<https://baekyeongmin.github.io/paper-review/structbert-review/>

StructBERT: Incorporating Language Structures into Pre-Training for Deep Language Understanding (Alibaba Group)

ALBERT에 이어 GLUE 벤치마크 89.0 (2위) – 옛날에 (블로그 글 당시 2019/10/14)

BERT pre-training 단계에서 문장 내부와 문장들 사이의 구조적 정보를 학습하도록 하는 새로운 전략 제시

**Main Idea**

1. Pre-training 단계에서 특정 토큰들 및 문장들의 순서를 예측하는 문제 추가  
   -> 잘 정제된 단어 구조와 문장간의 구조 학습
2. LM은 랜덤한 순서의 토큰들이 주어졌을 때 올바른 순서로 복구할 수 있어야함.  
   BERT로는 그것이 어려움(토큰들의 sequential 순서 & 토큰들의 고차원 의존성을 정확히는 모델링 못함) => 새로운 토큰(단어) 단위의 objective 설정
3. NSP는 97~98% 정확도(너무 쉬워서) => 확장해서 여러 문장들(segments) 사이의 관계를 모델링하는 objective 설정

**Model Architecture**

[CLS] + segment1 + [SEP] + segment2 + [SEP] / segment embedding / positional embedding / 모델 구조 => 기존 BERT와 동일

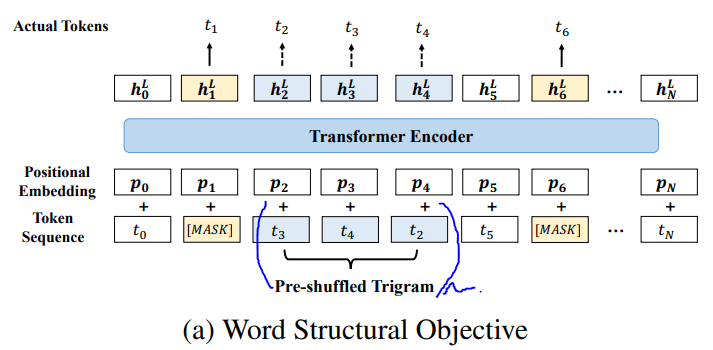
**Word Structural Objective**

단지 Masked LM의 Objective가 바뀜

기존에 BERT는 Mask 토큰이 원래 뭐였을까? 복원하는 문제였다. 이 문제는 그대로 두고  
새로운 토큰들 간의 순서 맞추기 문제를 추가하는 것이다.

아래 그림처럼 mask 토큰을 제외한 부분에서(mask 토큰 부분은 원래 단어가 무엇인지 맞추기만 한다.), K칸(여기서는 3칸) 단위(=Trigram) (질문: 그러면 중첩되게 stride는 아닌가?) 마다 단어를 섞어 원래는 어느 위치였을까 맞추는 것이다.

그림과 같이 (t3, t4, t2) -> (t2, t3, t4) 로 예측해야 한다. 따라서 기존 objective에 이 기능을 추가한다. (동시에 학습, 동일한 가중치로, jointly learned)



**Sentence Structural Objective**

기존의 NSP를 변형한다. 기존에는 IsNext, NotNext로 segments를 문서 순서대로 뽑기 or 임의의 문서에서 뽑기로 했지만 여기서는 문서 순서대로 뽑기 or 임의의 문서에서 뽑기 or 이전 순서에서 뽑기로 해서 IsNext, IsPrevious, Nothing 이렇게 3개의 클래스로 예측하게 된다.

