**基于预训练模型的情感分析**

**一、前言**

近年来纷繁复杂的大模型涌现，其中一些模型的参数量大概在几十到几百M，可以在个人PC上训练。本文即是以探求各模型之间、各模型不同参数量版本的性能差异为目的，系统性地比较多个大模型在情感分析下游任务上的能力。

**二、方法**

基于pyhon中现有NLP框架，对基于Transformer的ERINE,BERT,GPT,BART,

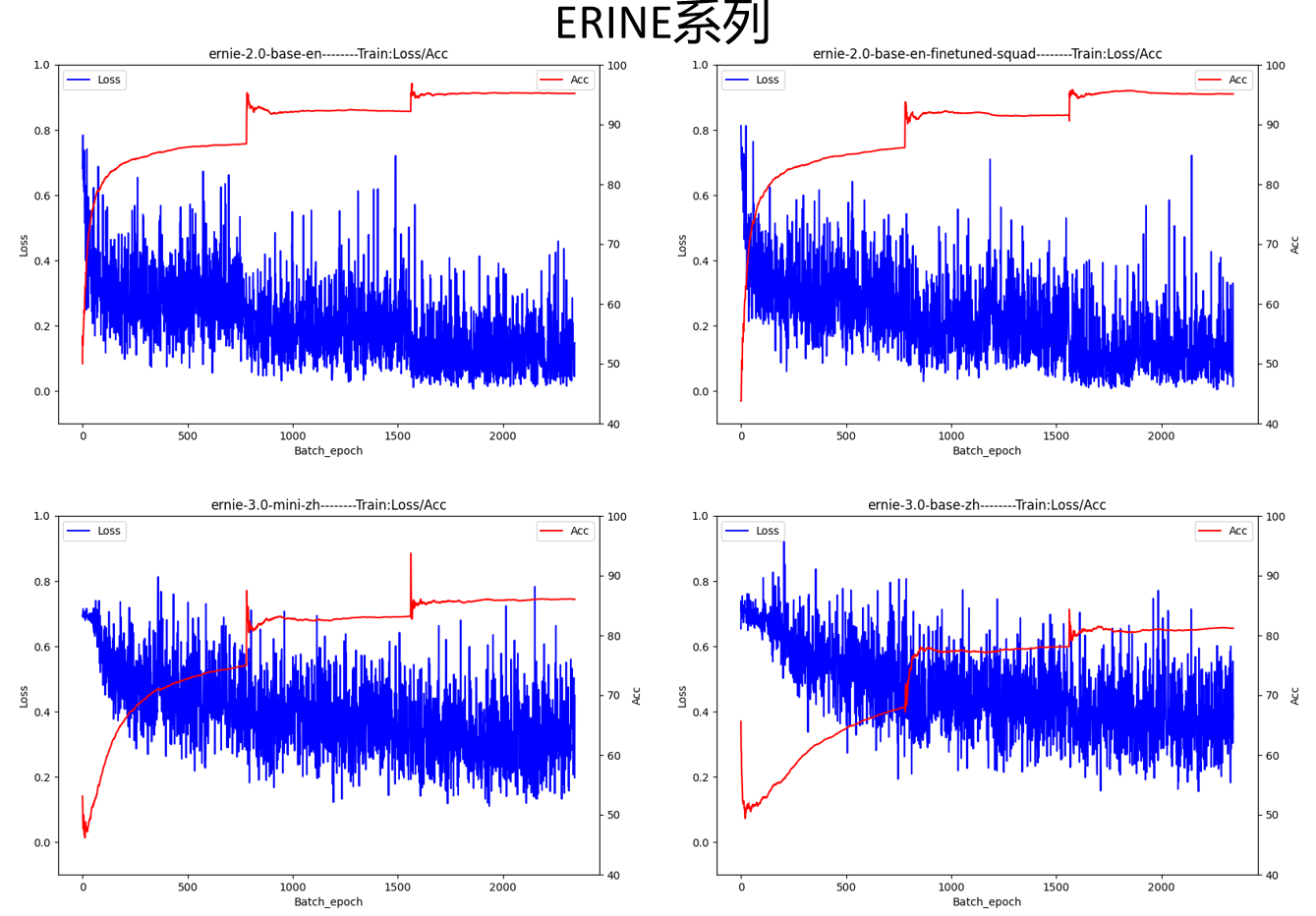
ELECTRA,Roberta的不同规模模型、改进型模型在IMDB数据集上微调。以情感分析正确率、收敛速度以及训练时间作为评估标准。并在微调后用于其他情感分析数据集,加入针对非预期结果的额外实验,以及与传统方法的效果进行比较。

