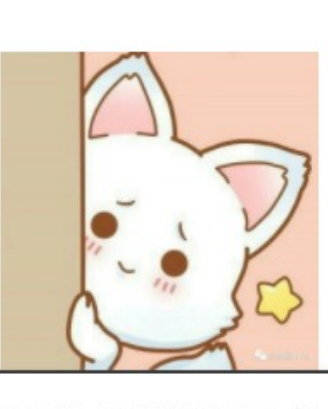
ACL20 | 让笨重的BERT问答匹配模型变快！

原创 曹庆庆 夕小瑶的卖萌屋 2020-06-04 22:23:54 手机阅读

来自专辑
卖萌屋@自然语言处理



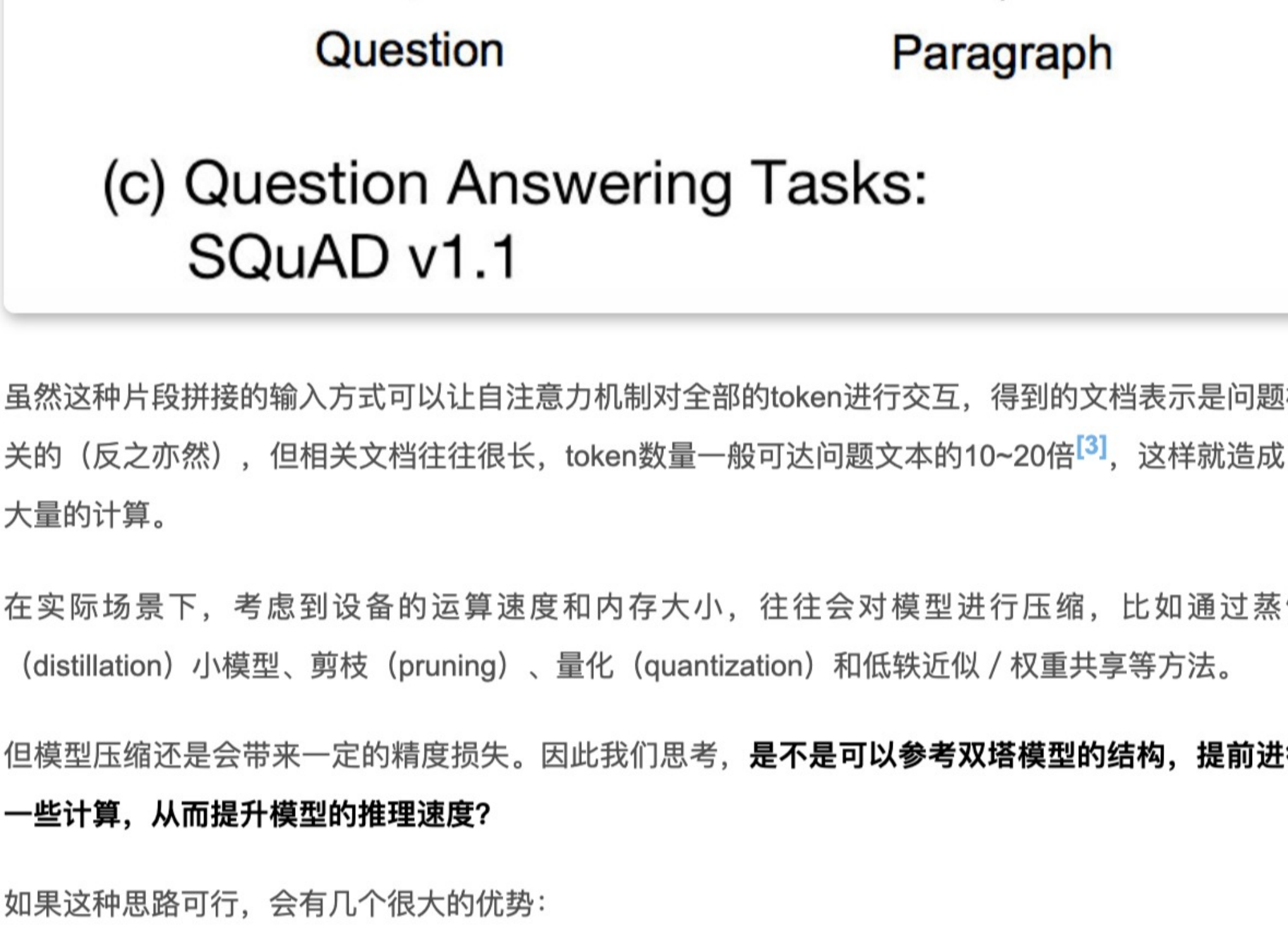
一只小狐狸带你解锁**炼丹术&NLP**秘籍

作者：曹庆庆（Stony Brook University 在读PhD，关注Efficient NLP、QA方向，详见awk.ai）

背景

BERT、XLNet、RoBERTa等基于Transformer[1]的预训练模型推出后，自然语言理解任务都获得了大幅提升。问答任务（Question Answering，QA）[2]也同样取得了很大的进步。

用BERT类模型来做问答或阅读理解任务，通常需要将问题和问题相关文档拼接一起作为输入文本，然后用自注意力机制对输入文本进行多层交互编码，之后用线性分类器判别文档中可能的答案序列。如下图：



虽然这种片段拼接的输入方式可以让自注意力机制对全部的token进行交互，得到的文档表示是问题相关的（反之亦然），但相关文档往往很长，token数量一般可达问题文本的10~20倍[3]，这样就造成了大量的计算。

在实际场景下，考虑到设备的运算速度和内存大小，往往会对模型进行压缩，比如通过蒸馏（distillation）小模型、剪枝（pruning）、量化（quantization）和低轶近似 / 权重共享等方法。

但模型压缩还是会带来一定的精度损失。因此我们思考，**是不是可以参考双塔模型的结构，提前进行一些计算，从而提升模型的推理速度？**

如果这种思路可行，会有几个很大的优势：

1. **它不需要大幅修改原来的模型架构**
2. **也不需要重新预训练，可以继续使用标准Transformer初始化+目标数据集fine-tune的精调方式**
