BERT

Bidirectional Encoder Representations From Transformers

ICLAB 김종현

BERT

BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding)

[Abstract]

Abstract

We introduce a new language representation model called **BERT**, which stands for **B**idirectional **E**ncoder **R**epresentations from **T**ransformers. Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications.

- BERT는 Unlabled Text에서 모든 Layer의 좌우 Token을 모두 반영하는 양방향(Bidirectional) Pre-Trained Model
- Pre-Trained BERT Model에 하나의 Output Layer만 추가하여도 State-of-the-art(SOTA) Model 제작 가능

BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding)

[Introduction]

1 Introduction

Language model pre-training has been shown to be effective for improving many natural language processing tasks (Dai and Le, 2015; Peters et al., 2018a; Radford et al., 2018; Howard and Ruder, 2018). These include sentence-level tasks such as natural language inference (Bowman et al., 2015; Williams et al., 2018) and paraphrasing (Dolan and Brockett, 2005), which aim to predict the relationships between sentences by analyzing them holistically, as well as token-level tasks such as named entity recognition and question answering, where models are required to produce fine-grained output at the token level (Tjong Kim Sang and De Meulder, 2003; Rajpurkar et al., 2016).

- 많은 NLP Task에서 Language model pre-training의 탁월한 효과는 알려져 있음
- 이러한 NLP Task는 Token-Level Task인 Named Entity Recognition(NER)에서부터 Question Answering까지 광범위하게 적용

BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding)

[Introduction]

There are two existing strategies for applying pre-trained language representations to downstream tasks: *feature-based* and *fine-tuning*. The feature-based approach, such as ELMo (Peters et al., 2018a), uses task-specific architectures that include the pre-trained representations as additional features. The fine-tuning approach, such as the Generative Pre-trained Transformer (OpenAI GPT) (Radford et al., 2018), introduces minimal task-specific parameters, and is trained on the downstream tasks by simply fine-tuning *all* pre-trained parameters. The two approaches share the same objective function during pre-training, where they use unidirectional language models to learn general language representations.

- Pre-Trained된 언어 표현을 적용시키는 방법에는 두 가지가 존재. (feature-base, fine-tuning)
- Feature-based 방식은 ELMo에서, Fine-tuning 방식은 Transformer와 OpenAI의 GPT에서 사용
- 두 방식 모두 동일한 Objective Function으로 pre-train 하지만, 모두 단방향(unidirectional) Language Model을 사용해 학습

* Feature-based: 특정 Task를 수행하는 Network에 Pre-Trained Language Representation을 추가 Feature로 제공 (Fixed Parameters)

* Fine-Tuning: 각 Task를 수행할 때 Pre-Trained된 Parameter들을 사용 (Non-fixed Parameters)

BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding)

[Introduction]

We argue that current techniques restrict the power of the pre-trained representations, especially for the fine-tuning approaches. The major limitation is that standard language models are unidirectional, and this limits the choice of architectures that can be used during pre-training. For example, in OpenAI GPT, the authors use a left-to-right architecture, where every token can only attend to previous tokens in the self-attention layers of the Transformer (Vaswani et al., 2017). Such restrictions are sub-optimal for sentence-level tasks, and could be very harmful when applying fine-tuning based approaches to token-level tasks such as question answering, where it is crucial to incorporate context from both directions.

- 기존의 Model(ELMo, GPT)들은 성능 저해 요소가 있다고 주장
- Unidirectional Model으로 학습 시 Pre-training의 architecture 선택에 제한이 있음
- 예로는 OpenAI GPT는 Token 참고 시 왼쪽에서 오른쪽으로만 참고가 가능함

BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding)

[Introduction]

In this paper, we improve the fine-tuning based approaches by proposing BERT: **B**idirectional **E**ncoder **R**epresentations from **T**ransformers. BERT alleviates the previously mentioned unidirectionality constraint by using a "masked language model" (MLM) pre-training objective, inspired by the Cloze task (Taylor, 1953). The masked language model randomly masks some of the tokens from the input, and the objective is to predict the original vocabulary id of the masked word based only on its context. Unlike left-to-right language model pre-training, the MLM objective enables the representation to fuse the left and the right context, which allows us to pre-train a deep bidirectional Transformer. In addition to the masked language model, we also use a "next sentence prediction" task that jointly pre-trains text-pair representations. The contributions of our paper are as follows:

- BERT는 기존 모델들의 Unidirection을 해결하기 위해 "masked language model(MLM)"를 사용해 pre-train
- MLM은 임의로 input token을 masking하고, 문맥기반만으로 masked token의 original vocab 예측이 목표
- 기존 모델들과 달리 MLM의 목적은 좌, 우 모든 문맥이 융화되도록 하는 것
- MLM과 더불어 "next sentence prediction" task를 함께 사용

BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding)

[Introduction]

- We demonstrate the importance of bidirectional pre-training for language representations. Unlike Radford et al. (2018), which uses unidirectional language models for pre-training, BERT uses masked language models to enable pre-trained deep bidirectional representations. This is also in contrast to Peters et al. (2018a), which uses a shallow concatenation of independently trained left-to-right and right-to-left LMs.
- We show that pre-trained representations reduce the need for many heavily-engineered task-specific architectures. BERT is the first fine-tuning based representation model that achieves state-of-the-art performance on a large suite of sentence-level *and* token-level tasks, outperforming many task-specific architectures.
- BERT advances the state of the art for eleven NLP tasks. The code and pre-trained models are available at `https://github.com/google-research/bert`.

- Bidrectional과 Unidirectional을 비교하는 학습을 진행
- 사전 학습을 통해 많은 engineering이 필요한 task-specific architectures 사용 필요성 감소를 도출
- Sentence, Token 단위 모두 SOTA 성능을 보이는 최초의 fine-tuning based 표현 모델
- BERT는 11개의 NLP task들에서 SOTA의 성능을 향상시킴

BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding)

[BERT - Model Architecture]

Model Architecture BERT's model architecture is a multi-layer bidirectional Transformer encoder based on the original implementation described in Vaswani et al. (2017) and released in the `tensor2tensor` library.[1] Because the use of Transformers has become common and our implementation is almost identical to the original, we will omit an exhaustive background description of the model architecture and refer readers to Vaswani et al. (2017) as well as excellent guides such as "The Annotated Transformer."[2]

In this work, we denote the number of layers (i.e., Transformer blocks) as L, the hidden size as H, and the number of self-attention heads as A.[3] We primarily report results on two model sizes: $\textbf{BERT}_{\textbf{BASE}}$ (L=12, H=768, A=12, Total Parameters=110M) and $\textbf{BERT}_{\textbf{LARGE}}$ (L=24, H=1024, A=16, Total Parameters=340M).

$\text{BERT}_{\text{BASE}}$ was chosen to have the same model size as OpenAI GPT for comparison purposes. Critically, however, the BERT Transformer uses bidirectional self-attention, while the GPT Transformer uses constrained self-attention where every token can only attend to context to its left.[4]

- BERT는 Transformer의 Encoder만 사용
- BERT-Base Model은 num_layers = 12, hidden_dim = 768, num_heads = 12. Total Parameter는 1억 1000만개
- BERT-Base Model은 OpenAI GPT와 동일 개수의 Parameter을 가짐

→ Hyper Parameter 개수가 동일하더라도 pre-training concept 변화만으로도 높은 성능을 낼 수 있음을 보여주기 위해

- BERT-Large Model은 num_layers = 24, hidden_dim = 1024, num_heads = 16. Total Parameter는 3억 4000만개

BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding)

[BERT - Input / Output Representations]

Input/Output Representations To make BERT handle a variety of down-stream tasks, our input representation is able to unambiguously represent both a single sentence and a pair of sentences (e.g., ⟨ Question, Answer ⟩) in one token sequence. Throughout this work, a "sentence" can be an arbitrary span of contiguous text, rather than an actual linguistic sentence. A "sequence" refers to the input token sequence to BERT, which may be a single sentence or two sentences packed together.

