AI Lab Seminar

RoBERTa: A Robustly Optimized BERT Pretraining Approach

-석사 4기 조충현-

Index

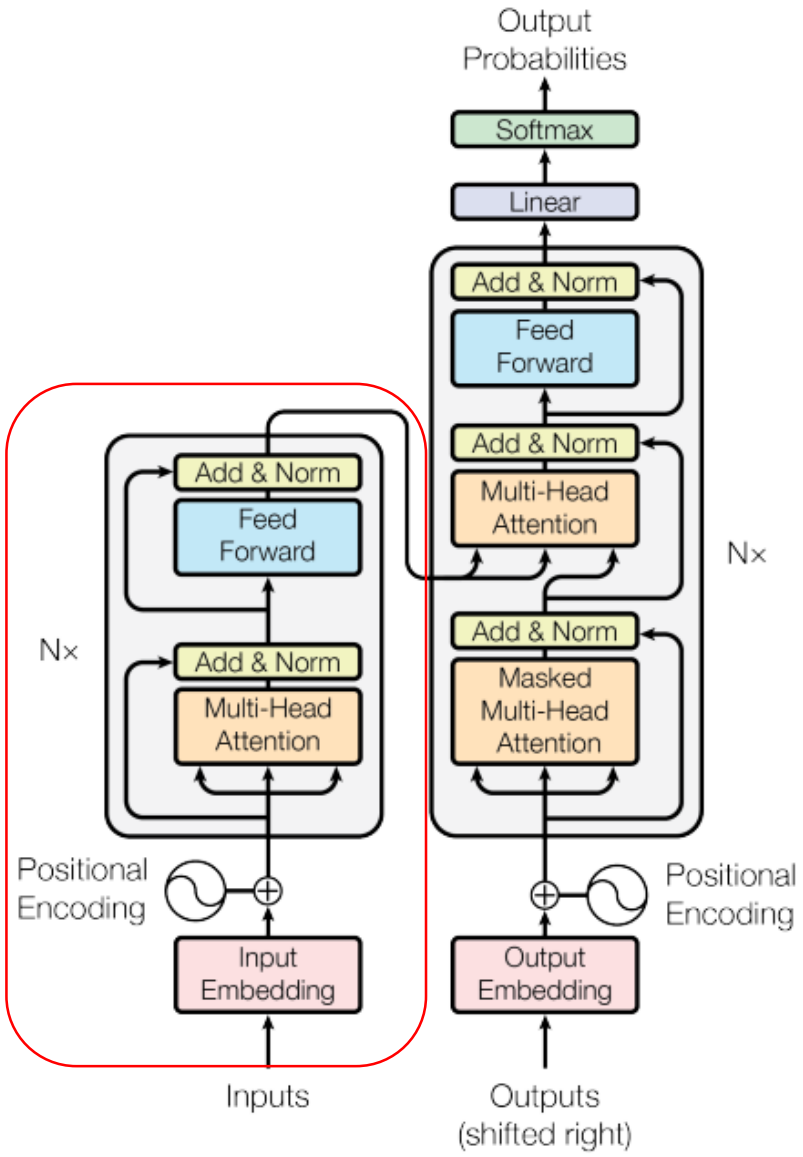
Introduction

Self-training Methods

- 최근 ELMo(Peters et al., 2018), GPT(Radford et al., 2018), BERT(Devlin et al., 2019), XLM(Lample and Conneau, 2019), XLNet(Yang et al., 2019)와 같은 모델들에서 중요한 성능 향상을 보임
- 현재 대부분의 자연어 처리 task에 이러한 모델들이 SOTA(state-of-the-art)를 찍고 있음
- 이러한 모델들의 특징은 파라미터 계수가 엄청 많기 때문에 (예. BERT 1.1억개) 학습하는데 계산량이 많이 듦
- 따라서 이러한 방법론들중 어떠한 측면에서 가장 큰 contribution이 있는지 찾기가 어렵다!
- 이 논문의 저자는 분석을 통하여 BERT가 학습이 덜 되어있어서 개선된 학습방식을 제안
- 그것이 RoBERTa -> BERT로 이룬 성능을 능가하는 성능을 보임