3. **还可以叠加模型压缩技术**

经过不断地尝试，我们提出了**《Deformer：Decomposing Pre-trained Transformers for Faster Question Answering》**[4]，在小幅修改模型架构且不更换预训练模型的情况下提升推理速度。下面将为大家介绍我们的思考历程。

论文链接：

https://awk.ai/assets/deformer.pdf

代码链接：

https://github.com/StonyBrookNLP/deformer

Arxiv访问慢的小伙伴也可以在订阅号后台回复关键词【**0604**】下载论文PDF。

模型结构

在开篇的介绍中，我们指出了QA任务的计算瓶颈主要在于自注意力机制需要交互编码的token太多了。因此我们猜想，**是否能让文档和问题在编码阶段尽可能地独立？**

这样的话，就可以提前将最难计算的文档编码算好，只需要实时编码较短的问题文本，从而加速整个QA过程。

部分研究表明，Transformer 的低层（lower layers）编码主要关注一些局部的语言表层特征（词形、语法等等），到高层（upper layers）才开始逐渐编码与下游任务相关的全局语义信息。因此我们猜想，**至少在模型的某些部分，“文档编码能够不依赖于问题”的假设是成立的。** 具体来说可以在 Transformer 开始的低层分别对问题和文档各自编码，然后再在高层部分拼接问题和文档的表征进行交互编码，如图所示：

为了验证上述猜想，我们设计了一个实验，测量文档在和不同问题交互时编码的变化程度。下图为各层输出的文档向量和它们中心点cosine距离的方差：

可以看到，对于BERT-Based的QA模型，如果编码的文档不变而问题变化，模型的低层表征往往变化不大。这意味着并非所有Transformer编码层都需要对整个输入文本的全部token序列进行自注意力交互。

