

1조(BERT)

▼ BERT: 양방향 인코더 트랜스포머를 통한 표현

BERT는 구글에서 개발하고 2018년에 출시한 자연어 처리(NLP) 모델입니다. 이는 자연어와 같은 순차적 데이터를 처리하기 위해 설계된 트랜스포머 아키텍처를 기반으로 합니다.

BERT는 다른 NLP 모델과 달리 양방향 모델이기 때문에 단어의 문맥을 왼쪽뿐만 아니라 오른쪽에서도 고려할 수 있습니다. 이를 통해 문장 내 단어 간의 관계를 더 잘 이해할 수 있습니다.

BERT는 대규모 텍스트 데이터에서 사전 학습되어 언어와 문맥의 미묘한 차이를 학습할 수 있습니다. 그런 다음 작업별로 세부 조정이 가능하며, 이를 위해 작업별 특화 데이터의 작은 양만 필요합니다. 이 작업에는 질문 답변이나 감정 분석 등이 포함됩니다.

BERT의 주요 장점 중 하나는 더 깊은 언어 이해가 필요한 작업을 처리할 수 있다는 점입니다. 예를 들어 문맥에 따라 "은행"이 금융 기관으로 쓰이는지, 강둑으로 쓰이는지 구분 가능합니다.

BERT는 챗봇, 검색 엔진, 언어 번역 등 다양한 분야에 활용되었으며, 많은 NLP 벤치마크에서 최고 성과를 기록하고 있어 연구원 및 전문가들 사이에서 인기가 높습니다.

총괄적으로, BERT는 NLP 분야에서 중요한 발전을 나타내며, 지속적인 개발로 인해 더욱 놀라운 결과가 나올 것으로 기대됩니다.

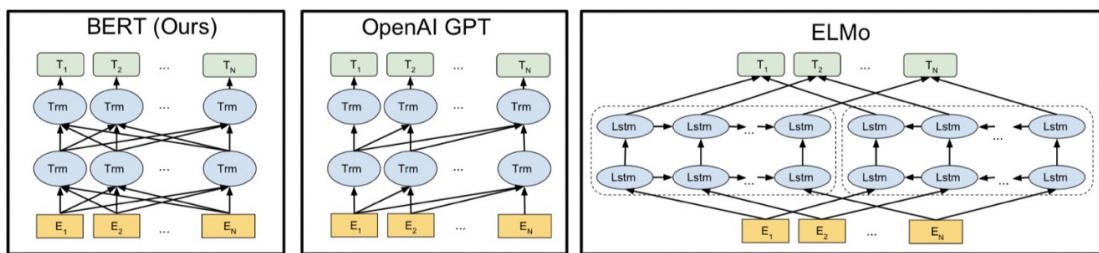
본문에 대한 질문을 넣으면 그에 맞는 답을 얻을 수 있음.

딥러닝 기반 자연어 언어모델 BERT

- 기본적으로, WIKI나 book data와 같은 대용량 unlabeled data로 모델을 미리 학습시킨 후, 특정 task를 가지고 있는 labeled data로 transfer learning을 하는 모델
- QA를 풀기 위해서 질문과 본문이라는 두 개의 텍스트의 쌍을 입력함. 대표적인 데이터셋으로 SQuAD(Stanford Question Answering Dataset)이 있음. 질문과 본문을 받으면, 본문의 일부분을 추출해 질문에 답변하는 것

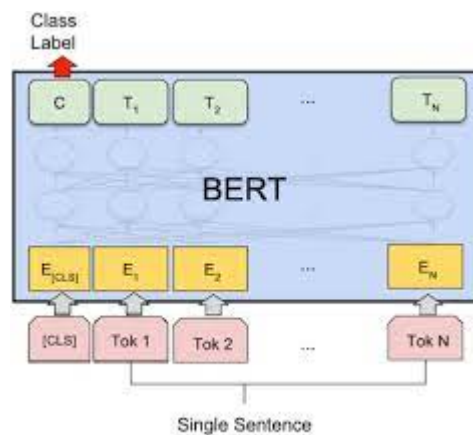
- 대용량 unlabeled data를 통해 language model을 학습하고, 이를 토대로 뒤쪽에 특정 task를 처리하는 network를 붙이는 방식 ex) ELMo, OpenAI GPT...
- 이전의 모델의 접근 방식은 shallow bidirectional(얕은 양방향성) 또는 unidirectional(한방향으로만) 하다고 봄.
- bert는 특정 task를 처리하기 위해 새로운 network를 붙일 필요 없이, bert 모델 자체의 fine-tuning을 진행함.
- bert의 기본 구조는 트랜스포머의 인코더를 쌓아올린 구조