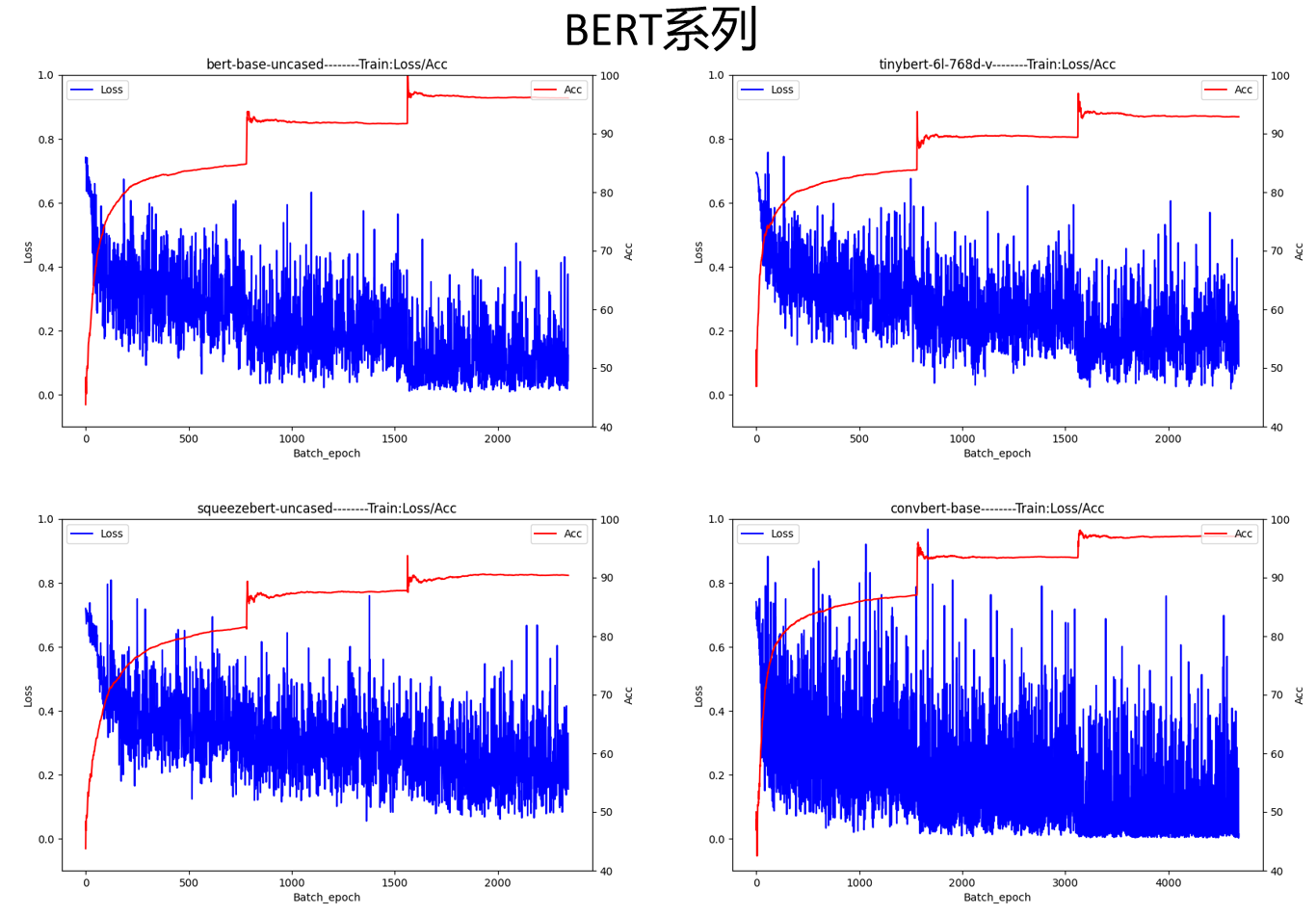
**三、实验设定和结果**

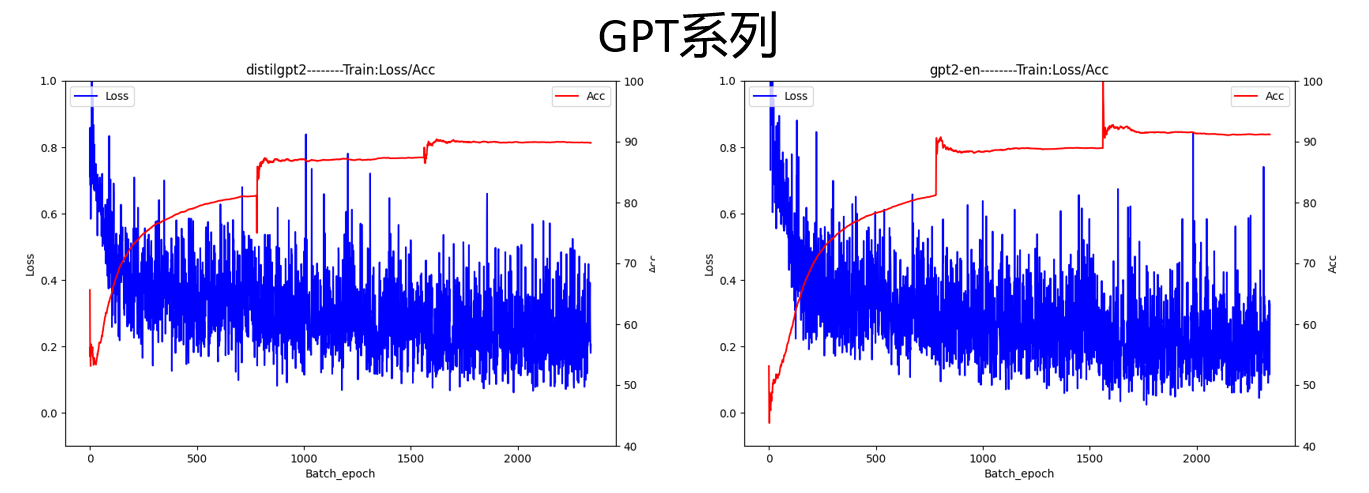
微调采用IMDB训练集，不设验证集，为减少每个模型实验时间，测试集只取其20%。将结果中较好和较差模型再在sst-2数据集测试,其他实验设定如下：

