BERT：深度双向变压器的预训练，用于语言理解

雅各布德夫林 张明伟 肯顿李 克里斯蒂娜Toutanova
Google AI语言
{雅各布德林，明伟长，肯托尔，KistOut} GoGoLe.com

摘要

我们引入了一种名为BERT的新语言表示模型，它代表变形金刚的双向编码器表示。与最近的语言表征模型不同彼得斯等人。，2018a；拉德- 福特等人。，2018），BERT旨在通过联合调节所有层中的左右上下文来预先训练来自未标记文本的深度双向表示。因此，预训练的BERT模型可以通过一个额外的输出层进行微调，为各种任务创建最先进的模型，例如问答和语言推理，而无需基本的任务特定架构修改。

BERT在概念上简单且经验丰富。它在11项自然语言处理任务中获得了最新的最新成果，包括将GLUE得分提高到80.5%（绝对改善率为7.7%），
MultiNLI精度达到86.7%（绝对值提高4.6%），SQuAD v1.1问题回答测试F1到93.2（1.5点绝对改进）和SQuAD v2.0测试F1到83.1（5.1点绝对改进）。

1 介绍

语言模型预训练已被证明可有效改善许多自然语言处理任务（戴和勒，2015；彼得斯等人。，2018a；Radford等。，2018；霍华德和罗德，2018）.这些包括句子级任务，如自然语言推理（鲍曼等人。，2015；威廉姆斯等人。，2018）和释义（杜兰 和布罗基特，2005），旨在通过整体分析句子以及令牌级任务（如命名实体识别和问答）来预测句子之间的关系，其中模型需要在令牌级别生成细粒度输出（Tjong Kim Sang和 德梅德尔，2003；Rajpurkar等人。，2016）.

将预训练语言表示应用于下游任务有两种现有策略：基于特征和微调。基于特征的方法，例如ELMo（彼得斯 等。，2018a），使用特定于任务的体系结构，其中包括预先训练的表示作为附加功能。微调方法，如Generative Pre-trained Transformer（OpenAI GPT）（Radford等。，2018），引入最小的任务特定参数，并通过简单地微调所有预训练参数来训练下游任务。这两种方法在预训练期间共享相同的目标函数，在这些方法中，他们使用单向语言模型来学习一般语言表示。

我们认为当前的技术限制了预训练表示的能力，特别是对于微调方法。主要限制是标准语言模型是单向的，这限制了在预训练期间可以使用的体系结构的选择。例如，在OpenAI GPT中，作者使用从左到右的架构，其中每个令牌只能处理Transformer的自我关注层中的先前令牌（Vaswani等。，2017）.这种限制对于句子级别的任务来说是次优的，并且在将基于微调的方法应用于诸如问答的令牌级任务时可能是非常有害的，其中从两个方向合并上下文是至关重要的。

在本文中，我们通过提出BERT：变换器的双向编码器表示来改进基于微调的方法。BERT通过使用受Cloze任务启发的“蒙面语言模型”（MLM）预训练目标来缓解前面提到的单向性约束（泰勒，1953）.蒙面语言模型从输入中随机屏蔽一些标记，目的是预测被屏蔽的原始词汇id

单词仅基于其上下文。与从左到右的语言模型预训练不同，MLM目标使得表示能够融合左右上下文，这允许我们预先训练深度双向变换器。除了蒙面语言模型，我们还使用联合预训练文本对表示的“下一句预测”任务。我们的论文的贡献如下：

- 我们证明了双向预训练对语言表达的重要性。不像Radford等。(2018)，它使用单向语言模型进行预训练，BERT使用掩蔽语言模型来实现预训练的深度双向表示。这也与之形成鲜明对比彼得斯等人。(2018a)，它使用由独立训练的从左到右和从右到左LM的浅层连接。
- 我们展示了预先训练的表示减少了对许多重型工程任务特定体系结构的需求。BERT是第一个基于微调的表示模型，它在大量句子级和令牌级任务上实现了最先进的性能，优于许多特定于任务的体系结构。
- BERT推进了11项NLP任务的最新技术。代码和预先训练的模型可在https://github.com/谷歌的研究/BERT.

2 相关工作

预培训通用语言表示有很长的历史，我们将简要回顾本节中使用最广泛的方法。

2.1 无监督的基于特征的方法

几十年来，学习广泛适用的词语表达一直是一个活跃的研究领域，包括非神经学（布朗等人。，1992; 安藤和张, 2005; Blitzer等人。，2006）和神经（Mikolov等。，2013; Pennington等。，2014）方法。预先训练的文字嵌入是现代NLP系统不可或缺的一部分，与从头学习的嵌入相比提供了显着的改进（Turian等。，2010）. 为了预先训练单词嵌入向量，使用了从左到右的语言建模目标（美国国立卫生研究院和辛顿, 2009），以及在左右语境中区分正确与错误词语的目标（Mikolov等。，2013).

这些方法已被推广到更粗糙的粒度，例如句子嵌入（Kiros等。，2015; Logeswaran和Lee，2018）或段落嵌入（勒和米科洛夫，2014）. 为了训练句子表示，先前的工作已经使用目标来对候选句子进行排名（杰尼特等人。，2017; Logeswaran和背风处, 2018），从左到右生成下一句话词，给出前一句话的表示（Kiros等。，2015），或去噪自动编码器派生的目标（希尔等人。，2016).

ELMo及其前身（彼得斯等人。，2017，2018a）将传统的词汇嵌入研究概括为不同的维度。它们从左到右和从右到左的语言模型中提取上下文相关的功能。每个标记的上下文表示是从左到右和从右到左表示的串联。在将上下文字嵌入与现有任务特定体系结构集成时，ELMo推进了几个主要NLP基准测试的最新技术水平（彼得斯等人。，2018a）包括问答（Rajpurkar等人。，2016），情绪分析（Socher等。，2013）和命名实体识别（Tjong Kim Sang和De Meulder，2003). Melamud等。(2016）提议通过任务学习上下文表示，以使用LSTM预测来自左右上下文的单个单词。与ELMo类似，他们的模型基于特征而非深度双向。费杜斯等。(2018）表明完形填空任务可用于提高文本生成模型的稳健性。

2.2 无监督的微调方法

与基于特征的方法一样，第一个在这个方向上工作的只是预先训练的来自未标记文本的字嵌入参数（同事洛贝特和韦斯顿, 2008).

最近，产生上下文令牌表示的句子或文档编码器已经从未标记的文本进行了预训练，并针对受监督的下游任务进行了微调（戴和勒, 2015; 霍华德和罗德，2018; 雷德福等。，2018）. 这些方法的优点是需要从头开始学习很少的参数。至少部分由于这一优势，OpenAI GPT（Radford等。，2018）从GLUE基准测试中获得了许多句子级任务的先前最新结果（王等。，2018a）. 从左到右的语言模型

