

# BERT

1. NLP에서 사전 훈련(Pre-training)
2. BERT(Bidirectional Encoder Representations from Transformers)

김지현

## NLP에서 사전 훈련(Pre-training)

### 1. 사전 훈련된 워드 임베딩

어떤 태스크를 수행할 때, 워드 임베딩(Word2Vec, FastText, GloVe 등)을 사용하는 방법

- 1) 임베딩 층(Embedding layer)을 랜덤 초기화하여 처음부터 학습하는 방법
- 2) 방대한 데이터로 사전에 학습된 임베딩 벡터들을 가져와 사용하는 방법
  - 데이터가 적을 때, 사전 훈련된 임베딩을 사용하면 성능 향상을 기대할 수 있음.

=> 문제점: 하나의 단어가 하나의 벡터로 맵핑 -> 다의어나 동음이의어 구분하지 못함.

ex) '사과'에 맵핑된 벡터값은 '용서를 빈다'는 의미와 '먹는 과일'의 의미를 구분할 수 없음.

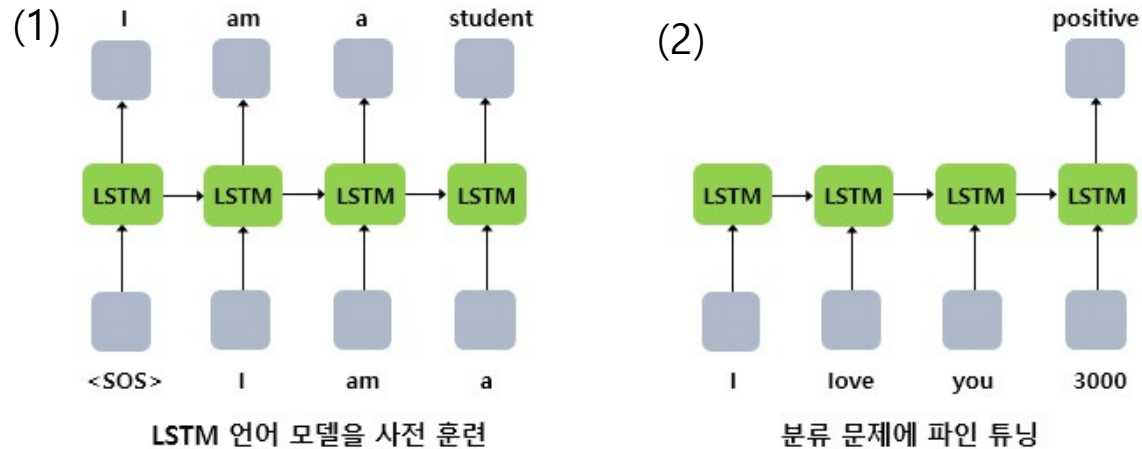
=> 이 한계는 사전 훈련된 언어 모델로 극복할 수 있음(ex. ELMo나 BERT 등)

## NLP에서 사전 훈련(Pre-training)

### 2. 사전 훈련된 언어모델

- 주어진 텍스트로부터 이전 단어들로부터 다음 단어를 예측하도록 언어모델을 학습
  - (장점) 기본적으로 별도의 레이블이 부착되지 않은 텍스트 데이터로도 학습 가능
- 문맥에 따라서 임베딩 벡터값이 달라지므로, 다의어를 구분할 수 없었던 문제점을 해결할 수 있음.

Semi-Supervised Sequence Learning, Google, 2015



2015년 구글- 'Semi-supervised Sequence Learning' 논문

- (1) LSTM 언어 모델을 학습
- (2) 학습한 LSTM을 텍스트 분류에 추가 학습

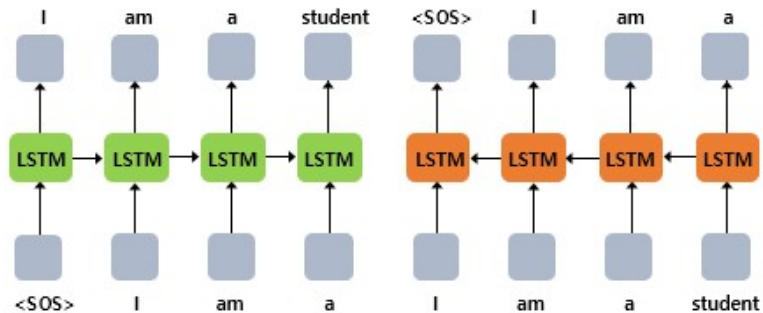
레이블이 없는 데이터로 학습된 LSTM VS 가중치가 랜덤으로 초기화된 LSTM  
텍스트 분류 문제와 같은 문제를 학습하여 전자의 경우가 더 좋은 성능을 얻음.

## NLP에서 사전 훈련(Pre-training)

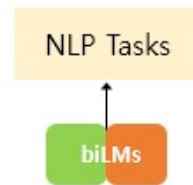
### 2. 사전 훈련된 언어모델

- 주어진 텍스트로부터 이전 단어들로부터 다음 단어를 예측하도록 언어모델을 학습
  - (장점) 기본적으로 별도의 레이블이 부착되지 않은 텍스트 데이터로도 학습 가능
- 문맥에 따라서 임베딩 벡터값이 달라지므로, 다의어를 구분할 수 없었던 문제점을 해결할 수 있음.

