后BERT时代：15个预训练模型对比分析与关键点探究

JayLou 夕小瑶的卖萌屋 2019-08-17

前言

在小夕之前写过的《NLP的游戏规则从此改写?从word2vec, ELMo到BERT》 一文中，介绍了从word2vec到ELMo再到BERT的发展路径。而在BERT出现之后的这大半年的时间里，模型预训练的方法又被Google、Facebook、微软、百度、OpenAI等极少数几个玩得起游戏的核心玩家反复迭代了若干版，一次次的刷新我们这些吃瓜群众的案板上的瓜。

有没有感觉出瓜速度太快以至于吃速跟不上？不用担心，小夕帮你们找来了这篇总结的恰到好处的文章，对ELMo以来的15个代表性的预训练语言模型进行了多维度的对比和分析。尤其是近期找工作的小伙伴们注意啦，这篇文章里面提出来的一些问题很适合作为面试考点（划掉，交流点）噢~

> 本文转载自知乎专栏《高能NLP之路》并进行了剪切和少量修改，作者JayLou，原文链接
> https://zhuanlan.zhihu.com/p/76912493

首先上一张镇楼专用图，看一下ELMo以来的预训练语言模型发展的概况

模型	语言模型	特征抽取	上下文表征	最大亮点
ELMO	BiLM	BiLSTM	单向	2个单向语言模型拼接；
ULMFiT	LM	AWD-LSTM	单向	引入逐层解冻解决finetune中的灾难性问题；
SiATL	LM	LSTM	单向	引入逐层解冻+辅助LM解决finetune中的灾难性问题；
GPT1.0	LM	Transformer	单向	统一下游任务框架，验证Transformer在LM中的强大；
GPT2.0	LM	Transformer	单向	没有特定模型的精调流程，生成任务取得很好效果；
BERT	MLM	Transformer	双向	MLM获取上下文相关的双向特征表示；
MASS	LM+MLM	Transformer	单向/双向	改进BERT生成任务：统一为类似Seq2Seq的预训练框架；
UNILM	LM+MLM+S2SLM	Transformer	单向/双向	改进BERT生成任务：直接从mask矩阵的角度出发；
ENRIE1.0	MLM(BPE)	Transformer	双向	引入知识：3种[MASK]策略(BPE)预测短语和实体；
ENRIE	MLM+DEA	Transformer	双向	引入知识：将实体向量与文本表示融合；
MTDNN	MLM	Transformer	双向	引入多任务学习：在下游阶段；
ENRIE2.0	MLM+Multi-Task	Transformer	双向	引入多任务学习：在预训练阶段，连续增量学习；
SpanBERT	MLM+SPO	Transformer	双向	不需要按照边界信息进行mask；
RoBERTa	MLM	Transformer	双向	精细调参，舍弃NSP；
XLNet	PLM	Transformer-XL	双向	排列语言模型+双注意力流+Transformer

然后上本文正餐，一个高能的question list，这也是本文写作的主线。

Question List

- Q1：从不同维度对比各【预训练语言模型】？
- Q2：基于深度学习的NLP特征抽取机制有哪些？各有哪些优缺点？
- Q3：自回归和自编码语言模型各有什么优缺点？
- Q4：单向模型的内核机制是怎样的？有哪些缺点？
- Q5：Transformer内部机制的深入理解：
  - 为什么是缩放点积，而不是点积模型？

    - 相较于加性模型，点积模型具备哪些优点？
    - 多头机制为什么有效？
- Q6-Q10：BERT内核机制探究
    - BERT为什么如此有效？
    - BERT存在哪些优缺点？
    - BERT擅长处理哪些下游NLP任务？
    - BERT基于“字输入”还是“词输入”好？（对于中文任务）
    - BERT为什么不适用于自然语言生成任务（NLG）？
- Q11-Q15：针对BERT原生模型的缺点，后续的BERT系列模型是：
    - 如何改进【生成任务】的？
    - 如何引入【知识】的？
    - 如何引入【多任务学习机制】的？
    - 如何改进【mask策略】的？
    - 如何进行【精细调参】的？
- Q16：XLNet提出的背景是怎样的？
- Q17：XLNet为何如此有效：
    - 为什么PLM可以实现双向上下文的建模？
    - 怎么解决没有目标(target)位置信息的问题？
- Q18：Transformer-XL怎么实现对长文本建模？

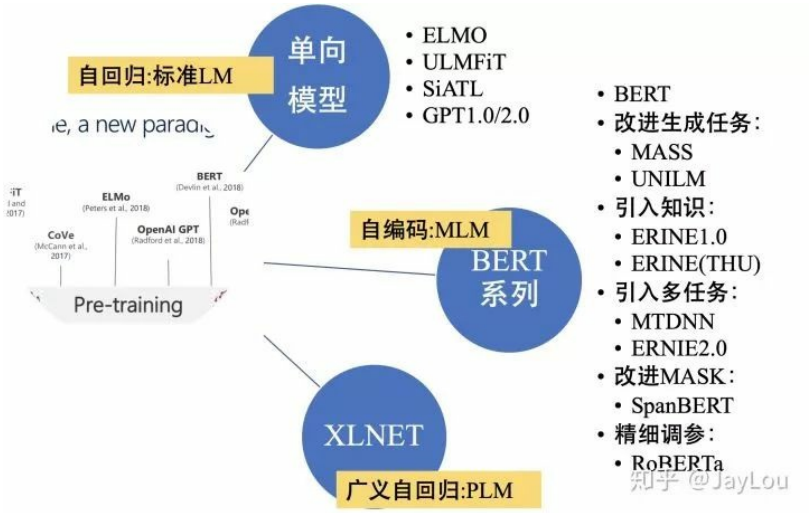
下面本文将从以下几个方面来对上述问题一一探讨

> 一. 不同视角下的预训练语言模型对比
> 二.预训练语言模型的基础：特征抽取机制+语言模型的分类
> 三.单向模型回顾+内核机制探究
> 四.BERT的内核机制探究
> 五.BERT系列模型进展介绍
> 六.XLNET的内核机制探究
> 七.预训练语言模型的未来

