**BERT: Pre-training of Deep Bidirectional Transformers for**

**Language Understanding**

**Bert: 用于语言理解的预训练深度双向Transformer**

Jacob Devlin

Ming-Wei Chang Kenton Lee Kristina Toutanova

Google AI Language

[{jacobdevlin,mingweichang,kentonl,kristout}@google.com](mailto:{jacobdevlin,mingweichang,kentonl,kristout}@google.com)

## Abstract 摘要

We introduce a new language representation model called BERT, which stands for Bidirectional Encoder Representations from Transformers. Unlike recent language representation models (Peters et al., 2018; Radford et al., 2018), BERT is designed to pre-train deep bidirectional representations by jointly

conditioning on both left and right context in all layers. As a result, the pre-trained BERT representations can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such

as question answering and language inference, without substantial ask-specific architecture modifications.

本文介绍了一种新的语言表示模型BERT，它是（Bidirectional Encoder Representations from Transformers基于Transformer 的双向编码表征器）的缩写。与最近的语言表示模型(Peters等人，2018;Radford等人，2018)不同，BERT被设计为在所有层中对左-上下文和右-上下文进行联合训练进行深度双向表示的预训练。因此，经过预先训练的BERT表示可以通过仅仅一个额外的输出层进行微调，从而为广泛的任务(如问答和语言推理)创建最先进的模型，而无需对特定任务的体系结构进行实质性的修改。

BERT is conceptually simple and empirically powerful. It obtains new state-of-the-art results on eleven natural language processing tasks, including pushing the GLUE benchmark to 80.4% (7.6% absolute improvement), MultiNLI accuracy to 86.7% (5.6% absolute improvement) and the SQuAD v1.1 question answering Test F1 to 93.2 (1.5 absolute improvement), outperforming human performance by 2.0.

Bert在概念上简单，在经验上强大。该系统在11个自然语言处理任务中取得了新的研究成果，其中GLUE基准值达到80.4%(绝对提高7.6%)，MultiNLI 准确度达到86.7%(绝对提高5.6%)，SQuAD v1.1问答测试F1达到93.2%(绝对提高1.5%)，比人类表现提高2.0%。

## **1 Introduction** 引言

Language model pre-training has shown to be effective for improving many natural language processing tasks (Dai and Le, 2015; Peters et al., 2017, 2018; Radford et al., 2018; Howard and Ruder, 2018). These tasks include sentence-level tasks such as natural language inference (Bowman et al., 2015; Williams et al., 2018) and paraphrasing (Dolan and Brockett, 2005), which aim

to predict the relationships between sentences by analyzing them holistically, as well as token-level tasks such as named entity recognition (Tjong

Kim Sang and De Meulder, 2003) and SQuAD question answering (Rajpurkar et al., 2016), where models are required to produce fine-grained output at the token-level.

语言模型训练已经被证明是改善许多自然语言处理任务的有效方法(2018)。这些任务包括句子层面的任务，如自然语言推理(2015); 和释义(Dolan，2005)，旨在通过全面分析预测句子之间的关系，以及象征层面的任务，如命名实体识别(2003)和 SQuAD问题解答(2016)，这几个模型需要产生输入符号级别（字或词）的细粒度输出。

There are two existing strategies for applying pre-trained language representations to downstream tasks: feature-based and fine-tuning. The feature-based approach, such as ELMo (Peters et al., 2018), uses tasks-specific architectures that include the pre-trained representations as additional features. The fine-tuning approach, such as the Generative Pre-trained Transformer (OpenAI GPT) (Radford et al., 2018), introduces minimal task-specific parameters, and is trained on the downstream tasks by simply fine-tuning the pre-trained parameters. In previous work, both approaches share the same objective function during pre-training, where they use unidirectional language models to learn general language representations.

现有的将预先训练好的语言表示应用到下游任务的策略有两种:基于特征的和微调的。基于特征的方法，如ELMo(2018)，使用特定于任务的体系结构，它将预先训练好的表示形式作为附加特征(例如加到glove词向量上)。微调方法，如生成式预训练Transformer (OpenAIGPT)(2018)，引入了最小特定任务参数，并通过简单的预训练参数微调来训练下游任务。在以往的研究中，两种方法在前期训练中都使用了相同的目标函数，即使用单向语言模型来学习通用的语言表述。

We argue that current techniques severely restrict the power of the pre-trained representations,especially for the fine-tuning approaches. The major limitation is that standard language models are unidirectional, and this limits the choice of architectures that can be used during pre-training. For example, in OpenAI GPT, the authors use a left-to-right architecture, where every token can only attended to previous tokens in the self-attention layers of the Transformer (Vaswani et al., 2017). Such restrictions are sub-optimal for sentence-level tasks, and could be devastating when applying fine-tuning based approaches to token-level tasks such as SQuAD question answering (Ra

jpurkar et al., 2016), where it is crucial to incorporate context from both directions.

我们认为当前的技术严重限制了预训练表示的能力，特别是对于微调方法。其局限性在于标准的语言模型是单向的，这就限制了预训练可以使用的体系结构的选择。例如，在OpenAIGPT中，作者使用一种从左到右的架构，其中每个令牌只能处理Transformer中自注意层中的先前令牌(Vaswanietal.2017年)。这种限制对于句子层面的任务来说不是最好的，并且当应用基于微调的方法来处理输入符号级别的任务时可能是毁灭性的，比如SQuAD问答(Ra-jpurkaretal.，2016)，这对于从两个方向切入或分析上下文是至关重要的。

In this paper, we improve the fine-tuning based approaches by proposing BERT: Bidirectional Encoder Representations from Transformers. BERT addresses the previously mentioned uni-directional constraints by proposing a new pre-training objective: the “masked language model” (MLM), inspired by the Cloze task (Taylor, 1953). The masked language model randomly masks some of the tokens from the input, and the objective is to predict the original vocabulary id of the masked word based only on its context. Unlike left-to-right language model pre-training, the MLM objective allows the representation to fuse the left and the right context, which allows us to pre-train a deep bidirectional Transformer. In addition to the masked language model, we also

introduce a “next sentence prediction” task that jointly pre-trains text-pair representations.

