**BERT(Devlin 2018)**

<https://ebbnflow.tistory.com/151>

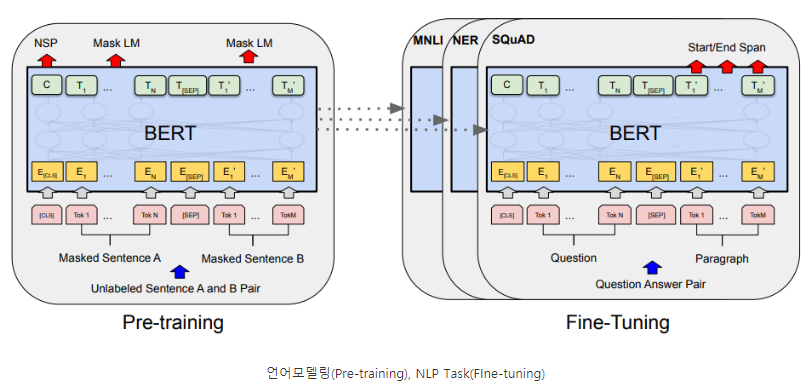
Pre-training of Deep Bidirectional Transformers for Language Understanding

모든 자연어 처리 tasks에서 좋은 성능을 내는 범용 LM

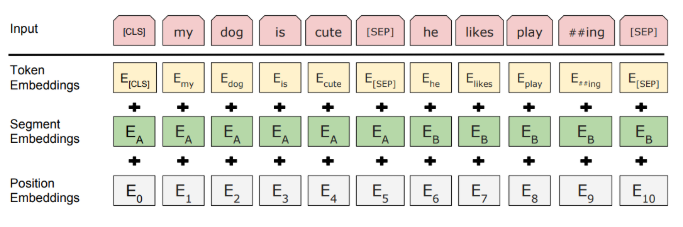
LM은 왜 오직 Left or right context를(unidirectional, ELMo도 그저 단방향을 두 번 이용한 것일 뿐이다.) 사용? Bidirectional도 필요하다.

Directionality는 좋은 확률 분포 생성에 있어서 꼭 필요하며 bidirectional 인코더에서 “see themselves” 해야 한다.(self-attention 인가??)

이것 역시 ELMo 처럼 word embedding을 위한 LM인데 그 모델로 Transformer를 이용한 것



그럼 저 그림에서 BERT라고 쓰여진 부분이 모델의 부분이다.

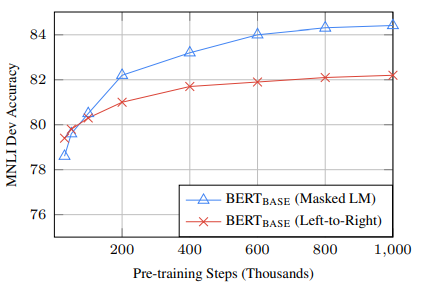


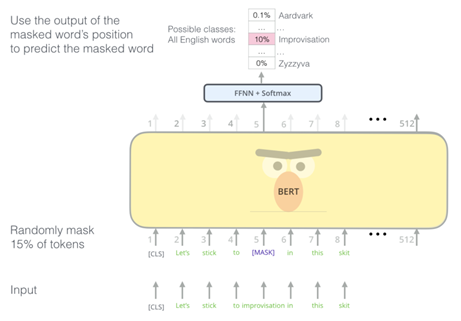
Token Embedding: Word Piece (OOV, sub-word, char)

Segment Embedding: sub-word를 하나의 문장으로 다시 묶는 정체성을 주는 과정(하나의 세그먼트에서 보통 sub-word 최대 60개 안 넘어감, 제한을 512개로 설정했고 문장 구분을 위해 입력 시 [SEP]를 이용한다.

Position Embeddings: BERT는 Transformer의 인코더 부분만 이용하는데 따라서 Position 정보가 필요하다.

