

RoBERTa: A Robustly Optimized BERT Pretraining Approach

Paper 소개

- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- <https://arxiv.org/abs/1907.11692>
- BERT 모델을 더 잘 학습하는 방법 제시

BERT 모델 복습

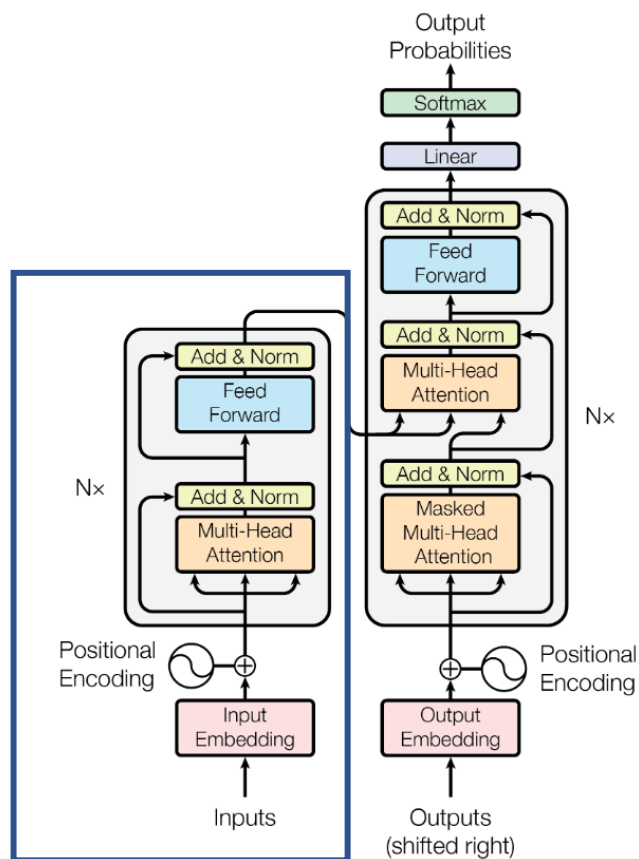
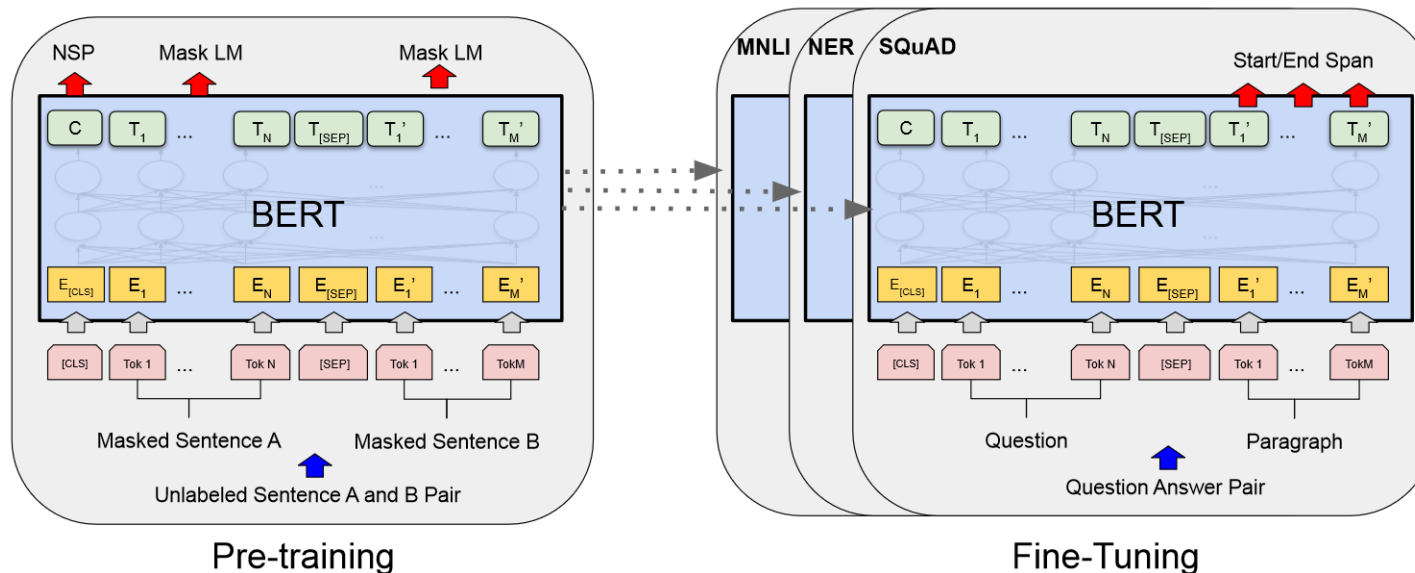
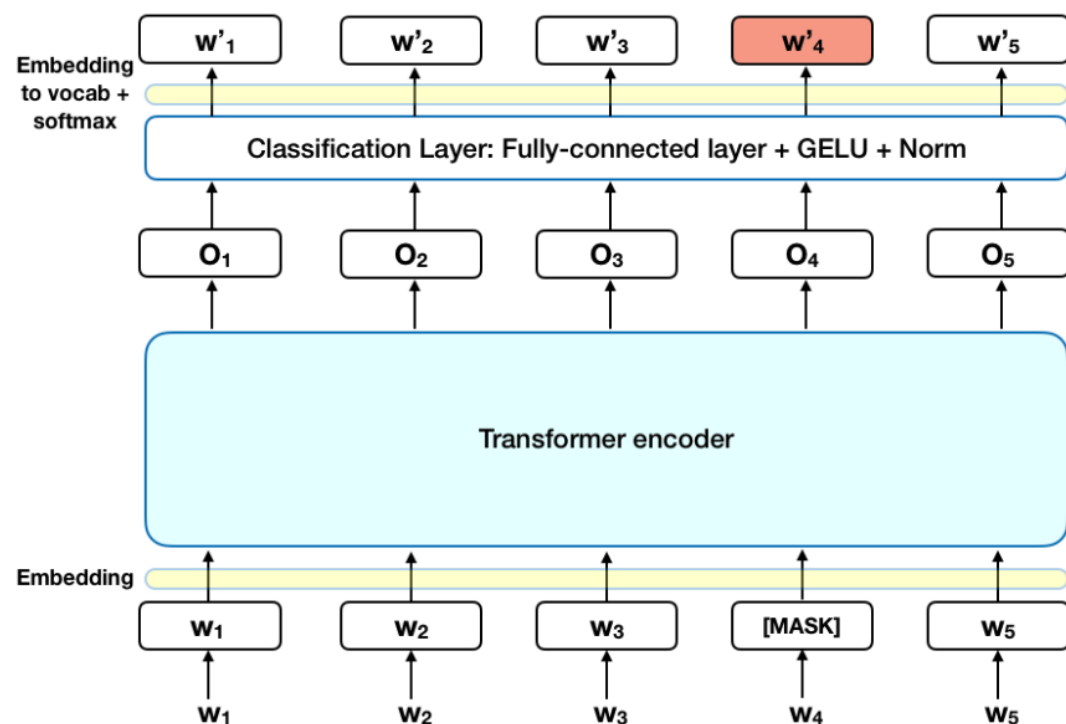


