题目：

BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension

收录会议：ACL 2020

摘要

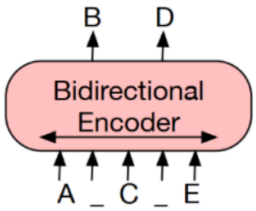
文章提出一个预训练sequence-to-sequence的去噪自编码器：BART。BART的训练主要由2个步骤组成：(1)使用任意噪声函数破坏文本(2）模型学习重建原始文本。BART 使用基于 Transformer 的标准神经机器翻译架构，可视为BERT(双向编码器)、GPT(从左至右的解码器)等近期出现的预训练模型的泛化形式。文中评估了多种噪声方法，最终发现通过随机打乱原始句子的顺序，再使用首创的新型文本填充方法(即用单个 mask token 替换文本片段，换句话说不管是被mask掉多少个token，都只用一个特定的mask token表示该位置有token被遮蔽了)能够获取最优性能。BART 尤其擅长处理文本生成任务，不过在自然语言理解任务中也颇有可圈可点之处。在同等训练资源下，BART 在 GLUE 和 SQuAD 数据集上的效果与 RoBERTa 不相伯仲，并在对话、问答和文本摘要等任务中斩获得新的记录(PS：特指BART刚出道之际。比如CNN / Daily Mail当下冠军模型是Big Bird加持下的BigBird-Pegasus)，在 XSum 数据集上的性能比之前的最佳结果高出了6个ROUGE。在机器翻译任务中，BART 在仅使用目标语言预训练的情况下，获得了比回译系统高出 1.1 个 BLEU 值的结果。此外，文章还使用控制变量法在BART 框架内使用其他预训练机制，从而更好地评估影响下游任务性能的因素。

模型

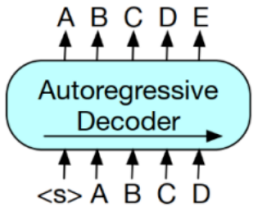
**

**

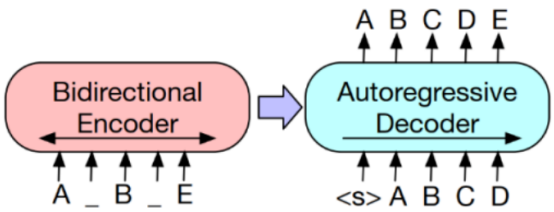
BART结合双向(比如BERT)和自回归(比如GPT) Transformer对模型进行预训练。BART还参考了GPT中的激活函数，将ReLU也改为GeLU。BART、BERT和GPT之间的对比如 Figure1所示。



(a) BERT: Random tokens are replaced with masks, and the document is encoded bidirectionally. Missing tokens are predicted independently, so BERT cannot easily be used for generation.



(b) GPT: Tokens are predicted auto-regressively, meaning GPT can be used for generation. However words can only condition on leftward context, so it cannot learn bidirectional interactions.



(c) BART: Inputs to the encoder need not be aligned with decoder outputs, allowing arbitary noise transformations. Here, a document has been corrupted by replacing spans of text with a mask symbols. The corrupted document (left) is encoded with a bidirectional model, and then the likelihood of the original document (right) is calculated with an autoregressive decoder. For fine-tuning, an uncorrupted document is input to both the encoder and decoder, and we use representations from the final hidden state of the decoder.

