8篇论文梳理BERT相关模型进展与反思

2019-09-05 | 作者：陈永强

BERT 自从在 arXiv 上发表以来获得了很大的成功和关注，打开了 NLP 中 2-Stage 的潘多拉魔盒。随后涌现了一大批类似于“BERT”的预训练（pre-trained）模型，有引入 BERT 中双向上下文信息的广义自回归模型 XLNet，也有改进 BERT 训练方式和目标的 RoBERTa 和 SpanBERT，还有结合多任务以及知识蒸馏（Knowledge Distillation）强化 BERT 的 MT-DNN 等。除此之外，还有人试图探究 BERT 的原理以及其在某些任务中表现出众的真正原因。以上种种，被戏称为 BERTology。本文尝试汇总上述内容，作抛砖引玉。

目录

近期 BERT 相关模型一览

1. XLNet 及其与 BERT 的对比

我们的讨论从 XLNet 团队的一篇博文开始，他们想通过一个公平的比较证明最新预训练模型 XLNet 的优越性。但什么是 XLNet 呢?

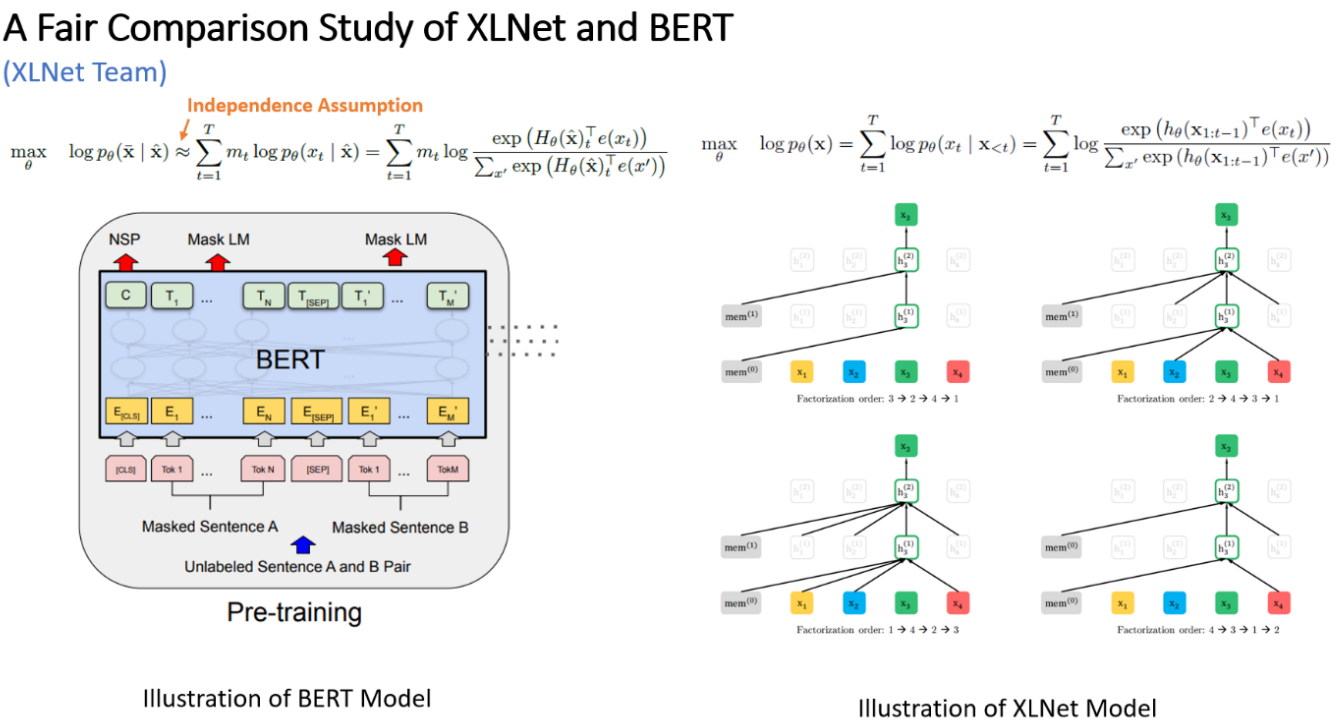


图1：XLNet 和 BERT 对比图

我们知道，BERT 是典型的自编码模型（Autoencoder），旨在从引入噪声的数据重建原数据。而 BERT 的预训练过程采用了降噪自编码（Variational Autoencoder）思想，即 MLM（Mask Language Model）机制，区别于自回归模型（Autoregressive Model），最大的贡献在于使得模型获得了双向的上下文信息，但是会存在一些问题：