_

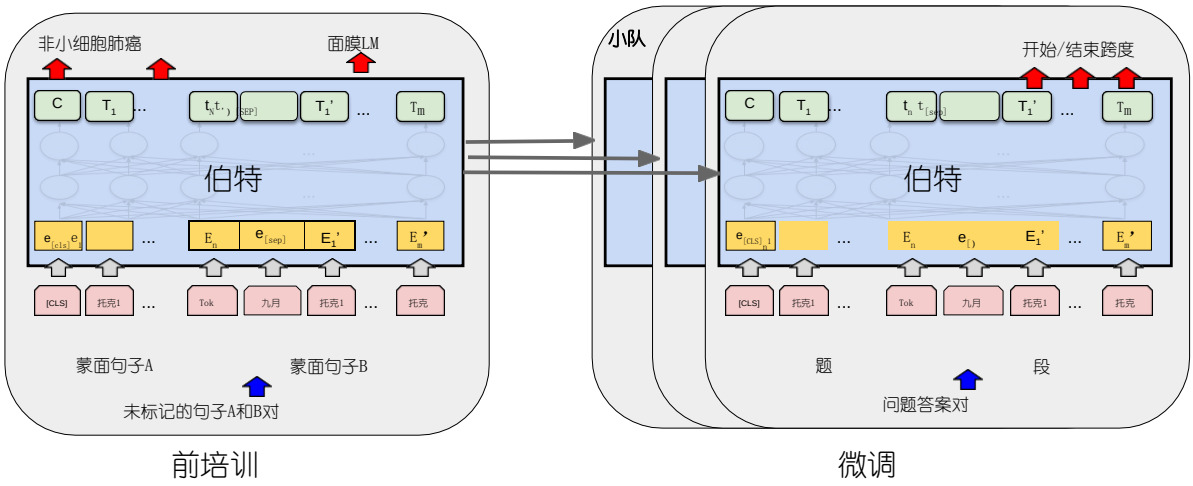

图1：BERT的整体预训练和微调程序。除了输出层之外，在预训练和微调中使用相同的架构。相同的预训练模型参数用于初始化不同下游任务的模型。在微调期间，所有参数都经过微调。[CLS]是在每个输入示例前添加的特殊符号，[SEP]是一个特殊的分隔符号（例如，分隔问题/答案）。

ing和自动编码器目标已被用于预训这样的模型（霍华德和罗德，2018; Radford等。，2018; 戴和勒，2015).

2.3 从监督数据转移学习

还有一些工作显示了有监督的任务与大型数据集的有效转移，例如自然语言推理（Conneau等。，2017）和机器翻译（麦肯等人。，2017）.计算机视觉研究也证明了从大型预训练模型转移学习的重要性，其中一个有效的方法是微调用Ima-geNet预训练的模型（邓等人。，2009; Yosinski等人。，2014).

3 伯特

我们在本节介绍BERT及其详细实现。我们的框架有两个步骤：预训练和微调。在预训练期间，模型在不同的预训练任务上训练未标记的数据。对于微调，首先使用预先训练的参数初始化BERT模型，并且使用来自下游任务的标记数据来微调所有参数。每个下游任务都有单独的微调模型，即使它们是ini

使用相同的预先训练的参数进行tialized。该问答案例如图1将作为本节的运行示例。

BERT的一个显着特点是它跨越不同任务的统一架构。有迷你

预训练架构与最终下游架构之间的差异。

模型体系结构BERT的模型体系结构是一个多层双向变换器编码器，它基于上面描述的原始实现Vaswani等。（2017）并在tensor2tensor库中发布。[1] 因为变形金刚的使用已经变得很普遍，而且我们的实现几乎与原始实现相同，所以我们将省略模型体系结构的详尽背景描述并引用读者Vaswani等。(2017）以及诸如“注释变形金刚”等优秀指南。[2]

在这项工作中，我们将层数（即变换器块）表示为L，将隐藏大小表示为H，将自我关注头的数量表示为A.[3] 我们主要报告两种模型尺寸的结果：BERT$_{基础}$（L = 12，H = 768，A = 12，总参数= 110M）和BERT$_{大}$（L = 24，H = 1024，
A = 16，总参数= 340M）。

选择BERT$_{基础}$具有与OpenAI GPT相同的模型尺寸用于比较目的。然而，重要的是，BERT变换器使用双向自我关注，而GPT变换器使用受限制的自我关注，其中每个令牌只能处理其左侧的上下文。[4]

[1] https://github.com/tensorflow/tensor2tensor

[2] http://nlp.seas.harvard.edu/2018/04/03/attention.html

[3] 在所有情况下，我们将前馈/滤波器大小设置为4H，即H = 768时为3072，H = 1024时为4096。

[4] 我们注意到在文献中双向Trans-

输入/输出表示为了使BERT处理各种下游任务，我们的输入表示能够在一个标记序列中明确地表示单个句子和一对句子（例如，问题，答案）。在整个这项工作中，“句子”可以是连续文本的任意跨度，而不是实际的语言句子。“序列”指的是BERT的输入令牌序列，其可以是单个句子或两个句子打包在一起。

我们使用WordPiece嵌入（吴等人。，2016）有30,000个令牌词汇表。每个序列的第一个标记始终是一个特殊的分类标记（[CLS]）。对应于该标记的最终隐藏状态用作分类任务的聚合序列表示。句子对被打包成一个序列。我们以两种方式区分句子。首先，我们用特殊标记（[SEP]）将它们分开。其次，我们在每个令牌上添加一个学习嵌入，指示它是属于句子A还是句子B.如图所示1，我们将输入嵌入表示为E，将特殊[CLS]标记的最终隐藏向量表示为$C \in R^h$，并将i^{th}输入标记的最终隐藏向量表示为$T_i \in R^h$。

对于给定的标记，其输入表示是通过对相应的标记，段和位置嵌入求和来构造。可以在图中看到这种结构的可视化2.

3.1 预训练BERT

不像彼得斯等人。（2018a）和Radford等。(2018)，我们不使用传统的从左到右或从右到左的语言模型来预训练BERT。相反，我们使用本节中描述的两个无人监督的任务来预训练BERT。该步骤显示在图的左侧部分1.

任务#1：蒙面LM直观地说，有理由相信深度双向模型比从左到右模型或从左到右和从右到左模型的浅层连接更严格。.遗憾的是，标准的条件语言模型只能从左到右或从右到左进行训练，因为双向调节会允许每个单词间接地“看到自己”，并且模型可以在多层次中平凡地预测目标单词上下文。

为了训练深度双向表示，我们简单地随机掩盖一定比例的输入令牌，然后预测那些被掩盖的令牌。我们将此过程称为“蒙面LM”（MLM），尽管它在文献中通常被称为完形任务（泰勒，1953).在这种情况下，对应于掩码令牌的最终隐藏向量被馈送到词汇表上的输出softmax，如在标准LM中。在我们的所有实验中，我们随机地屏蔽每个序列中所有WordPiece标记的15%。与去噪自动编码器相比（文森特等人。，2008），我们只预测被掩盖的单词而不是重建整个输入。

虽然这允许我们获得双向预训练模型，但缺点是我们在训练前和微调之间产生不匹配，因为在微调期间不会出现[MASK]标记。为了缓解这种情况，我们并不总是用实际的[MASK]令牌替换“蒙面”字。训练数据生成器随机选择15%的令牌位置进行预测。如果选择了第i个令牌，我们用（1）[MASK]令牌替换第一个令牌80%的时间（2）随机令牌10%的时间（3）未更改的第i个令牌10%的时间。然后，T_i将用于预测具有交叉熵损失的原始令牌。我们比较了附录中这个程序的变化C.2.

