17장 BERT

Bidirectional Encoder Representations

from Transformers

엄호용

목차 Table of Contents

01 NLP에서의 사전 훈련(Pre-training)

02 버트(BERT)

03 구글 BERT의 마스크드 언어 모델 실습

04 한국어 BERT의 마스크드 언어 모델 실습

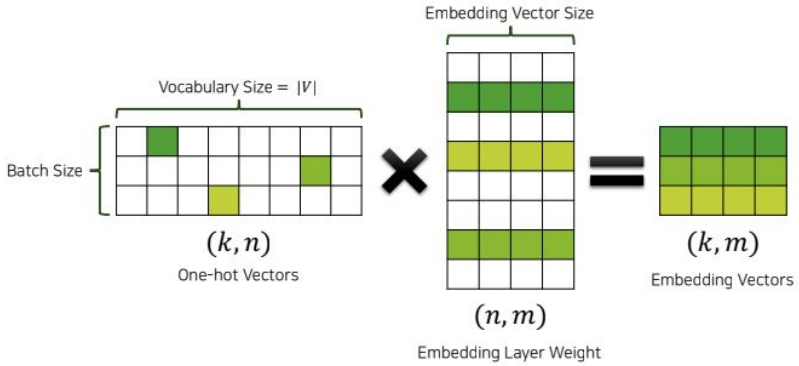
05 구글 BERT의 다음 문장 예측

06 한국어 BERT의 다음 문장 예측

07 센텐스 버트(SBERT)

01 NLP에서의 사전 훈련 Pre-training

워드 임베딩



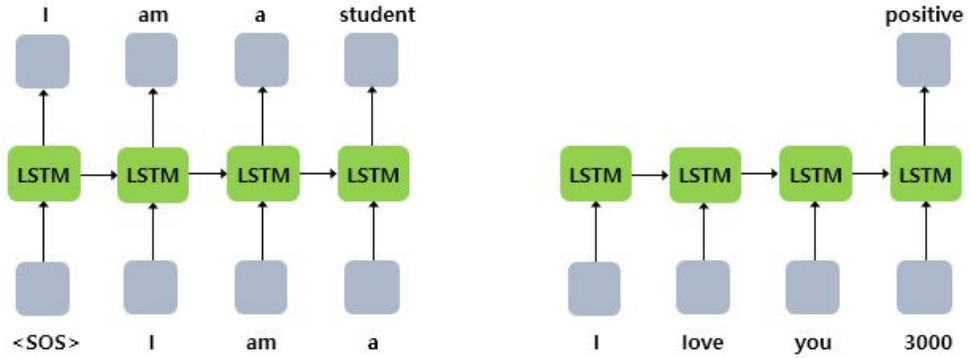
- 랜덤 초기화 후 학습
- 사전 학습된 벡터를 가져와 사용

단어 - 벡터값 일대일 mapping

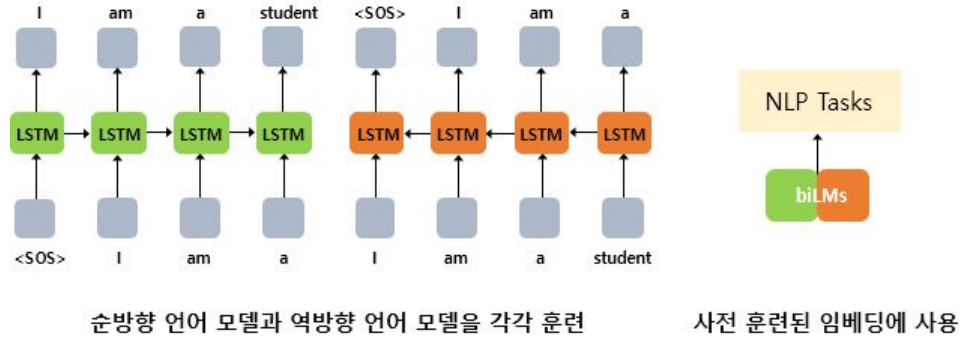
다의어/동음이의어 구분 불가

LSTM 사전 학습

Semi-Supervised Sequence Learning, Google, 2015



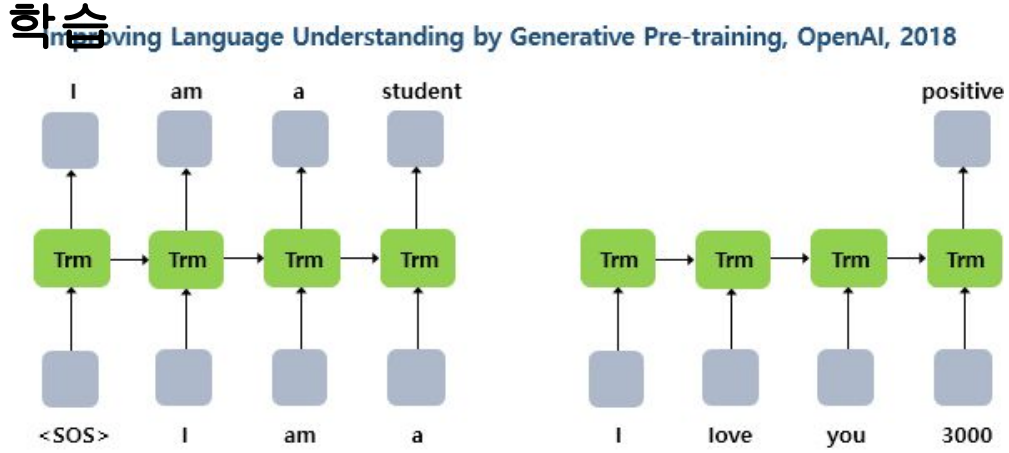
ELMo: Deep Contextual Word Embedding, AI2 & University of Washington, 2017



