Small BERT. Which one is the best?

- Compression Rate vs. MNLI accuracy

EXTREME LANGUAGE MODEL COMPRESSION WITH OPTIMAL SUBWORDS AND SHARED PROJECTIONS

Sanqiang Zhao, Raghav Gupta, Yang Song, Denny Zhou
Google AI, USA
{sanqiang, raghavgupta, yangso, dennyzhou}@google.com

Model	Hidden Dim	Vocab Size	Compress Factor	MRPC (F1/Acc)	MNLI-m (Acc)	MNLI-mm (Acc)	SST-2 (Acc)
Teacher $BERT_{BASE}$	768	30522	1x	88.5/84.3	84.0	82.8	93.5
PKD, 6 layers (Sun et al., 2019)	768	30522	1.64x	85.0/79.9	81.5	81.0	92.0
PKD, 3 layers (Sun et al., 2019)			2.40x	80.7/72.5	76.7	76.3	87.5
NoKD Baseline	192	4928	5.74x	82.6/74.1	77.4	76.5	87.1
DualTrain				82.5/76.6	78.1	77.3	**88.4**
DualTrain + SharedProjDown				83.6/76.9	**78.2**	**77.7**	**88.4**
DualTrain + SharedProjUp				**84.9/78.5**	77.5	76.7	88.0
NoKD Baseline	96	4928	19.41x	84.6/77.3	76.2	75.1	85.4
DualTrain				**86.1/80.5**	76.1	74.7	85.4
DualTrain + SharedProjDown				83.7/77.5	**76.5**	**75.2**	**85.6**
DualTrain + SharedProjUp				84.9/78.1	76.4	**75.2**	84.7
NoKD Baseline	48	4928	61.94x	76.3/66.1	70.9	70.2	79.5
DualTrain				77.5/66.8	70.6	69.9	79.8
DualTrain + SharedProjDown				78.0/68.2	**71.3**	70.4	80.0
DualTrain + SharedProjUp				**79.3/68.6**	71.0	**70.8**	**82.2**

Table 3: Results of the distilled models, the teacher model and baselines on the downstream language understanding task test sets, obtained from the GLUE server, along with the size parameters and compression ratios of the respective models compared to the teacher $BERT_{BASE}$. MNLI-m and MNLI-mm refer to the genre-matched and genre-mismatched test sets for MNLI.

BERT-BASE:
1/6: 84 => 78
1/20: 84 => 76
1/60: 84 => 71

Severe Performance Drop

MOBILEBERT: TASK-AGNOSTIC COMPRESSION OF BERT BY PROGRESSIVE KNOWLEDGE TRANSFER

Anonymous authors
Paper under double-blind review

BERT-BASE:
1/4: 84(?) => 84

Retrain the teacher
New tearcher:
1/10: 87 => 84

Table 3: The test results on the GLUE benchmark (except WNLI). The number below each task denotes the number of training examples. The metrics for these tasks can be found in the GLUE paper (Wang et al., 2018). For tasks with multiple metrics, the metrics are arithmetically averaged to compute the GLUE score. "OPT" denotes the operational optimizations introduced in Section 4.3.

	#Params	#FLOPS	CoLA 8.5k	SST-2 67k	MRPC 3.7k	STS-B 5.7k	QQP 364k	MNLI-m/mm 393k	QNLI 108k	RTE 2.5k	GLUE
ELMo-BiLSTM-Attn	-	-	33.6	90.4	84.4	72.3	63.1	74.1/74.5	79.8	58.9	70.0
OpenAI GPT	109M	-	47.2	93.1	87.7	84.8	70.1	80.7/80.6	87.2	69.1	76.9
$\text{BERT}_{\text{BASE}}$	109M	22.5B	**52.1**	**93.5**	**88.9**	**85.8**	71.2	**84.6/83.4**	90.5	66.4	78.3
$\text{BERT}_{\text{BASE}}$-6L-PKD	66.5M	11.3B	-	92.0	85.0	-	70.7	81.5/81.0	89.0	65.5	-
$\text{BERT}_{\text{BASE}}$-3L-PKD	45.3M	5.7B	-	87.5	80.7	-	68.1	76.7/76.3	84.7	58.2	-
MobileBERT	25.3M	5.7B	50.5	92.8	88.8	84.4	70.2	83.3/82.6	90.6	66.2	77.7
MobileBERT w/o OPT	25.3M	5.7B	51.1	92.6	88.8	84.8	70.5	84.3/**83.4**	**91.6**	**70.4**	**78.5**

