**RoBERTa: A Robustly Optimized BERT Pre-Training Approach (GLUE 벤치마크 4등)**

Pre-Training에서 많은 시간 & 리소스 소요 => BERT 및 이후 접근법들을 엄밀하게 비교하기 힘듦(테스트 정확도가 높더라도 시간과 리소스가 많이 들면 좋지 않다고 볼 수 있는 그런 뜻?)

또한, 어떤 Hyper Parameter가 결과에 많은 영향을 미쳤는지 검증하기 힘듦

여러 실험을 통해 데이터의 양 및 Key-Hyper Parameter의 영향을 분석할 것

**Main Idea**

BERT는 아직 UnderTrain이다.

Pre-Training 시에 Hyper Parameter 튜닝으로 더 좋은 결과 가져올 수 있다.

1. Dataset: BERT에 비해 더 많은 데이터 & 더 많은 Epoch & 더 큰 배치 사이즈
2. Objective: NSP Task 과정은 제거 (왜?)
3. Downstream Task 를 대비하여 0.1 확률로 최대 길이보다 더 짧게 인위적으로 세그먼트 길이를 조정 했었는데 이 로직 제거, 그냥 최대로만.
4. BERT의 static masking이 아닌 입력을 만들 때 마다 Masking을 다시 ! (Dynamic Masking)

**Background – BERT**

두 개의 segment와 special tokens로 구성 = BERT의 입력 총 문장

CLS Token: classification task의 representation을 위한 토큰

SEP Token: 두 segment 분리하는 토큰

Dataset은 영어 위키피디아 & Book Corpus (16GB)

**MLM**

기존 BERT에서는 만약 40 Epoch라면 data A에 대해서 같은 부분을 Masking 했다.  
하지만 RoBERTa는 duplication factor 를 둔다. 만약 그 값이 10이라면 모델은 data A에 대해서 다른 경우의 masking 10 상황을 만들게 해서 각 상황을 40 Epoch / 10 duplication factor 만큼 보게 된다.

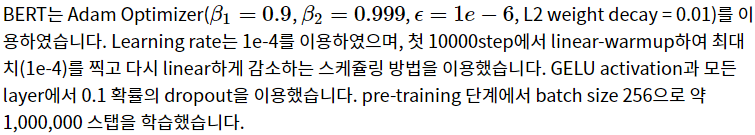
기존 BERT에서는 미리 dataset에 마스킹을 해놨다.

**NSP**

두 개의 segment가 자연스러운지 따진다. (NLI & text similarity 고려)

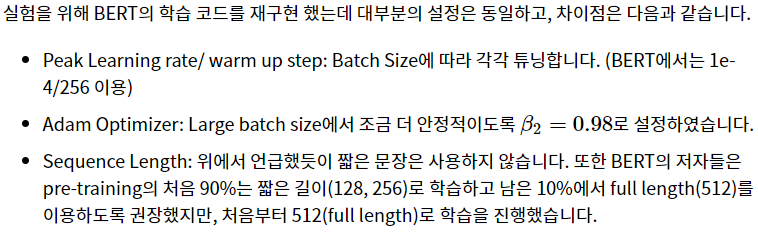
Positive sample( 같은 문서에서 연속적으로 segments를 가져옴 )  
Negative sample( 다른 문서에서 segments를 가져옴 )

**Optimization**



**Experiments**

**Reimplementation**

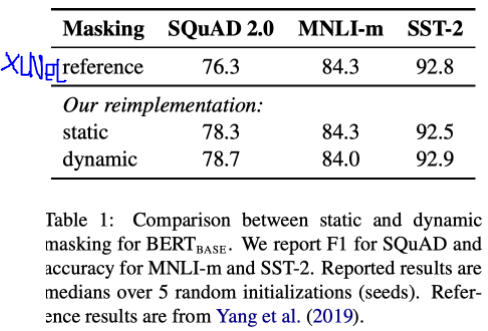


Adam Optimizer 구조 잘 모름

**Dynamic Masking**

더 많은 step & dataset일수록 Dynamic Masking은 중요하다. 동일한 환경에서 Static vs Dynamic Masking을 구분해서 실험했다.

Static Masking에 비해 비슷하거나 조금 향상되었다.



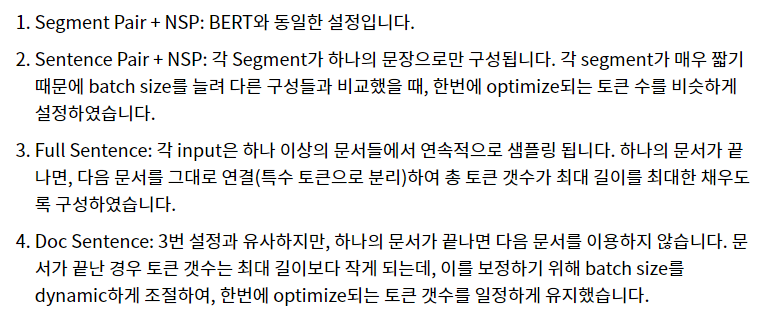
**Model Input Format & NSP**

기존 BERT는 Pair Segment 단위의 Task들을 위해 NSP 추가했다.

