关于ERNIE、Transformer-XL、XLNET的理解

1. ERNIE

BERT的相关知识跟问题

双向的Transformer模型，通过MASKed LM和NSP任务来预训练任务。

对于MASKed LM存在，预训练任务中mask与下游任务的不一致问题，同时mask中文针对的是单字，而中文的更多是词语和短语，这些会影响单字的情况。

ERNIE1.0整体结构上跟Bert保持一致，一方面通过先验知识根据实体和短语的mask能够学习语法和句法信息的语言模型，为了解决OOV的问题和节约词汇表大小的情况，它只是将实体进行mask,并没有作为输入，预测输出。同时增加了其他的数据集训练。

ERNIE1.0还使用了多轮对话修改了NSP，去掉了sep Embedding,增加了dialogue Embedding，在其中识别哪些是Q随机的R。

ERNIE2.0又重新设计了任务来提升效果。它使用了多任务持续学习预训练框架，构建三种类型的无监督任务。

连续多任务学习，是为了不遗忘之前的训练结果，多任务高效进行训练，同时使用上一任务的参数，并且新旧任务一起训练。将每个任务分成多次迭代，框架完成不同迭代的训练自动分配。模型结构上面多了一个Tasking embedding用来区分哪个任务。

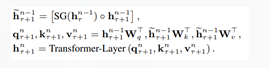
多任务训练任务包括词法级别、语言结构级别、语法级别。

1. Transformer-XL

Transfomer-XL是将Transformer和RNN的优点进行结合的，Transformer优点在于能并行计算（Attention）和长距离问题，缺点在于位置，所以添加了position embedding，长度问题一但超过了固定长度，要么切分或者补padding。RNN优点在于它是时序序列，所以任意长度都可以处理。

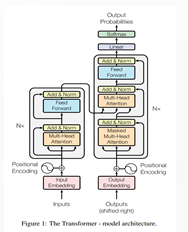
Transfomer-XL它是一种循环的transformer建模机制，采用了相对位置编码方式。

它的传递方式是不同的segment输入，采用拼接的方式，下一层接受上一层的信息，下一层的接受上一时刻的上层信息

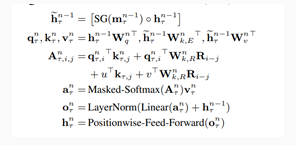


两个segment拼接变成2\*segment\*dk 在计算Q, K,V,其中Q还是采用t+1时刻上一层的输入，K，V则是采用拼接后的来计算，再将他们经过Transformer-Layer之后变成这一层的输出，注意此时的维度变为了segment\*dk。

采用相对位置信息一方面是为了不同segment的位置信息，另一方面描述token之间的种间关系。



最终形态



它跟Transformer相比，变动在于计算attention时候采用相对位置编码的，attention来源于Q、K、V,Q不变，其中KV是经过拼接的结果。

1. XLNET

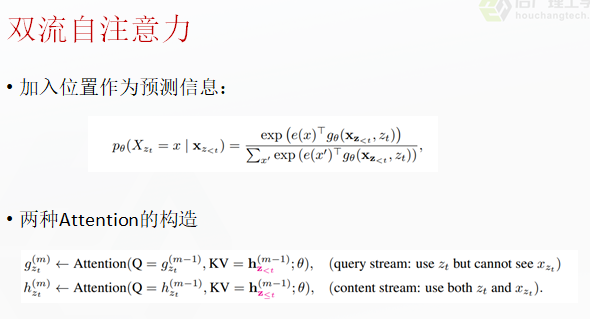
它是融合了AR和AE优点的模型，AR是单向的自回归语言模型，只能利用上文或者下文的信息，可以完成生成类的任务，AE则是自编码语言模型，双向语言模型，可以根据上下文预测mask,缺点在于会造成预训练任务和Fine-tuning不一致。

XLNET它是单向的语言模型，为了解决只能利用一边的信息，它采用了一种排列组合的方法，它不会改变原始词的顺序，通过Attention的Mask来对应不同的分解方法

双流自注意力的思想

一方面要预测后面的字符是哪个，另一方面要预测自己是哪个字符。

所以它采用了两套attention计算Content Stream和Query Stream.Content Stream在计算attention时候允许看到自己，Query Stream则不允许。

训练方法

XLNET的数据输入50%来源于连续的句子，50%是不连续的，随机组合数据随机采样。由于它是用融合了AR的，单向，CLS放在后面。

与BERT相比，它尽量规避Bert中词的独立假设，预测词是出于性能考虑 。因为有些词拼接在一起是对于词的预测是有利的。