How Powerful are BERTs ?

	Rank	Name	Model	URL	Score	CoLA	SST-2	MRPC	STS-B	QQP	MNLI-m	MNLI-mm	QNLI	RTE	WNLI	AX
	1	Facebook AI	RoBERTa		88.5	67.8	96.7	92.3/89.8	92.2/91.9	74.3/90.2	90.8	90.2	98.9	88.2	89.0	48.7
	2	XLNet Team	XLNet-Large (ensemble)		88.4	67.8	96.8	93.0/90.7	91.6/91.1	74.2/90.3	90.2	89.8	98.6	86.3	90.4	47.5
+	3	Microsoft D365 AI & MSR AI	MT-DNN-ensemble		87.6	68.4	96.5	92.7/90.3	91.1/90.7	73.7/89.9	87.9	87.4	96.0	86.3	89.0	42.8
	4	GLUE Human Baselines	GLUE Human Baselines		87.1	66.4	97.8	86.3/80.8	92.7/92.6	59.5/80.4	92.0	92.8	91.2	93.6	95.9	-
+	5	王玮	ALICE large ensemble (Alibaba DAMO NLP)		86.3	68.6	95.2	92.6/90.2	91.1/90.6	74.4/90.7	88.2	87.9	95.7	83.5	80.8	43.9
	6	Stanford Hazy Research	Snorkel MeTaL		83.2	63.8	96.2	91.5/88.5	90.1/89.7	73.1/89.9	87.6	87.2	93.9	80.9	65.1	39.9
	7	XLM Systems	XLM (English only)		83.1	62.9	95.6	90.7/87.1	88.8/88.2	73.2/89.8	89.1	88.5	94.0	76.0	71.9	44.7
	8	张倬胜	SemBERT		82.9	62.3	94.6	91.2/88.3	87.8/86.7	72.8/89.8	87.6	86.3	94.6	84.5	65.1	42.4
	9	Danqi Chen	SpanBERT (single-task training)		82.8	64.3	94.8	90.9/87.9	89.9/89.1	71.9/89.5	88.1	87.7	94.3	79.0	65.1	45.1
	10	Kevin Clark	BERT + BAM		82.3	61.5	95.2	91.3/88.3	88.6/87.9	72.5/89.7	86.6	85.8	93.1	80.4	65.1	40.7
	11	Nitish Shirish Keskar	Span-Extractive BERT on STILTs		82.3	63.2	94.5	90.6/87.6	89.4/89.2	72.2/89.4	86.5	85.8	92.5	79.8	65.1	28.3
	12	Jason Phang	BERT on STILTs		82.0	62.1	94.3	90.2/86.6	88.7/88.3	71.9/89.4	86.4	85.6	92.7	80.1	65.1	28.3
+	13	Jacob Devlin	BERT: 24-layers, 16-heads, 1024-hidden		80.5	60.5	94.9	89.3/85.4	87.6/86.5	72.1/89.3	86.7	85.9	92.7	70.1	65.1	39.6
	14	Neil Houlsby	BERT + Single-task Adapters		80.2	59.2	94.3	88.7/84.3	87.3/86.1	71.5/89.4	85.4	85.0	92.4	71.6	65.1	9.2
	15	Zhuohan Li	Macaron Net-base		79.7	57.6	94.0	88.4/84.4	87.5/86.3	70.8/89.0	85.4	84.5	91.6	70.5	65.1	38.7
	16	Linyuan Gong	StackingBERT-Base		78.4	56.2	93.9	88.2/83.9	84.2/82.5	70.4/88.7	84.4	84.2	90.1	67.0	65.1	36.6

GLUE Benchmark Leaderboard

What will we talk about today

- Recent Highlights of BERT-like models
  - XLNet and A Fair Comparison Study of XLNet and BERT
  - RoBERTa
  - SpanBERT
  - MT-DNN and MT-DNN with Knowledge Distillation
  - ERNIE
- Recent In-depth Analyses of BERT-like Models in NLP Tasks
  - BERT in Argument Reasoning Comprehension Task
  - BERT in Natural Language Inference Task

A Fair Comparison Study of XLNet and BERT

(XLNet Team)

Independence Assumption

$$\max_{\theta} \quad \log p_{\theta}(\bar{\mathbf{x}} \mid \hat{\mathbf{x}}) \approx \sum_{t=1}^{T} m_t \log p_{\theta}(x_t \mid \hat{\mathbf{x}}) = \sum_{t=1}^{T} m_t \log \frac{\exp\left(H_{\theta}(\hat{\mathbf{x}})_t^{\top} e(x_t)\right)}{\sum_{x'} \exp\left(H_{\theta}(\hat{\mathbf{x}})_t^{\top} e(x')\right)}$$

$$\max_{\theta} \quad \log p_{\theta}(\mathbf{x}) = \sum_{t=1}^{T} \log p_{\theta}(x_t \mid \mathbf{x}_{<t}) = \sum_{t=1}^{T} \log \frac{\exp\left(h_{\theta}(\mathbf{x}_{1:t-1})^{\top} e(x_t)\right)}{\sum_{x'} \exp\left(h_{\theta}(\mathbf{x}_{1:t-1})^{\top} e(x')\right)}$$

