2021 DeepSleep Paper Review

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

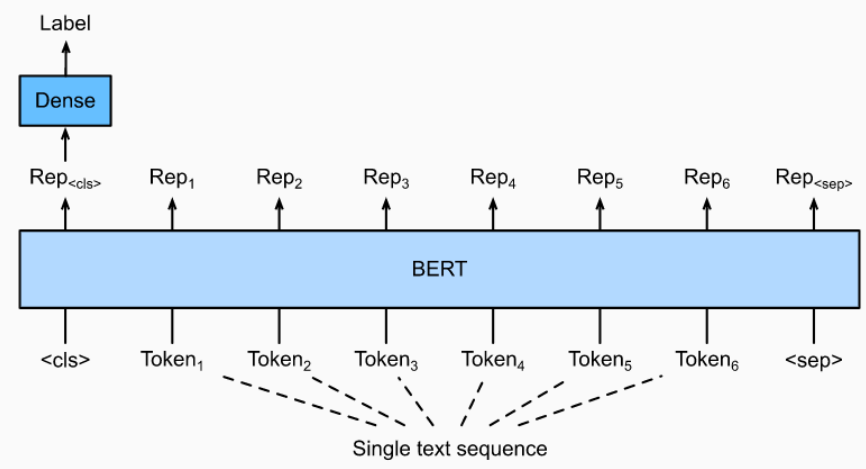
Presenter : Haram Lee

출처 : https://d2l.ai/chapter_natural-language-processing-applications/finetuning-bert.html

