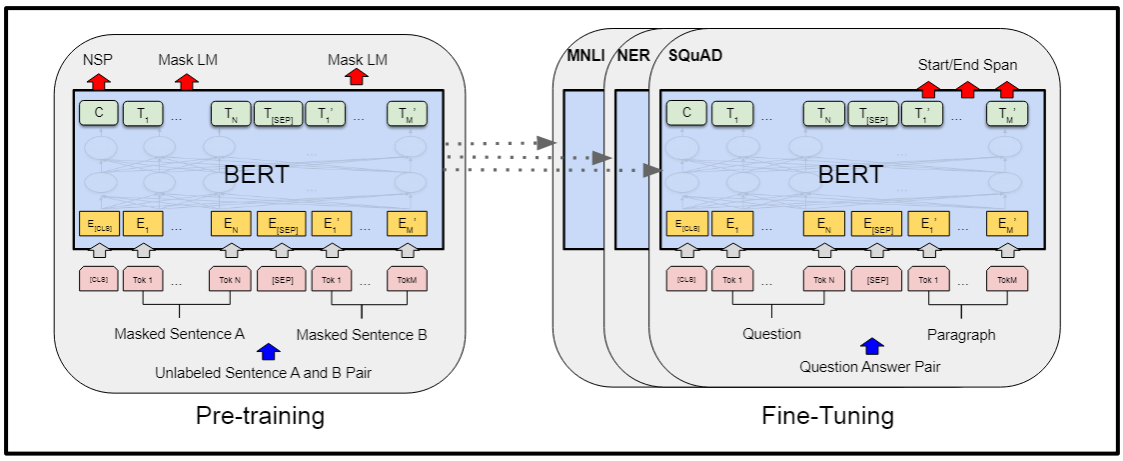
《BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding》

摘要：

1. **论文背景：**语言模型的预训练早已被证明对改进许多自然语言处理任务是有效的。而自 Transformer 框架提出后，NLP 领域中的问题不断在此框架上获得改进，但 Transformer 框架之下的巨大潜力仍亟待各方研究者进一步开发。
2. **论文的贡献：**以 Transformer 框架下的编码器结构为基础，提出了一种新的语言表示模型 BERT。它展示了双向预训练对语言表示的重要性，并且还表明了利用预训练的表示能够减少对许多精心设计的特定任务架构的需求。
3. **主要创新点：**不同于 GPT，BERT 模型采用了**双向结构**，即能通过左右上下文来对未标记的文本进行预训练，而在下游任务上也能通过微调方式有效地实现任务。
4. **实验结果：**BERT 框架在 11 个自然语言处理任务上获得了新的最好成绩，其中包括将$GLUE$得分提高到 80.5%（7.7%的绝对改进）、$MultiNLI$ 准确率提高到 86.7%（4.6%的绝对改进）、$SQuAD\ v1.1$问答测试$F1$得分提高到 93.2（1.5 分的绝对改进）以及$SQuAD\ v2.0$测试$F1$得分提高到 83.1（5.1 分的绝对改进）。

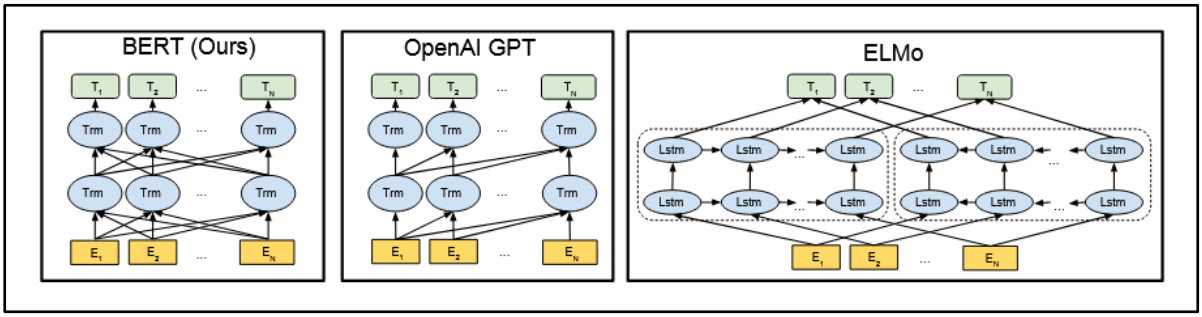
BERT 的整体框架：



如上图所示，BERT 的具体实现框架中包含两个步骤：**预训练（Pre-training）**和**微调（Fine-Tuning）**。在预训练期间，我们可以通过不同的预训练任务（NSP 和 MLM）对未标记的数据进行模型训练。而在微调阶段，我们可以首先使用预

