BERT
Pre-training of Deep Bidirectional Transformers for Language Understanding(2018)

FOM

Focus On Data Mining

INDEX

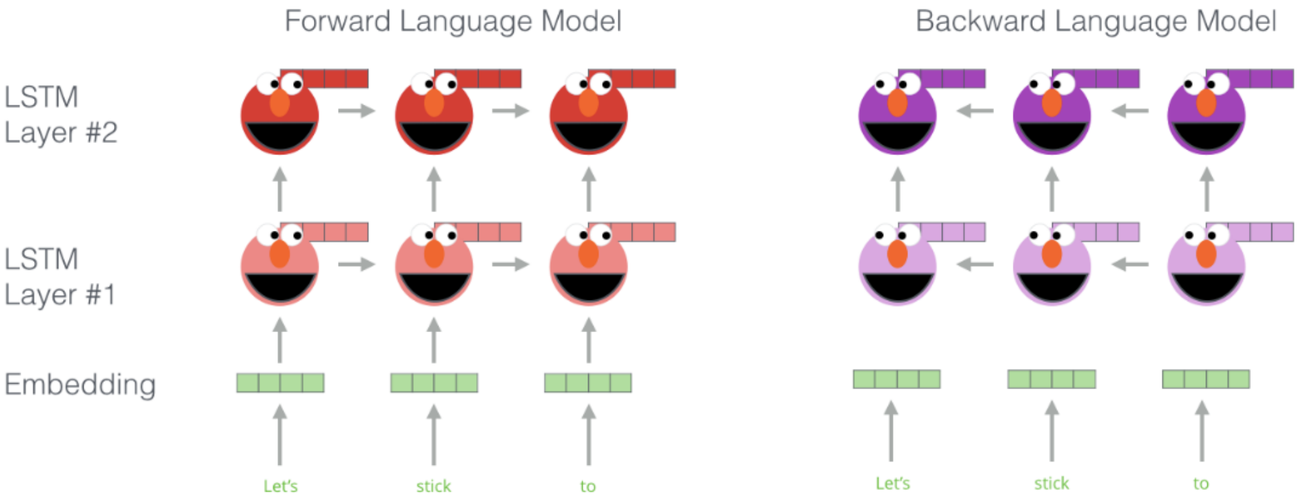
01 Introduction

Background

- Transformer의 등장
  - 2017년 Transformer의 등장으로 attention mechanism을 사용한 병렬적인 encoding-decoding이 등장
- Pre-training의 중요성
  - 2018년 ELMo의 등장으로 pre-training을 통한 전이 학습의 등장
- Bidirectional LM의 기능
  - ELMo등 Bi-LM의 좋은 예측 성능으로 Bi-LM의 등장
- Large Language Model (LLM)의 등장
  - Open-AI의 GPT-1의 등장으로 LLM의 중요성이 올라감

02 Related Works

Bidirectional LM(Language Model)/ Contextual Embedding - ELMo



- **Forward, Backward LM**을 통해 **Bidirectional** 학습 수행 후 **context에 민감한 embedding** 및 정보를 학습함

- 일반적인 NLP 모델에서는 통상적으로 순방향(left-to-right)오 학습 및 예측을 진행

-> 하지만 양방향 LM은 양방향 학습 때문에 문장에 대한 leakage가 발생

-> **BERT**에서는 **masked word**를 예측하는 방식으로 **Bi-LM**을 활용

-> 추가적으로 ELMo에서는 BERT와 비슷한 embedding 방식인 **Character embedding**을 사용

One-directional

Bi-directional

02 Related Works

Unsupervised Feature-based Learning

- Unsupervised Learning -> Not Labeling learning

  - 일반적인 NLP model(Seq2Seq,RNN based model)들은 순차적으로 문장들을 읽으면서 다음 단어가 어떤 단어가 등장할 지 예측 (Object to rank candidate next sentence)

  - ELMo는 양방향(순방향/역방향) 학습을 통해 contextual word embedding을 수행

- ELMo는 여러 개의 benchmarking이 수행 됨

  - QA(Question-Answering),sentiment analysis, 그리고 named entity recognition(pos tagging)
  - Bidirectional model의 좋은 성능