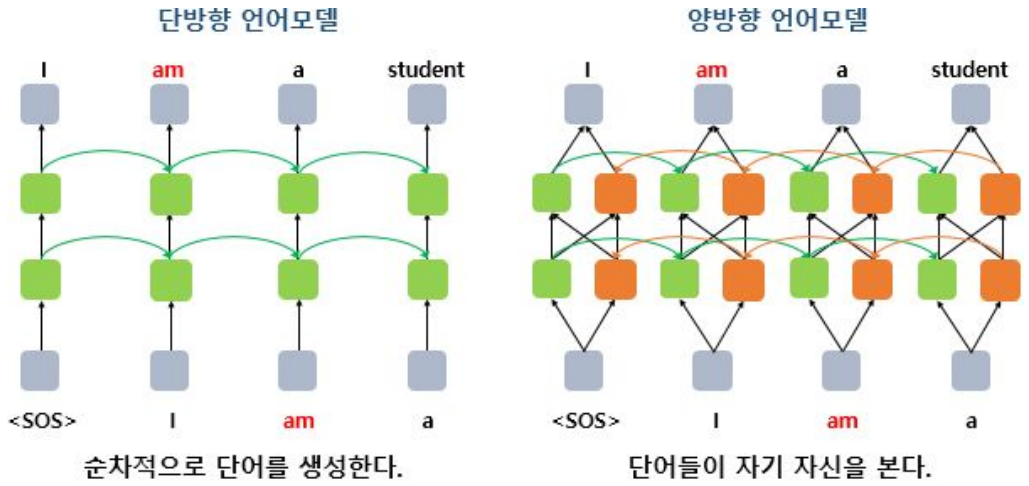
다의어/동음이의어 문제 해결

01 NLP에서의 사전 훈련 Pre-training

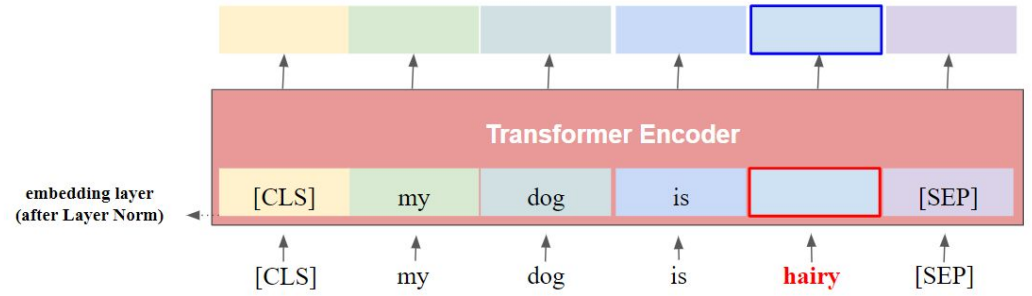
트랜스포머 사전 학습



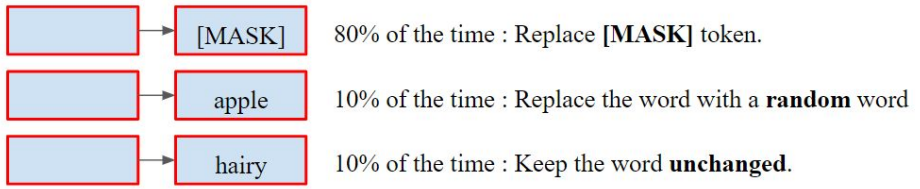
양방향 언어 모델



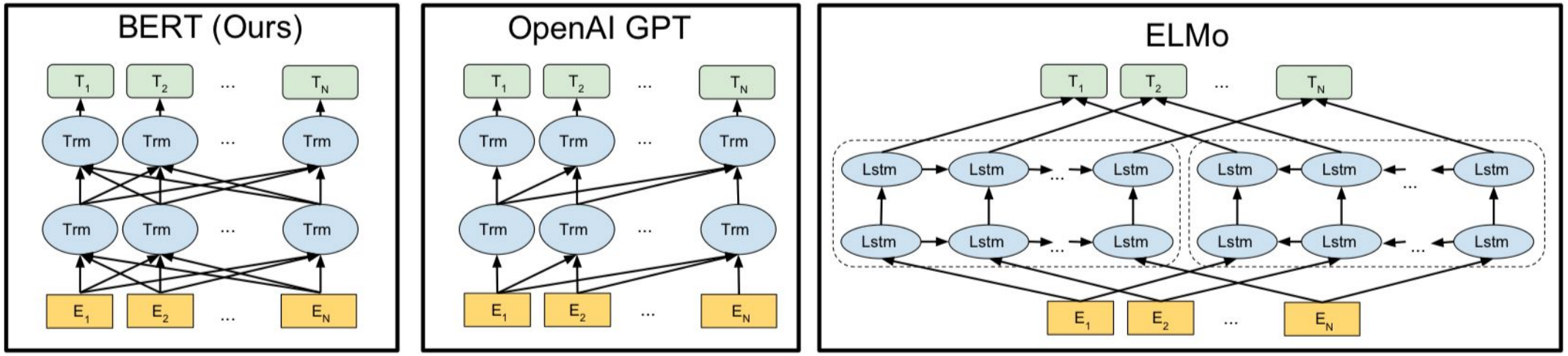