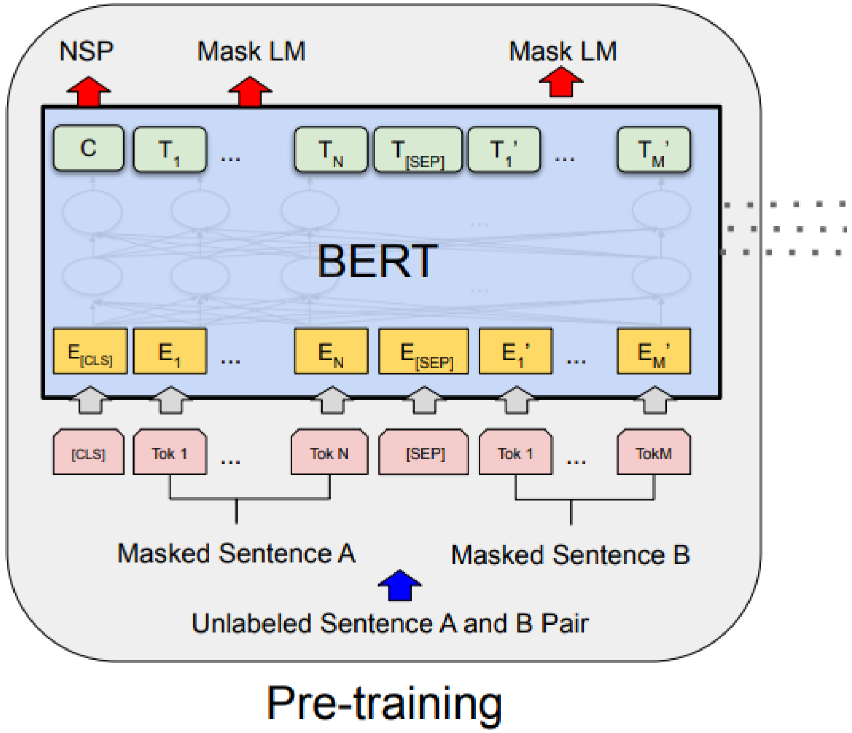


Illustration of BERT Model

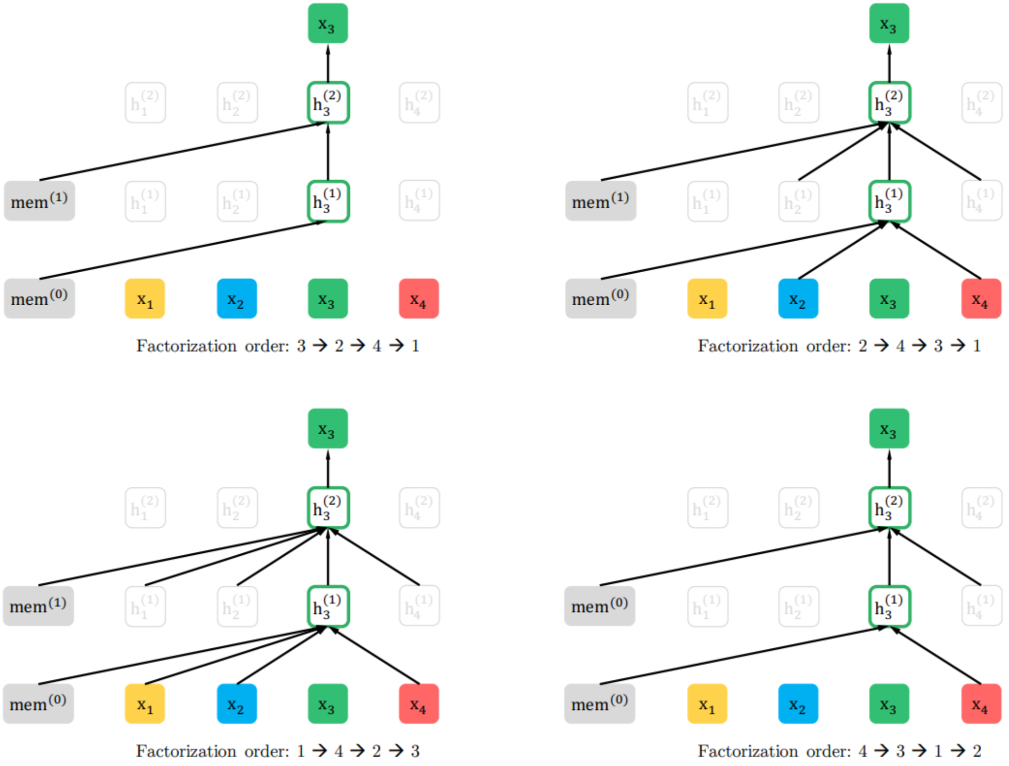
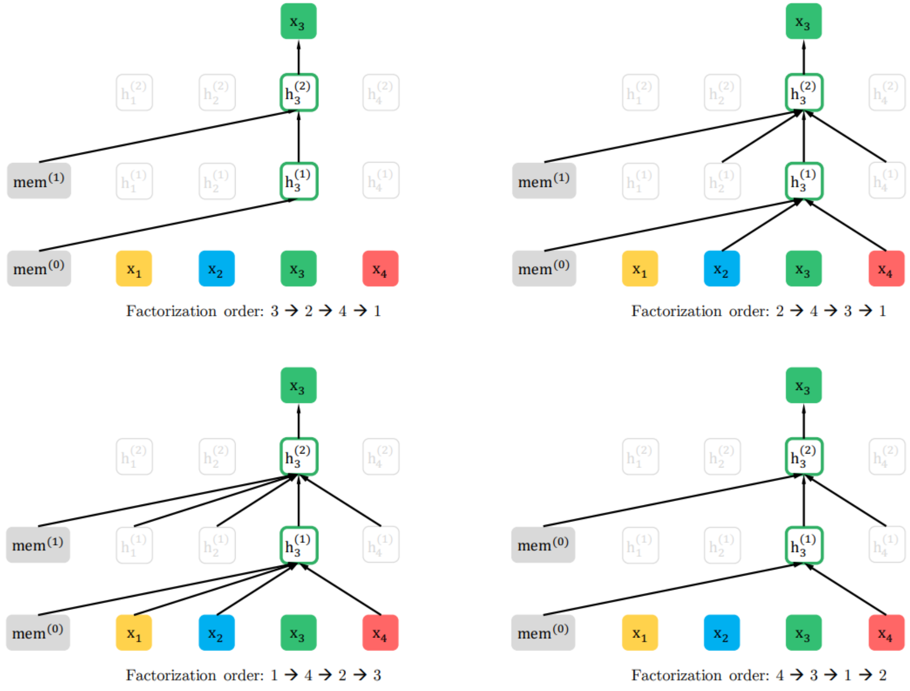


Illustration of XLNet Model

XLNet: Generalized Autoregressive Pretraining for Language Understanding

(Yang et al. CoRR abs/1906.08237)



New Target: $$\max_{\theta} \quad \mathbb{E}_{\mathbf{z}\sim\mathcal{Z}_T}\left[\sum_{t=1}^{T} \log p_\theta(x_{z_t} \mid \mathbf{x}_{\mathbf{z}_{<t}})\right]$$

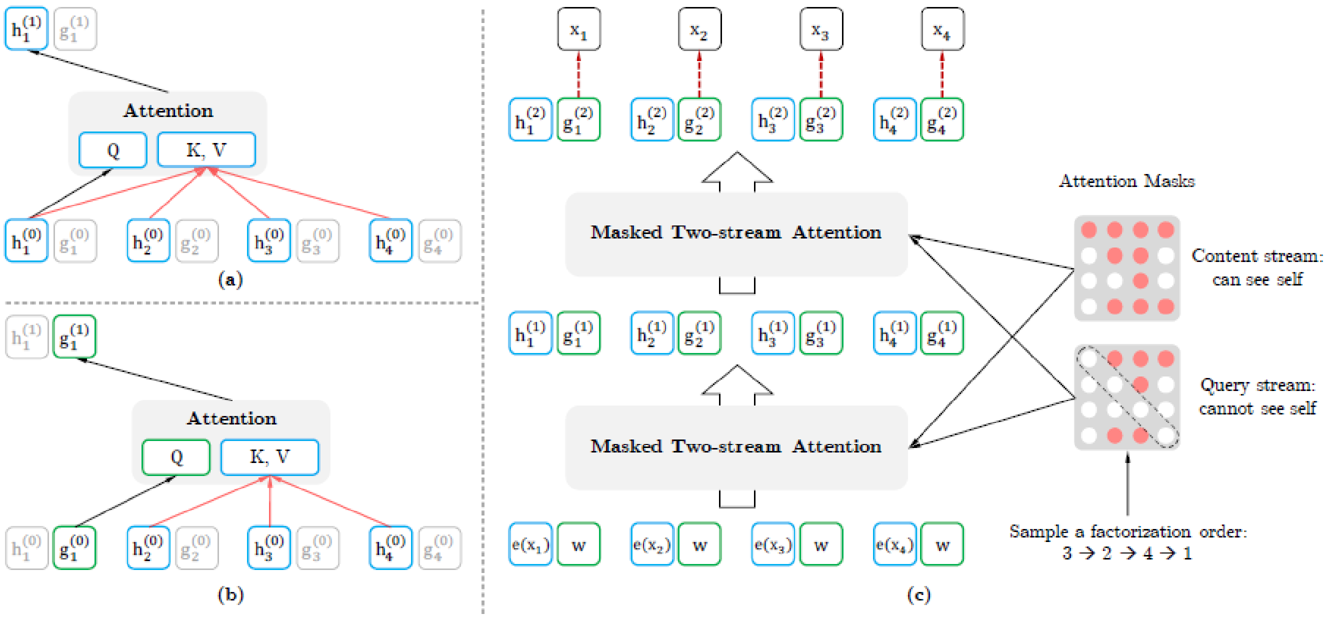
Position Info: $$p_\theta(X_{z_t} = x \mid \mathbf{x}_{z_{<t}}) = \frac{\exp\left(e(x)^\top g_\theta(\mathbf{x}_{\mathbf{z}_{<t}}, z_t)\right)}{\sum_{x'} \exp\left(e(x')^\top g_\theta(\mathbf{x}_{\mathbf{z}_{<t}}, z_t)\right)}$$