#### ELMo: Deep Contextual Word Embedding, AI2 & University of Washington, 2017



순방향 언어 모델과 역방향 언어 모델을 각각 훈련



사전 훈련된 임베딩에 사용

#### ELMo

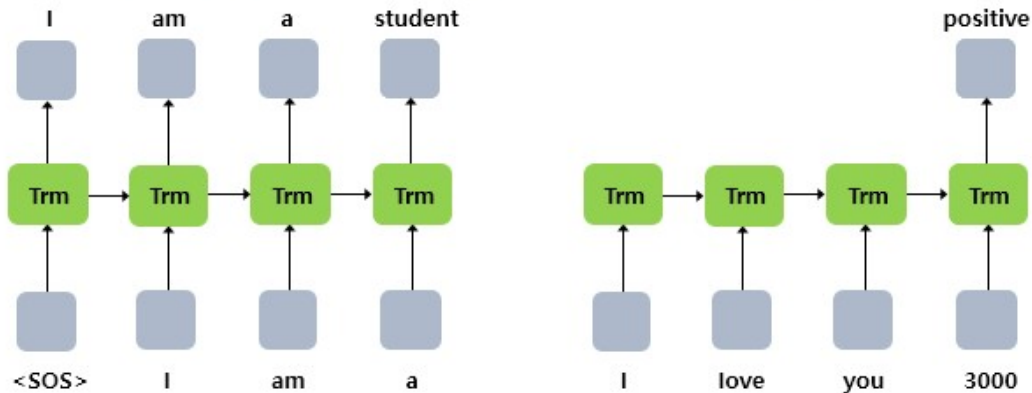
- (1) 순방향과 역방향 언어 모델 각각 학습
- (2) 사전 훈련된 임베딩에 사용

## NLP에서 사전 훈련(Pre-training)

### 2. 사전 훈련된 언어모델

- RNN 계열의 신경망에서 탈피
- 트랜스포머가 번역기와 같은 인코더-디코더 구조에서 LSTM을 뛰어넘는 좋은 성능을 얻자, LSTM이 아닌 트랜스포머로 사전 훈련된 언어 모델을 학습하는 시도가 등장

#### Improving Language Understanding by Generative Pre-training, OpenAI, 2018



Deep(12-layer) Trm 언어 모델을 사전 훈련

분류 문제에 파인 튜닝

#### GPT-1

- Open AI는 트랜스포머 디코더로 총 12개의 층을 쌓은 후에 방대한 텍스트 데이터를 학습시킨 언어 모델을 만듦.
- 다양한 태스크에서 높은 성능을 얻을 수 있음을 입증함.

## NLP에서 사전 훈련(Pre-training)

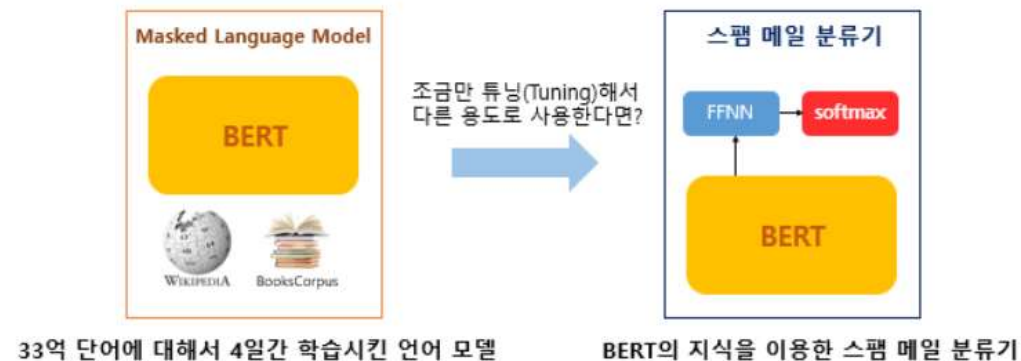
### 3. 마스크드 언어 모델(Masked Language Model)

- 언어의 문맥이라는 것은 실제로는 양방향
- 이전 단어로부터 다음 단어를 예측하는 언어 모델의 특성으로 인해 양방향 언어 모델을 사용할 수 없음.  
(ELMo에서는 순방향과 역방향이라는 두 개의 단방향 언어 모델을 따로 준비하여 학습하는 방법을 사용)
- 양방향 구조를 도입하기 위해서 2018년에는 “마스크드 언어 모델”이라는 새로운 구조의 언어 모델이 탄생
- 입력 텍스트의 단어 집합의 15%의 단어를 랜덤으로 마스킹(Masking)
- 마스킹 된 단어들을(Masked words) 예측하도록 함.

# BERT(Bidirectional Encoder Representations from Transformers, BERT)

## 1. BERT 개요

- 2018년에 구글이 공개한 사전 훈련된 모델
- 이전 챕터에서 배웠던 트랜스포머를 이용하여 구현되었으며, 위키피디아(25억 단어)와 BooksCorpus(8억 단어)와 같은 레이블이 없는 텍스트 데이터로 사전 훈련된 언어 모델
- 레이블이 없는 방대한 데이터로 사전 훈련된 모델을 가지고, 레이블이 있는 다른 작업(Task)에서 추가 훈련과 함께 하이퍼파라미터를 재조정(Fine-tuning)
- 수많은 NLP 태스크에서 최고 성능을 보여주며 NLP의 한 획을 그은 모델로 평가받음.