在本文中，我们通过提出BERT：基于Transformer 的双向编码器表示来改进基于微调的方法。 BERT通过提出一个新的预训练目标来解决前面提到的单向约束：“遮罩语言模型”（MLM），受到完形任务的启发（Taylor，1953）。 遮罩语言模型在输入中随机地掩盖一些标记，目标是仅基于其上下文来预测被掩盖位置的原始词汇id。 与从左到右的语言模型预训练不同，MLM目标允许嵌入表示融合左右的上下文，这允许我们预训练深度的双向Transformer 。 除了遮罩语言模型，我们也引入“下一句预测”任务，用于联合预训练句子对的嵌入表示。

The contributions of our paper are as follows: 本文的贡献如下:

* We demonstrate the importance of bidirectional pre-training for language representations. Unlike Radford et al. (2018), which uses unidirectional language models for pre-training, BERT uses masked language models to enable pre-trained deep bidirectional representations. This is also in contrast to Peters et al. (2018), which uses a shallow concatenation of independently trained left-to-right and right-to-left LMs.

我们证明了双向预训练对语言表征的重要性。不同于Radford等人(2018年)，使用单向语言模型进行预训练，BERT使用语言模型来支持预先训练的深度双向表示。这也是相反的彼得斯等人(2018年)，其中使用浅层连接独立训练左至右和右至左LMs。

* We show that pre-trained representations eliminate the needs of many heavily-engineered task-specific architectures. BERT is the first fine-tuning based representation model that achieves state-of-the-art performance on a large suite of sentence-level and token-level tasks, outperforming many systems with task-specific architectures.

我们表明，经过预先训练的表示方法消除了许多重度设计的特定于任务的体系结构的需求。Bert是第一个基于微调的表示模型，它可以在大型的句子级和令牌级任务上实现最先进的性能，在任务特定的架构上超越了许多系统。

* BERT advances the state-of-the-art for eleven NLP tasks. We also report extensive ablations of BERT, demonstrating that the bidirectional nature of our model is the single most important new contribution. The code and pre-trained model will be available at goo.gl/language/bert. 1

我们表明，经过预先训练的表示方法消除了许多重度设计的特定于任务的体系结构的需求。Bert是第一个基于微调的表示模型，它可以在大型的句子级和令牌级任务上实现最先进的性能，在任务特定的架构上超越了许多系统。

1 Will be released before the end of October 2018.

## **2 Related Work** 相关工作

There is a long history of pre-training general language representations, and we briefly review the most popular approaches in this section.

预训练通用语言表达方式有着悠久的历史，我们在这一部分简要回顾一下最流行的方式。

### **2.1 Feature-based Approaches** 基于特征的方法

Learning widely applicable representations of words has been an active area of research for decades, including non-neural (Brown et al., 1992; Ando and Zhang, 2005; Blitzer et al., 2006) and neural (Collobert and Weston, 2008; Mikolov et al., 2013; Pennington et al., 2014) methods. Pre-trained word embeddings are considered to be an integral part of modern NLP systems, offering significant improvements over embeddings learned from scratch (Turian et al., 2010).

学习广泛应用表征的词已经成为一个活跃的研究几十年，包括非神经(布朗等，1992年;2014)methods，2008。预训练单词嵌入被认为是现代NLP系统的一个必要组成部分，提供了从0学习嵌入的重大改进(图里安等人，2010年)。

These approaches have been generalized to coarser granularities, such as sentence embeddings (Kiros et al., 2015; Logeswaran and Lee, 2018) or paragraph embeddings (Le and Mikolov, 2014). As with traditional word mbeddings, these learned representations are also typically used as features in a downstream model.

这些方法被推广到更粗糙的粒度，如句子嵌入(Kirosetal.，2015;Logeswaran和Lee，2018)或段落嵌入(LeandMikolov，2018)这些方法被推广到更粗糙的粒度，如句子嵌入(Kirosetal.，2015;Logeswaran和Lee，2018)或段落嵌入(LeandMikolov，2018)

ELMo (Peters et al., 2017) generalizes traditional word embedding research along a different dimension. They propose to extract context-sensitive features from a language model. When integrating contextual word embeddings with ex-

isting task-specific architectures, ELMo advances the state-of-the-art for several major NLP benchmarks (Peters et al., 2018) including question an-

swering (Rajpurkar et al., 2016) on SQuAD, sentiment analysis (Socher et al., 2013), and named entity recognition (Tjong Kim Sang and De Meulder, 2003).

Elmo (Peters 等人，2017)将传统的嵌入式研究从一个不同的维度进行了推广。 他们建议从语言模型中提取上下文敏感的特征。 当整合上下文词语嵌入到特定任务架构中时，ELMo 推进了几个主要 NLP 基准测试的最佳分数(Peters 等人，2018) ，包括在 SQuAD 上的提问(Rajpurkar 等人，2016)、情感分析(Socher 等人，2013)和命名实体识别(Tjong Kim Sang 和 De Meulder，2003)。

### **2.2 Fine-tuning Approaches** 微调方法

A recent trend in transfer learning from language models (LMs) is to pre-train some model architecture on a LM objective before fine-tuning that same model for a supervised downstream task (Dai and Le, 2015; Howard and Ruder, 2018;

Radford et al., 2018). The advantage of these approaches is that few arameters need to be learned from scratch. At least partly due this advantage, OpenAI GPT (Radford et al., 2018) achieved previously state-of-the-art results on many sentence-level tasks from the GLUE benchmark (Wang et al., 2018).

