

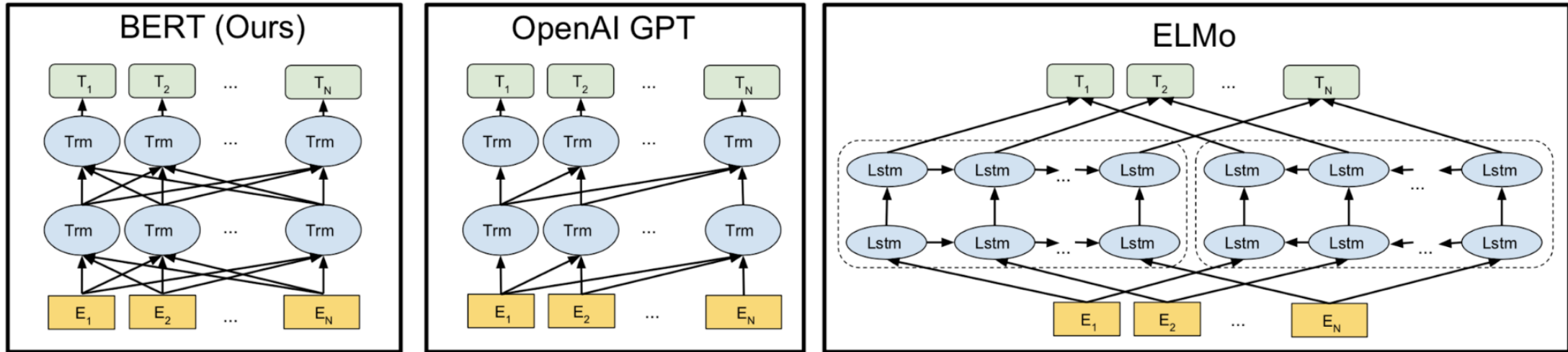
NLP 스터디 4주차

한국어 임베딩

5.6 BERT

이영현

BERT, ELMo, GPT



- BERT 성공비결: **트랜스포머 블록** 사용, 모델의 속성이 **양방향**을 지향
- GPT: 단어 시퀀스를 왼쪽에서 오른쪽으로 한 방향으로만 보는 아키텍처
- ELMo: Bi-LSTM 레이어의 상단은 양방향이지만 중간 레이어는 한 방향

BERT와 GPT 모두 트랜스포머 블록 사용, 근데 GPT는 왜 단어들을 양방향으로 보지 못하는 걸까?


- GPT가 언어모델 주어진 단어 시퀀스를 가지고 그 다음 단어를 예측하는 과정에서 학습이기 때문
 맞춰야 하는데 정답을 미리 알려줄 수 없음


마스크 언어 모델

masked language model

- 주어진 시퀀스 다음 단어를 맞추는 것에서 벗어나, 일단 **문장 전체**를 모델에 알려 주고, **빈칸(MASK)**에 해당하는 단어가 어떤 단어일 지 예측하는 과정에서 학습을 해보자는 아이디어

- 양방향, 단방향 언어 모델

① 나는 어제 _____


② 나는 어제 _____ 먹었다


BERT는 2번

GPT의 학습

예측해야 할 단어를 보지 않기 위해 소프트맥스
스코어 행렬의 일부 값을 0으로 만듦

$$\text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V =$$

	뜨끈한	국밥	한 그릇
뜨끈한	0	0	0
국밥	1.0	0	0
한 그릇	0.9	0.1	0

$$\begin{pmatrix} V_{\text{뜨끈한}} \\ V_{\text{국밥}} \\ V_{\text{한 그릇}} \end{pmatrix}$$

$$=$$

	뜨끈한	국밥	한 그릇
뜨끈한	0. $V_{\text{뜨끈한}}$	0. $V_{\text{국밥}}$	0. $V_{\text{한 그릇}}$
국밥	1.0 $V_{\text{뜨끈한}}$	0. $V_{\text{국밥}}$	0. $V_{\text{한 그릇}}$
한 그릇	0.9 $V_{\text{뜨끈한}}$	0.1 $V_{\text{국밥}}$	0. $V_{\text{한 그릇}}$