Figure 1: The Transformer - model architecture.



- Transformer의 encoder 구조를 그대로 사용하여 language representation 학습
- 학습 방법: Masked Language Model, Next Sentence Prediction
- 문제점: 모델 학습을 시키면 underfit된다.

Pre-train #1: Masked Language Model



- BERT로 Masked LM을 학습할 때 15%의 Token을 선택하여 아래와 같은 처리를 한다.
- 80%는 [MASK] token으로 바꾼다.
 - 예: my dog is hairy → my dog is [MASK]
- 10%는 전혀 임의의 단어로 바꾼다.
 - my dog is hairy → my dog is apple
- 10% 정도는 변화 없이 둔다.
 - My dog is hairy → my dog is hairy

<https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270>

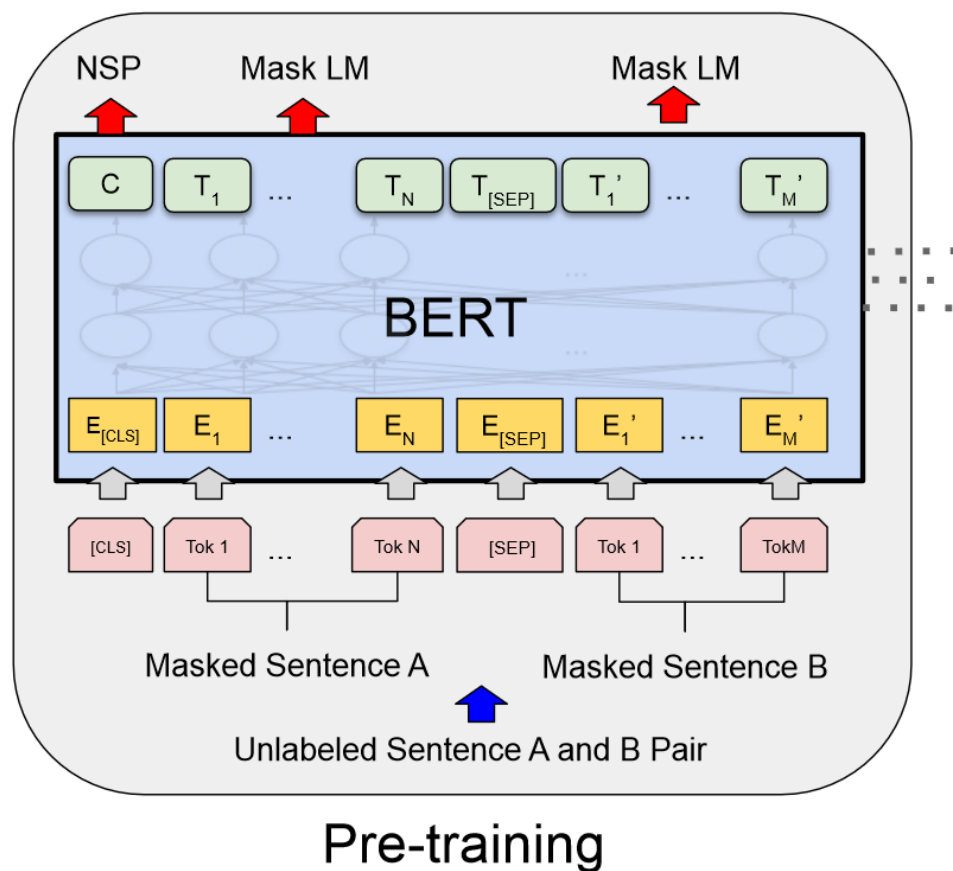
Static vs. Dynamic Masking

- Static masking
 - Masking은 train 시작할 때 한 번만 한다.
 - 이렇게 처리된 sentence는 학습 내내 똑같이 사용한다
- Dynamic masking
 - 한 문장에 대해 masking을 10가지 다른 방법으로 수행한다.
 - 40epoch 만큼 학습하므로, 같은 masking은 4번만 학습에 사용한다.
- Static은 reference(BERT)와 성능이 비슷하다.
- Dynamic은 reference보다 살짝 더 좋다.

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

Table 1: Comparison between static and dynamic masking for BERT_{BASE}. We report F1 for SQuAD and accuracy for MNLI-m and SST-2. Reported results are medians over 5 random initializations (seeds). Reference results are from [Yang et al. \(2019\)](#).

Pre-train #2: Next Sentence prediction



- 두 문장이 문맥상 이어지는 문장인지 아닌지 예측하는 Task
- IsNext: dataset에 있는 sentence를 순서대로 뽑은 case
- NotNext: 두 sentence를 랜덤하게 뽑은 case
- BERT에서 'sentence'는 자연어에서 말하는 여러 문장을 포함할 수 있다.
- Pre-train Task 1과 마찬가지로, 일부 token을 [MASK]로 대체한다.
- NSP를 사용하지 않으면 성능이 떨어진다는 연구도 있었다.
- 그러나 2019년에, NSP loss가 정말 필요한 것인지 의문을 표하는 논문이 발표되었다.

Model Input Format and NSP

- SEGMENT-PAIR+NSP
 - 원래 BERT 논문에서 사용한 것
 - Segment 하나는 여러 문장을 포함할 수 있음
- SENTENCE-PAIR+NSP
 - 두 자연어 문장(natural sentences)의 쌍을 pair로 사용한 것
- FULL-SENTENCES
 - 하나 이상의 document에서 sentence sampling. NSP loss 미사용
- DOC-SENTENCES
 - 하나의 document에서만 sentence sampling. NSP loss 미사용

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

Table 2: Development set results for base models pretrained over BOOKCORPUS and WIKIPEDIA. All models are trained for 1M steps with a batch size of 256 sequences. We report F1 for SQuAD and accuracy for MNLI-m, SST-2 and RACE. Reported results are medians over five random initializations (seeds). Results for BERT_{BASE} and XLNet_{BASE} are from [Yang et al. \(2019\)](#).