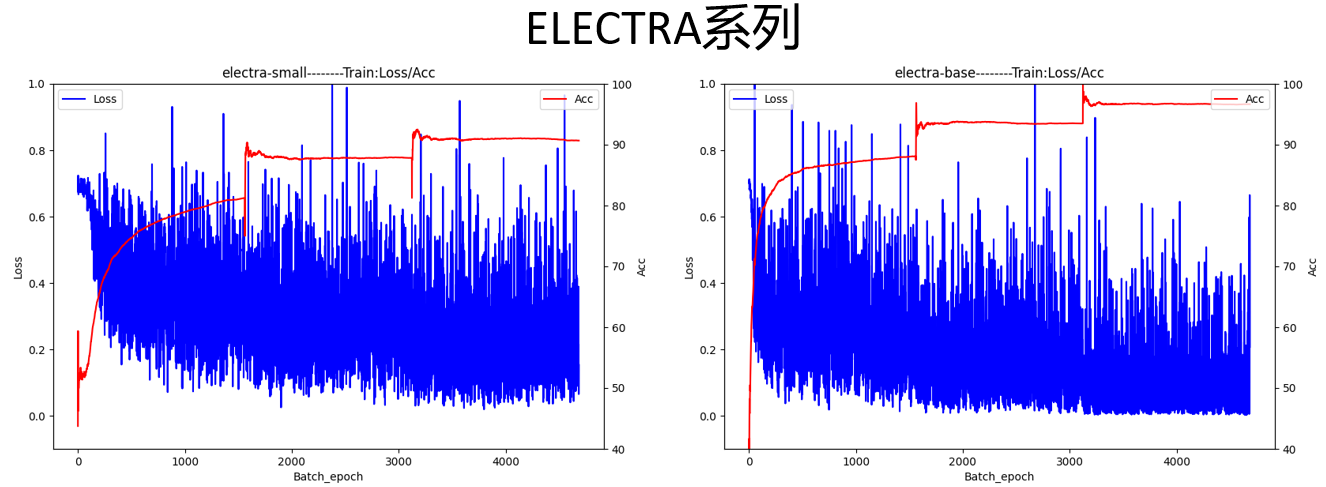
|  |  |
| --- | --- |
| 设定项目 | 内容 |
| Epoch | 3 |
| Batch\_size | 16 或 32（根据参数量改变） |
| Optimizer | AdamW（初始学习率2e-5） |