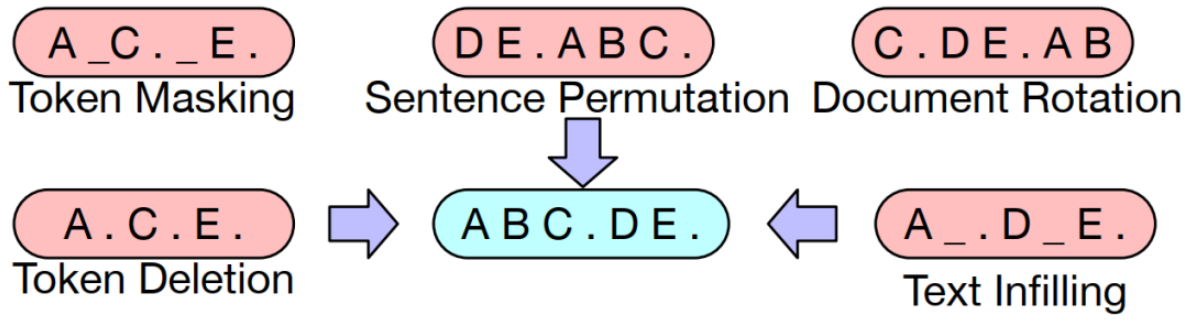
BART 的结构在上图中已经很明确了：就是一个BERT+GPT的结构；但是不同之处在于（也是作者通篇在强调的），相对于BERT中单一的noise类型(只有简单地用[MASK] token进行替换这一种noise)，BART在encoder端尝试了多种noise。其原因和目的也很简单:

- BERT的这种简单替换导致的是encoder端的输入携带了有关序列结构的一些信息（比如序列的长度等信息），而这些信息在文本生成任务中一般是不会提供给模型的。
- BART采用更加多样的noise，**意图是破坏掉这些有关序列结构的信息**，防止模型去"依赖"这样的信息。

预训练

**

BART的损失函数是decoder的输出与原始文本之间的交叉熵。与其他去噪自编码器(一般需要定制特定的噪声方案)不同的是BART可以使用任何的加噪方式。在极端情况下，源信息可以全部缺失，此时的BART就蜕化成了一个语言模型。文章中用到的加噪方案(即原始文本如何被破坏)如Figure 2所示。



具体来说主要有：

(1)Token Masking：与 BERT 一样，BART 随机采样 token，并用 [MASK] 这一预定义的特殊token进行替换。

(2)Token Deletion：从输入中随机删除 token。与 Token Masking不同，模型必须同时确定输入中缺失的位置。

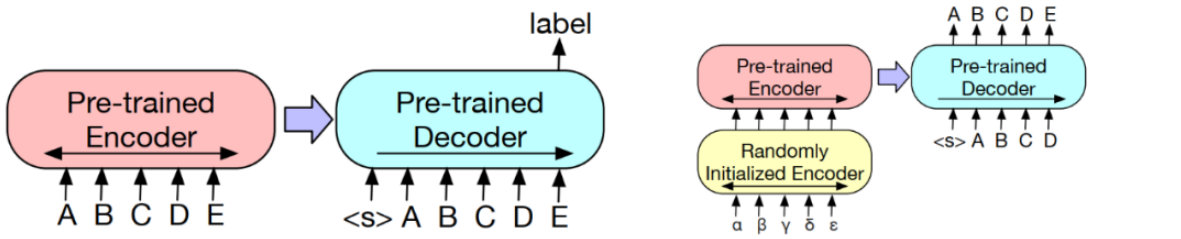
(3)Text Infilling：采样多个文本片段，每个文本片段长度服从λ = 3的泊松分布。每个文本片段用单个 [MASK] token替换。从泊松分布中采样出长度为 0 的文本片段对应 插入 [MASK] token。这种文本填充方法的思想源于SpanBERT，但SpanBERT采样的文本片段长度服从的是几何分布，且用等长的[MASK] token 序列替换掉文本片段。因此，BART能够迫使模型学习到一个片段中所缺失的token数量。

(4)Sentence Permutation：这里的句子排列变换是指按**句号**将文档分割成多个句子，然后随机打乱这些句子。

(5)Document Rotation：随机均匀地选择 一个token，再旋转文档使文档以该 token 作为起始。该任务的目的是训练模型识别文档开头。

Fine-tuning

BART在文本分类和翻译任务中的微调如Figure 3所示。以下具体介绍 BART 在各个下游任务的微调。



(a) To use BART for classification problems, the same input is fed into the encoder and decoder, and the representation from the final output is used.

(b) For machine translation, we learn a small additional encoder that replaces the word embeddings in BART. The new encoder can use a disjoint vocabulary.

Figure 3: Fine tuning BART for classification and translation.

