BERTology

Jie Tang

Department of Computer Science and Technology

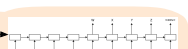
Tsinghua University

May 20, 2020

Overview

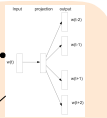
seq2seq (2014)

Quoc V. Le
Google Brain

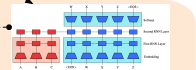


Semi-supervised Sequence Learning (2015)

Tomas Mikolov
Google->FB



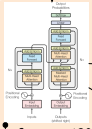
word2vec (2013)



pretrained seq2seq (2017)

context2vec (2016)

Ido Dagan
Bar-Ilan University



Transformer (2017)

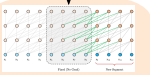
Google Research
Google Brain

OpenAI GPT (2018)

ELMo (2018)

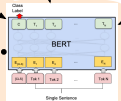
GPT-2 (2019)

Allen AI



Transformer-XL (2019)

Jacob Devlin
Google

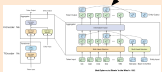


BERT (2018)

Ilya Sutskever
OpenAI

Zhilin Yang
Recurrent AI

XLNet (2019)



ERNIE Tsinghua (2019)

ERNIE Baidu (2019)

RoBERTa (2019)
SpanBERT (2019)
ALBERT (2019)
ELECTRA (2020)

By Yukuo @KEG, THU

Overview

Background

- Language model: given an sequence of length n, it assigns a probability $p\left(x_1, x_2, x_3, \ldots, x_n\right)$ to the whole sequence. The probability of standard LM can be decomposed as $p\left(x_1, x_2, \ldots, x_n\right) = \prod_{t=1}^{n} p\left(x_t | x_1, \ldots, x_{t-1}\right)$.
- Sequence to sequence (seq2seq) learning: given an input sequence $x_1, x_2, ..., x_m$ and an output sequence $y_1, y_2, ..., y_n$, the objective of seq2seq learning is to maximize the likelihood $p(y_n, y_{n-1}, ..., y_1 | x_1, x_2, ..., x_m)$. Common seq2seq methods decompose this objective as $p\left(y_n, y_{n-1}, \ldots, y_1 | x_1, x_2, \ldots, x_m\right) = \prod_{t=1}^{n} p\left(y_t | y_{t-1}, \ldots, y_1; x_1, x_2, \ldots, x_m\right)$.

Pre-BERT Era

- Semi-supervised Sequence Learning
- context2vec: Learning Generic Context Embedding with Bidirectional LSTM
- Pre-trained seq2seq: Unsupervised Pretraining for Sequence to Sequence Learning
- ELMo: Deep contextualized word representations
- OpenAI GPT: Improving Language Understanding by Generative Pre-Training

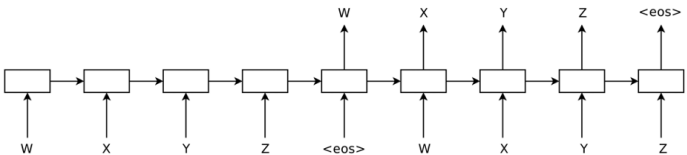
Semi-supervised Sequence Learning[1]

- This paper presents two approaches that use unlabeled data to improve sequence learning with recurrent networks.
- One is to predict what comes next in a sequence, which is a conventional language model.
- The other is to use a sequence autoencoder, which reads the input sequence into a vector and predicts the input sequence again.
- These two methods can be used as a "pretraining" step for a later supervised sequence learning algorithm.

[1] Dai, Andrew M., and Quoc V. Le. "Semi-supervised sequence learning." Advances in neural information processing systems. 2015.

Semi-supervised Sequence Learning (cont.)