Background

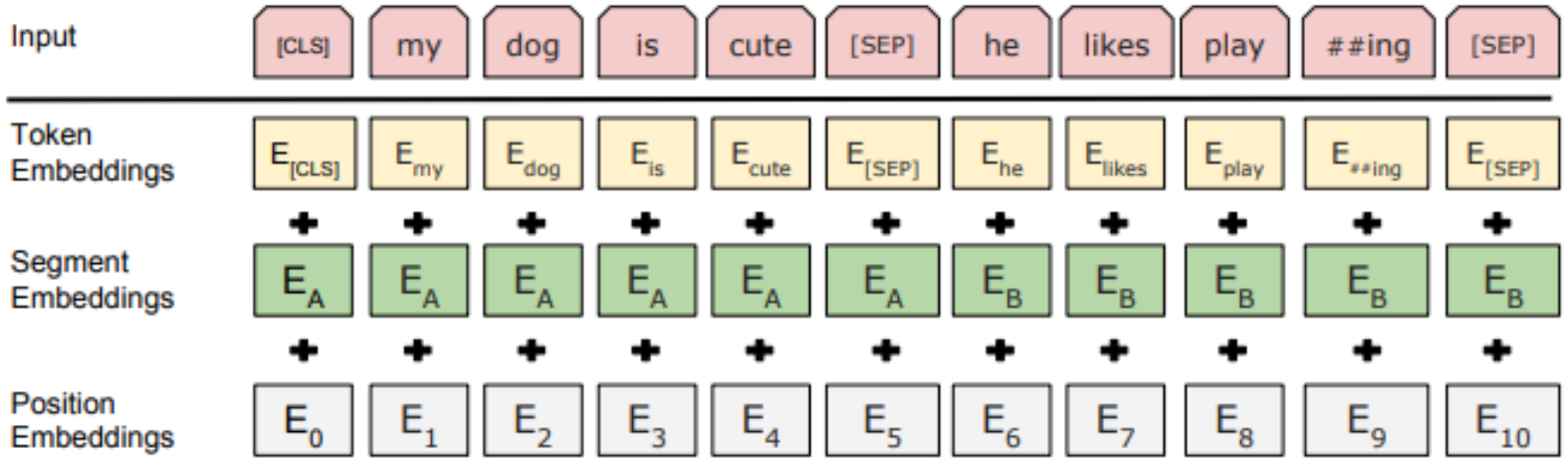
Background



BERT

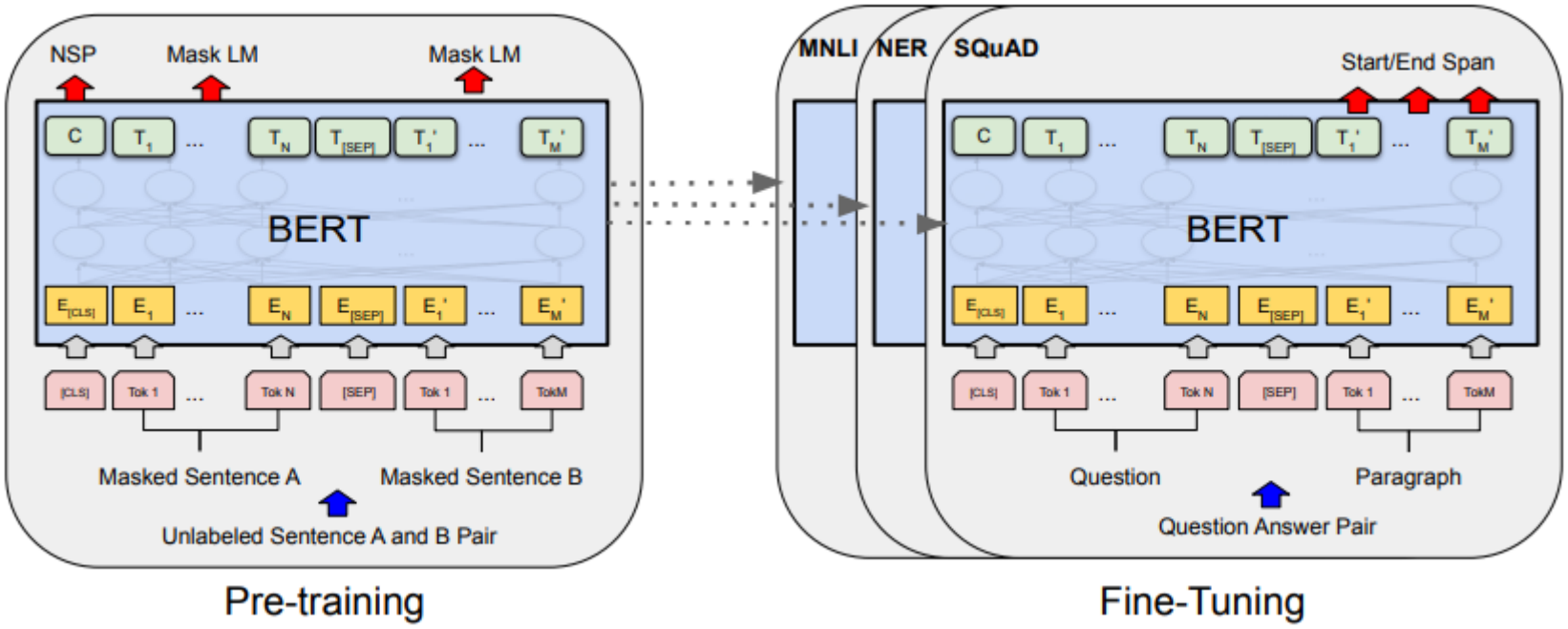
- BERT는 Transformer에서 encoder부분만 사용하여 만든 모델
- 현재 대부분의 자연어 처리 분야의 SOTA가 BERT로 부터 파생된 모델로 이루어짐
- L개의 Transformer layer, A개의 self-attention head, H의 hidden dimension으로 이루어짐