Masked 언어 모델

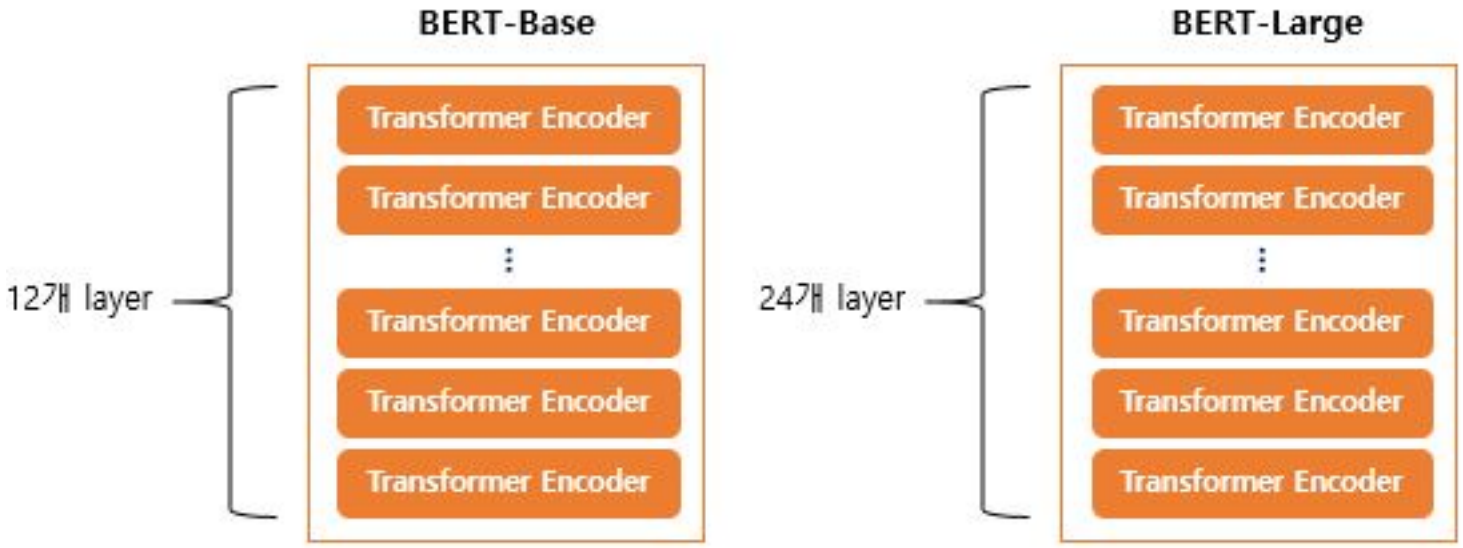


Mask **15%** of all WordPiece tokens in each sequence at **random**. (e.g., **hairy**)



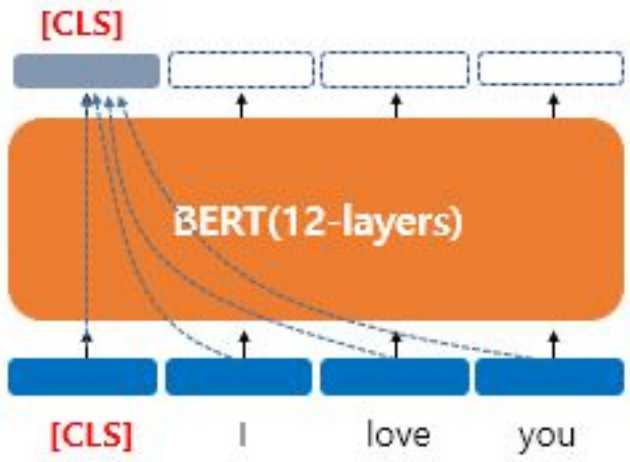
02 BERT Bidirectional Encoder Representations from Transformers