从语言模型(LMs)学习的一个最近的趋势是预先对LM目标进行一些模型构造，然后对监督下游任务的同一模型进行微调(Dai和Le，2015;Howard和Ruder，2018;Radford等，2018)。这些应用程序的优点是很少有参数需要从头学习。OpenaiGPT(Radfordetal.2018)至少部分得益于这一优势，它在GLUE基准测试(Wangetal.2018)的许多句子级任务中取得了前所未有的最先进成果。

### **2.3 Transfer Learning from Supervised Data** 监督数据的迁移学习

While the advantage of unsupervised pre-training is that there is a nearly unlimited amount of data available, there has also been work showing effective transfer from supervised tasks with large datasets, such as natural language inference (Conneau et al., 2017) and machine translation (McCann et al., 2017). Outside of NLP, computer vision research has also demonstrated the importance of transfer learning from large pre-trained models, where an effective recipe is to fine-tune models pre-trained on ImageNet (Deng et al.,

2009; Yosinski et al., 2014).

虽然无监督预训练的优点是几乎有无限的数据可用，也有工作显示有效的转移监督任务与大数据集，如自然语言推理和机器翻译。在NLP之外，计算机视觉研究也证明了从大型预先训练的模型进行迁移学习的重要性，其中一个有效的方法是微调模型预训练在ImageNet(邓等人，2009年;约西纽斯基等人，2014年)。

## 3 BERT

We introduce BERT and its detailed implementation in this section. We first cover the model architecture and the input representation for BERT. We then introduce the pre-training tasks, the core innovation in this paper, in Section 3.3. The pre-training procedures, and fine-tuning procedures are detailed in Section 3.4 and 3.5, respectively. Finally, the differences between BERT and OpenAI GPT are discussed in Section 3.6.

本节介绍了BERT技术及其具体实现。我们首先介绍了模型的体系结构和BERT输入表示。然后在第3.3节中介绍了本文的核心创新——培训任务。培训前的程序和调整过程在第3.4和3.5节详细说明。最后，在第3.6节中讨论了BERT技术和OpenAIGPT技术之间的差异。

### **3.1 Model Architecture** 模型架构

BERT’s model architecture is a multilayer bidirectional Transformer encoder based on the original implementation described in Vaswani et al.(2017) and released in the tensor2tensor library. 2 Because the use of Transformers has become ubiquitous recently and our implementation is effectively identical to the original, we will omit an exhaustive background description of the model architecture and refer readers to Vaswani et al. (2017) as well as excellent guides such as “The Annotated Transformer.” 3

Bert的模型架构是一个基于Vaswani等人(2017年)所描述的原始实现的多层bidi-rectional编码器，并在tensor2tensorli-brary.2中发布。由于Transformer 的使用最近已经无处不在，我们的实现实际上与原始模型相同，我们将省略模型架构的详尽的背景描述，并将读者介绍给Vaswani等人(2017年)，以及优秀的指南，如《Transformer 注释》

In this work, we denote the number of layers (i.e., Transformer blocks) as L, the hidden size as H, and the number of self-attention heads as A. In all cases we set the feed-forward/filter size to be 4H, i.e., 3072 for the H = 768 and 4096 for the H = 1024. We primarily report results on two model sizes:

在这项工作中，我们表示的数量层(即，Transformer 块)为l，隐藏的大小为h，和数量的自我注意头为a。在所有情况下，我们都将前向/过滤大小设置为4H，即h768的3072和h1024的4096。我们主要报告两种型号的测试结果:

• BERTBASE : L=12, H=768, A=12, Total Parameters=110M

• BERTLARGE : L=24, H=1024, A=16, Total Parameters=340M

BERT BASE was chosen to have an identical model size as OpenAI GPT for comparison purposes. Critically, however, the BERT Transformer uses bidirectional self-attention, while the GPT Transformer uses constrained self-attention where every token can only attend to context to its left.

We note that in the literature the bidirectional Transformer is often referred to as a “Transformer encoder” while the left-context-only version is referred to as a “Transformer decoder” since it can be used for text generation. The comparisons between BERT, OpenAI GPT and ELMo are shown visually in Figure 1.

Bertcabase被选为有一个相同的模型尺寸作为比较密集型openaigpt。然而，关键的是，BERTTransformer使用双向自注意，而GPTTransformer使用受限的自注意，每个令牌符号只能注意左边的上下文。我们注意到在文献中双向转换器经常被称为"Transformerencoder"，而只有左上下文的版本则被转换为"Transformerdecoder"，因为它可以用于文本生成。Bert、OpenAIGPT和ELMo之间的比较如图1所示。

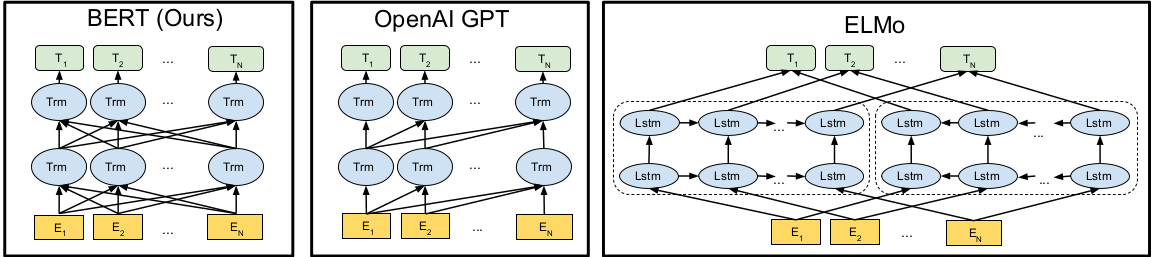


Figure 1: Differences in pre-training model architectures. BERT uses a bidirectional Transformer. OpenAI GPT uses a left-to-right Transformer. ELMo uses the concatenation of independently trained left-to-right and right-to-left LSTM to generate features for downstream tasks. Among three, only BERT representations are jointly conditioned on both left and right context in all layers. 图1:培训前模型架构的差异。使用一个双向Transformer 。OpenaiGPT使用的是一个从左到右的Transformer。Elmo使用独立培训的左到右和右到左LSTM的串联来生成下游任务的特性。在三个层次中，只有BERT表示法在所有层次中都受到左右上下文的共同制约。