## BERT(Bidirectional Encoder Representations from Transformers, BERT)

### 2. BERT의 크기

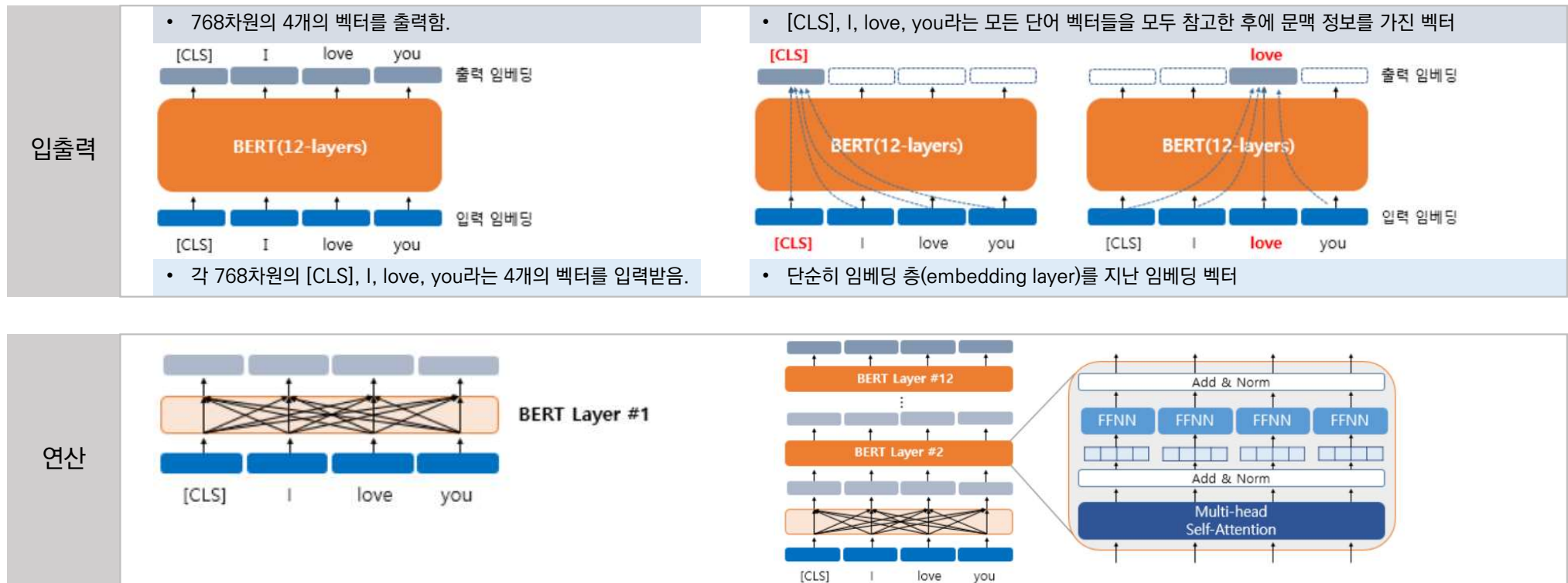
- 기본 구조는 트랜스포머의 인코더를 쌓아올린 구조
- Base 버전에서는 총 12개를 쌓고 Large 버전에서는 총 24개를 쌓았음.
- Large 버전은 Base 버전보다 d\_model의 크기나 셀프 어텐션 헤드(Self Attention Heads)의 수가 더 큼.
- 트랜스포머 인코더 층의 수를 L, d\_model의 크기를 D, 셀프 어텐션 헤드의 수를 A라고 할 때,
  - BERT-Base : L=12, D=768, A=12 : 110M개의 파라미터 (GPT-1과 성능을 비교하기 위해서 GPT-1과 동등한 크기로 BERT-Base를 설계)
  - BERT-Large : L=24, D=1024, A=16 : 340M개의 파라미터 (BERT의 최대 성능을 보여주기 위함)
- cf) 초기 트랜스포머 모델(<https://wikidocs.net/31379>) 파라미터 정보: L=6, D=512, A=8



## BERT(Bidirectional Encoder Representations from Transformers, BERT)

### 3. BERT의 문맥을 반영한 임베딩(Contextual Embedding)

- ELMo나 GPT-1과 마찬가지로 문맥을 반영한 임베딩을 사용함.
- [입력] 각 단어들은 임베딩 층(Embedding layer)를 지나 768차원의 임베딩 벡터가 되어 입력으로 사용됨.
- [출력] 내부적인 연산을 거친 후, 동일하게 각 단어에 대해서 768차원의 벡터를 출력
- BERT의 연산을 거친 후의 출력 임베딩은 문장의 문맥을 모두 참고한 문맥을 반영한 임베딩이 됨.



## BERT(Bidirectional Encoder Representations from Transformers, BERT)

### 4. BERT의 서브워드 토크나이저 : WordPiece

- BERT는 단어보다 더 작은 단위로 쪼개는 서브워드 토크나이저를 사용
- WordPiece 토크나이저는 글자로부터 서브워드들을 병합해가는 방식으로 최종 단어 집합(Vocabulary)을 만드는 것은 바이트 페어 인코딩(Byte Pair Encoding, BPE)와 유사
- 서브워드 토크나이저는 기본적으로 자주 등장하는 단어는 그대로 단어 집합에 추가하지만, 자주 등장하지 않는 단어의 경우에는 더 작은 단위인 서브워드로 분리되어 서브워드들이 단어 집합에 추가됨.
- 이렇게 단어 집합이 만들어지고 나면, 이 단어 집합을 기반으로 토큰화를 수행 (BERT의 단어 집합의 크기는 30,522개임.)

#### 〈 BERT에서 토큰화를 수행하는 방식 〉

준비물 : 이미 훈련 데이터로부터 만들어진 단어 집합

1. 토큰이 단어 집합에 존재한다.

=> 해당 토큰을 분리하지 않는다.

2. 토큰이 단어 집합에 존재하지 않는다.

=> 해당 토큰을 서브워드로 분리한다.