국밥을 맞추려면 뜨끈한만 참고 가능

BERT의 학습

문장 내 단어 쌍 사이의 관계를 모두 볼 수 있음

$$\text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V =$$

$$\begin{array}{c} \text{드디어} \text{ } \text{금요일} \text{ } \text{이다} \\ \text{드디어} \begin{pmatrix} 0.2 & 0.7 & 0.1 \\ \dots & \dots & \dots \\ \dots & \dots & \dots \end{pmatrix} \begin{pmatrix} V_{\text{드디어}} \\ V_{\text{금요일}} \\ V_{\text{이다}} \end{pmatrix} \\ \text{금요일} \\ \text{이다} \end{array}$$

$$\begin{array}{c} \text{드디어} \begin{pmatrix} 0.2V_{\text{드디어}} & 0.7V_{\text{금요일}} & 0.1V_{\text{이다}} \\ \dots & \dots & \dots \\ \dots & \dots & \dots \end{pmatrix} \\ \text{금요일} \\ \text{이다} \end{array}$$

※ 같은 BERT 모델이라도 프리트레이닝을 할 때 한 방향만 보게 할 경우 그 성능이 크게 감소
→ 그만큼 양방향 전후 문맥을 모두 보게 하는 것이 중요!

마스크 언어 모델 태스크 수행 학습 데이터 구축

- 학습 데이터 한 문장 토큰의 15%를 마스크
- 마스크 대상 토큰 가운데 80%는 실제 빈칸으로 만들고, 모델은 그 빈칸을 채운다. 예: 발 없는 말이 [MASK] 간다 -> 천리
- 마스크 대상 토큰 가운데 10%는 랜덤으로 다른 토큰으로 대체하고, 모델은 해당 위치의 정답 단어가 무엇일지 맞추도록 한다. 예: 발 없는 말이 [컴퓨터] 간다 -> 천리
- 마스크 대상 토큰 가운데 10%는 토큰 그대로 두고, 모델은 해당 위치의 정답 단어가 무엇일지 맞추도록 한다. 예: 발 없는 말이 [천리] 간다 -> 천리

마스크 언어 모델 태스크 수행 학습 데이터 구축

〈기대〉

- 발 없는 말이 [MASK] 간다의 빈칸을 채워야 하기 때문에 문장 내 어느 자리에 어떤 단어를 쓰는 게 자연스러운지 앞뒤 문맥을 읽어낼 수 있게 된다.
- 발 없는 말이 천리 간다 발 없는 말이 컴퓨터 간다를 비교해 보면서 주어진 문장이 의미/문법상 비문인지 아닌지 가려낼 수 있다.
- 모델은 어떤 단어가 마스크될지 전혀 모르기 때문에 문장 내 모든 단어 사이의 의미적, 문법적 관계를 세밀히 살피게 된다.

NSP를 맞추기 위한 학습 데이터 구축

- 모든 학습 데이터는 1건당 **문장 두 개**로 구성된다.
- 이 가운데 절반은 동일한 문서에서 실제 이어지는 문장을 두 개 뽑고, 그 정답으로 **참**을 부여한다.
- 나머지 절반은 서로 다른 문서에서 문장 하나씩 뽑고, 그 정답으로 **거짓**을 부여한다.
- max_num_tokens를 정의한다.
 - ① 학습 데이터의 90%는 max_num_tokens 가 사용자가 정한 max_sequence_length가 되도록 한다.
 - ② 나머지 10%는 max_num_tokens 가 max_sequence_length 보다 짧게 되도록 랜덤으로 정한다.
- 이전에 뽑은 문장 두 개의 단어 총 수가 max_num_tokens을 넘지 못할 때까지 두 문장 중 단어 수가 많은 쪽을 50%의 확률로 문장 맨 앞 또는 맨 뒤 단어 하나씩 제거한다.

