BERT:Pre-training of Deep Bidirectional Transformers for Language Understanding

2020-01-21

백서인

Abstract

- BERT stands for **B**idirectional **E**ncoder **R**epresentations from **T**ransformers
- BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers
- 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
- BERT is conceptually simple and empirically powerful
- It obtains new state-of-the-art results on eleven natural language processing tasks

Introduction

Two existing strategies for applying pre-trained language representations to down-stream tasks

(1) Feature-based

- Ex) ELMo
- Uses task-specific architectures that include the pre-trained representations as additional features

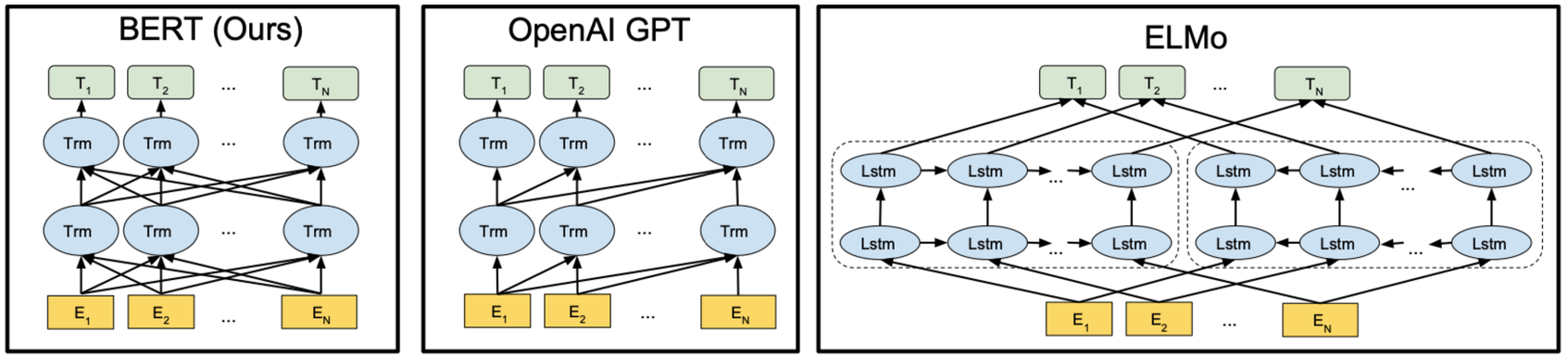
(2) Fine-tuning

- Ex) Generative Pre-trained Transformer (OpenAI GPT)
- 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

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



Introduction

- BERT alleviates the previously mentioned unidirectionality constraint by using a “masked language model” (MLM) pre-training objective
- 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
- The MLM objective enables the representation to fuse the left and the right context
- We also use a “next sentence prediction” (NSP) task that jointly pre-trains text-pair representation

BERT

There are two steps in our framework

: *pre-training* and *fine-tuning*

(1) During pre-training, the model is trained on unlabeled data over different pre-training task

(2) For fine-tuning, the BERT model is first initialized with the pre-trained parameters, and all of the parameters are fine-tuned using labeled data from the downstream tasks