1. Pretrain-finetune Discrepancy：预训练时的[MASK]在微调（fine-tuning）时并不会出现，使得两个过程不一致，这不利于 Learning。

2. Independence Assumption：每个 token 的预测是相互独立的。而类似于 New York 这样的 Entity，New 和 York 是存在关联的，这个假设则忽略了这样的情况。

自回归模型不存在第二个问题，但传统的自回归模型是单向的。XLNet 团队想做的，就是让自回归模型也获得双向上下文信息，并避免第一个问题的出现。

他们主要使用了以下三个机制：

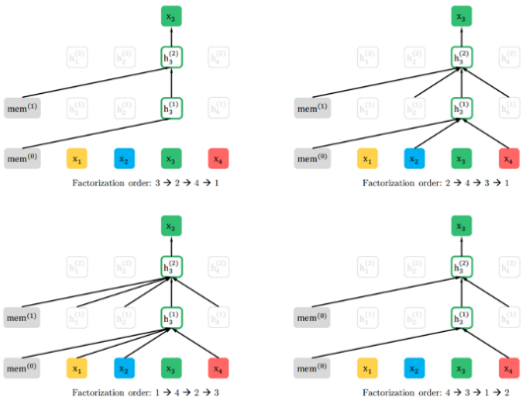
- Permutation Language Model
- Two-Stream Self-Attention
- Recurrence Mechanism

接下来我们将分别介绍这三种机制。

Permutation Language 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:

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

图2：XLNet 模型框架图

在预测某个 token 时，XLNet 使用输入的 permutation 获取双向的上下文信息，同时维持自回归模型原有的单向形式。这样的好处是可以不用改变输入顺序，只需在内部处理。

它的实现采用了一种比较巧妙的方式：使用 token 在 permutation 的位置计算上下文信息。如对于，当前有一个 2 -> 4 ->3 ->1 的排列，那么我们就取出 token_2 和 token_4 作为AR 的输入预测 token_3。不难理解，当所有 permutation 取完时，我们就能获得所有的上下文信息。

这样就得到了我们的目标公式：

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

但是在原来的公式中，我们只使用了 h_θ (x_(Z < t)) 来表示当前token“上文”的 hidden representation，使得不管模型要预测哪个位置的 token，如果“上文”一致，那么输出就是一致的。因此，新的公式做出了改变，引入了要预测的 token 的位置信息。

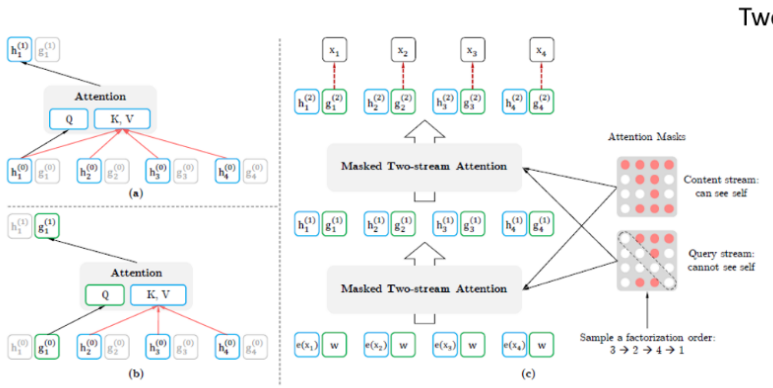
$$\text{Position Info:}\quad p_\theta(X_{z_t}=x\mid \mathbf{x}_{z<t})=\frac{\exp\left(e(x)^\top g_\theta(\mathbf{x}_{\mathbf{z}<t},z_t)\right)}{\sum_{x'}\exp\left(e(x')^\top g_\theta(\mathbf{x}_{\mathbf{z}<t},z_t)\right)}$$

此外，为了降低模型的优化难度，XLNet 使用了 Partial Prediction，即只预测当前 permutation 位置 c 之后的 token，最终优化目标如下所示。

Partial Prediction:

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

Two-Stream Self-Attention



Two Attention Streams:

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

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

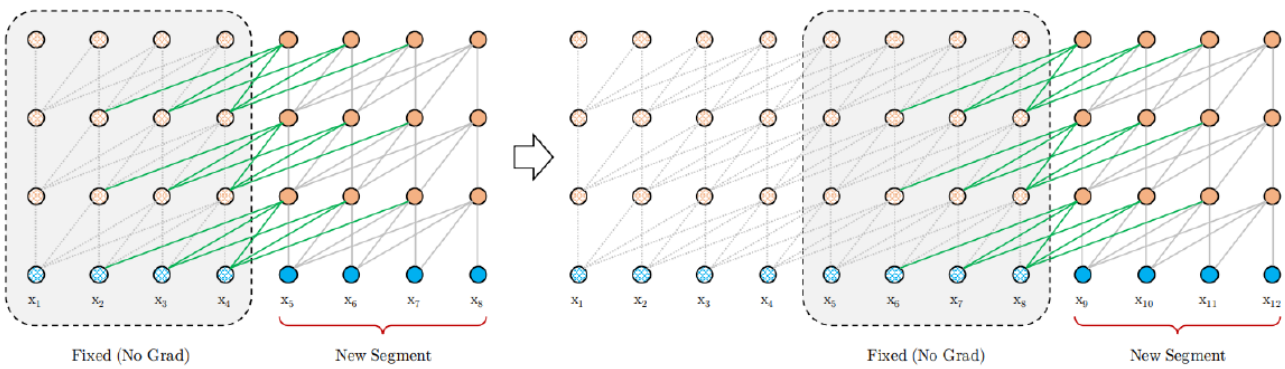
content stream: use both z_t and x_{z_t}

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

图3：Two-Stream Self-Attention 机制

该机制所要解决的问题是，当我们获得了g_θ (x_{Z<t},z_t)后，我们只有该位置信息以及“上文”的信息，不足以去预测该位置后的 token；而原来的 h_θ (x_{Z<t}) 则因为获取不到位置信息，依然不足以去预测。因此，XLNet 引入了 Two-Stream Self-Attention 机制，将两者结合起来。

Recurrence Mechanism



Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context
(Dai et al. CoRR abs/1901.02860)

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

图4：Recurrence Mechanism 机制

该机制来自 Transformer-XL，即在处理下一个 segment 时结合上个 segment 的 hidden representation，使得模型能够获得更长距离的上下文信息。而在 XLNet 中，虽然在前端采用相对位置编码，但在表示 h_θ (x_{Z<t}) 的时候，涉及到的处理与permutation 独立，因此还可以沿用这个机制。该机制使得 XLNet 在处理长文档时具有较好的优势。

XLNet 与 BERT 的区别示例

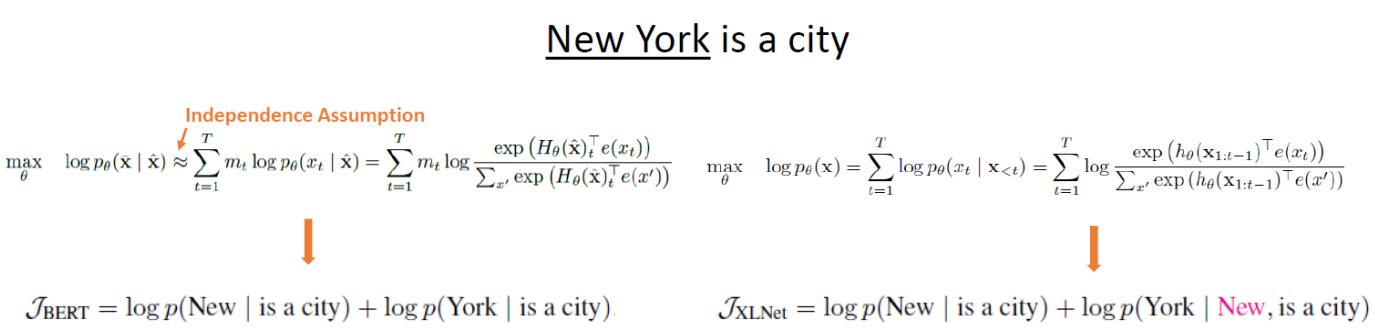


图5：XLNet 与 BERT 的区别示例

为了说明 XLNet 与 BERT 的区别，作者举了一个处理“New York is a city”的例子。这个可以直接通过两个模型的公式得到。假设我们要处理 New York 这个单词，BERT 将直接 mask 这两个 tokens，使用“is a city”作为上下文进行预测，这样的处理忽略了 New 和 York 之间的关联；而 XLNet 则通过 permutation 的形式，可以使得模型获得更多如 York | New, is a city 这样的信息。

公平地比较 XLNet 与 BERT

为了更好地说明 XLNet 的优越性，XLNet 团队发表了开头提到的博文“A Fair Comparison Study of XLNet and BERT”。

在这篇博文中，XLNet 团队控制 XLNet 的训练数据、超参数（Hyperparameter）以及网格搜索空间（Grid Search Space）等与 BERT 一致，同时还给出了三个版本的 BERT 进行比较。BERT 一方则使用以下三个模型中表现最好的模型。

- 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

实验结果如下。

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.