NSP를 맞추기 위한 학습 데이터 구축

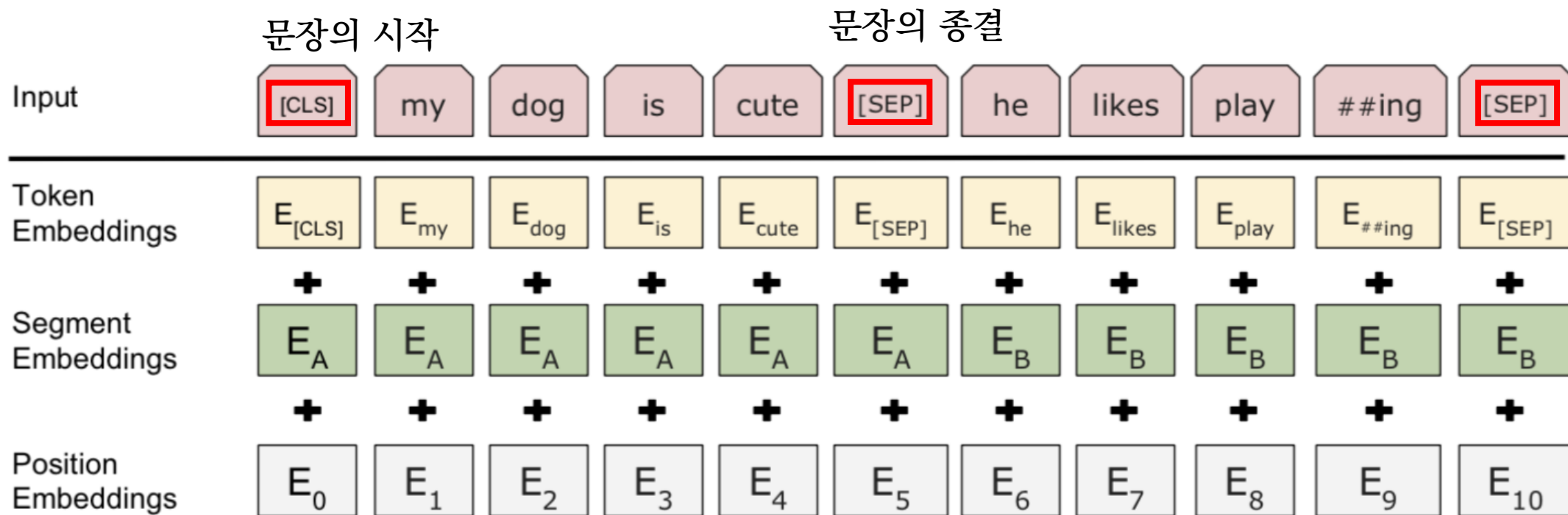
<기대>

- 모델은 **애비는 종이였다, 밤이 깊어도 오지 않았다**가 이어진 문장인지 아닌지 **반복 학습**한다. 따라서 문장 간 의미 관계를 이해할 수 있다.
- 일부 문장 성분이 없어도 전체 의미를 이해하는 데 큰 무리가 없다. NSP 테스트가 너무 쉬워지는 것을 방지하기 위해 문장 맨 앞 또는 맨 뒤쪽 단어 일부를 삭제했기 때문이다.
- 학습 데이터에 짧은 문장이 포함돼 있어도 성능이 크게 떨어지지 않는다. 학습 데이터의 10%는 사용자가 정한 최대 길이(max_sequence_length)보다 짧은 데이터로 구성돼 있기 때문이다.