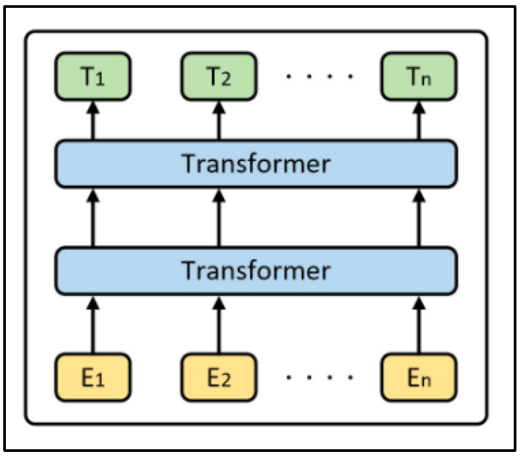
训练的参数来初始化 BERT 模型，然后使用下游任务（MNLI、NER、SQuAD）中的标记数据对所有参数进行微调。（**注意：**每个下游任务都有单独的微调模型，即便它们采用了相同的预训练参数进行了初始化。）

BERT 的模型结构：



BERT 的模型结构如上图左一所示，其中最下层为词向量表示，中间层为 Transformer 块，最上层为输出 Token。而对比上图中间的 OpenAI GPT（Generative pre-trained transformer）的模型结构，我们可以发现 BERT 它不同于 GPT 的最主要一点就是**它是双向的 Transformer block 连接**。此外，对比上图右一的 ELMo 模型结构，我们可以发现虽然二者都是“双向的”，但它们最终的目标函数和中间层都是不同的。

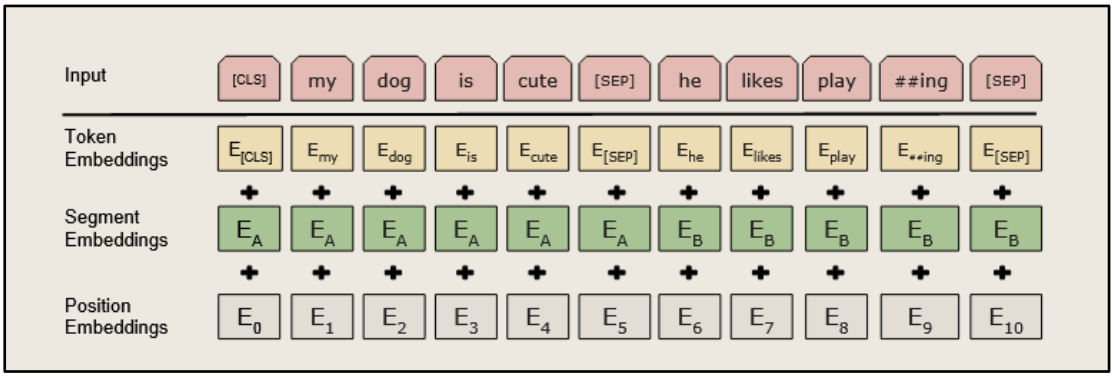
除此之外，我们还可以对上述 BERT 的模型结构作进一步简化：



如图所示，实际上 BERT 的网络结构并不复杂。

BERT 的输入表示

在 BERT 中，Input Embedding 主要包含三个部分：**Token Embedding**、**Positional Embedding** 和 **Segment Embedding**，如下图所示：



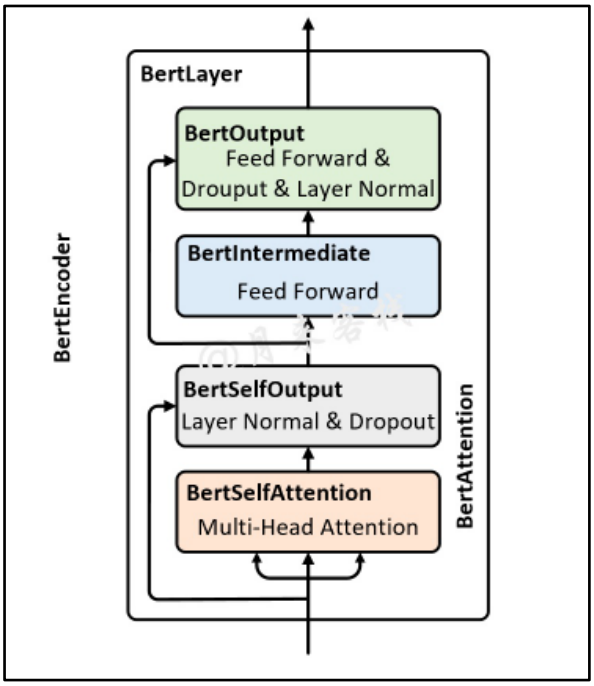
Token Embedding：即对输入序列中的每个单词根据字典进行向量化，其中[CLS]是在每个输入示例前添加的特殊符号，而[SEP]是特殊的分隔符标记。