- BERT는 다양한 down-stream task 수행을 위해 유연한 representation 허용
- Single Sentence만을 받을 수 있고, Pair of Sentences (ex: Question-Answer)도 받을 수 있음

논문에서의 Sentence와 Sequence 정의

* Sentence: 연속적인 text의 span(나열). 우리가 알고 있는 문장이 아니어도 됨(문장 전체일 수도 있고 문장의 일부일 수도 있음)

* Sequence: BERT의 input token. Single sentence 또는 two sentences packed together. 이 때 sentence는 우리가 알고 있는 sentence가 아닌 BERT에서 정의하는 sentence의 의미

BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding)

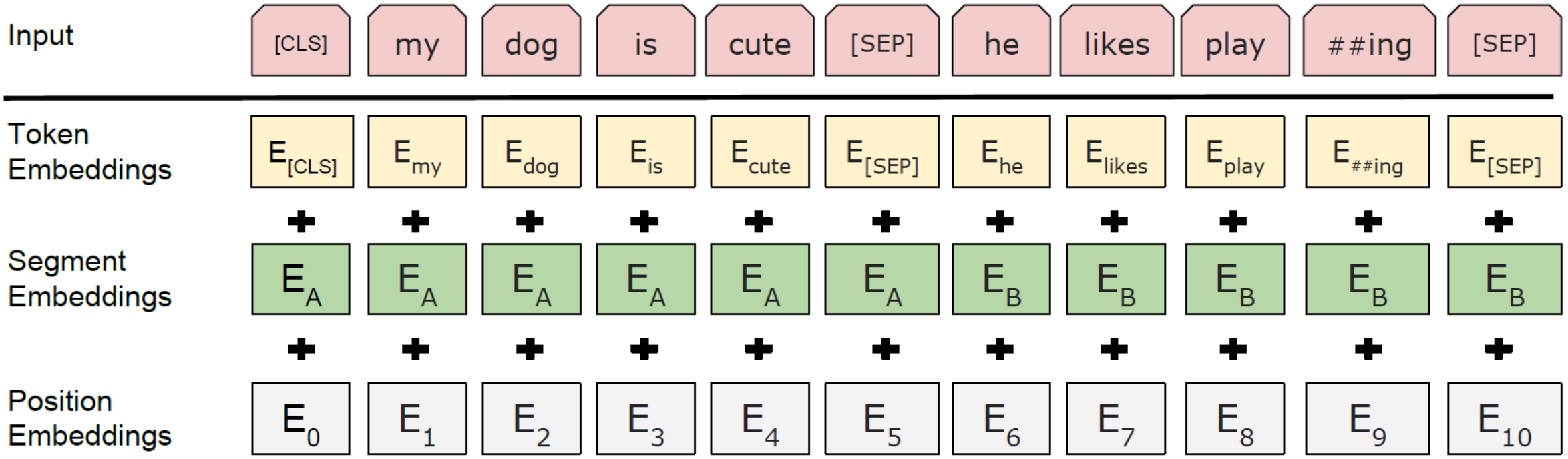
[BERT - Input / Output Representations]

We use WordPiece embeddings (Wu et al., 2016) with a 30,000 token vocabulary. The first token of every sequence is always a special classification token (`[CLS]`). The final hidden state corresponding to this token is used as the aggregate sequence representation for classification tasks. Sentence pairs are packed together into a single sequence. We differentiate the sentences in two ways. First, we separate them with a special token (`[SEP]`). Second, we add a learned embedding to every token indicating whether it belongs to sentence `A` or sentence `B`. As shown in Figure 1, we denote input embedding as E, the final hidden vector of the special `[CLS]` token as $C \in \mathbb{R}^H$, and the final hidden vector for the i^{th} input token as $T_i \in \mathbb{R}^H$.