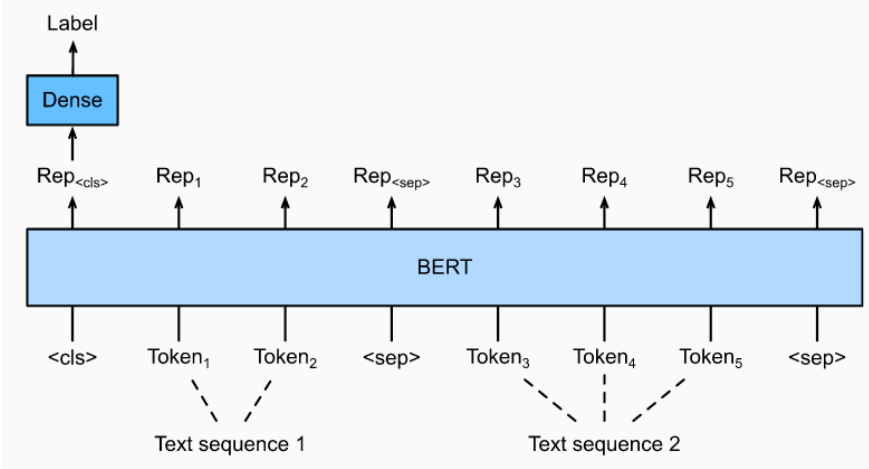
Introduction

NLP tasks

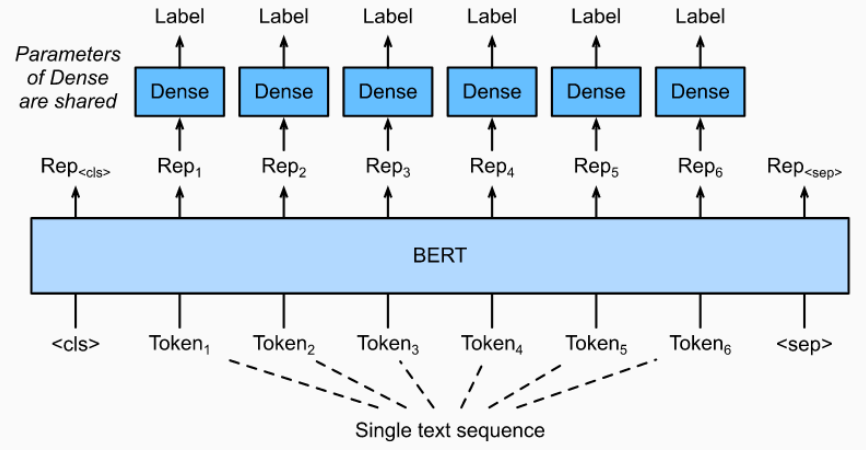
Single Text Classification



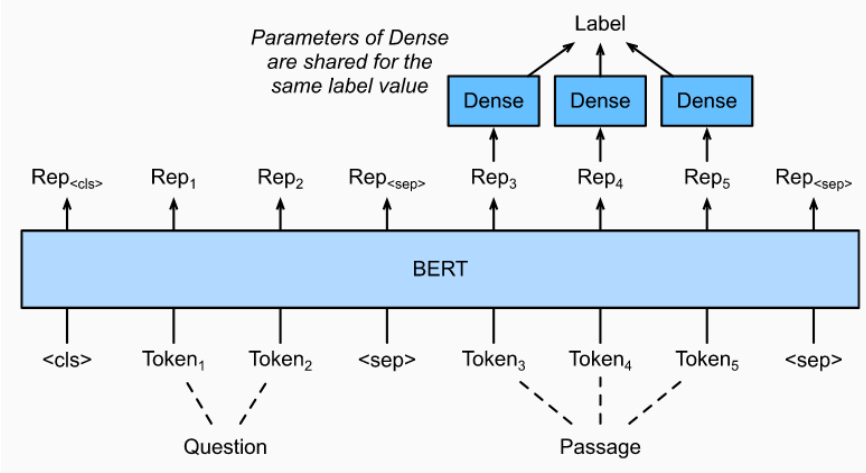
Text Pair Classification or Regression



Text Tagging



Question Answering



Introduction

사전 학습된 언어 표현을 여러 task에 적용하기 위한 방법

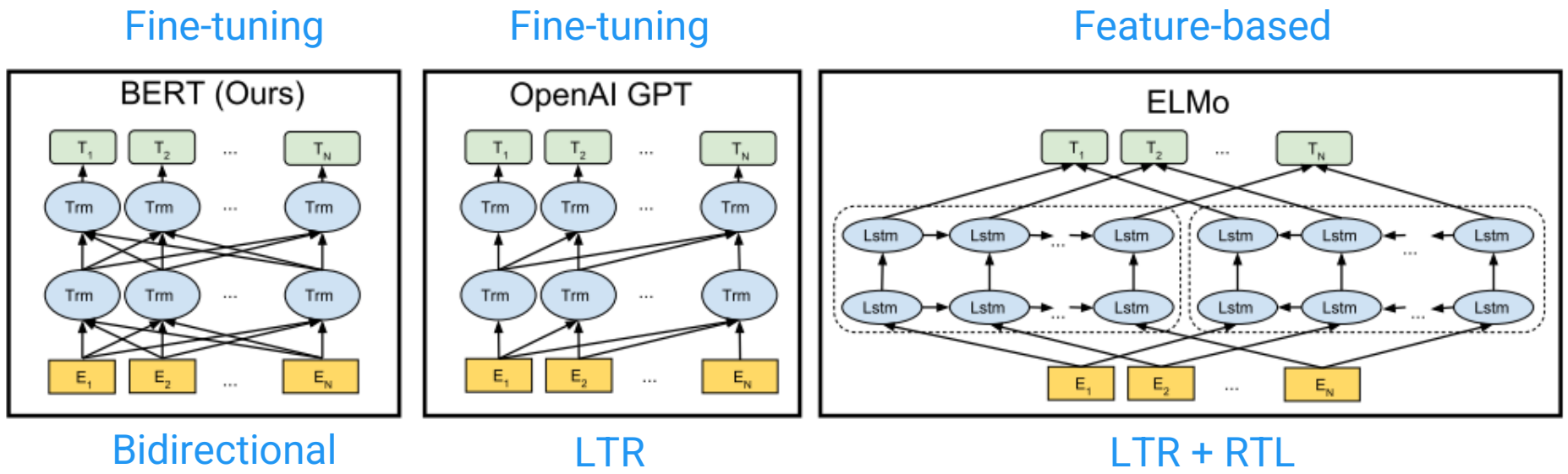


Figure 3: Differences in pre-training model architectures. BERT uses a bidirectional Transformer. OpenAI GPT uses a left-to-right Transformer. ELMo uses the concatenation of independently trained left-to-right and right-to-left LSTMs to generate features for downstream tasks. Among the three, only BERT representations are jointly conditioned on both left and right context in all layers. In addition to the architecture differences, BERT and OpenAI GPT are fine-tuning approaches, while ELMo is a feature-based approach.

