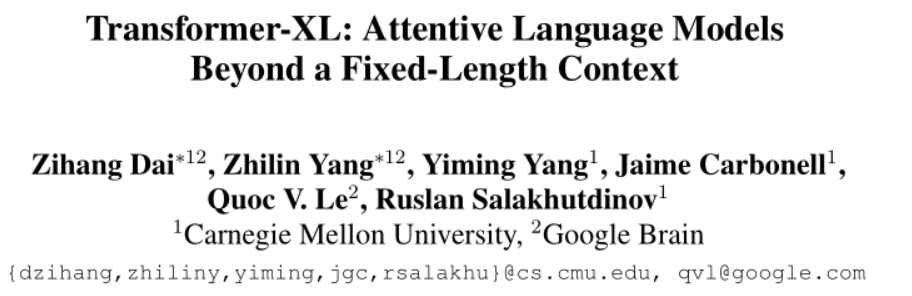
Transformer-XL



本文提出了一种新型的神经网络结构Transformer-XL，它能够学习超过固定长度的dependency，而不会破坏时间连续性。其包含了两个部分：段级递归机制(segment-level recurrence mechanism)和一种新颖的位置编码方案（positional encoding scheme）。该神经网络模型不仅可以获取长期依赖（long-term dependency），而且可以解决语境碎片化问题（contextual fragmentation）。

从结果来看，Transformer-XL能够学习依赖关系长度比RNNs长80%，比vanilla Transformer长450%，并且在长序列和短序列中都取得了更好的表现。

Transformer-XL is the first self-attention model that achieves substantially better results than RNNs on both character-level and word-level language modeling.

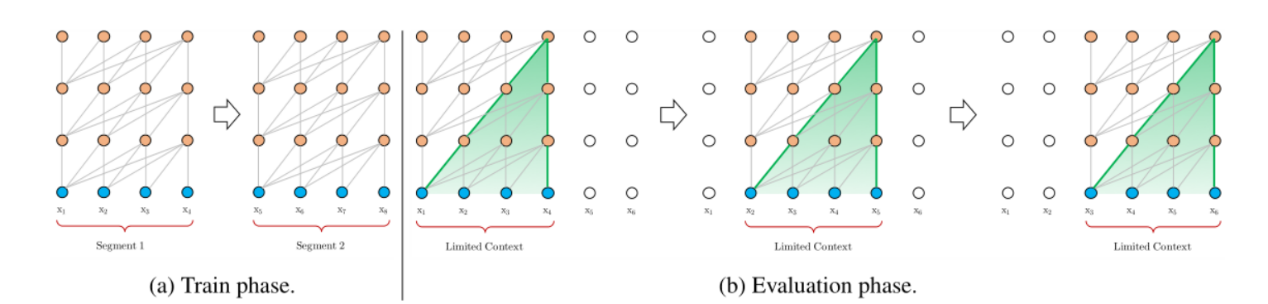
语言建模（language modeling）中存在一个重要的问题就是需要模型具有长期依赖关系。循环神经网络（RNNs）,尤其是LSTM和GRU等神经网络，一直是语言建模中的标准解决方案，并且也在实际测试中取得了不错的结果。在基于LSTM的模型中，为了建模长距离依赖，提出了门控机制（gating mechanism）和梯度裁剪（gradient clipping），目前可以编码的最长距离在200左右，对更长文本编码效果则变差。

Transformer把要处理的文本分割成等长的片段（segment），通常不考虑句子（语义）边界，片段之间没有信息交互。因此，模型无法获取超出既定上下文长度的依赖关系，导致模型优化效率不高且表达能力有限。

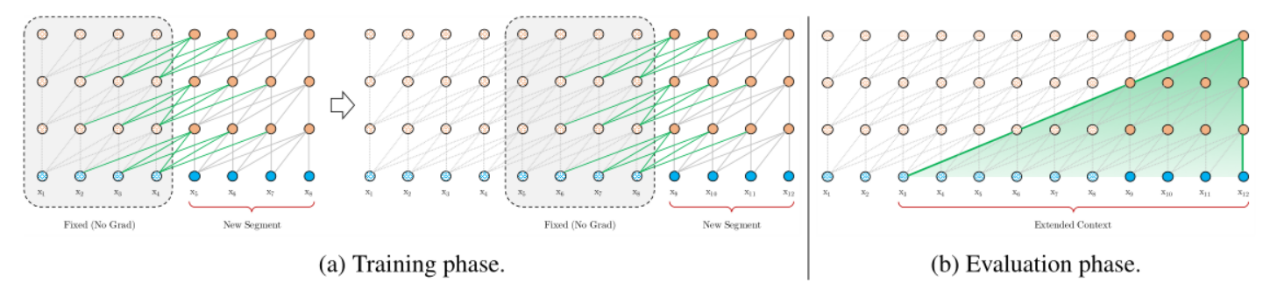
针对以上的问题，本文提出的Transformer-XL采取了两种策略：

* 1. **采用段级递归机制（segment-level recurrence mechanism）：**

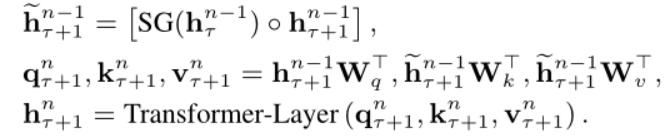
在介绍段级递归机制之前，先了解一下vanilla Transformer处理段序列的机制：将整个语料库拆分为可管理大小的较短段，并且仅在每个段内训练模型，而忽略先前段中的所有上下文信息。