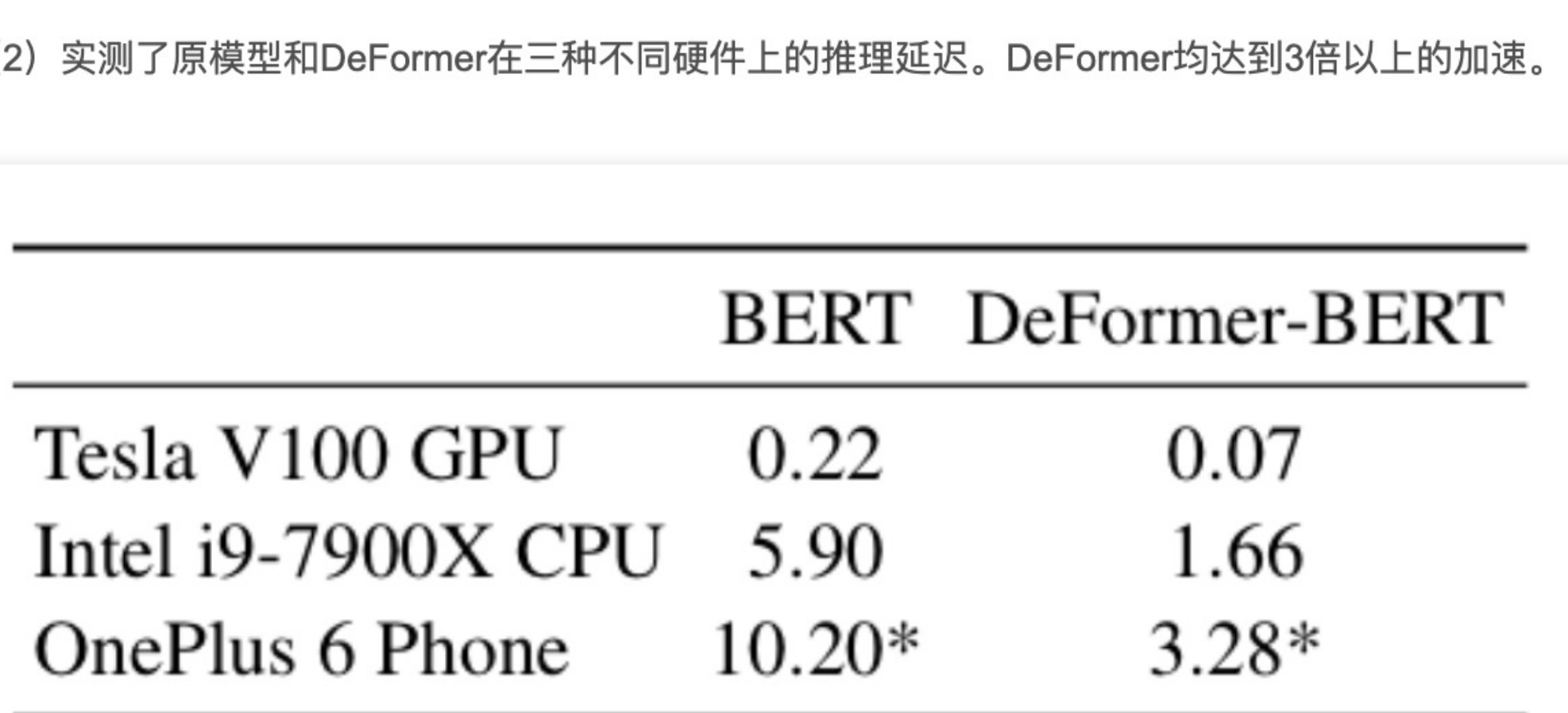
因此，我们提出Transformer模型的一种变形计算方式（称作 DeFormer）：在前k层对文档编码离线计算得到第 k 层表征，问题的第k层表征通过实时计算，然后拼接问题和文档的表征输入到后面$k+1$到n层。下面这幅图示意了DeFormer的计算过程：

值得一提的是，这种方式在有些QA任务（比如SQuAD）上有较大的精度损失，所以我们添加了两个蒸馏损失项，目的是**最小化Deformer的高层表征和分类层logits与原始BERT模型的差异**，这样能控制精度损失在1个点左右。