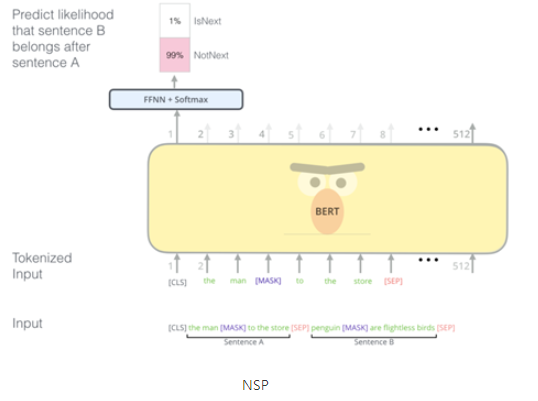
이제 LM을 실행하면 된다.(왼쪽에서 오른쪽으로 다음 단어 예측 기계 or 예측할 단어 좌우 문맥 고려하여 예측(word2vec처럼)) 그런데 BERT는 언어의 특성을 잘 학습하도록 MLM, NSP 이용





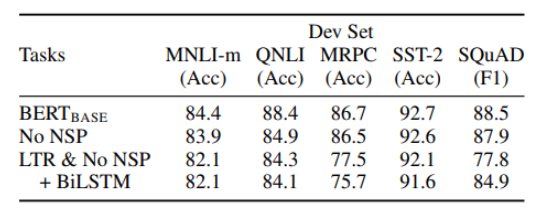
MLM(Masked Language Model): 입력 문장에서 임의로 토큰을 버리고(Mask), 그 토큰을 맞추는 방식으로 학습을 진행

Too little masking: Too expensive to train / Too much masking: Not enough context



NSP(Next Sentence Prediction): 두 문장의 순서를 예측(두 문장이 주어졌을 때), NLI(Natural Language Inference)나 QA 같은 Task를 위해 필요하며

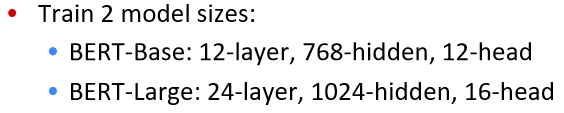
아래는 Ablation Study

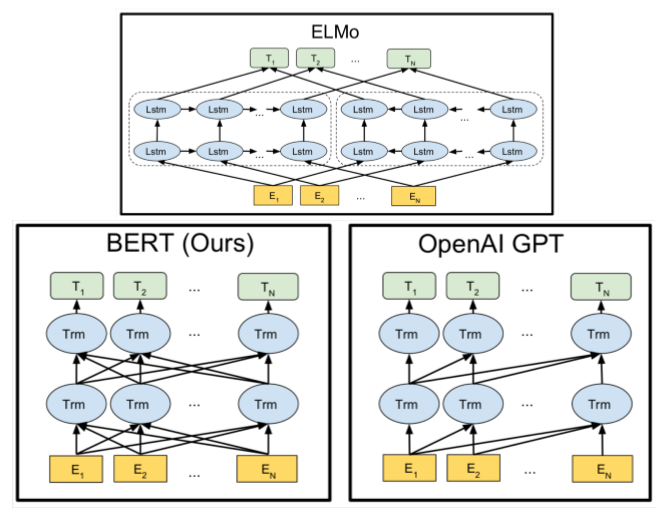


No NSP: MLM만 적용

LTR & No NSP: MLM과 NSP 둘 다 적용하지 않은 기존 좌우 모델

BERT base(MLM + NSP)