Model Input Format and NSP

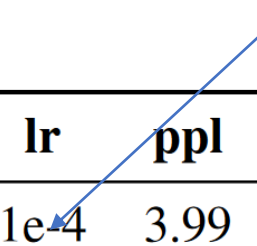
- BERT에서 사용하는 sentence 대신 자연 언어의 sentence 단위를 사용하면, downstream task에서의 성능이 하락
- NSP Loss를 사용하지 않으면 성능이 조금 향상된다.
- 이 때, sequence를 만들 때 sentence는 하나의 document에서만 고르는 것이 좋다.

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

Table 2: Development set results for base models pretrained over BOOKCORPUS and WIKIPEDIA. All models are trained for 1M steps with a batch size of 256 sequences. We report F1 for SQuAD and accuracy for MNLI-m, SST-2 and RACE. Reported results are medians over five random initializations (seeds). Results for BERT_{BASE} and XLNet_{BASE} are from [Yang et al. \(2019\)](#).

Training with large batches

- 2018년에, 기계번역 연구에서 batch size를 매우 크게 키우면 성능이 향상된다는 보고가 있다
- BERT 또한, batch size를 원래 논문의 32에서 훨씬 더 키울 수 있다는 논문이 있다.
- Batch size를 키워서 학습해 보니,
 - Perplexity가 좋아짐
 - 따라서 end-task accuracy 또한 좋아짐
- BERT와 메모리 사용량이 똑같아서 GPU에선 batch size를 무조건 늘리기는 어렵다.



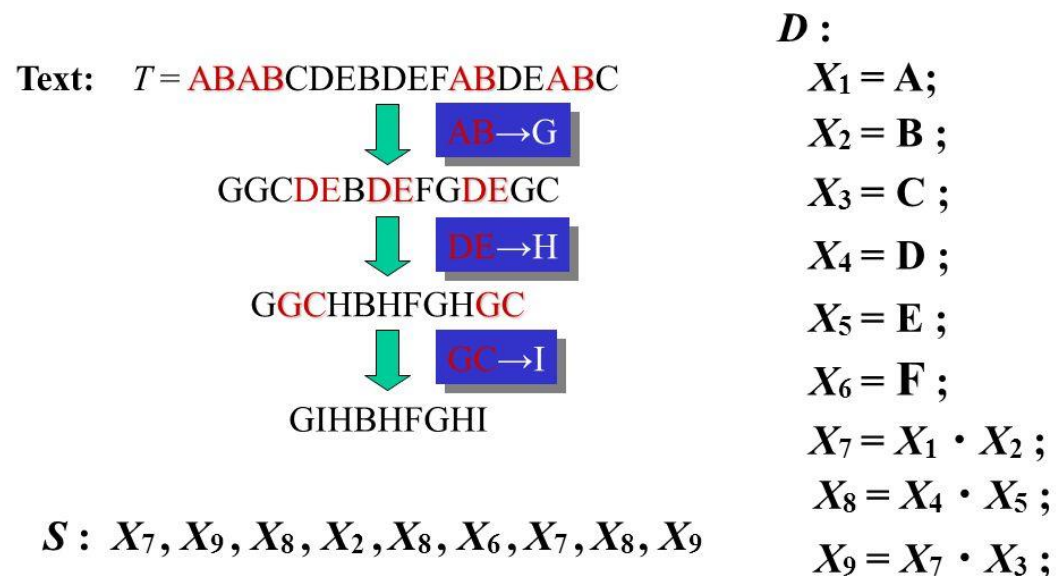
bsz	steps	lr	ppl	MNLI-m	SST-2
256	1M	1e-4	3.99	84.7	92.7
2K	125K	7e-4	3.68	85.2	92.9
8K	31K	1e-3	3.77	84.6	92.8

Table 3: Perplexity on held-out training data (*ppl*) and development set accuracy for base models trained over BOOKCORPUS and WIKIPEDIA with varying batch sizes (*bsz*). We tune the learning rate (*lr*) for each setting. Models make the same number of passes over the data (epochs) and have the same computational cost.