实验

这里简要描述下四组关键的实验结果：

（1）在三个QA任务上，BERT和XLNet采用DeFormer分解后，取得了**2.7-3.5倍的加速，节省内存65.8-72.0%，效果损失只有0.6-1.8%**。BERT-base（$n=12$）在SQuAD上，设置$k=9$能加快推理3.2倍，节省内存70%。

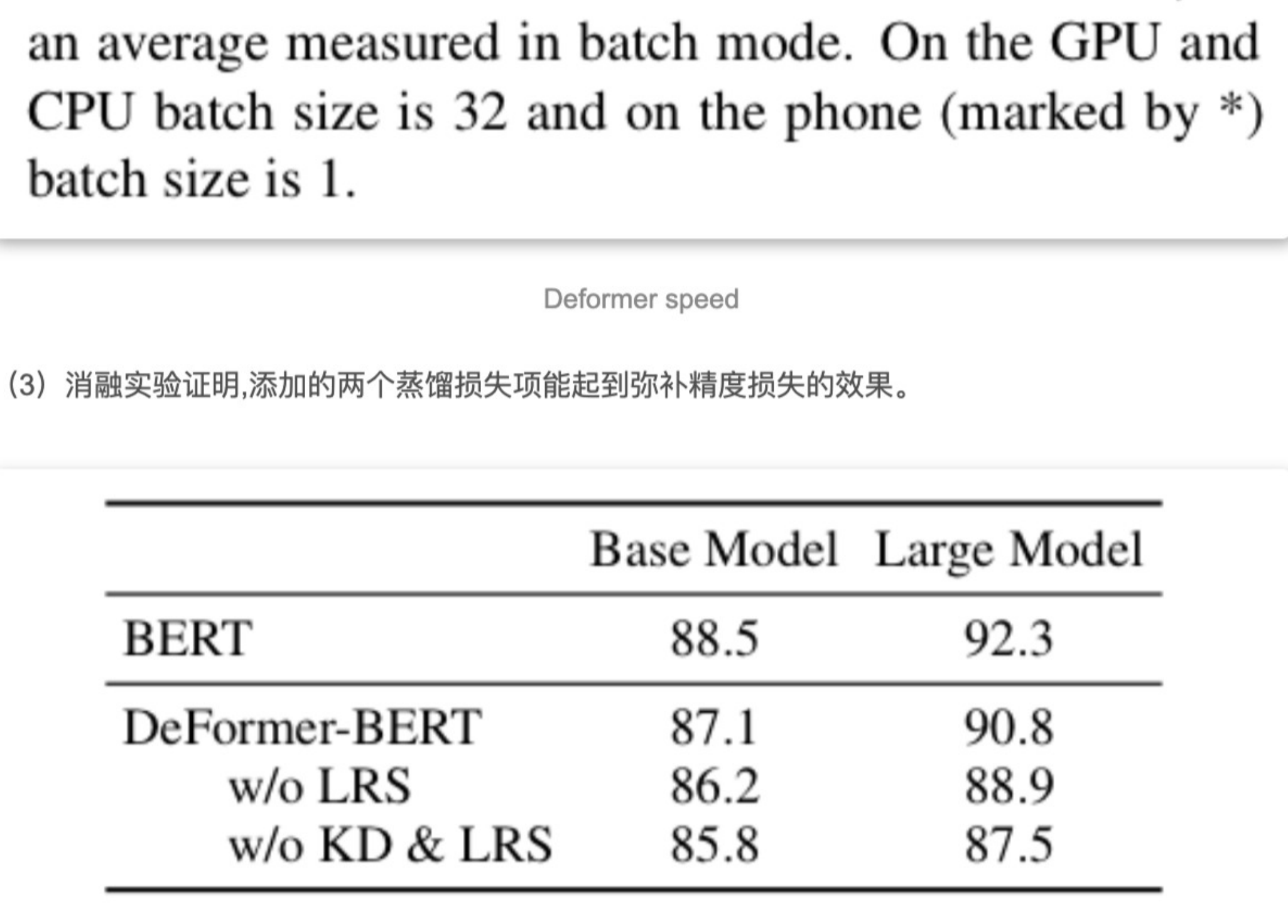


Model	Datasets	Avg. Input Tokens	Original base	DeFormer-base	Performance Drop (absolute | %age)	Inference Speedup (times)	Memory Reduction (%age)
BERT	SQuAD	320	88.5	87.1	1.4 | 1.6	3.2x	70.3
	RACE	2048	66.3	64.5	1.8 | 2.7	3.4x	72.9
	BoolQ	320	77.8	76.8	1.0 | 1.3	3.5x	72.0
XLNet	SQuAD	320	91.6	90.4	1.2 | 1.3	2.7x	65.8
	RACE	2048	70.3	68.7	1.6 | 2.2	2.8x	67.6
	BoolQ	320	80.4	78.8	0.6 | 0.7	3.0x	68.3

Table 1: (i) Performance of original fine-tuned vs fine-tuned models of DeFormer-BERT-base and DeFormer-XLNet-base, (ii) Performance drop, inference speedup and inference memory reduction of DeFormer- over original models for 3 QA tasks. DeFormer-BERT-base uses nine lower layers, and three upper layers with caching enabled, DeFormer-XLNet-base use eight lower layers, and four upper layers with caching enabled. For SQuAD and RACE we also train with the auxiliary losses, and for the others we use the main supervision loss – the settings that give the best effectiveness during training. Note that the choice of the loss doesn't affect the efficiency metrics.

