1、ERNIE 1.0 , XLNET, RoBERTa, ALBERT 分别基于 BERT 做了哪些改进?

ERNIE 1.0

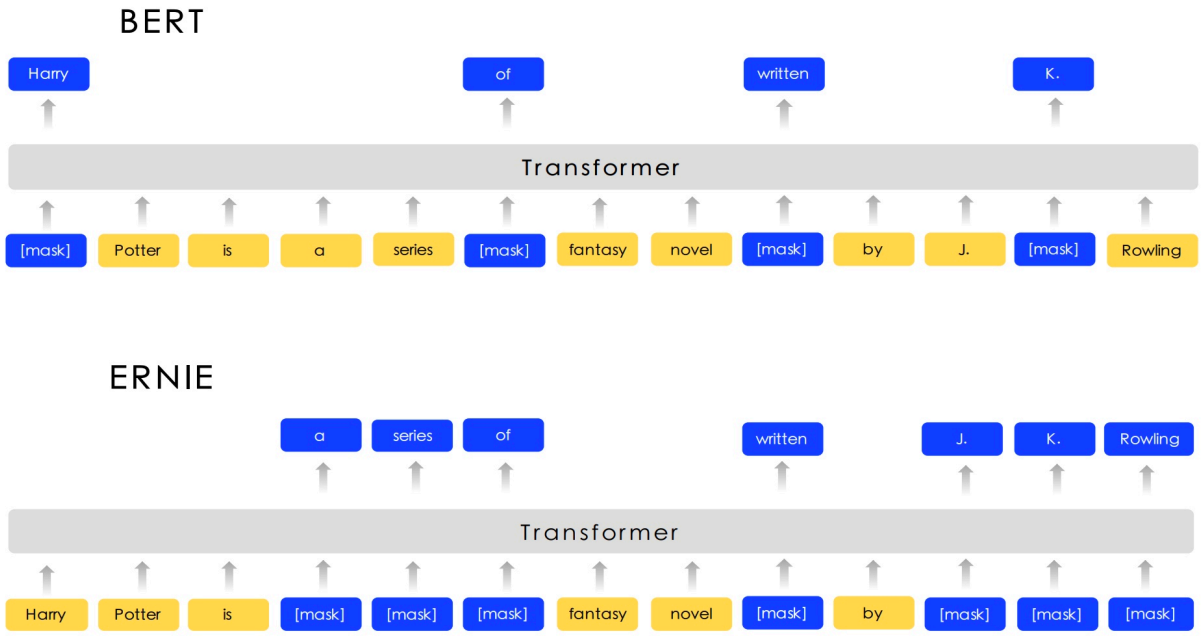

Figure 1: The different masking strategy between BERT and ERNIE

> BERT是对token-level进行建模，这样并没有学到语义单元的完整含义，ERNIE模型中加入了entity-level和phrase-level，用来学习命名实体和语义单元的知识。
> ERNIE引入了对话语料库，从而构建了一个Dialogue LM（DLM）。训练中，通过生成一些假的Question-Response（QR）对，让模型来判断多轮对话是否真实。

XLNET

> BERT 是典型的自编码模型（Autoencoder），旨在从引入噪声的数据重建原数据。而 BERT 的预训练过程采用了降噪自编码（Variational Autoencoder）思想，即 MLM（Mask Language Model）机制，区别于自回归模型（Autoregressive Model），最大的贡献在于使得模型获得了双向的上下文信息，但是会存在一些问题：
>
> 1. Pretrain-finetune Discrepancy: 预训练时的[MASK]在微调（fine-tuning）时并不会出现，使得两个过程不一致，这不利于 Learning。
> 2. Independence Assumption: 每个 token 的预测是相互独立的。而类似于 New York 这样的 Entity，New 和 York 是存在关联的，这个假设则忽略了这样的情况。

> 为了解决这些问题，XLNET引入了以下机制：
>
> 1. Permutation Language Model: XLNet 使用输入的 permutation 获取双向的上下文信息，同时维持自回归模型原有的单向形式。
> 2. Two-Stream Self-Attention: 该机制所要解决的问题是，当我们获得了