- 3만개의 token vocabulary를 가지는 WordPiece Embedding 사용. Split word의 경우 ##으로 표현
- 모든 sequence의 First token은 special-classification token인 [CLS]를 넣어줌
- Sentence Pair는 합쳐져서 Single Sequence로 입력됨. 각각의 Sentence는 여러 개의 sentence로 이루어 질 수 있음
- 두 가지 방법으로 문장 쌍을 구별
  1. 두 문장 사이에 [SEP] 토큰 삽입
  2. 학습된 문장 A의 Embedding 값을 앞쪽 문장 모든 token에 더해주고 문장 B의 Embedding 값을 뒤쪽 문장 모든 token에 더해줌
- input이 단일 문장일 경우 문장 A의 Embedding 값만 사용

BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding)

[BERT - Input / Output Representations]



For a given token, its input representation is constructed by summing the corresponding token, segment, and position embeddings. A visualization of this construction can be seen in Figure 2.

- Input Representation은 Embedding Token, Segment Embedding, Position Embedding의 합으로 표현

BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding)

[Pre-training BERT]

3.1 Pre-training BERT

Unlike Peters et al. (2018a) and Radford et al. (2018), we do not use traditional left-to-right or right-to-left language models to pre-train BERT. Instead, we pre-train BERT using two unsupervised tasks, described in this section. This step is presented in the left part of Figure 1.

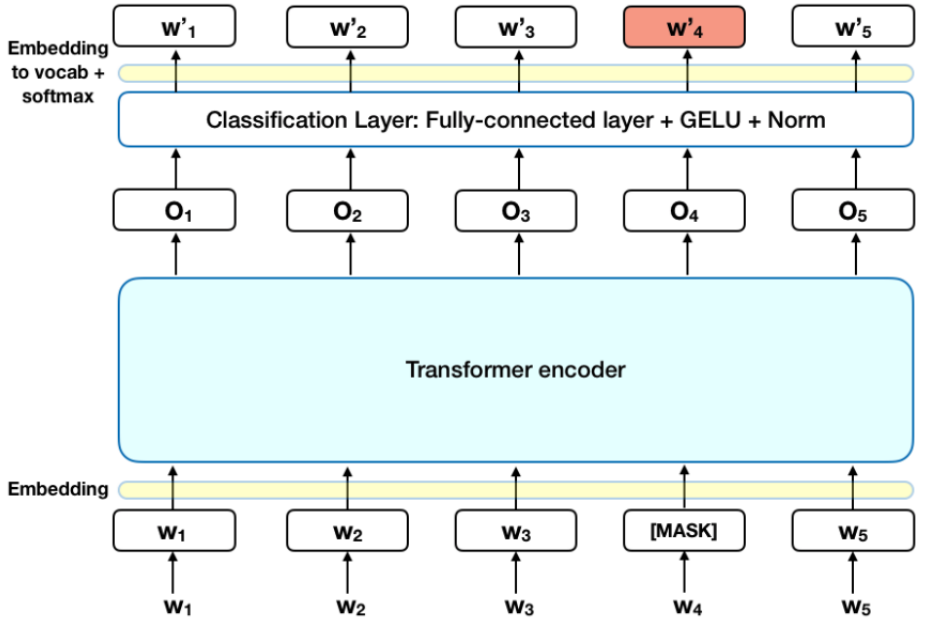
- 기존의 left-to-right나 right-to-left 방식이 아닌 2개의 unsupervised task로 pre-training 수행
- 1. Masked Language Model(MLM)
- 2. Next Sentence Prediction(NSP)

BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding)

[Pre-training BERT – Task #1: Masked Language Model (MLM)]

Task #1: Masked LM Intuitively, it is reasonable to believe that a deep bidirectional model is strictly more powerful than either a left-to-right model or the shallow concatenation of a left-to-right and a right-to-left model. Unfortunately, standard conditional language models can only be trained left-to-right *or* right-to-left, since bidirectional conditioning would allow each word to indirectly "see itself", and the model could trivially predict the target word in a multi-layered context.