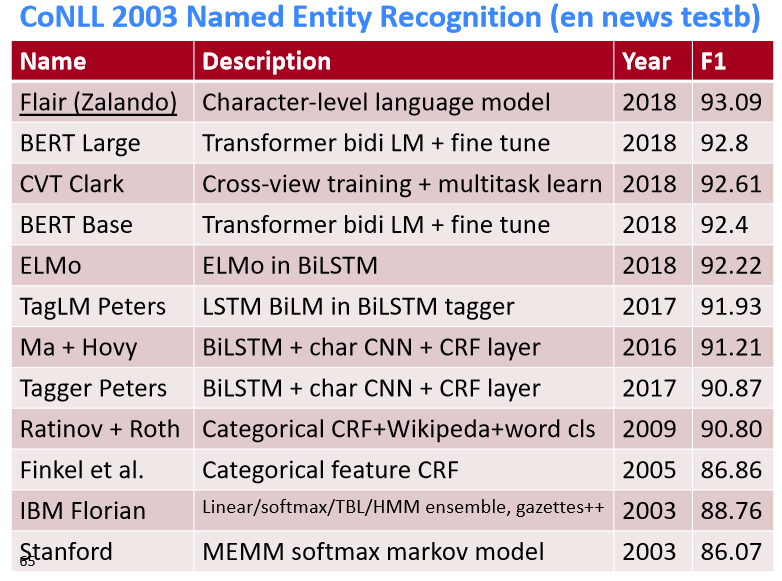


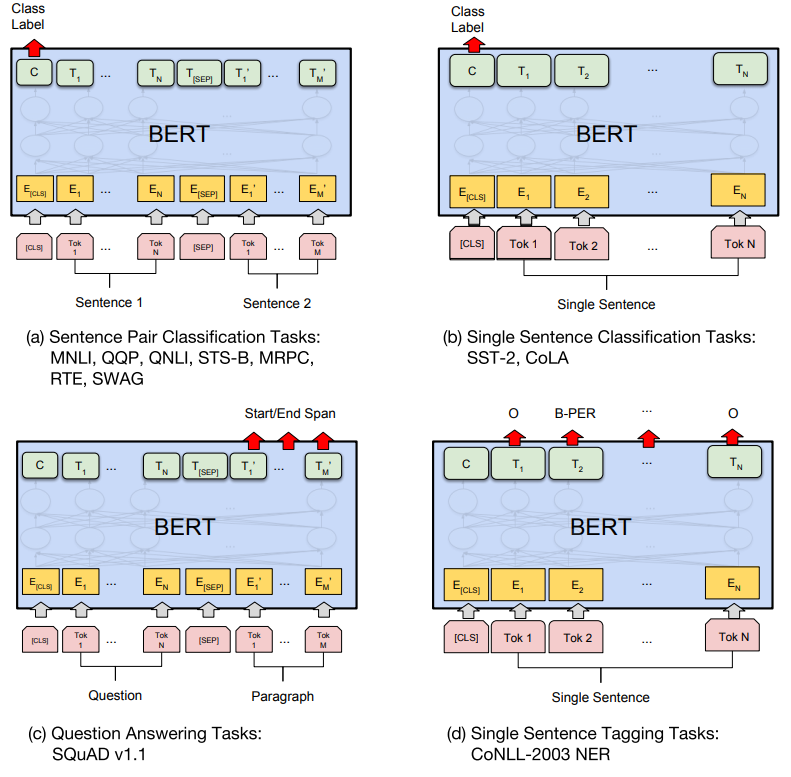


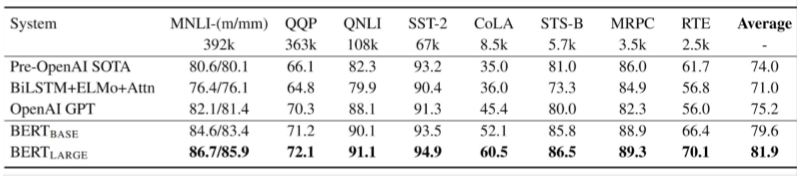
학습된 언어모델을 다양한 Task에 수행한다.

BERT의 한계점: 일반 NLP에서는 잘 작동하지만 아직 Bio, Science, Finance 등의 특정 도메인에서 잘 적용이 안된다. 사용 단어들이 다르고 언어의 특성이 다르기 때문이다.

따라서 특정 분야의 특성을 포함한 언어 데이터로 finetuning이 필요하다.







BERT(Bidirectional Encoder Representations from Transformers), 2018 google.