BERT

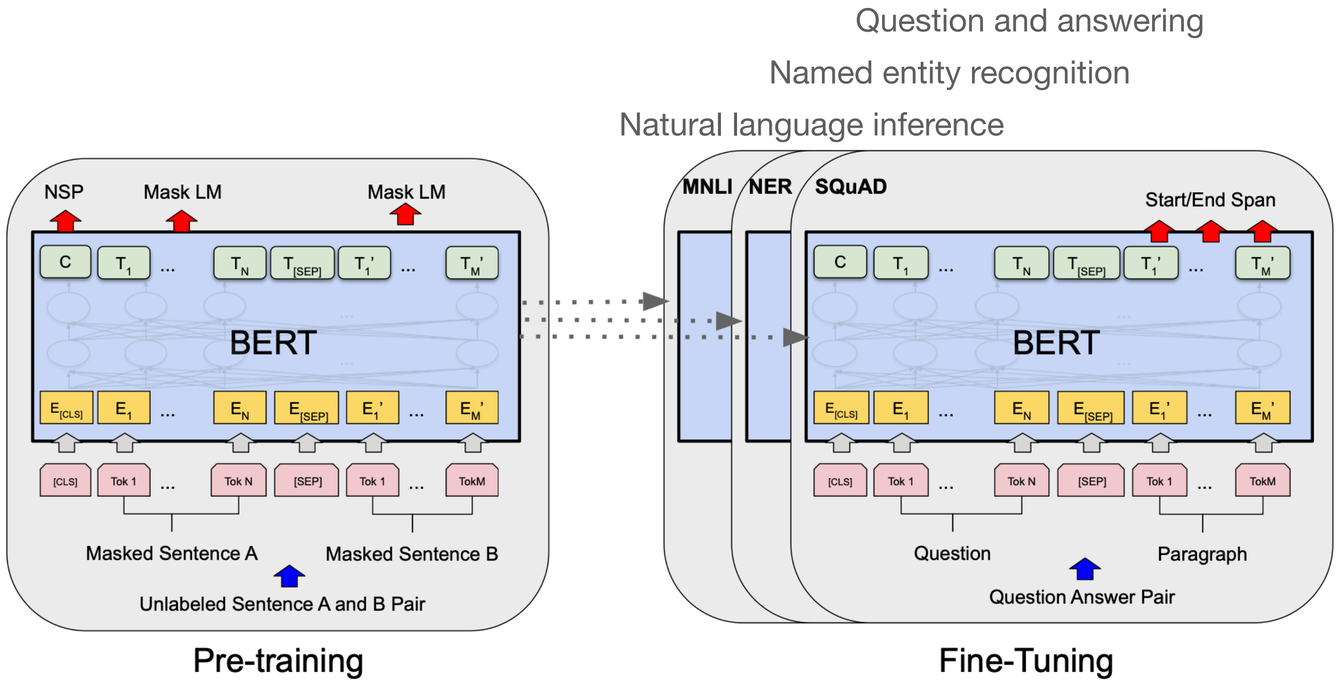


Figure 1: Overall pre-training and fine-tuning procedures for BERT. Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different down-stream tasks. During fine-tuning, all parameters are fine-tuned. [CLS] is a special symbol added in front of every input example, and [SEP] is a special separator token (e.g. separating questions/answers).

Output layer를 제외하고는 pre-training 과 fine-tuning 은 same architecture을 갖고 있음

fine-tuning할 때는 모든 parameters들을 fine-tuned

[CLS] 는 모든 input example 맨 앞에 추가하는 special symbol

[SEP] 는 separator token (e.g. separating questions/answers)

Model Architecture

- 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

- In this work, we denote
  - L: number of layers (i.e., Transformers blocks)
  - H: hidden size
  - A: number of self-attention heads

- $\text{BERT}_{\text{BASE}}$ (L=12, H=768, A=12, Total Parameters=110M)

- $\text{BERT}_{\text{LARGE}}$ (L=24, H=1024, A=16, Total Parameters=340M)

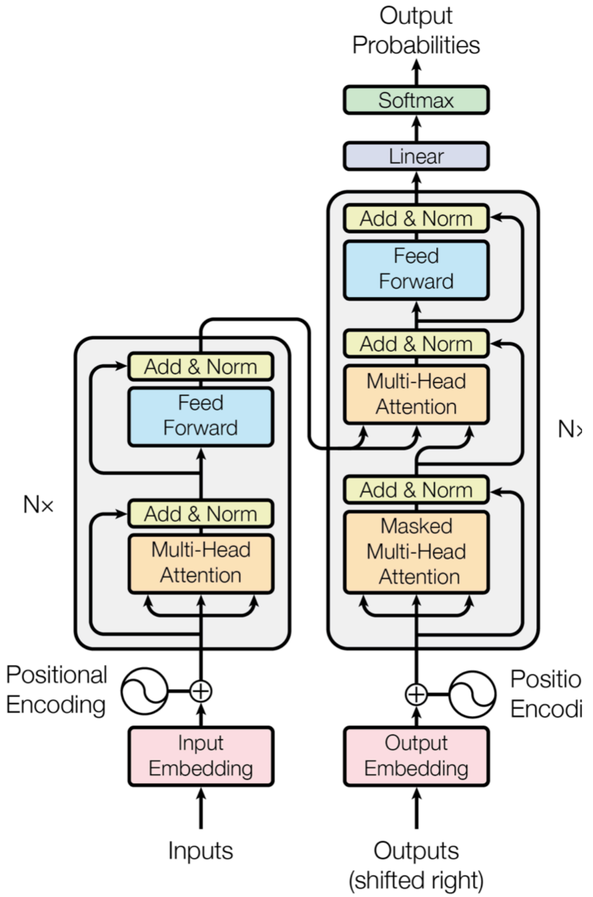


Figure 1: The Transformer - model architecture.