Partial Prediction:

$$\max_{\theta} \quad \mathbb{E}_{\mathbf{z}\sim\mathcal{Z}_T}\left[\log p_\theta(\mathbf{x}_{\mathbf{z}_{>c}} \mid \mathbf{x}_{\mathbf{z}_{\le c}})\right] = \mathbb{E}_{\mathbf{z}\sim\mathcal{Z}_T}\left[\sum_{t=c+1}^{|\mathbf{z}|} \log p_\theta(x_{z_t} \mid \mathbf{x}_{\mathbf{z}_{<t}})\right]$$

XLNet: Generalized Autoregressive Pretraining for Language Understanding

(Yang et al. CoRR abs/1906.08237)



Two Attention Streams:

query stream: use z_t but cannot see x_{z_t}

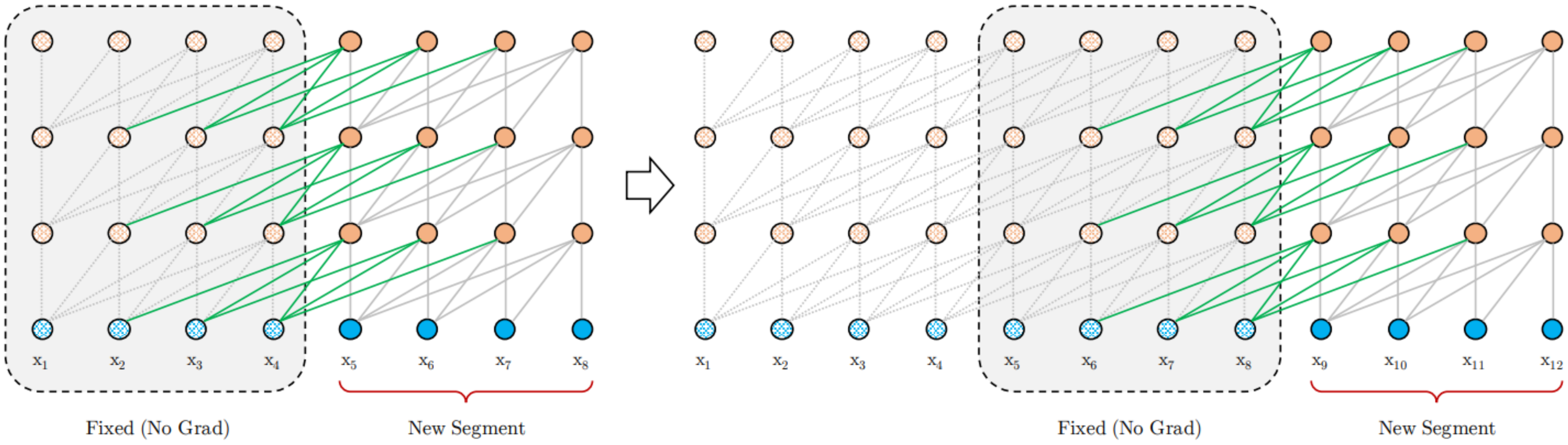
$$g_{z_t}^{(m)} \leftarrow \text{Attention}(\mathrm{Q} = g_{z_t}^{(m-1)}, \mathrm{KV} = \mathbf{h}_{\mathbf{z}_{<t}}^{(m-1)}; \theta)$$

content stream: use both z_t and x_{z_t}

$$h_{z_t}^{(m)} \leftarrow \text{Attention}(\mathrm{Q} = h_{z_t}^{(m-1)}, \mathrm{KV} = \mathbf{h}_{\mathbf{z}_{\leq t}}^{(m-1)}; \theta)$$

XLNet: Generalized Autoregressive Pretraining for Language Understanding

(Yang et al. CoRR abs/1906.08237)



Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context

(Dai et al. CoRR abs/1901.02860)

$$h_{z_t}^{(m)} \leftarrow \text{Attention}(\mathrm{Q} = h_{z_t}^{(m-1)}, \mathrm{KV} = \left[\tilde{\mathrm{h}}^{(m-1)}, \mathrm{h}_{\mathbf{z}_{\leq t}}^{(m-1)}\right]; \theta)$$

Recurrence Mechanism

XLNet: Generalized Autoregressive Pretraining for Language Understanding

(Yang et al. CoRR abs/1906.08237)

<u>New York</u> is a city

Independence Assumption

$$\max_{\theta} \quad \log p_{\theta}(\bar{\mathbf{x}} \mid \hat{\mathbf{x}}) \approx \sum_{t=1}^{T} m_t \log p_{\theta}(x_t \mid \hat{\mathbf{x}}) = \sum_{t=1}^{T} m_t \log \frac{\exp\left(H_{\theta}(\hat{\mathbf{x}})_t^{\top} e(x_t)\right)}{\sum_{x'} \exp\left(H_{\theta}(\hat{\mathbf{x}})_t^{\top} e(x')\right)}$$

$$\mathcal{J}_{\text{BERT}} = \log p(\text{New} \mid \text{is a city}) + \log p(\text{York} \mid \text{is a city})$$

$$\max_{\theta} \quad \log p_{\theta}(\mathbf{x}) = \sum_{t=1}^{T} \log p_{\theta}(x_t \mid \mathbf{x}_{<t}) = \sum_{t=1}^{T} \log \frac{\exp\left(h_{\theta}(\mathbf{x}_{1:t-1})^{\top} e(x_t)\right)}{\sum_{x'} \exp\left(h_{\theta}(\mathbf{x}_{1:t-1})^{\top} e(x')\right)}$$

$$\mathcal{J}_{\text{XLNet}} = \log p(\text{New} \mid \text{is a city}) + \log p(\text{York} \mid \text{New}, \text{is a city})$$

Comparison of BERT and XLNet

- Permutation Language Model
- More Data (32.89B > 3.87B)
- Transformer-XL
  - Relative Positional Encoding
  - Segment Recurrence Mechanism