一、不同视角下的预训练语言模型对比

Q1：从不同维度对比【预训练语言模型】

从特征抽取、预训练语言模型目标、BERT系列模型的改进方向、特征表示4个视角，对比预训练语言模型：



- **不同的特征抽取机制**
  - RNNs：ELMO/ULMFiT/SiATL；
  - Transformer：GPT1.0/GPT2.0/BERT系列模型；
  - Transformer-XL：XLNet；
- **不同的预训练语言目标**
  - 自编码（AutoEncode）：BERT系列模型；
  - 自回归（AutoRegression）：单向模型（ELMO / ULMFiT / SiATL / GPT1.0 / GPT2.0）和XLNet；
- **BERT系列模型的改进**
  - 引入常识：ERNIE1.0 / ERNIE(THU) / ERNIE2.0（简称为“ERNIE系列”）；
  - 引入多任务学习：MTDNN/ERNIE2.0；
  - 基于生成任务的改进：MASS/UNILM；
  - 不同的mask策略：WWM/ERNIE系列/SpanBERT；
  - 精细调参：RoBERTa；
- **特征表示（是否能表示上下文）**
  - 单向特征表示：单向模型（ELMO/ULMFiT/SiATL/GPT1.0/GPT2.0）；
  - 双向特征表示：BERT系列模型+XLNet；

二、预训练语言模型的基础：特征抽取机制+语言模型的分类

Q2：基于深度学习的NLP特征抽取机制有哪些？各有哪些优缺点？

1）能否处理长距离依赖问题

长距离依赖建模能力：Transformer-XL > Transformer > RNNs > CNNs

- MLP：不考虑序列（位置）信息，不能处理变长序列，如NNLM和word2vec；
- CNNs：考虑序列（位置）信息，不能处理长距离依赖，聚焦于n-gram提取，pooling操作会导致序列（位置）信息丢失；
- RNNs：天然适合处理序列（位置）信息，但仍不能处理长距离依赖（由于BPTT导致的梯度消失等问题），故又称之为“较长的短期记忆单元(LSTM)”；
- Transformer/Transformer-XL：self-attention解决长距离依赖，无位置偏差；

2）前馈/循环网络 or 串行/并行计算

- MLP/CNNs/Transformer：前馈/并行
- RNNs/ Transformer-XL：循环/串行：

3）计算时间复杂度（序列长度n，embedding size为d，filter大小k）

- CNNs：

- RNNs：

- Self Attention：

Q3：自回归和自编码语言模型各有什么优缺点?

1）自回归语言模型

$$\max_{\theta} \quad \log p_\theta(\mathbf{x}) = \sum_{t=1}^{T} \log p_\theta(x_t \mid \mathbf{x}_{<t}) = \sum_{t=1}^{T} \log \frac{\exp\left(h_\theta(\mathbf{x}_{1:t-1})^\top e(x_t)\right)}{\sum_{x'} \exp\left(h_\theta(\mathbf{x}_{1:t-1})^\top e(x')\right)},$$

- **优点：**
  - 文本序列**联合概率的密度估计**，即为传统的语言模型，天然适合处理自然生成任务；

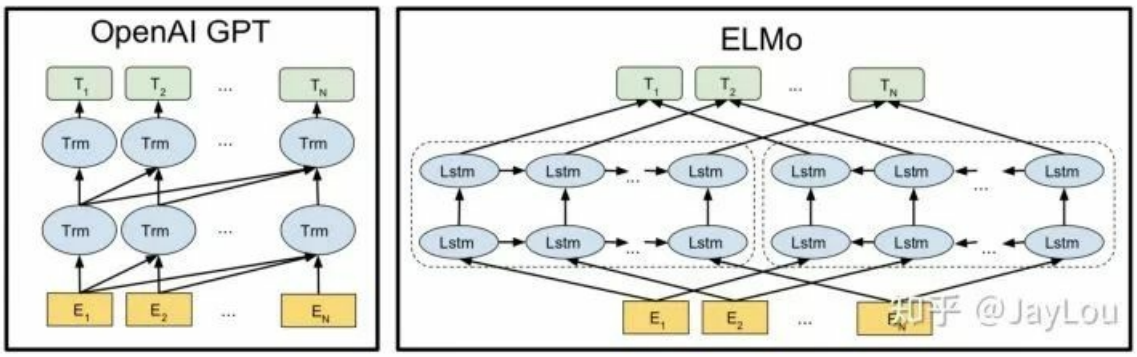
- **缺点：**
  - 联合概率按照文本序列从左至右分解（**顺序拆解**），无法通过上下文信息进行双向特征表征；
- **代表模型：**ELMO/GPT1.0/GPT2.0；
- **改进：**XLNet将传统的自回归语言模型进行推广，将顺序拆解变为**随机拆解**（排列语言模型），产生上下文相关的双向特征表示；

2）自编码语言模型

$$\max_{\theta} \quad \log p_\theta(\bar{\mathbf{x}} \mid \hat{\mathbf{x}}) \approx \sum_{t=1}^{T} m_t \log p_\theta(x_t \mid \hat{\mathbf{x}}) = \sum_{t=1}^{T} m_t \log \frac{\exp\left(H_\theta(\hat{\mathbf{x}})_t^\top e(x_t)\right)}{\sum_{x'} \exp\left(H_\theta(\hat{\mathbf{x}})_t^\top e(x')\right)},$$

