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

1.Introduction

pre-train된 언어표현을 down-stream task에 적용하는 데는 2가지 방식이 존재한다. feature-based 방식과 fine-tuning 방식이 그것이다. feature-based 방식(ex.ELMO)으로는 pre-train된 표현을 추가적인 피쳐로 포함시킨다. fine-tuning 방식(ex.ChatGPT)은 단방향(왼>오) 언어 모델이다. 이는 최소한의 작업별 매개변수를 도입하고 하위 작업에서 모든 사전 학습된 매개변수를 fine-tuning한다.

해당 논문의 저자는은 기존의 fine-tuning접근 방식이 그 단방향성으로 인해 pre-train된 표현의 기능을 제한한다고 말한다. 단방향 방식에서는 각 토큰이 이전 토큰에만 영향을 받기 때문이다. 이러한 제한은 문장 수준 작업에는 최적이지 않으며 질문-답변같은 token 수준의 작업에서는 양방향적 문맥 통합이 중요할 수 있기 때문이다.

이러한 한계를 극복하기 위해 저자는 BERT방식을 제안한다. BERT는 클로즈 작업에서 영감을 받은 “Masked Language Model(MLM)”을 통해 사전 학습 목표를 사용하여 단방향성을 극복하려 한다.

2. Related Work

데이터 표현 학습 방법과 모델의 적용 방식을 위주로

2.1 Unsupervised Feature-based Approaches

사전에 대량의 데이터를 사용하여 학습된 모델이나 파라미터 접근방법.

- **단어 임베딩의 발전:** 초기 연구는 주로 비신경망 방법에 초점을 맞췄으나, 시간이 지나면서 신경망을 이용한 방법(예: Mikolov et al., 2013; Pennington et al., 2014)이 등장. 간단히 설명하면 일단 기존에는 '단어'에 집중하여 워드 임베딩했다면 문장 전체의 context를 담아서 임베딩을 하려 시도
- **ELMo**: 미리 학습된 (pretrain) LSTM 모델에 문장 전체를 넣어 각 단어의 임베딩 벡터를 구하는 방식. 이 때 bidirection LSTM을 사용.

2.2 Unsupervised Fine-tuning Approaches

비지도 학습을 통한 미세 조정(fine-tuning) 접근 방법 -> 사전 학습된 모델을 특정 작업에 최적화하여 전체 모델을 미세 조정하는 데 초점

초기 접근법: 단어 임베딩의 사전 학습은 레이블이 없는 텍스트로부터 단어 임베딩 파라미터를 사전 학습하는 것에 초점

문장 및 문서 인코더의 발전: 이후 연구들(Dai and Le, 2015; Howard and Ruder, 2018; Radford et al., 2018)은 문장이나 문서 전체를 인코딩하여 문장 시퀀스를 예측하기 위해 문맥적 토큰 표현을 생성하는 인코더를 사전 학습. 이러한 접근법은 지도 학습 작업에 미세 조정될 수 있으며, 이 과정에서 새로운 파라미터를 거의 학습할 필요가 없다는 장점이 있다. 비지도 학습 기반 특징 추출 방법은 사전 학습된 임베딩을 고정된 특징으로 사용하는 반면, 비지도 학습 기반 미세 조정 방법은 사전 학습된 모델 전체를 특정 작업에 맞게 조정하여 더욱 맞춤화된 표현을 학습