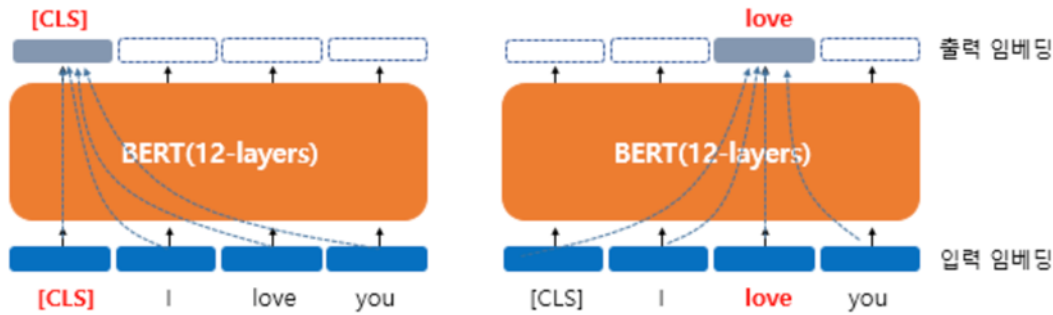
BERT의 pre-training 방법론



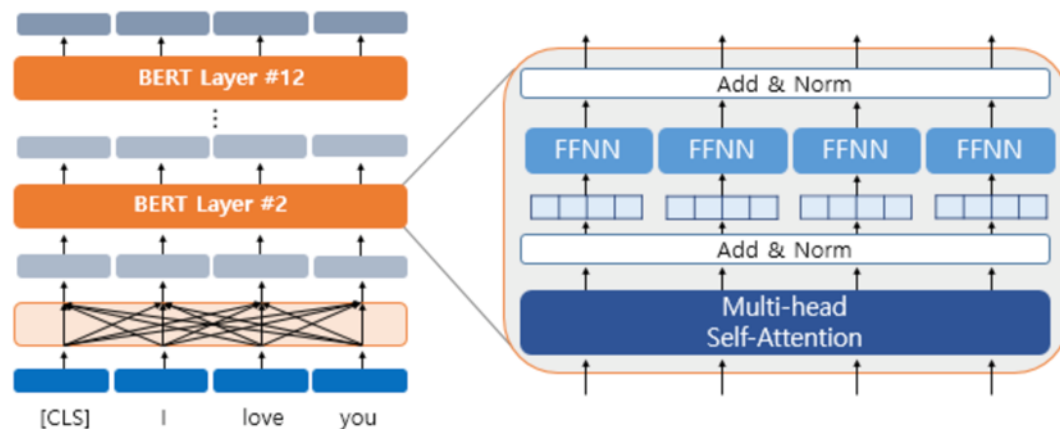
▼ BERT 모델 구조도

bert는 양방향으로 화살표가 뻗어나감, 예측할 단어의 좌우 문맥을 고려하여 예측함.





왼쪽 그림 [cls] 벡터는 버트의 초기 입력으로 입력 임베딩 당시 임베딩 층을 지난 임베딩 벡터였지만, 버트를 지나고 나서는 [cls],I,love,you 모든 단어를 반영한 문맥 정보를 가진 벡터가 됩니다.



Pre-training

MLM과 NSP를 위해 self-attention을 수행하는 transformer encoder구조를 사용했음.

1. MLM(Masked Language Model) :

- input에서 무작위하게 몇 개의 token을 mask시킵니다. 그리고 이를 transformer 구조에 넣어서 주변 단어의 context만을 보고 mask된 단어를 예측하는 모델
- BERT에서는 input 전체와 mask된 token을 한번에 transformer encoder에 넣고 원래 token 값을 예측하므로 deep bidirectional 가능해짐.

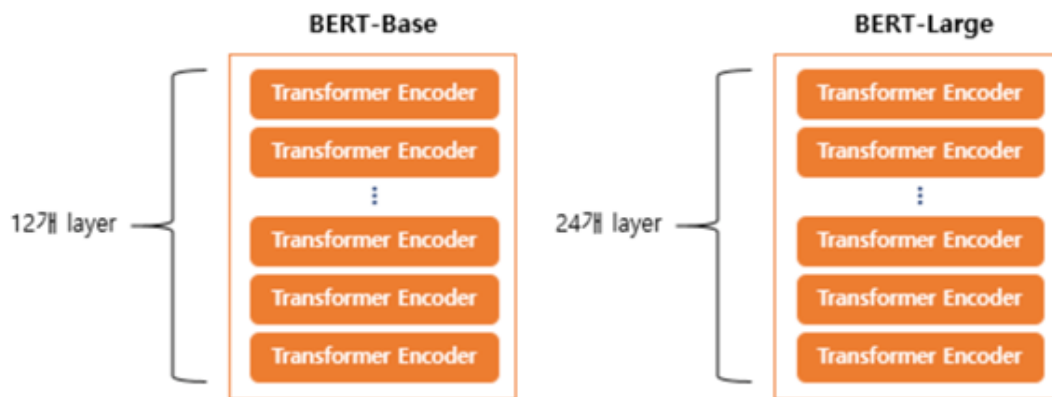
2. next sentence prediction:

- 두 문장을 pre-training시에 같이 넣어줘서 두 문장이 이어지는 문장인지 아닌지 맞추는 것

- pre-training시에는 50:50 비율로 실제로 이어지는 두 문장과 랜덤하게 추출된 두 문장을 넣어줘서 맞춘.
- natural language inference와 같은 task 수행에 도움이 됨

Bert 모델 크기

- base모델 : L=12, H=768, A=12, Total Parameters = 110M
- large 모델 : L=24, H=1024, A=16, Total Parameters = 340M
- L: : transform block의 layer 수, H: hidden size, A: self-attention heads 수, feed-forward/filter size = 4H



- 학습 코퍼스 데이터

English Wikipedia(2500M) words without lists, tables, and headers

30,000 token vocabulary

- 데이터의 tokenizing

WordPiece tokenizing :

He likes playing → He/ likes/ play/ ing

입력 문장을 tokenizing하고, 그 token들로 'token sequence'를 만들어 학습에 사용함.

임베딩

버트는 총 3개의 임베딩 층 사용됨. token embedding, position embedding, segment embedding

Input	[CLS]	my	dog	is	cute	[SEP]	he	likes	play	##ing	[SEP]
Token Embeddings	$E_{[CLS]}$	E_{my}	E_{dog}	E_{is}	E_{cute}	$E_{[SEP]}$	E_{he}	E_{likes}	E_{play}	$E_{\# \# ing}$	$E_{[SEP]}$
	+	+	+	+	+	+	+	+	+	+	+
Segment Embeddings	E_A	E_A	E_A	E_A	E_A	E_A	E_B	E_B	E_B	E_B	E_B
	+	+	+	+	+	+	+	+	+	+	+
Position Embeddings	E_0	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}

먼저 토큰 임베딩은 wordpiece 임베딩 방식을 사용함. 버트는 서브워드 토큰라이저를 사용

Bert의 WordPiece tokenizing

빈도수에 기반해 단어를 의미 있는 패턴(subword)으로 잘라서 tokenizing

서브워드 토큰라이저

- 기본적으로 자주 등장하는 단어는 그대로 단어 집합에 추가하지만, 자주 등장하지 않는 단어의 경우에는 더 작은 단위인 서브워드로 분리되어 서브워드들이 단어 집합에 추가 됩니다.
- 만들어진 단어 집합을 기반으로 토큰화 수행
- transformers 패키지를 사용하여 bert tokenizer 수행

Token Embedding

- Word Piece 임베딩 방식 사용
- 자주 등장하면서 가장 긴 길이의 sub-word를 하나의 단위로 생성함.
- 기존 임베딩 방법은 Out-of-vocabulary(OOV) 문제가 존재하며, 희귀 단어, 이름, 숫자 나 단어장에 없는 단어에 대한 학습, 번역에 어려움이 있음. 그러나, word piece 임베딩 은 모든 언어에 적용 가능하며, sub-word 단위로 단어를 분절하므로 OOV 처리에 효과 적이고 정확도 상승함.

Segment Embedding

- 두 개의 문장 task를 해결할 때 첫번째 문장에는 sentence0 임베딩, 두번째 문장에는 sentence1 임베딩을 더해주는 방식
- 임베딩 벡터는 두개만 사용함. 즉, 한두문장만 계산함.

Position Embedding

- 위치 정보를 학습을 통해서 얻는 것
- 첫번째 단어의 임베딩 벡터에는 0번 포지션 임베딩 벡터, 두번째 단어의 임베딩 벡터에는 +1번 포지션 임베딩 벡터
- 실제 버트에서는 문장의 최대 길이를 512로 하고 있으므로 총 512개의 포지션 임베딩 벡터가 학습됨.

MLM

단어 중의 15%를 [mask] token으로 바꾸어 줌.

- 80%는 token을 [MASK]로 바꿈. ex) my dog is cute. → my dog is [mask].
- 10%는 token을 random word로 바꾸어 줌. ex) my dog is cute → my dog is smart.
- 10%는 token을 원래의 단어로 그대로 놔둡니다.

[mask] token은 pre-training에만 사용되고, fine-tuning에는 사용되지 않습니다.


Next Sentence Prediction

- QA(질문에 맞는 답, question-answer)나 Natural Language Inference(NLI)와 같이 두 문장 사이의 관계를 이해하기 위함.
- Binarized next sentence prediction task
 - 50%는 sentence A, B 가 실제 next sentence
 - 50%는 sentence A, B가 corpus에서 random으로 뽑힌(관계가 없는) 두 문장

ex) [CLS] the man went to [MASK] store [SEP] he bought a gallon [MASK] milk [SEP]

[CLS] the man [MASK] to the store [SEP] penguin [MASK] are flight ##less birds [SEP]

pre-training이 완료되면, 이 task로 97-98%의 accuracy를 달성함.

 BERT 네이버 영화 리뷰 분류 (1).