Introduction

BERT : Bidirectional Encoder Representations from Transformers

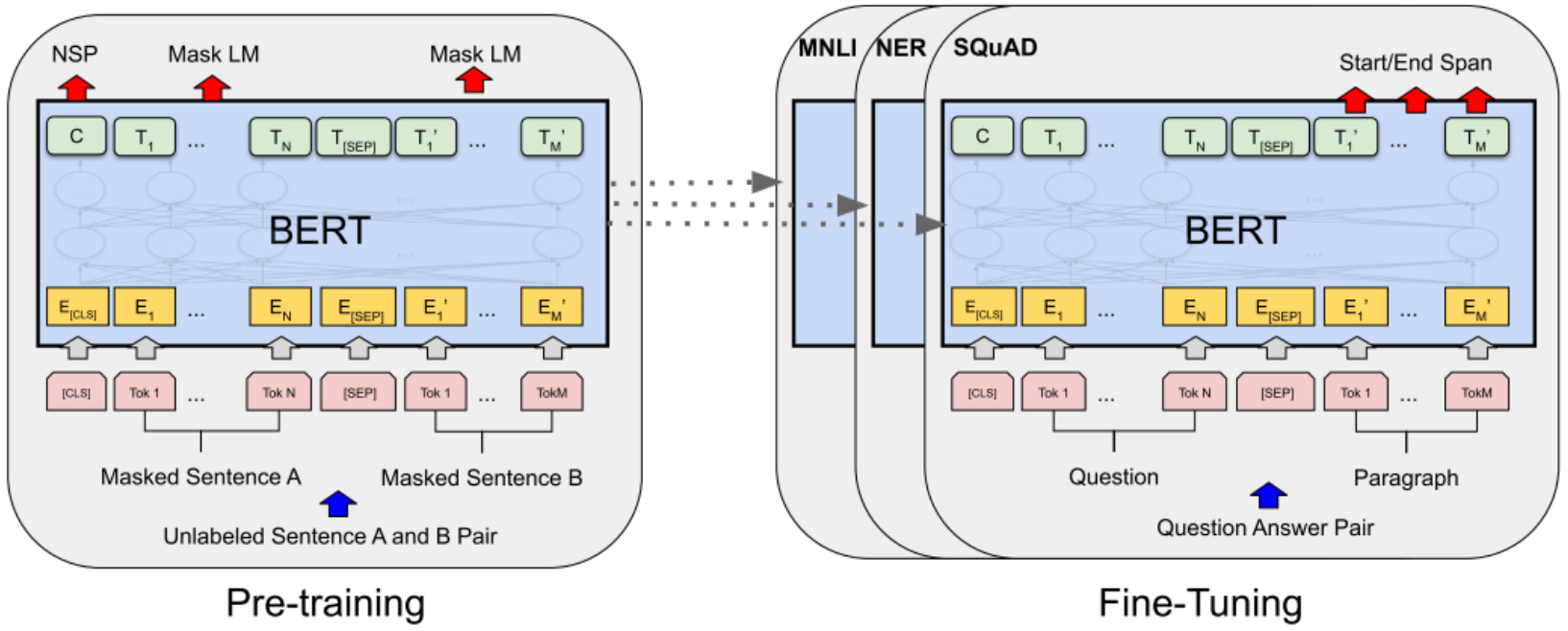
- 사전학습(pre-training)하도록 설계
  - task-specific 한 아키텍쳐를 만들기 위한 노력을 줄일 수 있다.
  - 하나의 output layer만 붙이면 fine tuning 가능
- 양방향 사전학습(Bidirectional pre-training)
  - MLM(Masked Language Model)
  - NSP(Next Sentence Prediction)
- 주요 기여
  - 사전 훈련된 동일한 모델을 이용하여 11개의 NLP과제에서 SOTA를 이뤘다.
  - unsupervised pre-training이 많은 언어 이해 시스템의 필수적인 부분이라는 것을 입증
  - Low-resource tasks도 deep unidirectional architectures의 이점을 누릴 수 있다.

