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

Pre-training of Deep Bidirectional Transformers for Language Understanding(BERT).... 757.0...

1. introduction

language model 인 pretraining language model 은 자연어 처리의 성능을 향상 시켰다. plm을 적용하는 방법은 2가지로 구분된다. feature-based, fine-tuning이다. Feature-based 방식의 경우, (대표적으로 ELMo) 특정한 task에 알맞는 구조를 보유하고, pre-trained representation을 특성으로 추가해서 사용하게 된다. 반면에 fine-tuning 방식의 경우 (대표적으로 GPT), 특정 task에 특화된 파라미터를 최소화 하고, 사전 학습된 파라미터를 fine-tuning 한다. 두 방식 모두 pre-training 하는 과정에서는 동일한 objective를 사용하고, 동일하게 unidirectional language model을 사용한다. 하지만 이런 방식은 성능을 제한한다. 따라서 이 논문에서는 fine-tuning 방식을 개선하는 BERT (Bidirectional Encoder Representations from Transformers) 를 제안한다.BERT는 masked language model (MLM) 을 사용해서 성능을 향상시킨다.

2. Unsupervised Feature-based Approaches

단어 임베딩을 학습하기 위해, letf-to-right language modeling 이나 등의 objective 로 학습을 진행했다. ELMo 는 2개의 biLM(left-to-right LM, right-to-left LM)을 이용해서 문맥을 고려한 단어 임베딩을 제공한다. 한 단어의 representation을 left-to-right representation과 right-to-left representation을 연결해서 제공한다.

3. Unsupervised Fine-tuning Approaches

최근에는 레이블 되지않은 텍스트를 통해 pre-trained 된 contextual token representation을 생성하고, supervised downstream task에서 fine-tunin 하는 방식이 . 장점은 시작부터 학습이 완료될 때까지 적은 수의 파라미터가 학습된다는 것이다.

4. Transfer Learning from Supervised Data

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

5. BERT

BERT 는 크게 pre-training 단계와 fine-tuning 단계, 두가지 단계로 나뉜다. Pre-training 단계에서는 레이블링 하지 않는 데이터를 기반으로 학습을 진행한다. Fine-tuning 과정에서 모델은 우선적으로 pre-train된 파라미터로 초기화된다. 이후 모델을 레이블링된 데이터로 fine-tuning 한다. 실제 task에서 사용하는 모델은 초기에 동일한 파라미터로 시작하지만, 마지막은 seperated된 fine-tuned 된 모델을 보유하게 된다. BERT 는 pre-trained 된 모델과 fine-tuned 된 모델 사이의 구조적 차이가 거의 없게 된다.

6. Model Architecture

BERT는 multi-layer bidirectional Transformer encoder를 사용한다.

- Transformer block layers = L
- hidden size = H
- number of self-attention heads = A

이번 논문에서는 크게 두 가지 종류의 모델을 크기를 기준으로 제안한다.

- BERTBASE : L=12,H=768,A=12,totalparameter=110M

  L=12,H=768,A=12,totalparameter=110M
- BERTLARGE : L=24,H=1024,A=16,totalparameter=240M

BERTBASE 의 경우 OpenAI 의 GPT 와의 비교를 목적으로 동일한 크기로 제작했다. BERT 의 경우는 현재 토큰의 좌/우를 모두 참조할 수 있는 bidirectional self-attention을 수행하는 반면에, GPT 의 경우는 현재 토큰의 왼쪽에 있는 문맥만 참조가 가능하다는 차이점이 있다.

7. Input/Output Representations

단일 문장과 쌍으로 구성된 문장을 모두 표현할 수 있다. "sequence" 은 앞으로 단일 문장 또는 쌍으로 이루어진 문장을 말한다.단어 임베딩으로는 WordPiece embedding 을 사용하게 된다. Sequence 의 첫 토큰은 [CLS] 사용한다.모든 입력 토큰에 대해서 token segment와 해당 토큰의 position embeddin를 더해서 Input representation 이 생성된다.

8. Pre-training BERT

전통적인 left-to-right/right-to-left LM을 사용해서 pre-train하는 ELMo, GPT와는 다르게, BERT는 2개의 unsupervised task를 이용해서 학습하게 된다.

9. Task #1 : Masked LM

상식적으로 left-to-right LM이나, right-to-left LM, 또는 두 결과를 연결한 결과보다 deep bidirectional model 이 성능이 더 좋다고 이해할 수 있다. 하지만, 전통적인 conditional LM의 경우 left-to-right/right-to-left 의 방식으로 밖에 수행하지 못한다. Bidirectional 하게 처리하면 간접적으로 예측하려는 단어를 참조하게 되고, multi-layered 구조에서 해당 단어를 예측할 수 있기 때문이다. 이러한 문제를 해결하기 위해서 BERT는 일정 비율의 token을 mask하게 된다. (이러한 이유로 masked LM 이라고 부른다.) 이번 논문에서는 전체 토큰의 15%를 mask 하게 된다.최종적으로 cross-entropy loss 를 사용해서 기존의 토큰을 예측하도록 학습된다.

10. Task #2 : Next Sentence Prediction (NSP)

Question-answering(QA), Natural Language Interference(NLI) 등의 task는 두 문장 사이의 관계를 이해해야 하는 task이다. 이러한 특징은 LM을 통해서 학습하기 쉽지 않기 때문에, NSP라는 task에 대해서도 함께 학습을 진행한다.

11. Pre-training Data

Pre-training corpus로는 다음과 같은 데이터를 사용했다. Wikipedia의 경우 text passage 만 사용했고, 목록이나 표 등은 모두 제외했다. 긴 문맥을 학습하기 위해서 Billion Word Benchmark 와 같이 섞인 문장으로 구성된 데이터를 사용하지 않았다.

12. Fine-tuning

Fine-tuning하는 방법은 task에 알맞는 입력과 출력을 모델에 입력으로 제공해서 파라미터들을 해당 task에 맞게 end-to-end로 업데이트한다.

13. Experiments

-No NSP : masked LM(MLM) 으로만 학습되고 NSP는 사용하지 않는 경우. 이런 경우 QNLI, MNLI, SQuAD 1.1에서 성능이 떨어지는 것을 확인할 수 있었다.

-LTR & No NSP : left-context-only model 을 사용하고 NSP도 사용하지 않는 경우. No NSP와 함께 bidirectional representation이 아닌, LTR을 사용하는 경우 모든 task에 대해서 MLM에 비해 성능이 떨어지고, 특히 MRPC와 SQuAD 에서 큰 폭의 성능 저하를 확인할 수 있었다. bidirectional 한 요소를 제외해서 나타난 성능 저하인지 확인하기 위해서, 임의로 초기화된 BiLSTM을 추가했을 때 SQuAD의 성능이 상당히 향상된다는 것을 확인할 수 있었다. (pre-trained bidirectional model 에 비해서는 성능이 떨어진다.)

-ELMo 처럼 LTR과 RTL 모델을 각각 학습해서 representation을 합치는 방식을 생각해볼 수 있다. 하지만 이러한 방식은 bidirectional model 에 비해 2배의 비용이 든다는 단점이 있다.