=> 해당 토큰의 첫번째 서브워드를 제외한 나머지 서브워드들은 앞에 "##"를 붙인 것을 토큰으로 한다.

## BERT(Bidirectional Encoder Representations from Transformers, BERT)

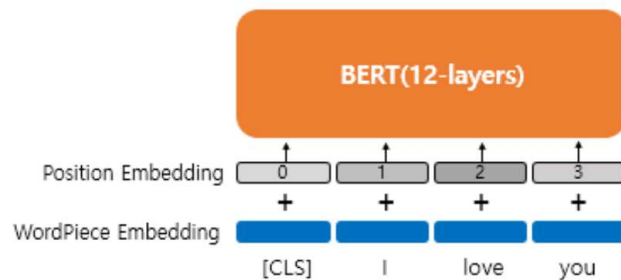
### 5. 포지션 임베딩(Position Embedding)

- 학습 가능한 임베딩 층을 구성해 위치 임베딩 벡터를 만드는 방법

Q. 트랜스포머에서의 포지셔널 인코딩(Positional Encoding)\*과 다른 점은?

A. 둘 다 단어의 위치 정보를 표현하기 위한 위치 임베딩 벡터를 만드는 방법인데

Positional Encoding은 사인함수와 코사인함수를 이용해 임베딩 벡터를 만들고 Position Embedding은 학습을 통해서 만듦.

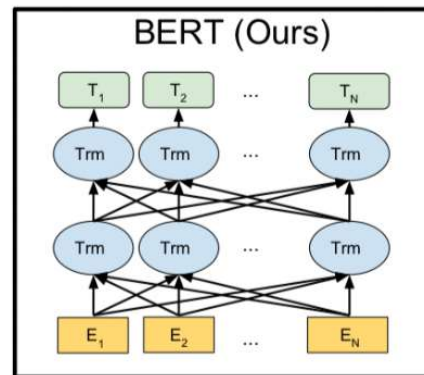


- 4개의 포지션 임베딩 벡터를 학습시킴.
- BERT의 입력마다 다음과 같이 포지션 임베딩 벡터를 더해줌.
  - 첫번째 단어의 임베딩 벡터 + 0번 포지션 임베딩 벡터
  - 두번째 단어의 임베딩 벡터 + 1번 포지션 임베딩 벡터
  - 세번째 단어의 임베딩 벡터 + 2번 포지션 임베딩 벡터
  - 네번째 단어의 임베딩 벡터 + 3번 포지션 임베딩 벡터

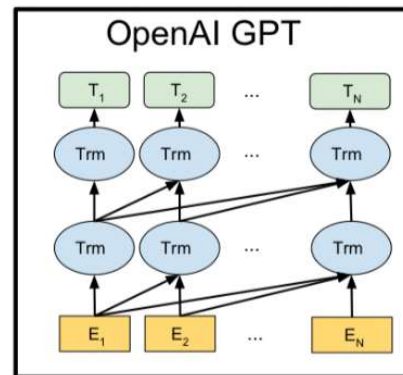
- 실제 BERT에서는 문장의 최대 길이를 512로 하고 있으므로, 총 512개의 포지션 임베딩 벡터 학습
- BERT에서는 총 두 개의 임베딩 층이 사용
  - 단어 집합의 크기가 30,522개인 단어 벡터를 위한 임베딩 층
  - 512개의 포지션 벡터를 위한 임베딩 층

## BERT(Bidirectional Encoder Representations from Transformers, BERT)

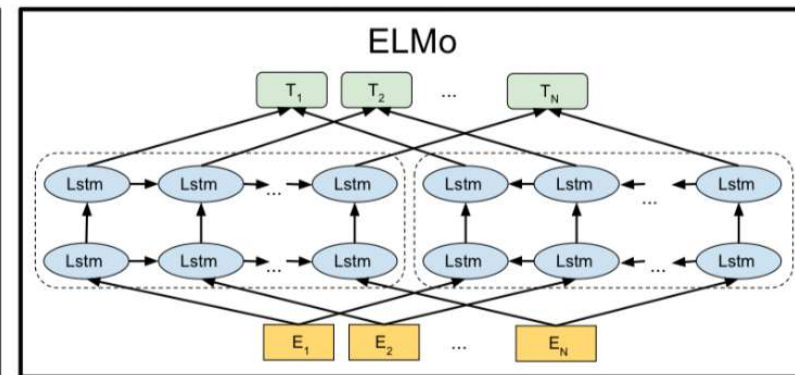
### 6. BERT의 사전 훈련(Pre-training)



- 양방향 언어 모델
- 마스크드 언어 모델(Masked Language Model)를 통해 양방향 향성을 얻음



- 단방향 언어 모델
- 트랜스포머의 디코더를 이용하여 이전 단어들로부터 다음 단어를 예측



- 양방향 언어 모델
- 정방향 LSTM과 역방향 LSTM을 각각 훈련시킴

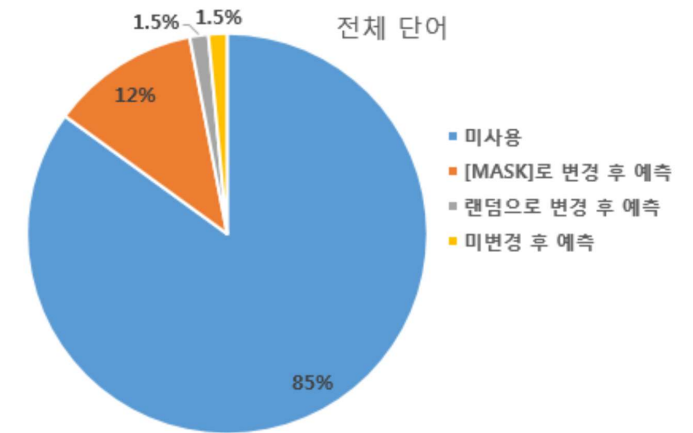
- BERT는 BookCorpus(8억 단어)와 위키피디아(25억 단어)로 학습
- BERT의 사전 훈련 방법
  - 마스크드 언어 모델(Masked Language Model)
  - 다음 문장 예측(Next sentence prediction, NSP)