任务#2：下一句话预测（NSP）许多重要的下游任务，例如问答（QA）和自然语言推理（NLI），都是基于理解两个句子之间的关系，而这两个句子并不是由语言建模直接捕获的。为了训练理解句子关系的模型，我们预先训练二进制化的下一句预测任务，该任务可以从任何单语语料库中平凡地生成。具体来说，当为每个预训练的例子选择句子A和B时，50%的时间B是跟随A的实际下一个句子（标记为IsNext），50%的时间是来自语料库的随机句子（标记的）作为NotNext）。正如我们在图中所示1，C用于下一句子预测（NSP）。[5] 尽管它很简单，我们在Section中演示5.1 对此任务的预训练对QA和NLI都非常有益。[6]

前者通常被称为“变换器编码器”，而左上下文版本被称为“变换器解码器”，因为它可以用于文本生成。

[5]最终型号在NSP上达到97%-98%的准确度。

[6]矢量C不是没有微调的有意义的句子表示，因为它是用NSP训练的。

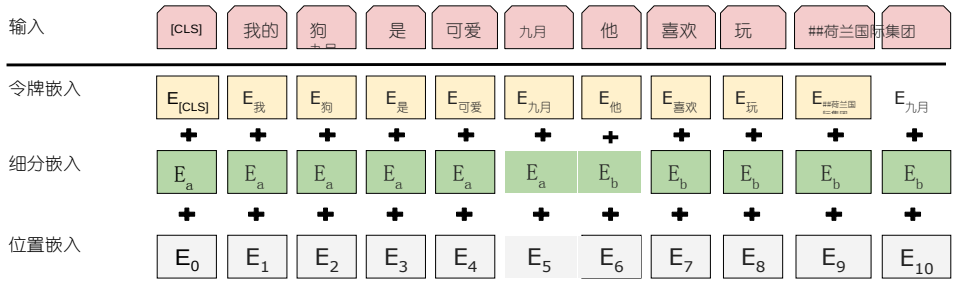

图2：BERT输入表示。输入嵌入是令牌嵌入，分段嵌入和位置嵌入的总和。

NSP任务与表达学习目标密切相关杰尼特等人。(2017）和Logeswaran和Lee (2018).但是，在以前的工作中，只有句子嵌入被转移到下游任务，其中BERT传输所有参数以初始化最终任务模型参数。

训练前数据训练前程序主要遵循现有的语言模型预训练文献。对于预训练语料库，我们使用BooksCorpus （800M字） （朱等人。，2015）和英语维基百科（2500万字）。对于维基百科，我们只提取文本段落并忽略列表，表格和标题。使用文档级语料库而不是改组的句子级语料库（如Billion Word Benchmark）至关重要（Chelba等人。，2013）以提取长的连续序列。

3.2 微调BERT

微调很简单，因为Transformer中的自我关注机制允许BERT通过交换适当的输入和输出来模拟许多下游任务 - 无论它们是单文本还是文本对。对于涉及文本对的应用程序，常见的模式是在应用双向交叉注意之前独立编码文本对，例如Parikh等。(2016); Seo等人。(2017).BERT使用自我关注机制来统一这两个阶段，因为编码具有自我关注的连接文本对有效地包括两个句子之间的双向交叉关注。

对于每个任务，我们只需将特定于任务的输入和输出插入到BERT中，并对端到端的所有参数进行微调。在...
来自训练前的句子A和句子B
类似于（1）释义中的句子对，（2）蕴涵中的假设 - 前提对，（3）问题回答中的问题 - 通道对，以及

（4）文本分类或序列标记中的简并text-∅对。在输出处，令牌表示被馈送到输出层以用于令牌级任务，例如序列标记或问答，并且[CLS]表示被馈送到输出层以进行分类，例如蕴涵或情绪分析。

与预训练相比，微调相对便宜。本文中的所有结果可以在单个云TPU上最多1小时复制，或者在GPU上几小时复制，从完全相同的预训练模型开始。[7] 我们在Section的相应小节中描述了特定于任务的细节4.更多细节可以在附录中找到A.5.

4 实验

在本节中，我们将介绍11个NLP任务的BERT微调结果。

4.1 胶

一般语言理解评估（GLUE）基准（王等人。，2018a）是各种自然语言理解任务的集合。GLUE数据集的详细说明包含在附录中B.1.

为了微调GLUE，我们代表输入序列（对于单个句子或句子对），如章节中所述3，并使用与第一个对应的最终隐藏向量$C \in R^h$
输入令牌（[CLS]）作为聚合表示
灰。在微调期间引入的唯一新参数是分类层
权重$W \in R^{K \times H}$，其中K是标记的数量。我们com-
用C和W表示标准分类损失，
即log（softmax（CW ^{T} ））。

[7]例如，BERT SQuAD模型可以在单个云TPU上训练大约30分钟，以获得91.0%的Dev F1得分。

[8]见（10）inhttps://gluebenchmark.com/faq.

系统	MNLI-（米/毫米） 392k	QQP 363k	肯利 108k	SST-2 67k	可乐 8.5k	STS-B 5.7k	MRPC 3.5k	即食 2.5k	平均 -
预先开放的SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM + ELMO +经办人	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
伯特	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
伯特	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

表1：评估服务器评分的GLUE测试结果（https://gluebenchmark.com/leaderboard）.每个任务下面的数字表示训练样例的数量。“平均”列与官方GLUE分数略有不同，因为我们排除了有问题的WNLI集。[8] BERT和OpenAI GPT是单一模型，单一任务。报告QQP和MRPC的F1分数，报告STS-B的Spearman相关性，并报告其他任务的准确性分数。我们排除使用BERT作为其组件之一的条目。

我们使用批量大小为32并对所有GLUE任务的数据进行3个时期的微调。对于每项任务，我们在Dev set上选择了最佳的微调学习率（5e-5,4e-5,3e-5和2e-5）。此外，对于BERT大，我们发现微调有时在小数据集上不稳定，因此我们运行了几次随机重启并在Dev集上选择了最佳模型。通过随机重启，我们使用相同的预训练检查点，但执行不同的微调数据混洗和分类器层初始化。[9]

结果列于表中1. 都伯特 基础 和BERT 大 优于所有系统对所有任务进行大幅度的改进，相对于现有技术水平，相应的平均准确度提高了4.5%和7.0%。请注意，除了注意力掩蔽之外，BERT 基础 和OpenAI GPT在模型架构方面几乎相同。对于最大和最广泛报道的GLUE任务，MNLI，BERT获得4.6%的绝对精度改进。在官方的GLUE排行榜上[10]BERT大 得分为80.5，与获得的OpenAI GPT相比
截至编写之日为止。

我们发现BERT大 在所有任务中都明显优于BERT基础，特别是那些训练数据非常少的人。在Section中更深入地探讨了模型尺寸的影响5.2.

4.2 V1.1车队

斯坦福问题答疑数据集（SQuAD v1.1）是100k众包问答对的集合（Rajpurkar等人。，2016).给出一个问题和一段话维基百科包含答案，任务是预测文章中的答案文本跨度。

如图所示1在问题回答任务中，我们将输入问题和段落表示为单个打包序列，问题使用A嵌入和使用B嵌入的段落。我们仅在微调期间引入起始载体SR^h和末端载体ER^h。单词i作为答案跨度开始的概率计算为T_i 和S之后的一个点产品 $\in$ 然后是softmax over $\in$ 段落中的所有词语：P $i = \frac{e^{S \cdot T_i}}{\sum_j e^{S \cdot T_j}}$ 类似的公式用于答案范围的结束。从位置i到位置j的候选跨度的得分被定义为$ST_i + ET_j$，以及其中ji用作预测的最大得分跨度。≥训练目标是正确的开始和结束位置的对数似然的总和。我们对3个时期进行微调，学习率为5e-5，批量为32。

表2 显示顶级排行榜条目以及顶级发布系统的结果（Seo等人。，2017; 克拉克和加德纳，2018; 彼得斯等人。，2018a; 胡等人。，2018).SQuAD排行榜的最高结果没有最新的公共系统描述，[11] 并且在培训他们的系统时允许使用任何公共数据。因此，我们首先在TriviaQA上进行微调，在我们的系统中使用适度的数据增强（乔希 等。，2017）对SQUAD进行微调。

我们表现最佳的系统在整个排行榜中的表现优于顶级排行榜系统+1.5 F1
+1.3 F1作为单一系统。事实上，我们的单一BERT模型在F1得分方面优于顶级合奏系统。没有TriviaQA罚款 -

