ALBERT: A LITE BERT FOR SELF-SUPERVISIED LEARNING OF LANGUAGE REPRESENTATIONS
(ICLR'20 papers)

25th Jan, 2022

JongHyeon Kim

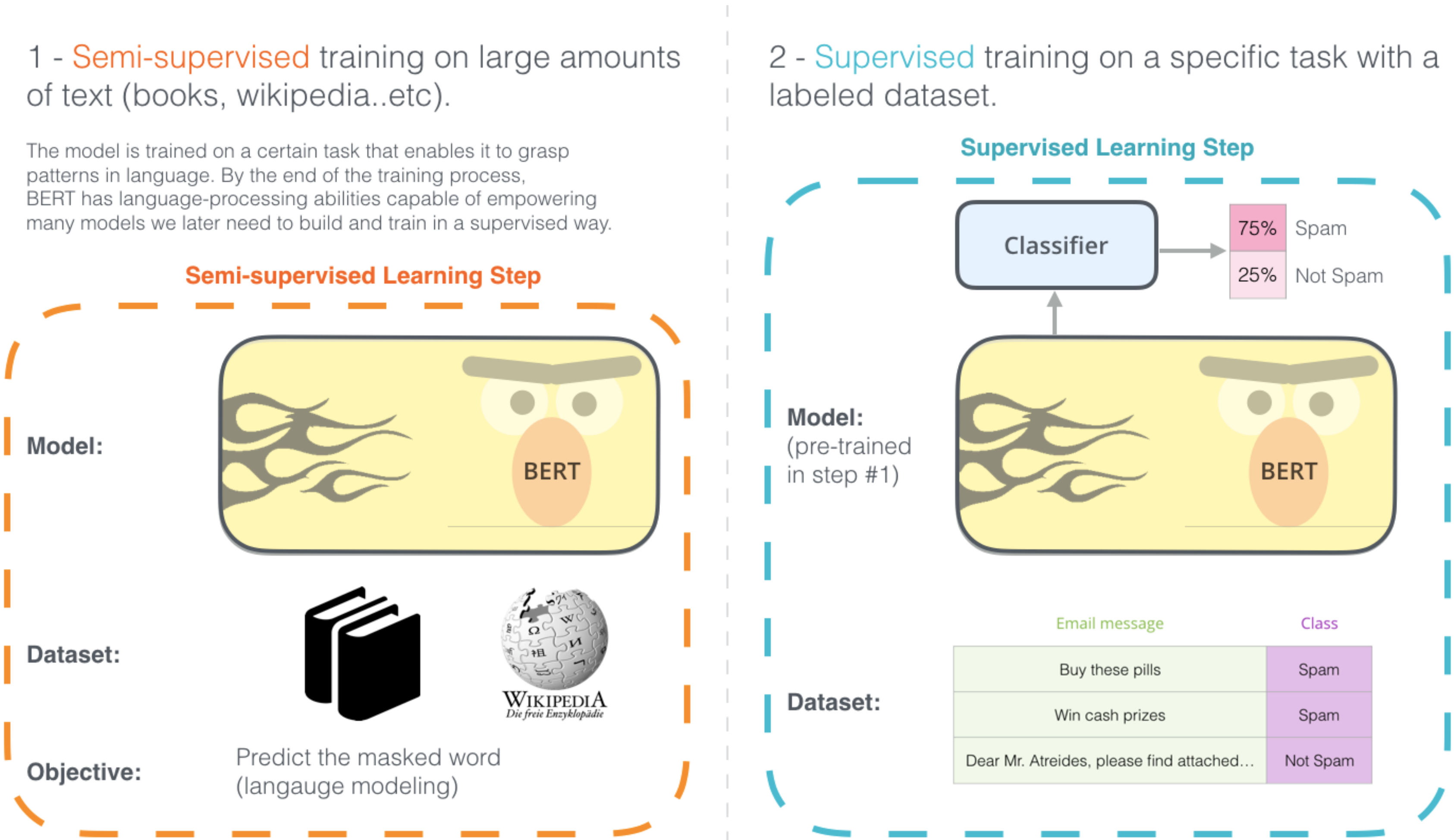
github.com/bellhyeon
bellhyeon@naver.com

INTRODUCTION

Prior to the start of the presentation, I inform you in advance that the prior knowledge has been summarized in references, and that this presentation will be presented mainly by ALBERT.

INTRODUCTION

Full Network Pre-Training



Reference: http://jalammar.github.io/illustrated-bert/

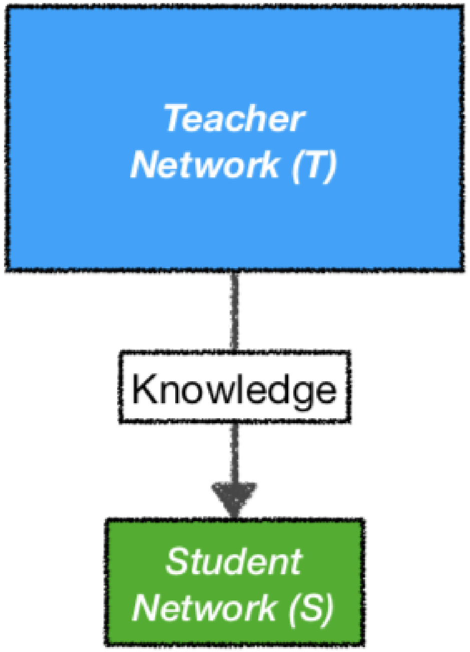
INTRODUCTION

Race Dataset

Model	Report Time	Institute	RACE	RACE-M	RACE-H
Base					
Gated Attention Reader*	Apr 15, 2017	CMU	44.1	43.7	44.2
RoBERTa (SOTA)					
RoBERTa	Jul 26, 2019	Facebook AI	83.2	86.5	81.8
ALBERT					
ALBERT (ensemble)	Sep 26, 2019	Google Research & TTIC	$\boxed{89.4}$	91.2	88.6