Deformer results

（2）实测了原模型和DeFormer在三种不同硬件上的推理延迟。DeFormer均达到3倍以上的加速。

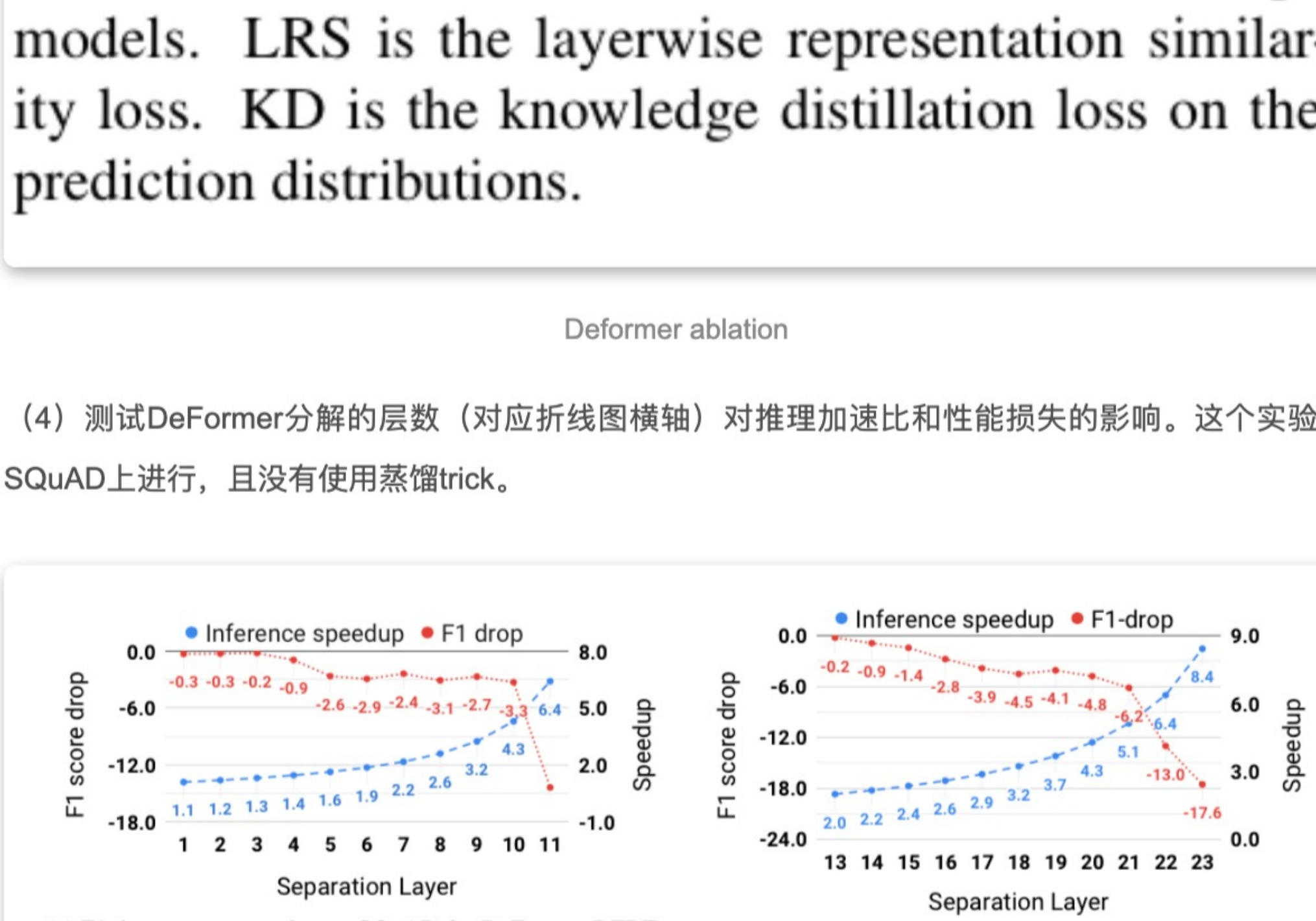


	BERT	DeFormer-BERT
Tesla V100 GPU	0.22	0.07
Intel i9-7900X CPU	5.90	1.66
OnePlus 6 Phone	10.20*	3.28*

Table 4: Inference latency (in seconds) on SQuAD datasets for BERT-base vs DeFormer-BERT-base, as an average measured in batch mode. On the GPU and CPU batch size is 32 and on the phone (marked by *) batch size is 1.