SQuAD1.1	EM	F1	SQuAD2.0	EM	F1
Dev set results without data augmentation					
BERT [10]	84.1	90.9	BERT† [10]	78.98	81.77
XLNet	**88.95**	**94.52**	XLNet	**86.12**	**88.79**
Test set results on leaderboard, with data augmentation (as of June 19, 2019)					
Human [27]	82.30	91.22	BERT+N-Gram+Self-Training [10]	85.15	87.72
ATB	86.94	92.64	SG-Net	85.23	87.93
BERT* [10]	87.43	93.16	BERT+DAE+AoA	85.88	88.62
XLNet	**89.90**	**95.08**	XLNet	**86.35**	**89.13**

Comparison in SQuAD

#	Model	RACE	SQuAD2.0 F1	SQuAD2.0 EM	MNLI m/mm	SST-2
1	BERT-Base	64.3	76.30	73.66	84.34/84.65	92.78
2	DAE + Transformer-XL	65.03	79.56	76.80	84.88/84.45	92.60
3	XLNet-Base ($K = 7$)	66.05	**81.33**	**78.46**	**85.84/85.43**	92.66
4	XLNet-Base ($K = 6$)	66.66	80.98	78.18	85.63/85.12	**93.35**
5	- memory	65.55	80.15	77.27	85.32/85.05	92.78
6	- span-based pred	65.95	80.61	77.91	85.49/85.02	93.12
7	- bidirectional data	66.34	80.65	77.87	85.31/84.99	92.66
8	+ next-sent pred	**66.76**	79.83	76.94	85.32/85.09	92.89

Ablation Study for Pure Model Comparison

A Fair Comparison Study of XLNet and BERT

(XLNet Team)

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)
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.

Experiment Results

- 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

- XLNet improves performance
- XLNet-Large could be better optimized

RoBERTa: A Robustly Optimized BERT Pretraining Approach

(Liu et al. CoRR abs/1907.11692)

- More data
- Bigger Batch
- Train Longer
- Remove Next Sentence Prediction
- Dynamically Change Mask Pattern

	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**	-
Ensembles on test (from leaderboard as of July 25, 2019)										
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**

RoBERTa in GLUE Test

RoBERTa: A Robustly Optimized BERT Pretraining Approach

(Liu et al. CoRR abs/1907.11692)

- Dynamically Change Mask Pattern

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

- Larger Batch Size

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

- Remove Next Sentence Prediction

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

- Larger Byte-Pair Encoding Vocabulary

  from 30K to 50K

RoBERTa: A Robustly Optimized BERT Pretraining Approach

(Liu et al. CoRR abs/1907.11692)

- Longer Training and Larger Trainset size

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

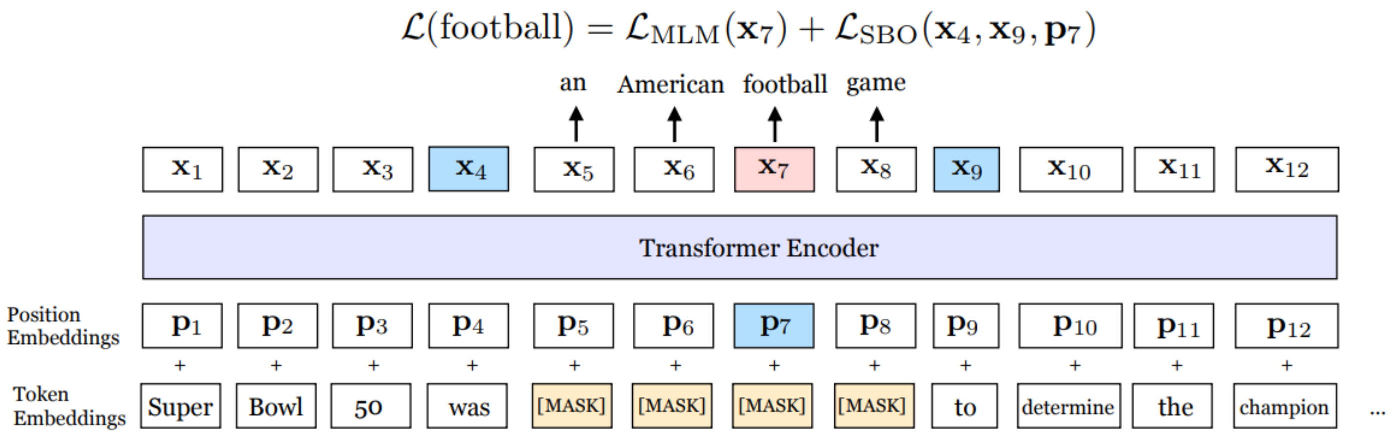
	MNLI	QNLI	QQP	RTE	SST	MRPC	CoLA	STS	WNLI	Avg
Single-task single models on dev										
$\text{BERT}_{\text{LARGE}}$	86.6/-	92.3	91.3	70.4	93.2	88.0	60.6	90.0	-	-
$\text{XLNet}_{\text{LARGE}}$	89.8/-	93.9	91.8	83.8	95.6	89.2	63.6	91.8	-	-
RoBERTa	**90.2/90.2**	**94.7**	**92.2**	**86.6**	**96.4**	**90.9**	**68.0**	**92.4**	**91.3**	-
Ensembles on test (from leaderboard as of July 25, 2019)										
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**

Language Models are Unsupervised Multitask Learners
GPT 2.0
(Radford et al. ICML 2019)

RoBERTa in GLUE Test

SpanBERT: Improving Pre-training by Representing and Predicting Spans

(Joshi et al. CoRR abs/1907.10529)



Model Architecture

	CoLA	SST-2	MRPC	STS-B
Google BERT	59.3	**95.2**	88.5/84.3	86.4/88.0
Our BERT	58.6	93.9	90.1/86.6	88.4/89.1
Our BERT-1seq	63.5	94.8	**91.2**/87.8	89.0/88.4
SpanBERT	**64.3**	94.8	90.9/**87.9**	**89.9/89.1**