5.2 IMPLEMENTATION DETAILS

REWEIGHTED PROXIMAL PRUNING FOR LARGE-SCALE LANGUAGE REPRESENTATION

Fu-Ming Guo[1], Sijia Liu[2], Finlay S. Mungall[3], Xue Lin[1] &Yanzhi Wang[1]

[1]Northeastern University
[2]MIT-IBM Watson AI Lab, IBM Research
[3]United States Federal Aviation Administration
guo.fu@husky.neu.edu, sijia.liu@ibm.com, fmungall@gmail.com
{*xue.lin,yanz.wang*}*@northeastern.edu*

BERT-Large:
1/2: 86 (?) => 86
1/10: 86 (?) => 82

Retrain the model

Table 1: $\text{BERT}_{\text{LARGE}}$ pruning results on a set of transfer learning tasks. The degradation is contrasted with the original BERT (without pruning) for transfer learning.

Method	Prune Ratio(%)	SQuAD 1.1	QQP	MNLI	MRPC	CoLA
NIP	50.0	85.3 (-5.6)	85.1 (-6.1)	77.0 (-9.1)	83.5 (-5.5)	76.3 (-5.2)
	80.0	75.1 (-15.8)	81.1 (-10.1)	73.81 (-12.29)	68.4 (-20.5)	69.13 (-12.37)
RPP	59.3	90.23 (-0.67)	91.2 (-0.0)	86.1 (-0.0)	88.1 (-1.2)	82.8 (+1.3)
	88.4	81.69 (-9.21)	89.2 (-2.0)	81.4 (-4.7)	81.9 (-7.1)	79.3 (-2.2)
Method	Prune Ratio(%)	SQuAD 2.0	QNLI	MNLIM	SST-2	RTE
NIP	50.0	75.3 (-6.6)	90.2 (-1.1)	82.5 (-3.4)	91.3 (-1.9)	68.6 (-1.5)
	80.0	70.1 (-11.8)	80.5 (-10.8)	78.4 (-7.5)	88.7 (-4.5)	62.8 (-7.3)
RPP	59.3	81.3 (-0.6)	92.3 (+1.0)	85.7 (-0.2)	92.4 (-0.8)	70.1 (-0.0)
	88.4	80.7 (-1.2)	88.0 (-3.3)	81.8 (-4.1)	90.5 (-2.7)	67.5 (-2.6)

FASTER AND JUST AS ACCURATE: A SIMPLE DECOMPOSITION FOR TRANSFORMER MODELS

	Avg. Input Tokens	BERT base	Decomp-BERT base	Performance Drop (absolute | %age)	Inference Speedup (times)	Memory Reduction (%age)
SQuAD	320	88.5	87.1	1.4 | 1.6	3.2x	70.3
RACE	2048	66.3	64.5	1.8 | 2.7	3.4x	72.9
BoolQ	320	77.8	76.8	1.0 | 1.3	3.5x	72.0
MNLI	120	84.4	82.6	1.8 | 2.1	2.2x	56.4
QQP	100	90.5	90.3	0.2 | 0.2	2.0x	50.0

Table 1: (i) Performance of BERT-base vs Decomp-BERT-base, (ii) Performance drop, inference speedup and inference memory reduction of Decomp-BERT-base over BERT-base for 5 tasks. Decomp-BERT-base uses nine lower layers, and three upper layers with caching enabled. For SQuAD and RACE we also train with the auxiliary losses, and for the others we use the main supervision loss – the settings that give the best effectiveness during training. Note that the choice of the loss doesn't affect the efficiency metrics.

	Performance (Squad-F1)	Speed (GFLOPs)	Memory (MB)
BERT-large	92.3	204.1	1549.6
BERT-base	88.5	58.4	584.2
Decomp-BERT-large	90.8	47.7	359.7

Table 2: Performance, Inference Speed and Memory for different models on SQuAD.

	Tesla V100 GPU	Intel i9-7900X CPU	OnePlus 6 Phone
BERT-base	0.22	5.90	10.20*
Decomp-BERT-base	0.07	1.66	3.28*

Table 3: Inference latency (in seconds) on SQuAD datasets for BERT-base vs Decomp-BERT-base, as an average measured in batch mode. On the GPU and CPU we use a batch size 32 and on the phone (marked by *) we use a batch size of 1.