- 일반적인 조건부 Language Model은 오직 left-to-right나 right-to-left 방식으로만 학습 가능
- Bidirectional Model은 각 단어가 간접적으로 스스로를 참조하도록 허용하고, multi-layered context의 target word를 세밀하게 예측가능



BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding)

[Pre-training BERT – Task #1: Masked Language Model (MLM)]

In order to train a deep bidirectional representation, we simply mask some percentage of the input tokens at random, and then predict those masked tokens. We refer to this procedure as a "masked LM" (MLM), although it is often referred to as a *Cloze* task in the literature (Taylor, 1953). In this case, the final hidden vectors corresponding to the mask tokens are fed into an output softmax over the vocabulary, as in a standard LM. In all of our experiments, we mask 15% of all WordPiece tokens in each sequence at random. In contrast to denoising auto-encoders (Vincent et al., 2008), we only predict the masked words rather than reconstructing the entire input.

- Bidirectional Representation의 train을 위해 input을 특정 percentage만큼 random masking하고, mask된 token을 예측
- 각 input sequence의 15%를 무작위로 정해 Mask를 씌움
- 전체 문장 중의 일부(15%)를 [Mask] token으로 변환해 예측에 사용
- 전체 input을 재구성 하지 않고 masked word만 예측

BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding)

[Pre-training BERT – Task #1: Masked Language Model (MLM)]

Although this allows us to obtain a bidirectional pre-trained model, a downside is that we are creating a mismatch between pre-training and fine-tuning, since the [MASK] token does not appear during fine-tuning. To mitigate this, we do not always replace "masked" words with the actual [MASK] token. The training data generator chooses 15% of the token positions at random for prediction. If the i-th token is chosen, we replace the i-th token with (1) the [MASK] token 80% of the time (2) a random token 10% of the time (3) the unchanged i-th token 10% of the time. Then, T_i will be used to predict the original token with cross entropy loss. We compare variations of this procedure in Appendix C.2.

- MLM을 적용함으로써 Bidirectional Training을 가능하게 하였지만, 문제점 존재
  - pre-training 과정에서는 "[MASK]" token을 사용하지만, fine-tuning에서는 사용하지 않아 pre-train과 fine-tuning간 간극 발생
    → masked word를 전부 [MASK] token으로 바꾸지 않고 일부만 변경하는 방법으로 해결
- Masked word의 80%는 [MASK] token으로 바꾸고, 10%는 token을 random word로 바꾸어 주고, 나머지 10%는 원래 단어 그대로 둠

Masked word 중 10%를 원본 단어로 두는 이유?

→ 90%의 Predicted Word에 대한 Original Word의 표상을 더해 줌으로써 성능을 높이기 위해

BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding)

[Pre-training BERT – Task #1: Masked Language Model (MLM)]

MLM Example

Input: My Dog's name is Polly and My Dog is hairy
Masking Target: My Dog's name is Polly and My Dog is hairy

(80%) My Dog is hairy → My Dog is [MASK]
(10%) My Dog is hairy → My Dog is apple
(10%) My Dog is hairy → My Dog is hairy

이 때 Masking 비율은 전체 문장의 15%, Masking Target의 10%를 random token으로 바꾸지만 전체 문장 비율로 보면 1.5%에 불과
→ Language understanding에 끼치는 영향력 미미

BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding)

[Pre-training BERT – Task #2: Next Sentence Prediction (NSP)]

Task #2: Next Sentence Prediction (NSP) Many important downstream tasks such as Question Answering (QA) and Natural Language Inference (NLI) are based on understanding the *relationship* between two sentences, which is not directly captured by language modeling. In order to train a model that understands sentence relationships, we pre-train for a binarized *next sentence prediction* task that can be trivially generated from any monolingual corpus. Specifically, when choosing the sentences `A` and `B` for each pre-training example, 50% of the time `B` is the actual next sentence that follows `A` (labeled as `IsNext`), and 50% of the time it is a random sentence from the corpus (labeled as `NotNext`). As we show in Figure 1, C is used for next sentence prediction (NSP).[5] Despite its simplicity, we demonstrate in Section 5.1 that pre-training towards this task is very beneficial to both QA and NLI. [6]