段级递归机制则是：在训练过程中，将前一片段的隐藏状态序列保存，在计算下一片段状态时再利用。



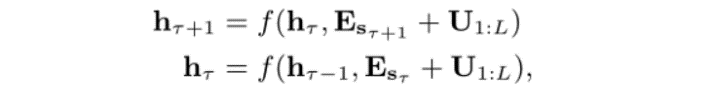
假设两个长度为L的连续片段为s τ =[x τ,1 ,··· ,x τ,L ] and s τ+1 = [x τ+1,1 ,··· ,x τ+1,L ]，由s τ片段产生的第n层隐藏状态序列为，则下一片段s τ+1的第n层隐藏状态为：



与Transformer最大的区别在于，下一片段的,由决定，而包含了前一片段的状态信息。递归的结果，使得有效的上下文信息可以超过两个片段，并且，最大可能的依赖长度线性增长，达到。

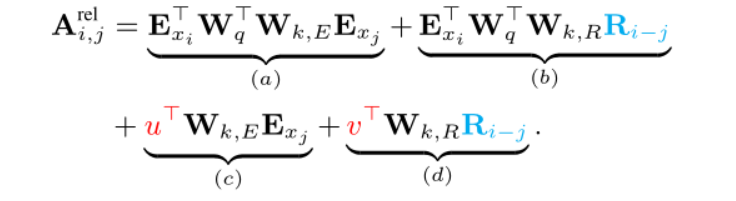
* 1. **使用相对位置编码（relative positional encoding）取代绝对位置编码：**

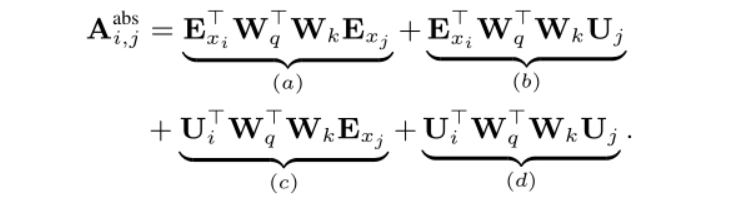
在进行段级递归的时候，如果对输入向量采用绝对位置编码，公式如下图所示，**会出现片段之间不同向量位置编码相同的情况，这样的话，无法在段级递归机制中保证位置信息正确。**



其中，表示片段的词向量，​表示绝对位置向量，可以看到，x τ,j 和x τ+1,j没有任何位置差异。

为了解决以上问题，本文采用了相对位置编码，其同一片段中query q i 和 key k j的attention为：



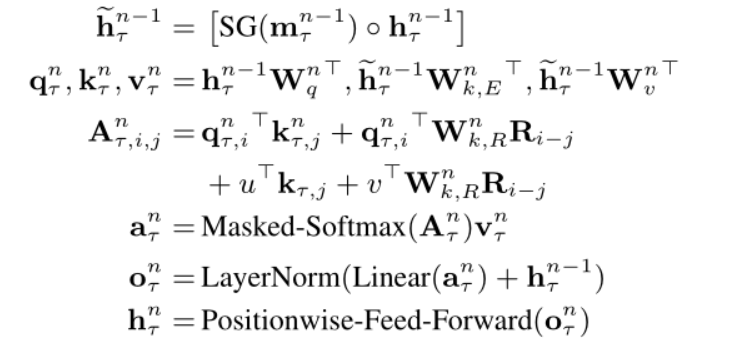


相对位置编码与绝对位置编码的不同在于以下几点：

* + 在（b）和（d）计算key向量部分中，用相对位置编码代替绝对位置编码，这里的R用的是Transformer里用的sinusoid encoding matrix，没有要学习的参数。
  + 在（c）和（d）中分别用一个可学习的参数u和v代替，在计算self-attention时，由于query所有位置对应的query向量是一样的，因此对不同单词的attention偏差应保持相同。
  + 将拆分成分别产生基于内容的key向量和基于位置的key向量。

采用新的参数设置后，每一个部分都有了直接的含义：（a）项代表了content-based addressing,（b）项获取了content-based positional bias，（c）项代表了global content bias，（d）项编码了global positional bias。