Input/Output Representations

- Our input representation is able to unambiguously represent both a single sentence and a pair of sentences (e.g., <Question, Answer>)
- For a given token, its input representation constructed by summing the corresponding token segment, and position embeddings

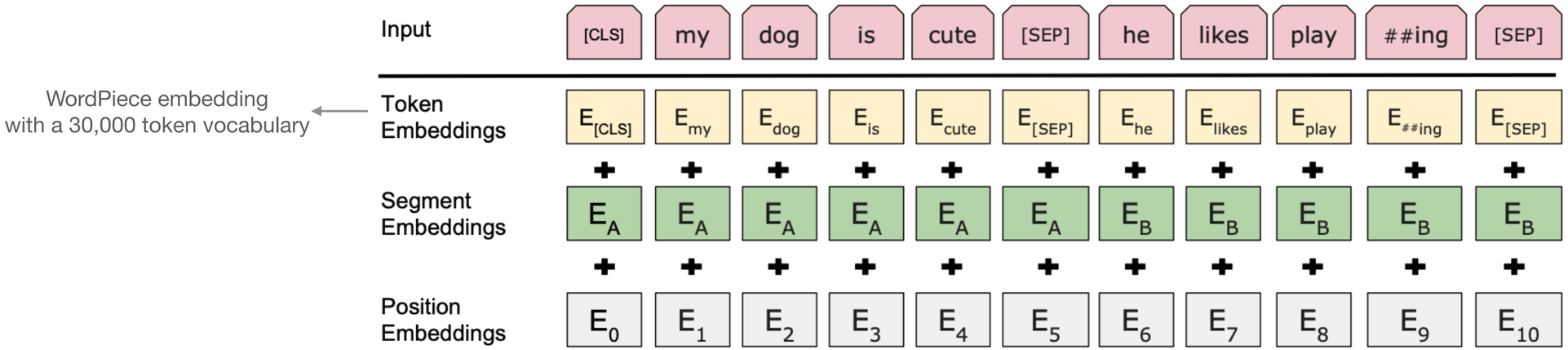


Figure 2: BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.

Pre-training BERT

- We pre-train BERT using two unsupervised tasks

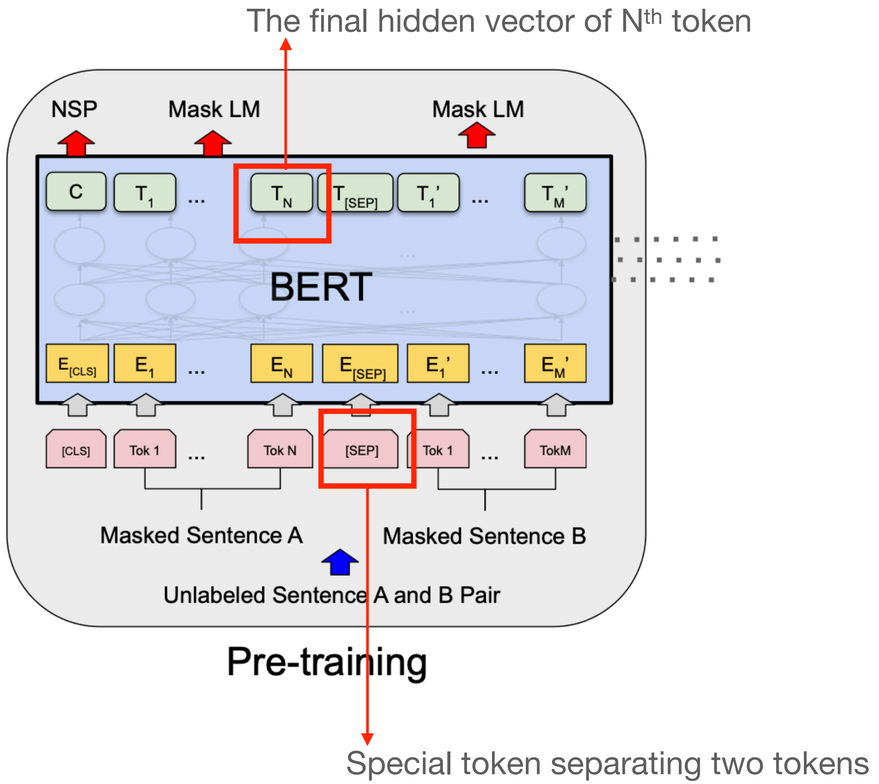
- Task #1: Masked LM (MLM)
  - In order to train a deep bidirectional representation, we simply mask some percentage of the input tokens at random, and then predict those masked tokens
  - In all of our experiments, we mask 15% of all WordPiece tokens in each sequence at random
  - A downside is that we are creating mismatch between pre-training and fine-tuning, since [MASK] token does not appear during fine-tuning
  - To mitigate this, we do not always replace “masked” words with the actual [MASK] token
    - The [MASK] token 80% of the time
    - A random token 10% of the time
    - The unchanged i-th token 10% of the time