- Question Answering, Natural Language Inference등의 Task들은 두 문장 사이의 관계를 이해하는 것이 중요
  → Language Modeling에서는 관계 capture 불가
- BERT에서는 corpus(말뭉치)로 구성된 두 개의 문장에 대해 서로 관계가 있는지 없는지 분류하는 binary next sentence prediction task를 통해 pre-train
- 두 문장 A, B를 매 pre-training 마다 선택해 B가 A 바로 다음에 오는지 예측
- 학습을 위해 50%는 연결된 문장, 나머지 50%는 랜덤하게 뽑힌(관계가 없는) 문장으로 학습
- 문장이 연결된 경우 label = IsNext, 연결되지 않았다면 label = NotNext

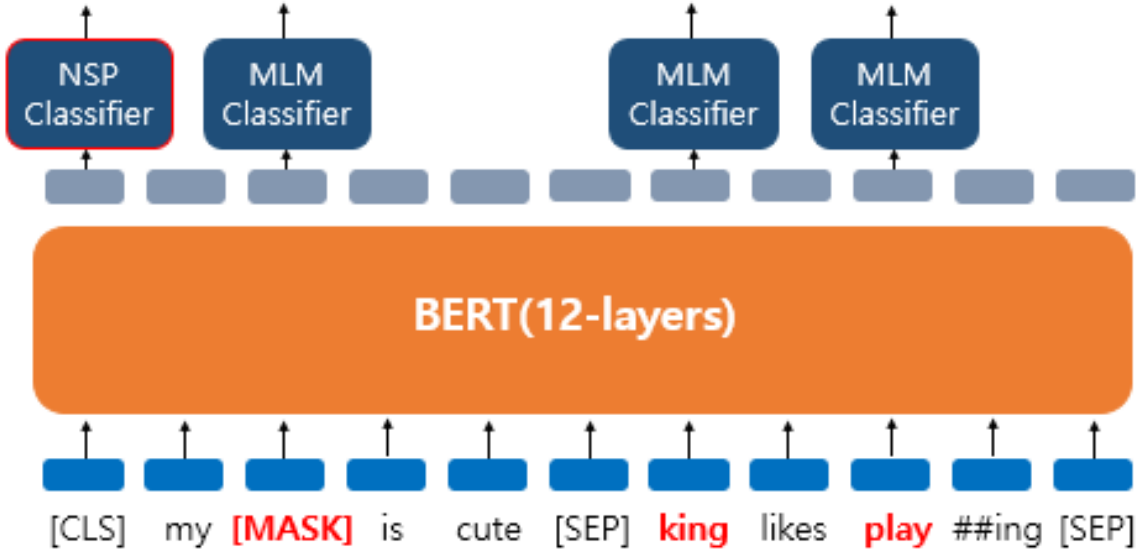
BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding)

[Pre-training BERT – Task #2: Next Sentence Prediction (NSP)]

NSP Example

1. Input = [CLS] the man went to [MASK] store [SEP] he bought a gallon [MASK] milk [SEP]
   → Label = IsNext

2. Input = [CLS] the man [MASK] to the store [SEP] penguin [MASK] are fight ##less birds [SEP]
   → Label = NotNext



1. 첫 번째 문장의 끝에 [SEP] Token 삽입
2. 두 번째 문장이 끝나면 [SEP] Token 삽입
3. 두 문장이 실제 이어지는 문장인지 아닌지를 첫 번째 문장의 [CLS] Token에서 binary-classification

BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding)