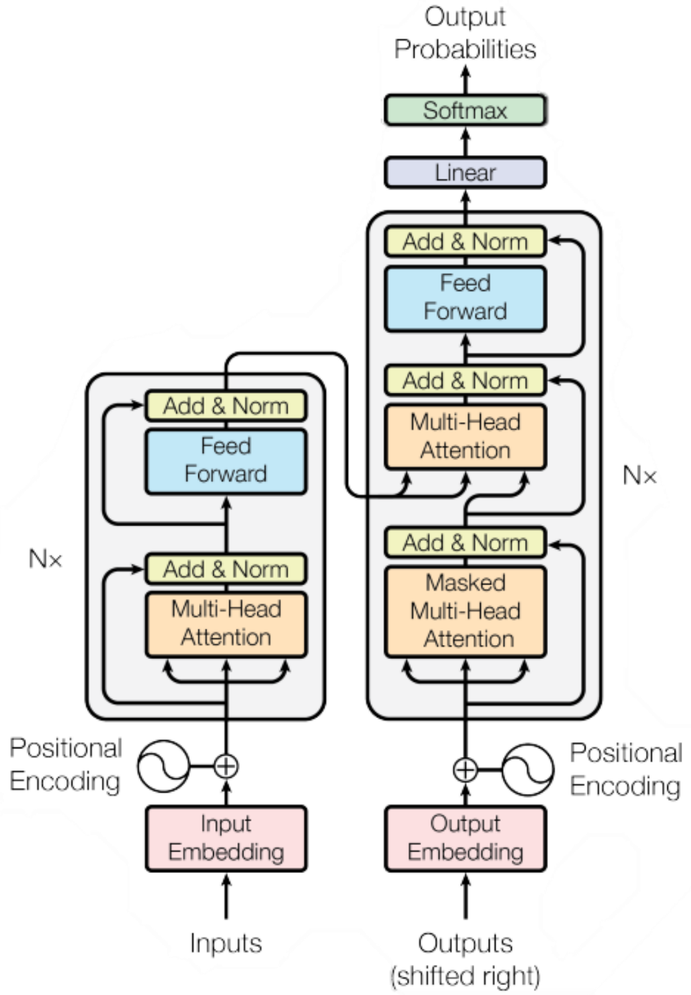
02 | Related Works

Transformer

- **Attention**을 **병렬적 처리**를 통해 엄청난 속도 상승을 수행
  - Multi-head Attention으로 encoder-decoder layer를 구성
  - 각각 6개의 encoder-decoder layer들로 transformer를 구성
  - 각 encoder-decoder들의 input, output dim이 모두 같아 연속적인 학습이 가능

- Multi-head Attention을 통해 contextual embedding진행
  - context에 민감한 encoding 결과물을 얻을 수 있음

- RNN을 완전히 대체하며 GPU를 사용한 빠른 속도
  - positional encoding을 통해 RNN의 기능도 완벽히 수행



02 Related Works

Positional encoding

- **Positional encoding**은 RNN의 기능 중 하나인 문장 속 단어의 위치를 파악하기 위한 encoding

  - Positional encoding에는 **Relative, Absolute position**이 있는데 **Transformer**에서는 **relative position**을 사용함

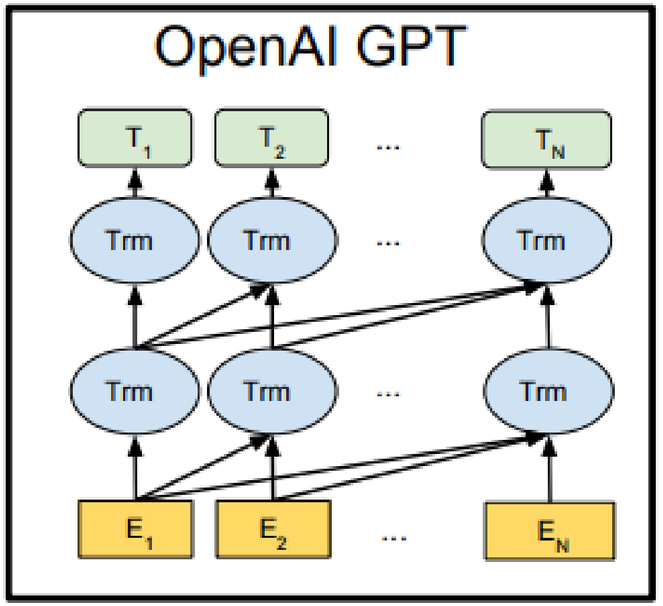
  - **Relative position**는 **sin,cos등 사인파 함수**를 사용해서 상대적 위치를 추가 -> 학습한 데이터의 길이보다 **긴 문장의 데이터를 예측 가능**

  - **Absolute position**는 순차적으로 진행함에 따라 **encoding값들이 +1**이 됨 (0,1,2,3,...,n) 이런 식으로 -> 학습 데이터의 길이보다 **긴 문장의 길이는 예측이 불가능** (문장이 중간에서 짤림,BERT에서는 512개)

02 Related Works

GPT-1 (OpenAi)

