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

1. Introduction

Pre-trained Language Model은 다양한 NLP task의 성능을 향상시킬 수 있습니다. Pre-trained Language Model을 적용하는 방법에는 크게 2가지가 있는데, feature-based와 fine-tuning의 접근 방법이 있습니다. ELMo와 같은 feature-based approach는 pre-trained representation을 추가적인 features로 task-specific architectures에 포함하여 사용하는 방식입니다. OpenAI GPT와 같은 fine-tuning approach는 모든 pre-trained parameters를 down-stream tasks에 학습시키고, 최소한의 task-specific parameters를 도입하는 방식입니다.

지금까지 ELMo와 OpenAI GPT는 모두 unidirectional의 정보만 사용하였습니다. 그러나 BERT 연구자들은, 기존 방법들이 unidirectional의 한계점으로 인해 pre-trained representation을 잘 나타내지 못한다는 단점이 있다고 말하며, MLM(masked language model) 방법으로 language model을 학습시키는 방법을 제안하고 있습니다. BERT는 masked language model(MLM)을 사용하여 unidirectionality를 완화하며, fine-tuning based접근법을 향상했습니다. 그리고 MLM은 representation이 양방향의 context를 융합하므로 bidirectional Transformer를 pre-training할 수 있습니다.

2. Related Work

Pre-training language representation 방법론(feature based, fine tuning, transfer learning)의 몇가지를 소개합니다.

2-1. Unsupervised Feature-based Approaches

단어혹은 문장의 representation을 학습하는 것은 크게 non-neural method와 neural method(word2vec, GloVe 등)로 나눠집니다. word embedding의 Pre-trained word embedding을 NLP 시스템의 성능 향상을 이뤄냈습니다. 또한 word embedding을 통한 접근 방식은 자연스레 sentence embedding 혹은 paragraph embedding으로 이어졌습니다. ELMo와 그 후속 모델들은 2개의 biLM(left-to-right LM, right-to-left LM)을 이용해서 문맥을 고려한 단어 임베딩을 제공합니다. 한 단어의 representation을 left-to-right representation과 right-to-left representation을 연결해서 제공합니다. ELMo를 사용함으로서 몇 가지의 NLP benchmark에서 SOTA의 성능을 제공할 수 있었습니다.

2-2. Unsupervised Fine-tuning Approaches

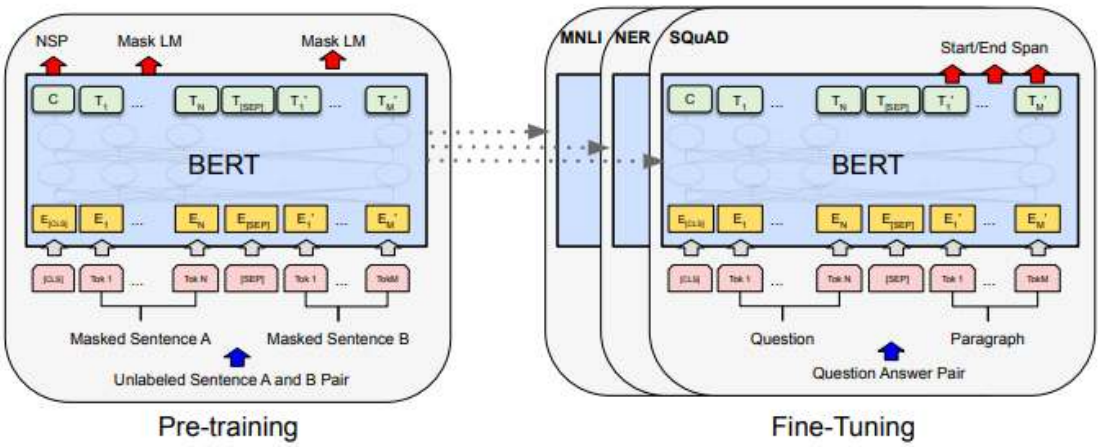
최근에는 레이블링 하지 않는 텍스트를 통해 pre-trained 된 contextual token representation을 생성하고, supervised downstream task에서 fine-tunin 하는 방식이 제안되었습니다. 이러한 방식의 장점은 시작부터 학습이 완료될때까지 적은 수의 파라미터가 학습된다는 것입니다. 이러한 장점을 기반으로 OpenAI의 GPT 역시 이러한 방식을 사용하여 GLUE Benchmark의 다양한 task에서 SOTA의 성능을 제공할 수 있었습니다.