Pre-training BERT

- We pre-train BERT using two unsupervised tasks

- Task #1: Masked LM (MLM)

  - 80% of the time: Replace the word with the `[MASK]` token, e.g., `my dog is hairy` → `my dog is [MASK]`
  - 10% of the time: Replace the word with a random word, e.g., `my dog is hairy` → `my dog is apple`
  - 10% of the time: Keep the word unchanged, e.g., `my dog is hairy` → `my dog is hairy`. The purpose of this is to bias the representation towards the actual observed word.



Pre-training BERT

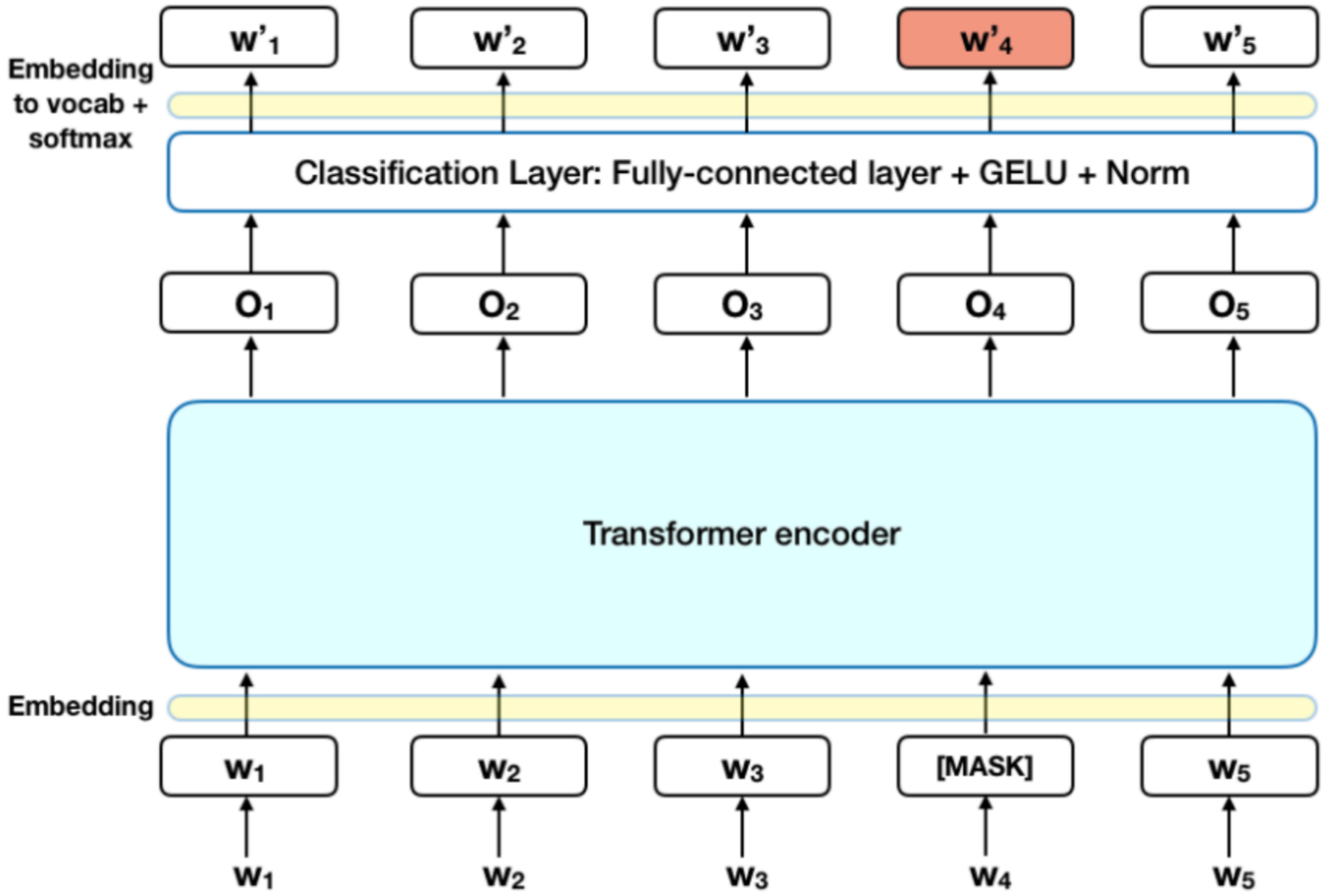
- We pre-train BERT using two unsupervised tasks