BERT-Base:
1/4: 84 => 82

COMPRESSING BERT: STUDYING THE EFFECTS OF WEIGHT PRUNING ON TRANSFER LEARNING

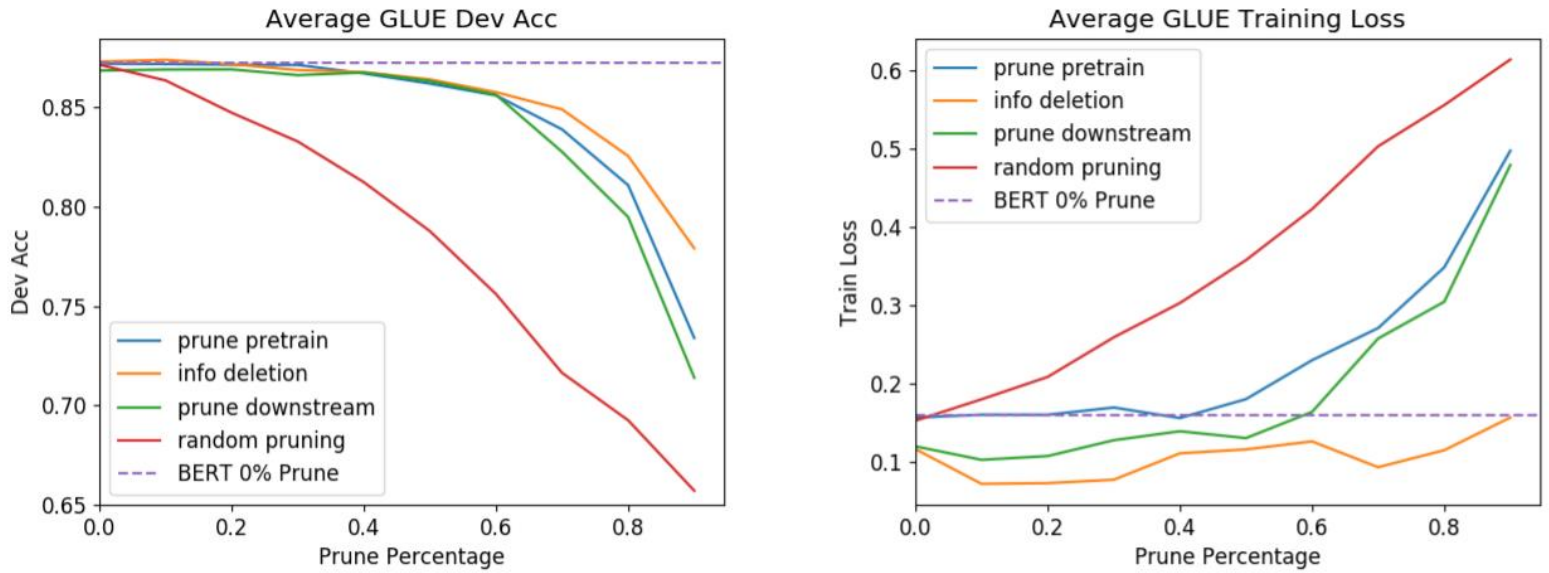

Figure 1: (Blue) The best GLUE dev accuracy and training losses for models pruned during pre-training, averaged over 5 tasks. Also shown are models with information deletion during pre-training (orange), models pruned after downstream fine-tuning (green), and models pruned randomly during pre-training instead of by lowest magnitude (red). 30-40% of weights can be pruned using magnitude weight pruning without decreasing dowsntream accuracy. Notice that information deletion fits the training data better than un-pruned models at all sparsity levels but does not fully recover evaluation accuracy. Also, models pruned after downstream fine-tuning have the same or worse development accuracy, despite achieving lower training losses. Note: none of the pruned models are overfitting because un-pruned models have the lowest training loss and the highest development accuracy. While the results for individual tasks are in Table 1, each task does not vary much from the average trend, with an exception discussed in Section 4.3.