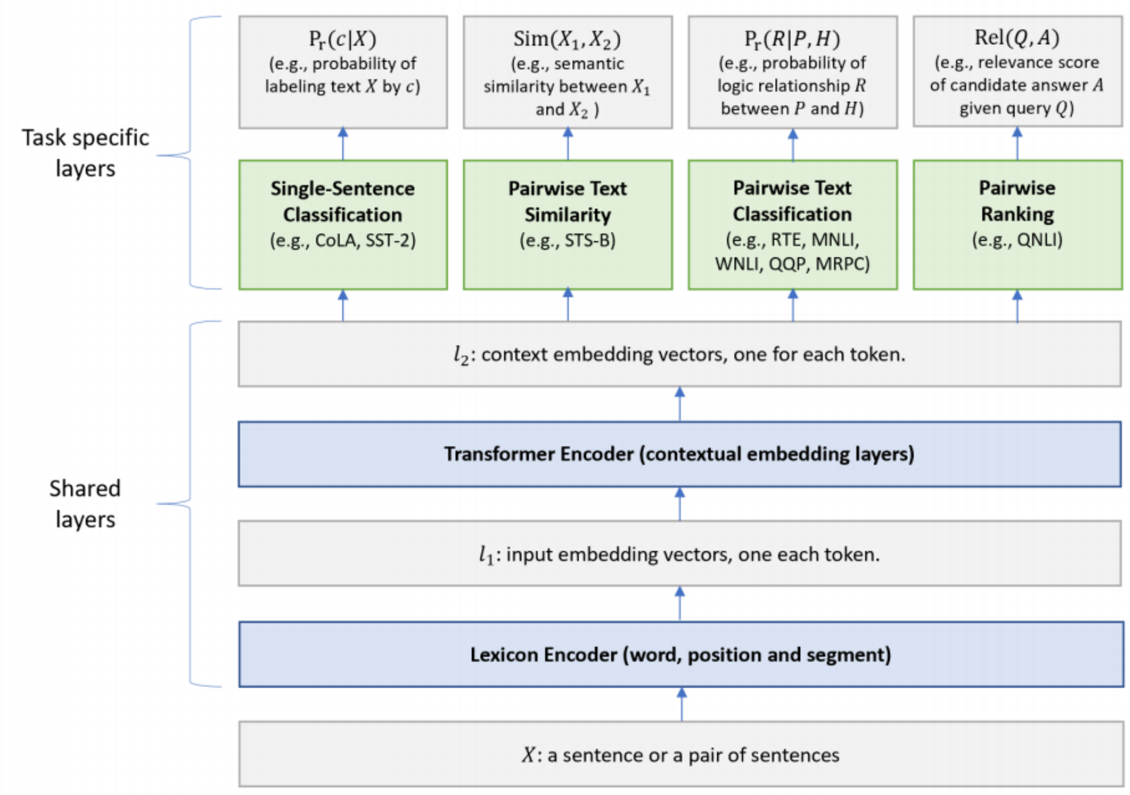
QQP	MNLI	QNLI	RTE	(Avg)
71.2/89.0	86.1/85.7	93.0	71.1	80.4
71.8/89.3	87.2/86.6	93.0	74.7	81.1
72.1/89.5	88.0/87.4	93.0	72.1	81.7
71.9/**89.5**	**88.1/87.7**	**94.3**	**79.0**	**82.8**

SpanBERT in GLUE Test

- Span Masking
- Span Boundary Objective
- Single-Sequence Training

Multi-Task Deep Neural Networks for Natural Language Understanding

(Liu et al. Microsoft Research. CoRR abs/1901.11504)



Model Architecture

Algorithm 1: Training a MT-DNN model.

Initialize model parameters Θ randomly.
Pre-train the shared layers (i.e., the lexicon encoder and the transformer encoder).
Set the max number of epoch: $epoch_{max}$.
//Prepare the data for T tasks.
for t *in* $1, 2, ..., T$ **do**
  Pack the dataset t into mini-batch: D_t.
end
for $epoch$ *in* $1, 2, ..., epoch_{max}$ **do**
  1. Merge all the datasets:
  $D = D_1 \cup D_2 ... \cup D_T$
  2. Shuffle D
  for b_t *in* D **do**
    //b_t is a mini-batch of task t.
    3. Compute loss : $L(\Theta)$
    $L(\Theta) =$ Eq. 6 for classification
    $L(\Theta) =$ Eq. 7 for regression
    $L(\Theta) =$ Eq. 8 for ranking
    4. Compute gradient: $\nabla(\Theta)$
    5. Update model: $\Theta = \Theta - \epsilon\nabla(\Theta)$
  end
end

$$-\sum_c \mathbb{1}(X,c)\log(P_r(c|X))$$

$$(y - \text{Sim}(X_1, X_2))^2$$

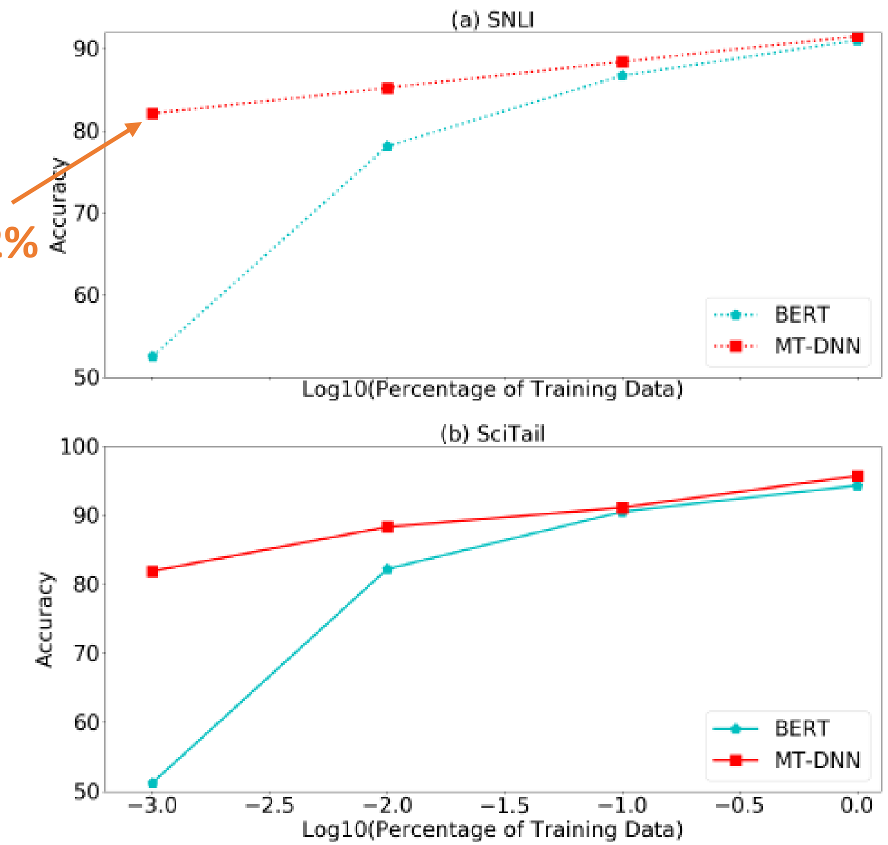
$$-\sum_{(Q,A^+)} P_r(A^+|Q)$$

Training Algorithm

Multi-Task Deep Neural Networks for Natural Language Understanding

(Liu et al. Microsoft Research. CoRR abs/1901.11504)