- Pre-training model과 LLM이 시작점
  - OpenAi 에서 Transformer를 사용한 LLM을 고안
  - pre-training을 통해 학습한 모델을 fine-tuning하여 여러가지 task를 수행
  - GPT-1은 pre-training를 순방향 학습을 수행 (left-to-right)
  - GPT는 encoding부분 보다는 decoding부분에 집중한 모델
    -> left-to-right direction Decoder



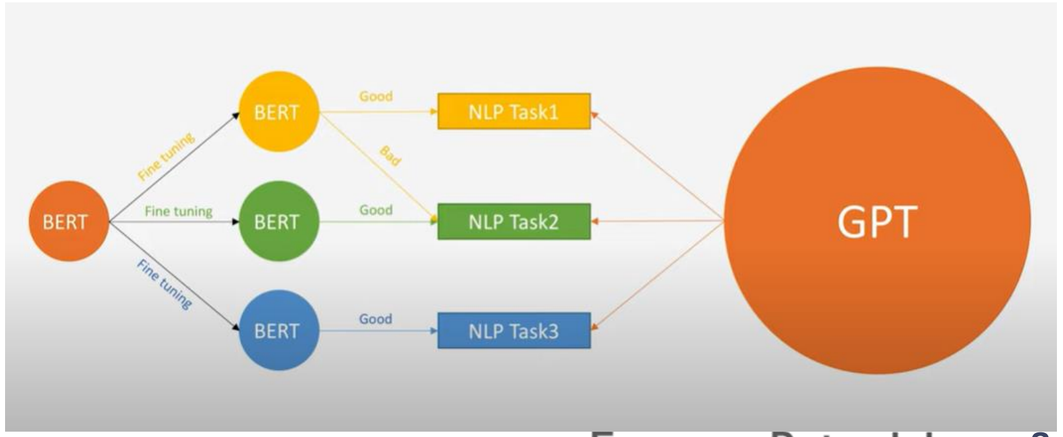
Ex)

I like nlp but, hate hard coding

-> I like nlp but, _____ (GPT-1)

I like nlp but, hate hard coding

-> I like nlp but, hate hard _____ (BERT(MLM))

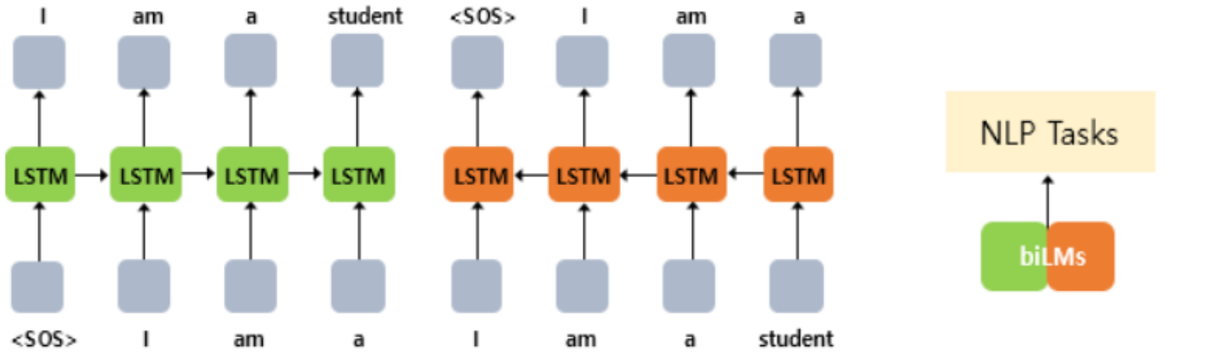


02 Related Works

Pre-training / Transfer Learning (Fine-tuning)

- ELMo

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

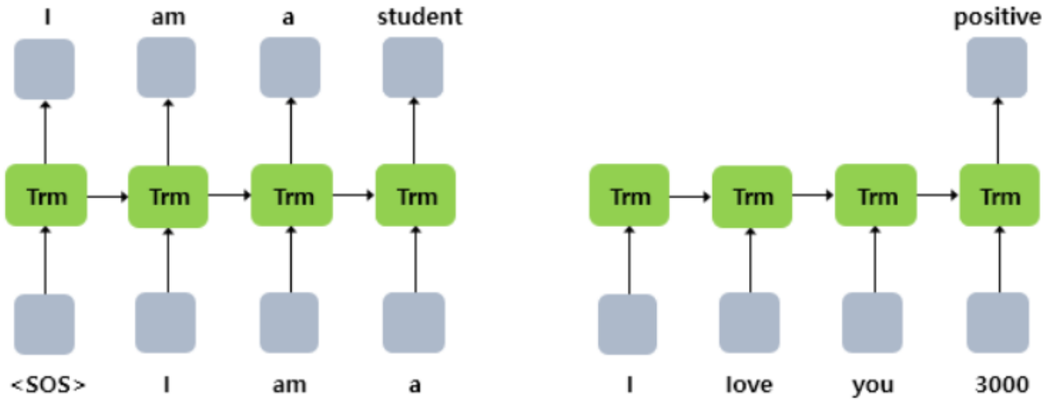


순방향 언어 모델과 역방향 언어 모델을 각각 훈련 사전 훈련된 임베딩에 사용

- **ELMo**에서는 **순방향(Forward LM),역방향(Backward LM)**을 각각 **LSTM**으로 학습 시킨 후 이렇게 학습 된 모델에서 임베딩 값들을 얻어 **다른 NLP Task**에 사용하는 방식으로 **transfer learning**을 진행