$g_\theta(x_{Z<t}, z_t)$ 后，我们只有该位置信息以及“上文”的信息，不足以去预测该位置后的 token；而原来的$h_\theta(x_{Z<t})$ 则因为获取不到位置信息，依然不足以去预测。因此，XLNet 引入了 Two-Stream Self-Attention 机制，将两者结合起来。

3. Recurrence Mechanism: 该机制来自 Transformer-XL，即在处理下一个 segment 时结合上个 segment 的 hidden representation，使得模型能够获得更长距离的上下文信息。

ALBERT

ALBERT主要对BERT做了3点改进，缩小了整体的参数量，加快了训练速度，增加了模型效果。

1. Factorized embedding parameterization:
2. Cross-layer parameter sharing: ALBERT通过跨层共享所有参数进一步提高了参数效率。这意味着前馈网络参数和注意力参数都是共享的。与BERT相比，ALBERT从一层到另一层的转换更平滑。
3. Sentence-order prediction: ALBERT利用NSP开发了SOP - 句子顺序预测，使用了两个句子，都来自同一个文档。正样本测试用例是这两句话的顺序是正确的，负样本是两个句子的顺序颠倒。这避免了主题预测的问题，并帮助ALBERT学习更细粒度的语篇或句子间衔接。

RoBERTa

1. 去掉下一句预测(NSP)任务
2. 动态掩码。BERT 依赖随机掩码和预测 token。原版的 BERT 实现在数据预处理期间执行一次掩码，得到一个静态掩码。 而 RoBERTa 使用了动态掩码：每次向模型输入一个序列时都会生成新的掩码模式。这样，在大量数据不断输入的过程中，模型会逐渐适应不同的掩码策略，学习不同的语言表征。
3. 文本编码。Byte-Pair Encoding（BPE）是字符级和词级别表征的混合，支持处理自然语言语料库中的众多常见词汇。原版的 BERT 实现使用字符级别的 BPE 词汇，大小为 30K，是在利用启发式分词规则对输入进行预处理之后学得的。Facebook 研究者没有采用这种方式，而是考虑用更大的 byte 级别 BPE 词汇表来训练 BERT，这一词汇表包含 50K 的 subword 单元，且没有对输入作任何额外的预处理或分词。

对比

XLNet vs Bert

- 双流自注意力

  它包含两种自注意力。一个是**content stream attention**，它是Transformer中的标准自注意力。另一个是**query stream attention**。XLNet引入它来替换BERT中的[MASK] token。

ERNIE vs BERT

- 实体和短语mask

  不同级别的mask（单字、实体、短语）替换bert字符级mask，

RoBERTa vs BERT

- 静态mask vs 动态mask

  原来Bert对每一个序列随机选择15%的Tokens替换成[MASK]，为了消除与下游任务的不匹配，还对这15%的Tokens进行
  （1）80%的时间替换成[MASK]；
  （2）10%的时间不变；
  （3）10%的时间替换成其他词。但整个训练过程，这15%的Tokens一旦被选择就不再改变，也就是说从一开始随机选择了这15%的Tokens，之后的N个epoch里都不再改变了。这就叫做静态Masking。

  而RoBERTa一开始把预训练的数据复制10份，每一份都随机选择15%的Tokens进行Masking，也就是说，同样的一句话有10种不同的mask方式。然后每份数据都训练N/10个epoch。这就相当于在这N个epoch的训练中，每个序列的被mask的tokens是会变化的。这就叫做动态Masking。

- NSP

  原本的Bert为了捕捉句子之间的关系，使用了NSP任务进行预训练，就是输入一对句子A和B，判断这两个句子是否是连续的。在训练的数据中，50%的B是A的下一个句子，50%的B是随机抽取的。

  而RoBERTa去除了NSP，而是每次输入连续的多个句子，直到最大长度512（可以跨文章）。这种训练方式叫做（FULL - SENTENCES），而原来的Bert每次只输入两个句子。