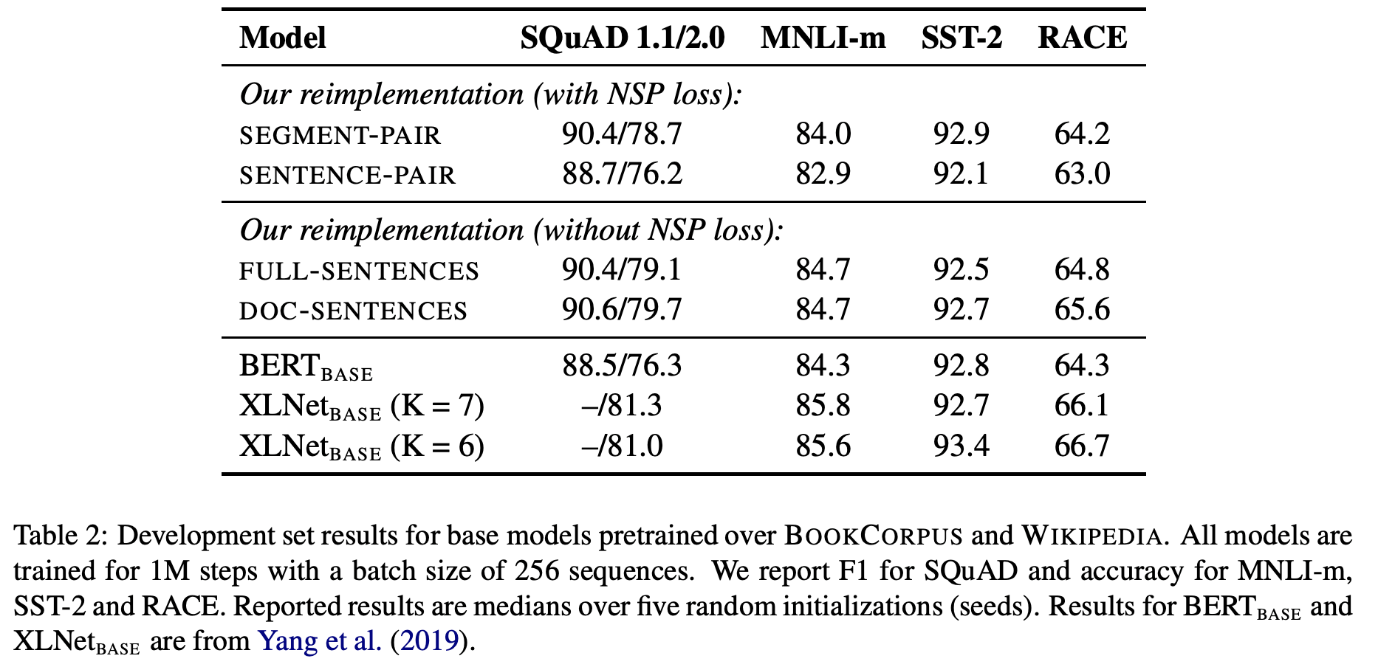
해당 논문의 ablation study에서는 NSP가 Pair 단위 task들에서는 효과가 있었다.

하지만 NSP Ablation 에서는 단지 NSP Loss만 제거할 뿐 두 개의 segment가 입력으로 들어가는 구조는 유지했기 때문에 제대로 제거를 했다고 볼 수 없다. ( 즉, 입력 자체도 두 segment를 주지 않아야 NSP를 쓰지 않아 성능이 떨어졌다고 볼 수 있다. ) = BERT 입장에서는 입력 자체는 유지하는게 더 좋았을 거라 생각

하지만 NSP Loss 제거 + 입력 자체 구성 변경하니까 오히려 성능 좋아짐(pair task에 대해서)



