값으로 하는 RNN과 유사한 특징을 갖고 있음

Multi-Head Attention



헤드가 여러개인 어텐션, 서로 다른 가중치 행렬을 이용해 각각 어텐션을 계산, 그리고 서로 연결한다

Scaled Dot-Product Attention



처음에는 임베딩의 fully-connected 결과, 이전 인코더 블록의 결과를 다음 인코더 블럭의 입력으로 사용한다

- Transformer는

Q : 디코더의 히든 스테이트

K : 인코더의 히든 스테이트

V : K에 어텐션을 부여 받은 Normalized Weights

- Bert에서는 Q,K,V의 초기값이 모두 동일하며, 디코더를 쓰지도 않음(구성만 동일함)

=> Scaled_Dot_Product Attention을 여러번 계산한 결과들의 Concatenate임

Masked Attention(Optional)

제로 패딩으로 입력된 토큰은, 마스킹 처리 및 패널티를 부과해 어텐션 점수를 못 받게 함

Feed-Forward Network

마지막 어텐션 결과를 FFN을 통과시킴

두 개의 Linear Transformations, Bert는 중간에 Gelu를 써서 음수를 소실시키지 않음

(잘 이해가 안감)

가장 중요 : 학 습

가장 큰 차이점이 뭐냐 ? -> Bidirectional 하다!

어떻게 구현했냐 ?

```
1. Masked Language Model
```

2. Next Sentence Prediction

[SEP] : 문장의 끝을 나타냄 (구분함) -> 이를 통해 QA 문제 해결과 Next_Sentence_Prediction

[CLS] : 분류 문제를 해결하는데 사용

pre-training에선

- BooksCorpus(800M words), English Wikipedia(2,500M words)
    - 특이 위키피디아에선 텍스트만 가져옴(긴 문장을 뽑기 위함)

Masked Language Model

(Word2Vec CBOW와 유사)

- 전체 단어의 15%를 마스킹해서, 이 토큰이 뭔지 맞추도록 학습한 결과를 벡터로 가질 수 있음
- 하지만 전체 단어의 15%를 마스킹하고, 그 중 80%만 가 되고, 10%는 랜덤단어,

  나머지 10%는 정상적인 단어

(이유) : 밖에 없으면 Fine-tuning시 이 토큰이 없어서 아무것도 예측 할 필요가 없다고 생각하기에

  토큰이 아닌것도 예측하도록 학습해서 문맥 표현이 학습되도록 한다

Next Sentence Prediction

두 문장을 주고, 두 번째 문장이 코퍼스 내에서 첫 문장의 바로 다음에 오는지 여부를 예측

QA와 Natural Language Inference(자연어 추론)을 하기에 MLM으론 부족

두 문장이 실제로 이어지는지 여부는 50% 비율로 참 / 랜덤 추출 거짓 문장의 비율로 구성

[CLS] 벡터로 Binary Classification을 통해 맞추도록 학습시킴