- **优点：**本质为降噪自编码特征表示，通过引入噪声[MASK]构建MLM，获取上下文相关的双向特征表示；
- **缺点：**引入独立性假设，为**联合概率的有偏估计**，没有考虑预测[MASK]之间的相关性
  - 不适合直接处理生成任务，MLM预训练目标的设置造成预训练过程和生成过程不一致；
  - 预训练时的[MASK]噪声在finetune阶段不会出现，造成两阶段不匹配问题；
- **代表模型：**BERT系列模型；

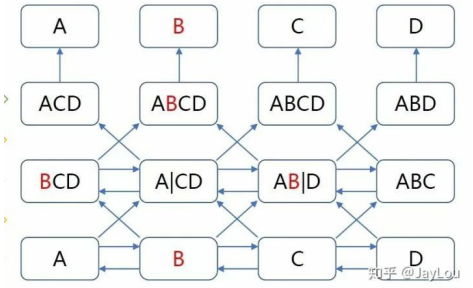
三、单向模型回顾+内核机制探究



Q4：单向模型的内核机制是怎样的？有哪些缺点？

1）ELMo (Allen Institute)[6]

- **要点：**
  - 引入双向语言模型，其实是2个单向语言模型（前向和后向）的集成；
  - 通过保存预训练好的2层biLSTM，通过特征集成或finetune应用于下游任务；
- **缺点：**
  - 本质上为自回归语言模型，只能获取单向的特征表示，不能同时获取上下文表示；
  - LSTM不能解决长距离依赖。
- **为什么不能用biLSTM构建双向语言模型？**
  - 不能采取2层biLSTM同时进行特征抽取构建双向语言模型，否则会出现**标签泄漏**的问题；因此ELMO前向和后向的LSTM参数独立，共享词向量，独立构建语言模型；



2）ULMFiT (fast.ai) / SiATL

2.1）ULMFiT[7]要点：

- 三阶段训练：LM预训练+精调特定任务LM+精调特定分类任务；
- 特征抽取：3层AWD-LSTM；
- 精调特定分类任务：逐层解冻；

2.2) SiATL[8]要点：

- 二阶段训练：LM预训练+特定任务精调分类任务（引入LM作为辅助目标，辅助目标对于小数据有用，与GPT相反）；
  - 特征抽取：LSTM+self-attention；
- 精调特定分类任务：逐层解冻；
  - 都通过一些技巧解决finetune过程中的灾难性遗忘问题：如果预训练用的无监督数据和任务数据所在领域不同，逐层解冻带来的效果更明显[9]；

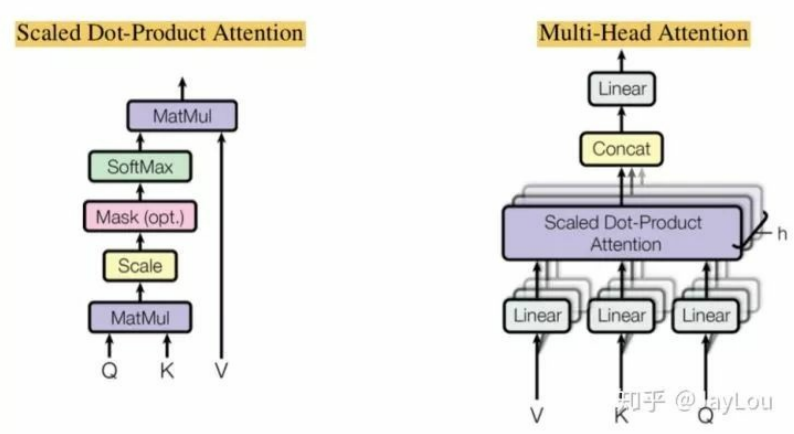
3）GPT1.0 / GPT2.0 (OpenAI)

- **GPT1.0[10]要点：**
  - 采用Transformer进行特征抽取，首次将Transformer应用于预训练语言模型；
  - finetune阶段引入语言模型辅助目标（辅助目标对于大数据集有用，小数据反而有所下降，与SiATL相反），解决finetune过程中的灾难性遗忘；
  - 预训练和finetune一致，统一二阶段框架；
- **GPT2.0[11]要点：**
  - 没有针对特定模型的精调流程：GPT2.0认为预训练中已包含很多特定任务所需的信息。
  - 生成任务取得很好效果，使用覆盖更广、质量更高的数据；
- **缺点：**
  - 依然为单向自回归语言模型，无法获取上下文相关的特征表示；

四、BERT内核机制探究

这一部分对BERT的内核机制进行介绍，在回答“BERT为什么如此有效？”之前，首先介绍Transformer的内核机制。

Q5：Transformer[12]内部机制的深入理解（回顾）



1）Multi-Head Attention和Scaled Dot-Product Attention

本质是self attention通过attention mask动态编码变长序列，解决长距离依赖、无位置偏差、可并行计算

- **为什么是缩放点积，而不是点积模型?**
  - 当输入信息的维度 d 比较高，点积模型的值通常有比较大方差，从而导致 softmax 函数的梯度会比较小。因此，缩放点积模型可以较好地解决这一问题。
- **为什么是双线性点积模型（经过线性变换Q**

K）？

- 双线性点积模型，引入非对称性，更具健壮性（Attention mask对角元素值不一定是最大的，也就是说当前位置对自身的注意力得分不一定最高）。

- **相较于加性模型，点积模型具备哪些优点？**
  - 常用的Attention机制为**加性模型**和**点积模型**，理论上加性模型和点积模型的复杂度差不多，但是点积模型在实现上可以更好地利用矩阵乘积，从而计算效率更高（实际上，随着维度d的增大，加性模型会明显好于点积模型）。

- **多头机制为什么有效?**
  - 类似于CNN中通过多通道机制进行特征选择;
  - Transformer中先通过切头(spilt)再分别进行Scaled Dot-Product Attention,可以使进行点积计算的维度d不大(防止梯度消失），同时缩小attention mask矩阵。

2）Position-wise Feed-Forward Networks

- FFN 将每个位置的Multi-Head Attention结果映射到一个更大维度的特征空间,然后使用ReLU引入非线性进行筛选,最后恢复回原始维度。
- Transformer在抛弃了 LSTM 结构后，FFN 中的 ReLU成为了一个主要的提供非线性变换的单元。

3）Positional Encoding

将Positional Embedding改为Positional Encoding,主要的区别在于Positional Encoding是用公式表达的、不可学习的,而Positional Embedding是可学习的(如BERT),两种方案的训练速度和模型精度差异不大;但是Positional Embedding位置编码范围是固定的,而Positional Encoding编码范围是不受限制的。

- 为什么引入

和

建模Positional Encoding?