Reference: http://www.qizhexie.com/data/RACE_leaderboard.html

Model Distillation



1. **Teacher Network (T)**
   - cumbersome model
   ex) ensemble / a large generalized model
   - (pros) excellent performance
   - (cons) computationally expansive
   - can not be deployed when limited environments

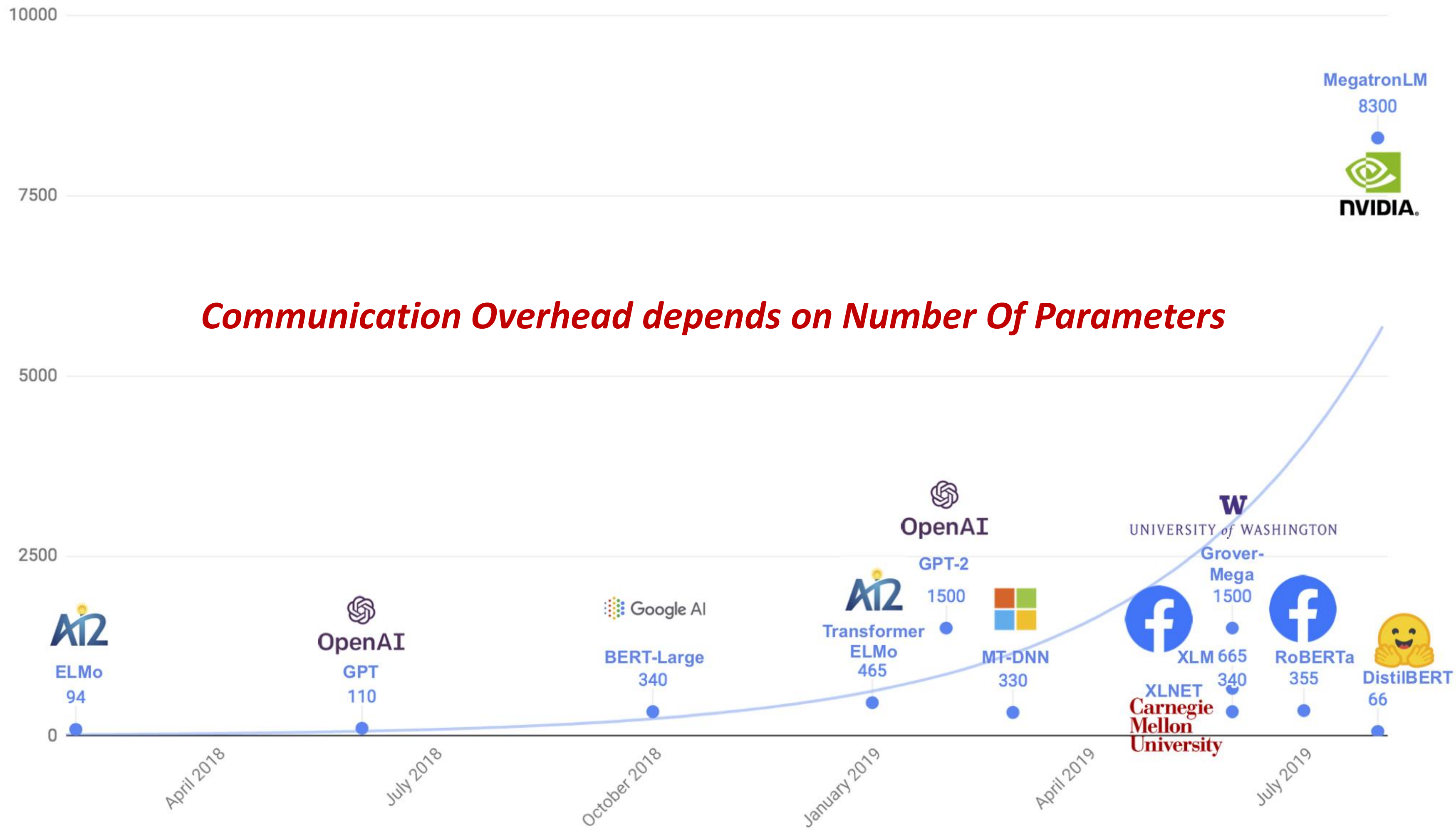
2. **Student Network (S)**
   - small model
   - suitable for deployment
   - (pros) fast inference
   - (cons) lower performance than T

Reference: https://baeseongsu.github.io/posts/knowledge-distillation/

Is having better NLP models as easy as having larger models?