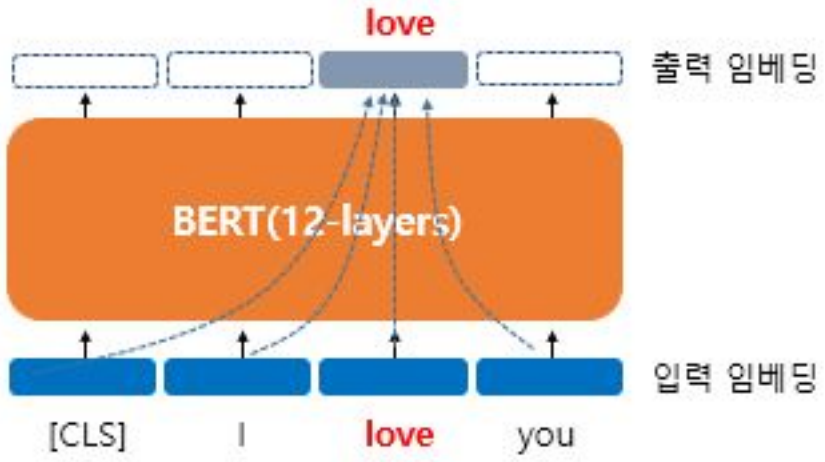


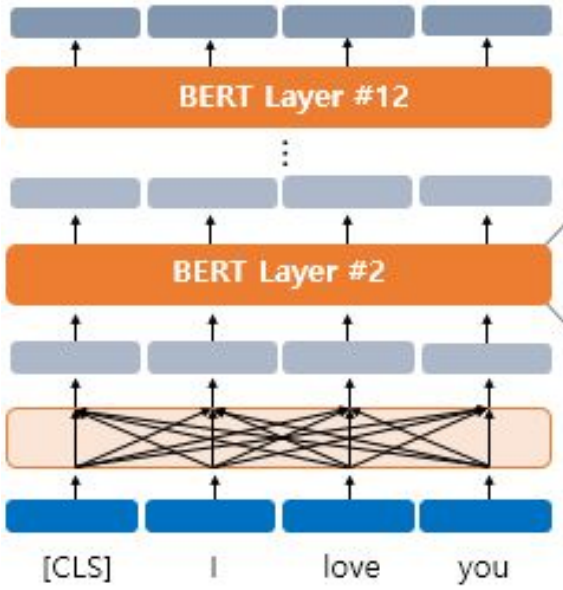


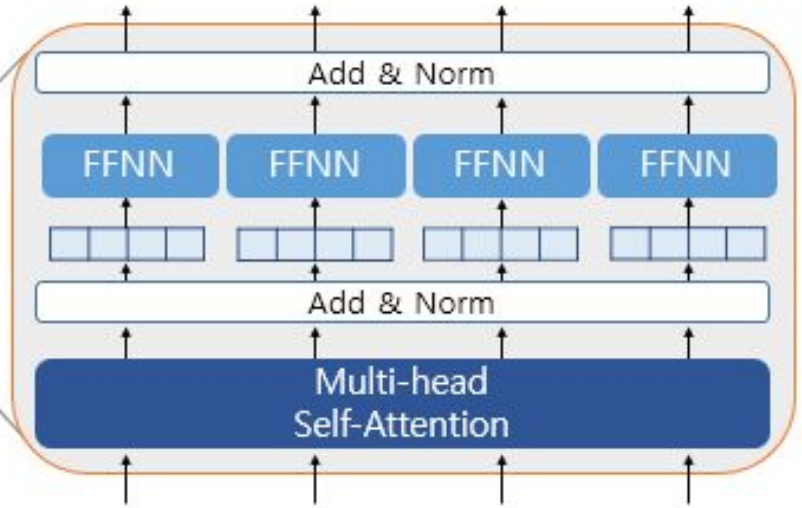
- BERT-Base : L=12, D=768, A=12 : 110M개의 파라미터
- BERT-Large : L=24, D=1024, A=16 : 340M개의 파라미터