表1：XLNet 与 BERT 实验结果对比

从中可以看出，在相同设定情况下，XLNet 完胜 BERT。但有趣的是：

- XLNet 在使用 Wikibooks 数据集时，在MRPC（Microsoft Research Paraphrase Corpus: 句子对来源于对同一条新闻的评论，判断这一对句子在语义上是否相同）和 QQP（Quora Question Pairs: 这是一个二分类数据集。目的是判断两个来自于 Quora 的问题句子在语义上是否是等价的）任务上获得了不弱于原版 XLNet 的表现；
- BERT-WWM 模型普遍表现都优于原 BERT；
- 去掉 NSP（Next Sentence Prediction）的 BERT 在某些任务中表现会更好；

除了 XLNet，还有其他模型提出基于 BERT 的改进，让 BERT 发挥更大的潜能。

2. RoBERTa: A Robustly Optimized BERT Pretraining Approach

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										
$BERT_{LARGE}$	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	**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

表2：RoBERTa 在 GLUE 中的实验结果

RoBERTa 是最近 Facebook AI 联合 UW 发布的 BERT 预训练模型，其改进主要是如图所示几点，除了调参外，还引入了 Dynamically Change Mask Pattern 并移除 Next Sentence Prediction，使得模型在 GLUE Benchmark 排名第一。作者的观点是：BERT is significantly undertrained。

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
$BERT_{BASE}$	88.5/76.3	84.3	92.8	64.3
$XLNet_{BASE}$ (K = 7)	–/81.3	85.8	92.7	66.1
$XLNet_{BASE}$ (K = 6)	–/81.0	85.6	93.4	66.7

- Larger Byte-Pair Encoding Vocabulary from 30K to 50K

表3：RoBERTa 各个机制的效果比较实验

不同于原有的 BERT 的 MLM 机制，作者在总共40个 epoch 中使用10种不同的 Mask Pattern，即每种 Mask Pattern 训练4代，作为 static 策略；作者还引入了 dynamic masking 策略，即每输入一个 sequence 就为其生成一个 mask pattern。最终发现，新策略都比原 BERT 好，而 dynamic 总体上比 static 策略要好一些，并且可以用于训练更大的数据集以及更长的训练步数，因此最终选用 dynamic masking pattern。

作者还通过替换 NSP 任务进行预训练。虽然 BERT 中已经做了尝试去掉 NSP 后的对比，结果在很多任务中表现会下降，但是包括前文 XLNet 团队所做的实验都在质疑这一结论。

选用的新策略包括：

- Sentence-Pair+NSP Loss：与原 BERT 相同；
- Segment-Pair+NSP Loss：输入完整的一对包含多个句子的片段，这些片段可以来自同一个文档，也可以来自不同的文档；
- Full-Sentences：输入是一系列完整的句子，可以是来自同一个文档也可以是不同的文档；
- Doc-Sentences：输入是一系列完整的句子，来自同一个文档；

结果发现完整句子会更好，来自同一个文档的会比来自不同文档的好一些，最终选用 Doc-Sentences 策略。

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

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

	MNLI	QNLI	QQP	RTE	SST
Single-task single models on dev					
$\text{BERT}_{\text{LARGE}}$	86.6/-	92.3	91.3	70.4	93.2
$\text{XLNet}_{\text{LARGE}}$	89.8/-	93.9	91.8	83.8	95.6
RoBERTa	**90.2/90.2**	**94.7**	**92.2**	**86.6**	**96.4**
Ensembles on test (from leaderboard as of July 25, 2019)					
ALICE	88.2/87.9	95.7	**90.7**	83.5	95.2
MT-DNN	87.9/87.4	96.0	89.9	86.3	96.5
XLNet	90.2/89.8	98.6	90.3	86.3	**96.8**
RoBERTa	**90.8/90.2**	**98.9**	90.2	**88.2**	96.7

MRPC	CoLA	STS	WNLI	Avg
88.0	60.6	90.0	-	-
89.2	63.6	91.8	-	-
90.9	**68.0**	**92.4**	**91.3**	-
92.6	**68.6**	91.1	80.8	86.3
92.7	68.4	91.1	89.0	87.6
93.0	67.8	91.6	**90.4**	88.4
92.3	67.8	**92.2**	89.0	**88.5**

RoBERTa in GLUE Test

表4：RoBERTa 在更多训练数据和更久训练时间下的实验结果

作者还尝试了更多的训练数据以及更久的训练时间，发现都能提升模型的表现。

这种思路一定程度上与 OpenAI 前段时间放出的 GPT2.0 暴力扩充数据方法有点类似，但是需要消耗大量的计算资源。

3. SpanBERT: Improving Pre-training by Representing and Predicting Spans