[9]GLUE数据集分布不包括测试标签，我们只制作了一个GLUE评估服务器
提交每个BERT基础 和BERT大。

[10]https://gluebenchmark.com/leaderboard

[11]QANet描述于Yu等人。(2018)，但该系统在出版后已大幅改善。

系统	开发		测试	
	EM	F1	相对长度单位 F1	
排行榜榜首（2018年12月10日）人类	-	82.3	91.2	
第1乐团-NLNET	-	-	86.0	91.7
第2乐团- QANet	-	-	84.5	90.5
发布时间				
BiDAF + ELMo（单）	-	85.6	-	85.8
RM阅读器（合奏）	81.2	87.9	82.3	88.5
我们的				
BERT基础（单）	80.8	88.5	-	-
BERT大（单）	84.1	90.9	-	-
BERT大（合奏）	85.8	91.8	-	-
BERT大（Sgl。+ TriviaQA）	84.2	91.1	85.1	91.8
BERT大（Ens。+ TriviaQA）	86.2	92.2	87.4	93.2

表2：SQuAD 1.1结果。BERT系列是7x系统，使用不同的训练前检查点和微调种子。

系统	开发		测试	
	EM	F1	相对长度单位 F1	
排行榜榜首（2018年12月10日）人类	86.3	89.0	86.9	89.5
γ 1单miR MRC（F网）	-	-	74.8	78.0
2单NLNET	-	-	74.2	77.1
发布时间				
unet（合奏）	-	-	71.4	74.9
SLQA +（单）	-		71.4	74.4
我们的				
BERT大（单人）	78.7	81.9	80.0	83.1

表3：SQuAD 2.0结果。我们排除使用BERT作为其组件之一的条目。

调整数据，我们只损失0.1-0.4 F1，仍然大幅超越所有现有系统。[12]

4.3 V2.0小队

SQuAD 2.0任务通过允许在提供的段落中不存在简短答案的可能性来扩展SQuAD 1.1问题定义，从而使问题更加真实。

我们使用一种简单的方法来扩展此任务的SQuAD v1.1 BERT模型。我们将没有答案的问题视为在[CLS]标记处具有开始和结束的答案范围。开始和结束答案跨度位置的概率空间被扩展为包括[CLS]标记的位置。为了预测，我们比较无答案跨度的得分：$s_{\text{空值}} =$ $S \cdot C + E \cdot C$得分为最佳非零跨度

[12]我们使用的TriviaQA数据包括来自TriviaQA-Wiki的段落，这些段落由文档中的前400个令牌组成，其中包含至少一个提供的可能答案。

系统	开发	测试
ESIM +手套	51.9	52.7
ESIM_ELMo	59.1	59.2
OpenAI GPT	-	78.0
伯特基础	81.6	-
伯特大	**86.6**	**86.3**
人（专家）†	-	
	85.0人	
类（5个注释）†	-	88.0

表4：SWAG开发和测试精度。†如SWAG论文所述，用100个样品测量人的表现。

$s_{i,j} = \max_{j \geq i} S \cdot T_i + E \cdot T_j$。当$s_{i,j} > s_{\text{空值}} + \tau$时，我们预测非空答案，其中在开发集上选择阈值$\tau$以使F1最大化。我们没有将TriviaQA数据用于此模型。我们对2个时期进行了微调，学习率为5e-5，批量为48。

结果与之前的排行榜参赛作品和最新出版的作品相比较Sun等人。，2018；王等人。，2018b）如表所示3，不包括使用BERT作为其中一个的系统

ponents。我们观察到+5.1 F1的改进以前最好的系统。

4.4 赃物

具有对抗性世代的情况（SWAG）数据集包含113k个句子对完成示例，用于评估基础常识推理（Zellers等。，2018）。给出一个句子，任务是在四个选择中选择最合理的延续。

在对SWAG数据集进行微调时，我们构造了四个输入序列，每个序列包含给定句子（句子）的串联A）和可能的继续（句子B）。引入的唯一任务特定参数是矢量，其具有[CLS]标记表示C的点积表示用softmax层标准化的每个选择的分数。

我们对模型进行了3个时期的微调，学习率为2e-5，批量大小为16。结果如表所示4。BERT大优于作者的基线ESIM + ELMo系统+ 27.1%，OpenAI GPT优于8.3%。

5 消融研究

在本节中，我们将对BERT的多个方面进行消融实验，以便更好地了解它们的相对重要性。额外

任务	开发集 MNLi-M (Acc)	QNLI (Acc)	MRPC (Acc)	SST-2 (Acc)	小班 (F1)
伯特	84.4	88.4	86.7	92.7	88.5
没有NSP	83.9	84.9	86.5	92.6	87.9
LTR与NSP	82.1	84.3	77.5	92.1	77.8
+比斯特	82.1	84.1	75.7	91.6	84.9

表5：使用BERT$_{基础}$ 架构消除预训练任务。没有下一句话预测任务就训练“没有NSP”。“LTR&No NSP”被训练为从左到右的LM而没有下一个句子预测，如OpenAI GPT。“+ BiLSTM”在微调期间在“LTR + No NSP”模型上添加随机初始化的BiLSTM。

消融研究可以在附录中找到C.

5.1 预训练任务的效果

我们通过使用与BERT$_{基础}$完全相同的预训练数据，微调方案和超参数来评估两个预训练目标，证明了BERT深度双向性的重要性：

无NSP：使用“屏蔽LM”（MLM）但没有“下一句预测”（NSP）任务训练的双向模型。

LTR&No NSP：仅使用左上下文的模型，使用标准的从左到右（LTR）LM而不是MLM进行训练。左侧约束也适用于微调，因为移除它会引入预列车/微调不匹配，从而降低下游性能。此外，该模型在没有NSP任务的情况下进行了预训练。这与OpenAI GPT直接相当，但使用我们更大的训练数据集，输入表示和我们的微调方案。

我们首先考察NSP任务带来的影响。在表中5，我们表明删除NSP会严重损害QNLI，MNLI和SQuAD 1.1的性能。接下来，我们通过比较“No NSP”与“LTR&No NSP”来评估训练双向表示的影响。在所有任务中，LTR模型的性能都比MLM模型差，在MRPC和SQuAD上有大幅下降。

对于SQuAD，直观清楚的是LTR模型在令牌预测时表现不佳，因为令牌级隐藏状态没有右侧上下文。为了真诚地尝试加强LTR系统，我们在顶部添加了一个随机初始化的BiLSTM。这确实显着改善了SQuAD的结果，但是结果仍然比预训练的双向模型差。BiLSTM会损害GLUE任务的性能。

我们认识到，也可以训练单独的LTR和RTL模型，并将每个标记表示为两个模型的串联，如ELMo所做的那样。但是：（a）这是单一双向模型的两倍；（b）对于像QA这样的任务来说，这是不直观的，因为RTL模型无法对问题的答案作出规定；（c）它的强度远低于深度双向模型，因为它可以在每一层使用左右上下文。

5.2 模型尺寸的影响

在本节中，我们将探讨模型大小对微调任务准确性的影响。我们训练了许多具有不同层数，隐藏单元和注意头的BERT模型，否则使用与前面描述的相同的超参数和训练过程。

选定的GLUE任务的结果显示在表中6. 在此表中，我们报告了5次随机重启微调的平均开发设置精度。我们可以看到，较大的模型导致所有四个数据集的严格精度提高，即使对于仅具有3,600个标记的训练样例的MRPC，并且与训练前任务有很大不同。同样令人惊讶的是，我们能够在相对于现有文献已经非常大的模型之上实现这种显着的改进。例如，最大的变形金刚探索过Vaswani等。（2017）（L = 6，H = 1024，A = 16），编码器有100M参数，我们在文献中找到的最大变压器是（L = 64，H = 512，A = 2），参数为235M（Al-Rfou等人。，2018）. 相比之下，BERT$_{基础}$ 包含110M参数，BERT$_{大}$ 包含340M参数。