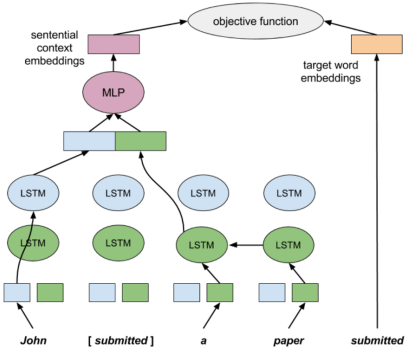
- The sequence autoencoder is inspired by seq2seq, except that it is an unsupervised learning model. The objective is to reconstruct the input sequence itself.



- The weights obtained from the sequence autoencoder can be used as an initialization of downstream LSTM networks.

context2vec[1]

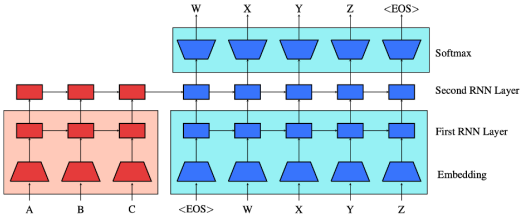
- context2vec is an unsupervised model for efficiently learning a generic context embedding function from large corpora, using bidirectional LSTM. The architecture is based on word2vec's CBOW but replaces its context modeling with LSTM.



[1]Melamud, Oren, Jacob Goldberger, and Ido Dagan. "context2vec: Learning generic context embedding with bidirectional lstm." CoNLL 2016.

Pre-trained Seq2seq[1]

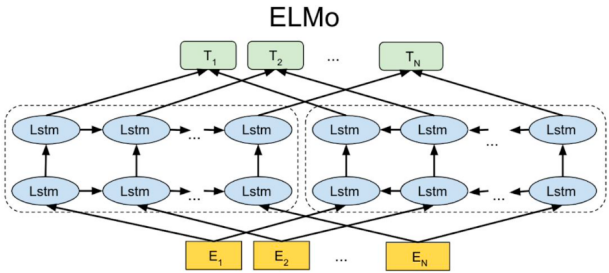
- This work presents a general unsupervised learning method to improve the accuracy of seq2seq models.
- The weights of a seq2seq model are initialized with the pretrained weights of two language models and then fine-tuned with labeled data.
- All parameters in a shaded box are pretrained from language models.



[1]Ramachandran, Prajit, Peter J. Liu, and Quoc V. Le. "Unsupervised pretraining for sequence to sequence learning." arXiv preprint arXiv:1611.02683 (2016).

ELMo[1]

- ELMo introduces a new type of deep contextualized word presentation, which are functions of the internal states of a deep bidirectional language model (biLM).
- ELMo uses the concatenation of independently trained left-to-right and right-to-left LSTMs to generate features for downstream tasks.



[1]Peters, Matthew E., et al. "Deep contextualized word representations." arXiv preprint arXiv:1802.05365 (2018).

ELMo (cont.)

- ELMo formulation jointly maximizes the log likelihood of the forward and backward directions:

$$\sum_{k=1}^{N} \left(\log p \left(t_k | t_1, \ldots, t_{k-1}; \Theta_x, \vec{\Theta}_{LSTM}, \Theta_s \right) + \log p \left(t_k | t_{k+1}, \ldots, t_N; \Theta_x, \overleftarrow{\Theta}_{LSTM}, \Theta_s \right) \right) \quad (1)$$

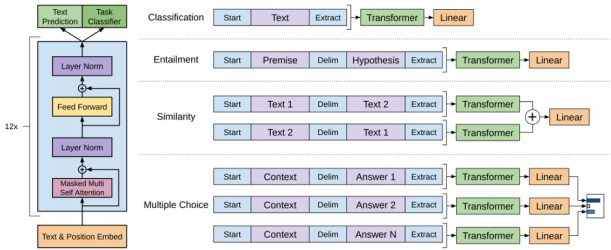
where Θ_x and Θ_s are the parameters of token representation and softmax layer.