## BERT(Bidirectional Encoder Representations from Transformers, BERT)

### 6. BERT의 사전 훈련(Pre-training)

#### 1) 마스크드 언어 모델(Masked Language Model, MLM)

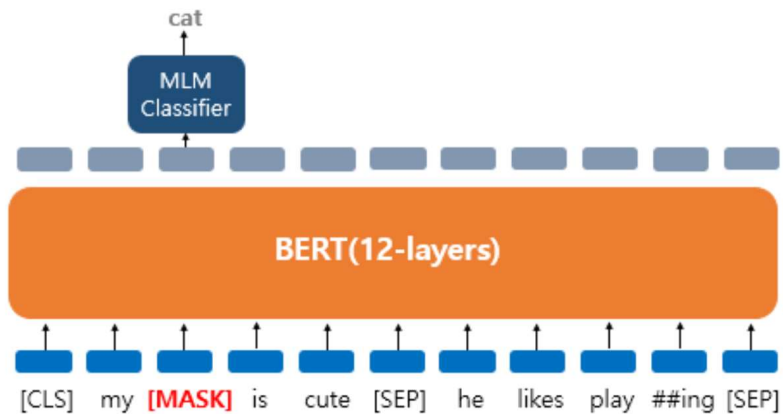
- 인공 신경망의 입력으로 들어가는 입력 텍스트의 15%의 단어를 랜덤으로 마스킹(Masking)
- 인공 신경망에게 이 가려진 단어들을(Masked words) 예측하도록 함.
- 랜덤으로 선택된 15%의 단어들은 다시 다음과 같은 비율로 규칙이 적용
  - 80%의 단어들은 [MASK]로 변경  
Ex) The man went to the store → The man went to the [MASK]
  - 10%의 단어들은 랜덤으로 단어가 변경  
Ex) The man went to the store → The man went to the dog
  - 10%의 단어들은 동일하게 둔다.  
Ex) The man went to the store → The man went to the store
- [MASK] 토큰이 파인 튜닝 단계에서는 나타나지 않으므로 사전 학습 단계와 파인 튜닝 단계에서의 불일치가 발생하는 문제가 있음.  
이 문제를 완화하기 위해서 랜덤으로 선택된 15%의 단어들의 모든 토큰을 [MASK]로 사용하지 않음.



## BERT(Bidirectional Encoder Representations from Transformers, BERT)

### 6. BERT의 사전 훈련(Pre-training)

#### 1) 마스크드 언어 모델(Masked Language Model, MLM)



문장 : 'My dog is cute. he likes playing'

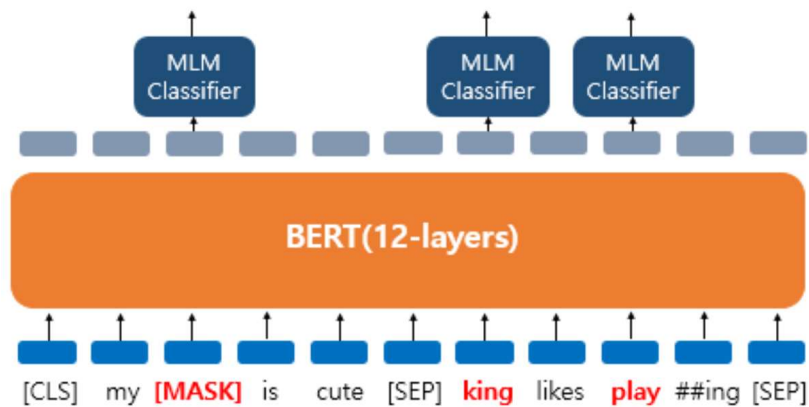
- 'dog' 토큰이 [MASK]로 변경

- BERT 모델이 [MASK]로 변경된 토큰을 맞추려고 함.
- 여기서 출력층에 있는 다른 위치의 벡터들은 예측과 학습에 사용되지 않고, 오직 'dog' 위치의 출력층의 벡터만이 사용됨.

## BERT(Bidirectional Encoder Representations from Transformers, BERT)

### 6. BERT의 사전 훈련(Pre-training)

#### 1) 마스크드 언어 모델(Masked Language Model, MLM)



문장 : 'My dog is cute. he likes playing'

- 'dog' 토큰이 [MASK]로 변경
- 'he'는 랜덤 단어 'king'으로 변경
- 'play'는 변경되진 않았지만 예측에 사용됨.

BERT는

- [MASK]로 변경된 토큰에 대해 원래 단어가 무엇인지
- 랜덤 단어 'king'으로 변경된 토큰에 대해서도 원래 단어가 무엇인지
- 'play'는 변경되지 않았지만 BERT 입장에서는 이것이 변경된 단어인지 아닌지 모르므로 마찬가지로 원래 단어를 예측

## BERT(Bidirectional Encoder Representations from Transformers, BERT)

### 6. BERT의 사전 훈련(Pre-training)

#### 2) 다음 문장 예측(Next Sentence Prediction, NSP)

- 두 개의 문장을 준 후에 이 문장이 이어지는 문장인지 아닌지를 맞추는 방식으로 훈련시킴.
- 50:50 비율로 실제 이어지는 두 개의 문장과 랜덤으로 이어붙인 두 개의 문장을 주고 훈련

- 이어지는 문장의 경우

Sentence A : The man went to the store.