Model Architecture

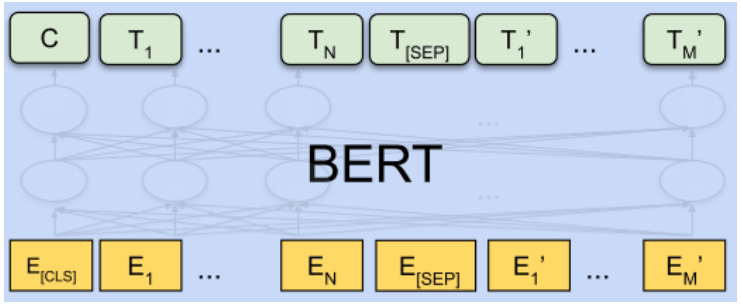
Multi-layer bidirectional Transformer encoder

- L: 레이어 개수
- H: 히든 사이즈
- A: self-attention head 개수

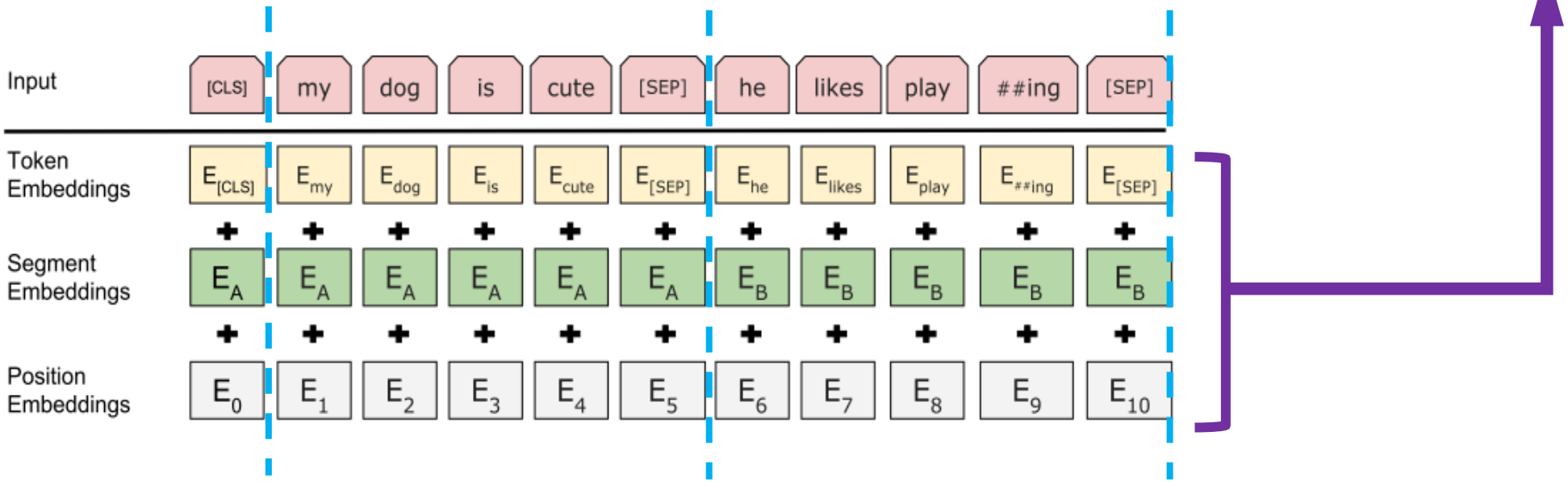
- $BERT_{BASE}$: L=12, H=768, A=12, Total Parameters=110M (OpenAI GPT와 같은 사이즈)
- $BERT_{LARGE}$: L=24, H=1024, A=16, Total Parameters=340M