- ELMo Results

Task	Previous SOTA		Our baseline	ELMo + baseline	Increase (absolute/ relative)
SQuAD	Liu et al. (2017)	84.4	81.1	85.8	4.7 / 24.9%
SNLI	Chen et al. (2017)	88.6	88.0	88.7 ± 0.17	0.7 / 5.8%
SRL	He et al. (2017)	81.7	81.4	84.6	3.2 / 17.2%
Coref	Lee et al. (2017)	67.2	67.2	70.4	3.2 / 9.8%
NER	Peters et al. (2017)	91.93 ± 0.19	90.15	92.22 ± 0.10	2.06 / 21%
SST-5	McCann et al. (2017)	53.7	51.4	54.7 ± 0.5	3.3 / 6.8%

OpenAI GPT[1]

- OpenAI GPT uses generative pre-training of a language model on a diverse corpus of unlabeled text, followed by discriminative fine-tuning on each specific task.
- OpenAI GPT uses a left-to-right Transformer.



[1]Radford, Alec, et al. "Improving language understanding by generative pre-training." URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf (2018).

OpenAI (cont.)

- GPT uses a standard language modeling for pre-training

$$\sum_i \log P\left(u_i | u_{i-k}, \ldots, u_{i-1}; \Theta\right) \tag{2}$$

where P is modeled using a multi-layer Transformer

$$\begin{aligned} h_0 &= UW_e + W_p \\ h_l &= \text{transformer\_block}\left(h_{l-1}\right) \forall i \in [1, n] \\ P(u) &= \text{softmax}\left(h_n W_e^T\right) \end{aligned} \tag{3}$$

where W_e is the token embedding and W_p is the position embedding.

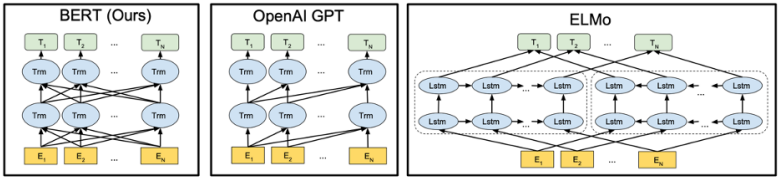
- Results on GLUE

Method	Classification		Semantic Similarity			GLUE
	CoLA (mc)	SST2 (acc)	MRPC (F1)	STSB (pc)	QQP (F1)	
Sparse byte mLSTM [16]	-	**93.2**	-	-	-	-
TF-KLD [23]	-	-	**86.0**	-	-	-
ECNU (mixed ensemble) [60]	-	-	-	81.0	-	-
Single-task BiLSTM + ELMo + Attn [64]	35.0	90.2	80.2	55.5	66.1	64.8
Multi-task BiLSTM + ELMo + Attn [64]	18.9	91.6	83.5	72.8	63.3	68.9
Finetuned Transformer LM (ours)	**45.4**	91.3	82.3	**82.0**	**70.3**	**72.8**

Overview

BERT[1]

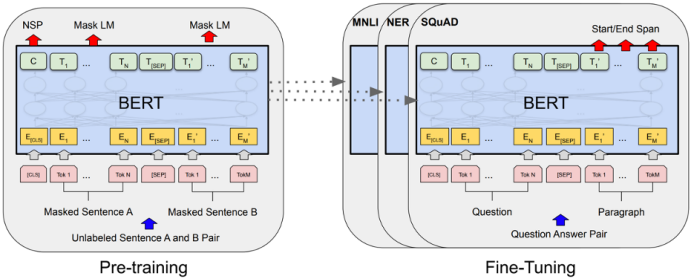
- BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers.
- The pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks. range of tasks



[1]Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

BERT (cont.)

- Overall pre-training and fine-tuning procedures for BERT.
- BERT uses two unsupervised tasks: masked language model (MLM) and next sentence prediction (NSP).