INTRODUCTION

Memory Limitation Problem & Training Time



Reference: https://medium.com/huggingface/distilbert-8cf3380435b5

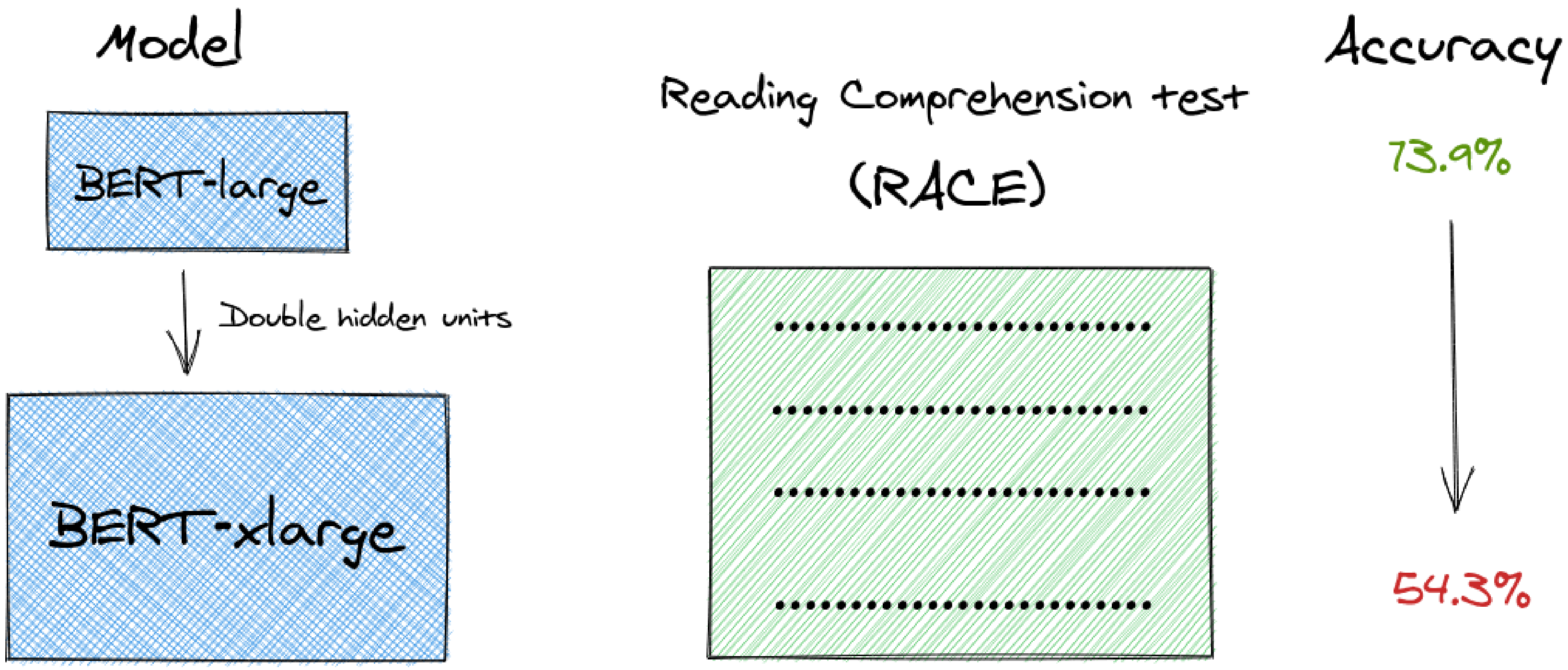
INTRODUCTION

Memory Limitation Problem & Training Time

Model	Size	TPU ($ per hour)	TPU Count (device)	Training Time	Cost (USD)	CO_2 emissions (lbs)
BERT	24 Layers (340M)	v2 ($4.5)	16	4 days	$6,912 (약 850만원)	1428
GPT-2	48 Layers (1542M)	v3 ($8)	32	7 days	$43,008 (약 5,100만원)	2516
XLNet	24 Layers (365M)	v3 ($8)	128	2.5 days	$61,440 (약 7,300만원)	-

Reference: https://medium.com/huggingface/distilbert-8cf3380435b5

Model Degradation - Are Large Models Always The Answer?



Reference: https://amitness.com/2020/02/albert-visual-summary/

Model Degradation - Are Large Models Always The Answer?

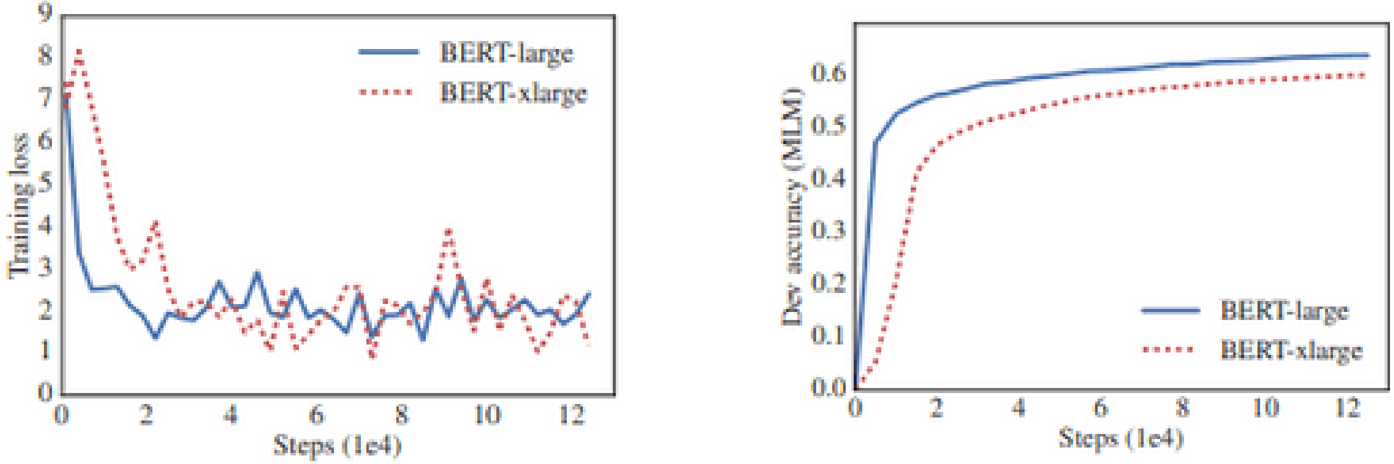


Figure 1: Training loss (left) and dev masked LM accuracy (right) of BERT-large and BERT-xlarge (2x larger than BERT-large in terms of hidden size). The larger model has lower masked LM accuracy while showing no obvious sign of over-fitting.

Model	Hidden Size	Parameters	RACE (Accuracy)
BERT-large (Devlin et al., 2019)	1024	334M	72.0%
BERT-large (ours)	1024	334M	73.9%
BERT-xlarge (ours)	2048	1270M	54.3%

Table 1: Increasing hidden size of BERT-large leads to worse performance on RACE.