Figure 3中a：当使用 BART 解决分类问题，用相同的输入文本输入到encoder和decoder，使用最终输出的表征。b：对于机器翻译任务，训练一个额外的小型encoder来替换 BART 中的词嵌入。新encoder可使用不同的词汇。

序列分类任务：

同一个输入同时输入到encoder 和decoder，将最后decoder的token的最终隐层状态被输入到一个新的多类别线性分类器中。该方法与 BERT 中的 CLS token 类似，不过 BART 在decoder最后额外添加了一个 token，如此该 token 在decoder中的表征可以关注到完整输入的decoder状态（见Figure 3a）。

token 分类任务：

token的分类任务，比如SQuAD中答案端点位置的分类。将完整文档输入到encoder和decoder中，使用decoder最上方的隐状态作为每个token的表征以判断该 token 的类别，比如是否为答案端部。

序列生成任务：

由于 BART 具备自回归解码器，因此可以直接应用到序列生成任务(如生成式问答和文本摘要)进行微调。在这两项任务中，从输入复制经过处理的信息，这与去噪预训练目标紧密相关。encoder的输入是输入序列，decoder以自回归的方式生成输出。

机器翻译：

BART用以机器翻译的时候，将整个BART(包括encoder和decoder)作为一个单独的预训练decoder，并增加一系列的从双语语料学习而得的encoder，如 Figure 3b所示。具体操作上是用一个新的随机初始化encoder替换 BART encoder的嵌入层。该模型以端到端的方式训练，即训练一个新的encoder将其他语种词映射到输入(BART可将其去噪为英文）。这个新的encoder可以使用不同于原始 BART 模型的词汇表。

源encoder的训练分两步，均需要将BART模型输出的交叉熵损失进行反向传播。

(1)冻结 BART 的大部分参数，仅更新随机初始化的源encoder、BART 位置嵌入和 BART encoder第一层的自注意力输入投影矩阵。

(2)将所有模型参数进行少量迭代训练。

实验

预训练目标对比

文章中还充分对比了不同预训练目标的影响，包括：

(1)语言模型：与GPT类似，训练一个从左到右的Transformer语言模型。该模型相当于BART的decoder，只是没有交叉注意(cross-attention)。

(2)排列语言模型：该模型基于XLNet，采样1/6的token，并以自回归的随机顺序生成。为了与其他模型保持一致，这里没有引入相对位置编码和XLNet中的片段级的循环注意力机制。

(3)带遮蔽的语言模型：与BERT相同，15%的token用 [MASK] token替换，训练模型重建出这些被遮蔽掉的token。

(4)多任务遮蔽的语言模型：与 UniLM 一样，使用额外self-attention mask训练带遮蔽的语言模型。自注意力遮蔽按如下比例随机选择:1/6从左到右；1/6从右到左；1/3未遮蔽；剩余的1/3中前50%的未遮蔽，其余的从左到右遮蔽。

(5)带遮蔽的seq-to-seq：与MASS模型类似，遮蔽一个片段中50%的token，并训练一个序列到序列模型预测被遮蔽的tokens。

实验过程对比了两种方案：

(1)将所有任务视为sequence-to-sequence问题，source端输入到encoder，decoder端的输出即为target结果。

(2)在decoder端将source作为target的一个前缀，且只在序列的target部分有损失函数。

实验发现前者对BART模型更有效，后者对其他模型更有效。更加详细的实验结果如 Table 1所示。

Model	SQuAD 1.1 F1	MNLI Acc	ELI5 PPL	XSum PPL	ConvAI2 PPL	CNN/DM PPL
BERT Base (Devlin et al., 2019)	88.5	**84.3**	-	-	-	-
Masked Language Model	90.0	83.5	24.77	7.87	12.59	7.06
Masked Seq2seq	87.0	82.1	23.40	6.80	11.43	6.19
Language Model	76.7	80.1	**21.40**	7.00	11.51	6.56
Permuted Language Model	89.1	83.7	24.03	7.69	12.23	6.96
Multitask Masked Language Model	89.2	82.4	23.73	7.50	12.39	6.74
BART Base						
w/ Token Masking	90.4	84.1	25.05	7.08	11.73	6.10
w/ Token Deletion	90.4	84.1	24.61	6.90	11.46	5.87
w/ Text Infilling	**90.8**	84.0	24.26	**6.61**	**11.05**	5.83
w/ Document Rotation	77.2	75.3	53.69	17.14	19.87	10.59
w/ Sentence Shuffling	85.4	81.5	41.87	10.93	16.67	7.89
w/ Text Infilling + Sentence Shuffling	**90.8**	83.8	24.17	6.62	11.12	**5.41**

