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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova
Google AI Language, 2018.10

ETRI-UST
Jihyeon Song

2021.05.28. FRI

목차
INDEX

Abstract

- BERT: Bidirectional Encoder Representations from Transformers

- 모든 레이어의 left and right context를 함께 사용하여 unlabeled data로 모델을 pre-training 한 후, downstream task의 labeled data로 fine-tuning
  - Downstream task: 구체적으로 풀고 싶은 문제

- Pre-trained 된 BERT 모델은 task 별 architecture 수정없이, 자체 fine-tuning을 통해 광범위한 task에 대한 state-of-the-art (SOTA) 모델을 생성할 수 있음

1 Introduction

- Language model pre-training은 NLP tasks 개선에 효과적

- Pre-trained language representation을 downstream tasks에 적용하는 방식
  - Feature-based approach: 특정 task를 수행하기 위한 모델에 pre-trained language representation을 추가 feature로 제공, pre-trained 파라미터가 고정값 ex) ELMo
  - Fine-tuning approach: task-specific한 파라미터를 최대한 줄이고, pre-trained 된 파라미터를 downstream task 학습을 통해 바꾸는 방식 ex) OpenAI GPT

- 본 논문에서는 fine-tuning 기반 approach의 개선을 위해 BERT를 제안

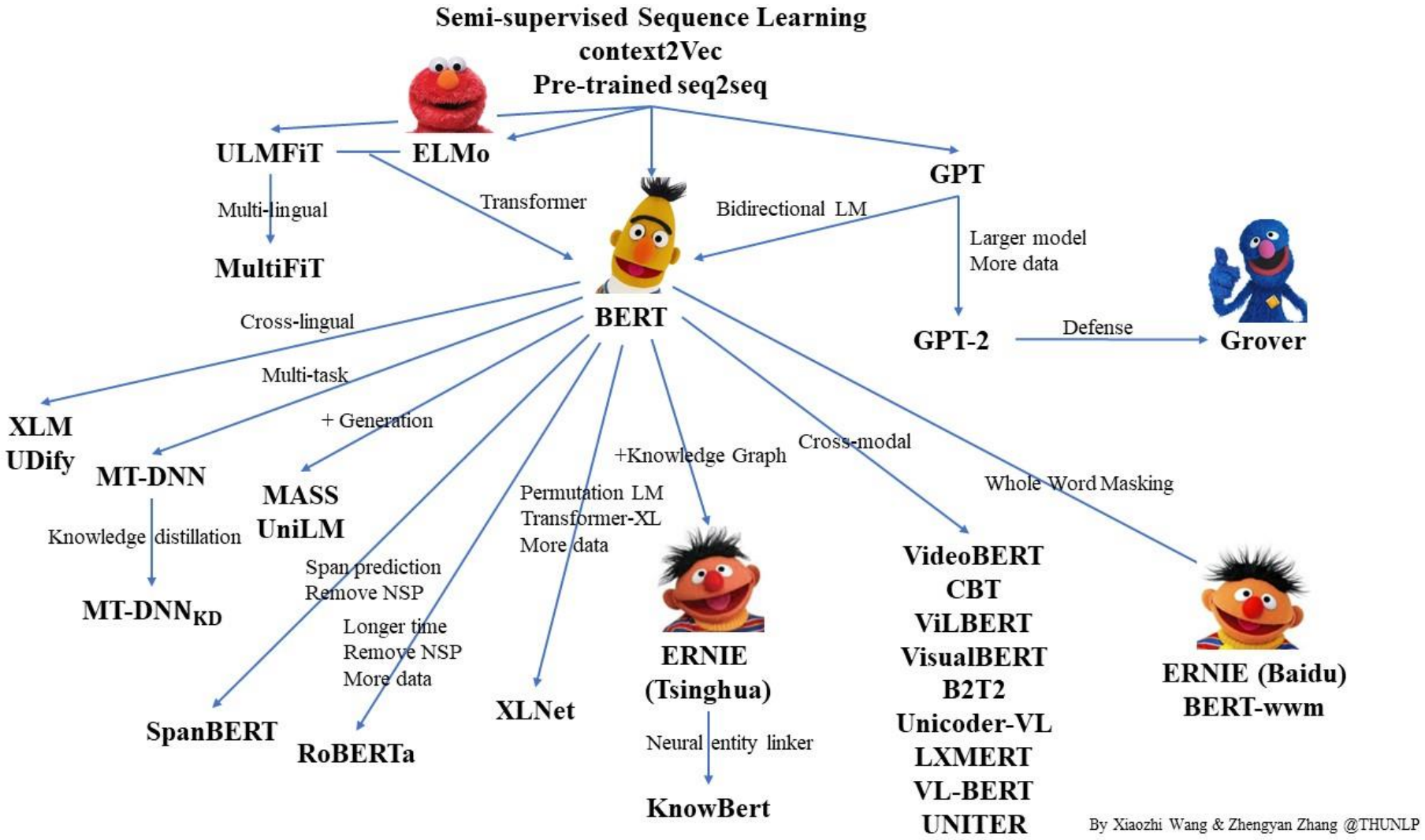
1 Introduction

- Pre-train BERT using two unsupervised tasks
  - Masked Language Model (MLM): 입력에서 일부 토큰을 무작위로 마스킹하고, 주변 context를 기반으로 마스킹 된 단어를 예측