Text Encoding

- BPE(Byte Pair Encoding)
 - <https://www.aclweb.org/anthology/P16-1162.pdf>
 - 모든 글자(characters) 또는 유니코드(unicode) 단위로 단어 집합(vocabulary)를 만들고,
 - 가장 많이 등장하는 유니그램을 하나의 유니그램으로 통합
- BERT: character level BPE 사용 (vocab: 30k)
- RoBERTa: byte level BPE 사용 (vocab: 50k)
 - 50k 정도의 vocab size만으로, UNK token 없이 학습이 가능함
 - 한글은 utf-8 기준 3byte이므로, 한글 한 글자는 3개로 쪼개질 수 있음

Byte Pair Encoding “collage system”



<https://slideplayer.com/slide/8527360/>

RoBERTa

- RoBERTa
 - Robustly optimized **BERT** approach
- BERT와의 차이점
 - Dynamic masking
 - FULL-SENTENCES without NSP loss
 - Large mini-batches
 - Larger byte-level BPE
- 위와 같은 개선을 통해, BERT에 비해 end task의 성능이 올라감

Results

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
BERT _{LARGE} ensembles, t						
with BOOKS + WIKI	13GB	256	1M	90.9/81.8	86.6	93.7
XLNet _{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

Table 4: Development set results for RoBERTa as we pretrain over more data (16GB \rightarrow 160GB of text) and pretrain for longer (100K \rightarrow 300K \rightarrow 500K steps). Each row accumulates improvements from the rows above. RoBERTa matches the architecture and training objective of BERT_{LARGE}. Results for BERT_{LARGE} and XLNet_{LARGE} are from [Devlin et al. \(2019\)](#) and [Yang et al. \(2019\)](#), respectively. Complete results on all GLUE tasks can be found in the Appendix.

- BERT large보다 훨씬 좋은 성능을 보여준다
- Pre-train 학습 데이터만 추가해도 성능이 향상된다
- Pre-train step을 늘려나가면 XLNet_large보다도 성능이 좋다

Results (GLUE dataset)

	MNLI	QNLI	QQP	RTE	SST	MRPC	CoLA	STS	WNLI	Avg
<i>Single-task single models on dev</i>										
BERT _{LARGE}	86.6/-	92.3	91.3	70.4	93.2	88.0	60.6	90.0	-	-
XLNet _{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	-
<i>Ensembles on test (from leaderboard as of July 25, 2019)</i>										
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. BERT_{LARGE} and XLNet_{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.

- Single-task single models: 각 TASK마다 각각 fine-tuning
- Ensembles on test: multi-task fine-tuning으로 학습한 모델
- Single-task single-model 일때는 RoBERTa가 XLNet보다도 성능이 좋음을 확인할 수 있다.

Result (SQuAD, RACE dataset)

Model	SQuAD 1.1		SQuAD 2.0	
	EM	F1	EM	F1
<i>Single models on dev, w/o data augmentation</i>				
BERT _{LARGE}	84.1	90.9	79.0	81.8
XLNet _{LARGE}	89.0	94.5	86.1	88.8
RoBERTa	88.9	94.6	86.5	89.4
<i>Single models on test (as of July 25, 2019)</i>				
XLNet _{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. BERT_{LARGE} and XLNet_{LARGE} results are from [Devlin et al. \(2019\)](#) and [Yang et al. \(2019\)](#), respectively.

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

Table 7: Results on the RACE test set. BERT_{LARGE} and XLNet_{LARGE} results are from [Yang et al. \(2019\)](#).

- 질문-정답 데이터셋을 이용한 모델 성능 비교
- SQuAD 1.1은 XLNet와 동등한 수준의 성능, 2.0은 XLNet보다 좋은 성능을 보여준다.
- RACE Dataset 기준으로 RoBERTa가 더 좋은 성능을 보여준다.