02 BERT Bidirectional Encoder Representations from Transformers

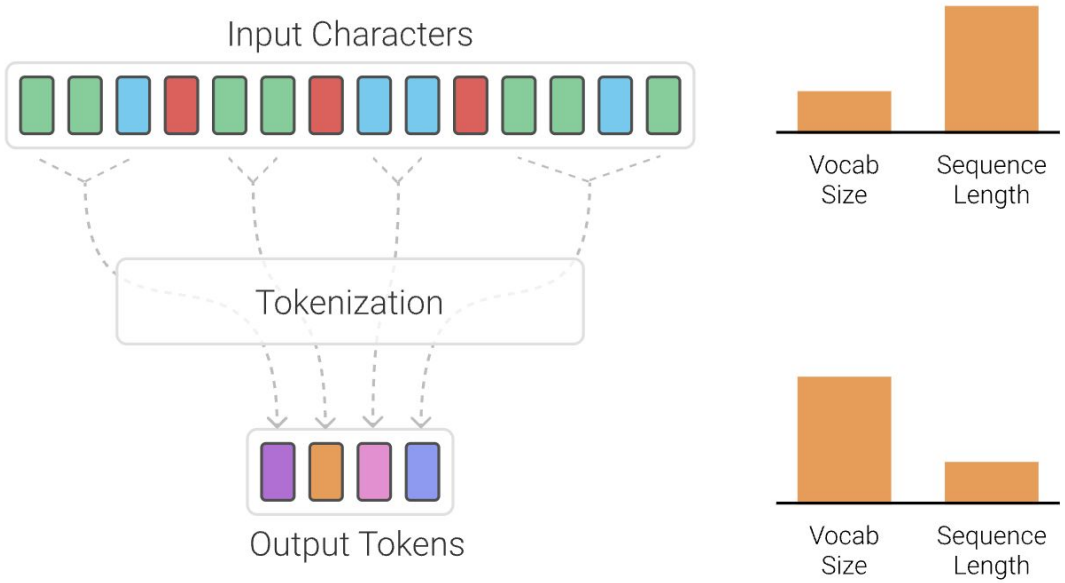








02 BERT Bidirectional Encoder Representations from Transformers

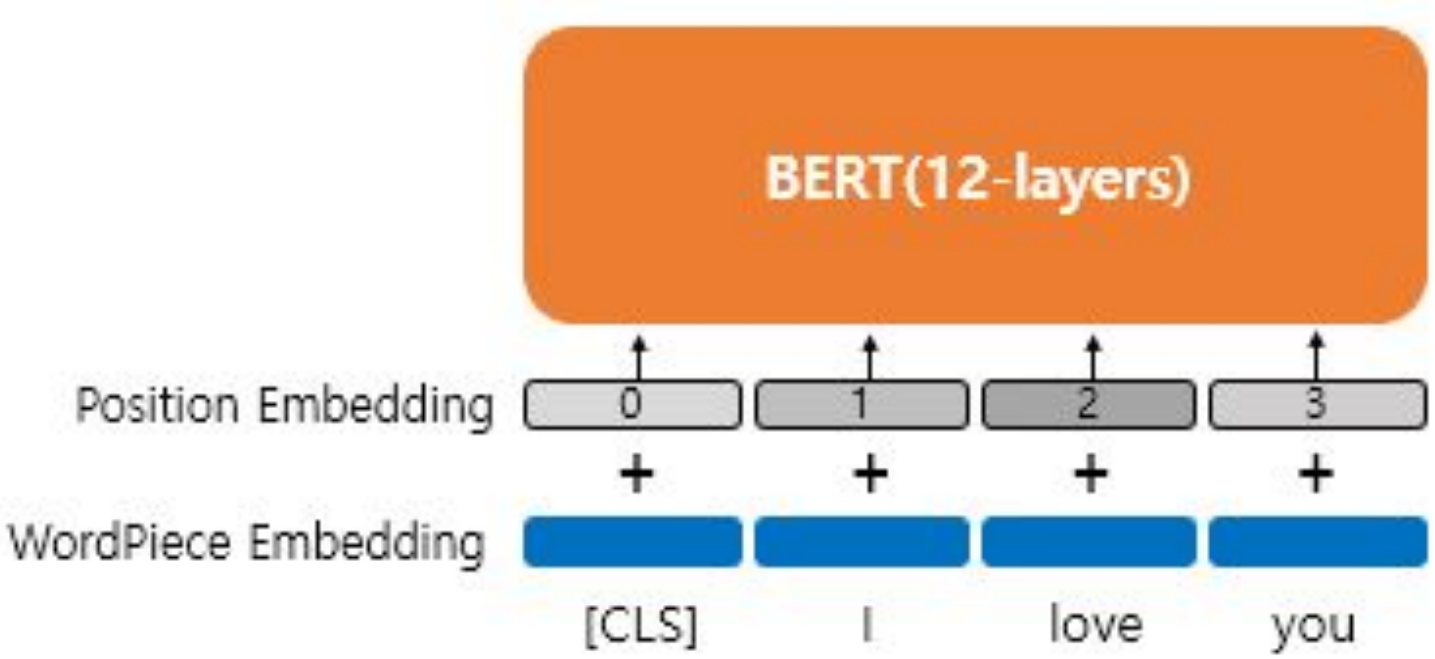



```
result = tokenizer.tokenize('Here is the sentence I want embeddings for.')
print(result)
```

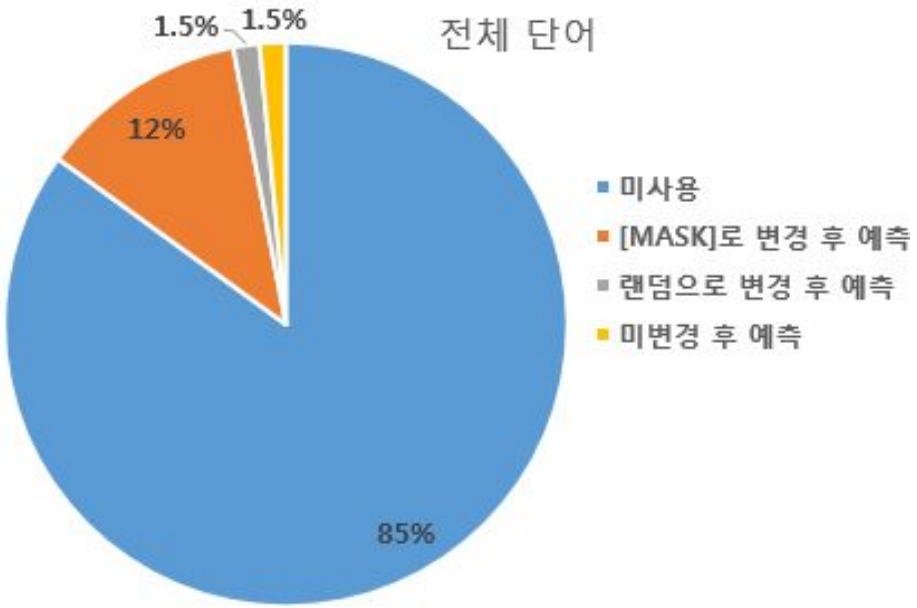
Copy

```
['here', 'is', 'the', 'sentence', 'i', 'want', 'em', '##bed', '##ding', '##s', 'for', '.']
```