2.3 Transfer Learning from Supervised Data

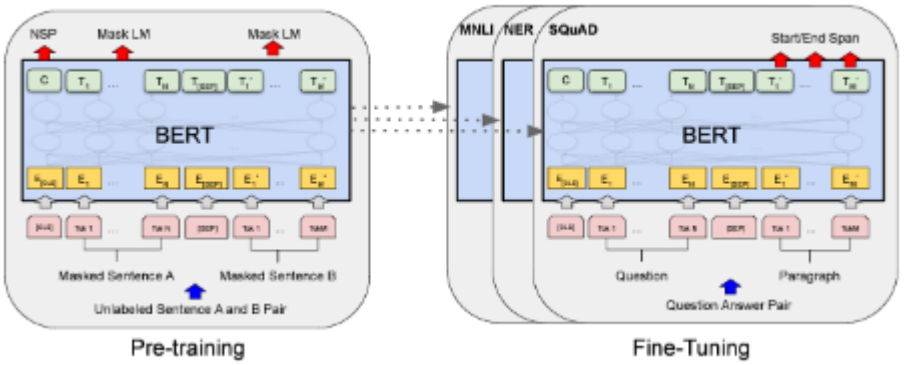
다양한 분야에서 대규모 데이터셋을 활용한 감독 학습 작업을 통한 성공적인 전이 학습 사례들을 소개

자연어 처리(NLP) 분야에서는 자연어 추론과 기계 번역 작업에서의 전이 학습이 효과적임을 보여주고 있으며, 컴퓨터 비전 분야에서는 ImageNet으로 사전 훈련된 모델들을 특정 작업에 맞게 세부 조정함으로써 성능을 개선할 수 있음을 강조

- 전이학습(Transfer Learning)은 기계학습의 한 분야로, 한 작업에서 학습한 지식을 다른 작업에 적용하여 모델의 학습을 개선하는 방법
- **도메인 전이학습**: 소스 도메인(학습 데이터)과 타겟 도메인(적용하고자 하는 데이터)이 다를 때 사용된다. 예를 들어, 인터넷에서 수집한 이미지로 학습한 이미지 분류 모델을 특정 의료 이미지 분류 작업에 적용하는 경우
- **작업 전이학습**: 소스 작업(학습한 작업)과 타겟 작업(적용하고자 하는 작업)이 다를 때 사용된다. 예를 들어, 자연어 처리에서 문장의 감정을 분류하는 모델을 학습한 후, 이 모델을 다른 언어의 문서 분류 작업에 적용하는 경우

3. BERT

Model Architecture



- multi-layer bidirectional Transformer encoder
- BERT Transformer uses bidirectional self-attention (“Transformer encoder”)
- GPT Transformer uses constrained self-attention (left-context-only version: “Transformer decoder” (can only be used for text generation))

Input/Output Representations

- “sentence” = 어법적 의미의 문장이 아닌 연결성 있는 텍스트의 부분
- “sequence” = BERT에게 input되는 token sequence (하나 또는 두 개의 sentence일 수 있음)
- sentence 구분 방법
  - special token [SEP]로 separate 시킴
  - 각 token에 learned embedding을 추가하여 어느 sentence에 속하는지 indicate시킴
- 각 토큰의 input representation은 corresponding token, segment embeddings과 position embeddings의 합임

3.1 Pre-training BERT

unlabeled data로 두 가지 unsupervised tasks를 훈련시킴

Task #1: Masked LM (Language Modeling)

- 15%의 토큰을 랜덤하게 선택하여 맥락에 따라 예측하도록 훈련
- 선택된 토큰은
  - 80% 의 확률로 [MASK] 토큰으로 대체됨
  - 10%의 확률로 랜덤한 단어 토큰으로 대체됨

    - 10%의 확률로 대체되지 않음

Task #2: Next Sentence Prediction (NSP)

- 두 “sentences” (spans of text, separated by [SEP]) 가 순서있이 주어졌을 때 이 둘이 either [IsNext]인지 [NotNext]인지 분류 예측
- 이 훈련 과정에 따라 BERT는 맥락 속 단어와 sentences에 따른 latent representations (은닉층)을 학습함

3.2 Fine-tuning BERT

- pre-training 후 더 적은 resources를 가지고 더 구체적인 task를 수행할 수 있음. (pre-training 에 비해 훨씬 비용이 적게 듬
- 구체적인 task 예시
    - NLP tasks
        - language inference
        - 텍스트 분류
    - sequence-to-sequence based language generation tasks
        - 질문 답하기
        - 대화적 답변 생성

4. Experiments

Bert fine-tuning experiment

4.1 GLUE

parameters: classification layer weights, learning rate

배치 사이즈 32, fine-tuning 3에포크 진행

Bert Large: 작은 데이터셋에서 불안정하여 랜덤니스를 더해주어 보완.

bert large, base 모델 모두 다른 모델보다 훨씬 좋은 결과를 보임.