- Task #1: Masked LM (MLM)

Masking Rates			Dev Set Results		
MASK	SAME	RND	MNLI	NER	
			Fine-tune	Fine-tune	Feature-based
80%	10%	10%	84.2	95.4	94.9
100%	0%	0%	84.3	94.9	94.0
80%	0%	20%	84.1	95.2	94.6
80%	20%	0%	84.4	95.2	94.7
0%	20%	80%	83.7	94.8	94.6
0%	0%	100%	83.6	94.9	94.6

Table 8: Ablation over different masking strategies.

Pre-training BERT



Pre-training BERT

- We pre-train BERT using two unsupervised tasks

- Task #2: Next Sentence Prediction (NSP)
  - 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 follow A (labeled as IsNext), and 50% of the time it is a random sentence from corpus (labeled as NotNext)

Pre-training BERT

- We pre-train BERT using two unsupervised tasks

- Task #2: Next Sentence Prediction (NSP)

$$\text{Input} = \texttt{[CLS] the man went to [MASK] store [SEP]}$$
$$\texttt{he bought a gallon [MASK] milk [SEP]}$$
$$\text{Label} = \texttt{IsNext}$$

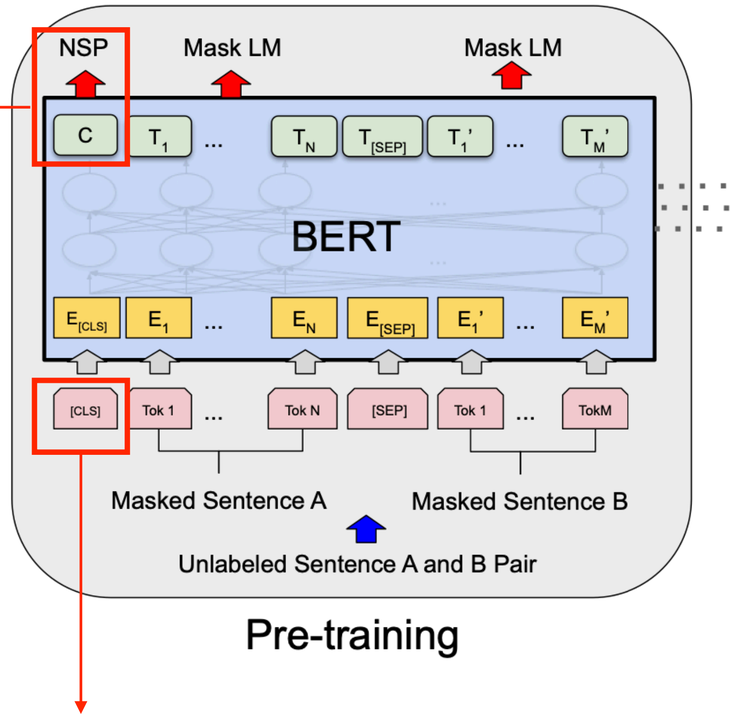
$$\text{Input} = \texttt{[CLS] the man [MASK] to the store [SEP]}$$
$$\texttt{penguin [MASK] are flight ##less birds [SEP]}$$
$$\text{Label} = \texttt{NotNext}$$

Pre-training BERT

- We pre-train BERT using two unsupervised tasks

- Task #2: Next Sentence Prediction (NSP)

The final hidden vector of special [CLS] token



The first token of every sequence

C is used for next sentence prediction (NSP)

