BERT: Bidirectional Encoder Representations from Transformer

Pre-training of Deep Bidirectional Transformers for Language Understanding

- transformer를 활용한 language representation
- transfer-learning 방식으로 학습. 즉, 대용량 unlabeled data로 pre-training 후, 특정 task를 가지고 있는 labeled data로 fine-tuning.
- GPT처럼 unidirectional 하거나, ELMo처럼 shallow bidirectional 한 모델과 달리, BERT는 deeply bidirectional 한 모델.

  * unidirectional: 텍스트를 한 방향으로만 처리. 어떠한 같은 모델은 텍스트를 왼쪽에서 오른쪽으로 읽음. 이러한 모델은 주어진 단어의 오른쪽 컨텍스트(이후의 단어들)는 고려하지 않으며, 주로 생성적인 task(ex. 텍스트생성)에 적합.

  * shallow bidirectional: 두 개의 분리된 레이어를 사용하여 텍스트를 양방향으로 처리. 하나의 레이어는 왼쪽에서 오른쪽으로, 다른 하나는 오른쪽에서 왼쪽으로 텍스트를 읽음. 그러나 이 두 레이어는 서로 독립적으로 작동하며, 각 레이어는 텍스트의 한 방향의 컨텍스트만을 고려. ELMo는 이 두 레이어의 출력을 결합하여 더 풍부한 단어 표현을 생성.

  * deeply bidirectional: 모든 레이어에서 양방향의 컨텍스트를 동시에 고려. 이는 각 단어가 그보다 앞뒤 모든 단어(왼쪽과 오른쪽 모두)의 컨텍스트를 바탕으로 이해될 수 있게 만듦. BERT는 이러한 깊은 양방향 특성으로 인해 텍스트의 의미를 더 정확하게 파악하고, 다양한 NLP 태스크에서 뛰어난 성능을 보임.
- BERT에는 pre-training 과정과 그 이후의 fine-tuning 과정이 있음.

  * pre-training에서 unlabeled 된 corpus를 transformer의 encoder architecture와 MLM과 NSP 과정을 통해서 feature가 나옴.

  * 이 feature를 task specific 하게 적용해서 fine-tuning을 통해 모델링을 하는 것.
- BERT는 모델의 크기에 따라 base 모델과 large 모델로 나뉨.

  * BERT_base: L=12, H=768, A=12, Total Parameters=110M

  * BERT_large: L=24, H=1024, A=16, Total Parameters=340M

  (L: transformer block의 layer 수, H: hidden size, A: self-attention heads 수)

Input/output Representations

- BERT의 input은 세 가지 embedding 값의 합으로 이루어져 있음.
- 모든 sentence의 첫번째 token은 [CLS](special classification token)임.

  * 문장 전체의 의미를 나타내는 역할.

  * classification을 위한 embedding.

  * classification task일 경우 cls token 위치에 해당하는 hidden unit output을 feature로 생각하여 classification 진행. 아닌 경우 무시.
- 문장 구분

  * [SEP] token 사용.

  * segment embedding 사용: 앞의 문장에는 sentence A embedding, 뒤의 문장에는 sentence B embedding을 더해줌. (모두 학습된 값)

Pre-training

① Masked Language Model(MLM)

- 무작위하게 몇 개의 토큰을 mask 시킴.
- transformer 구조에 넣어서 주변 단어의 context만을 보고 mask된 단어를 예측.

② Next Sentence prediction

- 두 문장을 pre-training 시킬 때 같이 넣어서 두 문장이 이어지는 문장인지 아닌지 맞추는 것.

fine tuning

- pre-training을 거친 모델을 가지고 downstream 한 방법을 적용하여 fine tuning 시행.

* 위 과정을 제외하고 pre-training 때의 hyper parameter 와 대부분 동일함.

(추천) batch size : 16, 32 / learning rate (Adam) : 5e-5, 3e-5, 2e-5 / # of epochs : 2, 3, 4