  - Next Sentence Prediction (NSP): 두 문장을 함께 pre-train하여 next sentence예측

- Pre-trained representations가 task-specific한 architectures의 필요성을 줄여줄 수 있다는 것을 증명

- 대규모의 문장 및 토큰 수준 task에서 최고 수준의 성능을 보였으며, task-specific architectures를 능가하는 최초의 fine-tuning 기반 representation 모델
  - 11개 NLP tasks에 대한 SOTA를 제공함

1 Introduction



Pre-trained Language Model (PLM)

1 Introduction: ELMo & OpenAI GPT

- Embedding from Language Model (ELMo)
  - Contextualized word embeddings
  - Bidirectional Language Model (biLM) 사용
  - 독립적으로 학습된 left-to-right 및 right-to-left LSTM
  - Feature-based approach

- Generative Pre-trained Transformer (OpenAI GPT)
  - 이전 단어를 보고 다음 단어를 예측
  - Unidirectional Language Model
  - Fine-tuning approach

1 Introduction

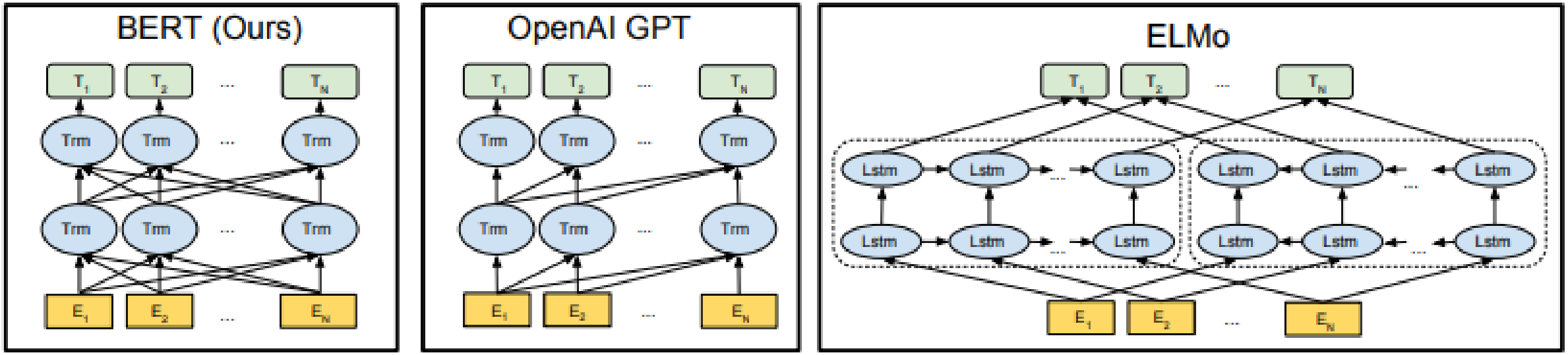


Figure 3: Differences in pre-training model architectures. BERT uses a bidirectional Transformer. OpenAI GPT uses a left-to-right Transformer. ELMo uses the concatenation of independently trained left-to-right and right-to-left LSTMs to generate features for downstream tasks. Among the three, only BERT 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.