3. 은 1 segment + positive sample 느낌이고 4. 는 2 segment + negative sample 느낌



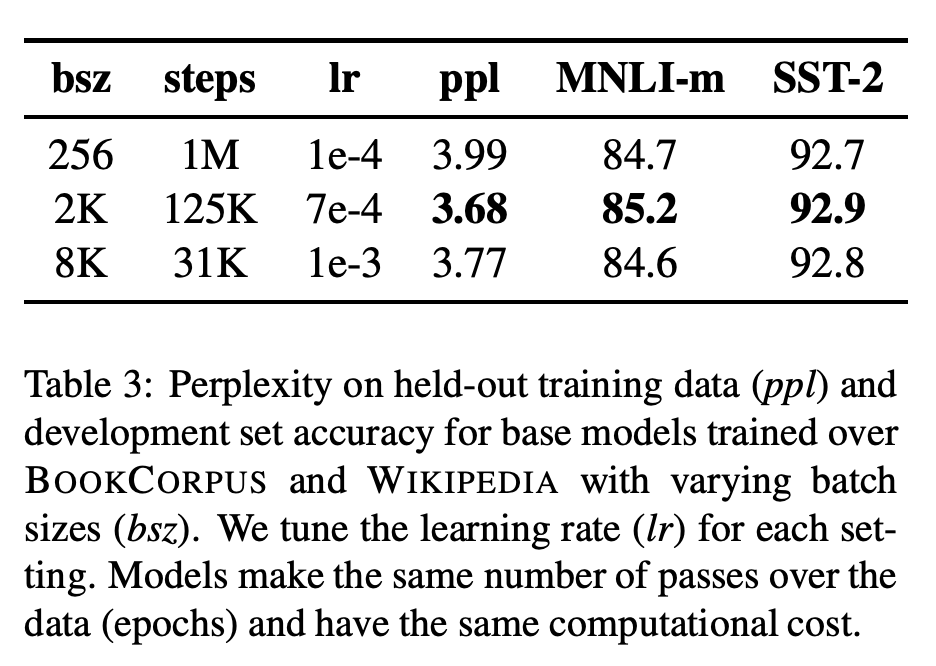
XLNet 은 dataset이 더 컸기 때문에 성능이 좋은 것으로 예상

Training with Large Batches

Batch size에 맞는 Learning Rate를 이용한다면 => optimization 속도 & 성능 향상

Pre-tarin 자체의 LM & downstream task의 성능 같이 향상

기존 BERT의 계산 비용을 유지하면서 다른 방식(배치 사이즈 늘림)을 하며 실험 진행했다는 소리



**Text Encoding**

기존 BERT의 WPM에서 그냥 BPE 사용?

50k의 사전 사이즈, 사실 character 단위가 더 좋은 성능 but universal encoding 이점 때문에 byte 단위를 이용하기로 했다.

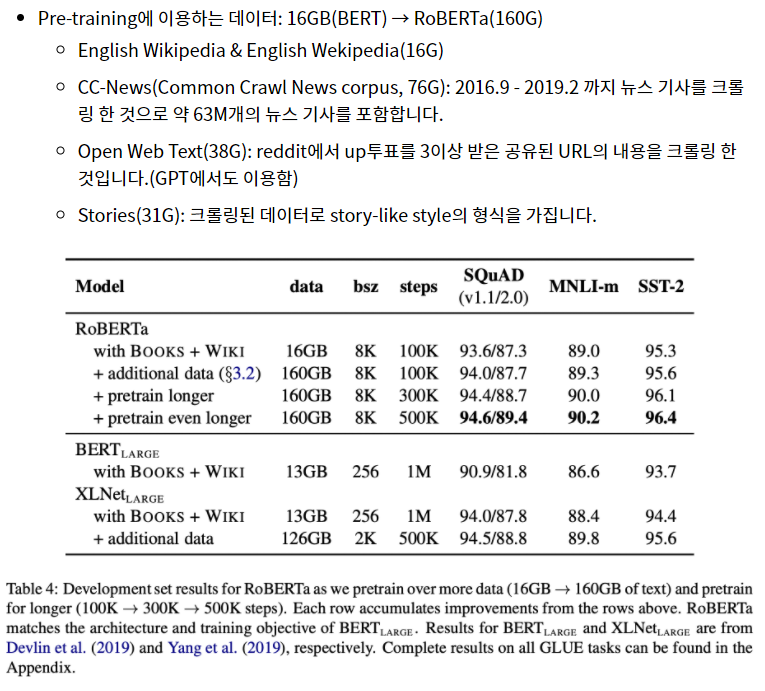
질문 => 결국 BPE도 사전을 만들 뿐 토큰을 one-hot 후에 embedding 하는 건 word2vec을 이용하는 건가..?

**RoBERTa**

아래 그림은 pre-train 데이터셋

=> 더 많은 & 더 오래 => 성능 향상

아직 모델이 underfit에 가깝다는 소리



Bemchmark Result

Single model 설정: 모든 task에 대해 single-training, 다른 논문과 유사하게 적절한 hyper parameter, BERT & 대부분의 논문 = 3 epoch But RoBERTa는 10 epoch + early stopping

Ensemble model 설정: 더 넓은 hyper parameter 범위에서 5~7개 모델들을 학습해 앙상블

또한, Multi-task Finetuning을 이용하면 성능이 오르지만 dataset을 다시 구성? 해야 하니 안함.

또한, RTE, STS, MRPC는 pre-training 모델 시작보다 MNLI로 fine-tuning한 것을 다시 fine-tuning 하면 성능이 더 좋다.(아마 pair task 중 MNLI가 데이터 양 많아서.)

아래 결과 처럼, single 에서는 SOTA(현재 GLUE는 조금 바뀜)

ensemble에서는 몇 개 높음, 평균은 1등

RoBERTa는 BERT 와 모델 구조 같고 단지 Masked-LM objective를 이용(단지 방식만 바꿈) => 다른 모델 구조인 XLNet & objective 변경한 XLNet, StructBERT(ALICE) 보다 능가

하지만 현재 StructBERT(2019 10 19) 가 능가.

즉 모델 구조나 training objective 보다 학습시간 & 데이터 양이 더 중요(XLNet을 robust하게 바꾸면?? 어떨까), SQuAD나 RACE에서도 SOTA