- GPT-1

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



Deep(12-layer) Trm 언어 모델을 사전 훈련 분류 문제에 파인 튜닝

- **ELMo**가 RNN 계열의 pre-training이었다면, **Transformer**를 사용한 **pre-training model GPT-1**이 OpenAI에서 고안됨 -> **GPT-1은 Transformer를 12개의 layer**로 쌓아 **순차적으로 (left-to-right)** 다음 단어를 예측

- 현재 NLP 분야는 **LLM**의 연구가 주를 이루고 있기 때문에 얼마나 많은 데이터들을 이용해 **large pre-trained model**을 만들고 이를 특정 task에 추가 학습 시켜 높은 성능을 얻는 트랜드

03 | Proposed Method

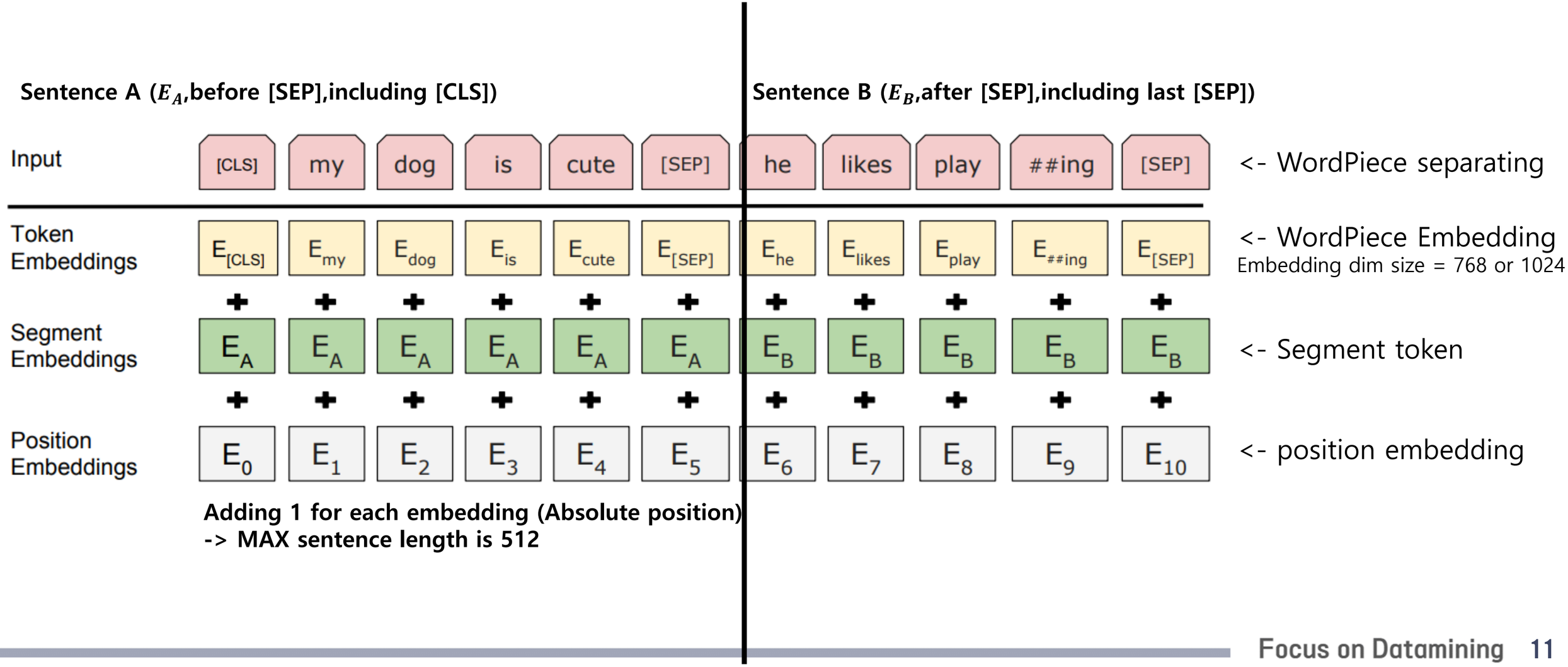
Input/Output Representations

- BERT는 **두개의 문장을 한번에 처리** 가능
  - both a single sentence, a pair of sentence를 다루기 쉽게 하기 위해 **하나의 sequence로 packing**후 input sequence로 사용
- BERT는 **WordPiece embedding**을 진행
  - **WordPiece**는 단어를 세부적으로 쪼갬 (character embedding보다 상세히)