Positional Embedding：位置表示，需要注意的是这里对于每个单词的位置处理并不是像 Transformer 中采用公式计算出来的，而是直接根据单词的位置（如1、2、3）来 Embedding 的。

Segment Embedding：片段表示，用来区分输入序列中的不同序列，其本质就是通过一个普通的词嵌入来区分每一个序列所处的位置。例如，在 NSP 任务中，对于任意一个序列的每一位置都将用同一个向量来进行表示。

最后，BERT 模型简单将这三部分的 Embedding 结果相加（并进行标准化）然后得到最终的 Input Embedding 部分的输出。

BERT 的 Encoder 结构：（如图所示）



具体的，在论文中作者分别用L来表示 BertLayer 的层数，即 BertEncoder 是由L个 BertLayer 所构成的；用H来表示模型的维度；用A来表示多头注意力中多头的个数。同时，在论文中作者分别就$BERT_{base}(L=12, H=768, A=12)$和$BERT_{large}(L=24, H=1024, A=16)$这两种尺寸的 BERT 模型进行了实验对比。

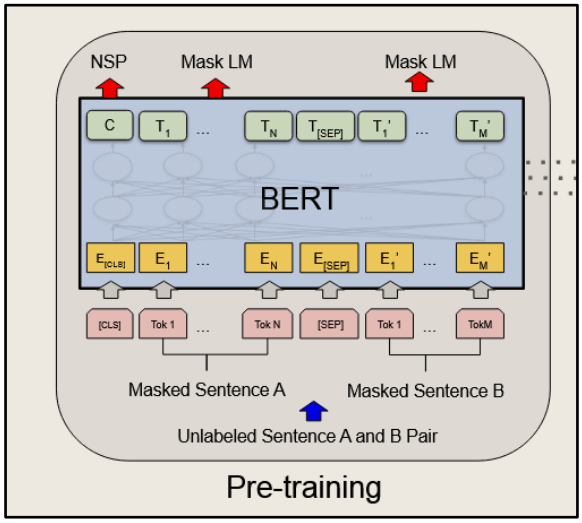
预训练任务——MLM 与 NSP：

对于 **MLM 任务**来说，其做法是随机掩盖掉输入序列中 15%的 Token（即用“[MASK]”替换掉原有的 Token），然后在 BERT 的输出结果中取对应掩盖位置上的向量进行真实值预测。不过作者接着提到，虽然 MLM 的这种做法能够得到一个很好的预训练模型，但是仍旧存在着不足之处——在 fine-tuning 时，由于输入序列中并不存在“[MASK]”这样的 Token，因此这将导致 pre-training 和 fine-tuning 之间存在匹配不一致的问题（GAP）。

所以，为了解决这一问题，作者在原始 MLM 的基础了做了部分改动，即先选定 15%的 Token，然后将其中的 80%替换为“[MASK]”、10%随机替换为其它 Token、剩下的 10%不变。最后，取这 15%的 Token 对应的输出做分类来预测其真实值。此外，作者还给出了一个不同掩盖策略下的对比结果，如下图所示：