XLNet 논문 Update

- 2020. 01. 02. 에 XLNet 논문이 업데이트되었다.
- RoBERTa 논문에서, multi-task fine tunin을 하지 않는 이상 XLNet보다 RoBERTa가 더 좋다는 실험 결과를 내었다.
- XLNet 발표 이후 RoBERTa와 ALBERT가 나왔는데, 이 중 연산량이 동일한 RoBERTa와 비교 수행
- XLNet 저자들이 다시 실험한 결과 RoBERTa보다 XLNet의 성능이 더 좋음을 확인
- RoBERTa 논문과 XLNet 논문에서 사용한 데이터셋은 다르다
 - RoBERTa: BookCorpus + Wikipedia
 - XLNet: BookCorpus + Wikipedia + Giga5 + ClueWeb + Common Crawl

Results (from XLNet paper)

Model	MNLI	QNLI	QQP	RTE	SST-2	MRPC	CoLA	STS-B	WNLI
<i>Single-task single models on dev</i>									
BERT [2]	86.6/-	92.3	91.3	70.4	93.2	88.0	60.6	90.0	-
RoBERTa [21]	90.2/90.2	94.7	92.2	86.6	96.4	90.9	68.0	92.4	-
XLNet	90.8/90.8	94.9	92.3	85.9	97.0	90.8	69.0	92.5	-
<i>Multi-task ensembles on test (from leaderboard as of Oct 28, 2019)</i>									
MT-DNN* [20]	87.9/87.4	96.0	89.9	86.3	96.5	92.7	68.4	91.1	89.0
RoBERTa* [21]	90.8/90.2	98.9	90.2	88.2	96.7	92.3	67.8	92.2	89.0
XLNet*	90.9/90.9[†]	99.0[†]	90.4[†]	88.5	97.1[†]	92.9	70.2	93.0	92.5

Table 5: Results on GLUE. * indicates using ensembles, and † denotes single-task results in a multi-task row. All dev results are the median of 10 runs. The upper section shows direct comparison on dev data and the lower section shows comparison with state-of-the-art results on the public leaderboard.

- Single-task single models: 각 TASK마다 각각 fine-tuning
- Ensembles on test: multi-task fine-tuning으로 학습한 모델
- 거의 모든 task에 대해 XLNet이 BERT나 RoBERTa보다 성능이 좋다.

Results (from XLNet paper)

RACE	Accuracy	Middle	High	Model	NDCG@20	ERR@20
GPT [28]	59.0	62.9	57.4	DRMM [13]	24.3	13.8
BERT [25]	72.0	76.6	70.1	KNRM [8]	26.9	14.9
BERT+DCMN* [38]	74.1	79.5	71.8	Conv [8]	28.7	18.1
RoBERTa [21]	83.2	86.5	81.8	BERT [†]	30.53	18.67
XLNet	85.4	88.6	84.0	XLNet	31.10	20.28

Table 2: Comparison with state-of-the-art results on the test set of RACE, a reading comprehension task, and on ClueWeb09-B, a document ranking task. * indicates using ensembles. † indicates our implementations. “Middle” and “High” in RACE are two subsets representing middle and high school difficulty levels. All BERT, RoBERTa, and XLNet results are obtained with a 24-layer architecture with similar model sizes (aka BERT-Large).

- 질문-정답 데이터셋을 이용한 모델 성능 비교
- XLNet이 RoBERTa보다 성능이 좋다.

Results (from XLNet paper)

SQuAD2.0	EM	F1	SQuAD1.1	EM	F1
<i>Dev set results (single model)</i>					
BERT [10]	78.98	81.77	BERT† [10]	84.1	90.9
RoBERTa [21]	86.5	89.4	RoBERTa [21]	88.9	94.6
XLNet	87.9	90.6	XLNet	89.7	95.1
<i>Test set results on leaderboard (single model, as of Dec 14, 2019)</i>					
BERT [10]	80.005	83.061	BERT [10]	85.083	91.835
RoBERTa [21]	86.820	89.795	BERT* [10]	87.433	93.294
XLNet	87.926	90.689	XLNet	89.898‡	95.080‡

Table 3: Results on SQuAD, a reading comprehension dataset. † marks our runs with the official code. * indicates ensembles. ‡: We are not able to obtain the test results of our latest model on SQuAD1.1 from the organizers after submitting our result for more than one month, and thus report the results of an older version for the SQuAD1.1 test set.

- 질문-정답 데이터셋을 이용한 모델 성능 비교
- SQuAD 1.1, 2.0 모두 XLNet의 성능이 RoBERTa보다 좋다.