### **3.2 Input Representation** 输入表示法

4 Throughout this work, a “sentence” can be an arbitrary span of contiguous text, rather than an actual linguistic sentence. A “sequence” refers to the input token sequence to BERT, which may be a single sentence or two sentences

packed together. 本论文中，一个"句子"可以是任意跨度的连续文本，而不是实际的语法上的句子概念（就是可能包涵多个句号）。一个"序列"指的是输入BERT的标记序列，这可能是一个单句或两个句子打包在一起。

Our input representation is able to unambiguously represent both a single text sentence or a pair of text sentences (e.g., [Question, Answer]) in one token sequence. 4 For a given token, its input representation is constructed by summing the corresponding token, segment and position embeddings. A visual representation of our input representation is given in Figure 2.

我们的输入表示能够在一个记号序列中清楚地表示一个单独的文本句子或一对文本句子(例如，[Question，Answer]])。对于给定的标记，其输入表示通过相应的标记，段落和位置嵌入求和来构造。我们的输入表示的直观表示如图2所示。

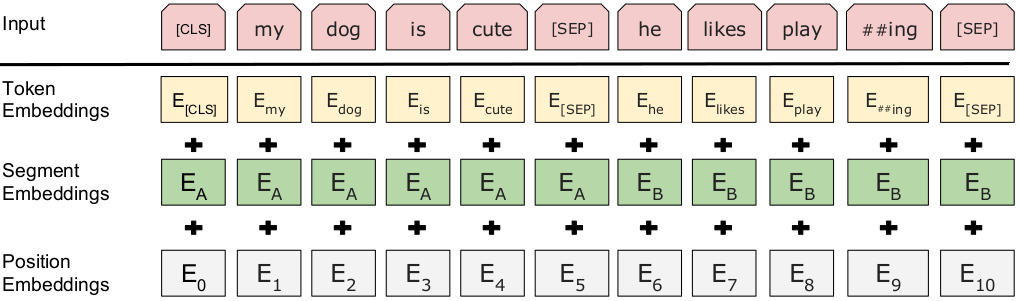


Figure 2: BERT input representation. The input embeddings is the sum of the token embeddings, the segmentation embeddings and the position embeddings. 图2:BERT输入表示法。输入嵌入是令牌嵌入、分段嵌入和位置嵌入的总和。

The specifics are: 具体内容如下:

* We use WordPiece embeddings (Wu et al., 2016) with a 30,000 token vocabulary. We denote split word pieces with ##. 我们使用WordPiece嵌入（Wu et al。，2016）和30,000个令牌词汇表。 我们用##表示分词。
* We use learned positional embeddings with supported sequence lengths up to 512 tokens. 我们使用学习的位置嵌入，支持的序列长度最多为512个令牌。
* The first token of every sequence is always the special classification embedding ([CLS]). The final hidden state (i.e., output of Transformer) corresponding to this token is used as the aggregate sequence representation for classification tasks. For non classification tasks, this vector is ignored. 每个序列的第一个标记是特殊的分类嵌入([CLS])。与之对应的最终隐藏状态(即Transformer的out-put)用作分类任务的聚合序列表示。对于非分类任务，此向量将被忽略。
* Sentence pairs are packed together into a single sequence. We differentiate the sentences in two ways. First, we separate them with a special token ([SEP]). Second, we add a learned sentence A embedding to every token of the first sentence and a sentence B embedding to every token of the second sentence. 成对的句子被打包成一个单序列。我们用两种方法区分这些句子。首先，我们用一个特殊的辅币([SEP])来分离它们。其次，我们在第一个句子的每个标记中加入一个学习句子a嵌入，在第二个句子的每个标记中加入一个句子b嵌入。
* For single-sentence inputs we only use the sentence A embeddings. 对于单句输入，我们只使用a语句嵌入。

### 3.3 Pre-training Tasks

Unlike Peters et al. (2018) and Radford et al. (2018), we do not use traditional left-to-right or right-to-left language models to pre-train BERT. Instead, we pre-train BERT using two novel unsupervised prediction tasks, described in this section. 不像Peters等人(2018)和Radford等人(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先训练BERT。相反，我们使用两个新颖的非监督预测任务对BERT进行预训练，本节将对此进行描述。

**3.3.1 Task #1: Masked LM 任务1：遮罩语言模型**

Intuitively, it is reasonable to believe that a deep bidirectional model is strictly more powerful than either a left-to-right model or the shallow concatenation of a left-to-right and right-to-left model. Unfortunately, standard conditional

language models can only be trained left-to-right or right-to-left, since bidirectional conditioning would allow each word to indirectly “see itself”

in a multi-layered context.

直观地说，有理由相信深度双向模型比左向右模型或左向右和右向左模型的浅连接更强大。遗憾的是，标准的条件语言模型只能进行从左到右或从右到左的训练，因为双向映射将允许每个单词在多层的上下文里间接地"看到自己"。

In order to train a deep bidirectional representation, we take a straightforward approach of masking some percentage of the input tokens at random, and then predicting only those masked tokens. We refer to this procedure as a “masked LM” (MLM), although it is often referred to as a Cloze task in the literature (Taylor, 1953). In this case, the final hidden vectors corresponding to the mask tokens are fed into an output softmax over the vocabulary, as in a standard LM. In all of our experiments, we mask 15% of all WordPiece tokens in each sequence at random. In contrast to denoising auto-encoders (Vincent et al., 2008), we only predict the masked words rather than reconstructing the entire input.