2-3. Transfer Learning from Supervised Data

큰 데이터셋을 보유하고 있는 task (NLI, MT 등) 들을 이용해서 transfer learning 을 수행하는 방식도 존재합니다.

3. BERT

Bert는 pre-training part와 fine-tuning part이라는 두가지 framework로 구성되어 있습니다. pre-training동안에는, unlabled data로 pre-training task들을 수행합니다. Fine tuning 단계에서는, pre-training된 parameter와 labled data로 finetuning task를 수행합니다. Bert는 fine-tunnign과정에서 pre training의 모델 구조와 parameter를 전부 활용하고, 약간의 레이어 추가만으로 fine tuning task를 수행한다는 점에서 장점을 가지고 있습니다.

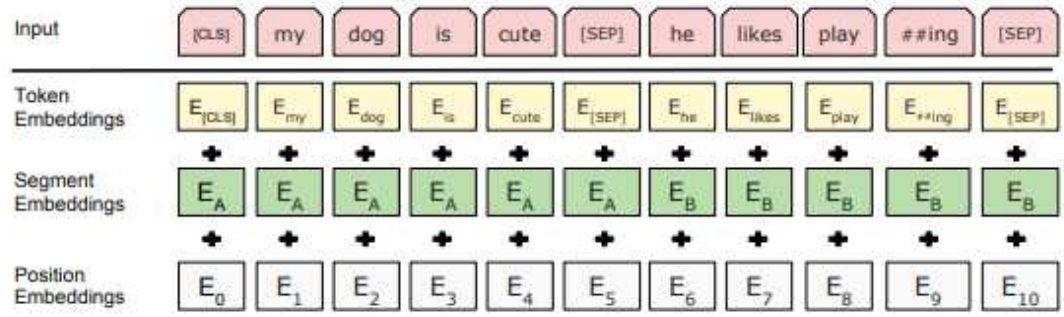


Model Architecture

Bert의 모델 아키텍처는 multi-layer bidirectional Transformer encoder입니다.
즉, 양방향 Transformer encoder를 여러 층 쌓은 것입니다.
BERT base의 경우 L = 12, H = 768, A = 12, Total Parameters=110M를 사용하였고, BERT large의 경우 L = 24, H = 1024, A = 16, Total Parameters=340M를 사용하였습니다.

Input/Output Representations

single sentence 혹은 two sentences packed together를 하나의 토큰 시퀀스로 분명하게 표현해야 합니다. 이를 위해 BERT에서는 총 3가지의 Embedding vector를 합쳐서 input으로 사용합니다. 모든 Input 시퀀스의 첫번째 토큰은 [CLS] 토큰인데, [CLS] 토큰과 대응되는 최종 hidden state는 분류 문제를 해결하기 위해 sequence representation들을 종합합니다. 또한, Input 시퀀스는 문장의 한 쌍으로 구성됩니다. 문장 쌍의 각 문장들은 [SEP] 토큰으로 분리됩니다. 또한 각 문장이 A문장인지, B문장인지 구분하기 위한 임베딩(Segment Embeddings) 역시 진행합니다. Token Embeddings는 WordPiece embedding을 사용하고, Position Embeddings는 Transformer에서 사용한 방식과 동일합니다. Input representation은 이러한 대응되는 토큰 token embedding, segment embedding, positional embedding의 합으로 나타냅니다.



3.1 Pre-training BERT

ELMo나 GPT의 방식과 달리 BERT는 기존의 Left to Right, Right to Left 방식을 사용하지 않습니다. 그 대신, BERT에서는 두가지 unsupervised task를 통해서 문장을 이해하는 능력을 습득하게 됩니다.

Task #1 Masked LM

BERT는 양방향 모델을 학습하기 위해서, 15%의 단어를 랜덤으로 선택하고 이중, 80%는 [MASK]토큰으로, 10% 는 랜덤한 단어로, 그리고 나머지 10%는 그대로 두어 [MASK] 를 맞추게 하는 task를 진행하였습니다. BERT는 80%의 [MASK] 토큰에 대해서는 [MASK] 토큰에 들어갈 단어를 예측하게 되고, 나머지 20%의 경우에는 원래 단어를 예측하는 task를(바뀌지 않은 상태라고 하더라도) 진행하게 됩니다.