  (ex, embeddings -> em+ ##bed + ##ding + ##s)
- BERT의 **special token**
  - input의 첫번째 토큰은 항상 **[CLS] token** (Final hidden state도 해당 토큰을 포함)
  - 두개의 문장을 구분하기 위해 **[SEP] token** 사용
  - 두 문장을 구분하기 위해 $\boldsymbol{E_A, E_B}$ **(sentence A, B)**를 추가적으로 embedding <- **segment token**
  - 각 단어들의 position을 학습하기 위해 **position embedding**을 수행 (**absolute position**)

03 Proposed Method

Input/Output Representations



03 | Proposed Method

Fine-tuning BERT

- **Pre-training** in BERT
  - BERT는 BooksCorpus (800M words)와 English Wikipedia (2500M words) (Wikipedia에선 text passage만 추출)에서 Corpus들을 추출하여서 pre-training
- BERT는 **self-attention mechanism** 사용
  - 2개의 문장을 **하나로 묶음** (bidirectional cross attention between two sentence)
- BERT에 input,output을 task에 **specific**하게 만들어 사용
  - fine-tuning을 통해 **parameter들을 end-to-end로 tuning**
- 여러가지 방식들로 **fine-tuning** 진행 후 여러 task 수행
  - sentence pairs in paraphrasing, hypothesis-premise pairs in entailment, question answering, text classification or sequence tagging

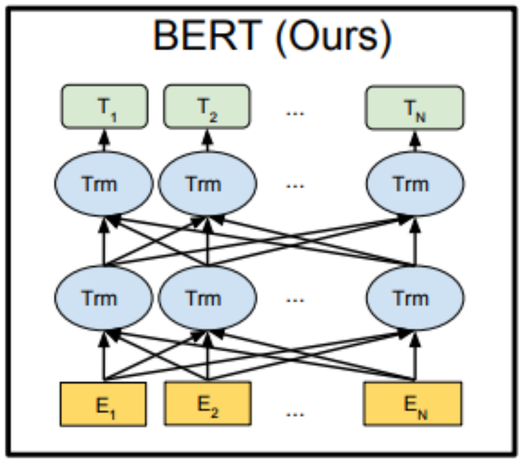
03 | Proposed Method

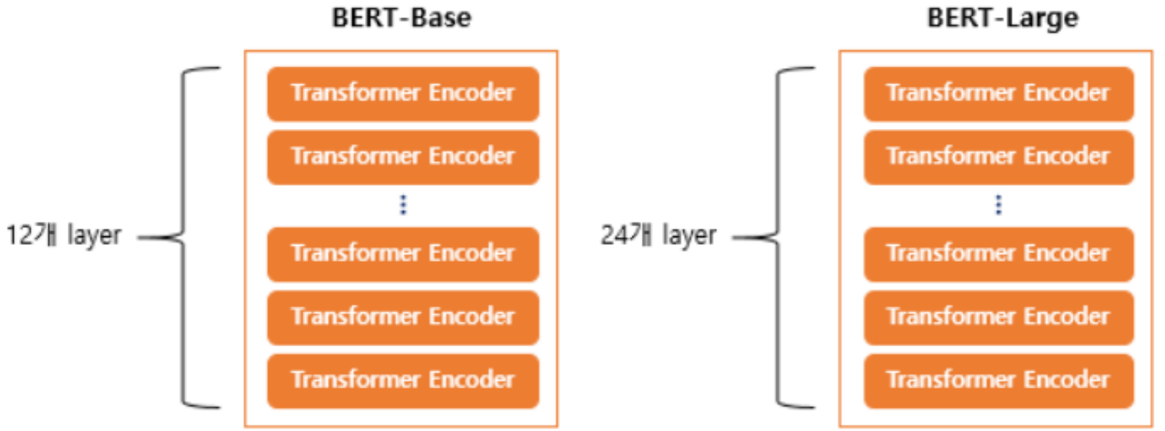
Model Architecture

- **BERT's Model Architecture**는 multi-layer bidirectional Transformer **encoder**

- $\mathbf{BERT_{BASE}}$: L = 12, H = 768, A = 12, D = 768, Total parameters = 110M
  -> GPT-1과 비교를 위해 parameter수를 조절한 모델 (**same size as GPT-1**)
- $\mathbf{BERT_{LARGE}}$: L = 24, H = 1024, A = 16, D = 1024, Total parameters = 340M
  -> **BERT의 성능을 최대화** 시킨 최적의 BERT model
  -> BERT의 성능이라 하면 LARGE의 성능을 의미