为了训练深度双向表示，我们采用直接的方法随机遮罩一定比例的输入令牌，然后仅预测那些被遮罩的令牌。 尽管它在文献中通常被称为完形任务（Taylor，1953），在这里我们将这个过程称为“遮罩LM”（MLM）。 在这种情况下，对应于遮罩令牌的最终隐藏向量被送到词汇表上的softmax输出，像在标准LM中一样。 在我们的所有实验中，我们随机地遮罩每个序列中所有WordPiece标记的15％。 与去噪自动编码器（Vincent et al。，2008）相反，我们只预测掩蔽的单词而不是重建整个输入。

Although this does allow us to obtain a bidirectional pre-trained model, there are two downsides to such an approach. The first is that we are creating a mismatch between pre-training and fine-tuning, since the [MASK] token is never seen during fine-tuning. To mitigate this, we do not always replace “masked” words with the actual [MASK] token. Instead, the training data generator chooses 15% of tokens at random, e.g., in the sentence my dog is hairy”。 it chooses hairy . It then performs the following procedure:

虽然这确实允许我们获得一个双向预训练模型，但是这种方法有两个缺点。首先，我们发现预训练和微调之间存在不匹配，因为[MASK]令牌从未在微调期间出现过。为了减轻这种情况，我们并不总是用实际的[MASK]标记替换被遮罩词语。相反，训练数据生成器随机选择15%的令牌，例如，在句子“我的狗毛茸茸的“。选择把“毛茸茸”遮掉。然后执行以下步骤:

* Rather than always replacing the chosen words with [MASK], the data generator will do the following: 数据生成器不总是用[MASK]替换所选择的词，而是执行以下操作
* 80% of the time: Replace the word with the [MASK] token, e.g., my dog is hairy → my dog is [MASK] 80%：用[MASK]替换遮罩单词
* 10% of the time: Replace the word with a random word, e.g., my dog is hairy → my dog is apple 用一个随机的单词替换这个单词。
* 10% of the time: Keep the word unchanged, e.g., my dog is hairy → my dog is hairy . The purpose of this is to bias the representation towards the actual observed word. 保持被遮罩单词不变。这样做的目的是使表示偏向于实际观察到的单词

The Transformer encoder does not know which words it will be asked to predict or which have been replaced by random words, so it is forced to keep a distributional contextual representation of every input token. Additionally, because random replacement only occurs for 1.5% of all tokens (i.e., 10% of 15%), this does not seem to harm the model’s language understanding capability. Transformer编码器不知道它将被要求预测哪个单词，或者哪些单词已经被随机单词替代，因此它必须保持每个输入标记的上下文表示。此外，由于所有令牌中只有1.5%发生随机替换(即15%中的10%)，因此这似乎不会损害模型的语言理解能力。

The second downside of using an MLM is that only 15% of tokens are predicted in each batch, which suggests that more pre-training steps may

be required for the model to converge. In Section 5.3 we demonstrate that MLM does converge marginally slower than a left-to-right model (which predicts every token), but the empirical improvements of the MLM model far outweigh the increased training cost. 使用MLM的第二个缺点是每批令牌的预测值只有15%，这意味着模型需要更多的预训练步骤才能收敛。在5.3节中，我们证明MLM的收敛速度略慢于从左到右的模型（预测每个标记），但MLM模型的实证改进远远超过增加的培训成本。（性价比高）

**3.3.2 Task #2: Next Sentence Prediction 任务2：下句预测**

Many important downstream tasks such as Question Answering (QA) and Natural Language Inference (NLI) are based on understanding the relationship between two text sentences, which is not directly captured by language modeling. In order to train a model that understands sentence relationships, we pre-train a binarized next sentence prediction task that can be trivially generated from any monolingual corpus. Specifically, when choosing the sentences A and B for each pre-training example, 50% of the time B is the actual next sentence that follows A, and 50% of the time it is a random sentence from the corpus. For example:

许多重要的下游任务，如问答(QA)和自然语言推理(NLI)，都是基于对两个文本句子之间关系的理解，而语言建模没有直接捕捉到这一点。为了训练一个能够理解句子关系的模型，我们预先训练了一个可以从任何单语料库中轻而易举地实现的双语下一步预测任务。具体来说，在为每个预训练的例子选择句子a和句子b时，50%的时间b是a之后的下一个句子，50%的时间b是语料库中的随机句子。例如

**Input =** [CLS] the man went to [MASK] store [SEP] he bought a gallon [MASK] milk [SEP] 输入[CLS]这个人去[MASK]商店[SEP] 他买了一加仑牛奶

**Label =** IsNext

**Input =** [CLS] the man [MASK] to the store [SEP] penguin [MASK] are flight ##less birds [SEP]

输入[CLS]人[MASK]到存储器[SEP] 企鹅[MASK]是飞行##少数鸟类[SEP]

**Label =** NotNext

We choose the Not Next sentences completely at random, and the final pre-trained model achieves 97%-98% accuracy at this task. Despite its simplicity, we demonstrate in Section 5.1 that pre-training towards this task is very beneficial to both QA and NLI. 我们完全随机选择NotNext句子，最终预训模型在该任务中达到97%-98%的准确率。尽管如此，我们在第5.1节展示了针对这项任务的前期培训对QA和NLI都是非常有益的。