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**

4.2 SQuAD v1.1

Question & Answer Dataset

input: question and passage as a single packed sequence

TriviaQA: fine-tuning 후 data augmentation 진행

ensemble 진행하고 TriviaQA를 한 bert large 모델의 결과가 가장 좋다.

System	Dev EM	Dev F1	Test EM	Test 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**

4.3 SQuAD v2.0

passage에 답이 없는 경우도 포함된 데이터셋.

예측을 위해 답이 없는 span의 점수와 비교도 함.

fine-tuning 2 에포크, 학습률: 0.00005, batch size: 48

single Bert Large 모델의 결과가 가장 좋다

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				
$\text{BERT}_{\text{LARGE}}$ (Single)	78.7	81.9	80.0	83.1

4.4 SWAG

task: 이어지는 문장 고르기

parameter: classification token

fine-tuning 3 에포크, 학습률: 0.00002, batch size: 16

softmax layer로 정규화

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

5. Ablation Studies

- BERT의 여러 측면에 대한 절제 실험을 수행하면서 상대적 중요성을 확인함.

	Dev Set				
Tasks	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

5.1 Effect of Pre-training Tasks

- Deep bidirectionality of BERT의 중요성을 확인하기 위해 똑같은 pre-training data, fine-tuning scheme, and hyperparameters as BERT(BASE)를 사용하여 두 개의 pre-training 모델을 평가함.
- NO NSP: masked LM(bidirectional model) 사용, “next sentence prediction’ task는 수행x
- LTR & NoNSP: left-context-only-model (left-to-right) (mlm x)
- 1. NSP task의 영향에 대해 조사.

Table5에서 NSP를 없앤 것이 QNLI, MNLI, SQuAD 1.1에서 성능이 떨어지는 것을 확인할 수 있었음.

- 2. Training Bidirectional representations의 영향

LTR, 즉 bidirectional하지 않은 것이 MRPC, SQuAD에서 크게 성능이 떨어지는 것을 확인함. -> SQuAD에서 LTR의 성능이 매우 안 좋은 것을 확인했음. 하지만 randomly initialized BiLSTM을 추가했을 때 LTR의 성능이 향상되었지만 MLM 보다는 그렇지 못함.

- 마지막 단락에서 LTR, RTL를 각각 train하여 합치는 방법을 언급함. 이 방법은 하나의 모델을 train하는 것보다 비싸고, 직관적이지 않고, deep bidirectional model에 비해 less powerful함.

5.2 Effect of Model Size

- 모델의 사이즈가 fine-tuning task의 정확도에 어떤 영향을 주는지 확인함.
- > Table6를 확인해보면, 모델의 크기가 클수록 성능이 향상됨. 사전 훈련이 충분하게 되었다면, 모델의 크기가 클 때 small scale task에서도 큰 성능의 향상을 가지고 옴.

5.3 Feature-based Approach with BERT

- BERT를 feature-based 방식으로 사용했을 때의 장점.

1. 모든 tasks가 transformer encoder architecture로 쉽게 표현되지 못해, task-specific model architecture를 추가해야한다.

2. Train data의 비싼 representation을 미리 계산하고 저렴한 모델들로 많은 학습을 하면 비용이 줄어듦.

BERT는 fine-tuning & feature-based 접근 모두에서 효과적임.

6. Conclusion

최근 언어모델의 전이 학습에 대한 개선들이 비지도 사전 학습에서 정의됨.

특히 깊은 단방향 아키텍처의 장점도 가지고 있음

양방향 아키텍처를 일반화하는 게 가장 큰 포인트

NLP 태스크의 방대한 양의 데이터에 대해서 같은 사전 학습된 모델 허용.