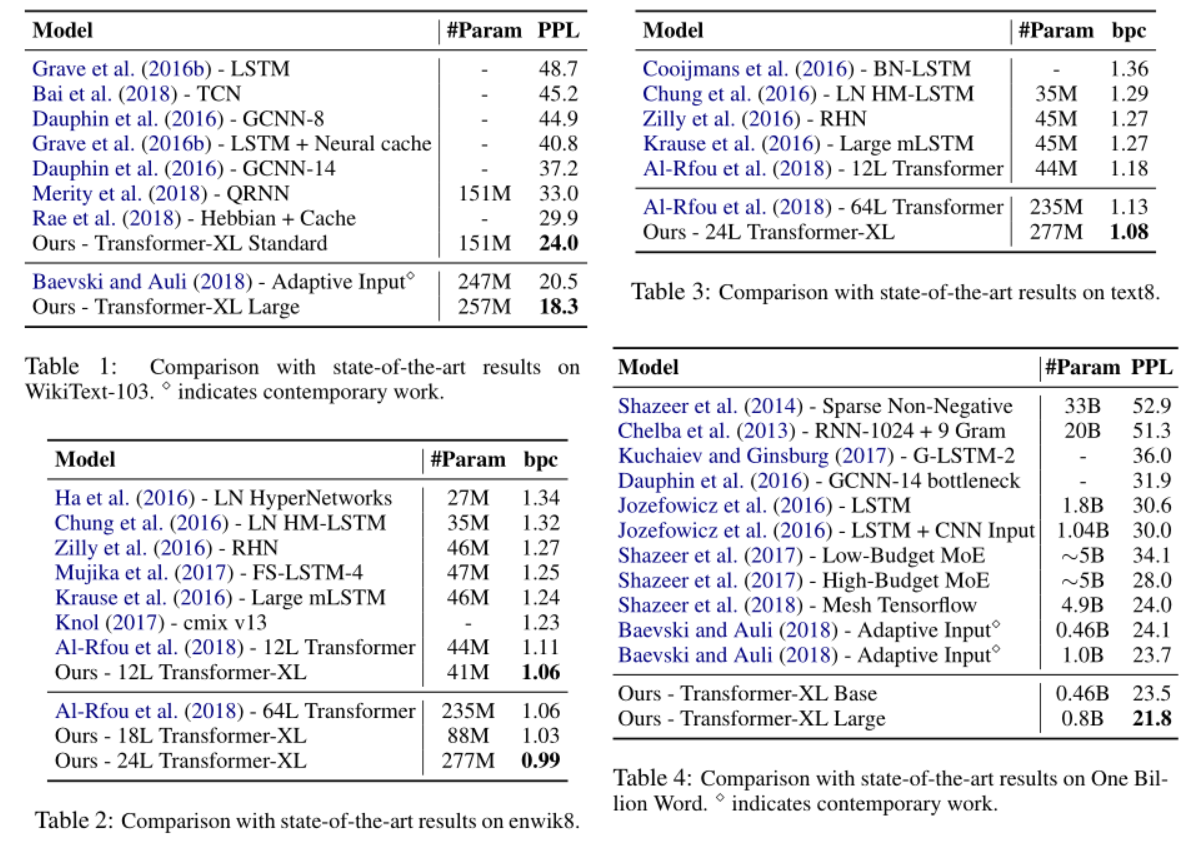
在介绍了段级递归机制和相对位置编码之后，Transformer-XL模型single attention head计算过程如下图所示：



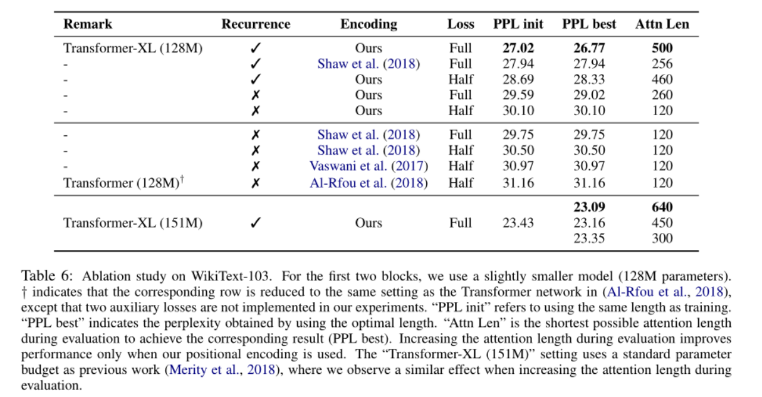
可以看到，在计算完attention后，还要经过Masked-Softmax和LayerNorm和一个基于位置的前馈网络。

实验结果：

Transformer-XL在WikiText-103 , enwik8 , text8 ,One Billion Word , Penn Treebank数据集进行了字符级和词级建模，并于其他模型进行了比较，其结果如下图所示：



本文也进行相应的**ablation study**，来研究段级递归机制和相对位置编码对模型带来的影响。其实验结果如下图所示：



个人总结：Transformer-XL在Transformer基础上加入了段级递归机制和相对位置编码，有效地解决了原模型长距离依赖能力不足带来的上下文碎片化问题，但是也增加了内存的消耗以及计算复杂度。在Transformer模型中加入循环机制是一个不错的想法。