모든 자연어 처리 Tasks에서 좋은 성능을 내는 범용 LM



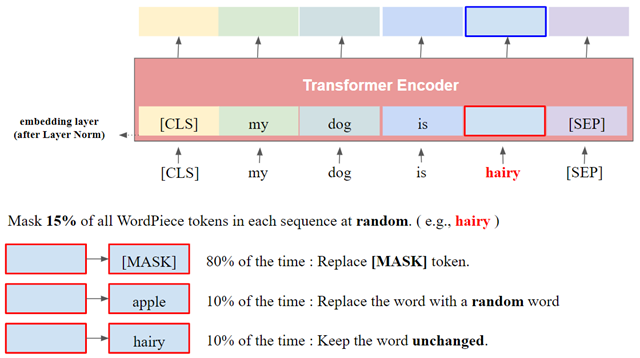
KorQuAD의 Ranking. 가볍게 인간을 이기며 1~10위 모두 BERT를 사용하고 있다.

MLM

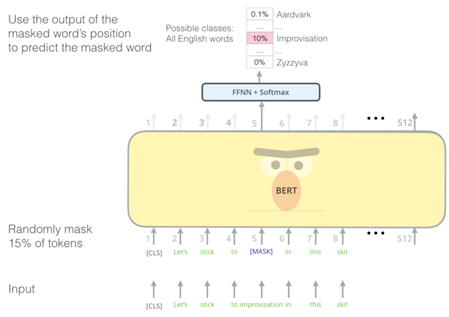
입력 문장에서 임의로 Token을 가리고 그 Token을 맞추는 학습 방식

BERT에서는 sub-word 단위로 token을 마스킹 하는데 이후 변종들에서는 whole word masking, 한국어에서는 형태소 단위 masking을 하곤 했다.

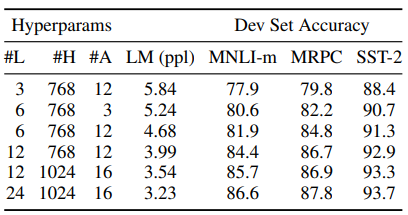
Mask하는 입력 단어는 랜덤하게 진행된다. => 경우에 따라 예측하는 차이가 있을 것으로 예상 => 그래서 앙상블을 하게 되는 것이 아닐까?



Fine-tune 때에는 올바른 예측을 돕도록 마스킹에 노이즈를 섞음?

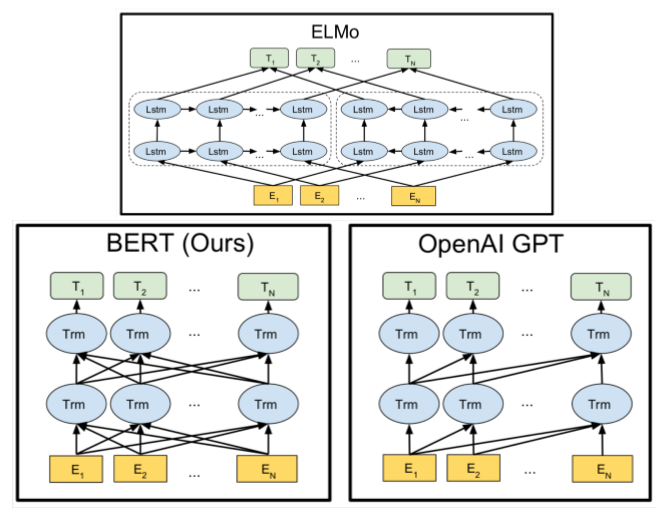


위 그림처럼 WordPiece의 vocab은 30522로 softmax가 class를 3만개 가진다.



하이퍼 파라미터에 따라..(하지만 리소스 문제가 있겠지..?)

왜 근데 이게 Bidirectional인지 궁금..



ELMo: 단지 left-to-right, right-to-left 독립적으로 사용  
GPT1: unidirectional Transformer decoders(디코더의 특징?? 트랜스포머를 보면은 앞에 나온 단어만 처리하고 아직 안 나타난 단어는 mask 처리 -> left-to-right 일 수 밖에 없나..?) 그리고 인코더와 함께하는 어텐션 부분은 사라졌음.

BERT: Bidirectional Transformer encoders