A Lite BERT (ALBERT)

- Solved aforementioned Methods

- Fewer parameters than BERT

- Propose parameter reduction techniques
  - Factorized embedding parameterization
  - Cross-layer parameter sharing

- Sentence-order prediction (SOP)

Reference: https://github.com/cybertronai/gradient-checkpointing

RELATED WORK

Scaling Up Representation Learning For Natural Language

- Often shown that larger model size improves performance
- Larger model size always leads to better performance in BERT
  (Larger Hidden Size, More Hidden Layers & Attention Heads)

GLUE

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

SQuAD 1.1

System	Dev EM	Dev F1	Test EM	Test F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	-	-	82.3	91.2
#1 Ensemble - nlnet	-	-	86.0	91.7
#2 Ensemble - QANet	-	-	84.5	90.5
Published				
BiDAF+ELMo (Single)	-	85.6	-	85.8
R.M. Reader (Ensemble)	81.2	87.9	82.3	88.5
Ours				
$\text{BERT}_{\text{BASE}}$ (Single)	80.8	88.5	-	-
$\text{BERT}_{\text{LARGE}}$ (Single)	84.1	90.9	-	-
$\text{BERT}_{\text{LARGE}}$ (Ensemble)	85.8	91.8	-	-
$\text{BERT}_{\text{LARGE}}$ (Sgl.+TriviaQA)	**84.2**	**91.1**	**85.1**	**91.8**
$\text{BERT}_{\text{LARGE}}$ (Ens.+TriviaQA)	**86.2**	**92.2**	**87.4**	**93.2**

SWAG Dev & Test

System	Dev	Test
ESIM+GloVe	51.9	52.7
ESIM+ELMo	59.1	59.2
OpenAI GPT	-	78.0
$\text{BERT}_{\text{BASE}}$	81.6	-
$\text{BERT}_{\text{LARGE}}$	**86.6**	**86.3**
Human (expert)†	-	85.0
Human (5 annotations)†	-	88.0

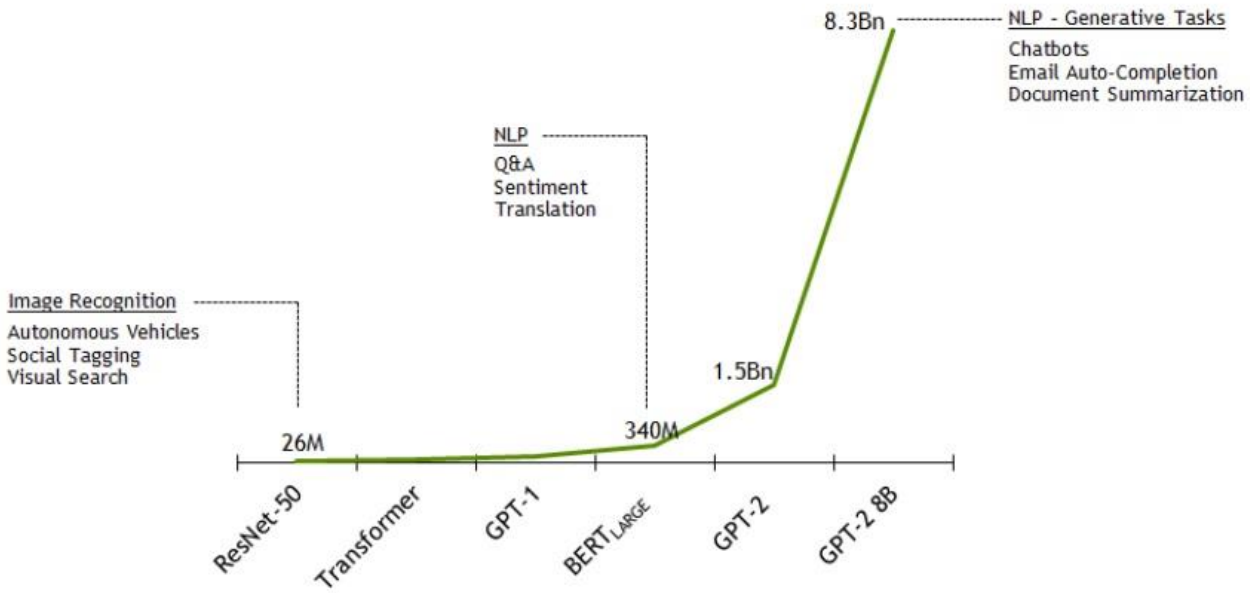
Reference: https://arxiv.org/pdf/1810.04805.pdf

RELATED WORK

Scaling Up Representation Learning For Natural Language

- Experiment with large models is difficult → due to computational constraints
  GPU/TPU memory limitations

- Current SOTA Models have millions or billions of parameters



Reference: https://han.gl/ULJWI

RELATED WORK

Scaling Up Representation Learning For Natural Language

- Gradient Checkpointing *(Training Deep Nets with Sublinear Memory Cost, Chen et al., 2016)*
- Reconstruct Each Layer's Activations From the Next Layer
  (The Reversible Residual Network: Backpropagation Without Storing Activations, Gomez et al., 2017)

These Solutions solved only Memory Limitation Problem !!