BERT Results

- Results on GLUE

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average**
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**

- Results on SQuAD

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**

(a) SQuAD 1.1

System	Dev EM	Dev F1	Test EM	Test F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	86.3	89.0	86.9	89.5
#1 Single - MIR-MRC (F-Net)	-	-	74.8	78.0
#2 Single - nlnet	-	-	74.2	77.1
Published				
unet (Ensemble)	-	-	71.4	74.9
SLQA+ (Single)	-		71.4	74.4
Ours				
$\text{BERT}_{\text{LARGE}}$ (Single)	78.7	81.9	80.0	83.1

(b) SQuAD 2.0

Overview

Post-BERT Era

- RoBERTa (2019)
- XLNet (2019)
- ERNIE (Tsinghua) (2019)
- ERINE (Baidu) (2019)
- ALBERT (2019)
- ELECTRA (2020)

RoBERTa[1]

- RoBERTa: A Robustly Optimized BERT Pretraining Approach
- RoBERTa finds that BERT is under-trained due to hyperparameters and training set.
- BERT can be improved by:
  - longer inputs (only full length sentence)
  - larger batch size and learning rate
  - larger dataset
  - No next sentence prediction
  - Dynamic masking

[1] Liu, Yinhan, et al. "Roberta: A robustly optimized bert pretraining approach." arXiv preprint arXiv:1907.11692 (2019).

RoBERTa Results

- Results on SQuAD

Model	SQuAD 1.1 EM	SQuAD 1.1 F1	SQuAD 2.0 EM	SQuAD 2.0 F1
Single models on dev, w/o data augmentation				
$\text{BERT}_{\text{LARGE}}$	84.1	90.9	79.0	81.8
$\text{XLNet}_{\text{LARGE}}$	**89.0**	94.5	86.1	88.8
RoBERTa	88.9	**94.6**	**86.5**	**89.4**
Single models on test (as of July 25, 2019)				
$\text{XLNet}_{\text{LARGE}}$			$86.3^{\dagger}$	$89.1^{\dagger}$
RoBERTa			86.8	89.8
XLNet + SG-Net Verifier			$\mathbf{87.0}^{\dagger}$	$\mathbf{89.9}^{\dagger}$

- Results on GLUE

	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**

XLNet[1]

- We first review and compare the conventional auto-regressive language modeling and BERT for language pretraining.
- The objective of auto-regressive language modeling:

$$\max_{\theta} \log p_\theta(\mathbf{x}) = \sum_{t=1}^{T} \log p_\theta\left(x_t | \mathbf{x}_{<t}\right) = \sum_{t=1}^{T} \log \frac{\exp\left(h_\theta\left(\mathbf{x}_{1:t-1}\right)^\top e\left(x_t\right)\right)}{\sum_{x'} \exp\left(h_\theta\left(\mathbf{x}_{1:t-1}\right)^\top e\left(x'\right)\right)} \tag{4}$$

- The objective of BERT is to reconstruct masked tokens $\overline{\mathbf{x}}$ from a corrupted version $\hat{\mathbf{x}}$:

$$\max_{\theta} \log p_\theta(\overline{\mathbf{x}}|\hat{\mathbf{x}}) \approx \sum_{t=1}^{T} m_t \log p_\theta\left(x_t|\hat{\mathbf{x}}\right) = \sum_{t=1}^{T} m_t \log \frac{\exp\left(H_\theta(\hat{\mathbf{x}})_t^\top e\left(x_t\right)\right)}{\sum_{x'} \exp\left(H_\theta(\hat{\mathbf{x}})_t^\top e\left(x'\right)\right)} \tag{5}$$

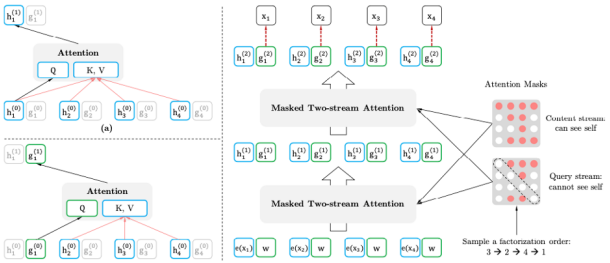
[1]Yang, Zhilin, et al. "Xlnet: Generalized autoregressive pretraining for language understanding." Advances in neural information processing systems. 2019.