BERT : Input representation



- [CLS], [SEP] sepecial token까지 포함하여 input embedding 만듦
- 기존 Input embedding에 segment embedding, position embedding을 더하여 Input representation 생성

BERT : Training Objectives



- 총 두가지의 방식으로 Pre-training을 진행
  - Masked Language Model (MLM)
  - Next Sentence Prediction (NSP)

BERT : Training Objectives

- MLM (Masked Language Model)
  - 입력 문장에서 랜덤하게 선택한 토큰(단어)들을 선택하고 special token ('[MASK]')으로 바꿈
  - 이렇게 Masking 된 토큰의 실제 토큰을 예측하도록 학습
  - 일반적으로 문장내의 토큰 중 15%정도를 possible replacement token으로 선택
  - 선택된 토큰들 중 80%는 '[MASK]' 토큰으로 바꾸고, 10%는 바꾸지 않고 그대로 두고, 나머지 10%는 vocabulary에 있는 토큰 중 랜덤하게 선택하여 바꾼다
- NSP (Next Sentence Prediction)
  - Binary classification(0 아니면 1)으로 두개의 문장이 이어진 문장인지 아닌지 판별하도록 학습
  - NSP objective는 Natural Language Inference나 두 쌍의 문장간의 관계를 추론하는 테스크의 성능 향상을 위해 쓰임

기존 BERT와 다른점

- 더 많은 데이터에 대해 더 큰 배치사이즈로 더 오래 훈련시킴
- NSP(next sentence prediction을 삭제)
- 더 긴 문장을 넣어줌
- Masking pattern을 dynamic하게 바꿈

Experimental Setup

Experimental Setup

- Implementation
  - 기존의 BERT의 셋팅을 따랐다
  - Max token 수를 512로 고정
  - BERT와는 다르게 랜덤하게 짧은 문장을 넣지 않음
- Dataset
  - Data의 수가 많으면 많을 수록 좋은 성능의 결과가 나오기 때문에 160GB의 corpus를 사용
    - BookCorpus + English Wikipedia : BERT에서 기본으로 사용한 데이터 (16GB)
    - CC-News : CommonCrawl News dataset으로 63M개의 영어 뉴스기사를 가짐 (76GB)
    - OpenWebText : Reddit에서 적어도 3개이상의 upvote를 가진 웹 text (38GB)
    - Stories : 이야기 스타일의 CommonCrawl data의 subset 데이터 (31GB)
- Evaluation
  - GLUE (General Language Understanding Evaluation) : 9개의 dataset의 collection으로 이루어짐
  - SQuAD (Stanford Question Answering Dataset) : QA dataset
  - RACE (ReAding Comprehension from Examinations) : 매우 큰 QA dataset

Training Procedure Analysis

Training Procedure Analysis

- Static vs. Dynamic Masking
  - 오리지널 BERT는 전처리 단계에서 masking을 하기 때문에 static하게 mask가 적용된다
  - 따라서 학습동안 같은 곳에 마스킹된 문장만 보게 된다
  - 따라서 본 논문에서는 dynamic masking을 사용