BERT 모델 구조

트랜스포머 인코더를 일부 변형한 아키텍처

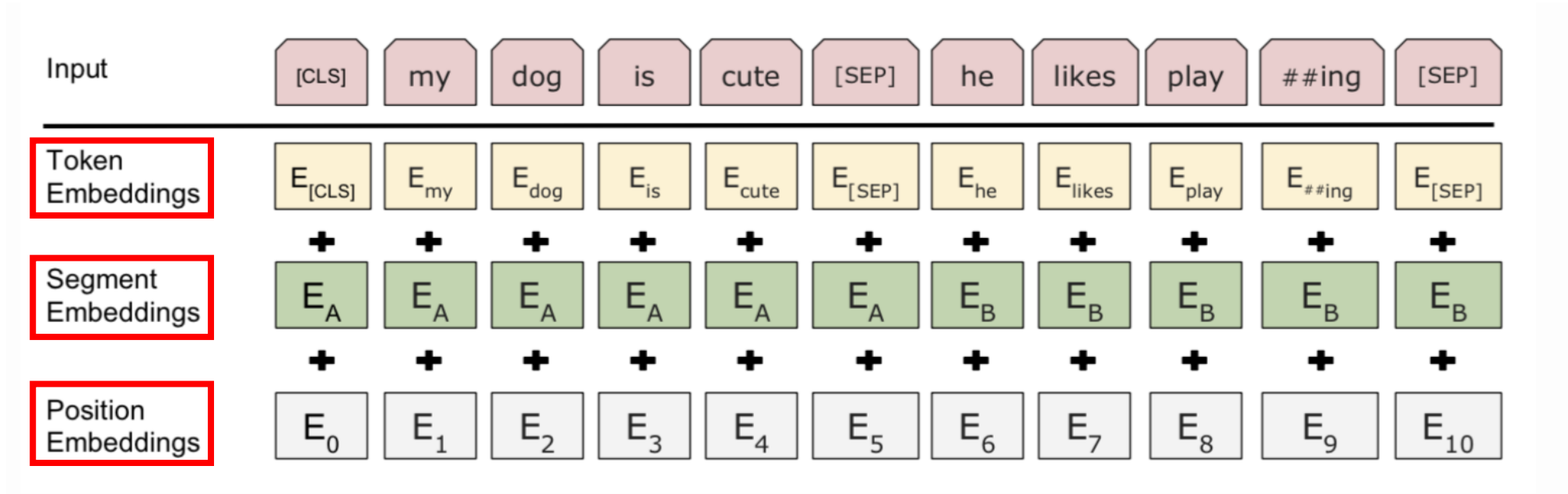
with 스페셜 토큰



- [MASK]: 마스크 토큰
- [PAD]: 배치 데이터의 길이를 맞춰 주기 위한 토큰

BERT 모델 구조

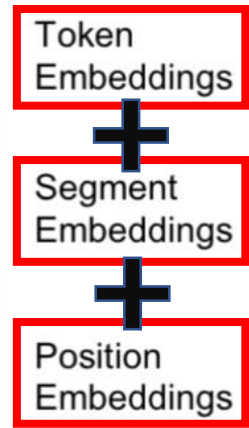
트랜스포머 인코더를 일부 변형한 아키텍처



- 토큰 임베딩: 입력 토큰에 해당하는 토큰 벡터를 참조해 만듦
- 세그먼트 임베딩: 해당 문장인지
- 포지션 임베딩: 입력 토큰의 문장 내 절대적인 위치

BERT 모델 구조

트랜스포머 인코더를 일부 변형한 아키텍처



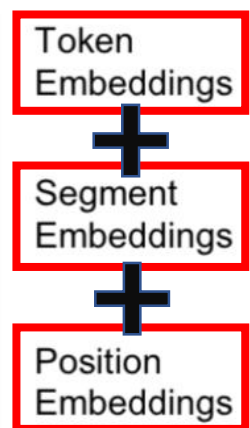
토큰, 세그먼트, 포지션 벡터를 만들 때 참조하는 행렬은 프리트레인 태스크 수행을 잘하는 방향으로 다른 학습 파라미터와 함께 업데이트 된다.

각각의 벡터에 레이어 정규화 & 드롭아웃

=> 첫 번째 트랜스포머 블록의 입력 행렬(11X hidden dim)