23, 82%



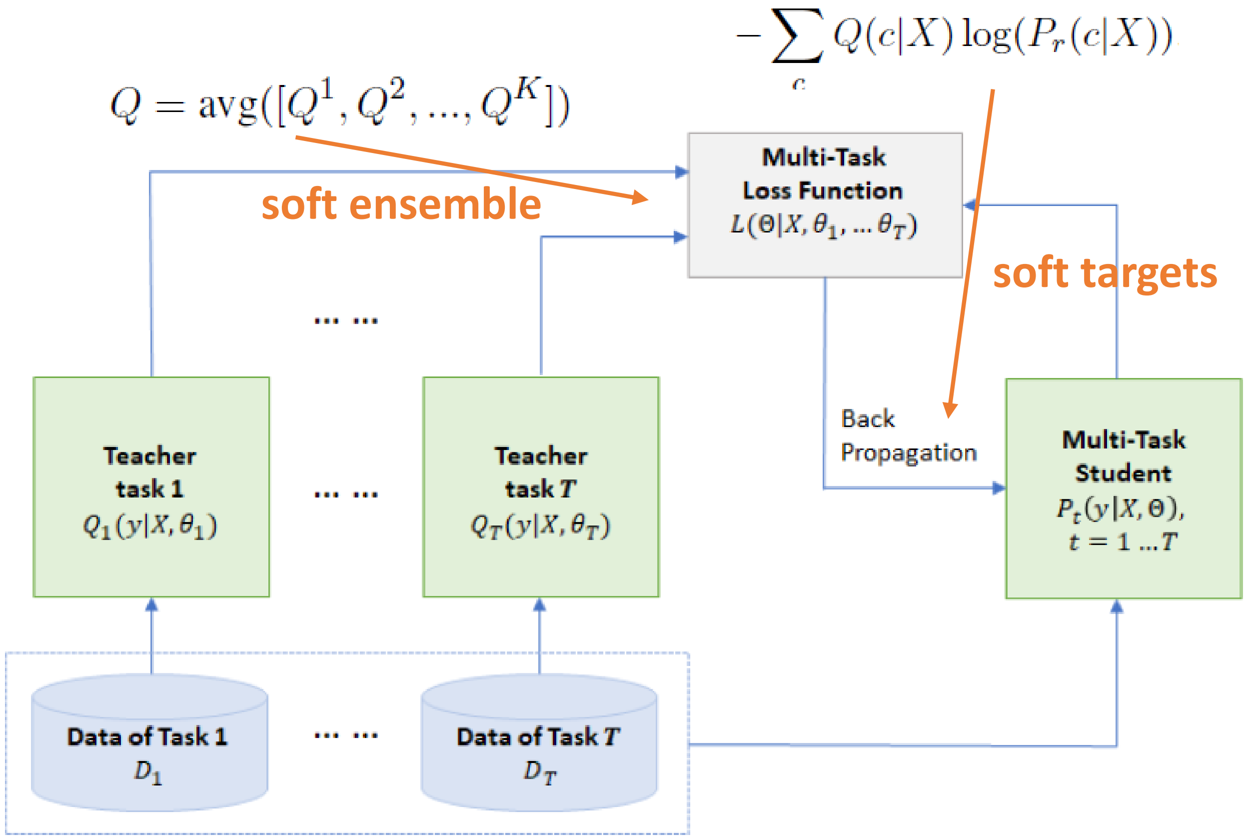
Domain Adaptation

Model	CoLA 8.5k	SST-2 67k	MRPC 3.7k	STS-B 7k	QQP 364k	MNLI-m/mm 393k	QNLI 108k	RTE 2.5k	WNLI 634	AX	**Score**
BiLSTM+ELMo+Attn [1]	36.0	90.4	84.9/77.9	75.1/73.3	64.8/84.7	76.4/76.1	-	56.8	65.1	26.5	70.5
Singletask Pretrain Transformer [2]	45.4	91.3	82.3/75.7	82.0/80.0	70.3/88.5	82.1/81.4	-	56.0	53.4	29.8	72.8
GPT on STILTs [3]	47.2	93.1	87.7/83.7	85.3/84.8	70.1/88.1	80.8/80.6	-	69.1	65.1	29.4	76.9
$\text{BERT}^{4}_{\text{LARGE}}$	60.5	94.9	89.3/85.4	87.6/86.5	72.1/89.3	86.7/85.9	92.7	70.1	65.1	39.6	80.5
$\text{MT-DNN}_{\text{no-fine-tune}}$	58.9	94.6	**90.1/86.4**	89.5/88.8	**72.7/89.6**	86.5/85.8	**93.1**	79.1	65.1	39.4	81.7
MT-DNN	**62.5**	**95.6**	**91.1/88.2**	**89.5/88.8**	**72.7/89.6**	**86.7/86.0**	**93.1**	**81.4**	65.1	**40.3**	**82.7**
Human Performance	66.4	97.8	86.3/80.8	92.7/92.6	59.5/80.4	92.0/92.8	91.2	93.6	95.9	-	87.1

MT-DNN in GLUE Test

Improving Multi-Task Deep Neural Networks via Knowledge Distillation for Natural Language Understanding

(Liu et al. Microsoft Research. CoRR abs/1904.09482)



Process of Knowledge Distillation

Model	CoLA 8.5k	SST-2 67k	MRPC 3.7k	STS-B 7k	QQP 364k	MNLI-m/mm 393k	QNLI 108k	RTE 2.5k	WNLI 634	AX	**Score**
BiLSTM+ELMo+Attn [1]	36.0	90.4	84.9/77.9	75.1/73.3	64.8/84.7	76.4/76.1	79.8	56.8	65.1	26.5	70.0
Singletask Pretrain Transformer [2]	45.4	91.3	82.3/75.7	82.0/80.0	70.3/88.5	82.1/81.4	87.4	56.0	53.4	29.8	72.8
GPT on STILTs [3]	47.2	93.1	87.7/83.7	85.3/84.8	70.1/88.1	80.8/80.6	-	69.1	65.1	29.4	76.9
$\mathrm{BERT}_{\mathrm{LARGE}}$ [4]	60.5	94.9	89.3/85.4	87.6/86.5	72.1/89.3	86.7/85.9	92.7	70.1	65.1	39.6	80.5
MT-DNN[5]	61.5	95.6	90.0/86.7	88.3/87.7	72.4/89.6	86.7/86.0	-	75.5	65.1	40.3	82.2
Snorkel MeTaL [6]	63.8	**96.2**	91.5/88.5	**90.1/89.7**	73.1/89.9	87.6/87.2	93.9	80.9	65.1	39.9	83.2
ALICE [*]	63.5	95.2	**91.8/89.0**	89.8/88.8	**74.0/90.4**	**87.9/87.4**	95.7	80.9	65.1	40.7	83.3
$\mathrm{MT\text{-}DNN}_{\mathrm{KD}}$	**65.4**	95.6	91.1/88.2	89.6/89.0	72.7/89.6	87.5/86.7	**96.0**	**85.1**	65.1	**42.8**	**83.7**
Human Performance	66.4	97.8	86.3/80.8	92.7/92.6	59.5/80.4	92.0/92.8	91.2	93.6	95.9	-	87.1

MT-DNN_{KD} in GLUE Test

Recent In-depth Analyses of BERT-like Models in NLP Tasks

Probing Neural Network Comprehension of Natural Language Arguments

(Niven et al. ACL 2019)

Topic: Tax Break for Sports.

Additional Information: Should pro sports leagues enjoy nonprofit status?

Premise (Reason): Government is already struggling to pay for basic needs.

And since

- ✔ **Warrant 0:** the government isn't required to pay for all the country's needs
- ✗ **Warrant 1:** the government is required to pay for the country's needs

Claim: Sport leagues should not enjoy nonprofit.