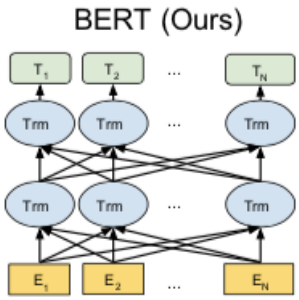
- 引入

和

是为了使模型实现对相对位置的学习，两个位置 pos 和 pos+k 的位置编码是固定间距k的线性变化：

- 可以证明：**间隔为k的任意两个位置编码的欧式空间距离是恒等的，只与k有关。**

Q6：BERT[13]为什么如此有效?



- 引入Masked Language Model(MLM)预训练目标，能够获取上下文相关的双向特征表示；
- 引入Next Sentence Prediction(NSP)预训练目标，擅长处理句子或段落的匹配任务；
- 引入强大的特征抽取机制Transformer(**多种机制并存**)：
  - Multi-Head self attention：多头机制类似于“多通道”特征抽取，self attention通过attention mask动态编码变长序列，解决长距离依赖（无位置偏差）、可并行计算；
  - Feed-forward ：在位置维度计算非线性层级特征；
  - Layer Norm & Residuals：加速训练，使“深度”网络更加健壮；
- 引入大规模、高质量的文本数据；

Q7：BERT存在哪些优缺点?

- 优点：能够获取上下文相关的双向特征表示；
- 缺点：
  - 生成任务表现不佳：预训练过程和生成过程的不一致，导致在生成任务上效果不佳；
  - 采取独立性假设：没有考虑预测[MASK]之间的相关性，是对语言模型联合概率的有偏估计（不是密度估计）；
  - 输入噪声[MASK]，造成预训练-精调两阶段之间的差异；
  - 无法文档级别的NLP任务，只适合于句子和段落级别的任务；

Q8：BERT擅长处理哪些下游NLP任务[14]?

1. 适合句子和段落级别的任务，不适用于文档级别的任务；
2. 适合处理高层语义信息提取的任务，对浅层语义信息提取的任务的提升效果不大（如一些简单的文本分类任务）；
3. 适合处理句子/段落的匹配任务；因此，在一些任务中可以构造辅助句（类似匹配任务）实现效果提升（如关系抽取/情感挖掘等任务）；
4. 不适合处理NLG任务；

Q9：BERT基于“字输入”还是“词输入”好？（对于中文任务）

1. 如果基于“词输入”，会加剧OOV问题，会增大输入空间，需要利用大得多的语料去学习输入空间到标签空间的函数映射。
2. 随着Transfomer特征抽取能力，分词不再成为必要，词级别的特征学习可以纳入为内部特征进行表示学习。

Q10：BERT为什么不适用于自然语言生成任务（NLG）？

1. 由于BERT本身在预训练过程和生成过程的不一致，并没有做生成任务的相应机制，导致在生成任务上效果不佳，不能直接应用于生成任务。
2. 如果将BERT或者GPT用于Seq2Seq的自然语言生成任务，可以分别进行预训练编码器和解码器，但是编码器-注意力-解码器结构没有被联合训练，BERT和GPT在条件生成任务中只是次优效果。

五、BERT系列模型进展介绍

这一部分介绍一些模型，它们均是对BERT原生模型在一些方向的改进。

Q11：针对BERT原生模型，后续的BERT系列模型是如何改进【生成任务】的?

1）MASS(微软)[15]

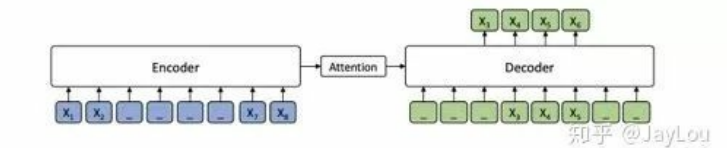
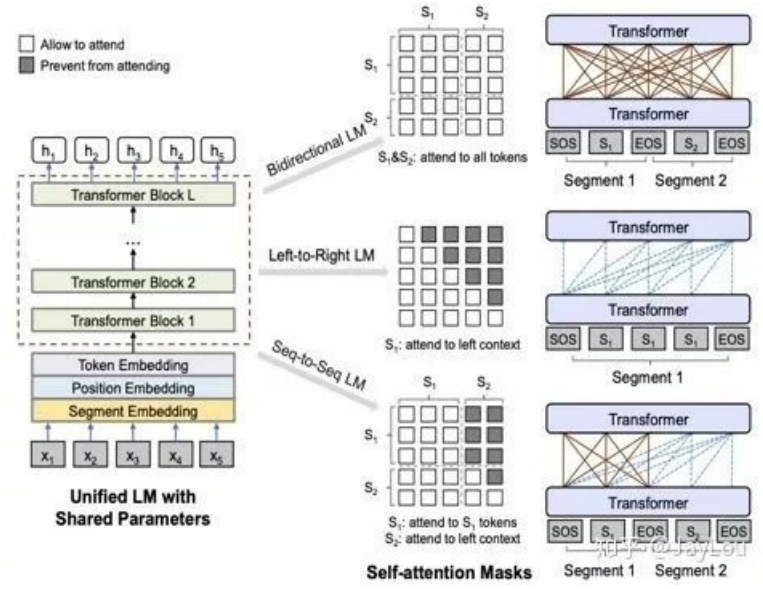


Figure 1. The encoder-decoder framework for our proposed MASS. The token "_" represents the mask symbol [M].