Table 1：预训练目标对比。各个预训练目标源于BERT, MASS, GPT, XLNet和UniLM。对比的模型都是尺寸近似，训练步数都是1M，预训练使用的数据也相同。

可以看出使用文本填充方案(Text Infilling)的BART战绩斐然。从中可以得出以下结论：

(1)在不同的任务中，预训练方法的表现有显著差异。换句话说，预训练方法的有效性高度依赖于任务本身。比如，一个简单的语言模型在ELI5数据集上可以夺冠，但是在SQUAD上的结果却是最差的。

(2)遮蔽Token至关重要。只使用旋转文档或句子组合的预训练目标则效果较差，效果较好的都是使用了token的删除或遮蔽作为预训练目标。此外，在生成任务上，删除token似乎比遮蔽token更胜一筹。

(3)从左到右的预训练目标有助于文本生成任务。遮蔽语言模型和排列语言模型在文本生成任务上不如其他模型。而这两种模型在预训练阶段都没有用到从左到右的自回归语言模型。

(4)对于SQuAD而言双向的encoder至关重要。因为上下文在分类决策中至关重要，BART仅用双向层数的一半就能达到BERT类似的性能。

(5)预训练目标并不是唯一重要的因素。这里的排列语言模型略逊于XLNet，其中一些差异可能是由于没有使用XLNet架构中的其他的改进，如相对位置编码和片段级的循环机制。

(6)纯语言模型在ELI5数据集上技压群雄，其困惑度远优于其他模型。这表明当输出仅受到输入的松散约束时，BART较为低效。

总而言之，使用文本填充预训练目标的BAR在多项任务上(除了ELI5之外)效果都很好。

Large版模型对比

**
**

自然语言理解任务

由于更大模型和更大batch size有助于下游任务性能的提升，所以文章还进一步对比各模型的large版。Large版的BART，encoder和decoder分别有12层，隐层大小为1024，batch size与RoBERTa一样都是8000，模型预训练了500000个step。tokenized方法借用 GPT-2 中的字节对编码(BPE)。各个模型在GLUE上的实验对比结果如 Table 2所示。

	MNLI m/mm	**SST** Acc	**QQP** Acc	**QNLI** Acc	**STS-B** Acc	**RTE** Acc	**MRPC** Acc	**CoLA** Mcc
BERT	86.6/-	93.2	91.3	92.3	90.0	70.4	88.0	60.6
UniLM	87.0/85.9	94.5	-	92.7	-	70.9	-	61.1
XLNet	89.8/-	95.6	91.8	93.9	91.8	83.8	89.2	63.6
RoBERTa	**90.2/90.2**	96.4	92.2	94.7	**92.4**	86.6	**90.9**	**68.0**
BART	89.9/90.1	**96.6**	**92.5**	**94.9**	91.2	**87.0**	90.4	62.8

Table 2：Large版模型在 SQuAD 和 GLUE 上的实验结果。BART 的效果可比肩 RoBERTa 和 XLNet，这表明 BART 的单向decoder层并不会降低模型在判别任务上的性能。

各个模型在SQuAD上的对比结果如Table 3所示。

	SQuAD 1.1 EM/F1	SQuAD 2.0 EM/F1
BERT	84.1/90.9	79.0/81.8
UniLM	-/-	80.5/83.4
XLNet	**89.0**/94.5	86.1/88.8
RoBERTa	88.9/**94.6**	**86.5/89.4**
BART	88.8/**94.6**	86.1/89.2