ARCT: Argument Reasoning Comprehension Task

(Habernal et al. NACCL 2018)

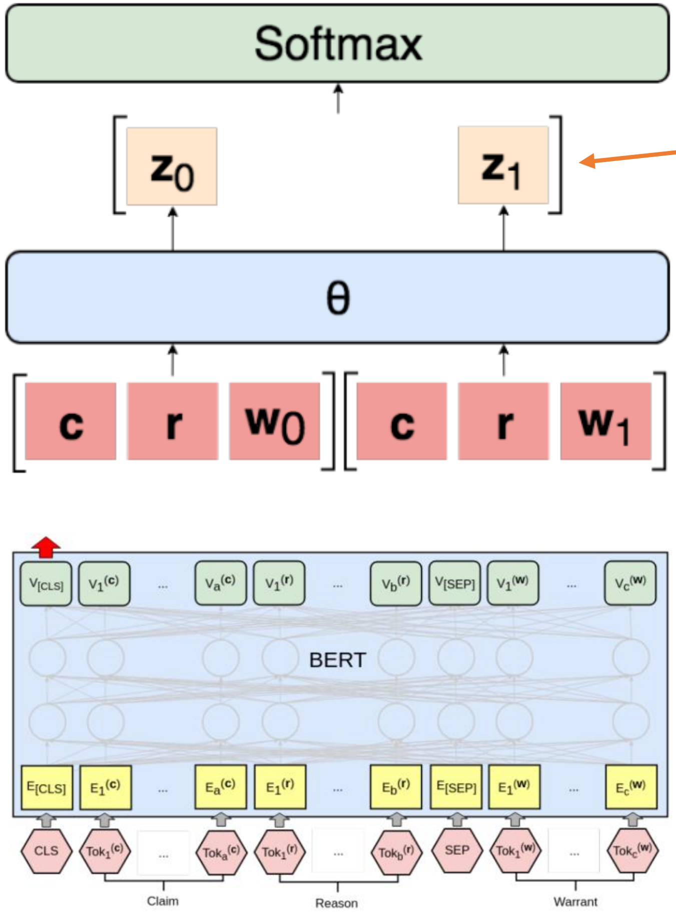
Rank	System	Accuracy
1	GIST	0.712
2	blcu_nlp	0.606
3	ECNU	0.604
4	NLITrans	0.590
5	Joker*	0.586
6	YNU_Deep	0.583
7	mingyan	0.581
8	ArcNet	0.577
8	UniMelb	0.577
10	TRANSRW	0.570
11	lyb3b	0.568
12	SNU_IDS	0.565
13	ArgEns-GRU	0.556
14	ITNLP-ARC	0.552
15	YNU-HPCC	0.550
16	TakeLab	0.541
17	HHU	0.534
18	**Random baseline**	0.527
19	Deepfinder	0.525
20	ART	0.518
21	RW2C	0.500
22	ztangfdu	0.464

SemEval-2018 Task 12: The Argument Reasoning Comprehension Task

(Habernal et al. SemEval-2018)

Probing Neural Network Comprehension of Natural Language Arguments

(Niven et al. ACL 2019)



Model Architecture

$$z_j^{(i)} = \boldsymbol{\theta}[\boldsymbol{c}^{(i)}; \boldsymbol{r}^{(i)}; \boldsymbol{w}_j^{(i)}]$$

	Dev	Test		
	Mean	Mean	Median	Max
Human (trained)		0.909 ± 0.11		
Human (untrained)		0.798 ± 0.16		
BERT (Large)	0.701 ± 0.05	0.671 ± 0.09	**0.712**	**0.770**
GIST (Choi and Lee, 2018)	**0.716** ± 0.01	**0.711** ± 0.01		
BERT (Base)	0.680 ± 0.02	0.623 ± 0.07	0.651	0.685
World Knowledge (Botschen et al., 2018)	0.674 ± 0.01	0.568 ± 0.03		0.610
BoV	0.639 ± 0.02	0.564 ± 0.02	0.569	0.595
BiLSTM	0.658 ± 0.01	0.552 ± 0.02	0.552	0.592

Experiment Result

Probing Neural Network Comprehension of Natural Language Arguments

(Niven et al. ACL 2019)

A Cue's Applicability: $\alpha_k = \sum_{i=1}^{n} \mathbb{1}\left[\exists j, k \in \mathbb{T}_j^{(i)} \wedge k \notin \mathbb{T}_{\neg j}^{(i)}\right]$

A Cue's Productivity: $\pi_k = \frac{\sum_{i=1}^{n} \mathbb{1}\left[\exists j, k \in \mathbb{T}_j^{(i)} \wedge k \notin \mathbb{T}_{\neg j}^{(i)} \wedge y_i = j\right]}{\alpha_k}$

A Cue's Coverage: $\xi_k = \alpha_k / n$

	Productivity	Coverage
Train	0.65	0.66
Validation	0.62	0.44
Test	0.52	0.77
All	**0.61**	**0.64**

The Cue "not" in Warrant

BERT(R,C) = 0.5

	Test		
	Mean	**Median**	**Max**
BERT	**0.671** $\pm$ 0.09	**0.712**	**0.770**
BERT (W)	0.656 $\pm$ 0.05	0.675	0.712
BERT (R, W)	0.600 $\pm$ 0.10	0.574	0.750
BERT (C, W)	0.532 $\pm$ 0.09	0.503	0.732
BoV	0.564 $\pm$ 0.02	0.569	0.595
BoV (W)	0.567 $\pm$ 0.02	0.572	0.606
BoV (R, W)	0.554 $\pm$ 0.02	0.557	0.579
BoV (C, W)	0.545 $\pm$ 0.02	0.544	0.589
BiLSTM	0.552 $\pm$ 0.02	0.552	0.592
BiLSTM (W)	0.550 $\pm$ 0.02	0.547	0.577
BiLSTM (R, W)	0.547 $\pm$ 0.02	0.551	0.577
BiLSTM (C, W)	0.552 $\pm$ 0.02	0.550	0.601

Probing Experiments

Probing Neural Network Comprehension of Natural Language Arguments

(Niven et al. ACL 2019)

	Original	**Adversarial**
Claim	Google is not a harmful monopoly	Google is a harmful monopoly
Reason	People can choose not to use Google	People can choose not to use Google
Warrant	Other search engines do not redirect to Google	All other search engines redirect to Google
Alternative	All other search engines redirect to Google	Other search engines do not redirect to Google

Adversarial Transfer

	Test		
	Mean	**Median**	**Max**
BERT	**0.504** $\pm$ 0.01	**0.505**	**0.533**
BERT (W)	0.501 $\pm$ 0.00	0.501	0.502
BERT (R, W)	0.500 $\pm$ 0.00	0.500	0.502
BERT (C, W)	0.501 $\pm$ 0.01	0.500	0.518

Adversarial Result

"with little to no understanding about the reality underlying these arguments, good performance shouldn't be feasible."

Probing Neural Network Comprehension of Natural Language Arguments

(Niven et al. ACL 2019)

Some Discussions:

- Adversarial Attack in Computer Vision
- Diverge or not ?
- What about other models like XLNet ?
- What about SOTA in ARCT, i.e. GIST ?

timniven commented 19 hours ago Member

Hi LFhase,

We haven't tested XLNet. A broader question though is why BERT can't solve this task, and whether XLNet is likely to have whatever BERT lacks? I think it is important to develop an intuition about this. Of course, you are welcome to conduct this experiment (and let me know the results!) since it actually doesn't cost very much to just try. But since my intuition is that XLNet is very unlikely to have the world knowledge needed for the task, I do not expect it to work, and therefore don't plan to conduct the experiment myself. However, I welcome you to prove me wrong.