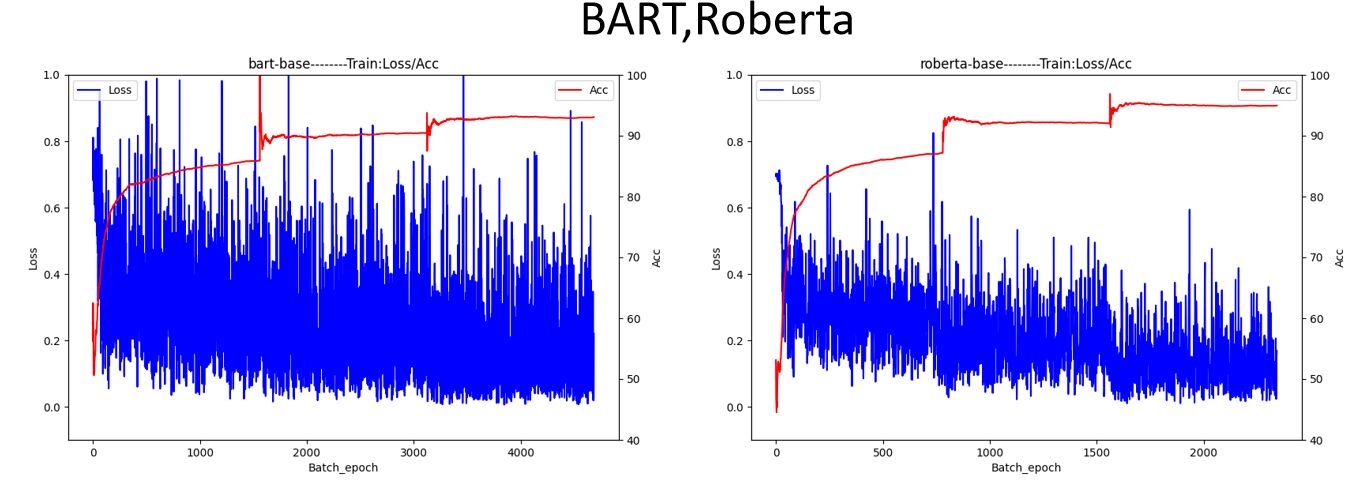
**训练集**结果：双y轴折线图展示，左蓝Loss，右红Acc。Loss以及Acc是训练集上每轮batch的累计值，且在新epoch中重新累计,模型名称在图标标题处。











PS:由于算力限制,参数较多的模型批次大小设置为16,x轴数据范围更大。

**测试集**实验结果及其他：以表格展示

其中Acc(i)是指第i轮epoch后在测试集上的正确率（单位%）；Loss(i)同理，参数为模型参数量（单位M），Time(单位s)为每个batch的平均用时

模型名称中base表示基准规模，规模大小为：

而en表示在英语文本上预训练,zh表示在中文文本(包括少量英文)上预训练;finetuned表示在别的数据集上先行微调过。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 模型 | Acc1 | Acc2 | Acc3 | Loss1 | Loss2 | Loss3 | 参数 | Time |
| ernie-2.0-base-en | 89.92 | 95.23 | 98.02 | 0.229 | 0.040 | 0.034 | 103 | 0.345 |
| ernie-2.0-base-en-finetuned-squad | 89.08 | 94.63 | 98.42 | 0.497 | 0.105 | 0.006 | 110 | 0.343 |
| ernie-3.0-mini-zh | 80.97 | 87.86 | 93.45 | 0.378 | 0.239 | 0.109 | 27 | 0.183 |
| ernie-3.0-base-zh | 81.13 | 86.84 | 90.48 | 0.312 | 0.302 | 0.300 | 118 | 0.352 |
| bert-base | 88.52 | 95.41 | 98.74 | 0.176 | 0.048 | 0.006 | 110 | 0.348 |
| squeezebert | 85.38 | 91.03 | 95.70 | 0.487 | 0.267 | 0.033 | 51 | 1.700 |
| tinybert | 86.82 | 92.05 | 95.95 | 0.464 | 0.294 | 0.142 | 67 | 0.198 |
| convbert-base | 89.96 | 95.70 | 98.50 | 0.341 | 0.055 | 0.005 | 106 | 0.472 |
| gpt2-en | 87.48 | 92.45 | 96.10 | 0.172 | 0.181 | 0.027 | 117 | 0.190 |
| distilgpt2 | 85.74 | 89.54 | 94.07 | 0.494 | 0.200 | 0.017 | 82 | 0.072 |
| electra-small | 86.52 | 91.65 | 95.79 | 0.358 | 0.485 | 0.043 | 14 | 0.218 |
| electra-base | 91.27 | 96.50 | 98.42 | 0.217 | 0.078 | 0.001 | 109 | 0.329 |
| roberta-base | 90.20 | 95.55 | 98.48 | 0.208 | 0.040 | 0.117 | 125 | 0.192 |
| bart-base | 89.96 | 95.21 | 98.24 | 0.041 | 0.280 | 0.016 | 217 | 0.346 |

