A Fair Comparison Study of XLNet and BERT with Large Models

X

XLNet Team Follow

Jul 22 · 4 min read

Several weeks ago, we released our new model XLNet, which outperforms BERT on a variety of benchmarks. Our largest model was trained on about 10x more data compared to BERT. For fair comparison, we included ablation study using the base model architecture and the same training data. Since then, our friends in both academia and industry have shown interest in how XLNet-Large would perform if only trained on the same data. We ourselves were also interested in investigating the gain of using additional data as we simply threw all the data we had (with decent quality) into training in our initial release. Therefore, we believe it will be of important scientific merits to conduct a fair comparison study between XLNet and BERT using the large model architecture and the same data.

. . .

In this study, we ensure that almost every possible hyperparameter is the same for the training recipes of both BERT and XLNet. These hyperparameters were used for BERT and published by the BERT

authors. In other words, they were chosen and possibly optimized for BERT, instead of XLNet. Specifically, we carefully control the following hyperparameters:

- The same batch size: 256
- The same number of training steps: 1M
- The same optimizer: Adam, learning rate 1e-4, warmup 10K, linear decay
- The same training corpora: Wikipedia + BooksCorpus. We used the same tool to process Wikipedia as described in the BERT repo. But for some unknown reason, our Wikipedia corpus has only 2B words, compared to 2.5B words used in BERT. As a result, XLNet was trained on slightly less data.
- The same model architecture parameters: 24 layers, 1024 hidden size, 16 heads
- The same finetuning hyperparameter search space

In addition, we modified a few data-related implementation details for an apples-to-apples comparison with BERT.

- In our previous implementation, the unmasked tokens do not see CLS and SEP in pretraining. In our current implementation, the unmasked tokens do see CLS and SEP, which is consistent with BERT.
- During finetuning, following BERT, we use the “BERT format” [CLS, A, SEP, B, SEP] instead of [A, SEP, B, SEP, CLS].

Furthermore, we consider three variants of BERT and report the best finetuning result for each individual task. The three variants are as follows:

- Model-I: The original BERT released by the authors
- Model-II: BERT with whole word masking, also released by the authors
- Model-III: Since we found that next-sentence prediction (NSP) might hurt performance, we use the published code of BERT to pretrain a new model without the NSP loss

Note that this setting might give BERT some advantages because the best performances of individual tasks might be obtained by different variants.

. . .

The dev set results on GLUE and SQuAD and the test set results on RACE are as follows. No data augmentation, ensembles, or multi-tasking learning were used.

Dataset	**XLNet-Large (as in paper)**	**XLNet-Large -wikibooks**	**BERT-Large -wikibooks** best of 3 variants
SQuAD1.1 EM	89.0	88.2	86.7 (II)
SQuAD1.1 F1	94.5	94.0	92.8 (II)

SQuAD1.1 F1	94.5	94.0	92.8 (II)
SQuAD2.0 EM	86.1	85.1	82.8 (II)
SQuAD2.0 F1	88.8	87.8	85.5 (II)
RACE	81.8	77.4	75.1 (II)
MNLI	89.8	88.4	87.3 (II)
QNLI	93.9	93.9	93.0 (II)
QQP	91.8	91.8	91.4 (II)
RTE	83.8	81.2	74.0 (III)
SST-2	95.6	94.4	94.0 (II)
MRPC	89.2	90.0	88.7 (III)
CoLA	63.6	65.2	63.7 (II)
STS-B	91.8	91.1	90.2 (III)

Comparison of different models. XLNet-Large (as in paper) was trained with more data and a larger batch size. For BERT, we report the best finetuning result of 3 variants for each dataset.

. . .

There are a few interesting observations from the table:

1. Trained on the same data with an almost identical training recipe, XLNet outperforms BERT by a sizable margin on all the datasets.
2. The gains of training on 10x more data (comparing XLNet-Large-wikibooks and XLNet-Large) are smaller than the gains of switching from BERT to XLNet on 8 out of 11 benchmarks.
3. On some of the benchmarks such as CoLA and MRPC, the model trained on more data underperforms the model trained on less data.

. . .

We believe we have valuable learnings from the above results.

XLNet improves performance. Observation #1 is consistent with our early ablation on base models, suggesting the advantages of XLNet over BERT given the same training conditions.

XLNet-Large could be better optimized. Observations #2 and #3 seem to suggest that our previous released XLNet-Large (trained on more data) did not fully leverage the data scale. Hence, we will continue to investigate how to properly scale up language pretraining with XLNet. From our current (limited) observations, we conjecture that the following training details might play important roles:

- Data related: data scale, data source, data cleaning, data encoding, data formatting
- Optimization related: learning rate (& schedule), batch size, number of training steps, optimizer
- Importantly, these hyper-parameters might have high-order interactions with each other.

Facebook AI's recent entry on the GLUE leaderboard seems to also suggest the importance of training details.

. . .

In conclusion, this study has more clearly decoupled the effects of **algorithms/models** from the other factors such as **training details**, **large computation**, and **big data**. Based on the results, we think that algorithms and models are at least as important as the other factors. It is likely that they are all necessary for achieving the final goal of natural language understanding. We will also update the XLNet paper with the above new results very soon.

Deep Lea NLP