- 统一预训练框架:通过类似的Seq2Seq框架，在预训练阶段统一了BERT和LM模型；
- Encoder中理解unmasked tokens；Decoder中需要预测连续的[mask]tokens，获取更多的语言信息；Decoder从Encoder中抽取更多信息；
- 当k=1或者n时，MASS的概率形式分别和BERT中的MLM以及GPT中标准的LM一致（k为mask的连续片段长度））

2）UNILM (微软)[16]



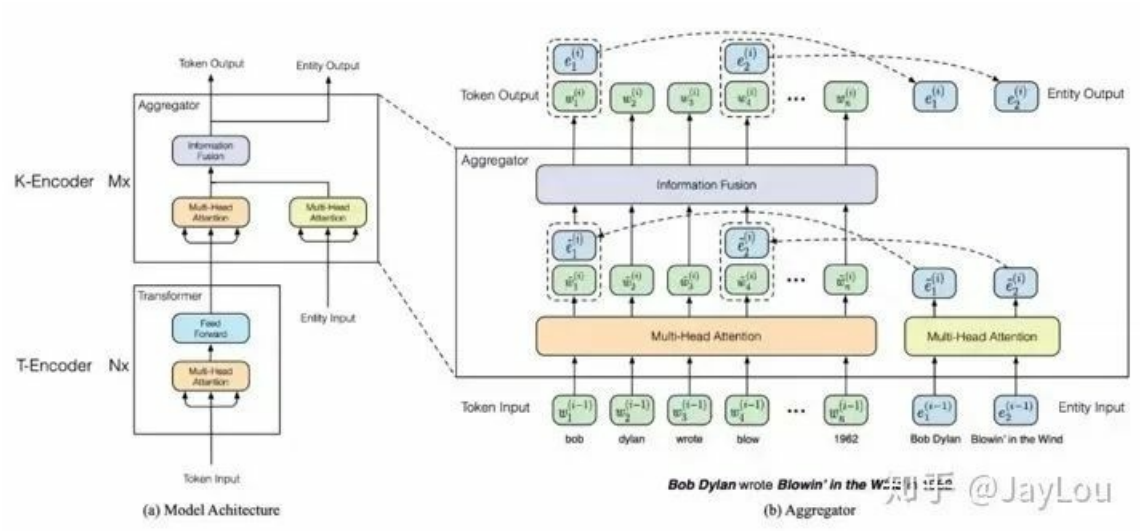
- 统一预训练框架:和直接从mask矩阵的角度统一BERT和LM；
- 3个Attention Mask矩阵：LM、MLM、Seq2Seq LM；
- 注意：UNILM中的LM并不是传统的LM模型，仍然是通过引入[MASK]实现的；

Q12：针对BERT原生模型，后续的BERT系列模型是如何引入【知识】的?

1）ERNIE 1.0 (百度)[17]

- 在预训练阶段引入知识（实际是预先识别出的实体），引入3种[MASK]策略预测：
  - Basic-Level Masking：跟BERT一样，对subword进行mask，无法获取高层次语义；
  - Phrase-Level Masking：mask连续短语；
  - Entity-Level Masking：mask实体；

2）ERNIE (THU)[18]

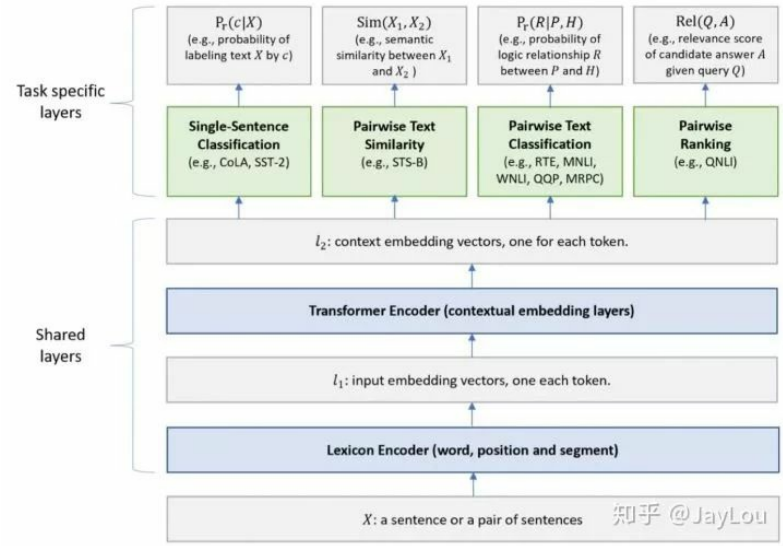


- 基于BERT预训练原生模型，将文本中的实体**对齐**到外部的知识图谱，并通过知识嵌入得到实体向量作为ERNIE的输入；
- 由于语言表征的预训练过程和知识表征过程有很大的不同，会产生两个独立的向量空间。为解决上述问题，在有实体输入的位置，将实体向量和文本表示通过**非线性变换进行融合**，以融合词汇、句法和知识信息；
- 引入改进的预训练目标 **Denoising entity auto-encoder** (DEA)：要求模型能够根据给定的实体序列和文本序列来预测对应的实体；