- L : the number of layers
- H : hidden size
- D : d_model size
- A : the number of Attention heads





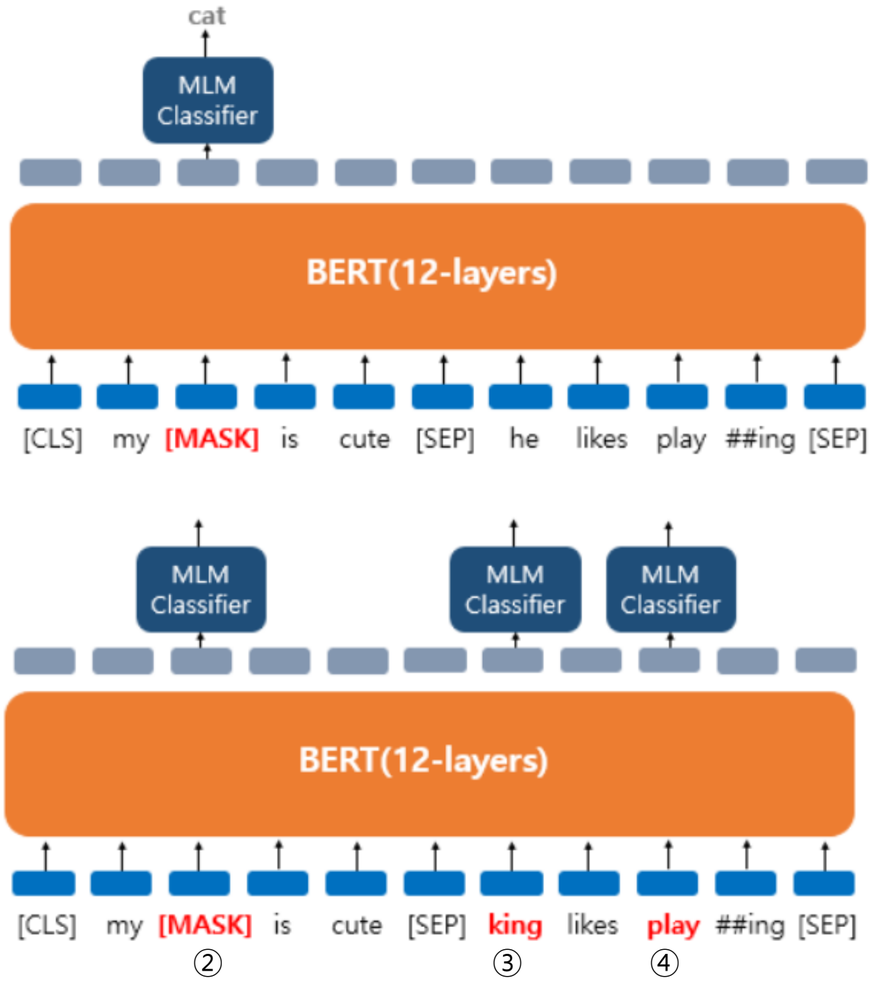
03 | Proposed Method

Masked Lanugage Model (MLM)

- Deep bidirectional model 은 left-to-right model 보다 훨씬 더 성능이 좋음

- 하지만 전통적인 language model에서는 **one-direction으로 학습** -> **“see itself”**를 방지하기 위함(“ see itself ” 는 논문에서 사용된 문장으로 문장에 대한 스포일러??같은 느낌)

- BERT는 **bidirectional model**을 사용하기 위해 Masked LM에 적용 – **Masked LM**이란, 문장 속 단어들을 **masking**후 해당 단어를 예측하는 LM



- **MLM mechanism**

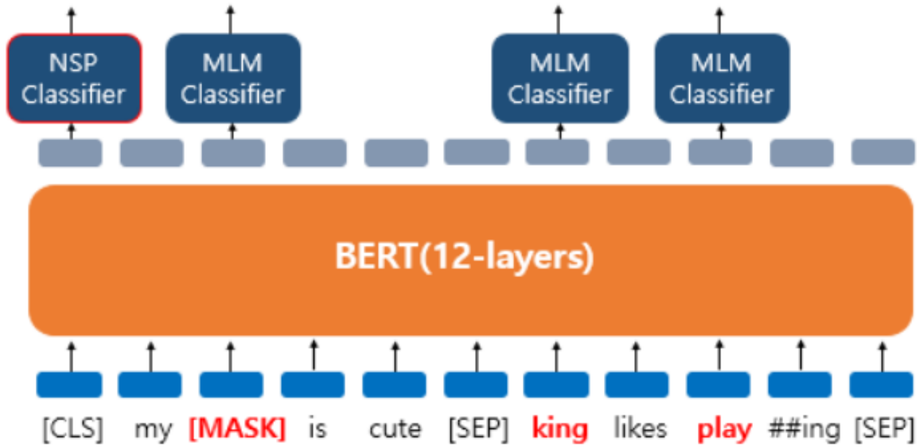
① Masking 15% words of sentence (기준은 WordPiece)
② 그 중 **80% masking words**를 실제로 **[MASK] token**을 부여
③ 그 중 **10% masking words**는 **다른 random token**을 부여
④ 그 중 **10% masking words**는 **original token**을 부여
Cross entropy loss를 통해 학습