人们早就知道，增加模型尺寸将导致机器翻译和语言建模等大规模任务的持续改进，这可以通过表中所示的LM延迟训练数据的复杂性来证明。6. 但是，我们认为，这是第一项令人信服地证明，如果模型已经过充分预先培训，那么扩展到极端模型尺寸也可以在非常小规模的任务上实现大幅改进。彼得斯等人。（2018b） 呈现

下游任务影响的混合结果将预训练的双LM尺寸从2层增加到4层Melamud等。（2016）顺便提到，隐藏的尺寸从200增加到600有助于增加，但进一步增加到1,000并没有带来进一步的改善。这两个先前的工作都使用了基于特征的方法 - 我们假设当模型直接在下游任务上进行微调并且仅使用非常少量的随机初始化的附加参数时，特定于任务的模型可以受益于更大的，即使下游任务数据非常小，也可以使用更具表现力的预训练表示。

5.3 基于特征的BERT方法

到目前为止所呈现的所有BERT结果都使用了微调方法，其中将简单的分类层添加到预训练模型，并且所有参数在下游任务上联合微调。然而，基于特征的方法具有某些优点，其中固定特征从预训练模型中提取。首先，并非所有任务都可以通过Transformer编码器体系结构轻松表示，因此需要添加特定于任务的模型体系结构。其次，预先计算训练数据的昂贵表示一次然后在该表示之上使用更便宜的模型进行许多实验具有主要的计算益处。在本节中，我们通过将BERT应用于CoNLL-2003命名实体识别（NER）任务来比较这两种方法（基姆演唱 和德梅尔德，2003）.在BERT的输入中，我们使用了一个保留大小写的WordPiece模型，并且我们包含了数据提供的最大文档上下文。按照标准做法，我们将其制定为标记任务，但不使用CRF

超帕拉斯				开发设置准确度		
#L	#H	#A	LM (ppl)	MNLI米	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

表6：BERT模型尺寸的烧蚀。#L =层数;#H =隐藏的大小;#A =关注头数量。“LM（ppl）”是保持训练数据的蒙面LM困惑。

系统	开发F1	测试F1
ELMo彼得斯等人。，2018a)	95.7	92.2
无级变速器克拉克等人。，2018)	-	92.6
CSE (Akbik等。，2018)	-	**93.1**
微调方法		
伯特大	96.6	92.8
伯特基础	96.4	92.4
基于特征的方法（BERT基础）		
的嵌入	91.0	-
倒数第二个隐藏	95.6	-
最后隐藏	94.9	-
加权和最后四个隐藏95.9	- 康卡特	
最后四个隐藏	96.1	
	- 加权总	
和所有12层	95.5	-

表7：CoNLL-2003命名实体识别结果。使用Dev set选择超参数。报告的开发和测试分数使用这些超参数在5次随机重启中取平均值。

输出中的图层。我们使用第一个子标记的表示作为NER标签集上的标记级分类器的输入。

为了消除微调方法，我们通过从一个或多个层中提取激活来应用基于特征的方法，而无需微调BERT的任何参数。这些上下文嵌入用作分类层之前的随机初始化的双层768维BiLSTM的输入。

结果列于表中7.BERT大 与最先进的方法竞争激烈。表现最佳的方法连接来自预训练变形金刚的前四个隐藏层的标记表示，这仅是0.3 F1后面对整个模型进行微调。这表明BERT对于微调和基于特征的方法都是有效的。

6 结论

由于使用语言模型进行转移学习，最近的经验改进表明，丰富的，无监督的预训练是许多语言理解系统的组成部分。特别是，这些结果使得即使是低资源任务也能从深度单向架构中受益。我们的主要贡献是将这些发现进一步推广到深度双向架构，允许相同的预训练模型成功解决一系列广泛的NLP任务。

参考

Alan Akbik，Duncan Blythe和Roland Vollgraf。2018.用于序列标记的上下文字符串嵌入。在第27届国际计算语言学会议论文集，第1638-1649页。

Rami Al-Rfou，Dokook Choe，Noah Constant，Mandy Guo和Llion Jones。2018.字符级语言建模，具有更深刻的自我关注。arXiv preprint arXiv：1808.04444。

Rie Kubota Ando和Tong Zhang。2005.从多个任务和未标记数据中学习预测结构的框架。机器学习研究杂志，6（11月）：1817-1853。

Luisa Bentivogli，Bernardo Magnini，Ido Dagan，Hoa Trang Dang和Danilo Giampiccolo。2009.第五届PASCAL承认文字蕴涵挑战。在TAC。NIST。

John Blitzer，Ryan McDonald和Fernando Pereira。2006.结构对应学习的领域适应。在2006年关于自然语言处理中经验方法会议的会议记录中，第120-128页。计算语言学协会。

Samuel R. Bowman，Gabor Angeli，Christopher Potts和Christopher D. Manning。2015.用于学习自然语言推理的大型注释语料库。在EMNLP中。计算语言学协会。

Peter F Brown，Peter V Desouza，Robert L Mercer，Vincent J Della Pietra和Jenifer C Lai。1992.基于类的自然语言n-gram模型。计算语言学，18（4）：467-479。

Daniel Cer，Mona Diab，Eneko Agirre，Inigo Lopez-Gazpio和Lucia Specia。2017年。SeValE-2017任务1：语义文本相似性多语言和跨领域的重点评估.*在第11届国际语义评估研讨会论文集（SemEval-2017），第1-14页，加拿大温哥华。计算语言学协会。*

Ciprian Chelba，Tomas Mikolov，Mike Schuster，Qi Ge，Thorsten Brants，Phillipp Koehn和Tony Robinson。2013.用于衡量统计语言建模进展的十亿字基准。arXiv preprint arXiv：1312.3005。

Z. Chen，H。Zhang，X。Zhang和L. Zhao。2018。Quora问题对.

克里斯托弗克拉克和马特加德纳。2018.简单有效的多段阅读理解。在ACL中。

Kevin Clark，Minh-Thang Luong，Christopher D Manning和Quoc Le。2018.采用交叉视图训练的半监督序列建模。在2018年自然语言处理经验方法会议论文集，第1914-1925页。

Ronan Collobert和Jason Weston。2008.自然语言处理的统一架构：具有多任务学习的深度神经网络。在第25届机器学习国际会议论文集，第160-167页。ACM。

Alexis Conneau，Douwe Kiela，Holger Schwenk，Lo¨ıcBarrault和Antoine Bordes。2017年。监督 从中学习通用句子表示 自然语言推断数据.在2017年自然语言处理经验方法会议论文集，第670-680页，丹麦哥本哈根。计算语言学协会。

Andrew M Dai和Quoc V Le。2015.半监督序列学习。在神经信息处理系统的进展中，第3079-3087页。

J.Deng，W。Dong，R。Socher，L.-J。Li，K。Li和L. FeiFei。ImageNet：一个大规模的分层图像数据库。在CVPR09。

William B Dolan和Chris Brockett。2005.自动构建句子释义语料库。在第三届国际释义研讨会论文集（IWP2005）中。

William Fedus，Ian Goodfellow和Andrew M Dai。2018. Maskgan：通过填写更好的文本生成。arXiv preprint arXiv：1801.07736。

Dan Hendrycks和Kevin Gimpel。2016年桥接 具有高斯的非线性和随机正则化器 sian错误线性单位.CoRR，abs / 1606.08415。

Felix Hill，Kyunghyun Cho和Anna Korhonen。2016.学习来自未标记数据的句子的分布式表示。在2016年计算语言学协会北美分会会议论文集：人类语言技术。计算语言学协会。

杰里米霍华德和塞巴斯蒂安罗德。2018。普遍 用于文本分类的语言模型微调.在ACL中。计算语言学协会。

胡明浩，彭宇兴，黄震，邱秋鹏，魏福茹，周明。2018.增强的记忆读取器，用于机器阅读理解。在IJCAI。

Yacine Jernite，Samuel R. Bowman和David Sontag。2017年。基于话语的目标，快速取消监督句子表示学习.CoRR，abs / 1705.00557。