PS:计算Time时基准批次大小为32，而以16为批次大小的会合并两个batch。

后续下游任务:在IMDB上微调后,将模型在SST-2验证集的20%上做情感分析

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 模型 | bert-base | tinybert | distilgpt2 | electra |
| 正确率% | 97.50 | 90.9 | 91.2 | 96.8 |

除以上展现的实验内容外,还进行了传统情感分析方法,小模型以及非预训练大模型,结果如下:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 方法/模型 | 基于词典 | BiLSTM | TextCNN | 非预训练BERT |
| 正确率% | 73.0 | 87.6 | 87.0 | 93.3 |

**四、分析讨论**

选取以上模型一方面有算力的限制,另一方面是基于:主要在在ERINE系列上进行模型版本以及规模的分析,用ELECTRA系列上验证这一分析;在BERT系列上进行改进方法的分析,在GPT系列上验证这一分析。

对以上折线图以及表格进行分析:

1.对ERINE系列模型:

主要是版本,规模以及预训练数据集不同。

(i)即使中文预训练集包含了一些英语单词,模型正确率也差英文预训练模型6%左右。本质上是单词表大小的影响。

(ii)在SQuAD问答数据集上先进行微调的模型和同等的正常数据集模型在正确率与时间上的表现相近,可以认为额外使用少量非下游任务相关的数据集影响很小。

(iii)erine3.0-zh的mini规模最终表现反而比base规模好,经查阅得知两个规模的模型只有隐藏层大小不一致,但结果却与预期相反。但是在ELECTRA系列模型中,大规模的base最终能力是优于small的,即使耗时更多。为进一步分析,补做ERINE系列的位于base和small之间的medium模型的实验:

|  |  |  |  |
| --- | --- | --- | --- |
| 模型 | Acc1 | Acc2 | Acc3 |
| ernie-3.0-medium-zh | 89.92 | 95.23 | 98.02 |

补做实验更与预期不符,估计只能归结于概率原因。

2.对BERT系列模型:

主要是在bert-base的基础上加入了优化和改进方法。

(i)[squeezebert使用了分组卷积来替换bert中自注意力层中的几个操作，](https://arxiv.org/abs/2006.11316" \t "https://www.bing.com/_blank)但在正确率与时间上的表现并不如bert。

(ii)tinybert其实是通过知识蒸馏,将大型预训练模型的知识转移到小型模型并实现压缩,以正确率而言,tinybert保证了原bert模型近乎97%的能力,并且减少了约50%的参数量和每批次时间。

(iii)convbert[使用基于区间的动态卷积来提升bert](https://arxiv.org/abs/2008.02496" \t "https://www.bing.com/_blank),初始模型能力略强于bert,但是最终能力持平且参数量和原来相当,每批次时间变长。

3.对GPT系列模型:

(i)distilgpt2同样采用知识蒸馏的方式对gpt2模型进行压缩,最终效果类似于tinybert和bert,以牺牲少许模型能力的代价降低了参数量和训练时间。

(ii)gpt是单向编码自回归模型而bert是双向编码模型,由于编码方式导致gpt只能利用上文信息,在NLU任务(包括情感分析)上的表现并不如bert,以上实验数据中gpt正确率比bert低2%左右。由于gpt单向编码,训练时间也比双向的bert少,实验数据中每批次时间约为bert的50%。

4.ELECTRA系列模型:

(i)electra模型的特征为替换掉输入序列中的一些单词,并让模型预测这些单词是否被替换,以使模型更好地理解输入序列中的上下文信息。实验数据表面electra-base与bert-base能力相当,但是electra-small用最少的参数量保证了相当高的正确率(参数量为tinybert的25%但正确率相当)。