03 | Proposed Method

Next Sentence Prediction (NSP)

- Main task in **NSP** (QA(Question Answering), NLI(Nature Language Inference))
  - **relationship between two sentences**
  - sentence A,B의 relation를 더 잘 찾기 위해 A->B 일 때, B의 정보를 조절하여 학습 진행
    -> B의 정보를 **이진화(binarized)** 시킴
    -> B의 **50%**를 **A와 이어지는 문장(IsNext),** 나머지 **50%**를 **A와 이어지지 않는 문장(NotNext)**으로 구성 후 A-B pair learning 진행
  - QA,NLI를 더 잘 수행하기 위해 위의 방법으로 NSP pre-training후 **Fine-Tuning** 거쳐 QA,NLI의 성능을 높임

- Pre-training in BERT
  - BERT use BooksCorpus (800M words) and English Wikipedia (2500M words) (Wikipedia에선 text passage만 추출)



03 | Proposed Method

Fine-tuning BERT on Different Tasks

a. Sentence Pair Classification

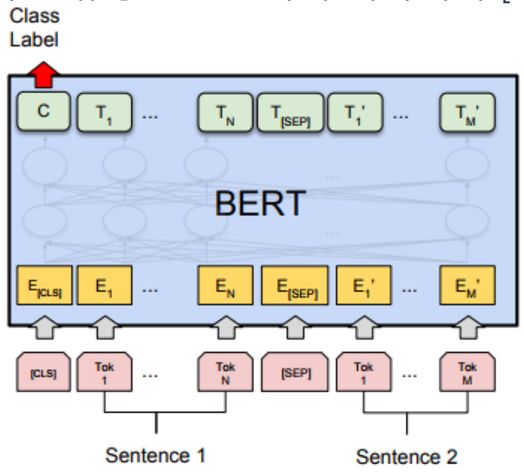
- **텍스트 쌍을 input**으로 받아 Task 수행 -> 주로 **NLI(자연어 추론) 수행**

자연어 추론이란, 한 쌍의 두 문장사이가 **어떤 관계인지를 분석** (관계에는 **모순 관계, 함의 관계, 중립 관계**가 있음)

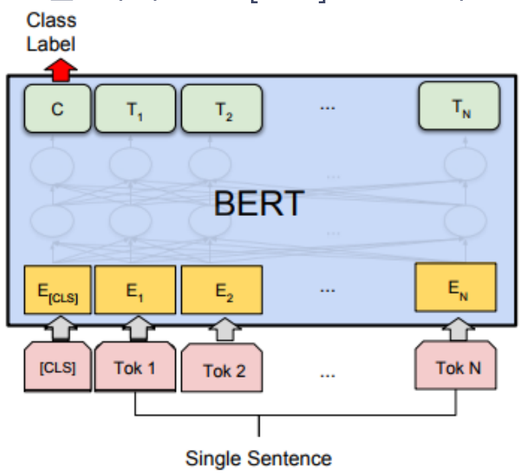
두 문장을 받아 Class Label의 출력값으로 **두 문장의 관계를 출력**

b. Single Sentence Classification Tasks

- **영화 리뷰 감성 분석** 등 입력된 문장에 대해서 분류를 하는 유형으로 문서의 시작에서 [CLS] token을 시작으로 [CLS] token의 출력층 위치에 Class Label 을 출력 (학습은 FC을 통해 수행)