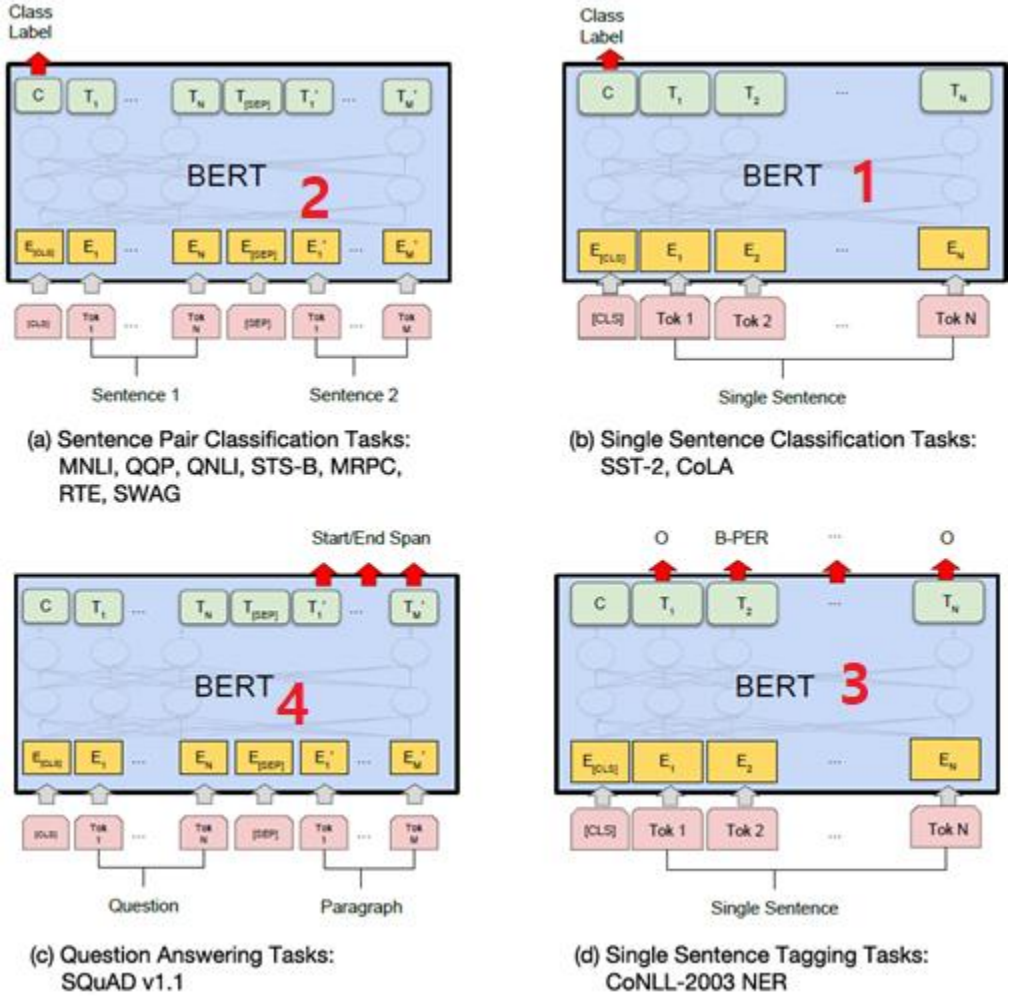
[Pre-training BERT – Pre-training Data]

Pre-training data The pre-training procedure largely follows the existing literature on language model pre-training. For the pre-training corpus we use the BooksCorpus (800M words) (Zhu et al., 2015) and English Wikipedia (2,500M words). For Wikipedia we extract only the text passages and ignore lists, tables, and headers. It is critical to use a document-level corpus rather than a shuffled sentence-level corpus such as the Billion Word Benchmark (Chelba et al., 2013) in order to extract long contiguous sequences.

- BookCorpus(800M words), Wikipedia(2,500M words)를 사용해 pre-trained
- Wikipedia Data에서는 lists, tables, headers를 모두 무시하고 text passages만 추출해 사용
  → long contiguous sequence만 학습시키기 위해

BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding)

[Fine-tuning BERT]



- BERT 논문에서는 4가지의 downstream task를 설명

1. Single Sentence Classification
   → 문장 하나가 주어졌을 때 어떠한 Label인지 예측

2. Sentence Pair Classification
   → 문장 두 개가 주어졌을 때, Label을 예측. 두 문장이 주어졌을 때 서로의 주장을 보완(entailment)하는지, 상충(contradiction)하는지, 중립(neutral)인지 예측

3. Single Sentence Tagging Tasks
   → 한 문장 내 들어있는 단어에 대한 레이블을 예측하는 문제. 대표적 예로는 개체명 인식(NER)

4. Question & Answering
   → BERT에게 질문과 본문이 주어졌을 때, 본문 속에 답이 있는 부분을 예측