### **3.4 Pre-training Procedure 预训练**程序

The pre-training procedure largely follows the existing literature on language model pre-training. For the pre-training corpus we use the concatenation of BooksCorpus (800M words) (Zhu et al., 2015) and English Wikipedia (2,500M words). For Wikipedia we extract only the text passages and ignore lists, tables, and headers. It is critical to use a document-level corpus rather than a shuffled sentence-level corpus such as the Billion Word Benchmark (Chelba et al., 2013) in order to extract long contiguous sequences. 预训练程序主要遵循以往包含预训练语言模型的文献。在预训练语料库中，我们使用了BooksCorpus(800M词)和英文维基百科(2500m词)。对于Wikipedia，我们只提取文本段落，而忽略列表、表格和标题。使用文档级语料库而不是混乱的句子级语料库(如BillionWordBenchmark(Chelba等人，2013年))来提取长的连续序列是非常重要的。

To generate each training input sequence, we sample two spans of text from the corpus, which we refer to as “sentences” even though they are typically much longer than single sentences (but can be shorter also). The first sentence receives the A embedding and the second receives the B embedding. 50% of the time B is the actual next sentence that follows A and 50% of the time it is a random sentence, which is done for the “next sentence prediction” task. They are sampled such that the combined length is ≤ 512 tokens. The LM masking is applied after WordPiece tokenization with a uniform masking rate of 15%, and no special consideration given to partial word pieces. 为了生成每个训练输入序列，我们从语料库中抽取两段文本，我们称之为"句子"，尽管它们通常比单个句子长得多(但也可以短一些)。第一句接受a嵌入，第二句接受b嵌入。50%的频率b是紧随a的下一个句子，50%的时间b是一个随机句子，这是为"下一个句子预测"任务而做的。对它们进行取样，使合并后的长度为512个标记。LM遮罩是在WordPiece生成之后使用的，统一的遮罩率为15%，对部分单词没有特殊考虑。

We train with batch size of 256 sequences (256 sequences \* 512 tokens = 128,000 tokens/batch) for 1,000,000 steps, which is approximately 40 epochs over the 3.3 billion word corpus. We use Adam with learning rate of 1e-4, β1 = 0.9, β2 = 0.999, L2 weight decay of 0.01, learning rate warm up over the first 10,000 steps, and linear decay of the learning rate. We use a dropout probability of 0.1 on all layers. We use a gelu activation (Hendrycks and Gimpel, 2016) rather than the standard relu, following OpenAI GPT. The training loss is the sum of the mean masked LM likelihood and mean next sentence prediction likelihood.

我们使用256个序列(256个序列\*512个令牌=128,000个令牌/一批)的批量训练1,000,000步，大约在33亿个单词语料库运行40个epochs。我们使用的Adam的学习速率为1e-4, β1 = 0.9, β2 = 0.999 , L2 weight decay是0.01，学习速率在最初的10,000步中开始升高，学习速率线性下降。我们在所有图层上使用0.1的dropout。我们使用gelu激活函数(henrycks和geimpl，2016)而不是标准relu，遵循OpenAIGPT。训练损失是平均遮罩LM似然值和平均下句预测似然值的总和。

Training of BERTBASE was performed on 4 Cloud TPUs in Pod configuration (16 TPU chips total). 5 Training of BERTLARGE was performed on 16 Cloud TPUs (64 TPU chips total). Each pre-training took 4 days to complete.

在4个云TPU上进行BERTBASE的训练(共16个TPU芯片)，在16个云TPU上进行BERTLARGE训练(共64个TPU芯片)。每次预训练需要4天才能完成。

### **3.5 Fine-tuning Procedure** 微调程序

For sequence-level classification tasks, BERT fine-tuning is straightforward. In order to obtain a fixed-dimensional pooled representation of the input sequence, we take the final hidden state (i.e., the output of the Transformer) for the first token in the input, which by construction corresponds to the the special [CLS] word embedding. We denote this vector as C ∈ R H . The only new parameters added during fine-tuning are for a classification layer W ∈ R K×H , where K is the number of classifier labels. The label probabilities P ∈ R K are computed with a standard softmax, P = softmax(CW T ). All of the parameters of BERT and W are fine-tuned jointly to maximize the log-probability of the correct label. For span-level and token-level prediction tasks, the above procedure must be modified slightly in a task-specific manner. Details are given in the corresponding subsection of Section 4.

对于序列级分类任务，BERT微调很简单。 为了获得输入序列的固定维度合并表示，我们对输入中的第一个标记采取最终隐藏状态（即Transformer的输出），其通过构造对应于特殊[CLS]嵌入。 我们将该向量表示为C∈RH. 在微调期间添加的唯一新参数是分类层W∈RK×H，其中K是分类器标签的数量。 计算标准概率P∈RK用标准softmax，P = softmax（CW T）。 BERT和W的所有参数都经过微调，以最大化正确标签的对数概率。 对于span级和令牌级预测任务，必须以特定于任务的方式稍微修改上述过程。 详情见第4节的相应小节。

For fine-tuning, most model hyperparameters are the same as in pre-training, with the exception of the batch size, learning rate, and number of training epochs. The dropout probability was always kept at 0.1. The optimal hyperparameter values are task-specific, but we found the following range of possible values to work well across all tasks: 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epoch除外。 dropout 始终保持在0.1。 最优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很好地适用于所有任务：对于微调，大多数模型超参数与预训练一样，批量大小，学习率和时期。

* Batch size: 16, 32
* Learning rate (Adam): 5e-5, 3e-5, 2e-5
* Number of epochs: 3, 4

We also observed that large data sets (e.g., 100k+ labeled training xamples) were far less sensitive to hyperparameter choice than small data sets. Fine-tuning is typically very fast, so it is reasonable to simply run an exhaustive search over the above parameters and choose the model that performs best on the development set. 我们还观察到，大型数据集(例如100k+标记的训练样本)对超参数选择的敏感性远远低于小型数据集。微调通常非常快速，因此只需对上述参数进行彻底搜索，并选择在开发集上表现最好的模型就可以了。