RELATED WORK

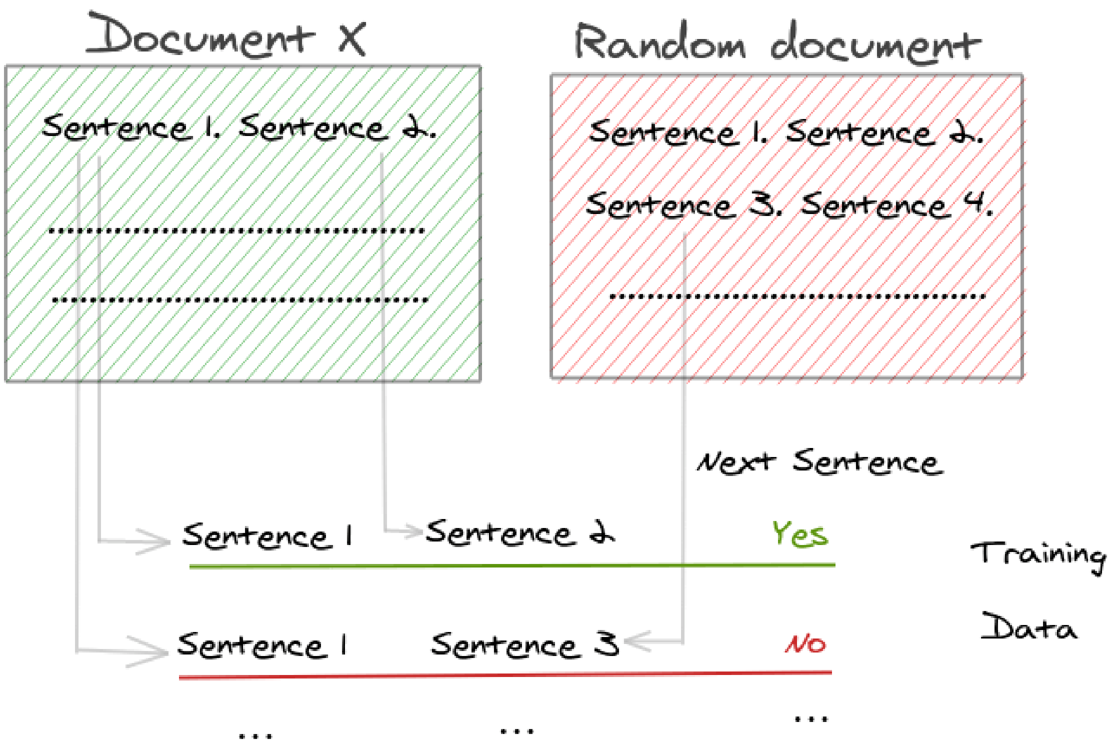
Cross-Layer Parameter Sharing

- Previously Explored with Transformer Architecture *(Attention is All You Need, Vaswani et al., 2017)*
  - Focused on Training Encoder-Decoder Architecture rather than Pretraining / Finetuning
- Universal Transformers *(Dehghani et al., 2018)*
  - Better Performance on Language Modeling and Subject-Verb Agreement than Vanilla Transformer
- Deep Equilibrium Models *(Bai et al., 2019)*
  - DQE can reach an equilibrium point for which the input embedding and the output embedding of a certain layer stay the same
- Modeling Recurrence for Transformers *(Hao et al., 2019)*
  - Vanilla Transformer Encoder + Recurrence Encoder Structure

RELATED WORK

Sentence Ordering Objectives

- BERT uses Next Sentence Prediction (NSP) Loss for downstream tasks
  - Take two segments that appear consecutively from the training corpus
  - Create a random pair of segments from the different document as negative samples



Reference: https://amitness.com/2020/02/albert-visual-summary/

RELATED WORK

Sentence Ordering Objectives

- Papers like RoBERTa and XLNET have shed light on the ineffectiveness of NSP and found its impact on the downstream tasks unreliable
- NSP sees not only the continuity, but also the topic of the sentence
- May be judged as a negative example by the different topic → Topic Prediction

THE ELEMENTS OF ALBERT

Model Architecture Choices

- Transformer Encoder + GELU Activation Function (Similar To BERT)
- Follows BERT notation conventions
  - Vocabulary Embedding Size: E
  - Number of Encoder Layers: L
  - Hidden Size: H
  - Feed-Ford-Network(FFN) Size: $4H$
  - Attention Heads: $H/64$

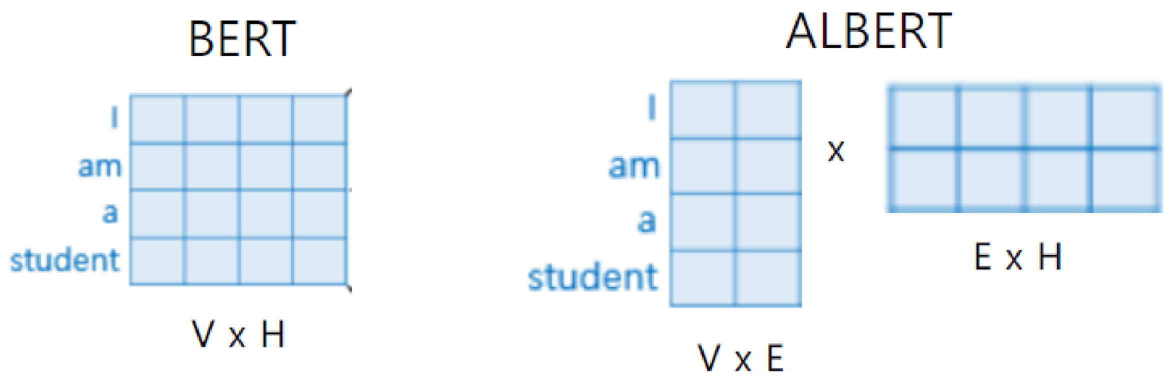
THE ELEMENTS OF ALBERT

Model Architecture Choices

1. Factorized embedding parameterization

- Modeling Perspective
  - WordPiece embedding learns context-independent representations
  - Hidden layer embedding learns context-dependent representations
  - WordPiece Embedding Information << Hidden Layer Embedding Information

- Practical Perspective
  - NLP usually require the large vocab size
  - If increase hidden size H when $\mathrm{E} = H$, embedding matrix = $(\mathrm{V} \times E(= H))$
  - If H != E, embedding matrix = $\mathrm{V} \times H + \mathrm{V} \times E$ $(H > \mathrm{E})$



Reference: https://han.gl/ULJWI

Model Architecture Choices

2. Cross-layer parameter sharing

- ALBERT proposes cross-layer paramaeter sharing
  - Only sharing feed-forward network
  - Only sharing multi-head attention parameters
  - Default: Sharing All parameters across layers