Sentence B : He bought a gallon of milk.

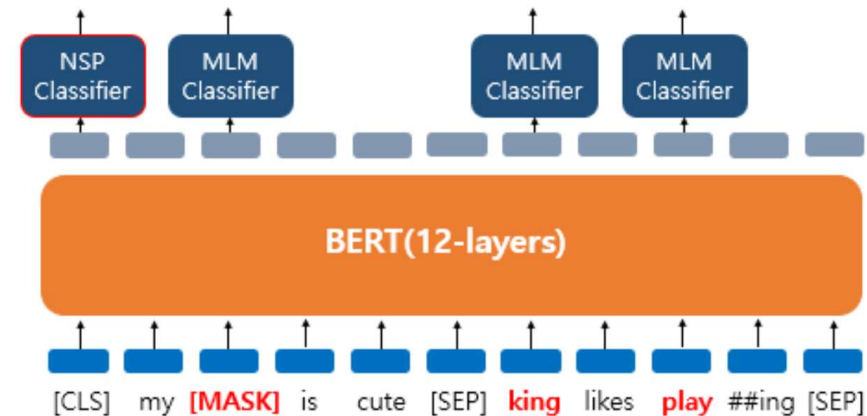
Label = IsNextSentence

- 이어지는 문장이 아닌 경우

Sentence A : The man went to the store.

Sentence B : dogs are so cute.

Label = NotNextSentence



- [SEP] : 문장의 끝을 식별하는 특별 토큰
- [CLS] : 두 문장이 실제 이어지는 문장인지 아닌지 풀도록 하는 특별 토큰

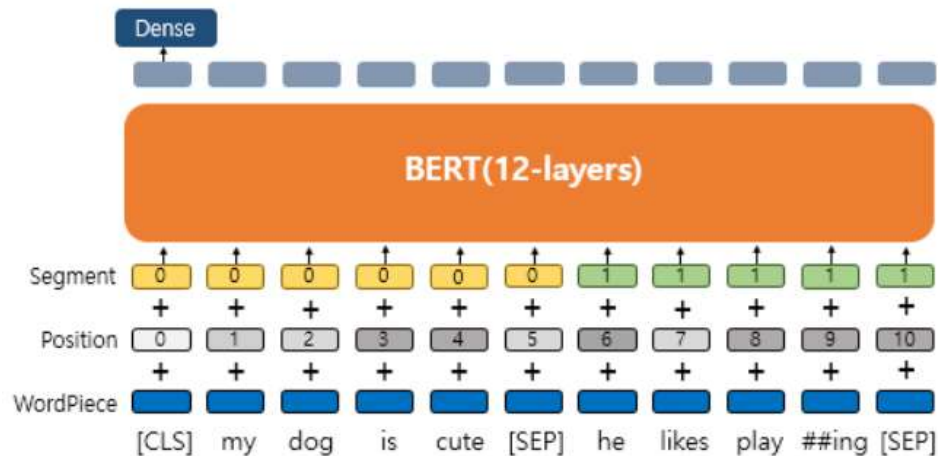
- 마스크드 언어 모델과 다음 문장 예측은 따로 학습하는 것이 아닌 loss를 합하여 학습이 동시에 이루어짐.



## BERT(Bidirectional Encoder Representations from Transformers, BERT)

### 7. 세그먼트 임베딩(Segment Embedding)

- BERT는 QA(Question Answering) 등 두 개의 문장 입력이 필요한 태스크를 풀기도 함.
- 문장 구분을 위해 세그먼트 임베딩이라는 임베딩 층(Embedding layer)를 사용
- 첫번째 문장에는 Sentence 0 임베딩, 두번째 문장에는 Sentence 1 임베딩을 더해주는 방식
- 감성 분류 태스크 등 한 개의 문서에 대해서만 분류를 하는 경우는 BERT의 전체 입력에 Sentence 0 임베딩만을 더해줌.



- [SEP]와 세그먼트 임베딩으로 구분되는 BERT의 입력에서의 두 개의 문장은 실제로는 두 종류의 텍스트, 두 개의 문서일 수 있음

BERT에서 사용되는 3개의 임베딩 층

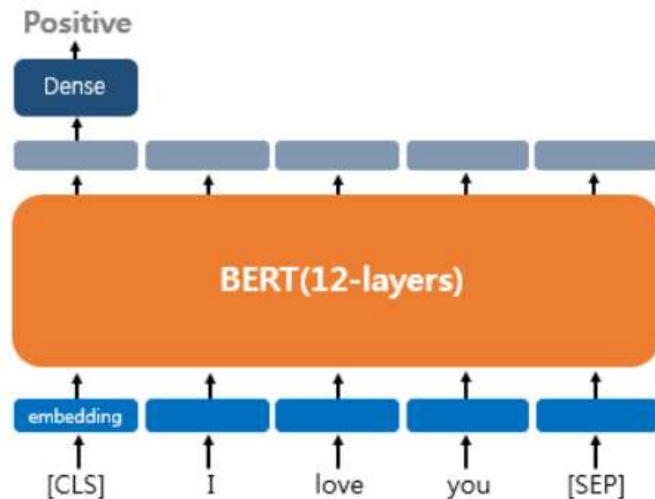
- WordPiece Embedding : 실질적인 입력이 되는 워드 임베딩. 임베딩 벡터의 종류는 단어 집합의 크기로 30,522개.
- Position Embedding : 위치 정보를 학습하기 위한 임베딩. 임베딩 벡터의 종류는 문장의 최대 길이인 512개.
- Segment Embedding : 두 개의 문장을 구분하기 위한 임베딩. 임베딩 벡터의 종류는 문장의 최대 개수인 2개.

