[预训练语言模型专题] RoBERTa: 捍卫BERT的尊严

原创 管扬 朴素人工智能

来自专辑

预训练语言模型

本文为预训练语言模型专题的第15篇。

快速传送门

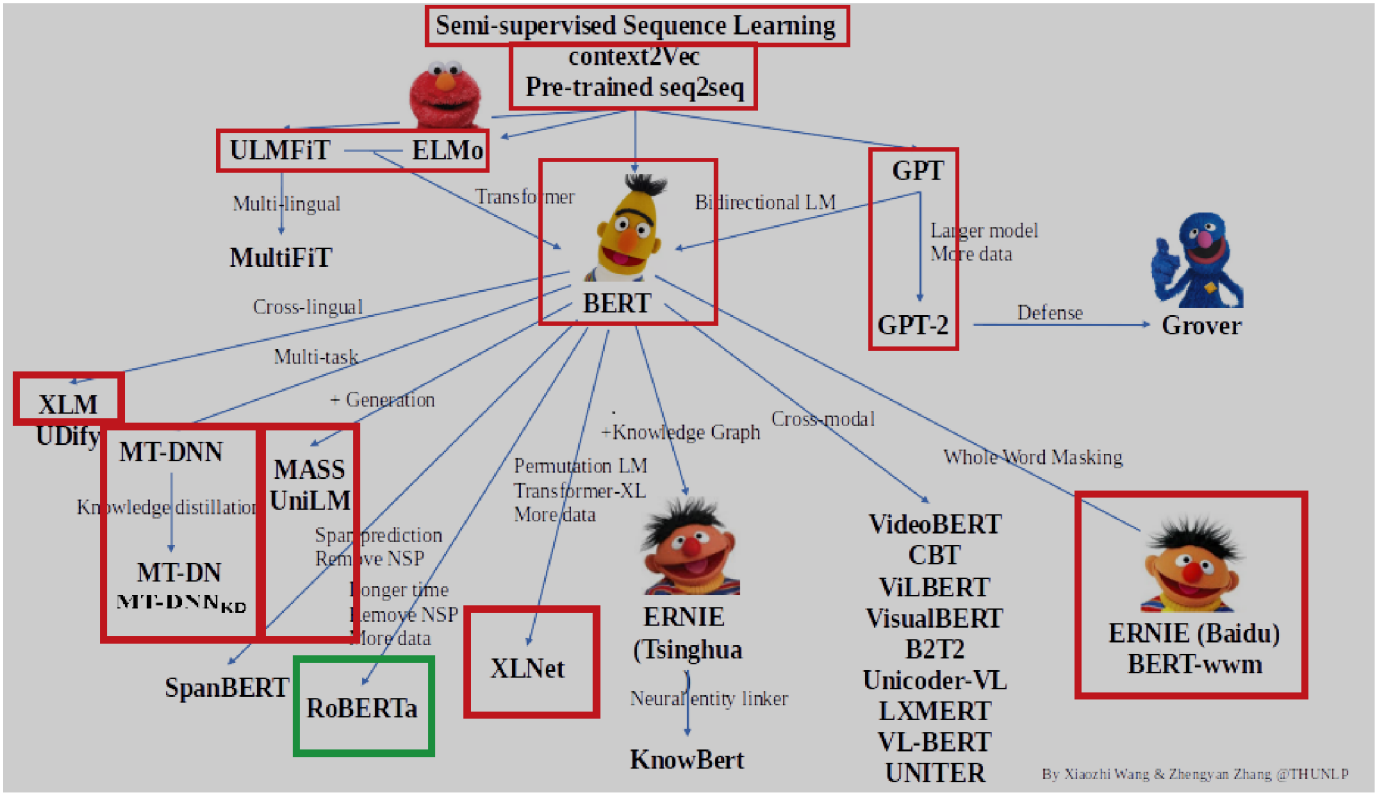
1-4:[萌芽时代]、[风起云涌]、[文本分类通用技巧] 、 [GPT家族]

5-8:[BERT来临]、[浅析BERT代码]、[ERNIE合集]、[MT-DNN(KD)]

9-12:[Transformer]、[Transformer-XL]、[UniLM]、[Mass-Bart]

13-14：[跨语种模型]、[XLNet]

感谢清华大学自然语言处理实验室对**预训练语言模型架构**的梳理，我们将沿此脉络前行，探索预训练语言模型的前沿技术，红框中为已介绍的文章，绿框中为本期介绍的模型，欢迎大家留言讨论交流。



RoBERTa: A Robustly Optimized BERT Pretraining Approach（2019）

众所周知，BERT对NLP领域的发展产生了极大的影响，刚一出现，它在当时的不少榜单上，都取得了压倒性的优势。在那之后，预训练语言模型领域有了蓬勃的发展，各种模型百花齐放，其中有一些还在各排行榜上超越了BERT，成为了当时的state-of-art。

而本文是对BERT预训练语言模型进行了一个重复性的研究。作者认为BERT实际上是大大地被**undertrained**了，将之充分地训练以后可以**反超**在它之后发布的所有模型的效果。同时，在GLUE，RACE，SQuAD等数据集上，作者进行了实验，确实又夺得了state-of-art的效果。作者认为，这显示人们以前忽略了设计选择的重要性，同时对最近的一些报告上，模型效果提升的来源（是否是结构导致的提升）提出了疑问。