02 BERT Bidirectional Encoder Representations from Transformers

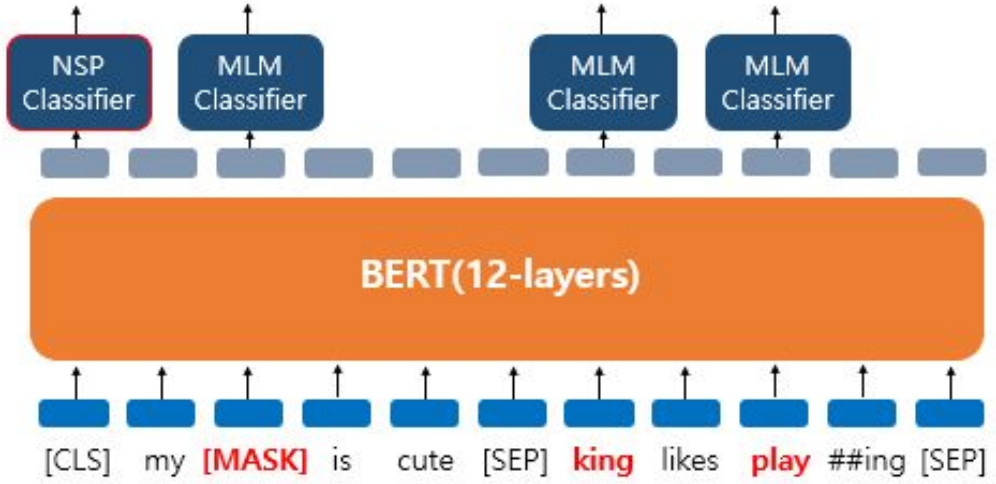


02 BERT Bidirectional Encoder Representations from Transformers



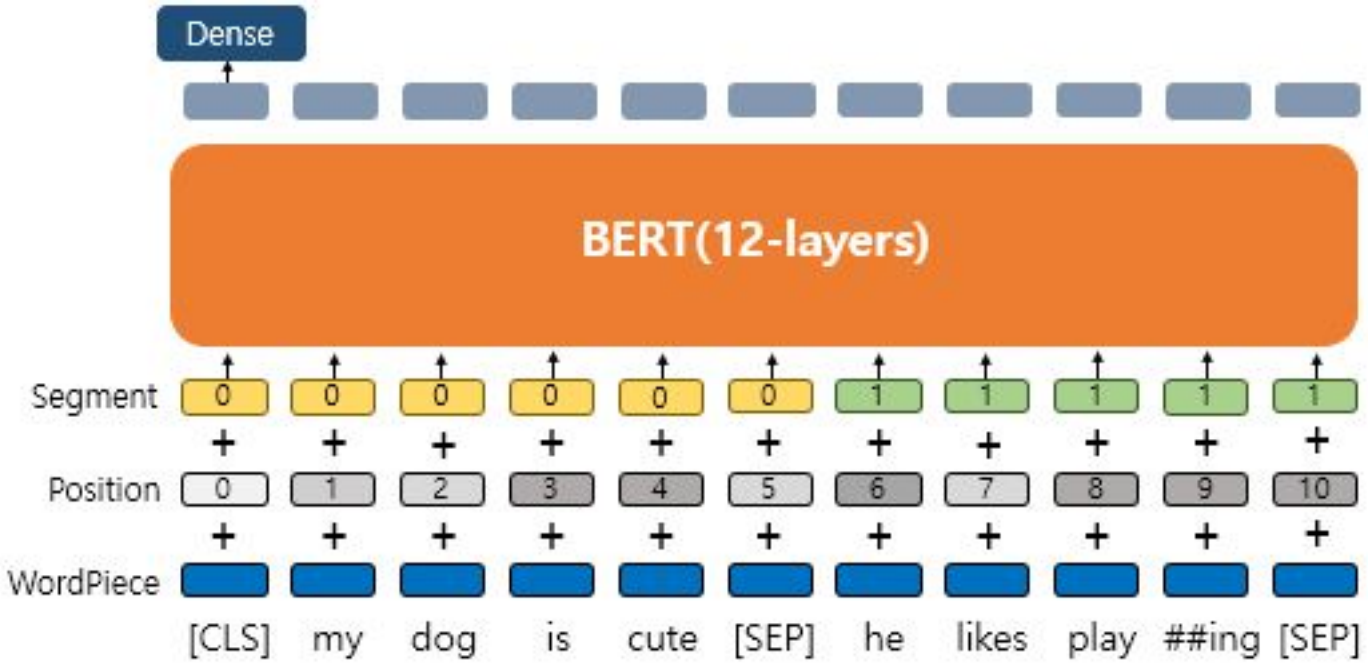
- 80%의 단어들은 [MASK]로 변경한다.
  Ex) The man went to the store → The man went to the [MASK]
- 10%의 단어들은 랜덤으로 단어가 변경된다.
  Ex) The man went to the store → The man went to the dog
- 10%의 단어들은 동일하게 둔다.
  Ex) The man went to the store → The man went to the store

'My dog is cute. he likes playing'