Task #2 Next Sentence Prediction(NSP)

두번째 task는 QA와 NLI(Natural Language Inference, 두 문장의 관계가 Entailment, Contradiction, Neutral중에 어떤것인지 파악하는 문제) 등에 활용하기 위해 두 문장 함께 학습하는 task입니다. Next Sentece Prediction은 이어지는 두 문장과, 이어지지 않는 두 문장을 같은 비율로 하여 dataset을 만든 뒤, 이어지는 문장이라면 IsNext를 아니라면 NotNext로 라벨링 하여 이를 맞추는 task입니다.

3.2 Fine-tuning BERT

Fine-tuning은 빠르고 간단하며, Transformer의 self-attention 메커니즘 덕분에 BERT는 많은 downstream task를 모델링 할 수 있습니다. 이때, 수행하고자 하는 downstream task에 따라 B=ERT는 task-specific input을 받아, fine-tuning을 진행합니다. 아래 그림은 각각의 NLP Task에 대해서 Fine-tuning이 어떻게 이뤄지는지를 도식화한 그림입니다.

1. Sentence pairs in paraphrasing
2. Hypothesis-Premise pairs in entailment
3. Question-Passage pairs in question answering
4. Degenerate-None pair in text classification or sequence tagging

Output:

1. token representation in sequence tagging or question answering
2. [CLS] representation in classification(entailment or sentiment analysis)

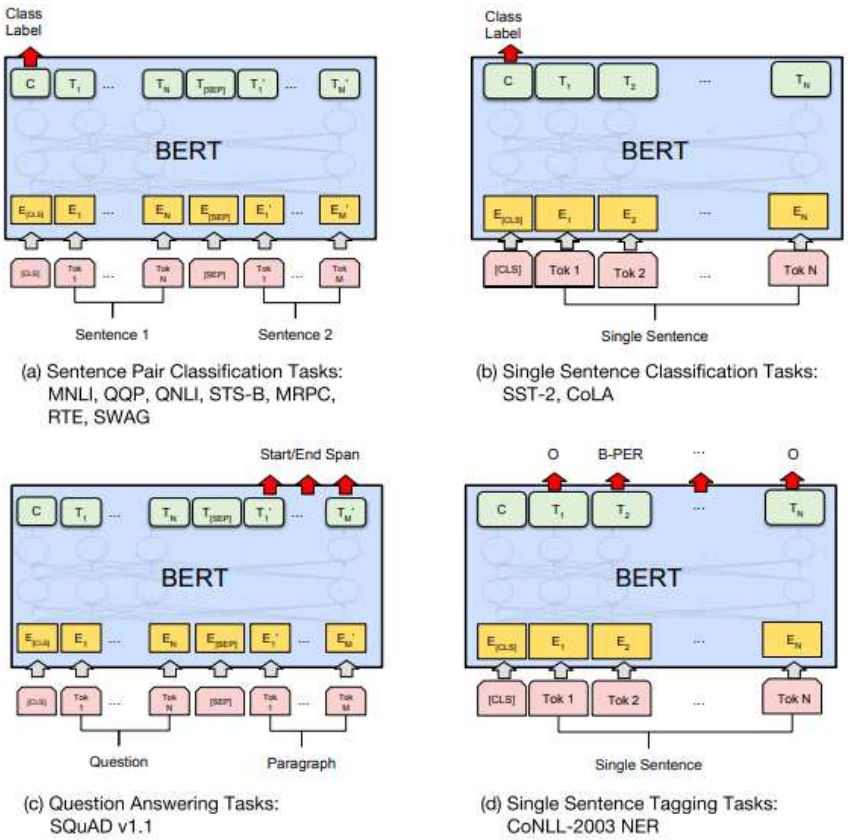


Figure 4: Illustrations of Fine-tuning BERT on Different Tasks.

4. Experiments

11개의 NLP 분야에서 BERT fine-tuning을 학습시켰습니다.

GLUE

GLUE는 다양한 분야의 general language understanding task를 포함하고 이를 평가합니다. GLUE에 맞추어 fine-tuning을 하기 위해서 분류에 있어 결과 개수에 따른 가중치 행렬 변화를 생성했습니다. 32개의 Batch size와 3 epoch 그리고 learning rate는 5e-5, 4e-5, 3e-5, 2e-5 중 가장 학습이 잘 진행된 것으로 선택했습니다.