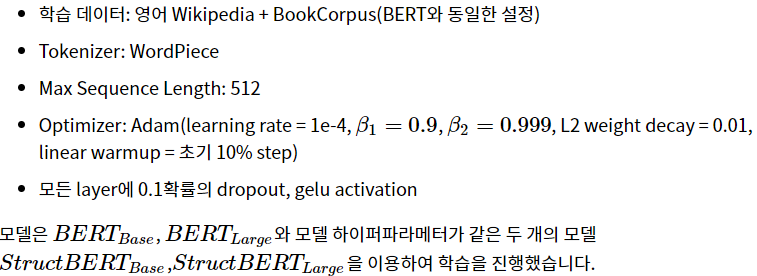
Segments 데이터셋 샘플링은 동일하게 1/3씩 한다. 또한 [CLS] 토큰 representation을 이용하여 classifier를 학습한다.

**Pre-training Setup**

Masking prob은 역시 15%

Word-ordering은 전체 중 5%의 trigram에 대해 셔플링 적용

그 외는 아래에



Experiment

3개의 Tasks, Batch: [16,24,32]

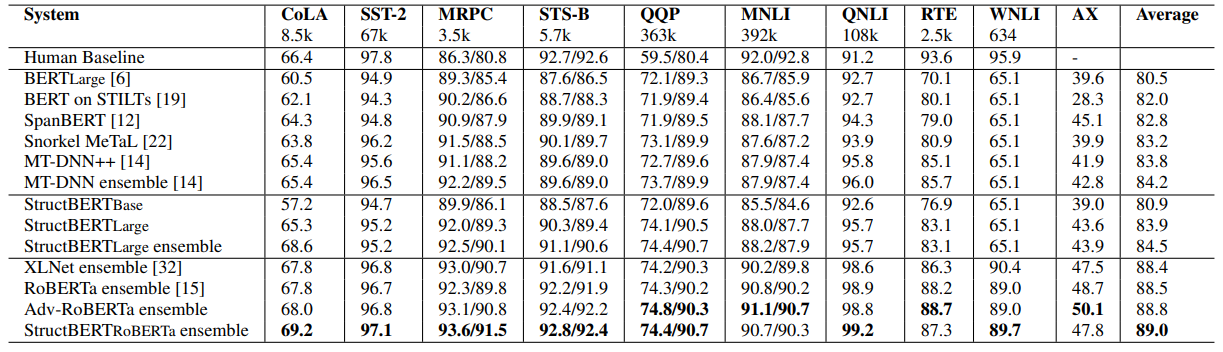
**GLUE(General Language Understanding Evaluation)**

MRPC/RTE/STS-B/MNLI – pair 단위로 진행하는 task들, MNLI가 데이터양이 많아 이걸 먼저 fine-tuning 후 성능 평가한 뒤 웨이트를 그대로 이용해 나머지 3개의 tasks에 대해서 다시 fine-tuning

나머지 tasks는 그 데이터셋만 이용

XLNet, RoBERTa 등 최신 연구는 데이터셋을 늘림 + 하이퍼 파라미터 조절 – 동등한 평가 불가  
따라서 StructBERTRoBERTa를 따로 만들었다.

추가적인 데이터로 StructBERT를 해도 성능이 더 뛰어남

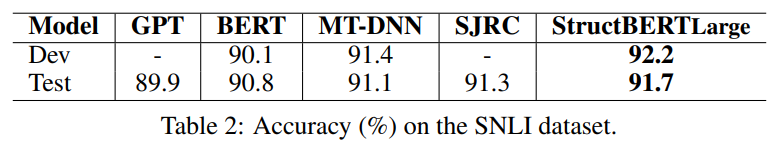


**SNLI(Stanford Natural Language Inference)**

두 문장간의 의미적 관계를 파악(MRPC, RTE, STS-B, MNLI와 유사)

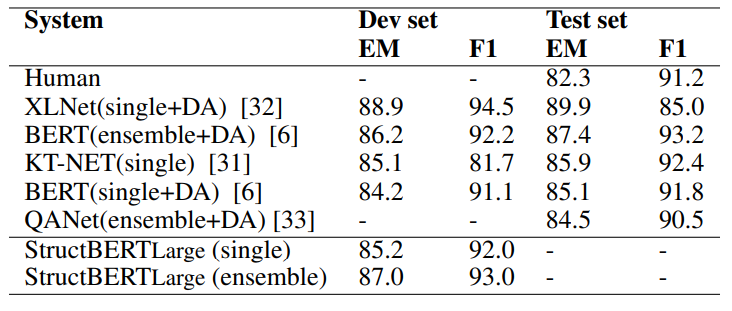
본문과 가설이 있을 때 가설이 본문에 entailment(포함,함축), contradiction(모순), neutral(그럭저럭, 중립) 인지 분류하는 문제