## BERT(Bidirectional Encoder Representations from Transformers, BERT)

### 8. BERT를 파인 튜닝(Fine-tuning)하기

- 사전 학습 된 BERT에 우리가 풀고자 하는 태스크의 데이터를 추가로 학습 시켜서 테스트하는 단계

#### 1) 하나의 텍스트에 대한 텍스트 분류 유형(Single Text Classification)



- [SEP]와 세그먼트 임베딩으로 구분되는 BERT의 입력에서의 두 개의 문장은 실제로는 두 종류의 텍스트, 두 개의 문서일 수 있음

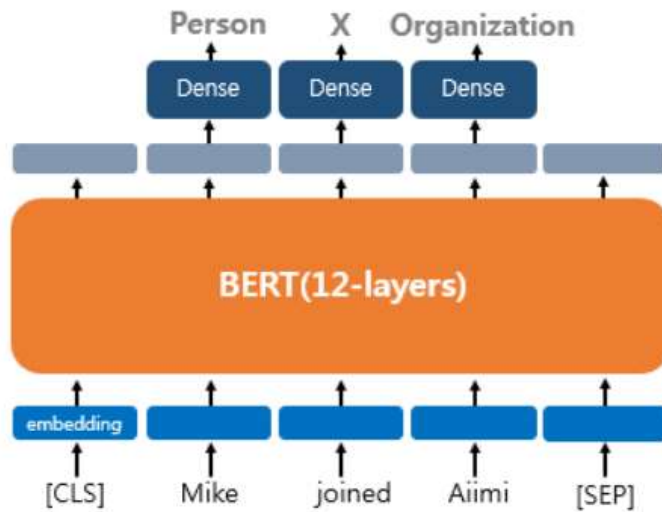
- 영화 리뷰 감성 분류, 로이터 뉴스 분류 등과 같이 입력된 문서에 대해서 분류를 하는 유형
- 문서의 시작에 [CLS] 라는 토큰을 입력
- 텍스트 분류 문제를 풀기 위해서 [CLS] 토큰의 위치의 출력층에서 밀집층(fully-connected layer)을 추가하여 분류에 대한 예측을 함.

## BERT(Bidirectional Encoder Representations from Transformers, BERT)

### 8. BERT를 파인 튜닝(Fine-tuning)하기

- 사전 학습 된 BERT에 우리가 풀고자 하는 태스크의 데이터를 추가로 학습 시켜서 테스트하는 단계

#### 2) 하나의 텍스트에 대한 태깅 작업(Tagging)



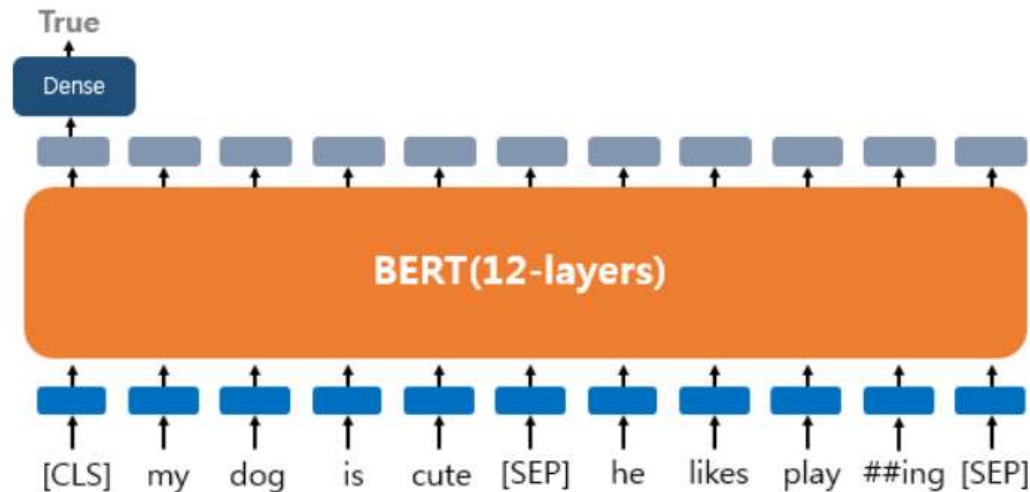
- 대표적으로 문장의 각 단어에 품사를 태깅하는 품사 태깅 작업과 개체를 태깅하는 개체명 인식 작업이 있음.
- 출력층에서는 입력 텍스트의 각 토큰의 위치에 밀집층을 사용하여 분류에 대한 예측을 함.