BERT 모델 구조

트랜스포머 인코더를 일부 변형한 아키텍처



첫 번째 트랜스포머 블록

멀티헤드 어텐션 레이어



Position-wise
Feedforward Networks

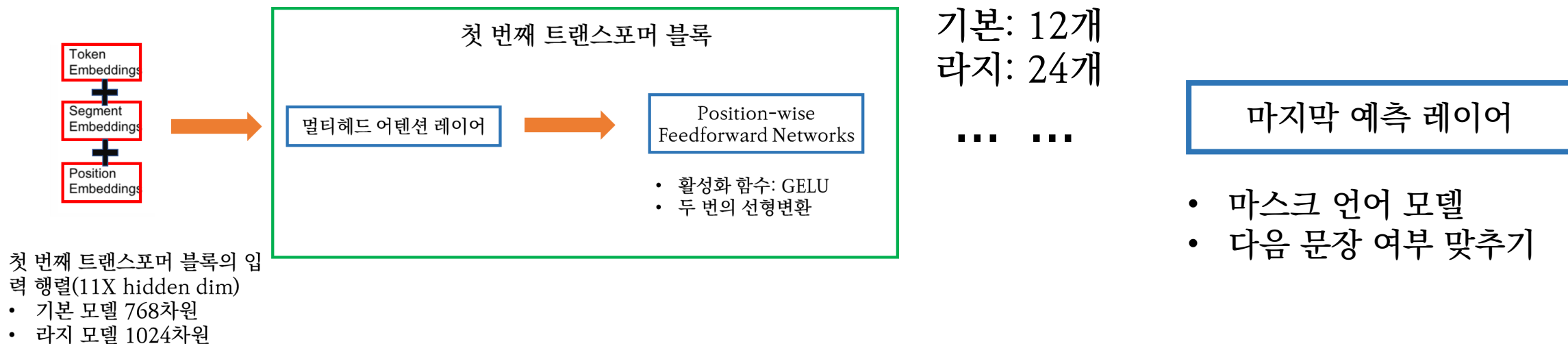
- 활성화 함수: GELU
- 두 번의 선형변환

첫 번째 트랜스포머 블록의 입력 행렬(11X hidden dim)

- 기본 모델 768차원
- 라지 모델 1024차원

BERT 모델 구조

트랜스포머 인코더를 일부 변형한 아키텍처



BERT 모델 구조

트랜스포머 인코더를 일부 변형한 아키텍처

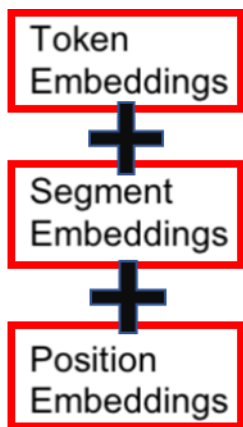
- 마스크 언어 모델

마지막 트랜스포머 블록의 마스크 위치에 해당하는 토큰 벡터



마지막 예측 레이어

- BERT 모델 입력 문장



발 없는 말이 [MASK] 간다

Input_tensor: 네 번째 벡터



입력 당시와 동일한 차원 수로 선형변환



레이어 정규화



로짓 벡터 생성



어휘 집합만큼의 차원수로 사영하는 가중치 행렬을 곱하고 output_bias 벡터를 더해 생성

소프트맥스를 취한 확률 벡터와 정답(천리라는 단어의 인덱스만 1이고 나머지는 0인 원핫벡터) 사이의 크로스 엔트로피를 구하고 이를 최소화하는 방향으로 모델 파라미터 업데이트

BERT 모델 구조

트랜스포머 인코더를 일부 변형한 아키텍처

- 다음 문장인지 여부를 맞추기 위한 레이어

마지막 트랜스포머 블록의 첫 번째 토큰([CLS])에 해당하는 벡터



마지막 예측 레이어

Input_tensor: [CLS]



2차원수로 사영하는
가중치 행렬을 곱하
고 2차원 크기의
output_bias 벡터
를 더해 생성



소프트맥스를 취한 확률 벡터와
정답(참 혹은 거짓) 사이의 크로스 엔트로피를 구하고 이를 최
소화하는 방향으로 모델 파라미
터 업데이트

BERT 모델 프리트레인

- 데이터 전처리

이것은 첫 번째 문서의 첫 번째 문장입니다.

이것은 첫 번째 문서의 두 번째 문장입니다.

이것은 두 번째 문서의 첫 번째 문장입니다.

- 어휘 집합 구축

구글 센텐스피스를 사용해 **바이트 페어 인코딩 BPE** 방식(비지도 학습 기반 형태소 분석기)의

어휘 집합 생성

['집에', '##좀', '가자']

BERT 모델 프리트레인

- 학습 데이터 구축

max_seq_length: 문서 하나에 속하는 토큰 최대 수

max_predictions_per_seq: 마스크 언어 모델로 예측할 토큰 수의 최대치

masked_lm_prob: 문서 하나당 마스킹하는 토큰 비율

dupe_factor: 동일한 말뭉치에서 학습 데이터를 몇 번을 반복해 만들지 정하는 옵션

* BERT는 다음 문장 예측용 데이터를 만들 때 앞뒤 문장을 랜덤하게 선택, 앞뒤 토큰 제거 등으로 인해 랜덤성 보유

- BERT 모델의 하이퍼파라미터

dropout: 드롭아웃 비율, hidden_act: feedforward 네트워크의 활성화함수 종류,
hidden_size, initializer_range, intermediate_size,
max_position_embeddings, num_attention_heads, num_hidden_layers,
vocab_size, type_vocab_size

- 모델 프리트레이닝

- 파인 튜닝