Mandar Joshi, Eunsol Choi, Daniel S Weld和Luke Zettlemoyer。2017.Triviaqa：一个用于阅读理解的大规模远程监督挑战数据集。在ACL中。

Ryan Kiros，Yukun Zhu，Ruslan R Salakhutdinov，Richard Zemel，Raquel Urtasun，Antonio Torralba和Sanja Fidler。2015. Skip-thought vectors。在神经信息处理系统的进展中，第3294-3302页。

Quoc Le和Tomas Mikolov。2014.句子和文件的分布式表示。在国际机器学习会议，第1188-1196页。

Hector J Levesque，Ernest Davis和Leora Morgenstern。2011. winograd架构挑战。在Aaai春季研讨会上：常识推理的逻辑形式化，第46卷，第47页。

Lajanugen Logeswaran和Honglak Lee。2018。一个学习句子的有效框架代表 - 论.*在国际学习代表会议上。*

Bryan McCann，James Bradbury，Caiming Xiong和Richard Socher。2017.Learnned in translation：Contextualized word vectors。在NIPS。

Oren Melamud，Jacob Goldberger和Ido Dagan。2016. context2vec：学习使用双向LSTM的通用上下文嵌入。在CoNLL。

Tomas Mikolov，Ilya Sutskever，Kai Chen，Greg S Corrado和Jeff Dean。2013.单词和短语的分布式表示及其组合性。在神经信息处理系统的进展26，第3111-3119页。Curran Associates，Inc。

Andriy Mnih和Geoffrey E Hinton。2009年。一个scal-能够分层的分布式语言模型.在D.Koller，D.Schuurmans，Y.Bengio，和L.Bot-tou，编辑，神经信息处理系统的进展21，第1081-1088页。Curran Associates，Inc。

Ankur P Parikh，OscarTa¨ckstrom，Dipanjan Das和Jakob Uszkoreit。2016.自然语言推理的可分解注意模型。在EMNLP中。

Jeffrey Pennington，Richard Socher和Christopher D. Manning。2014年手套：全球媒介 单词表示.*在自然语言处理中的经验方法（EMNLP），第1532-1543页。*

Matthew Peters，Waleed Ammar，Chandra Bhagavatula和Russell Power。2017.使用双向语言模型的半监督序列标记。在ACL中。

Matthew Peters，Mark Neumann，Mohit Iyyer，Matt Gardner，Christopher Clark，Kenton Lee和Luke Zettlemoyer。2018A。深层语境化词语表示。在NAACL。

Matthew Peters，Mark Neumann，Luke Zettlemoyer和Wen-tau Yih。2018B。解剖上下文词嵌入：架构和表示。在2018年自然语言处理经验方法会议论文集，第1499-1509页。

Alec Radford，Karthik Narasimhan，Tim Salimans和Ilya Sutskever。2018.通过无监督学习提高语言理解能力。技术报告，OpenAI。

Pranav Rajpurkar，张健，Konstantin Lopyrev和Percy Liang。2016.Squad：机器理解文本的100,000多个问题。在2016年自然语言处理经验方法会议论文集，第2383-2392页。

Minjoon Seo，Aniruddha Kembhavi，Ali Farhadi和Hannaneh Hajishirzi。2017.双向关注流程，用于机器理解。在ICLR。

Richard Socher，Alex Perelygin，Jean Wu，Jason Chuang，Christopher D Manning，Andrew Ng和Christopher Potts。2013.针对情感树库的语义组合的递归深度模型。在2013年自然语言处理经验方法会议论文集，第1631-1642页。

傅弘，李林阳，邱培鹏，刘洋。2018. U-net：机器阅读理解与无法回答的问题。arXiv preprint arXiv：1810.06638。

威尔逊L泰勒。Cloze程序：一种测量可读性的新工具。新闻公报，30（4）：415-433。

Erik F Tjong Kim Sang和Fien De Meulder。2003. conll-2003共享任务简介：与语言无关的命名实体识别。在CoNLL。

Joseph Turian，Lev Ratinov和Yoshua Bengio。Word表示：半监督学习的简单通用方法。在计算语言学协会第48届年会论文集中，ACL '10，第384-394页。

Ashish Vaswani，Noam Shazeer，Niki Parmar，Jakob Uszkoreit，Llion Jones，Aidan N Gomez，Lukasz Kaiser和Illia Polosukhin。2017.注意力就是你所需要的。在神经信息处理系统的进展，第6000-6010页。

Pascal Vincent，Hugo Larochelle，Yoshua Bengio和Pierre-Antoine Manzagol。2008.使用去噪自动编码器提取和组合强大的功能。在第25届机器学习国际会议论文集，第1096-1103页。ACM。

Alex Wang，Amanpreet Singh，Julian Michael，Felix Hill，Omer Levy和Samuel Bowman。2018A。胶水：一个多任务基准和分析平台

自*然语言理解。在2018年EMNLP研讨会论文集BlackboxNLP：分析和解释NLP的神经网络，第353-355页。*

王伟，明艳，陈武。2018B。用于阅读理解和问答的多粒度分层注意融合网络。“计算语言学协会第56届年会论文集”（第1卷：长篇论文）。计算语言学协会。

Alex Warstadt，Amanpreet Singh和Samuel R Bowman。2018.神经网络可接受性判断。arXiv preprint arXiv：1805.12471。

Adina Williams，Nikita Nangia和Samuel R Bowman。2018.通过推理理解句子的广泛覆盖挑战语料库。在NAACL。

Wu Yonghui Wu，Mike Schuster，Zhifeng Chen，Quoc V Le，Mohammad Norouzi，Wolfgang Macherey，Maxim Krikun，Yuan Cao，Qin Gao，Klaus Macherey，et al。2016.谷歌的神经机器翻译系统：缩小人机翻译的差距。arXiv preprint arXiv：1609.08144。

Jason Yosinski，Jeff Clune，Yoshua Bengio和Hod Lipson。深度神经网络中的特征如何可转移？在神经信息处理系统的进展中，第3320-3328页。

Adams Wei Yu，David Dohan，Minh-Thang Luong，Rui Zhao，Kai Chen，Mohammad Norouzi和Quoc V Le。2018. QANet：将局部卷积与全球自我关注结合起来进行阅读理解。在ICLR。

Rowan Zellers，Yonatan Bisk，Roy Schwartz和Yejin Choi。2018. Swag：用于扎根常识推理的大型对抗数据集。在2018年自然语言处理经验方法会议（EMNLP）的会议记录中。

Yukun Zhu，Ryan Kiros，Rich Zemel，Ruslan Salakhut-dinov，Raquel Urtasun，Antonio Torralba和Sanja Fidler。2015.对齐书籍和电影：通过观看电影和阅读书籍来实现故事般的视觉解释。在IEEE国际计算机视觉会议论文集，第19-27页。

“BERT：用于语言理解的深度双向变压器的预训练”附录

我们将附录分为三个部分：

- BERT的其他实施细节见附录A;
- 我们实验的其他细节见附录B;和
- 其他消融研究见附录C.