### 3.6 Comparison比较 of BERT and OpenAI GPT

The most comparable existing pre-training method to BERT is OpenAI GPT, which trains a left-to-right Transformer LM on a large text corpus. In fact, many of the design decisions in BERT were intentionally chosen to be as close to GPT as possible so that the two methods could be minimally compared. The core argument of this work is that the two novel pre-training tasks presented in Section 3.3 account for the majority of the empirical improvements, but we do note that there are several other differences between how BERT and GPT were trained: 现有的与BERT方法最有可比性的预训练方法是OpenAIGPT，它在一个大的文本语料库上训练一个从左到右的Transformer LM。事实上，BERT系统中的许多设计决策都被有意选择的尽可能接近GPT的结果，因此这两种方法可以进行最低限度的比较。这项工作的核心论点是，证券交易委员会第3.3号决议中提出的两个新颖的前期培训任务占了经验分析改进的大部分，但是我们注意到BERT和GPT的培训方式还有几个不同之处:

* GPT is trained on the BooksCorpus (800M words); BERT is trained on the BooksCorpus (800M words) and Wikipedia (2,500M words). Gpt用于训练bookscopus(800M字)，BERT用于训练bookscoor-pus(800M字)和Wikipedia(2500m字)。
* GPT uses a sentence separator ([SEP]) and classifier token ([CLS]) which are only introduced at fine-tuning time; BERT learns [SEP], [CLS] and sentence A/B embeddings during pre-training. Gpt使用一个句子分隔符([SEP])和分类器标记([CLS])，这两个标记只有在微调时才会引用;BERT在预训练时就能学到[SEP]、[CLS]和句子a/b的嵌入。
* GPT was trained for 1M steps with a batch size of 32,000 words; BERT was trained for 1M steps with a batch size of 128,000 words. Gpt用于训练1M步骤，批量大小为32,000字;BERT用于训练1M步骤，批量大小为128,000字
* GPT used the same learning rate of 5e-5 for all fine-tuning experiments; BERT chooses a task-specific fine-tuning learning rate which performs the best on the development set. Gpt在所有微调实验中使用5e-5相同的学习速率;BERT采用在特定任务的开发集上表现最佳的微调学习速率，

To isolate the effect of these differences, we perform ablation experiments in Section 5.1 which demonstrate that the majority of the improvements are in fact coming from the new pre-training tasks. 为了隔离这些差异的影响，我们在5.1节的每个形式的消融实验表明，大多数的改进实际上来自于新的预训练任务。

## **4 Experiments** 实验

In this section, we present BERT fine-tuning results on 11 NLP tasks.在本节中，我们介绍了BERT算法在11个NLP任务上的微调结果。

### 4.1 GLUE Datasets

The General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018) is a collection of diverse natural language understanding tasks. Most of the GLUE datasets have already existed for a number of years, but the purpose of GLUE is to (1) distribute these datasets with canonical Train, Dev, and Test splits, and (2) set up an evaluation server to mitigate issues with evaluation inconsistencies and Test set over-fitting. GLUE does not distribute labels for the Test set and users must upload their predictions to the GLUE server for evaluation, with limits on the number of submissions.

通用语言理解评估（GLUE）基准（Wang et al。，2018）是各种自然语言理解任务的集合。 大多数GLUE数据集已存在多年，但GLUE的目的是（1）使用规范的Train，Dev和Test拆分分发这些数据集，以及（2）设置评估服务以减轻问题评估不一致和测试集过度拟合。 GLUE不会为测试集分发标签，用户必须将其预测上传到GLUE服务器进行评估，并限制提交的数量。

The GLUE benchmark includes the following datasets, the descriptions of which were originally summarized in Wang et al. (2018): Glue基准包括以下数据集，其描述最初在Wangetal.(2018)中总结:

**MNLI** Multi-Genre Natural Language Inference is a large-scale, crowdsourced entailment classification task (Williams et al., 2018). Given a pair of sentences, the goal is to predict whether the second sentence is an entailment, contradiction, or neutral with respect to the first one. 多类型自然语言推理是一个大规模的、众包式的分类任务。给定一对句子，目标是预测第二句子相对于第一句是蕴涵式、矛盾式还是中性式。

**QQP** Quora Question Pairs is a binary classification task where the goal is to determine if two questions asked on Quora are semantically equivalent (Chen et al., 2018). Quora问答对是一个二分类任务，目标是确定Quora上提出的两个问题是否属于同义(Chen等人，2018)。

**QNLI** Question Natural Language Inference is a version of the Stanford Question Answering Dataset (Rajpurkar et al., 2016) which has been converted to a binary classification task (Wang et al., 2018). The positive examples are (question, sentence) pairs which do contain the correct answer, and the negative examples are (question, sentence) from the same paragraph which do not contain the answer. 自然语言推理是斯坦福问题回答数据集(Rajpurkaretal.，2016)的一个版本，该数据集已经被转换为二分类任务(Wangetal.，2018)。肯定式例句是包含正确答案的成对句，否定式例句是同一段落中不包含正确答案的成对句。

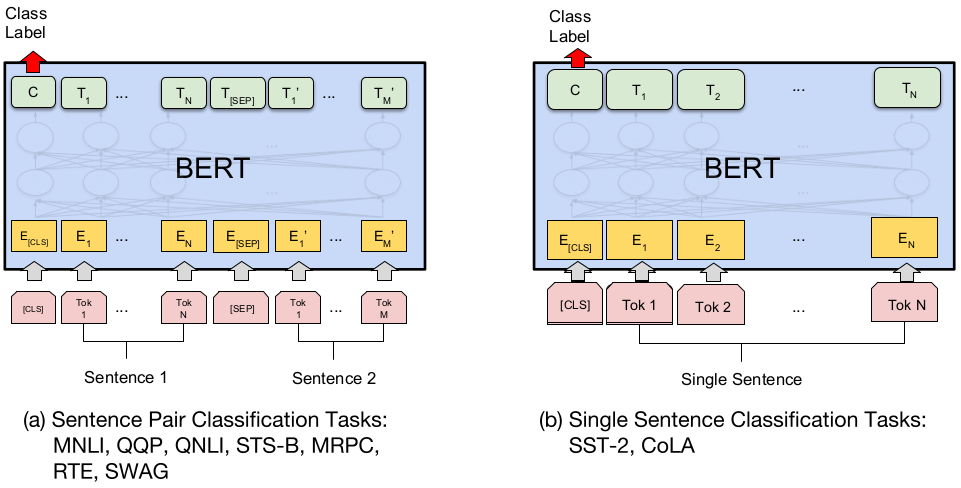
**SST-2** The Stanford Sentiment Treebank is a binary single-sentence classification task consisting of sentences extracted from movie reviews