Table 3：BART的结果与XLNet和RoBERTa不相伯仲。

总体而言，BART在自然语言理解任务上与其他先进模型不相上下。这表明BART在生成任务上的进一步突破并不是以牺牲自然语言理解性能为代价。

自然语言生成任务

在文本生成任务中选用了摘要生成(CNN/DailyMail 和XSum)、对话(CONVAI2)和生成式问答(ELI5，是一个长篇问答数据集)中对应的数据集进行评测，结果如 Table 4所示。

	CNN/DailyMail			XSum		
	R1	R2	RL	R1	R2	RL
Lead-3	40.42	17.62	36.67	16.30	1.60	11.95
PTGEN (See et al., 2017)	36.44	15.66	33.42	29.70	9.21	23.24
PTGEN+COV (See et al., 2017)	39.53	17.28	36.38	28.10	8.02	21.72
UniLM	43.33	20.21	40.51	-	-	-
BERTSUMABS (Liu & Lapata, 2019)	41.72	19.39	38.76	38.76	16.33	31.15
BERTSUMEXTABS (Liu & Lapata, 2019)	42.13	19.60	39.18	38.81	16.50	31.27
ROBERTASHARE (Rothe et al., 2019)	40.31	18.91	37.62	41.45	18.79	33.90
BART	**44.16**	**21.28**	**40.90**	**45.14**	**22.27**	**37.25**

Table 4：在两个标准摘要数据集上的结果。

从结果可以看出，在这两个摘要任务上，BART 在所有度量指标上均优于之前的模型。BART在更抽象的XSum 数据集上的比之前最优的RoBERTa模型高出3.5个点(所有的ROUGE指标)。此外，从人工评测的角度来看，BART也大幅优于之前的模型。但与人类的摘要结果相比仍然有差距。

各模型在CONVAI2上的实验结果如Table 5所示。

	ConvAI2	
	Valid F1	Valid PPL
Seq2Seq + Attention	16.02	35.07
Best System [2]	19.09	17.51
BART	**20.72**	**11.85**

Table 5：BART 在对话生成任务上的性能优于之前的模型。其中困惑度基于 ConvAI2 官方 tokenizer 进行了重新归一化。

在ELI5数据集上的评测结果如Table 6所示。

	ELI5		
	R1	R2	RL
Best Extractive	23.5	3.1	17.5
Language Model	27.8	4.7	23.1
Seq2Seq	28.3	5.1	22.8
Seq2Seq Multitask	28.9	5.4	23.1
BART	**30.6**	**6.2**	**24.3**

Table 6：BART在具有挑战性的ELI5长文档问答数据集上达到了最先进的结果。

发现BART的性能比之前最好的工作(指Seq2Seq Multi-task)高出1.2个 ROUGE-L。其实该数据集难度较大，因为数据集中的问题只对答案进行了微弱的指定。

leader board结果如下：

Model	Full ROUGE			Fill 20 ROUGE		
	1	2	L	1	2	L
BART [Lewis et al, 2019]	30.6	6.2	24.3	--	--	--
Local KB Construction [Fan et al, 2019]	30.0	5.8	24.0	--	--	--
Seq2Seq Multi-task [Fan et al, 2019]	28.9	5.4	23.1	37.2	14.6	33.0
Extractive [Fan et al, 2019]	23.5	3.11	17.5	--	--	--

PS：是因为数据比较新，所以知名度不够高，刷榜的人寥寥无几?

机器翻译任务

BART在WMT16 Romanian-English上与其他模型的对比结果如Table 8所示。

	RO-EN
Baseline	36.80
Fixed BART	36.29
Tuned BART	**37.96**

Table 8：BART 和基线模型(Transformer)在机器翻译任务上的性能对比情况。

参与对比的模型使用数据集包括 WMT16 RO-EN 和用回译系统做的扩增数据。可以看出BART使用单语英文预训练，性能结果优于基线模型。

总结

**
**

文本介绍了一种预训练模型：BART。该模型可以学习将被破坏的文档重建回原始文档。BART在分类任务上的性能结果与RoBERTa相当，并在几个文本生成任务上刷新记录。