大量的预训练模型如ELMo，GPT，BERT，XLM，XLNet等都给下游的任务带来了很大的收益，但是很难去确定这些收益来源于哪里，因为语言模型的预训练是很昂贵的。作者对BERT的预训练进行了仔细的评估，包括**超参数和训练集大小的配置**，发现BERT其实没有很充分地训练，从而提出了更好地训练BERT的方法，称为RoBERTa，它超过了在BERT之后发表的所有post-BERT方法的效果。

方法其实很简单：

1. 用**更长的时间，更大的batch size，更多的数据**进行训练
2. 去掉BERT中的**NSP**目标(next sentence prediction)
3. 在**更长的句子**上进行训练
4. 根据训练数据**动态地改变mask**的模式

同时作者总结了一下文章的主要贡献：

- 展示了一组重要的BERT的设计选择和新的训练策略，使模型在下游任务上取得更好的效果。
- 使用了新的dataset，CC-News，并确认了更多的数据预训练能给下游任务带来更好的效果
- 证明了在合适的设计选择下，masked language model（BERT）是和所有最新发布的模型相比都是极具竞争力的

模型细节

RoBERTa同样使用了Adam，β1为0.9，β2为0.999，ε=1e-6，L2 weight decay为0.01，全部层的dropout为0.1，线性激活为GELU。和BERT不同的是，其warmup的步数，以及warmup到达的最大学习率会根据设置而进行finetune，而且RoBERTa所有的训练样本几乎都是全长512的序列，这与BERT先通过小的序列长度进行训练不同。

数据

BERT类的模型依赖于大规模的文本进行训练，RoBERTa的训练样本就比BERT更多而且更有多样性。最后使用了BOOKCORPUS(16GB)，CC-NEWS(76GB)，OPENWEBTEXT(38GB)，STORIES(31GB) 总共160GB左右的数据，BOOKCORPUS是BERT原始的训练数据，可以看到，RoBERTA数据量**多了10倍**。

模型通过混合精度在8 * 32GB的V100上进行训练。

动态掩码（Dynamic Masking)

BERT的masking是在预处理时进行的，导致这种Masking是静态的，每个epoch的masking结果一致。原本的BERT为了避免这种情况，把数据复制了十份，然后进行了十种不同的静态masking。这样在40个epochs的训练中，同样masking的数据会在整个训练中出现四次。

而RoBERTa中使用Dynamic Masking，只是在序列送入模型中的时候才去进行**动态的masking**，这样在更大的数据集上或者更多步数的训练上会表现更好。

输入形式与NSP任务

作者比较了几种输入方式

- SEGMENT-PAIR+NSP 首先模型输入为包括两个segment的SEGMENT PAIR，其中每个segment可以包含多个句子，用[SEP]分割，这和BERT本来的输入方式相同，并且包含NSP任务。
- SENTENCE-PAIR+NSP 模型输入为包含两个句子的SENTENCE-PAIR，用[SEP]分割，这会导致序列的长度远小于512，同样保持了NSP任务。
- FULL-SENTENCES 模型输入为全量的句子，填满512的长度，采集样本的时候可以跨越文章的界限，去除了NSP loss
- DOC-SENTENCES 模型输入和FULL-SENTENCE类似，但是一个样本不能跨越两个document

比较了SEGMENT-PAIR和SENTENCE-PAIR后，作者发现，使用单个句子相比使用句子组成的Segment会**降低**下游任务的结果。

比较了SEGMENT-PAIR和DOC-SENTENCES两个模式后，作者发现**没有NSP**，下游任务的效果更好。

最后，作者比较DOC-SENTENCES和FULL-SENTENCES，作者结论是**不跨越document**进行取样效果更好。

文本编码方式

Byte-Pair Encoding（BPE)是一种介于字级别和词级别之间的编码表示，基于subwords units。BPE的字典一般从10K-100K的范围内，随着语料而变化。但有一种聪明的做法是利用bytes来替代subword中的unicode字符，这样就可以用一个不太大的（50Kunit）的字典来对任

意的输入进行编码而不引入未知的token。虽然相比于BERT使用30K的字符BPE的字典，Roberta会在Embedding层额外引入15M到20M的参数量，但是用一个固定的词典，相比而言是比较好的，而两者在性能上几乎没有区别。

效果比较

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

从上图中可以看到，数据增多，训练更长时间都能带来收益，和XLNET的比较中，RoBERTa并没有落于下风。

在GLUE，RACE等数据集上，也比当时的state-of-art，XLNet更好。

	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**

最后总结下，提高BERT模型下游任务上性能的方法有，用更多数据集以更大的batch size来训练更多的时间，去掉NSP任务目标，在更长的句子上预训练，动态地改变masking等。经过这些操作的RoBERTa，在各排行榜上都取得了state-of-art的效果。附上工程链接 https://github.com/pytorch/fairseq

未完待续

本期的论文就给大家分享到这里，感谢大家的阅读和支持，下期我们会给大家带来其他预训练语言模型的介绍，敬请大家期待！

欢迎关注朴素人工智能，这里有很多最新最热的论文阅读分享，有问题或建议可以在公众号下留言。

往期推荐

- 万字长文带你一览ICLR2020最新Transformers进展（上）
- [预训练语言模型专题] Transformer-XL 超长上下文注意力模型
- [预训练语言模型专题] XLNet：公平一战！多项任务效果超越BERT
- [预训练语言模型专题] Huggingface简介及BERT代码浅析
- [预训练语言模型专题] 结合HuggingFace代码浅析Transformer