Using the Transformer

RoBERTa (Liu et al., 2019)

Learning goals

- Understand the improvements over BERT
- Dynamic Masking

IMPROVEMENTS IN PRE-TRAINING

Short summary:

- Change of the `MASK`ing strategy
  $\rightarrow$ BERT masks the sequences once before pre-training
  $\rightarrow$ RoBERTa uses dynamic `MASK`ing
  $\Rightarrow$ RoBERTa sees the same sequence `MASK`ed differently
- RoBERTa does not use the additional NSP objective during pre-training
- Authors claim that BERT is seriously "undertrained"
  - 160 GB of pre-training resources instead of 13 GB
  - Pre-training is performed with larger batch sizes (8k)

DYNAMIC VS. STATIC MASKING

Static Masking (BERT):

- Apply `MASK`ing procedure to pre-training corpus once
- (additional for BERT: Modify the corpus for NSP)
- Train for approximately 40 epochs

Dynamic Masking (RoBERTa):

- Duplicate the training corpus *ten* times
- Apply `MASK`ing procedure to each duplicate of the pre-training corpus
- Train for 40 epochs
- Model sees each training instance in ten different "versions" (each version four times) during pre-training

DYNAMIC VS. STATIC MASKING

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

Table 1: Comparison between static and dynamic masking for $\text{BERT}_{\text{BASE}}$. 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).

Source: Liu et al. (2019)

NO NSP

- Described as important part of the pre-training process in BERT
- Liu et al. report that it hurts performance

→ Especially for QNLI, MNLI, and SQuAD 1.1

- Conduct experiments in multiple settings:
  - SEGMENT-PAIR+NSP
  - SENTENCE-PAIR+NSP
  - FULL-SENTENCES
  - DOC-SENTENCES

ROBERTA ‣ LIU ET AL., 2019

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

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 $\text{BERT}_{\text{BASE}}$ and $\text{XLNet}_{\text{BASE}}$ are from Yang et al. (2019).

Source: Liu et al. (2019)

Note: XLNet: see next Chapter.

CHANGES IN PRE-TRAINING

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.

Source: Liu et al. (2019)

CHANGES IN PRE-TRAINING

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

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 $\text{BERT}_{\text{LARGE}}$. Results for $\text{BERT}_{\text{LARGE}}$ and $\text{XLNet}_{\text{LARGE}}$ are from Devlin et al. (2019) and Yang et al. (2019), respectively. Complete results on all GLUE tasks can be found in the Appendix.

Source: Liu et al. (2019)

Note: XLNet: see next Chapter.

ROBERTA ‣ LIU ET AL., 2019

Architectural differences:

- Architecture (layers, heads, embedding size) identical to BERT
- 50k token BPE vocabulary instead of 30k
- Model size differs (due to the larger embedding matrix)
  $\Rightarrow \sim$ 125M (360M) for the BASE (LARGE) variant

Performance differences:

	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**	-

Source: Liu et al. (2019)

Note: Liu et al. (2019) report the accuracy for QQP while Devlin et al. (2018) report the F1 score (cf. results displayed in chapter 6.2.3); XLNet: see next Chapter.