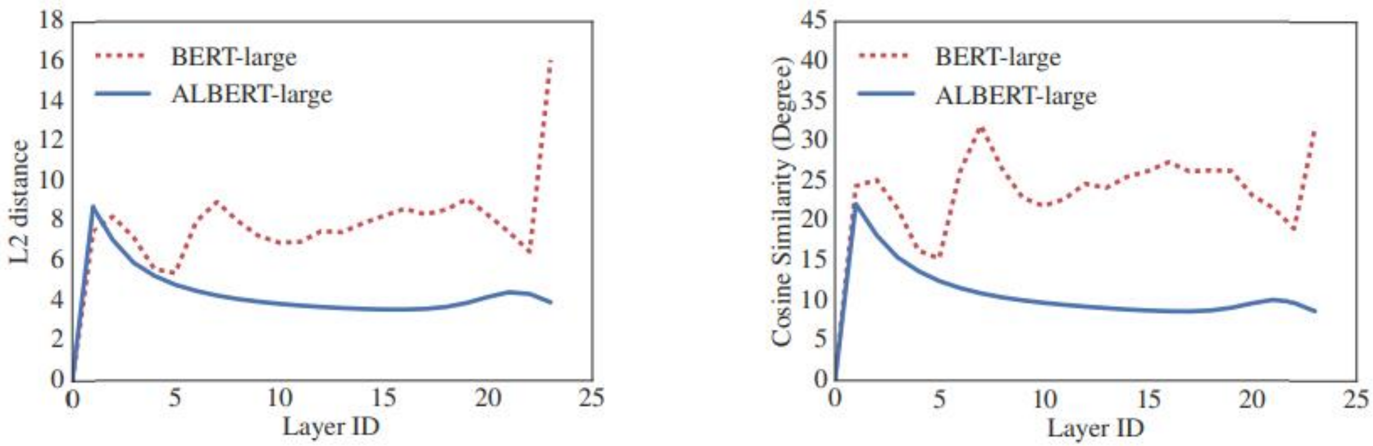


Figure 1: The L2 distances and cosine similarity (in terms of degree) of the input and output embedding of each layer for BERT-large and ALBERT-large.

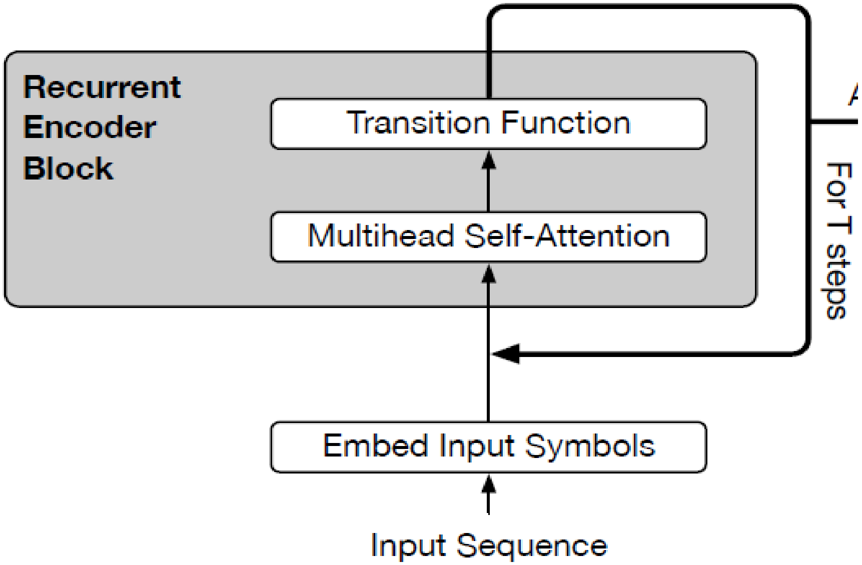
- Weight-Sharing has an effect on stabilizing network parameters

Model Architecture Choices

2. Cross-layer parameter sharing

- Similar to Universal Transformer

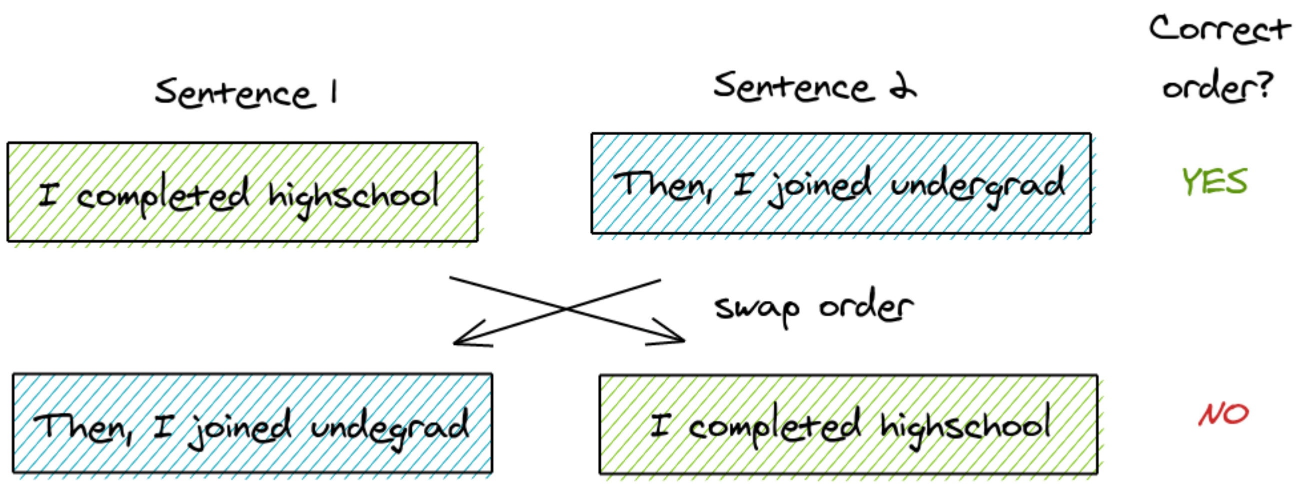
- Recurrent Transformer Encoder



Model Architecture Choices

3. Inter-sentence coherence loss

- ALBERT proposes Sentence Order Prediction (SOP)
  - Take two consecutive segments from the same document as a positive class
  - Swap the order of the same segment and use that as a negative example



EXPERIMENTAL RESULTS

EXPERIMENTAL SETUP

- Pre-train Corpora: Bookcorpus, English Wikipedia
- Same input format as BERT: [CLS] x_1 [SEP] x_2 [SEP]
- 30,000 Vocab tokenized by SentencePiece (Same as BERT)
- MLM targets: n-gram masking ($1 \leq n \leq 3$) $p(n) = \frac{1/n}{\sum_{k=1}^{N} 1/k}$
- Batch Size: 4096
- Optimizer: Lamb *(Large Batch Optimization for Deep Learning, You et al., 2019)*
- Learning Rate: 0.00176
- Used Google Cloud TPU v3: 64 ~ 512 used depending on model size
- Train Steps: 125,000 steps

EXPERIMENTAL RESULTS

EVALUATION BENCHMARKS

Overall Comparison Between BERT and ALBERT

- ALBERT-xxlarge
  - Only around 70% of BERT-large's parameters
  - Achieves significant improvements over BERT-large

