양방향 encoder transformers에 기초됨.

Label 없는 text로부터 깊은 양방향 표현 사전훈련함

이로인해 아키텍쳐 수정없이 질문 답변, 언어추론 같은 광범위한 작업 가능

기존꺼는 단방향이라 학습하는데 제한이있음

gpt의 경우 left to right 이전 self atteintion의 tokens을 받는 구조임

masked language model은 왼쪽과 오른쪽 맥락을 융합할 수 있게해서, 더 깊은 양방향 transformer 학습을 가능하게 해줌

뿐만 아니라, 다음 문장 예측도 가능하게 해줌

Melamud lstm으로 양방향 구현

ELMo 양방향 임베딩 방법

처음 지정할(배울?) 매개변수가 거의 없다

라벨 없는 텍스트로 사전훈련이 되고, downstream(?) 작업에 맞게 조정됨

Pretraining – fine tuning 2 section으로 설명

양방향 transformer encoder에 베이스됨, tensor2tensort library에 released 되어있음

Layer L,

hidden size H,

self-attention head : A

2개 모델 보고함

BERTBASE (L=12, H=768, A=12, Total Parameters=110M)

BERTLARGE (L=24, H=1024, A=16, Total Parameters=340M).

Bertbase는 openai gpt 와 비교를 위해 같게 설정

주의점은 gpt는 left to right지만, 우리는 양방향임

단일 문장 or question & answer가 포함된 연속된 문장 고루고루

Sequence는 입력 토큰으로, 한문장 or 두문장이 함께 포함된 것

Wordpiece embeddings 3만개 어휘 토큰 가진 것 사용

첫번째 토큰은 구분 위해 [CLS] 사용

1. [SEP]로 구분

2. 학습된 임베딩을 문장 A에 속하는지 B에속하는지에 대한 토큰 추가

입력임베딩 E

Finl hidden vector of the special token == C ∈ R^H

i번째 입력에 대한 final hidden vector == T(i) ∈ R^H

텍스트, 키보드이(가) 표시된 사진

자동 생성된 설명

Train 방법

#masked LM

깊은 양방향 모델을 학습하기 위해 토큰의 일부를 무작위로 마스킹한 다음, 마스킹 된 토큰을 예측하는 방법을 사용

문헌에서는 (1953 taylor) cloze task 라 부르지만, 우린 이 기법을 MLM(Masked LM) 이라 부를거임

전체 토큰의 15% 마스킹 함, auto encoder의 noise 제거와는 다르게, 전체 입력을 재구성하지 않고 마스킹 된 단어만 예측함.

이를 통해 양방향 pre-train 모델 얻지만, pre-triain과 fine-tuning에 mask 토큰의 일치가 일어남

이를 완화하기 위해 masking된 단어를 실제 mask 토큰으로 바꾸지 않음??

i번째 토큰 선택시, 80 of the time??인 mask 토큰으로 변경

변경되지 않은 i번째 토큰의 10% of the time의 10%인 임의 토큰으로 교체?

If the i-th token is chosen, we replace the i-th token with (1) the [MASK] token 80% of the time (2) a random token 10% of the time (3) the unchanged i-th token 10% of the time. Then, Ti will be used to predict the original token with cross entropy loss. We compare variations of this procedure in Appendix C.2.

단일 언어 말뭉치에서 세번째로 생성될 수 있는 이진화된 다음 문장 예측작업을 위해

Pretrain sentence 만들때,

A와 B선택할 때 50%는 실제 A뒤에 오는 B이고

50%는 임의 문장으로 구성

기존 Next Sentence Prediction은 이전 작업에서 문장 임베딩만 downstream 식으로 전송됨

Bert는 모든 매개변수를 전송하여 최종 작업 모델 매개 변수를 초기화 시킴

#Pre-training data

우리는 bookCorpus(800백만 단어) 영어 위키백과 (2500백만 단어) 사용

#fine-tuning Bert

Transformer의 self-attention은 bert가 적절한 입력과 출력을 교환하여 단일 텍스트 또는 텍스트 쌍을 포함하는 많은 다운스트림 작업을 모델링할 수 있도록 하기 때문에 미세 조정이 간단합니다.

텍스트 쌍을 포함한느 경우, 양방향 cross attention 적용 전에 텍스트 쌍을 독립적으로 인코딩\

조정되는 매개 변수들

(1) sentence pairs in paraphrasing, (2) hypothesis-premise pairs in entailment, (3) question-passage pairs in question answering, and (4) a degenerate text-∅ pair in text classification or sequence tagging.

사전훈련에 비해 fine-tuning의 비용은 적어서 빨리됨.

#GLUE

The General Language Understanding Evaluation (GLUE) benchmark 사용

Log(softmax(CW^T))사용해서 표준분류손실 계산

glue에서 32batch size, 3epochs 사용

learning rate는 5e-5 ~ 2e-5에서 좋은거 사용

squad v1.1 (100k question/answer pairs) 사용해서

3-epochs, 5e-5 learning rate, 32 batch size로 fine-tune했음

Squad로 fine-tuning하기전 Trivia qa사용해서 fine tuning 했음

텍스트, 영수증이(가) 표시된 사진

자동 생성된 설명

사람이김

Squad 2.0

텍스트, 영수증, 스크린샷이(가) 표시된 사진

자동 생성된 설명

단답 존재 여부를 무시해서? -> 문제를 현실적으로 만듬

Learning rate 5e-5, 48 batch size 사용

#swag

Situatuins with adversarial generations dataset contains 113k sentence-pair

3epochs, 2e-5, 16 batch size

ESIM+ELMo 보다 27.1% 좋고, GPT보다 8.3% 좋음

#Ablation studies

테이블이(가) 표시된 사진

자동 생성된 설명

No nsp는 다음 문장 예측 작업 없이 훈련

LTR&No NSP 는 GPT 처럼 문장 예측 없이 left to right 학습 방법

* 성능이 크게 저하됨
* Squad의 경우 ltr모델이 없어지자, 크게 떨어짐 이를 위해 bilstm 추가하자 크게 개선되지만, 사전 훈련 양방향 모델 보다 훨씬 안좋음
* Bilstm이 GLUE 작업 성능 크게 저하시키기 때문

+BiLSTM은 말 그대로.

# effect of model size

더 큰 모델이 성능 개선에 영향을 끼친다.

For example, the largest Transformer explored in Vaswani et al. (2017) is (L=6, H=1024, A=16) with 100M parameters for the encoder, and the largest Transformer we have found in the literature is (L=64, H=512, A=2) with 235M parameters (Al-Rfou et al., 2018). By contrast, BERTBASE contains 110M parameters and BERTLARGE con- tains 340M parameters.

버트 base는 110만개, bert large 는 340만개 파라미터

Pre train이 충분한경우 모델 크기를 확장시키면 작은 규모의 데이터셋에서도 큰 개선이 일어난다고 함. (peters et al. 2018)

하지만 hidden state를 200->600은 개선이 있었지만, 600->1000은 개선이 없었다.

테이블이(가) 표시된 사진

자동 생성된 설명

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

적당한게 좋다.

테이블이(가) 표시된 사진

자동 생성된 설명

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

Cocat last four hidden이 가장 좋은 효과