## BERT(Bidirectional Encoder Representations from Transformers, BERT)

### 8. BERT를 파인 튜닝(Fine-tuning)하기

- 사전 학습 된 BERT에 우리가 풀고자 하는 태스크의 데이터를 추가로 학습 시켜서 테스트하는 단계

#### 3) 텍스트의 쌍에 대한 분류 또는 회귀 문제(Text Pair Classification or Regression)



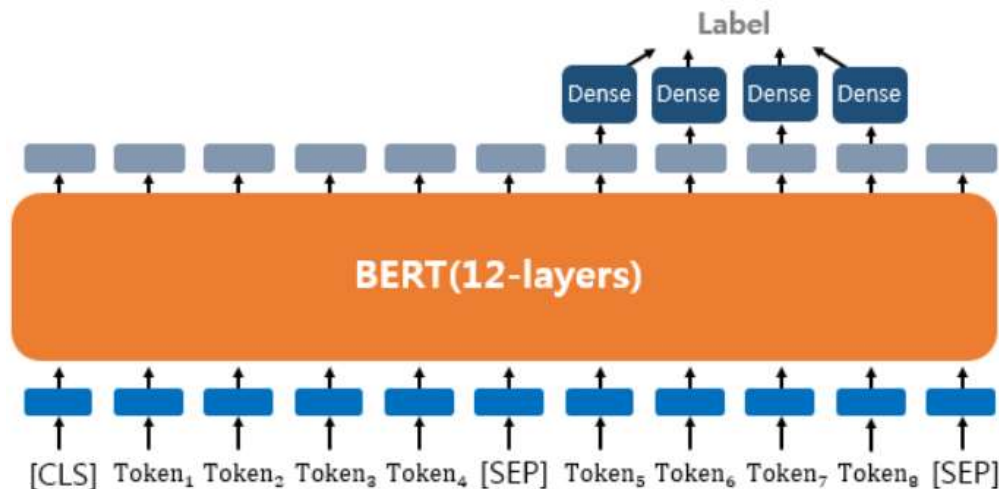
- 대표적인 태스크로 자연어 추론\*(Natural language inference)이 있음.
  - \* 자연어 추론 문제란, 두 문장이 주어졌을 때, 하나의 문장이 다른 문장과 논리적으로 어떤 관계에 있는지를 분류하는 것  
유형으로 모순 관계(contradiction), 함의 관계(entailment), 중립 관계(neutral)이 있음.
- 텍스트 사이에 [SEP] 토큰을 집어넣고, Sentence 0 임베딩과 Sentence 1 임베딩이라는 두 종류의 세그먼트 임베딩을 모두 사용하여 문서를 구분

## BERT(Bidirectional Encoder Representations from Transformers, BERT)

### 8. BERT를 파인 튜닝(Fine-tuning)하기

- 사전 학습 된 BERT에 우리가 풀고자 하는 태스크의 데이터를 추가로 학습 시켜서 테스트하는 단계

#### 4) 질의 응답(Question Answering)



- 대표적인 태스크로 QA(Question Answering)이 있으며 BERT로 QA를 풀기 위해서 질문과 본문이라는 두 개의 텍스트의 쌍을 입력함.
- 대표적인 데이터셋으로 SQuAD(Stanford Question Answering Dataset) v1.1이 있음.  
질문과 본문을 입력받으면, 본문의 일부분을 추출해서 질문에 답변함.
  - 질문 : "강우가 떨어지도록 영향을 주는 것은?"
  - 본문 : "기상학에서 강우는 대기 수증기가 응결되어 중력의 영향을 받고 떨어지는 것을 의미합니다. 강우의 주요 형태는 이슬비, 비, 진눈깨비, 눈, 싸락눈 및 우박이 있습니다."
  - 답 : "중력"

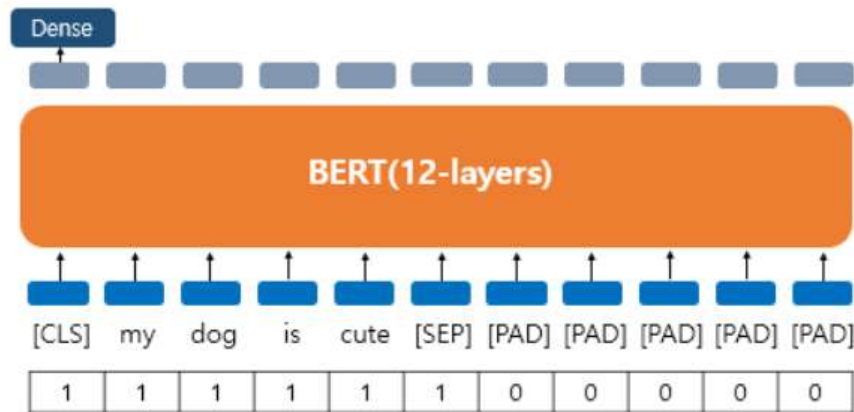
## BERT(Bidirectional Encoder Representations from Transformers, BERT)

### 9. 그 외 기타

- 훈련 데이터는 위키피디아(25억 단어)와 BooksCorpus(8억 단어)  $\approx$  33억 단어
- WordPiece 토큰라이저로 토큰화를 수행 후 15% 비율에 대해서 마스크드 언어 모델 학습
- 두 문장 Sentence A와 B의 합한 길이. 즉, 최대 입력의 길이는 512로 제한
- 100만 step 훈련  $\approx$  (총 합 33억 단어 코퍼스에 대해 40 에포크 학습)
- 옵티마이저 : 아담(Adam)
- 학습률(learning rate) :
- 가중치 감소(Weight Decay) : L2 정규화로 0.01 적용
- 드롭 아웃 : 모든 레이어에 대해서 0.1 적용
- 활성화 함수 : relu 함수가 아닌 gelu 함수
- 배치 크기(Batch size) : 256

## BERT(Bidirectional Encoder Representations from Transformers, BERT)

### 10. 어텐션 마스크(Attention Mask)



- 어텐션 마스크는 BERT가 어텐션 연산을 할 때, 불필요하게 패딩 토큰에 대해서 어텐션을 하지 않도록 실제 단어와 패딩 토큰을 구분할 수 있도록 알려주는 입력  
숫자 1: 해당 토큰은 실제 단어이므로 마스킹을 하지 않음  
숫자 0: 해당 토큰은 패딩 토큰이므로 마스킹을 함.