XLNet (cont.)

- Instead of using a fixed forward or backward factorization order as in conventional Auto-Regressive models, XLNet maximizes the expected log likelihood of a sequence w.r.t. all possible permutations of the factorization order.

$$\max_{\theta} \quad \mathbb{E}_{\mathbf{z}\sim\mathcal{Z}_T}\left[\sum_{t=1}^{T}\log p_\theta\left(x_{z_t}|\mathbf{x}_{\mathbf{z}<t}\right)\right] \tag{6}$$

where $\mathcal{Z}_T$ is the set of all possible permutations of the length-T index sequence.



XLNet Results

- Results on SQuAD

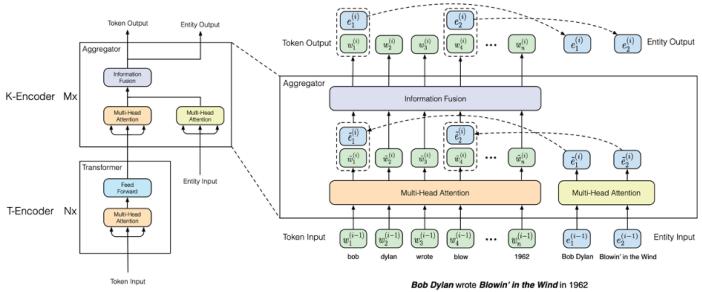
SQuAD2.0	EM	F1	SQuAD1.1	EM	F1
Dev set results (single model)					
BERT [10]	78.98	81.77	BERT† [10]	84.1	90.9
RoBERTa [21]	86.5	89.4	RoBERTa [21]	88.9	94.6
XLNet	**87.9**	**90.6**	XLNet	**89.7**	**95.1**
Test set results on leaderboard (single model, as of Dec 14, 2019)					
BERT [10]	80.005	83.061	BERT [10]	85.083	91.835
RoBERTa [21]	86.820	89.795	BERT* [10]	87.433	93.294
XLNet	**87.926**	**90.689**	XLNet	**89.898**‡	**95.080**‡

- Results on GLUE

Model	MNLI	QNLI	QQP	RTE	SST-2	MRPC	CoLA	STS-B	WNLI
Single-task single models on dev									
BERT [2]	86.6/-	92.3	91.3	70.4	93.2	88.0	60.6	90.0	-
RoBERTa [21]	90.2/90.2	94.7	92.2	**86.6**	96.4	**90.9**	68.0	92.4	-
XLNet	**90.8/90.8**	**94.9**	**92.3**	85.9	**97.0**	90.8	**69.0**	**92.5**	-
Multi-task ensembles on test (from leaderboard as of Oct 28, 2019)									
MT-DNN* [20]	87.9/87.4	96.0	89.9	86.3	96.5	92.7	68.4	91.1	89.0
RoBERTa* [21]	90.8/90.2	98.9	90.2	88.2	96.7	92.3	67.8	92.2	89.0
XLNet*	**90.9/90.9**†	**99.0**†	**90.4**†	**88.5**	**97.1**†	**92.9**	**70.2**	**93.0**	**92.5**

ERNIE (Tsinghua)[1]

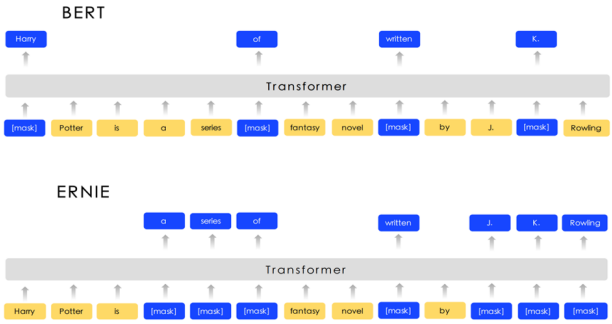
- ERNIE: Enhanced Language Representation with Informative Entities
- ERNIE utilizes both large-scale textual corpora and knowledge graphs to train an enhanced language representation model, which can take full advantage of lexical, syntactic, and knowledge information simultaneously.