2 Related Work: Unsupervised Feature-based Approaches

- Pre-trained word embeddings은 현대 NLP 시스템의 필수 요소
- Word embedding vector를 pre-train하기 위한 다양한 연구가 존재
- ELMo
  - Left-to-right 및 right-to-left language model에서 상황에 맞는 특징을 추출
  - contextual representation은 left-to-right 및 right-to-left representations의 concatenation
- ELMo는 질문 답변, 감정 분석 등 주요 NLP 벤치마크에 대해 SOTA를 제공

2 Related Work: Unsupervised Fine-tuning Approaches

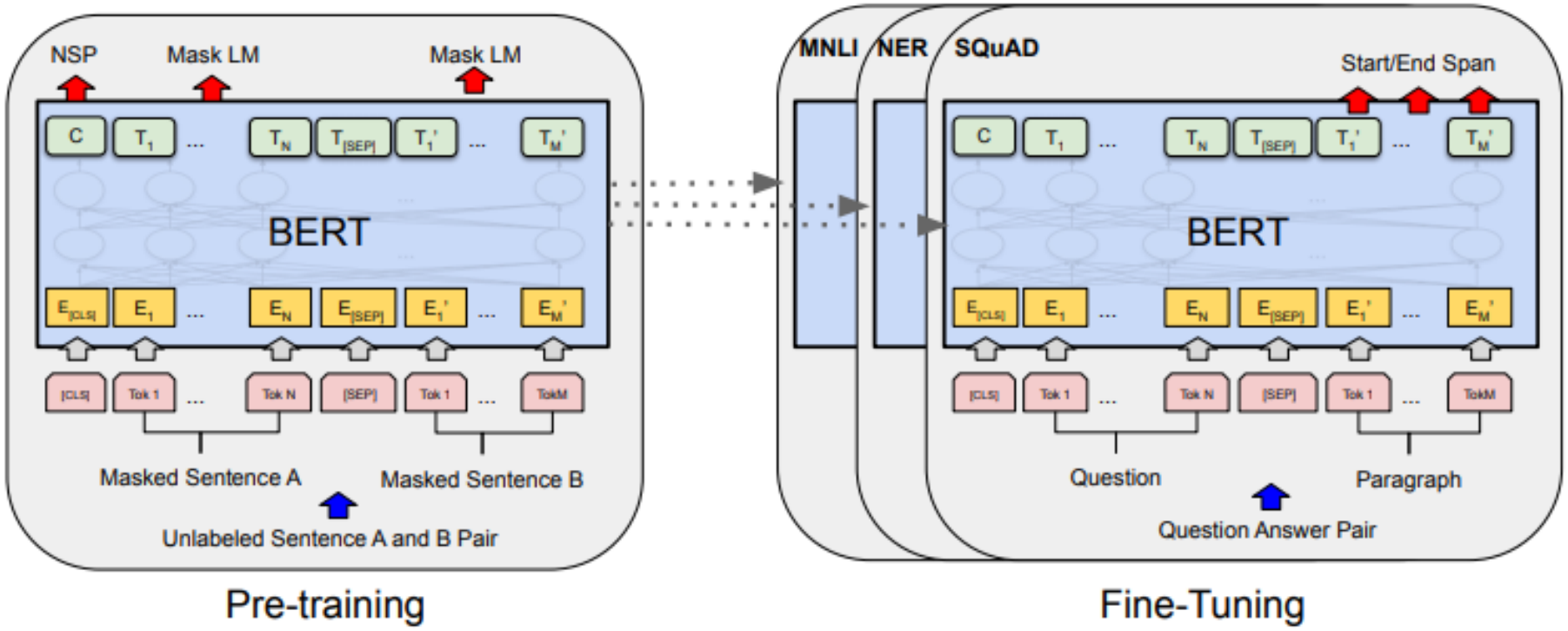
- 다양한 연구에서는 Contextual token representations를 생성하는 문장 또는 문서 인코더가 unlabeled text에서 pre-trained되어 downstream task에 맞게 fine-tuned 됨
- OpenAI GPT는 GLUE 벤치마크의 sentence-level tasks 에서 SOTA의 성능에 도달
- 이러한 모델을 pre-training 하기 위해 left-to-right language modeling을 사용