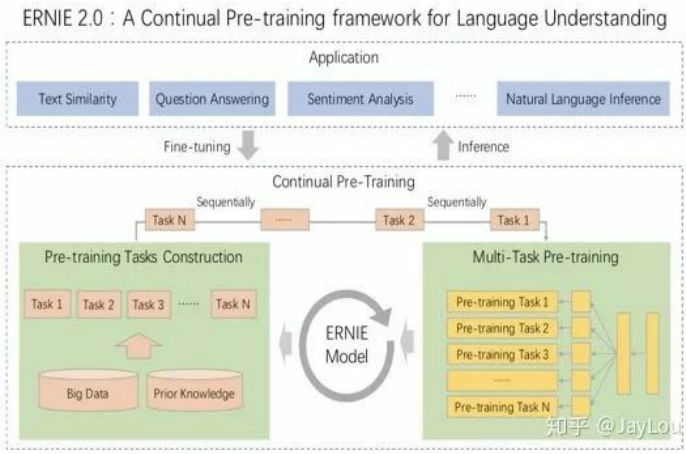
Q13：针对BERT原生模型，后续的BERT系列模型是如何引入【多任务学习机制】的?

多任务学习(Multi-task Learning)[19]是指同时学习多个相关任务，让这些任务在学习过程中共享知识，利用多个任务之间的相关性来改进模型在每个任务的性能和泛化能力。多任务学习可以看作是一种归纳迁移学习，即通过利用包含在相关任务中的信息作为归纳偏置(Inductive Bias)来提高泛化能力。多任务学习的训练机制分为同时训练和交替训练。

1）MTDNN(微软)[20]：在下游任务中引入多任务学习机制



2）ERNIE 2.0 (百度)[21]：在预训练阶段引入多任务学习



- MTDNN是在下游任务引入多任务机制的，而ERNIE 2.0 是在预训练引入多任务学习（与先验知识库进行交互），使模型能够从不同的任务中学到更多的语言知识。
- 主要包含3个方面的任务：
  - word-aware 任务：捕捉词汇层面的信息；
  - structure-aware 任务：捕捉句法层面的信息；
  - semantic-aware 任务：捕捉语义方面的信息；
- 主要的方式是构建**增量学习**（后续可以不断引入更多的任务）模型，通过多任务学习**持续更新预训练模型**，这种**连续交替**的学习范式不会使模型忘记之前学到的语言知识。
  - 将3大类任务的若干个子任务一起用于训练，引入新的任务时会将继续引入之前的任务，防止忘记之前已经学到的知识，具体是一个**逐渐增加任务数量**的过程[22]：

    (task1)->(task1,task2)->(task1,task2,task3)->…->(task1，task2,…,taskN)，

Q14：针对BERT原生模型，后续的BERT系列模型是如何改进【mask策略】的?

- 原生BERT模型：按照subword维度进行mask，然后进行预测；
- BERT WWM(Google)：按照whole word维度进行mask，然后进行预测；

- ERNIE等系列：引入外部知识，按照entity维度进行mask，然后进行预测；
- SpanBert：不需要按照先验的词/实体/短语等边界信息进行mask，而是采取随机mask：
  - **采用Span Masking**：根据几何分布，随机选择一段空间长度，之后再根据均匀分布随机选择起始位置，最后按照长度mask；通过采样，平均被遮盖长度是3.8 个词的长度；
  - **引入Span Boundary Objective**:新的预训练目标旨在使被mask的Span 边界的词向量能学习到 Span中被mask的部分;新的预训练目标和MLM一起使用；
- 注意:BERT WWM、ERNIE等系列、SpanBERT旨在 **隐式地学习**预测词(mask部分本身的强相关性)之间的关系[23],而在 XLNet中，是通过 PLM 加上自回归方式来**显式地学习**预测词之间关系；

Q15：针对BERT原生模型，后续的BERT系列模型是如何进行【精细调参】的?

RoBERTa(FaceBook)[24]

- 丢弃NSP，效果更好；
- 动态改变mask策略，把数据复制10份，然后统一进行随机mask；
- 对学习率的峰值和warm-up更新步数作出调整；
- 在更长的序列上训练：不对序列进行截短，使用全长度序列；

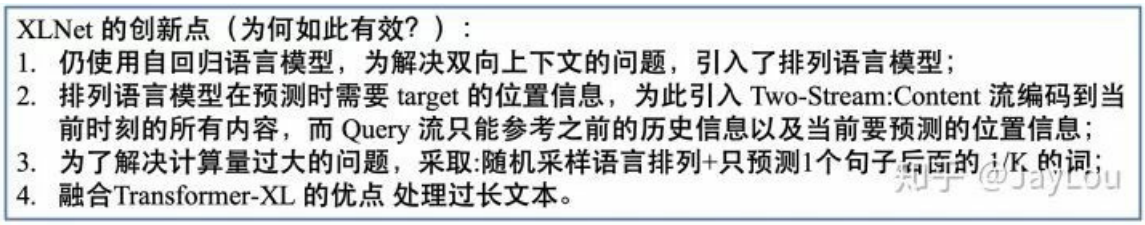
六、XLNet的内核机制探究

在BERT系列模型后，Google发布的XLNet在问答、文本分类、自然语言理解等任务上都大幅超越BERT；XLNet的提出是对标准语言模型（自回归）的一个复兴[25]，提出一个框架来连接语言建模方法和预训练方法。

Q16：XLNet[26]提出的背景是怎样的?

- 对于ELMO、GPT等预训练模型都是基于传统的语言模型（自回归语言模型AR），自回归语言模型天然适合处理生成任务，但是无法对双向上下文进行表征，因此人们反而转向自编码思想的研究（如BERT系列模型）；
- 自编码语言模型（AE）虽然可以实现双向上下文进行表征，但是：
  - BERT系列模型引入独立性假设，没有考虑预测[MASK]之间的相关性；
  - MLM预训练目标的设置造成预训练过程和生成过程不一致；
  - 预训练时的[MASK]噪声在finetune阶段不会出现，造成两阶段不匹配问题；
- 有什么办法能构建一个模型使得同时具有AR和AE的优点并且没有它们缺点呢?