- Training time faster under the same TPUS
  - Because of less communication and fewer computation
  - ALBERT-large is about 1.7 times faster than BERT-large
  - ALBERT-xxlarge is about 3 times slower than BERT-large (larger structure)

Model		Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg	Speedup
BERT	base	108M	90.4/83.2	80.4/77.6	84.5	92.8	68.2	82.3	4.7x
	large	334M	92.2/85.5	85.0/82.2	86.6	93.0	73.9	85.2	1.0
ALBERT	base	12M	89.3/82.3	80.0/77.1	81.6	90.3	64.0	80.1	5.6x
	large	18M	90.6/83.9	82.3/79.4	83.5	91.7	68.5	82.4	1.7x
	xlarge	60M	92.5/86.1	86.1/83.1	86.4	92.4	74.8	85.5	0.6x
	xxlarge	235M	**94.1/88.3**	**88.1/85.1**	**88.0**	**95.2**	**82.3**	**88.7**	0.3x

EXPERIMENTAL RESULTS

EVALUATION BENCHMARKS

Factorized Embedding Parameterization

- Under the non-shared condition (BERT-style)
  - Larger embedding sizes give better performance, but not by much

- Under the all-shared condition (ALBERT-style)
  - Best: Embedding size 128
  - Use embedding size E = 128 in all future settings

Model	E	Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
ALBERT base not-shared	64	87M	89.9/82.9	80.1/77.8	82.9	91.5	66.7	81.3
	128	89M	89.9/82.8	80.3/77.3	83.7	91.5	67.9	81.7
	256	93M	90.2/83.2	80.3/77.4	84.1	91.9	67.3	81.8
	768	108M	90.4/83.2	80.4/77.6	84.5	92.8	68.2	82.3
ALBERT base all-shared	64	10M	88.7/81.4	77.5/74.8	80.8	89.4	63.5	79.0
	128	12M	89.3/82.3	80.0/77.1	81.6	90.3	64.0	80.1
	256	16M	88.8/81.5	79.1/76.3	81.5	90.3	63.4	79.6
	768	31M	88.6/81.5	79.2/76.6	82.0	90.6	63.3	79.8

EXPERIMENTAL RESULTS

EVALUATION BENCHMARKS

Cross-Layer Parameter Sharing

- Not shared strategy is best
- Shared-Attention strategy hurts a little
- Shared-FFN strategy hurts more than Shared-Attention strategy
- Shared-Attention has more parameters than Shared-FFN

Model		Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
ALBERT base E=768	all-shared	31M	88.6/81.5	79.2/76.6	82.0	90.6	63.3	79.8
	shared-attention	83M	89.9/82.7	80.0/77.2	84.0	91.4	67.7	81.6
	shared-FFN	57M	89.2/82.1	78.2/75.4	81.5	90.8	62.6	79.5
	not-shared	108M	90.4/83.2	80.4/77.6	84.5	92.8	68.2	82.3
ALBERT base E=128	all-shared	12M	89.3/82.3	80.0/77.1	82.0	90.3	64.0	80.1
	shared-attention	64M	89.9/82.8	80.7/77.9	83.4	91.9	67.6	81.7
	shared-FFN	38M	88.9/81.6	78.6/75.6	82.3	91.7	64.4	80.2
	not-shared	89M	89.9/82.8	80.3/77.3	83.2	91.5	67.9	81.6

EXPERIMENTAL RESULTS

EVALUATION BENCHMARKS

Sentence Order Prediction (SOP)

- Compare None inter-sentence loss (XLNet / RoBERTa Style), NSP loss (BERT-Style), SOP loss (ALBERT Style)

- NSP loss does not effect to SOP task

- SOP loss effect to both NSP task and SOP task

- SOP loss appears to consistently improve downstream task performance for multi-sentence encoding

	Intrinsic Tasks			Downstream Tasks					
SP tasks	MLM	NSP	SOP	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
None	54.9	52.4	53.3	88.6/81.5	78.1/75.3	81.5	89.9	61.7	79.0
NSP	54.5	90.5	52.0	88.4/81.5	77.2/74.6	81.6	**91.1**	62.3	79.2
SOP	54.0	78.9	86.5	**89.3/82.3**	**80.0/77.1**	**82.0**	90.3	**64.0**	**80.1**

EXPERIMENTAL RESULTS

EVALUATION BENCHMARKS

What If We train For the Same Amout Of Time?

- ALBERT-xxlarge is significantly better than BERT-large: Avg (+1.5%) , Race (+5.2%)

Models	Steps	Time	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
BERT-large	400k	34h	93.5/87.4	86.9/84.3	87.8	94.6	77.3	87.2
ALBERT-xxlarge	125k	32h	**94.0/88.1**	**88.3/85.3**	87.8	**95.4**	**82.5**	**88.7**

EXPERIMENTAL RESULTS

EVALUATION BENCHMARKS

What If We train For the Same Amout Of Time?

- Compared by not number of training steps, but actual training time
  → Due to longer training usually leads to better performance

- ALBERT-xxlarge is significantly better than BERT-large: Avg (+1.5%) , Race (+5.2%)

Models	Steps	Time	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
BERT-large	400k	34h	93.5/87.4	86.9/84.3	87.8	94.6	77.3	87.2
ALBERT-xxlarge	125k	32h	**94.0/88.1**	**88.3/85.3**	87.8	**95.4**	**82.5**	**88.7**

EXPERIMENTAL RESULTS

EVALUATION BENCHMARKS

Additional Training Data And Dropout Effects

- With additional data improves performance
  → Except SQuAD: Due to ALBERT pretrained with Wikipedia corpus

- ALBERT-xxlarge is significantly better than BERT-large: Avg (+1.5%) , Race (+5.2%)

	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
No additional data	**89.3/82.3**	**80.0/77.1**	81.6	90.3	64.0	80.1
With additional data	88.8/81.7	79.1/76.3	**82.4**	**92.8**	**66.0**	**80.8**