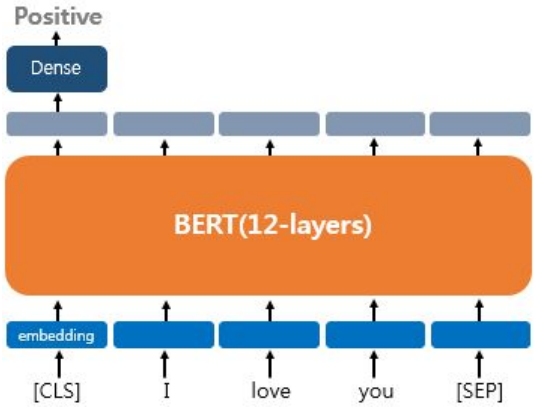
- 이어지는 문장의 경우
  Sentence A : The man went to the store.
  Sentence B : He bought a gallon of milk.
  Label = IsNextSentence
- 이어지는 문장이 아닌 경우 경우
  Sentence A : The man went to the store.
  Sentence B : dogs are so cute.
  Label = NotNextSentence

02 BERT Bidirectional Encoder Representations from Transformers

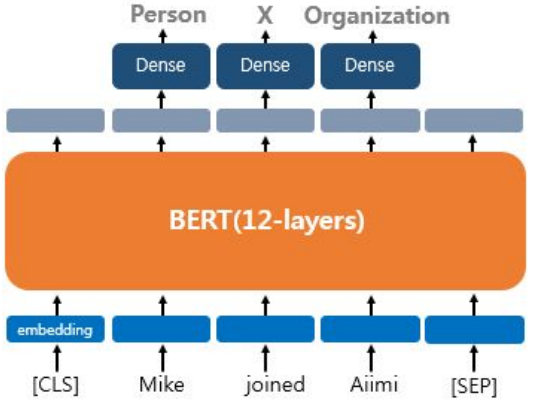


02 BERT Bidirectional Encoder Representations from Transformers

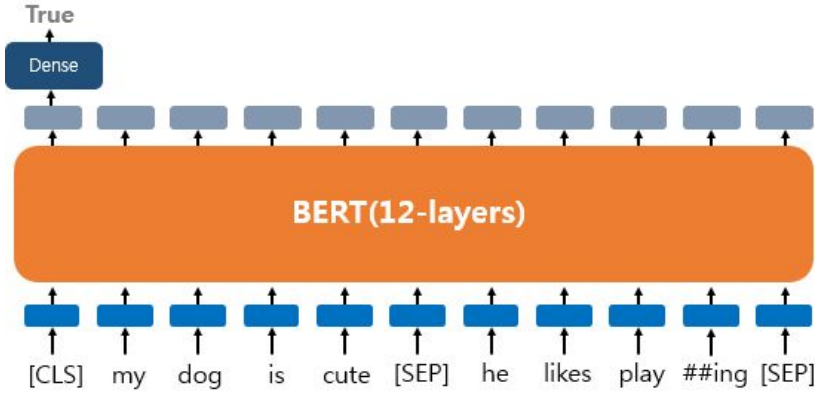
하나의 텍스트에 대한 텍스트 분류 유형



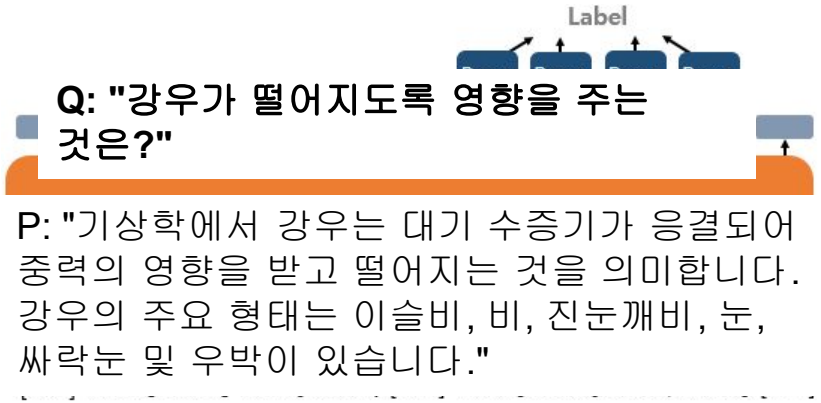
하나의 텍스트에 대한 유형 태깅 작업



텍스트의 쌍에 대한 분류 또는 회귀

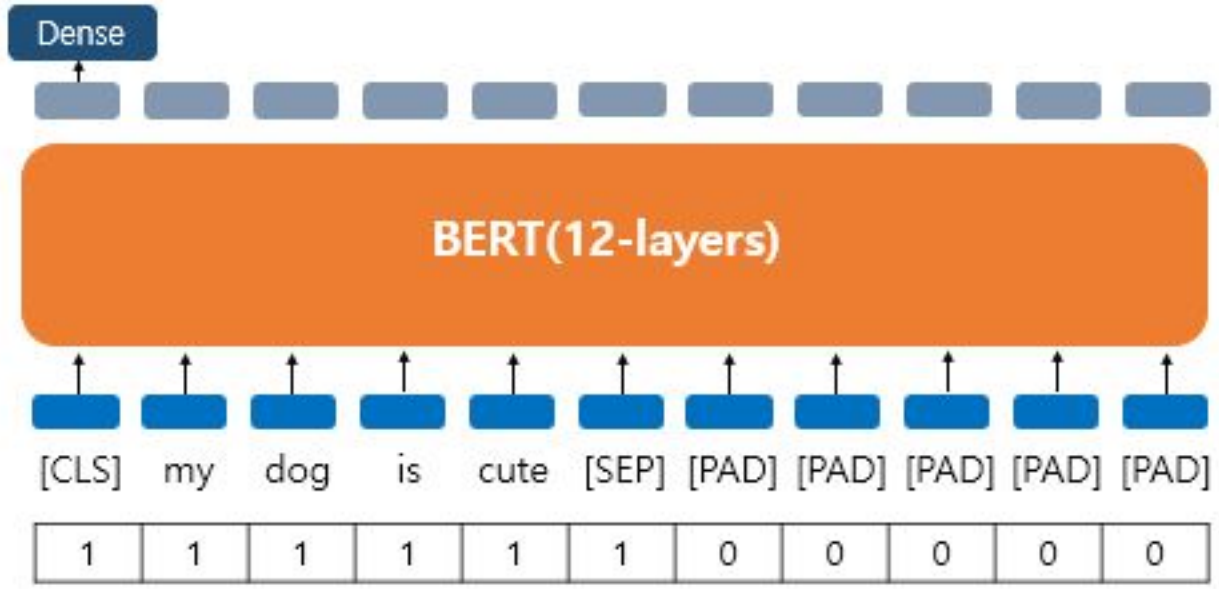


질의 응답(Question Answering)



02 BERT Bidirectional Encoder Representations from Transformers

어텐션 마스크



03 구글 BERT의 마스크드 언어 모델 실습
04 한국어 BERT의 마스크드 언어 모델 실습
05 구글 BERT의 다음 문장 예측
06 한국어 BERT의 다음 문장 예측

```
from transformers import FillMaskPipeline
pip = FillMaskPipeline(model=model, tokenizer=tokenizer)
```

```
pip('Soccer is a really fun [MASK].')
```

```
[{'score': 0.762112021446228,
  'sequence': 'soccer is a really fun sport.',
  'token': 4368,
  'token_str': 'sport'},
 {'score': 0.2034197747707367,
  'sequence': 'soccer is a really fun game.',
  'token': 2208,
  'token_str': 'game'},
 {'sco
  'seq
  'tok
  'tok
 {'sco
  'seq
  'tok
  'tok
 {'sco
  'seq
  'tok
  'tok
```

```