Q17：XLNet为何如此有效：内核机制分析



1）排列语言模型（Permutation LM，PLM）

如果衡量序列中被建模的依赖关系的数量,标准的LM可以达到上界,不像MLM一样,LM不依赖于任何独立假设。借鉴 NADE[27]的思想，XLNet将标准的LM推广到PLM。

- 为什么PLM可以实现双向上下文的建模?
  - **PLM的本质就是LM联合概率的多种分解机制的体现；**

- 将LM的顺序拆解推广到**随机拆解，但是需要保留每个词的原始位置信息（PLM只是语言模型建模方式的因式分解/排列，并不是词的位置信息的重新排列！）**
- 如果遍历 $T!$ 种分解方法，并且模型参数是共享的， **PLM就一定可以学习到各种双向上下文；**换句话说，当我们把所有可能的$T!$ 排列都考虑到的时候，对于预测词的所有上下文就都可以学习到了！
- 由于遍历 $T!$ 种路径计算量非常大（对于10个词的句子，10!=3628800）。因此实际只能随机的采样$T!$里的部分排列，并求期望；

$$\max_{\theta} \quad \mathbb{E}_{\mathbf{z}\sim\mathcal{Z}_T}\left[\sum_{t=1}^{T}\log p_\theta(x_{z_t}\mid \mathbf{x}_{\mathbf{z}<t})\right].$$

2）Two-Stream Self-Attention

如果采取标准的Transformer来建模PLM，会出现**没有目标(target)位置信息的问题**。问题的关键是模型并不知道要预测的到底是哪个位置的词，从而导致**具有部分排列下的PLM在预测不同目标词时的概率是相同的**。

XLNet内核机制2: Two-Stream Self-Attention

对于标准的Transformer建模PLM 时：

$$p_\theta(X_{z_t}=x|\mathbf{x}_{z<t})=\frac{exp(e(x)^T h_\theta(\mathbf{x}_{z<t}))}{\sum_{x'} exp(e(x')^T h_\theta(\mathbf{x}_{z<t}))}$$

具有部分排列下的PLM在预测不同目标词时的概率是相同的：

$$\underbrace{p_\theta(X_i=x\mid \mathbf{x}_{\mathbf{z}<t})}=\underbrace{p_\theta(X_j=x\mid \mathbf{x}_{\mathbf{z}<t})}=\frac{\exp\left(e(x)^\top h(\mathbf{x}_{\mathbf{z}<t})\right)}{\sum_{x'}\exp\left(e(x')^\top h(\mathbf{x}_{\mathbf{z}<t})\right)},$$

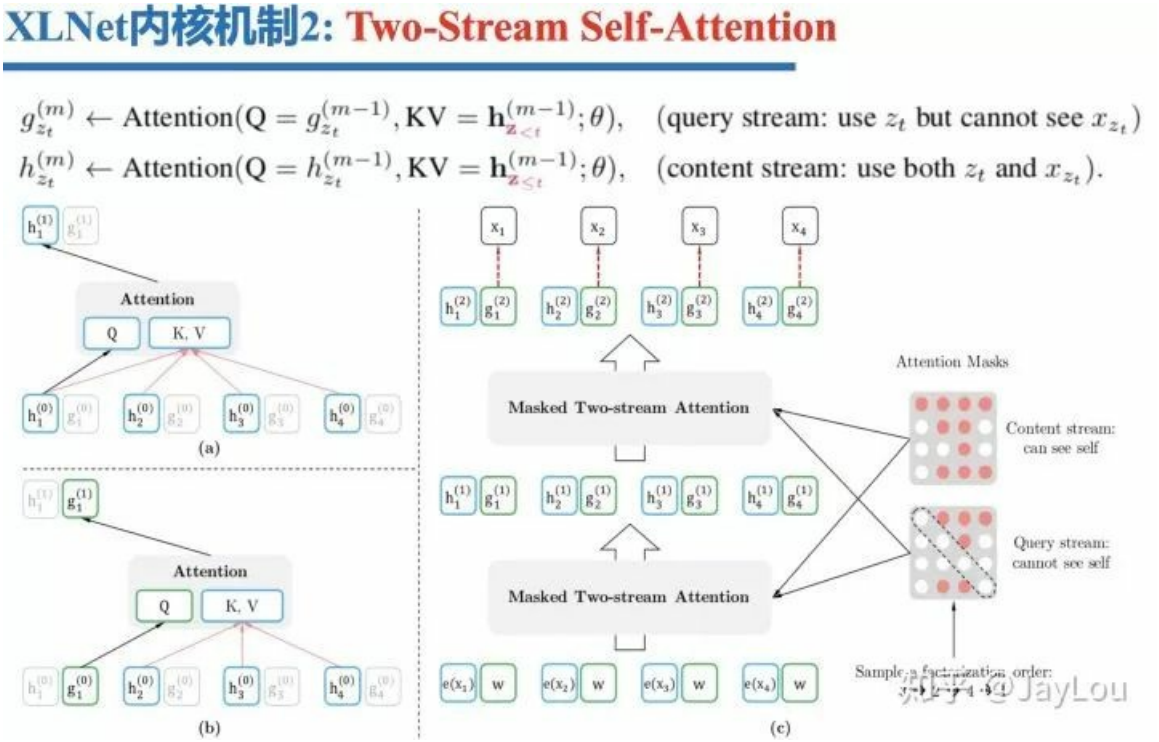
$$\mathbf{z}^{(1)}_{<t}=\mathbf{z}^{(2)}_{<t}=\mathbf{z}_{<t}\quad \text{but}\quad z^{(1)}_t=i\neq j=z^{(2)}_t.$$