Model Architecture



Input/Output Representations



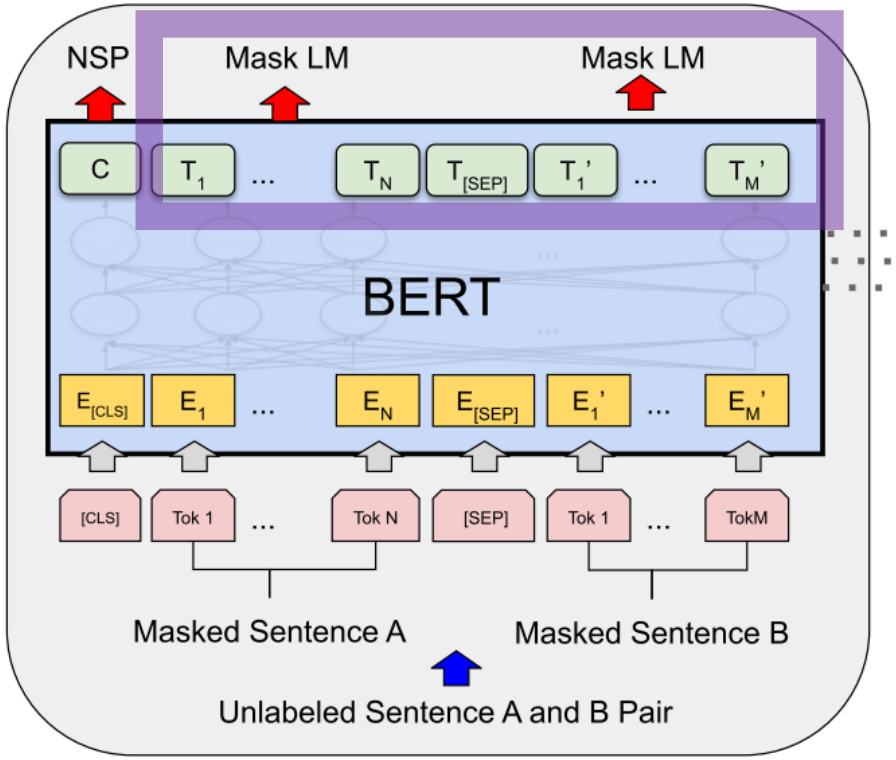
- “Sentence” : 한 문장이거나 두 문장을 이어 붙인 것
- WordPiece embeddings with a 30,000 token vocabulary
- [CLS] : 맨 앞에 있는 특별한 토큰
- [SEP] : 실제 문장 사이 구분

Model Architecture

Pre-training BERT

- **Masked Language Model(MLM)** (Cloze task)
  - 전체 토큰 중 **15%**를 마스킹하여 빈칸을 맞추기 위해 학습
  - i번째 토큰에 대한 최종 **hidden vector**인 T_i **vector**를 이용해서 예측한다.

- 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

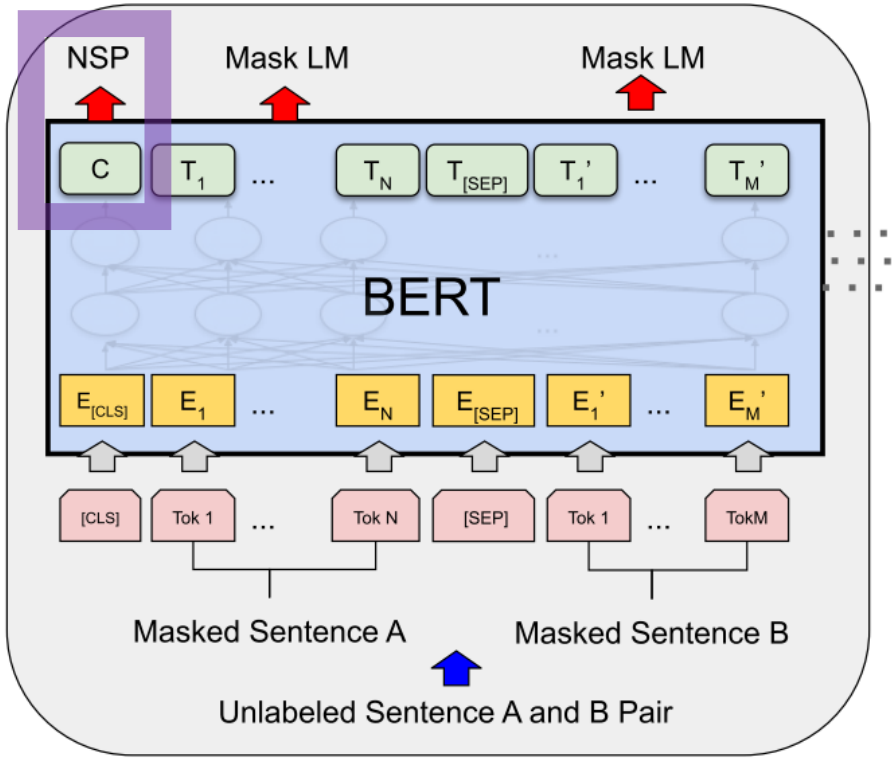
Model Architecture

Pre-training BERT

- **Next Sentence Prediction(NSP)**
  - Sentence A와 B를 선택할 때 50%는 B가 A의 다음 문장, 50%는 아닌 문장으로 구성
  - [CLS] 토큰의 최종 hidden vector인 C vector가 NSP에 사용된다.