5.RoBERTa和BART模型:

(i)RoBERTa其实也是BERT的改进,主要改动在去除了下一句的预测任务,采用动态掩码,即每次输入时生成一个新的掩码模式,以获得更好的文本理解能力.实验数据表明,该模型和bert-base正确率,参数量相当,但是批次时间更短。

(ii)BART模型采用一个双向Transformer编码器和一个单向Transformer解码,模型最终能力和BERT相近,但参数量更大。

6.其他分析:

(i)由于Acc是每轮epoch刷新,epoch中累计,故在新epoch开始时Acc会有小幅跳跃,但是由折线图仍然见得几乎每个模型都在2-3轮epoch时收敛,并且收敛速度相差无几。

(ii)更为合理的分析模型的时间效率的方法应该是,计算单位M参数量的批次时间,即批次时间/参数量,经计算distilgpt2最小,squeezebert最大。distilgpt2最小是能够解释的,因其单向编码且参数量较少。

(iii)Loss小时并不意味着Acc小,两者没有很强的相关性,这与经验认识有些许出入。

(iv)在IMDB微调后,再在SST-2数据集上进行测试,正确率下降大约1-3%,说明微调有效且效果不错。

(v)由最后的表格可知,预训练大模型的正确率比前预训练大模型方法高出10-20%。

**五、结论**

1.由以上分析基本可以得到经验性的结论:模型规模越大,表现越好,耗时越长以及改进模型优于原模型。但是另一方面,实验中也有例外,并且补做实验让分析更没法进行,只能归于随机因素。

2.大部分改进方法/模型在本实验的IMDB数据集上并不能保证性能更好,甚至由于随机因素在各方面更差。但是其中的distil蒸馏方法多次被验证了可以在牺牲极少正确率的情况下,大幅减少模型参数和训练时间。

3.以上各模型的收敛时间都相差无几,但是单位M参数的时间效率仍有较大差异,并且只能部分解释差异原因。

4.Loss和Acc没有很强的相关性。

5.相比于传统方法,小模型以及非预训练大模型,预训练大模型的正确率提升是非常显著的,达到10-20%。

**六、实验过程总结**

由于之前使用学习提供的算力平台,发现效率非常低,于是将选题防在一些较小的大模型上。刚开始实验时首先尝试的是部署Github上现成的项目,选择的是ELMO,但是经过耗时的配置环境,编写代码的努力后发现模型太大训练不了。后面发现了百度开发的paddle框架,提供了封装好的API,大大化简了部署流程。但是依然存在不少问题,即该框架没有详细的使用手册,很多不清楚的地方需要大量尝试。此外,有些API无法使用,翻看并尝试修改源码也无法解决。最终的实验内容是尝试成功后的内容。最后进行结果分析时还阅读了各模型原论文,以求给出实验结果的合理分析。实验项目放在

**七、组员分工**

余淼一人一组,负责所有内容。

**八、参考文献列表**

[1] 郝政, 等. "ERNIE 2.0: A Continual Pre-training Framework for Language Understanding"[D]. <https://arxiv.org/abs/1907.12412>

[2] 郝政, 等. "ERNIE 3.0: Large-Scale Knowledgeable Language Model"[D]. <https://arxiv.org/abs/2106.09436>

[3] Jacob Devlin, 等. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding"[D]. <https://arxiv.org/abs/1810.04805>

[4] Forrest N. Iandola, 等. "SqueezeBERT: What Can Computer Vision Teach NLP about Efficient Neural Networks?"[D]. <https://arxiv.org/abs/2006.11316>

[5] Zhaoxiang Zhang, 等. "TinyBERT: Distilling BERT for Natural Language Understanding"[D]. <https://arxiv.org/abs/1909.10351>

[6] Zihang Jiang, 等. "ConvBERT: Improving BERT with Span-based Dynamic Convolution"[D]. <https://arxiv.org/abs/2104.03759>

[7] Alec Radford, 等. "Language Models are Unsupervised Multitask Learners"[D]. <https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf>

[8] Alec Radford, 等. "Language Models are Few-Shot Learners"[D].

<https://arxiv.org/abs/2005.14165>

[9] Kevin Clark, 等. "ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators"[D]. <https://arxiv.org/abs/2003.10555>