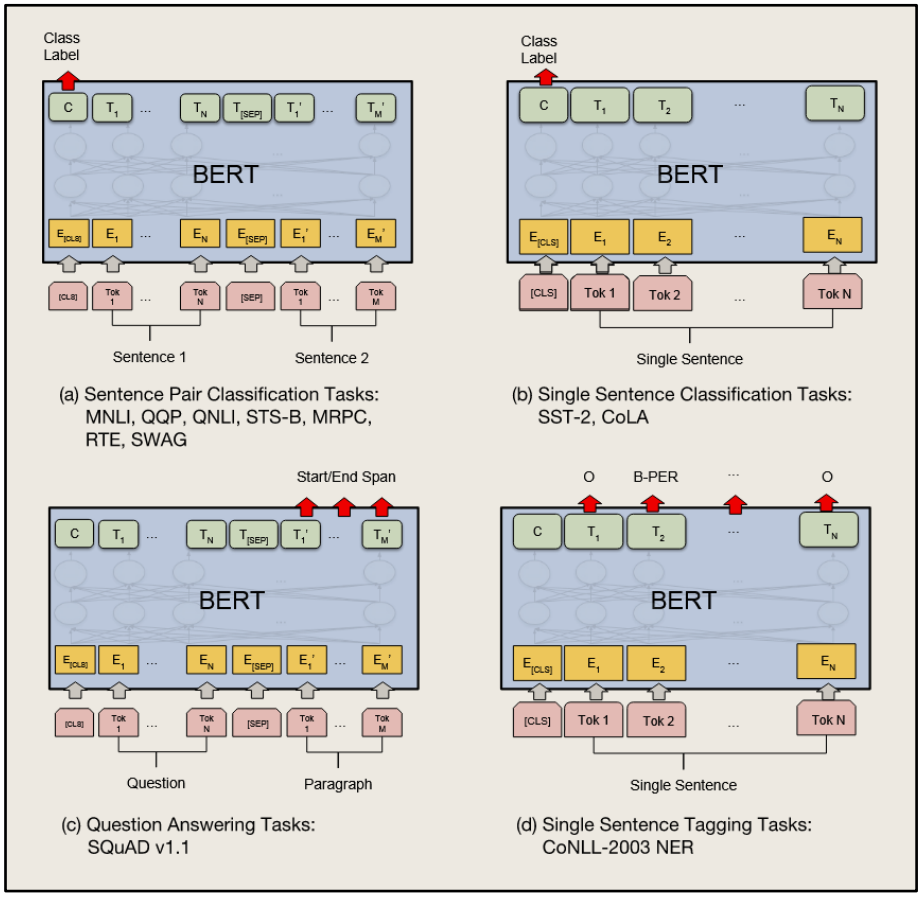
Masking Rates			Dev Set Results		
MASK	SAME	RND	MNLI Fine-tune	NER Fine-tune	NER Feature-based
80%	10%	10%	84.2	95.4	94.9
100%	0%	0%	84.3	94.9	94.0
80%	0%	20%	84.1	95.2	94.6
80%	20%	0%	84.4	95.2	94.7
0%	20%	80%	83.7	94.8	94.6
0%	0%	100%	83.6	94.9	94.6

而对于 **NSP 任务**，简单来说就是由于每个输入序列都是由 A 和 B 两句话拼接构成的，并且其中 50%的情况是 B 确实为 A 的下一句话，另外的 50%的情况是 B 为语料库中其它的随机句子（不是 A 的下一句话）。然后，我们利用 BERT 预训练模型来预测 B 是否为 A 的下一句话，即**二分类的下句预测任务**。



如上图所示便是 MLM 和 NSP 这两个任务在 BERT 预训练时的输入和输出示意图，其中最上层输出的C在预训练时用于 NSP 中的分类任务；其它位置上的T_i和T_j'则用于预测被掩盖的 Token（MLM）。

微调任务 Fine-Tuning：



如上图所示，不同的 NLP 任务有着不同的微调模型。

实验结果：（BERT 在 11 个 NLP 任务上的微调结果）

GLUE

对于所有的数据，模型参数设置如下——Batch size：32；学习率：5e-5,4e-5, 3e-5, 2e-5；epochs：3。

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	Average -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

SQuAD 1.1

说明：斯坦福问题解答数据集（SQuAD v1.1）是一个 10 万个众包问题/答案对的集合。给定一个问题以及 Wikipedia 中包含答案的段落，任务是预测段落中的答案文本范围。