Masking	SQuAD 2.0	MNLI-m	SST-2
reference	76.3	84.3	92.8
Our reimplementation:			
static	78.3	84.3	92.5
dynamic	78.7	84.0	92.9

SQuAD는 F1, MNLI-m와 SST-2는 Accuracy

Training Procedure Analysis

- Model Input Format and Next Sentence Prediction
  - NSP는 오리지널 BERT 학습에서 중요한 요소였다
  - 하지만 최근 연구에서는 NSP의 필요성에 대해 의문을 가짐
  - 따라서 4가지의 방법으로 테스트를 진행

Model	SQuAD 1.1/2.0	MNLI-m	SST-2	RACE
Our reimplementation (with NSP loss):				
SEGMENT-PAIR	90.4/78.7	84.0	92.9	64.2
SENTENCE-PAIR	88.7/76.2	82.9	92.1	63.0
Our reimplementation (without NSP loss):				
FULL-SENTENCES	90.4/79.1	84.7	92.5	64.8
DOC-SENTENCES	90.6/79.7	84.7	92.7	65.6
$\text{BERT}_{\text{BASE}}$	88.5/76.3	84.3	92.8	64.3
$\text{XLNet}_{\text{BASE}}$ (K = 7)	–/81.3	85.8	92.7	66.1
$\text{XLNet}_{\text{BASE}}$ (K = 6)	–/81.0	85.6	93.4	66.7

SQuAD는 F1, MNLI-m와 SST-2d와 RACE는 Accuracy

- Segment-Pair + NSP : 오리지널 BERT에서 사용하는 기법
- Sentence-Pair + NSP : 하나의 Segment에 한 개의 문장만 들어감
- Full-Sentences : 하나 이상의 document에 있는 연속된 문장이 들어감, 총 토큰 개수가 최대 길이를 최대한 채우도록 구성, 하나의 문서가 끝나면 다음 문서를 그대로 연결, NSP 제거
- Doc-Sentences : Full-sentences 와 비슷하지만 하나의 문서가 끝나면 다음 문서를 사용하지 않음, NSP 제거

Training Procedure Analysis

- Training with large batches
  - 최근 NMT 연구에서 learning rate를 적절하게 조절한다면 아주 큰 mini-batch 사이즈를 사용하는 것이 최적화 속도와 성능을 높일 수 있다는 것을 확인되었음
  - BERT도 똑같이 그러한 성향을 가짐
  - 따라서 동일한 계산 비용을 유지하면서 batch size를 늘려가면서 실험을 진행

bsz	steps	lr	ppl	MNLI-m	SST-2
256	1M	1e-4	3.99	84.7	92.7
2K	125K	7e-4	**3.68**	**85.2**	**92.9**
8K	31K	1e-3	3.77	84.6	92.8

Training Procedure Analysis

- Text Encoding
  - Byte-pair Encoding(BPE)은 Word 단위와 Character 단위의 사이에 있는 hybrid한 text encoding 방식
  - BPE는 입력 토큰을 sub-word unit들로 나누는데 그렇게 함으로써 OOV(out of vocabulary)를 피할 수 있음
  - BERT에서는 학습 corpus에 휴리스틱한 토크나이징을 한 뒤 character-level의 BPE를 학습
  - 본 논문에서는 GPT에서 소개한 byte 단위의 BPE를 학습

RoBERTa & Results

RoBERTa & Results

- RoBERTa
  - 이전 section에서 제안한 BERT pretraining procedure를 **RoBERTa** (Robustly optimized BERT approach) 라고 말하겠다
  - Dynamic Masking 사용
  - Full-Sentence의 입력 구성 및 NSP 제거
  - 더 큰 Batch size
  - Byte-level BPE

  - 추가적으로 pretraining에서의 data의 개수와 training step의 대하여서도 실험 진행

RoBERTa & Results

- RoBERTa

Model	data	bsz	steps	SQuAD (v1.1/2.0)	MNLI-m	SST-2
RoBERTa						
with BOOKS + WIKI	16GB	8K	100K	93.6/87.3	89.0	95.3
+ additional data (§3.2)	160GB	8K	100K	94.0/87.7	89.3	95.6
+ pretrain longer	160GB	8K	300K	94.4/88.7	90.0	96.1
+ pretrain even longer	160GB	8K	500K	**94.6/89.4**	**90.2**	**96.4**
$\text{BERT}_{\text{LARGE}}$						
with BOOKS + WIKI	13GB	256	1M	90.9/81.8	86.6	93.7
$\text{XLNet}_{\text{LARGE}}$						
with BOOKS + WIKI	13GB	256	1M	94.0/87.8	88.4	94.4
+ additional data	126GB	2K	500K	94.5/88.8	89.8	95.6