2 Related Work: Transfer Learning from Supervised Data

- 자연어 추론 및 기계 번역과 같은 대규모 데이터셋을 사용하는 supervised tasks 로부터 효과적인 transfer를 보여주는 연구가 존재
- 컴퓨터 비전 연구는 대규모 pre-trained models에서 transfer learning의 중요성을 입증
- ImageNet으로 pre-trained 된 모델을 fine-tune하는 방법이 가장 효과적

3 BERT

- BERT 모델은 pre-training 과 fine-tuning 단계로 나눌 수 있음
  - Pre-training 중에 모델은 pre-training tasks의 unlabeled data에 대해 학습함
  - BERT 모델은 먼저 pre-trained 파라미터로 초기화되고, 모든 파라미터는 downstream tasks의 labeled data를 사용하여 fine-tuned 됨



3 BERT: Model Architecture

- L: the number of layers
- H: the hidden size
- A: the number of self-attention heads

- BERTBASE
  - L=12, H=768, A=12, Total parameters=110M
  - OpenAI GPT와 동일한 모델 크기
  - 단, BERT는 bidirectional, OpenAI GPT는 left context 전용

- BERTLARGE
  - L=24, H=1024, A=16, Total parameters=340M

3 BERT: Input/Output Representations

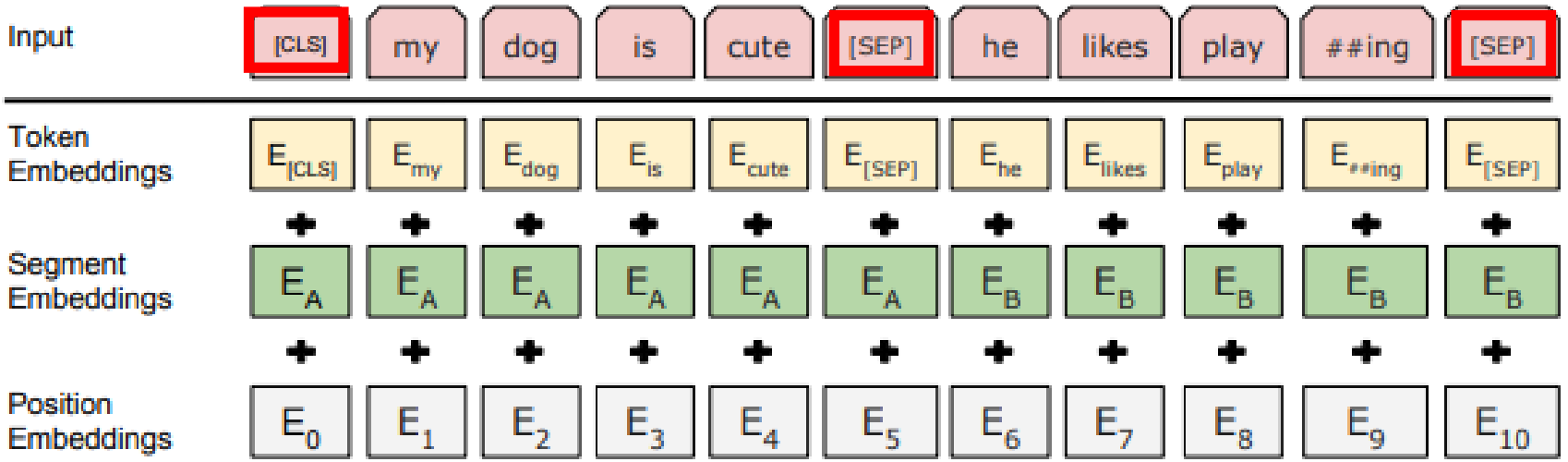
- BERT가 다양한 downstream tasks를 처리하도록 하기 위해 input representation은 단일 문장 및 한 쌍의 문장을 하나의 토큰 시퀀스로 나타낼 수 있음