[1]Zhang, Zhengyan, et al. "ERNIE: Enhanced language representation with informative entities." arXiv preprint arXiv:1905.07129 (2019).

ERNIE (Baidu)[1]

- ERNIE: Enhanced Representation through Knowledge Integration
- ERNIE is designed to learn language representation enhanced by knowledge masking strategies, which includes entity-level masking and phrase-level masking.



[1]Sun, Yu, et al. "Ernie: Enhanced representation through knowledge integration." arXiv preprint arXiv:1904.09223 (2019).

ALBERT[1]

- ALBERT uses two parameter-reduction techniques to lower memory consumption and increase the training speed of BERT.
- Factorize embedding parameterization: reduce the embedding parameters from $O(V \times H)$ to $O(V \times E + E \times H)$ where $H \gg E$.
- Cross-layer parameters sharing: a default decision for ALBERT is to share all parameters across layers.
- Sentence-order prediction replaces NSP

Model		Parameters	Layers	Hidden	Embedding	Parameter-sharing
BERT	base	108M	12	768	768	False
	large	334M	24	1024	1024	False
ALBERT	base	12M	12	768	128	True
	large	18M	24	1024	128	True
	xlarge	60M	24	2048	128	True
	xxlarge	235M	12	4096	128	True

[1]Lan, Zhenzhong, et al. "Albert: A lite bert for self-supervised learning of language representations." arXiv preprint arXiv:1909.11942 (2019).

ALBERT Results

- Results on GLUE

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**

- Results on SQuAD and RACE

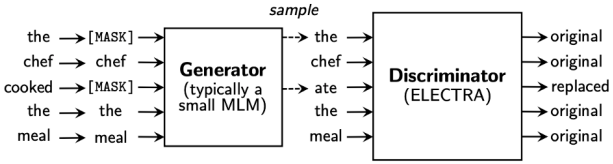
Models	SQuAD1.1 dev	SQuAD2.0 dev	SQuAD2.0 test	RACE test (Middle/High)
Single model (from leaderboard as of Sept. 23, 2019)				
BERT-large	90.9/84.1	81.8/79.0	89.1/86.3	72.0 (76.6/70.1)
XLNet	94.5/89.0	88.8/86.1	89.1/86.3	81.8 (85.5/80.2)
RoBERTa	94.6/88.9	89.4/86.5	89.8/86.8	83.2 (86.5/81.3)
UPM	-	-	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)**

ELECTRA[1]

- This paper proposes a replaced token detection task. This approach replaces some tokens with plausible alternatives sampled from a generator network, and then trains a discriminative model to predict whether each token was replaced by a generator sample or not.

$$\mathcal{L}_{\mathrm{MLM}}\left(\boldsymbol{x}, \theta_G\right) = \mathbb{E}\left(\sum_{i \in \boldsymbol{m}} -\log p_G\left(x_i | \boldsymbol{x}^{\mathrm{masked}}\right)\right) \tag{7}$$

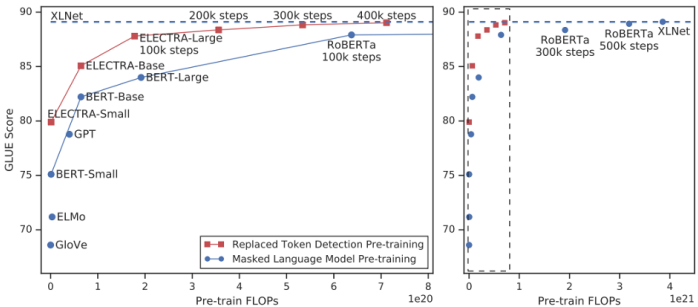
$$\min_{\theta_G, \theta_D} \sum_{\boldsymbol{x} \in \mathcal{X}} \mathcal{L}_{\mathrm{MLM}}\left(\boldsymbol{x}, \theta_G\right) + \lambda \mathcal{L}_{\mathrm{Disc}}\left(\boldsymbol{x}, \theta_D\right) \tag{8}$$