注意⚠️：
假设输入的句子是 [我爱生活]，有两种排列为 z=[1, 3, 4, 2]和z'=[1,3,2,4]时，
P(活 | 我生) == P(爱 | 我生)

$$p_\theta(X_{z_3}=x|x_{z_1z_2})=p_\theta(X_4=x|x_1x_3)=\frac{exp(e(x)^T h_\theta(x_1x_3))}{\sum_{x'} exp(e(x')^T h_\theta(x_1x_3))}$$

$$p_\theta(X_{z_3}=x|x_{z_1z_2})=p_\theta(X_2=x|x_1x_3)=\frac{exp(e(x)^T h_\theta(x_1x_3))}{\sum_{x'} exp(e(x')^T h_\theta(x_1x_3))}$$

- 怎么解决没有目标(target)位置信息的问题？
  - 对于没有目标位置信息的问题，XLNet 引入了Two-Stream Self-Attention：



- Query 流就为了预测当前词，只包含位置信息，不包含词的内容信息；
- Content 流主要为 Query 流提供其它词的内容向量，包含位置信息和内容信息；

3） 融入Transformer-XL的优点（具体见Q18）

Q18：Transformer-XL[28]怎么实现对长文本建模?

- BERT(Transformer)的最大输入长度为512，那么怎么对文档级别的文本建模?
  - vanilla model进行Segment，但是会存在上下文碎片化的问题（无法对连续文档的语义信息进行建模），同时推断时需要重复计算，因此推断速度会很慢；
- Transformer-XL改进
  - 对于每一个segment都应该具有不同的位置编码，因此Transformer-XL采取了相对位置编码；

$$\mathbf{A}_{i,j}^{\text{abs}} = \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{E}_{x_j}}_{(a)} + \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{U}_j}_{(b)} + \underbrace{\mathbf{U}_i^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{E}_{x_j}}_{(c)} + \underbrace{\mathbf{U}_i^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{U}_j}_{(d)}.$$

$$\mathbf{A}_{i,j}^{\text{rel}} = \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_{k,E} \mathbf{E}_{x_j}}_{(a)} + \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_{k,R} \mathbf{R}_{i-j}}_{(b)} + \underbrace{u^\top \mathbf{W}_{k,E} \mathbf{E}_{x_j}}_{(c)} + \underbrace{v^\top \mathbf{W}_{k,R} \mathbf{R}_{i-j}}_{(d)}.$$

  - 前一个segment计算的representation被修复并缓存，以便在模型处理下一个新的segment时作为扩展上下文resume；
  - 最大可能依赖关系长度增加了N倍，其中N表示网络的深度；
  - 解决了上下文碎片问题，为新段前面的token提供了必要的上下文；
  - 由于不需要重复计算，Transformer-XL在语言建模任务的评估期间比vanilla Transformer快1800+倍；
  - 引入**recurrence mechanism**(不采用BPTT方式求导)：
  - **引入相对位置编码方案：**

七、预训练语言模型的未来

上述的【预训练语言模型】主要从2大方面进行介绍：一是总的对比；二是分别介绍单向语言模型、BERT系列模型、XLNet模型。

可以看出，未来【预训练语言模型】更多的探索方向主要为[25]：

- 复兴语言模型：进一步改进语言模型目标，不断突破模型的上界；
- 大数据、大算力：将大数据、大算力推到极致；
- 更快的推断：轻量级模型是否有可能达到SOTA效果？
- 引入更丰富的知识信息，更精细的调参，更有价值的MASK策略；
- 统一条件生成任务框架，如基于XLNet统一编码和解码任务，同时可考虑更快的解码方式；

参考文献

[1] NLP将迎来黄金十年 https://www.msra.cn/zh-cn/news/executivebylines/tech-bylines-nlp
[2] a review of the recent history of nlp
[3] AIS：ACL2019进展报告
[4] ACL 主席周明：一起拥抱 ACL 和 NLP 的光明未来
[5] 自然语言处理中的语言模型预训练方法 https://www.jiqizhixin.com/articles/2018-10-22-3
[6] ELMO:Deep contextualized word representations
[7] ULMFiT：Universal Language Model Fine-tuning)
[8] SiATL：An Embarrassingly Simple Approach for Transfer Learning from Pretrained Language Models
[9] BERT时代与后时代的NLP https://zhuanlan.zhihu.com/p/66676144
[10] GPT:Improving Language Understanding by Generative Pre-Training
[11] GPT2.0:Language Models are Unsupervised Multitask Learners
[12] Transformer:Attention is all you need
[13] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
[14] Bert时代的创新（应用篇）：Bert在NLP各领域的应用进展 https://zhuanlan.zhihu.com/p/68446772
[15] MASS: Masked Sequence to Sequence Pre-training for Language Generation
[16] UNILM：Unified Language Model Pre-training for Natural Language Understanding and Generation
[17] ERNIE: Enhanced Representation through Knowledge Integration
[18] ERNIE: Enhanced Language Representation with Information Entities
[19] nndl：神经网络与深度学习
[20] MT-DNN：Multi-Task Deep Neural Net for NLU
[21] ERNIE 2.0: A CONTINUAL PRE-TRAINING FRAMEWORK FOR LANGUAGE UNDERSTANDING
[22] 陈凯：https://www.zhihu.com/question/337827682/answer/768908184
[23] SpanBert：对 Bert 预训练的一次深度探索
[24] RoBERTa: A Robustly Optimized BERT Pretraining Approach
[25] ab他们创造了横扫NLP的XLNet：专访CMU博士杨植麟
[26] XLnet: Generalized Autoregressive Pretraining for Language Understanding
[27] Neural autoregressive distribution estimation
[28] Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context

---