我们为BERT提供了额外的消融研究，包括：

  - 训练步数的影响;和
  - 消融不同的掩蔽程序。

A BERT的其他详细信息

A.1 预训练任务的插图

我们提供以下预训练任务的示例。

蒙面LM和掩蔽程序假设未标记的句子是我的狗是多毛的，并且在随机掩蔽过程中我们选择了第4个标记（对应于多毛），我们的掩蔽过程可以进一步说明

- 80％的时间：用[MASK]令牌替换单词，例如，我的狗是多毛的→我的狗是[面具]
- 10％的时间：用一个随机的单词替换单词，例如，我的狗是多毛的→我的狗是苹果
- 10％的时间：保持单词不变，例如，我的狗毛茸茸→我的狗毛茸茸。这样做的目的是将表示偏向于实际观察到的单词。

此过程的优点是Transformer编码器不知道将要求预测哪些单词或哪些单词已被随机单词替换，因此强制保留每个输入标记的分布式上下文表示。此外，因为随机替换只发生在所有令牌的1.5％（即15％的10％），这似乎不会损害模型的语言理解能力。在节中C.2，我们评估这个程序的影响。

与标准语言模型训练相比，蒙版LM仅对每批中15％的代币进行预测，这表明模型可能需要更多的预训练步骤

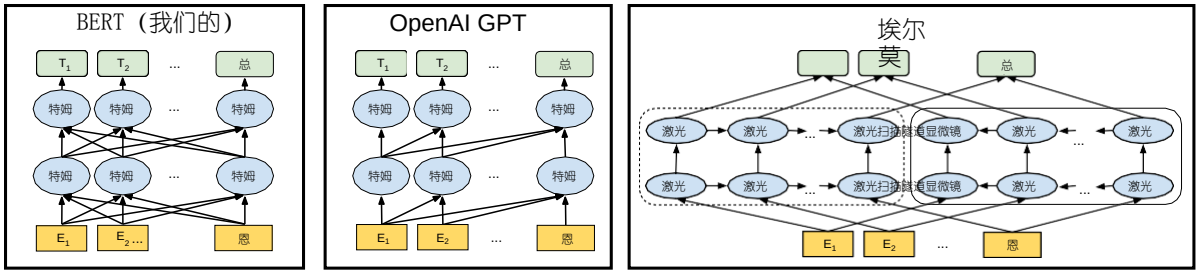

图3：训练前模型架构的差异。BERT使用双向变压器。OpenAI GPT使用从左到右的Transformer。ELMo使用经过独立训练的从左到右和右容忍LSTM的串联来生成下游任务的功能。在这三者中，只有BERT表示在所有层中共同依赖于左右上下文。除了架构差异之外，BERT和OpenAI GPT都是微调方法，而ELMo是一种基于特征的方法。

收敛。在节中C.1 我们证明MLM的收敛速度略慢于左右模型（预测每个标记），但MLM模型的经验改进远远超过增加的培训成本。

下一句子预测下一个句子预测任务可以在以下示例中说明。

输入= [CLS]男子去[MASK]商店[SEP]他买了一加仑[MASK]牛奶[SEP]

标签=下一个

输入= [CLS]男人[面具]到商店[SEP]企鹅[面具]是飞行##少鸟[SEP]

标签= NoT下一步

A.2 培训前程序

为了生成每个训练输入序列，我们从语料库中采样两个文本跨度，我们将其称为“句子”，即使它们通常比单个句子长得多（但也可以更短）。第一个句子接收A嵌入，第二个句子接收B嵌入。50%的时间B是跟随A的实际下一个句子，50%的时间是随机句子，这是为“下一句话预测”任务完成的。对它们进行采样，使得组合长度为512个令牌。在WordPiece标记化之后应用LM掩蔽，具有15%的统一掩蔽率，并且不特别考虑部分字块。

我们训练批量大小为256个序列（256个序列*512个令牌= 128,000个令牌/批次），持续1,000,000个步骤，大约40个超过33亿字语料库的时代。我们使用学习率为1e-4，$\beta_1 = 0.9$，$\beta_2 = 0.999$，L2权重衰减为0.01，学习率超过前10,000步的学习率和学习率的线性衰减的Adam。我们在所有层上使用0.1的丢失概率。我们使用gelu激活（亨德利克斯和金佩尔，2016而不是标准的relu，遵循OpenAI GPT。训练损失是平均掩蔽的LM可能性和平均下一句子预测可能性的总和。

在Pod配置中的4个云TPU上进行$\text{BERT}_{\text{基础}}$的训练（总共16个TPU芯片）。[13] 在16个云TPU（总共64个TPU芯片）上进行$\text{BERT}_{\text{大}}$的培训。每次预训练需要4天才能完成。

较长的序列不成比例地昂贵，因为注意力是序列长度的二次方。为了加速我们的实验中的预先训练，我们预先训练序列长度为128的模型，用于90%的步骤。然后，我们训练其余10%的512序列步骤来学习位置嵌入。

A.3 微调程序

对于微调，大多数模型超参数与预训练相同，但批量大小，学习率和训练时期数除外。辍学概率始终保持在0.1。最佳超参数值是特定于任务的，但我们发现以下范围的可能值可以在所有任务中很好地工作：

- **批量**：16,32

[13] https://cloudplatform.googleblog.com/2018/06/Cloud-TPU-现已提供，抢占定价及全局-availability.html

- 学习率（亚当）：5e-5,3e-5,2e-5
- 时代数：2,3,4

我们还观察到，大数据集（例如，100k +标记的训练示例）对超参数选择的敏感性远小于小数据集。微调通常非常快，因此简单地对上述参数进行详尽搜索并选择在开发集上表现最佳的模型是合理的。

A.4 BERT，ELMo和OpenAI GPT的比较

在这里，我们研究了最近流行的表示学习模型的差异，包括ELMo，OpenAI GPT和BERT。模型体系结构之间的比较在图中以可视方式显示3. 请注意，除了体系结构差异之外，BERT和OpenAI GPT都是微调方法，而ELMo是一种基于特征的方法。

与BERT最具可比性的现有预训练方法是OpenAI GPT，它在大型文本语料库中训练从左到右的Transformer LM。实际上，BERT中的许多设计决策都是为了使其尽可能接近GPT，因此可以将两种方法进行最低限度的比较。这项工作的核心论点是章节中提出的双向性和两个预训练任务3.1 考虑到大多数经验改进，但我们注意到BERT和GPT如何训练之间存在其他一些差异：

- GPT接受了BooksCorpus（800M字）的培训;BERT受过BooksCor-pus（800M字）和维基百科（2,500M字）的培训。
- GPT使用句子分隔符（[SEP]）和分类符标记（[CLS]），它们仅在微调时引入;BERT在预训练期间学习[SEP]，[CLS]和句子A / B嵌入。
- GPT接受了1M步骤的培训，批量为32,000字;BERT经过1M步骤的培训，批量为128,000字。
- 对于所有微调实验，GPT使用相同的5e-5学习率;BERT选择特定于任务的微调学习速率，该速率在开发集上表现最佳。

为了分离这些差异的影响，我们在Section中进行消融实验5.1 这表明大多数改进实际上来自两个预训练任务和它们实现的双向性。

A.5 微调不同任务的插图

可以在图中看到对不同任务的微调BERT的说明4. 我们的任务特定模型是通过将BERT与一个额外的输出层相结合而形成的，因此需要从头开始学习最少数量的参数。在这些任务中，（a）和（b）是序列级任务，而（c）和（d）是令牌级任务。在该图中，E表示输入嵌入，T_i 表示令牌i的上下文表示，[CLS]是用于分类输出的特殊符号，[SEP]是用于分离非连续令牌序列的特殊符号。

B 详细的实验设置

B.1 GLUE基准实验的详细描述。

我们的GLUE结果见表1 来自 https://gluebenchmark.com/排行榜 和 https://blog. 开放式语言/无监督语言.GLUE基准包括以下数据集，其描述最初总结于王等人。(2018a):

MNLI多类型自然语言推理是一项大规模的众包蕴涵分类任务（威廉姆斯等人。，2018).给定一对句子，目标是预测第二句话是否与第一句相关是蕴涵，矛盾或中立。

QQP Quora问题对是一个二进制分类任务，其目的是确定在Quora上提出的两个问题是否在语义上是等价的（陈等人。，2018).