[1]Clark, Kevin, et al. "Electra: Pre-training text encoders as discriminators rather than generators." arXiv preprint arXiv:2003.10555 (2020).

ELECTRA FLOPS

- ELECTRA substantially outperforms MLM-based methods such as BERT and XLNet given the same model size, data, and computation.



ELECTRA Results

- Results on GLUE

Model	Train FLOPs	CoLA	SST	MRPC	STS	QQP	MNLI	QNLI	RTE	WNLI	Avg.*	Score
BERT	1.9e20 (0.06x)	60.5	94.9	85.4	86.5	89.3	86.7	92.7	70.1	65.1	79.8	80.5
RoBERTa	3.2e21 (1.02x)	67.8	96.7	89.8	91.9	90.2	90.8	95.4	88.2	89.0	88.1	88.1
ALBERT	3.1e22 (10x)	69.1	**97.1**	**91.2**	92.0	90.5	**91.3**	–	89.2	91.8	89.0	–
XLNet	3.9e21 (1.26x)	70.2	**97.1**	90.5	**92.6**	90.4	90.9	–	88.5	**92.5**	89.1	–
ELECTRA	3.1e21 (1x)	**71.7**	**97.1**	90.7	92.5	**90.8**	**91.3**	**95.8**	**89.8**	**92.5**	**89.5**	**89.4**

- Results on SQuAD

Model	Train FLOPs	Params	SQuAD 1.1 dev		SQuAD 2.0 dev		SQuAD 2.0 test	
			EM	F1	EM	F1	EM	F1
BERT-Base	6.4e19 (0.09x)	110M	80.8	88.5	–	–	–	–
BERT	1.9e20 (0.27x)	335M	84.1	90.9	79.0	81.8	80.0	83.0
SpanBERT	7.1e20 (1x)	335M	88.8	94.6	85.7	88.7	85.7	88.7
XLNet-Base	6.6e19 (0.09x)	117M	81.3	–	78.5	–	–	–
XLNet	3.9e21 (5.4x)	360M	**89.7**	**95.1**	87.9	**90.6**	87.9	90.7
RoBERTa-100K	6.4e20 (0.90x)	356M	–	94.0	–	87.7	–	–
RoBERTa-500K	3.2e21 (4.5x)	356M	88.9	94.6	86.5	89.4	86.8	89.8
ALBERT	3.1e22 (44x)	235M	89.3	94.8	87.4	90.2	88.1	90.9
BERT (ours)	7.1e20 (1x)	335M	88.0	93.7	84.7	87.5	–	–
ELECTRA-Base	6.4e19 (0.09x)	110M	84.5	90.8	80.5	83.3	–	–
ELECTRA-400K	7.1e20 (1x)	335M	88.7	94.2	86.9	89.6	–	–
ELECTRA-1.75M	3.1e21 (4.4x)	335M	**89.7**	94.9	**88.0**	**90.6**	**88.7**	**91.4**

Overview

Conclusion

- Background
  - language model
  - sequence to sequence model
- Pre-BERT Era
  - semi-supervised sequence learning
  - context2vec
  - pre-trained seq2seq
  - ELMo
  - OpenAI GPT
- BERT
- Post-BERT Era
  - RoBERTa
  - XLNet
  - ERNIE (Tsinghua & Baidu)
  - ALBERT
  - ELECTRA

Thanks.

HP: http://keg.cs.tsinghua.edu.cn/jietang/
Email: jietang@tsinghua.edu.cn