Deformer speed

（3）消融实验证明,添加的两个蒸馏损失项能起到弥补精度损失的效果。

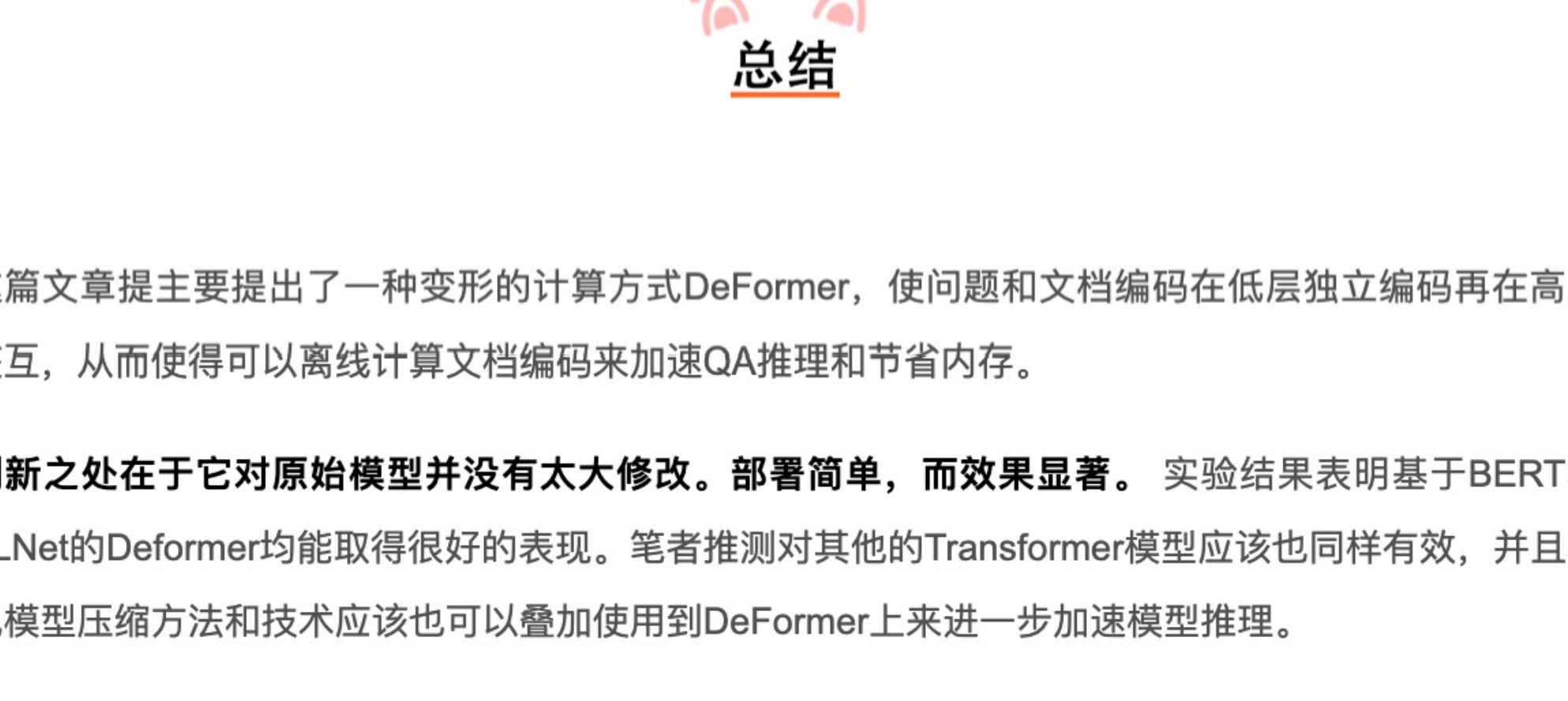


	Base Model	Large Model
BERT	88.5	92.3
DeFormer-BERT	87.1	90.8
w/o LRS	86.2	88.9
w/o KD & LRS	85.8	87.5

Table 5: Ablation analysis on SQuAD datasets for DeFormer-BERT-base and DeFormer-BERT-large models. LRS is the layerwise representation similarity loss. KD is the knowledge distillation loss on the prediction distributions.

Deformer ablation

（4）测试DeFormer分解的层数（对应折线图横轴）对推理加速比和性能损失的影响。这个实验在SQuAD上进行，且没有使用蒸馏trick。



(a) F1 drop versus speedup on SQuAD for DeFormer-BERT-base without auxiliary supervision.

(b) F1 drop versus speedup on SQuAD for DeFormer-BERT-large without auxiliary supervision.

Figure 4: F1 drop versus speedup of DeFormer-BERT model (without auxiliary supervision) when separating at different layers.

Deformer layers