- 30,000개의 토큰 vocabulary를 가진 WordPiece embeddings을 사용

- 시퀀스의 첫 번째 토큰은 특수 분류 토큰 ([CLS])
- 문장 쌍은 하나의 시퀀스로 합쳐져서 입력되며, 문장을 구분하기 위해 특수 토큰 ([SEP]) 을 사용하거나 모든 토큰에 학습된 embedding이 sentence A/B에 속하는지 여부를 나타내는 것을 추가

3 BERT: Input/Output Representations

- E: input embedding
- C: [CLS] token final hidden vector
- T_i: i^{th} input token final hidden vector
- [SEP]: 문장을 구분하기 위해 특수 토큰
- [CLS]: 시퀀스의 처음을 나타내는 특수 분류 토큰



3 BERT: Pre-training BERT

- Masked LM (MLM), Cloze task 라고도 부름
- Deep bidirectional representation을 학습하기 위해 입력 토큰의 일정 비율을 무작위로 마스킹 한 후, 마스킹 된 토큰을 예측
  - 마스킹 된 토큰에 해당하는 final hidden vectors는 output softmax에 전달됨
  - 본 논문의 모든 실험에서 WordPiece 토큰의 15%를 무작위로 마스킹

- [MASK] 토큰은 pre-training에만 사용되고 fine-tuning 할 때 사용되지 않음
  - Pre-training 및 fine-tuning 간에 mismatch가 발생할 수 있어, 마스킹 하는 토큰에 추가 작업을 수행
  - 80%: 토큰을 [MASK]로 변경
  - 10%: 토큰을 random word로 변경: 전체의 1.5%
  - 10%: 토큰을 원래 단어 그대로 둠

3 BERT: Pre-training BERT

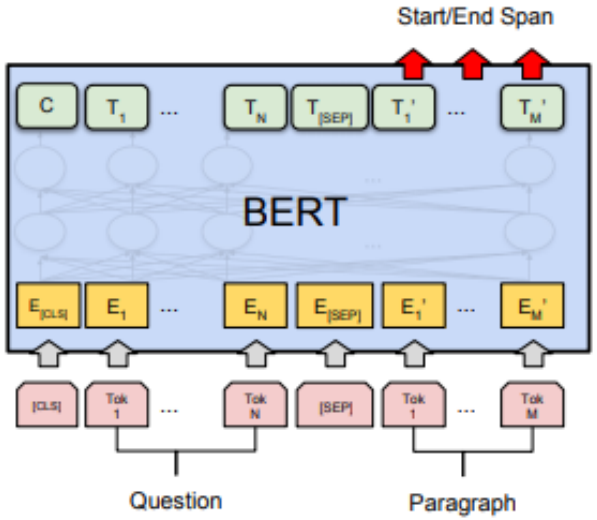
- Next Sentence Prediction (NSP)
- 문장 사이의 관계를 이해하는 모델을 훈련하기 위해, corpus에서 두 문장을 이어붙여 binarized next sentence prediction task를 pre-train함
- Pre-training 예제에 대해 sentence A와 B를 선택할 때,
  - 50%: sentence A, B가 실제 next sentence (IsNext)
  - 50%: corpus에서 무작위로 선택된 sentence A, B (NotNext)

- 사용한 pre-training corpus는 다음과 같음
  - BooksCorpus (800M words)
  - English Wikipedia (2,500M words, only text)