QNLI问题自然语言推理是斯坦福问题答疑数据集的一个版本（Rajpurkar等人。，2016）已转换为二进制分类任务（王 等。，2018a).积极的例子是（问题，句子）对包含正确的答案，而负面的例子是（问题，句子）来自同一段，不包含答案。

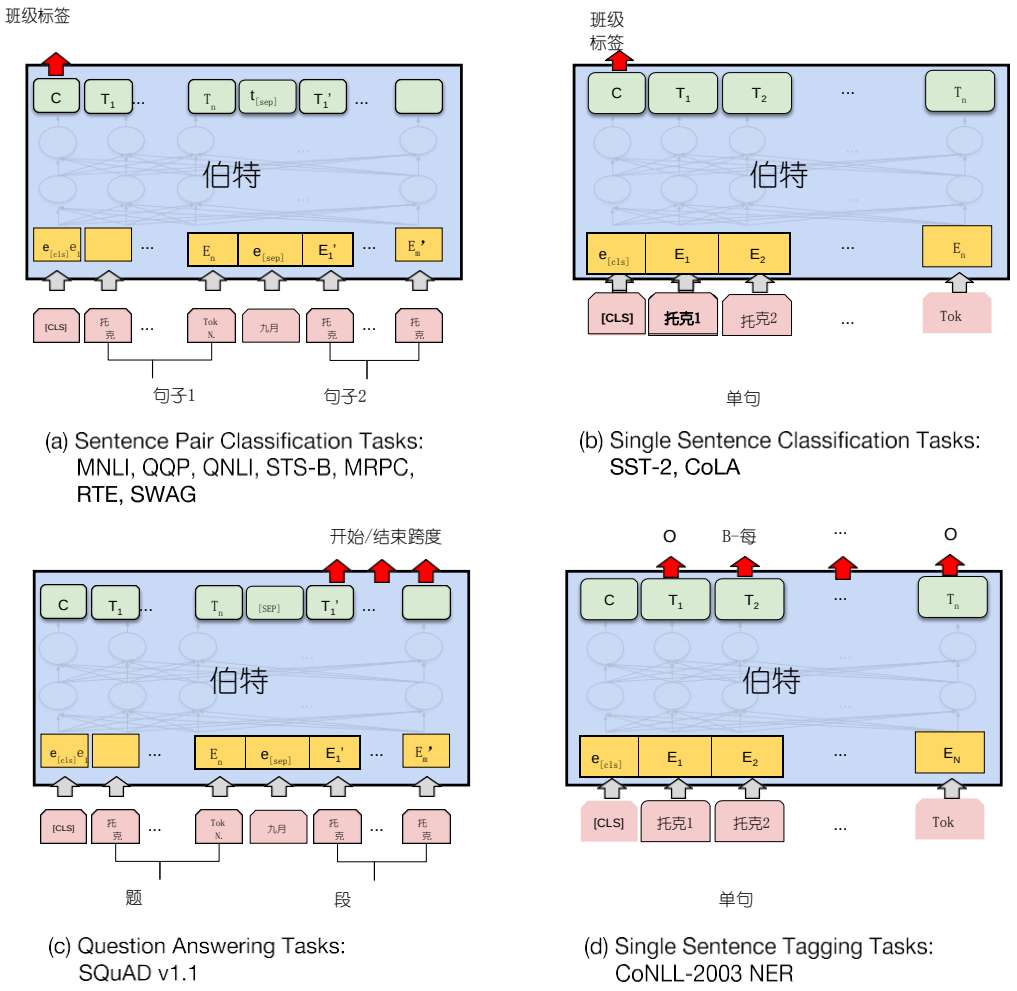

图4：在不同任务上微调BERT的插图。

SST-2斯坦福情感树库是一个二进制单句分类任务，由从电影评论中提取的句子和人类注释的情感组成（索赫尔 等。，2013).

CoLA语言可接受语料库是一个二进制单句分类任务，其目标是预测英语句子在语言上是否“可接受”（沃斯塔特 等。，2018).

STS-B语义文本相似性基准是从新闻标题和其他来源中提取的句子对的集合（Cer等人。，2017)．他们的评分为1分

5表示两个句子的相似程度
语义的术语。

MRPC Microsoft Research Disphrase Corpus包含从在线新闻源中自动提取的句子对，带有人类注释

对于该对中的句子是否在语义上是等价的（多兰和布罗基特，2005).

RTE识别文本蕴涵是类似于MNLI的二进制蕴涵任务，但训练数据少得多（Bentivogli等。，2009).[14]

WNLI Winograd NLI是一个小型自然语言推理数据集（Levesque等。，2011)．GLUE网页指出构建此数据集存在问题，[15] 并且每个已提交给GLUE的训练系统的表现都比预测大多数班级的65.1基线准确度差。因此，我们将此集排除在OpenAI GPT之外。对于我们的GLUE提交，我们总是预测ma-

[14]请注意，我们仅在本文中报告单任务微调结果。多任务微调方法可能会进一步提升性能。例如，我们确实观察到使用MNLI进行多任务培训的RTE的实质性改进。

[15]https://gluebenchmark.com/faq

jority class。

C 额外的消融研究

C.1 训练步数的影响

数字5 从已经预训练了k步的检查点进行微调后，显示MNLI Dev精度。这使我们可以回答以下问题：

1. 题： BERT真的需要这样吗？
   大量的预训练（128,000字/批*
   1,000,000步）来实现
   高微调精度？
   答：是的，$\text{BERT}_{\text{基础}}$ 在1M步骤训练后，对MNLI的准确度几乎提高1.0%，相比500k步。

2. 问题：MLM预训练比LTR预训练收敛慢，
   因为只有15%
   在每批中预测单词而不是每个单词？
   答案：MLM模型的收敛速度略慢于LTR模型。然而，就绝对精确度而言，MLM模型几乎立即开始优于LTR模型。

C.2 消融不同的掩蔽程序

在节中3.1，我们提到BERT使用混合策略在使用掩码语言模型（MLM）目标进行预训练时屏蔽目标令牌。以下是评估不同掩蔽策略效果的消融研究。

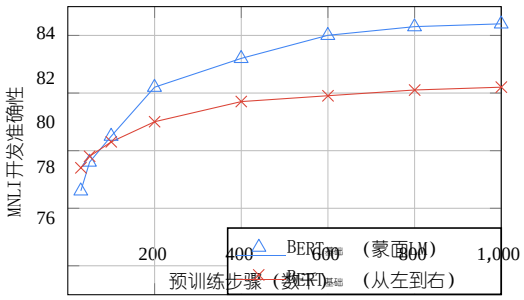

图5：多次训练步骤的消融。这显示了微调后的MNLI精度，从已经预训练了k步的模型参数开始。x轴是k的值。

请注意，屏蔽策略的目的是减少预训练和微调之间的不匹配，因为在微调阶段永远不会出现[MASK]符号。我们报告了MNLI和NER的Dev结果。对于NER，我们报告了微调和基于特征的方法，因为我们预计基于特征的方法将会放大不匹配，因为模型将无法调整表示。

掩蔽率			开发结果		
面具一样		隆德	姆恩利	纳	
			微调基于特征的微调		
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

表8：不同掩蔽策略的消融。

结果列于表中8. 在表中，MASK意味着我们用MLM的[MASK]符号替换目标令牌；SAME意味着我们保持目标令牌不变；RND意味着我们用另一个随机令牌替换目标令牌。

表格左侧的数字代表MLM预训练期间使用的特定策略的概率（BERT使用80%，10%，10%）。本文的右侧部分代表了Dev set结果。对于基于特征的方法，我们将最后4层BERT连接为特征，这被证明是剖面中的最佳方法5.3.

从表中可以看出，微调对于不同的掩蔽策略是惊人的稳健的。但是，正如预期的那样，在将基于特征的方法应用于NER时，仅使用MASK策略是有问题的。有趣的是，仅使用RND策略也比我们的策略更糟糕。