(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG



(b) Single Sentence Classification Tasks: SST-2, CoLA

03 | Proposed Method

Fine-tuning BERT on Different Tasks

c. Question Answering Tasks

- **분문의 일부분을 추출하여 질문에 대한 답변**을 출력하는 방식

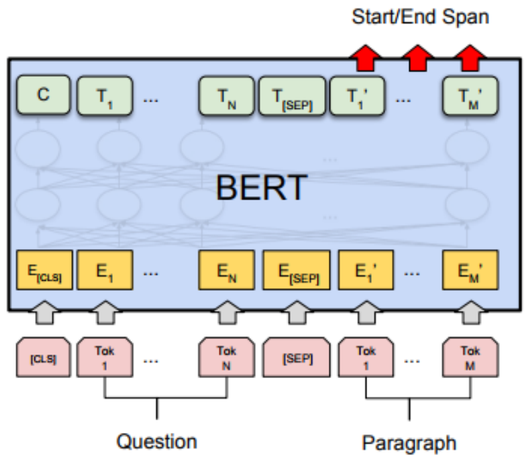
➔ 예로 들어서,

Q : "강우가 떨어지도록 영향을 주는 것은?",

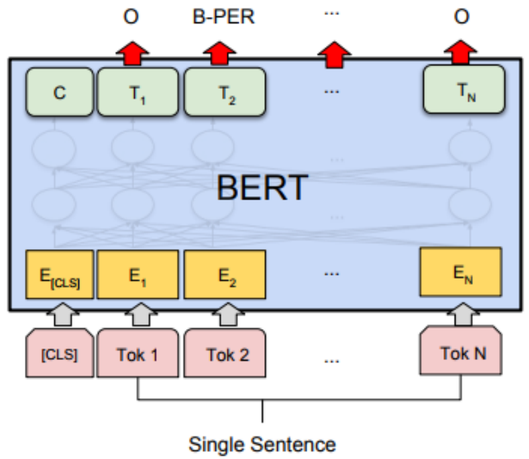
Paragraph : "기상학에서 강우는 대기 수증기가 응결되어 중력의 영향을 받고 떨어지는 것을 의미합니다. 강우의 주요 형태는 이슬비,비,진눈깨비,눈,씨락눈 및 우박이 있습니다." 라는 문장을 받았다면 **질문에 대한 정답으로는 "중력"**을 출력하는 모델

d. Single Sentence Tagging Tasks

- 하나의 문자에 대한 **태깅 작업**. 대표적으로 문장 속 각 단어들의 **품사 태깅** 작업 및 **개체명 인식** 작업으로 각각 단어들의 출력층에 **해당 단어의 품사등이 출력**되도록 수행 됨



(c) Question Answering Tasks: SQuAD v1.1



(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

04 | Experiment

VS GPT - 1

In GLUE score

GLUE : General Language Understanding Evaluation

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$BERT_{BASE}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$BERT_{LARGE}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

In SQuAD v2.0 score

System	Dev	Test
ESIM+GloVe	51.9	52.7
ESIM+ELMo	59.1	59.2
OpenAI GPT	-	78.0
$BERT_{BASE}$	81.6	-
$BERT_{LARGE}$	**86.6**	**86.3**
Human (expert)†	-	85.0
Human (5 annotations)†	-	88.0

BERT IS ! BEST !

SQuAD : The Stanford Question Answering Dataset

In SQuAD v1.1 score

System	Dev EM	Dev F1	Test EM	Test F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	-	-	82.3	91.2
#1 Ensemble - nlnet	-	-	86.0	91.7
#2 Ensemble - QANet	-	-	84.5	90.5
Published				
BiDAF+ELMo (Single)	-	85.6	-	85.8
R.M. Reader (Ensemble)	81.2	87.9	82.3	88.5
Ours				
$BERT_{BASE}$ (Single)	80.8	88.5	-	-
$BERT_{LARGE}$ (Single)	84.1	90.9	-	-
$BERT_{LARGE}$ (Ensemble)	85.8	91.8	-	-
$BERT_{LARGE}$ (Sgl.+TriviaQA)	**84.2**	**91.1**	**85.1**	**91.8**
$BERT_{LARGE}$ (Ens.+TriviaQA)	**86.2**	**92.2**	**87.4**	**93.2**

Table 2: SQuAD 1.1 results. The BERT ensemble is 7x systems which use different pre-training checkpoints and fine-tuning seeds.

System	Dev EM	Dev F1	Test EM	Test F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	86.3	89.0	86.9	89.5
#1 Single - MIR-MRC (F-Net)	-	-	74.8	78.0
#2 Single - nlnet	-	-	74.2	77.1
Published				
unet (Ensemble)	-	-	71.4	74.9
SLQA+ (Single)	-		71.4	74.4
Ours				
$BERT_{LARGE}$ (Single)	78.7	81.9	80.0	83.1

Table 3: SQuAD 2.0 results. We exclude entries that use BERT as one of their components.

05 Conclusion

BERT is POWERFUL

- GPT에 이은 Transformer 기반의 LLM (Large Lanugage Model)
- Unsupervised pre-training의 중요성을 증명
- Low-resource에서의 deep unidirectional architecture 학습 (Transfer Learning의 좋은 성능)
- Finding deep bidirectional architectures 그리고 allowing pre-trained model to NLP tasks

Q & A

감사합니다