=> 문장간의 관계에 더 집중하는 sentence ordering objective => 향상

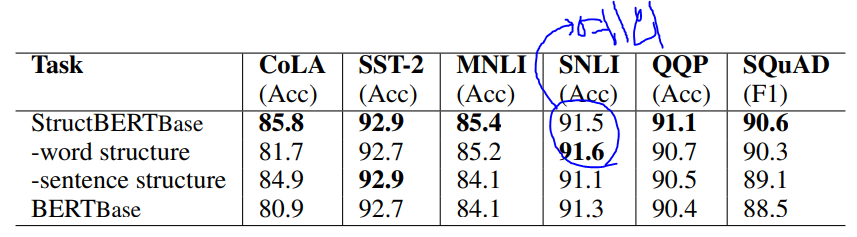


**Extractive Question Answering(SQuAD v1.1)**

데이터셋을 늘린 XLNet 보다는 약했지만 나머지보다는 우수



**Ablation Study**

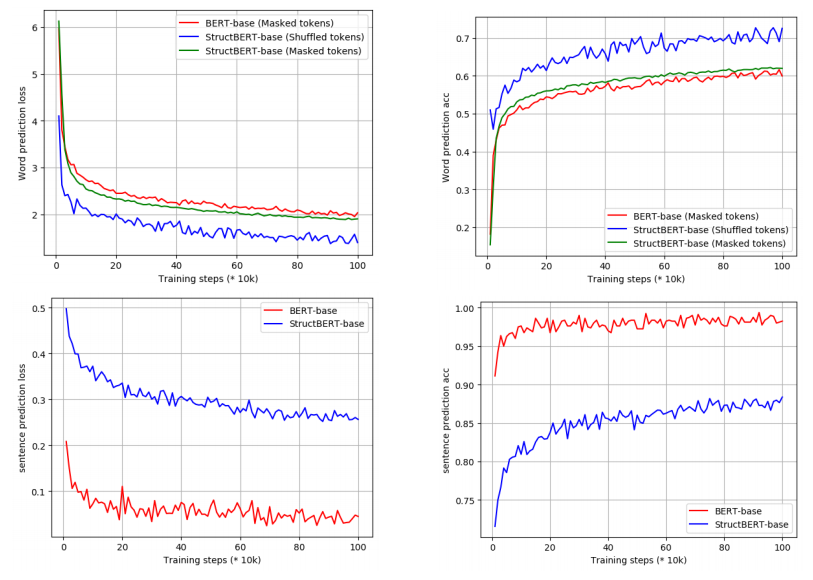


위 그림은 WS랑 SS를 각 뺐을 때이다.

대부분 pair 단위 task인 MNLI, SNLI, QQP, SQuAD에서 SS가 효과가 있었다.

그리고 single sentence task인 CoLA(문법 오류 교정), SST에서 WSO가 효과가 있었다. 특히 CoLA에서는 더 크게!

왜 근데 나머지 task는 보여주지 않는가.. 약했나?



위 두 그림: Masked-LM의 intrinsic task 성능인데 structbert가 더 강하다.(WSO 때문에), 추가적으로 StructBert만의 intrinsic task는 Word Ordering task로 여기서도 70% 가량의 어려운 문제임에도 잘 보인다.

아래 두 그림: BERT와 StructBERT의 intrinsic task인 NSP, SOP 에서의 성능이다. BERT가 intrinsic task에서는 높지만 downstream에서 실제로 쓸 때는 SOP 가 더 강하다.