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

Input = [CLS] the man [MASK] to the store [SEP]
penguin [MASK] are flight ##less birds [SEP]
Label = NotNext



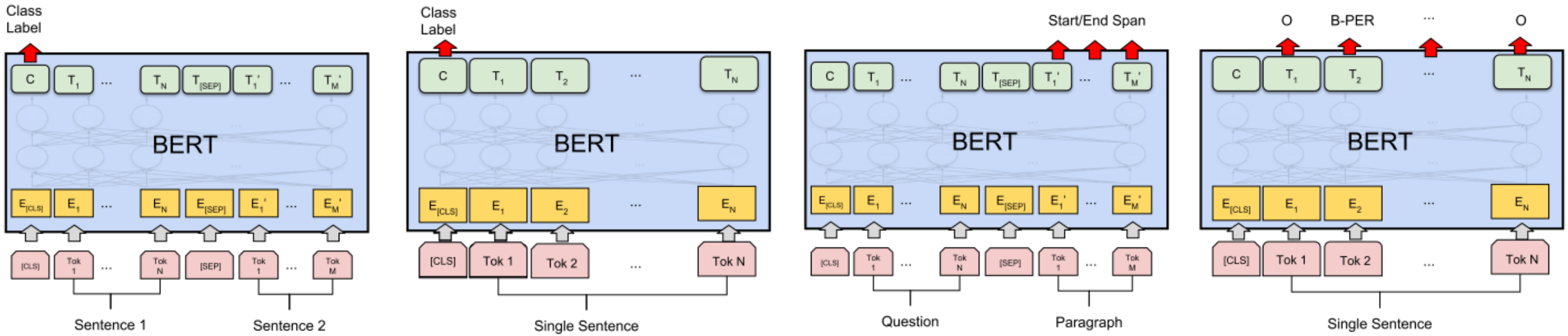
Pre-training

Model Architecture

Fine-tuning BERT

- 모든 매개 변수를 End-to-End Fine-tuning
- 각 task에 대해 task-specific한 입력과 출력을 연결

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



(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

Experiments

GLUE: The 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
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

Table 1: GLUE Test results, scored by the evaluation server (https://gluebenchmark.com/leaderboard). The number below each task denotes the number of training examples. The "Average" column is slightly different than the official GLUE score, since we exclude the problematic WNLI set.[8] BERT and OpenAI GPT are single-model, single task. F1 scores are reported for QQP and MRPC, Spearman correlations are reported for STS-B, and accuracy scores are reported for the other tasks. We exclude entries that use BERT as one of their components.

- [CLS] 토큰에 대한 최종 hidden vector인 C vector 이용 (Loss = $\log(softmax(CW^T))$
- $BERT_{BASE}$, $BERT_{LARGE}$ 가 모든 task에서 성능이 제일 좋았다.
- 데이터셋이 매우 작은 경우를 제외하면, $BERT_{LARGE}$가 $BERT_{BASE}$를 매우 큰 폭으로 이긴다.

Ablation Studies

Effect of Pre-training Tasks

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

- **No NSP** : MLM without NSP ➔ NSP의 영향 확인
- **LTR & No NSP** : MLM 대신 Left-to-Right(LTR) ➔ LTR과 MLM 비교
- **(+ BiLSTM)** : LTR & No NSP 에 양방향성을 위한 BiLSTM 추가

Ablation Studies

ELMO 처럼 LTR, RTL 모델을 각각 학습시켜서 concatenation하지 않은 이유

- Single bidirectional 모델보다 두 배로 비용이 많이 든다.
- RTL 모델을 학습시키기 어려운 task가 존재한다. (QA 등)

Effect of Model Size

- 모델이 충분히 사전 교육을 받았다면, 극단적인 모델 크기로 확장하는 것 또한 매우 작은 규모의 작업에서 큰 개선으로 이어진다.
- fine-tuning할 때, 랜덤으로 초기화되는 추가 매개변수를 매우 적은 수만 사용한다면, 데이터가 매우 작더라도 더 크고 더 표현적인 사전 학습된 표현으로부터 이익을 얻을 수 있다.

Ablation Studies

Feature-based Approach with BERT

- Feature-based approach의 장점
  - 태스크별 모델 아키텍처를 추가하면 더 잘 표현할 수 있다.
  - 사전학습된 표현 위에서 더 저렴한 모델로 많은 실험을 할 수 있다.
- 방법 : 레이어에서 활성화를 추출하여, 무작위로 초기화되는 2layer BiLSTM의 입력으로 사용

- 결론 : BERT는 Fine-tuning과 Feature-based 방식 모두에서 효과적이다.

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.

Q & A