# 상관없는 두 개의 문장
prompt = "In Italy, pizza served in formal settings, such as at a restaurant, is presented unsliced."
next_sentence = "The sky is blue due to the shorter wavelength of blue light."
encoding = tokenizer(prompt, next_sentence, return_tensors='tf')

logits = model(encoding['input_ids'], token_type_ids=encoding['token_type_ids'])[0]

softmax = tf.keras.layers.Softmax()
probs = softmax(logits)
print('최종 예측 레이블 :', tf.math.argmax(probs, axis=-1).numpy())
```

```
최종 예측 레이블 : [1]
```

```
pip('축구는 정말 재미있는 [MASK]다.')
```

```
[{'score': 0.8963505625724792,
  'sequence': '축구는 정말 재미있는 스포츠 다.',
  'token': 4559,
  'token_str': '스포츠'},
 {'score': 0.02595764957368374,
  'sequence': '축구는 정말 재미있는 거 다.',
  'token': 568,
  'token_str': '거'},
 {'score': 0.010033931583166122,
  'sequence': '축구는 정말 재미있는 경기 다.',
  'token': 3682,
  'token_str': '경기'},
 {'
  '
  '
  '
 {'
  '
  '
  '
```

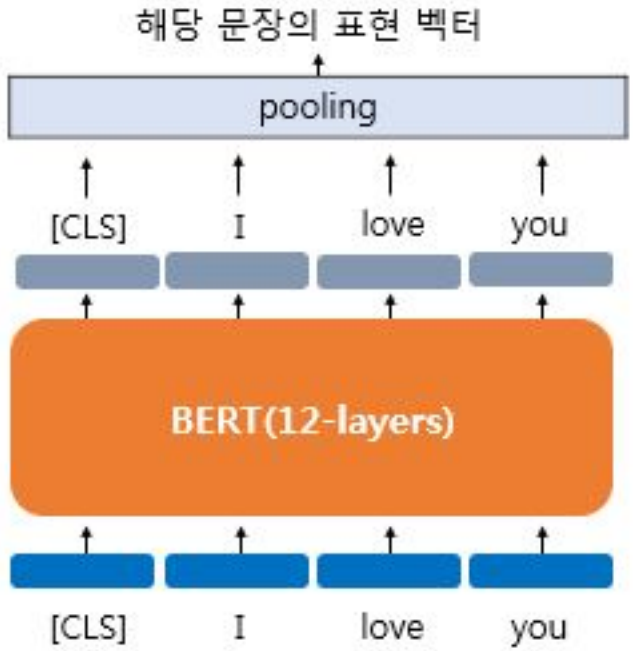
```
# 상관없는 두 개의 문장
prompt = "2002년 월드컵 축구대회는 일본과 공동으로 개최되었던 세계적인 큰 잔치입니다."
next_sentence = "극장가서 로맨스 영화를 보고싶어요"
encoding = tokenizer(prompt, next_sentence, return_tensors='tf')

logits = model(encoding['input_ids'], token_type_ids=encoding['token_type_ids'])[0]

softmax = tf.keras.layers.Softmax()
probs = softmax(logits)
print('최종 예측 레이블 :', tf.math.argmax(probs, axis=-1).numpy())
```

```
최종 예측 레이블 : [1]
```

07 센텐스 버트 SBERT



감사합니다

Q&A