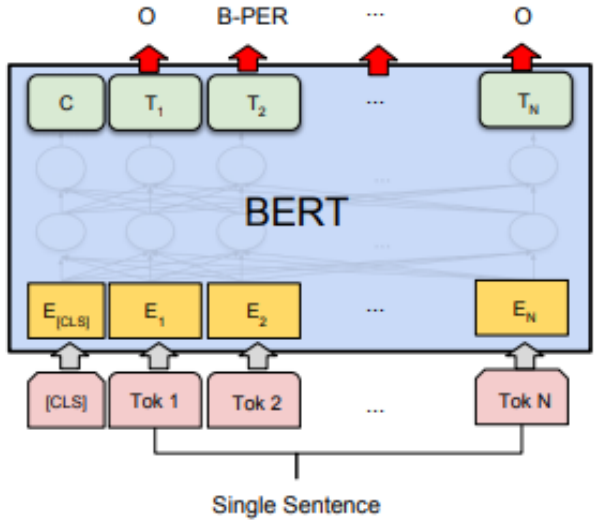
- Pre-training이 완료된 후, NSP 에 대한 accuracy는 97~98%를 달성

3 BERT: Fine-tuning BERT

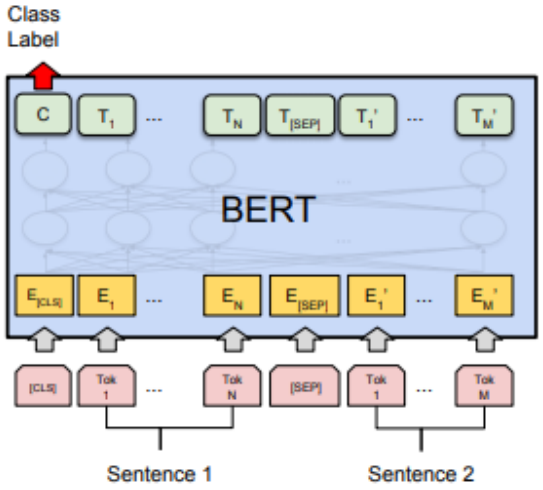
- Hyper-parameter
  - Batch size: 16, 32
  - Learning rate (Adam): 5e-5, 3e-5, 2e-5
  - Number of epochs: 2, 3, 4



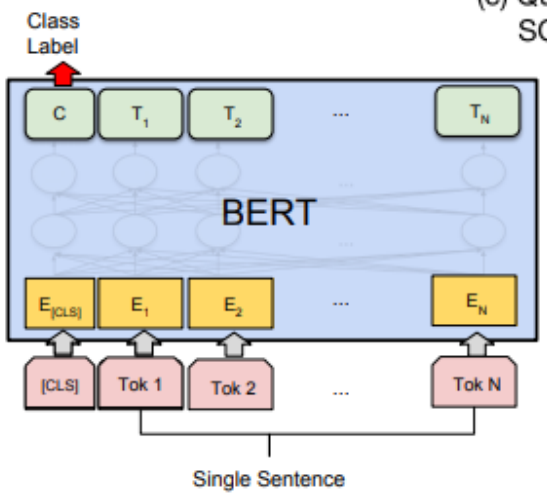
(c) Question Answering Tasks: SQuAD v1.1



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



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



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

▲ Token-level task
answer / token classification

◀ Sequence-level task
sentence classification

4 Experiments

- General Language Understanding Evaluation (GLUE) benchmark

- single-sentence tasks
  - CoLA (Corpus of Linguistic Acceptability)
  - SST-2 (Stanford Sentiment Treebank)
- similarity and paraphrase tasks
  - MRPC (Microsoft Research Paraphrase Corpus)
  - QQP (Quora Question Pairs)
  - STS-B (Semantic Textual Similarity Benchmark)
- inference tasks
  - MNLI (Multi-Genre Natural Language Inference)
  - QNLI (Question Natural Language Inference)
  - RTE (Recognizing Textual Entailment)
  - WNLI (Winograd Natural Language Inference): 데이터셋 구성에 문제가 있어 실험에서 제외함

4 Experiments

- BERTBASE 학습은 Cloud TPU 4개, BERTLARGE 학습은 Cloud TPU 16개로 수행했으며 pre-training 완료까지 4일이 소요됨

- GLUE
  - Sequence classification task
  - Batch size: 32
  - Epochs: 3
  - Dev set에서 최고의 fine-tuning learning rate를 선택: 5e-5, 4e-5, 3e-5, and 2e-5