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**

BERT Model은 GLUE의 모든 task에서 SOTA를 달성했고, BERT_BASE, BERT_LARGE 모두 다른 MODEL들 보다 성능이 좋게 나왔습니다. 또한 BERT_LARGE가 BERT_BASE 보다 모든 부분에서 성능이 좋습니다. 즉 Model의 크기가 클수록 성능이 좋다는 의미이다. BERT_BASE와 OpenAI GPT는 같은 조건이었지만 성능의 차이가 있습니다. 즉 bidirectional attention을 적용하는 것이 성능 향상에 큰 영향을 준다는 것을 알 수 있습니다.

SQuAD v1.1

SQuAD에 맞춰 Fine-tuning을 진행하였습니다. SQuAD는 question과 answer로 이루어진 pair 데이터셋입니다. Fine-tuning 학습 과정에서 answer 문장의 시작 token S와 끝 token E를 구한다. Transformer layer의 마지막 state를 이용해 해당 값이 S나 E일 확률을 구하는데 S와 T_i를 dot product한 후 softmax로 확률값으로 변환합니다. 나온 score를 이용해서 S*T_i와 E*T_J를 더한 값이 가장 큰 <i, j>을 최종 answer 영역으로 정한다. 32개의 batch-size와 3 epoch 그리고 learning rate는 5e-5로 선택했습니다.

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**

SQuAD v2.0

SQuAD v1.1에서 대답이 불가능한 질문을 포함한 dataset입니다. Fine-tuning 학습 중 CLS token을 이용하여 대답 가능 여부를 Binary Classification하면서 token C를 구한다. 대답이 불가능한 경우 snull=S・C+E・C로 score값으로 계산하고 대답이 가능한 경우 SQuAD v1.1과 동일하게 계산합니다.
대답 가능 여부는 두 score값을 비교하여 s^i,j>snull+r 식으로 판단한다. 48개의 batch-size와 2 epoch 그리고 learning rate는 5e-5로 선택했습니다.

SWAG

일반적인 추론을 하는 dataset으로 앞 문장이 주어지고 추가적으로 4개의 문장이 주어지고 4개의 문장 중 가장 적합한 것으로 선택하는 task입니다. 앞뒤 문장을 묶어 embedding A, B로 정의한 하나의 데이터셋을 만든다. CLS token에 대응하는 token C와 뒤 문장을 dot product한 뒤 softmax로 normalize하여 진행합니다.

5. Ablation Studies

특정 조건을 변경해 영향을 끼치는 정도를 분석합니다.

1. Pre-training 영향

pre-training의 효과를 알아보기 위해 기본모델과 BERT에서 NSP를 제거한 모델, MLM대신 LTR만 진행하고 NSP를 제거한모델 , BERT에 MLM 대신 BiLSTM를 추가한 모델 을 비교했습니다. 결과를 살펴보면 BERT와 No NSP 를 비교해봄으로써 NSP Pre -training 이 성능에 향상에 영향을 끼침을 알수 있습니다. NO NSP 와 LTR &No NSP 를 비교하면 MLM 이 성능향상에 매우 큰 영향을 끼침을 파악가능합니다. MLM 대신 BiLSTM 을 추가하면 SquAD 에서는 성능향상이 크지만 나머지(SST-2,MRPC)에서는 오히려 성능이 좋지 않아짐을 알 수 있습니다.

2. 모델크기가 미치는 영향

모델 크기가 증가할 수록 성능이 높아지는 경향을 확인할 수 있는데 특히 MRPC task는 pre-training task와 차이가 큰 task 이면서 3600개 적은 labeled training data 를 사용함에도 불구하고 모델크기가 증가함에 따라 성능도 향상됐다. 결과적으로 모델크기 증가는 번역 및 Langauge Modeling 과 같은 스케일이 큰 task 에서 도 성능 향상에 기여하고 충분한 pre- training 이 있는 작은 scale task 에서도 성능 향상에 기여함.

3. Feature-based Approach with BERT

feature-based Approach 의 장점은 Tranceformer Encoder 의 Output 에 몇몇 Layer 를 추가하는 간단한 작업 만으로는 해결할 수 없는 task 해결 가능 하다는 점과 매우 큰 pre-training 모델의 경우 pre -train 된 feature를 계속 업데이트 하지 않고 고정된 값으로 사용함으로써 연산량을 획기적으로 줄일 수 있다는 장점이 있습니다.