- 더 많은 데이터로 더 많이 학습할수록 좋은 결과를 보임
- 더 많은 데이터로 실험하였을 때 pretraining step을 증가해도 overfitting 되는 모습은 나타나지 않음
- 따라서 pre-training에서 데이터의 양과 다양성의 중요성을 증명

RoBERTa & Results

- RoBERTa (GLUE)

	MNLI	QNLI	QQP	RTE	SST	MRPC	CoLA	STS	WNLI	Avg
Single-task single models on dev										
$\text{BERT}_{\text{LARGE}}$	86.6/-	92.3	91.3	70.4	93.2	88.0	60.6	90.0	-	-
$\text{XLNet}_{\text{LARGE}}$	89.8/-	93.9	91.8	83.8	95.6	89.2	63.6	91.8	-	-
RoBERTa	**90.2/90.2**	**94.7**	**92.2**	**86.6**	**96.4**	**90.9**	**68.0**	**92.4**	**91.3**	-
Ensembles on test (from leaderboard as of July 25, 2019)										
ALICE	88.2/87.9	95.7	**90.7**	83.5	95.2	92.6	**68.6**	91.1	80.8	86.3
MT-DNN	87.9/87.4	96.0	89.9	86.3	96.5	92.7	68.4	91.1	89.0	87.6
XLNet	90.2/89.8	98.6	90.3	86.3	**96.8**	**93.0**	67.8	91.6	**90.4**	88.4
RoBERTa	**90.8/90.2**	**98.9**	90.2	**88.2**	96.7	92.3	67.8	**92.2**	89.0	**88.5**

Table 5: Results on GLUE. All results are based on a 24-layer architecture. $\text{BERT}_{\text{LARGE}}$ and $\text{XLNet}_{\text{LARGE}}$ results are from Devlin et al. (2019) and Yang et al. (2019), respectively. RoBERTa results on the development set are a median over five runs. RoBERTa results on the test set are ensembles of *single-task* models. For RTE, STS and MRPC we finetune starting from the MNLI model instead of the baseline pretrained model. Averages are obtained from the GLUE leaderboard.

RoBERTa & Results

- RoBERTa (SQuAD & RACE)

Model	SQuAD 1.1 EM	SQuAD 1.1 F1	SQuAD 2.0 EM	SQuAD 2.0 F1
Single models on dev, w/o data augmentation				
$\text{BERT}_{\text{LARGE}}$	84.1	90.9	79.0	81.8
$\text{XLNet}_{\text{LARGE}}$	**89.0**	94.5	86.1	88.8
RoBERTa	88.9	**94.6**	**86.5**	**89.4**
Single models on test (as of July 25, 2019)				
$\text{XLNet}_{\text{LARGE}}$			86.3†	89.1†
RoBERTa			86.8	89.8
XLNet + SG-Net Verifier			**87.0**†	**89.9**†

Table 6: Results on SQuAD. † indicates results that depend on additional external training data. RoBERTa uses only the provided SQuAD data in both dev and test settings. $\text{BERT}_{\text{LARGE}}$ and $\text{XLNet}_{\text{LARGE}}$ results are from Devlin et al. (2019) and Yang et al. (2019), respectively.

Model	Accuracy	Middle	High
Single models on test (as of July 25, 2019)			
$\text{BERT}_{\text{LARGE}}$	72.0	76.6	70.1
$\text{XLNet}_{\text{LARGE}}$	81.7	85.4	80.2
RoBERTa	**83.2**	**86.5**	**81.3**

Table 7: Results on the RACE test set. $\text{BERT}_{\text{LARGE}}$ and $\text{XLNet}_{\text{LARGE}}$ results are from Yang et al. (2019).

Conclusion

Conclusion

- RoBERTa는 BERT가 아직 과소 적합되어 있다는 것을 보여줌
- 더 좋은 성능을 보이는 training 전략을 제시함
- Pre-training을 위해 더 많은 데이터를 이용하는 것이 성능 향상에 도움을 준다는 것을 확인
- Masked language model이 pre-training에서 적합한 design choice였다는 것을 입증