The degenerate runs on small training sets are discussed in the original paper (I don't think you would call it "divergence," but rather a lack of good convergence - a "degenerate run" is what the original authors call it). In our case it is actually a rather subjective judgment. Looking at the training accuracies of BERT's runs, you can generally see that when BERT doesn't get over 80% on the training set it performs poorly on the validation and or test sets. I'm not 100% sure why this happens, it could be that with such a small dataset there are more local minima to get stuck in during optimization. Again, if you can develop your own intuition about this kind of question, then hopefully you can design an experiment to test your hypothesis. But we do not suggest using the original dataset anymore because of the bias coming from uneven distributions of linguistic artifacts over the labels. Since all models love to exploit these statistics, this is a meaningless exercise. What we have called the "adversarial" dataset (which may be not have been the best choice of words) is what you should use for any future work on ARCT.

Good luck with your studies and best wishes to you :)

Tim.

Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference

(McCoy et al. CoRR abs/1902.01007)

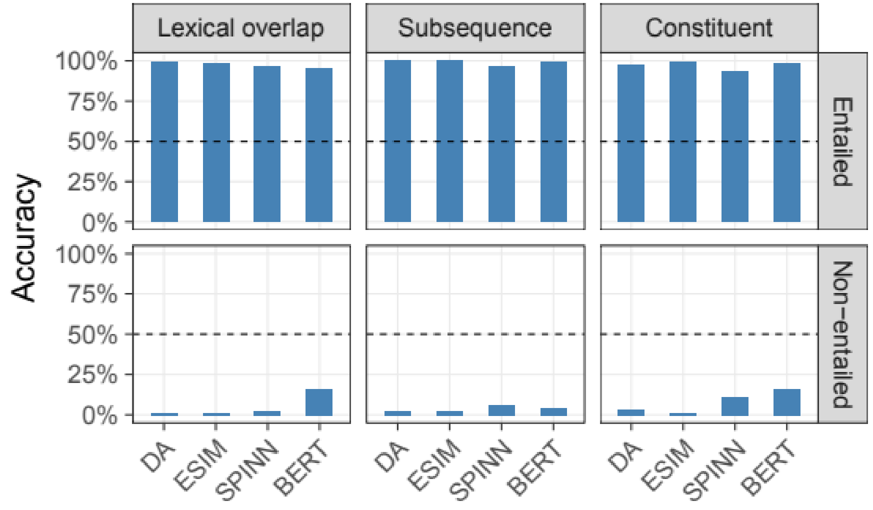
Heuristic	Definition	Example
Lexical overlap	Assume that a premise entails all hypotheses constructed from words in the premise	**The doctor** was **paid** by **the actor**. $\xrightarrow[\text{WRONG}]{}$ The doctor paid the actor.
Subsequence	Assume that a premise entails all of its contiguous subsequences.	The doctor near **the actor danced**. $\xrightarrow[\text{WRONG}]{}$ The actor danced.
Constituent	Assume that a premise entails all complete subtrees in its parse tree.	If **the artist slept**, the actor ran. $\xrightarrow[\text{WRONG}]{}$ The artist slept.

Heuristics

Heuristic	Premise	Hypothesis	Label
Lexical overlap heuristic	The banker near the judge saw the actor.	The banker saw the actor.	E
	The lawyer was advised by the actor.	The actor advised the lawyer.	E
	The doctors visited the lawyer.	The lawyer visited the doctors.	N
	The judge by the actor stopped the banker.	The banker stopped the actor.	N
Subsequence heuristic	The artist and the student called the judge.	The student called the judge.	E
	Angry tourists helped the lawyer.	Tourists helped the lawyer.	E
	The judges heard the actors resigned.	The judges heard the actors.	N
	The senator near the lawyer danced.	The lawyer danced.	N
Constituent heuristic	Before the actor slept, the senator ran.	The actor slept.	E
	The lawyer knew that the judges shouted.	The judges shouted.	E
	If the actor slept, the judge saw the artist.	The actor slept.	N
	The lawyers resigned, or the artist slept.	The artist slept.	N

Heuristic	Supporting Cases	Contradicting Cases
Lexical overlap	2,158	261
Subsequence	1,274	72
Constituent	1,004	58

Original Heuristic Distribution



Original Experiment Result

Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference

(McCoy et al. CoRR abs/1902.01007)

Some Analysis:

- Trainset too difficult?

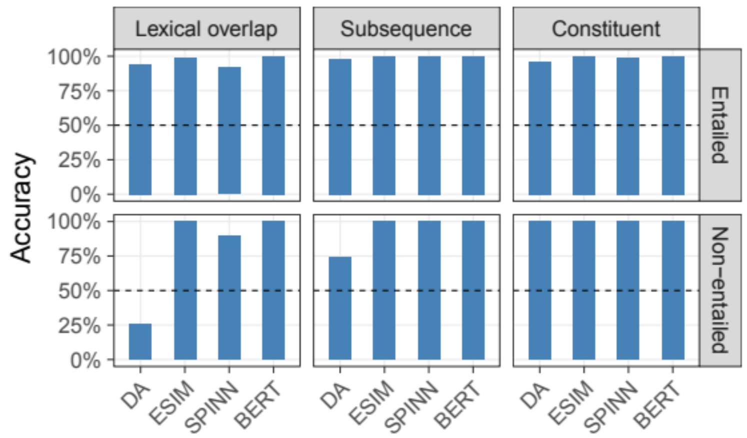
  No. Human 77% / 75%

- Lack of representation capabilities ?

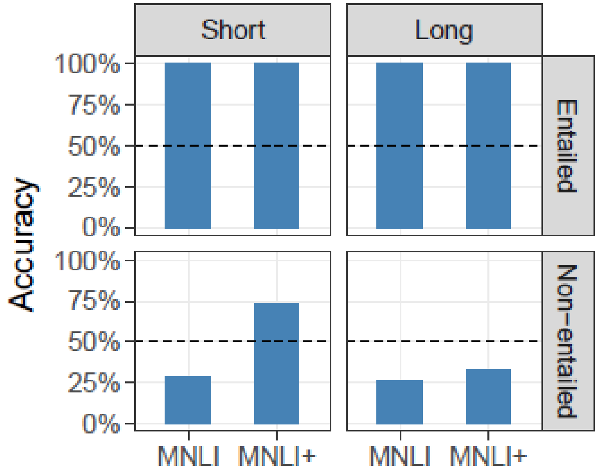
  No. RNNs implicitly implement

  tensor-product representations

  (McCoy et al. ICLR 2019)



HANS Result with Augmented Dataset



comp_same_long and comp_same_long

(Dasgupta et al. ACL 2018)

Result with Augmented Dataset

Conclusions

- BERTs are powerful because
  - It provides a novel way to pretrain representation models
  - It substantially push SOTA to a new level
- BERTs can be better with
  - More careful optimizing
  - Designing good pretraining tasks and objectives
  - More robust dataset
- BERTs don’t solve NLP because
  - Tasks like ARCT need more advanced high-level representation ability

Discussions/ Future Directions

- Two-Stage Pre-trained Models
  - What’ the best recipe: Multi-Task? Fine-tune in Downstream Task?
  - Should we embrace more robust training & data?
- Adversarial Attack in NLP
  - Adversarial attack in NLP like CV?
- Dataset Construction / Evaluation
  - Is the dataset robust to Model Exploitation?
  - How to evaluate such ability?