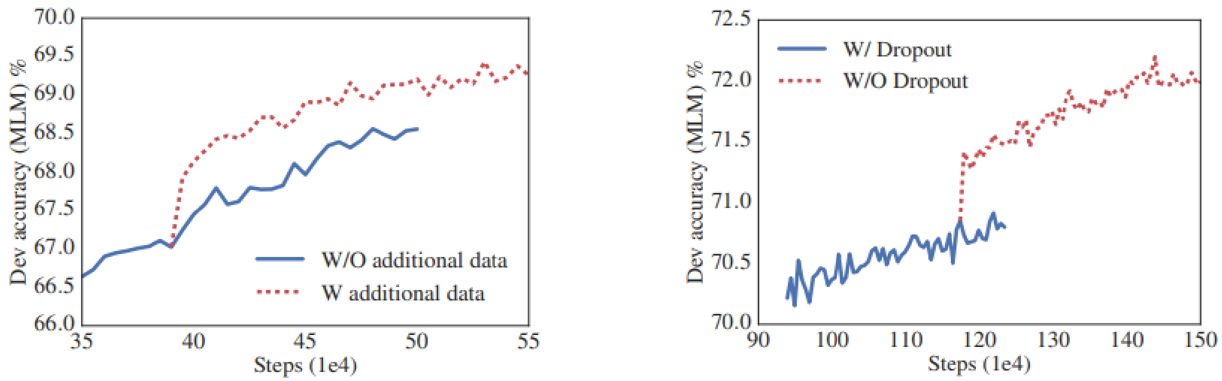
EXPERIMENTAL RESULTS

EVALUATION BENCHMARKS

Additional Training Data And Dropout Effects

- ALBERT-xxlarge does not overfitted after training 1M steps
  → Remove Dropout for increase model capacity

- Removing dropout leads to improvements for all downstream tasks
  → Needs further experimentation with other transformer-based architectures



(a) Adding data (b) Removing dropout

	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
With dropout	94.7/89.2	89.6/86.9	90.0	96.3	85.7	90.4
Without dropout	**94.8/89.5**	**89.9/87.2**	**90.4**	**96.5**	**86.1**	**90.7**

EXPERIMENTAL RESULTS

EVALUATION BENCHMARKS

Current State-Of-The-Art On NLU Tasks

Models	MNLI	QNLI	QQP	RTE	SST	MRPC	CoLA	STS	WNLI	Avg
Single-task single models on dev										
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-large	90.2	94.7	**92.2**	86.6	96.4	**90.9**	68.0	92.4	-	-
ALBERT (1M)	90.4	95.2	92.0	88.1	96.8	90.2	68.7	92.7	-	-
ALBERT (1.5M)	**90.8**	**95.3**	**92.2**	**89.2**	**96.9**	**90.9**	**71.4**	**93.0**	-	-
Ensembles on test (from leaderboard as of Sept. 16, 2019)										
ALICE	88.2	95.7	**90.7**	83.5	95.2	92.6	**69.2**	91.1	80.8	87.0
MT-DNN	87.9	96.0	89.9	86.3	96.5	92.7	68.4	91.1	89.0	87.6
XLNet	90.2	98.6	90.3	86.3	96.8	93.0	67.8	91.6	90.4	88.4
RoBERTa	90.8	98.9	90.2	88.2	96.7	92.3	67.8	92.2	89.0	88.5
Adv-RoBERTa	91.1	98.8	90.3	88.7	96.8	93.1	68.0	92.4	89.0	88.8
ALBERT	**91.3**	**99.2**	90.5	**89.2**	**97.1**	**93.4**	69.1	**92.5**	**91.8**	**89.4**

EXPERIMENTAL RESULTS

EVALUATION BENCHMARKS

Current State-Of-The-Art On NLU Tasks

Models	SQuAD1.1 dev	SQuAD2.0 dev	SQuAD2.0 test	RACE test (Middle/High)
Single model (from leaderboard as of Sept. 23, 2019)				
BERT-large	90.9/84.1	81.8/79.0	89.1/86.3	72.0 (76.6/70.1)
XLNet	94.5/89.0	88.8/86.1	89.1/86.3	81.8 (85.5/80.2)
RoBERTa	94.6/88.9	89.4/86.5	89.8/86.8	83.2 (86.5/81.3)
UPM	-	-	89.9/87.2	-
XLNet + SG-Net Verifier++	-	-	90.1/87.2	-
ALBERT (1M)	94.8/89.2	89.9/87.2	-	86.0 (88.2/85.1)
ALBERT (1.5M)	**94.8/89.3**	**90.2/87.4**	**90.9/88.1**	**86.5 (89.0/85.5)**
Ensembles (from leaderboard as of Sept. 23, 2019)				
BERT-large	92.2/86.2	-	-	-
XLNet + SG-Net Verifier	-	-	90.7/88.2	-
UPM	-	-	90.7/88.2	
XLNet + DAAF + Verifier	-	-	90.9/88.6	-
DCMN+	-	-	-	84.1 (88.5/82.3)
ALBERT	**95.5/90.1**	**91.4/88.9**	**92.2/89.7**	**89.4 (91.2/88.6)**

CONCLUSION & DISCUSSION

- ALBERT-xxlarge has less parameters than BERT-large, computation cost is expensive due to larger structure

- Next step to speed up the training and inference speed methods
  Sparse Attention(Child et al., 2019) / Block Attention (Shen et al., 2018)

- Next step to more better representation
  Hard Example Mining (Mikolove et al., 2013) / Efficient Language Model Training (Yang et al., 2019)

- Hypothesize that there could be another self-supervised training loss

CONCLUSION & DISCUSSION

- ALBERT-xxlarge has less parameters than BERT-large, computation cost is expensive due to larger structure

- 12 repeating layers
- 128 embedding dimension
- 4096 hidden dimension
- 64 attention heads
- 223M parameters

ALBERT-xxlarge Model Configuration

- 24-layer
- 1024 hidden dimension
- 16 attention heads
- 336M parameters.

BERT-large Model Configuration

Reference: https://huggingface.co/albert-xxlarge-v2; https://huggingface.co/bert-large-uncased

Q&A

Q&A 질문과 답변

Thank You!