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**

4 Experiments

- Stanford Question Answering Dataset (SQuAD) v1.1
  - 100K crowdsourced question/answer pairs
  - 답변이 포함된 wikipedia의 질문과 지문이 주어지면 해당 지문의 substring인 답변을 예측하는 task
  - 질문을 A embedding, 답변을 B embedding으로 처리
  - Batch size: 32
  - Epochs: 3
  - Learning rate: 5e-5

- System 훈련 시 public data를 사용할 수 있음
  - SQuAD에서 fine-tuning 하기 전, TriviaQA*으로 먼저 fine-tuning 하여 적당한 data augmentation을 사용함

- +1.5 F1 in ensembling and +1.3 F1 as a single system

System	Dev		Test	
	EM	F1	EM	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				
$\text{BERT}_{\text{BASE}}$ (Single)	80.8	88.5	-	-
$\text{BERT}_{\text{LARGE}}$ (Single)	84.1	90.9	-	-
$\text{BERT}_{\text{LARGE}}$ (Ensemble)	85.8	91.8	-	-
$\text{BERT}_{\text{LARGE}}$ (Sgl.+TriviaQA)	**84.2**	**91.1**	**85.1**	**91.8**
$\text{BERT}_{\text{LARGE}}$ (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.

❖ TriviaQA: A Large Scale Dataset for Reading Comprehension and Question Answering

4 Experiments

- SQuAD v2.0
  - 제공된 지문에 답변이 없을 가능성을 허용함으로써 SQuAD v1.1 문제 정의를 확장함
  - Batch size: 48
  - Epochs: 2
  - Learning rate: 5e-5
  - TriviaQA 사용하지 않음

- +5.1 F1 improvement

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$_{\text{LARGE}}$ (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.

4 Experiments

- The Situations With Adversarial Generations (SWAG)
  - 113k sentence-pair으로 구성되어 있으며, grounded common-sense inference 측정
  - 한 문장이 주어졌을 때, 보기의 네 가지 선택 중에서 가장 잘 이어지는 next sentence를 선택
  - Batch size: 16
  - Epochs: 3
  - Learning rate: 2e-5

- ESIM+ELMo system by +27.1%
- OpenAI GPT by 8.3%

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

Table 4: SWAG Dev and Test accuracies. †Human performance is measured with 100 samples, as reported in the SWAG paper.

5 Ablation Studies

- 본 논문에서 소개하는 MLM 및 NSP를 하나씩 제거하며 각각의 task의 효과를 증명
  - No NSP: MLM은 사용하지만 NSP 제거
  - LTR & No NSP: MLM 대신 Left-to-Right(LTR) 사용 NSP도 제거
  - LTR & No NSP + BiLSTM: LTR 시스템 강화를 위해 랜덤하게 초기화 된 BiLSTM 추가

- 모델 크기가 커질수록 fine-tuning task accuracy가 상승

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

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.

5 Ablation Studies

- Feature-based approach with BERT
  - 특정 task를 수행할 수 있는 모델을 추가하여 사용 가능
  - 학습 데이터의 expensive representation을 pre-compute한 후, cheaper 모델로 많은 실험을 수행하면 computational benefit을 얻을 수 있음

- CoNLL-2003 Named Entity Recognition (NER) 에 적용

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.

6 Conclusion

- Language 모델의 unsupervised pre-training이 많은 language understanding system에서 필수임을 증명
- Low-resource tasks에서도 deep unidirectional architectures의 이점을 얻을 수 있음
- 광범위한 NLP tasks에 대해 SOTA를 달성

Reference

- https://github.com/google-research/bert
- https://wikidocs.net/108730
- https://mino-park7.github.io/nlp/2018/12/12/bert-%EB%85%BC%EB%AC%B8%EC%A0%95%EB%A6%AC/
- 이동준, 김성동, “엄청 큰 언어 모델 공장 가동기 (LaRva: Language Representation by Clova)”, DEVIEW 2019 발표
- https://gluebenchmark.com/
- https://github.com/thunlp/PLMpapers
- https://hryang06.github.io/nlp/NLP/

Thank you