- 更大batch，更多训练数据

ALBert vs BERT

- Factorized Embedding Parameterization

  在BERT、XLNet、RoBERTa中，词表的embedding size(E)和transformer层的hidden size(H)都是相等的，这个选择有两方面缺点：

  i. 从建模角度来讲，wordpiece向量应该是不依赖于当前内容的(context-independent)，而transformer所学习到的表示应该是依赖内容的。所以把E和H分开可以更高效地利用参数，因为理论上存储了context信息的**H要远大于E**。
  ii. 从实践角度来讲，NLP任务中的vocab size本来就很大，如果E=H的话，模型参数量就容易很大，而且embedding在实际的训练中更新地也比较稀疏。

  因此作者使用了小一些的E(64、128、256、768)，训练一个独立于上下文的embedding(VxE)，之后计算时再投影到隐层的空间(乘上一个ExH的矩阵)，相当于做了一个因式分解。

- Cross-layer parameter sharing

  跨层参数共享，就是不管12层还是24层都只用一个transformer。

  分为三种：只共享attention相关参数；只共享FFN相关参数；共享所有参数。

- Sentence Order Prediction

  SOP预训练任务，关注于句子间的连贯性，而非句子间的匹配性。SOP正样本从原始语料中获得，负样本是原始语料的句子A和句子B交换顺序。举个例子说明NSP和SOP的区别，原始语料句子 A和B， NSP任务正样本是 AB，负样本是AC；SOP任务正样本是AB，负样本是BA。

2、ALBERT为什么用 SOP 任务替代BERT 中的 NSP 任务?

> NSP的作用和缺陷：在屏蔽语言模型（MLM）损失之外，BERT 使用了额外的损失，称为下一句预测（NSP）。NSP 是一个预测两段文本是否在原文本中连续出现的二元分类损失。NSP 是一种二进制分类损失，用于预测原始文本中是否有两个片段连续出现，如下所示：通过从训练语料库中获取连续片段来创建正样本；通过将来自不同文档的句段配对而创建负样本； 正样本和负样本均以相同的概率 (概率各自为0.5)采样。 NSP 目标旨在提高下游任务（例如自然语言推理）的性能，这些任务需要推理句子对之间的关系。 然而，随后的研究（包括 RoBERTa 等）发现NSP 的影响不可靠，因此决定消除它，这一决定由多项任务下游任务性能改善的结果支撑。
>
> ALBERT作者推测，与 MLM 相比，NSP 失效的主要原因是其缺乏任务难度。根据上

> 文叙述，NSP 在单个任务中融合了主题预测和连贯性预测。因为负样本是通过不同文档的句段进行构造的，NSP 与主题预测是相关的。但是，与连贯性预测相比，主题预测更容易学习，并且与使用 MLM 损失学习的内容重叠更多。通过 MLM 损失，模型就已能学习到一些主题相关的信息。而从 RoBERTa 等研究中可以发现，去除 NSP 对模型性能没有大的影响，说明 NSP 没有按预期学习到连贯性信息，或者说学习到的连贯性信息相比于主题信息不是决定性的。
>
> ALBERT作者强调，句间建模是语言理解的一个重要方面，作者提出了一个主要基于连贯性的损失。对于 ALBERT，作者使用句子顺序预测（SOP）损失，它避免了主题预测，而侧重于建模句子间的连贯性。 SOP 损失使用与 BERT（同一文档中的两个连续段）相同的技术作为正样本，而负样本使用相同的两个连续段，但顺序互换。这迫使模型学习关于句子级连贯性的细粒度区别。作者在后续实验中证明，NSP 根本无法解决 SOP 任务（即最终学习到更容易学习的主题预测信息，并在 SOP 任务上表现为随机基线水平），而 SOP 可以在一定程度上解决 NSP 任务，大概是基于分析错位的关系线索。结果，ALBERT 模型一致提高了多语句编码任务的各项下游任务性能。