The vector C is not a meaningful sentence representation without fine-tuning, since it was trained with NSP

Fine-tuning BERT

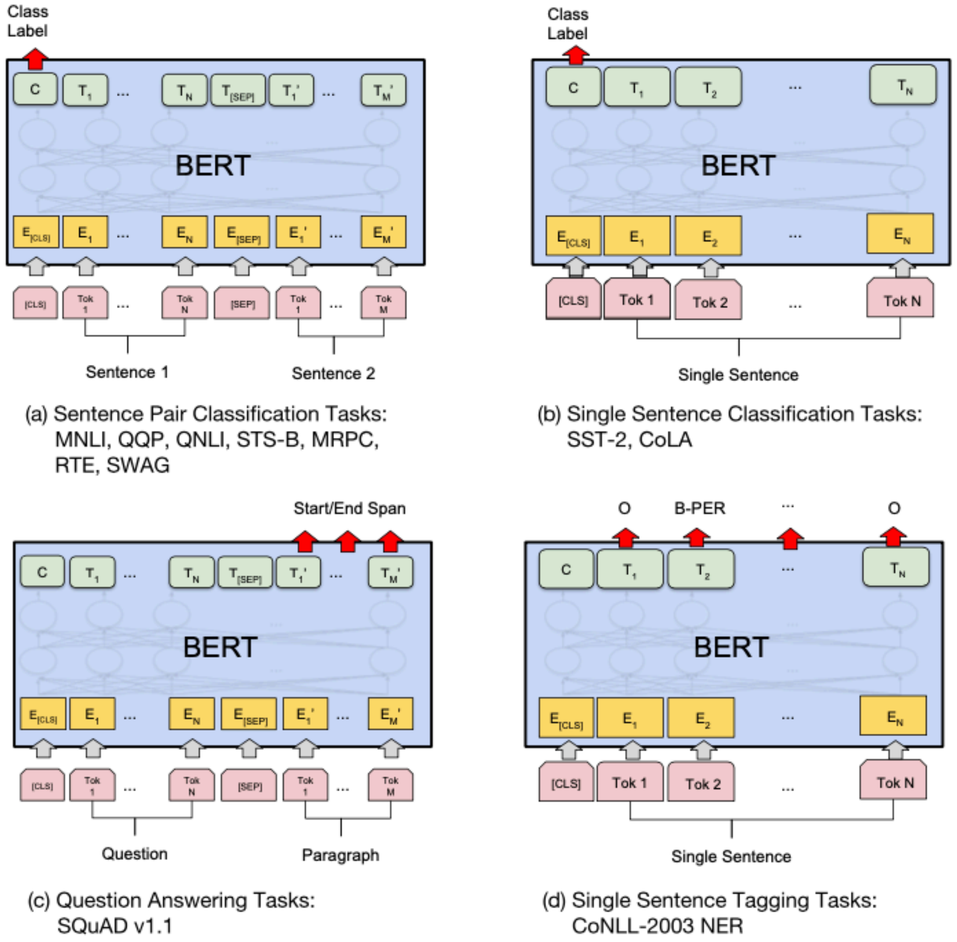
- Downstream tasks — whether they involve single text or text pairs
- For applications involving text pairs, a common pattern is to independently encode text pairs before applying bidirectional cross attention
- BERT instead uses the self-attention mechanism to unify these two stages, as encoding a concatenated text pair with self-attention effectively includes bidirectional cross attention between two sentences
- For each task, we simply plug in the task-specific inputs and outputs into BERT and fine-tune all the parameters end-to-end

Fine-tuning BERT

- At the input, sentence A and sentence B from pre-training are analogous to
  - (1) sentence pairs in paraphrasing
  - (2) hypothesis-premise pairs in entailment
  - (3) question-passage pairs in question answering
  - (4) a degenerate text - ∅ pair in text classification or sequence tagging

- At the output,
  - (1) the token representations are fed into an output layer for token-level tasks, such as sequence tagging or question answering
  - (2) the [CLS] representation is fed into an output layer for classification, such as entailment or sentiment analysis

Fine-tuning BERT



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

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

(c) Question Answering Tasks: SQuAD v1.1

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

Figure 4: Illustrations of Fine-tuning BERT on Different Tasks.