总结

这篇文章提主要提出了一种变形的计算方式DeFormer，使问题和文档编码在低层独立编码再在高层交互，从而使得可以离线计算文档编码来加速QA推理和节省内存。

创新之处在于它对原始模型并没有太大修改。部署简单，而效果显著。 实验结果表明基于BERT和XLNet的Deformer均能取得很好的表现。笔者推测对其他的Transformer模型应该也同样有效，并且其他模型压缩方法和技术应该也可以叠加使用到DeFormer上来进一步加速模型推理。

Arxiv访问慢的小伙伴也可以在订阅号后台回复关键词【**0604**】下载论文PDF。

本文收录于原创专辑：《卖萌屋@自然语言处理》

可能喜欢

- 卖萌屋原创专辑首发，算法镇魂三部曲！
- GPT-3诞生，Finetune也不再必要了！NLP领域又一核弹！
- ACL2020 | 线上搜索结果大幅提升！亚马逊提出对抗式query-doc相关性模型
- 别再蒸馏3层BERT了！变矮又能变瘦的DynaBERT了解一下
- All in Linux：一个算法工程师的IDE断奶之路

参考文献

[1]论文方面可以参考邱老师组的文献综述: Models for Natural Language Processing : A Survey (https://arxiv.org/abs/2003.08271)，实例代码可以参见 huggingface 的 Transfomer 库
[2]严格来说是机器阅读理解，即给出问题从相关文章中提取答案，一般QA系统还包括检索阶段来找到问题相关的文档
[3]比如SQuAD问题平均10个token，但文档平均有116个token
[4]Deformer：Decomposing Pre-trained Transformers for Faster Question Answering: https://awk.ai/assets/deformer.pdf