WELL-READ STUDENTS LEARN BETTER: ON THE IMPORTANCE OF PRE-TRAINING COMPACT MODELS

	Model	SST-2 (acc)	MRPC (f1/acc)	QQP (f1/acc)	MNLI (acc m/mm)	QNLI (acc)	RTE (acc)	Meta Score
test	TF (baseline)	90.7	85.9/80.2	69.2/88.2	80.4/79.7	86.7	63.6	80.5
	PF (baseline)	**92.5**	**86.8/81.8**	70.1/88.5	81.8/81.1	87.9	64.2	81.6
	PD (our work)	91.8	**86.8**/81.7	70.4/**88.9**	**82.8/82.2**	88.9	65.3	**82.1**
	Sun et al. (2019a)	92.0	85.0/79.9	**70.7/88.9**	81.5/81.0	**89.0**	**65.5**	81.7
dev	PF (baseline)	91.1	87.9/82.5	86.6/90.0	81.1/81.7	87.8	63.0	82.8
	PD (our work)	91.1	**89.4/84.9**	87.4/**90.7**	**82.5/83.4**	**89.4**	**66.7**	**84.4**
	Sanh (2019)	**92.7**	88.3/82.4	**87.7**/90.6	81.6/81.1	85.5	60.0	82.3

Table 3: **Model Quality.** All students are 6/768 BERT models, trained by 12/768 BERT teachers. Concurrent results are cited as reported by their authors. Our dev results are averaged over 5 runs. Our test results are evaluated on the GLUE server, using the model that performed best on dev. For anchoring, we also provide our results for MLM pre-training followed by fine-tuning (PF) and cite results from Sun et al. (2019a) for $\text{BERT}_{\text{BASE}}$ truncated and fine-tuned (TF). The meta score is computed on 6 tasks only, and is therefore not directly comparable to the GLUE leaderboard.

Figure 3: Pre-trained Distillation (PD) and concurrent work on model compression.

BERT-Base:
1/2: 84 => 82.5
1/3.5: 84 => 80
1/10: 84 => 78
1/25: 84=> 72

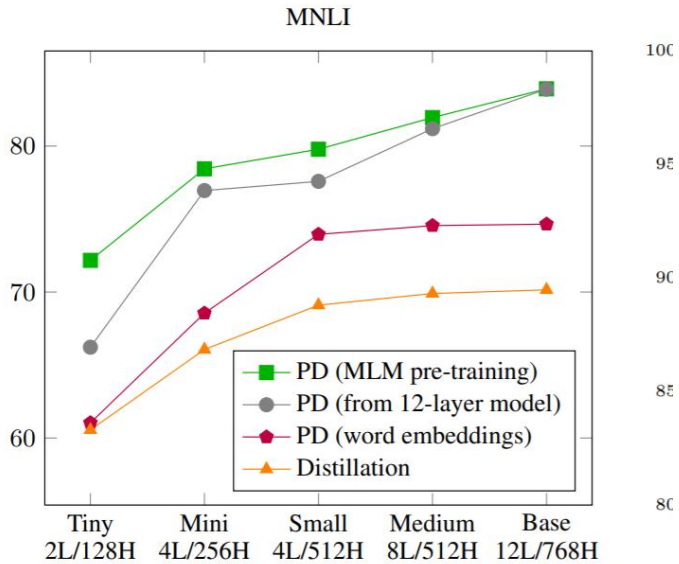

Figure 5: **Pre-training outperforms truncation.** Students initialized via LM pre-training (green) outperform those initialized from the bottom layers of 12-layer pre-trained models (gray). When only word embeddings are pre-trained (red), performance is degraded even further.

Winner:

REWEIGHTED PROXIMAL PRUNING FOR LARGE-SCALE LANGUAGE REPRESENTATION

Fu-Ming Guo[1], Sijia Liu[2], Finlay S. Mungall[3], Xue Lin[1] & Yanzhi Wang[1]

[1]Northeastern University
[2]MIT-IBM Watson AI Lab, IBM Research
[3]United States Federal Aviation Administration
guo.fu@husky.neu.edu, sijia.liu@ibm.com, fmungall@gmail.com
{xue.lin,yanz.wang}@northeastern.edu

> MOBILEBERT: TASK-AGNOSTIC COMPRESSION OF BERT BY PROGRESSIVE KNOWLEDGE TRANSFER

> FASTER AND JUST AS ACCURATE: A SIMPLE DECOMPOSITION FOR TRANSFORMER MODELS

> WELL-READ STUDENTS LEARN BETTER: ON THE IMPORTANCE OF PRE-TRAINING COMPACT MODELS

>> EXTREME LANGUAGE MODEL COMPRESSION WITH OPTIMAL SUBWORDS AND SHARED PROJECTIONS

Sanqiang Zhao, Raghav Gupta, Yang Song, Denny Zhou
Google AI, USA
{sanqiang, raghavgupta, yangso, dennyzhou}@google.com