Datasets for pre-training

- BooksCorpus (800M words) (Zhu et al., 2015)
- English Wikipedia (2,500M words)

Ablation Studies

- Effect of Pre-training Tasks
  - No NSP
  - LTR & No NSP

Tasks	Dev Set MNLI-m (Acc)	QNLI (Acc)	MRPC (Acc)	SST-2 (Acc)	SQuAD (F1)
$\text{BERT}_{\text{BASE}}$	84.4	88.4	86.7	92.7	88.5
No NSP	83.9	84.9	86.5	92.6	87.9
LTR & No NSP	82.1	84.3	77.5	92.1	77.8
+ BiLSTM	82.1	84.1	75.7	91.6	84.9

Table 5: Ablation over the pre-training tasks using the $\text{BERT}_{\text{BASE}}$ architecture. "No NSP" is trained without the next sentence prediction task. "LTR & No NSP" is trained as a left-to-right LM without the next sentence prediction, like OpenAI GPT. "+ BiLSTM" adds a randomly initialized BiLSTM on top of the "LTR + No NSP" model during fine-tuning.

Ablation Studies

- Effect of Model Size
  - We can see that larger models lead to a strict accuracy improvement
  - We believe that this is the first work to demonstrate convincingly that scaling to extreme model sizes also leads to large improvements on very small scaled tasks

* 펄플렉서티(perplexity, PPL)는 언어 모델을 평가하기 위한 내부 평가 지표
영어에서 'perplexed'는 '헷갈리는'과 유사한 의미, 여기서 PPL은 '헷갈리는 정도'로 이해 가능
PPL은 수치가 높으면 좋은 성능을 의미하는 것이 아니라, '낮을수록' 언어 모델의 성능이 좋다는 것을 의미

Hyperparams				Dev Set Accuracy		
#L	#H	#A	LM (ppl)	MNLI-m	MRPC	SST-2
3	768	12	5.84	77.9	79.8	88.4
6	768	3	5.24	80.6	82.2	90.7
6	768	12	4.68	81.9	84.8	91.3
12	768	12	3.99	84.4	86.7	92.9
12	1024	16	3.54	85.7	86.9	93.3
24	1024	16	3.23	86.6	87.8	93.7

Table 6: Ablation over BERT model size. #L = the number of layers; #H = hidden size; #A = number of attention heads. "LM (ppl)" is the masked LM perplexity of held-out training data.

Ablation Studies

- Feature-based Approach with BERT
  - Feature-based approach 장점
    - Not all tasks can be easily represented by a Transformer encoder architecture
    - There are major computational benefits
  - The best performing method concatenates the token representations from the top four hidden layers of the pre-trained Transformer, which is only 0.3 F1 behind fine-tuning the entire model
  - This demonstrates that BERT is effective for both fine-tuning and feature-based approaches.

System	Dev F1	Test F1
ELMo (Peters et al., 2018a)	95.7	92.2
CVT (Clark et al., 2018)	-	92.6
CSE (Akbik et al., 2018)	-	**93.1**
Fine-tuning approach		
$\text{BERT}_{\text{LARGE}}$	96.6	92.8
$\text{BERT}_{\text{BASE}}$	96.4	92.4
Feature-based approach ($\text{BERT}_{\text{BASE}}$)		
Embeddings	91.0	-
Second-to-Last Hidden	95.6	-
Last Hidden	94.9	-
Weighted Sum Last Four Hidden	95.9	-
Concat Last Four Hidden	96.1	-
Weighted Sum All 12 Layers	95.5	-

Table 7: CoNLL-2003 Named Entity Recognition results. Hyperparameters were selected using the Dev set. The reported Dev and Test scores are averaged over 5 random restarts using those hyperparameters.

Conclusion

- Recent empirical improvements due to transfer learning with language models have demonstrated that rich, unsupervised pre-training is an integral part of many language understanding systems
- In particular, these results enable even low-resource tasks to benefit from deep unidirectional architectures
- Our major contribution is further generalizing these findings to deep *bidirectional* architectures, allowing the same pre-trained model to successfully tackle a broad set of NLP tasks.

References

- https://mino-park7.github.io/nlp/2018/12/12/bert-논문정리/?fbclid=IwAR3S-8iLWEVG6FGUVxoYdwQyA-zG0GpOUzVEsFBd0ARFg4eFXqCyGLznu7w
- https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270
- https://wikidocs.net/21697
- https://www.youtube.com/watch?v=IwtexRHoWG0