System	Dev EM	Dev F1	Test EM	Test F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	-	-	82.3	91.2
#1 Ensemble - nlnet	-	-	86.0	91.7
#2 Ensemble - QANet	-	-	84.5	90.5
Published				
BiDAF+ELMo (Single)	-	85.6	-	85.8
R.M. Reader (Ensemble)	81.2	87.9	82.3	88.5
Ours				
$\text{BERT}_{\text{BASE}}$ (Single)	80.8	88.5	-	-
$\text{BERT}_{\text{LARGE}}$ (Single)	84.1	90.9	-	-
$\text{BERT}_{\text{LARGE}}$ (Ensemble)	85.8	91.8	-	-
$\text{BERT}_{\text{LARGE}}$ (Sgl.+TriviaQA)	**84.2**	**91.1**	**85.1**	**91.8**
$\text{BERT}_{\text{LARGE}}$ (Ens.+TriviaQA)	**86.2**	**92.2**	**87.4**	**93.2**

SQuAD 2.0

说明：SQuAD 2.0 任务通过允许在所提供的段落中不存在简短答案的可能性扩展了 SQuAD 1.1 问题定义，从而使问题看上去更加地贴合实际。

System	Dev EM	Dev F1	Test EM	Test F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	86.3	89.0	86.9	89.5
#1 Single - MIR-MRC (F-Net)	-	-	74.8	78.0
#2 Single - nlnet	-	-	74.2	77.1
Published				
unet (Ensemble)	-	-	71.4	74.9
SLQA+ (Single)	-		71.4	74.4
Ours				
$\text{BERT}_{\text{LARGE}}$ (Single)	78.7	81.9	80.0	83.1

SWAG

说明：该数据集包含 113k 个句子对完成示例，这些示例评估了基于常识的推理。给定一个句子，任务是在四个选择中选择最合理的答案。而模型的参数设置如下——Batch size：16；学习率：2e-5；epochs 数量：3。

System	Dev	Test
ESIM+GloVe	51.9	52.7
ESIM+ELMo	59.1	59.2
OpenAI GPT	-	78.0
$\text{BERT}_{\text{BASE}}$	81.6	-
$\text{BERT}_{\text{LARGE}}$	**86.6**	**86.3**
Human (expert)†	-	85.0
Human (5 annotations)†	-	88.0

关于 BERT 的消融研究：

Tasks	Dev Set MNLI-m (Acc)	QNLI (Acc)	MRPC (Acc)	SST-2 (Acc)	SQuAD (F1)
$\text{BERT}_{\text{BASE}}$	84.4	88.4	86.7	92.7	88.5
No NSP	83.9	84.9	86.5	92.6	87.9
LTR & No NSP	82.1	84.3	77.5	92.1	77.8
+ BiLSTM	82.1	84.1	75.7	91.6	84.9

如图所示是**使用$\boldsymbol{BERT}_{\text{BASE}}$架构的预训练任务消融**。“No NSP”是指在没有下一个句子预测任务的情况下进行训练；“LTR & No NSP”是指被训练为没有下一句话预测的从左到右的 LM，就像 OpenAI GPT 一样；“+ BiLSTM”是指在微调期间在“LTR + No NSP”模型之上添加随机初始化的 BiLSTM。

而从图中我们可以看到，$\boldsymbol{BERT}_{\text{BASE}}$的效果是最好的。

Hyperparams				Dev Set Accuracy		
#L	#H	#A	LM (ppl)	MNLI-m	MRPC	SST-2
3	768	12	5.84	77.9	79.8	88.4
6	768	3	5.24	80.6	82.2	90.7
6	768	12	4.68	81.9	84.8	91.3
12	768	12	3.99	84.4	86.7	92.9
12	1024	16	3.54	85.7	86.9	93.3
24	1024	16	3.23	86.6	87.8	93.7

如图所示是关于 **BERT 模型大小的消融实验**。其中，#L 为层数；#H为隐藏层大小；#A为注意力头数。**而从图中我们可以看到，BERT 模型越大（即参数量越多），模型的准确率就越高，并且其提升较为明显。**

总结：

之前的语言模型的迁移学习表明，丰富的、无监督的预训练是许多语言理解系统不可或缺的一部分，特别是这些结果能使即使是低资源的任务也能从深度单向架构中受益。**而本文的主要贡献是进一步将这些发现推广到深度双向架构中，允许相同的预训练模型能够成功地应用到广泛的 NLP 任务中。**

对本文的感悟：

由于 BERT 模型是基于 Transformer 架构的改进，因此其相对于 RNN 更加高效、能捕捉到更长距离的依赖。此外，对比在 BERT 提出之前的预训练模型，它能捕捉到**真正意义上的双向上下文信息**。而如今，BERT 也已经被广泛作为各种下游任务的预训练模型。总之，BERT 的提出对 NLP 领域的发展做出了巨大的贡献。