with human annotations of their sentiment (Socher et al., 2013). Sst-2"斯坦福情感树库"是一个单句二分类任务，包括从电影评论中提取的句子以及对其情感的人工注释(Socher等人，2013)。

**CoLA** The Corpus of Linguistic Acceptability is a binary single-sentence classification task, where the goal is to predict whether an English sentence is linguistically “acceptable” or not (Warstadt et al., 2018).语言可接受性语料库是一个单句二分类任务，其中我们的目标是预测一个英语句子在语言学上是否"可以接受"

**STS-B** The Semantic Textual Similarity Benchmark is a collection of sentence pairs drawn from news headlines and other sources (Cer et al., 2017). They were annotated with a score from 1 to 5 denoting how similar the two sentences are in terms of semantic meaning. Sts-b语义文本相似性标记是从新闻标题和其他来源获得的句子对的集合(Cer等，2017)。这些句子的注释分数从1分到5分不等，表明这两个句子在语义方面有多么相似。

**MRPC** Microsoft Research Paraphrase Corpus consists of sentence pairs automatically extracted from online news sources, with human annotations for whether the sentences in the pair are semantically equivalent (Dolan and Brockett, 2005). Mrpc微软研究院解释语料库包括自动从在线新闻源中提取的句子对，并附有人工注释，以判断句子是否为语义对等的句子(Dolan和Brockett，2005)。



**RTE** Recognizing Textual Entailment is a binary entailment task similar to MNLI, but with much less training data (Bentivogli et al., 2009). 6 RTE识别文字蕴涵是一项双重分配的任务，类似于MNLI，但训练数据少得多

**WNLI** Winograd NLI is a small natural language inference dataset deriving from (Levesque et al., 2011). The GLUE web page notes that there are issues with the construction of this dataset, 7 and every trained system that’s been submitted

to GLUE has has performed worse than the 65.1 baseline accuracy of predicting the majority class. We therefore exclude this set out of fairness to OpenAI GPT. For our GLUE submission, we always predicted the majority class. 是一个小型的自然语言推理数据集(莱韦斯克等人，2011年)。注意到在构建这个数据集时存在一些问题，并且每一个提交到GLUE的训练系统都表现得比预测多数类别的65.1基线准确性差。因此，我们排除了这一套出于公平的OpenAIGPT。为了我们的GLUE服务，我们总是预测大多数类别。

6 Note that we only report single-task fine-tuning results in this paper. Multitask fine-tuning approach could potentially push the results even further. For example, we did observe substantial improvements on RTE from multi-task training with MNLI.

7 <https://gluebenchmark.com/faq> 6注意，在本文中我们只报告单任务微调结果。多任务微调方法可能会进一步推动结果。例如，我们从MNLI的多任务培训中观察到RTE的实质性改进。

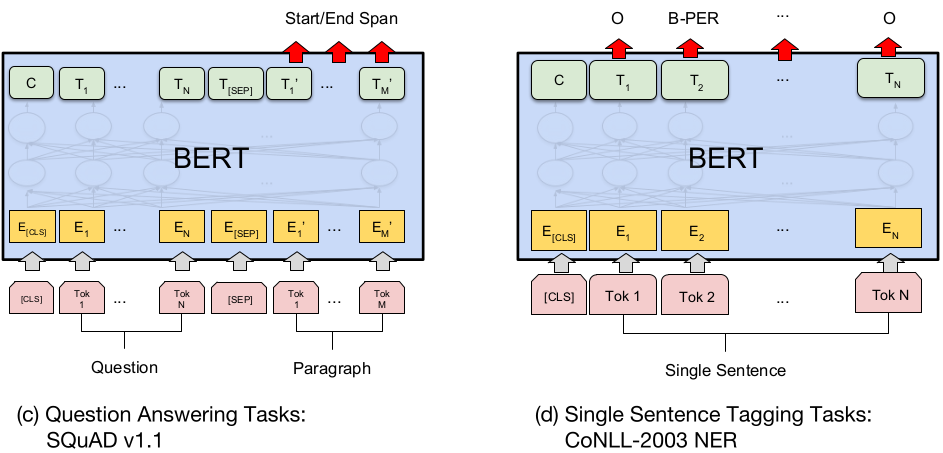


Figure 3: Our task specific models are formed by incorporating BERT with one additional output layer, so a minimal number of parameters need to be learned from scratch. Among the tasks, (a) and (b) are sequence-level tasks while (c) and (d) are token-level tasks. In the figure, E represents the input embedding, T i represents the contextual representation of token i, [CLS] is the special symbol for classification output, and [SEP] is the special symbol to separate non-consecutive token sequences. 图3:我们的任务特定的模型是通过将BERT与一个额外的输出层结合而形成的，所以需要从头开始学习的参数数量最少。在这些任务中，(a)和(b)是序列级任务，(c)和(d)是令牌级任务。在图中，e表示输入嵌入，Ti表示令牌i的上下文表示，[CLS]是分类输出的特殊符号，[SEP]是分隔非连续令牌序列的特殊符号。

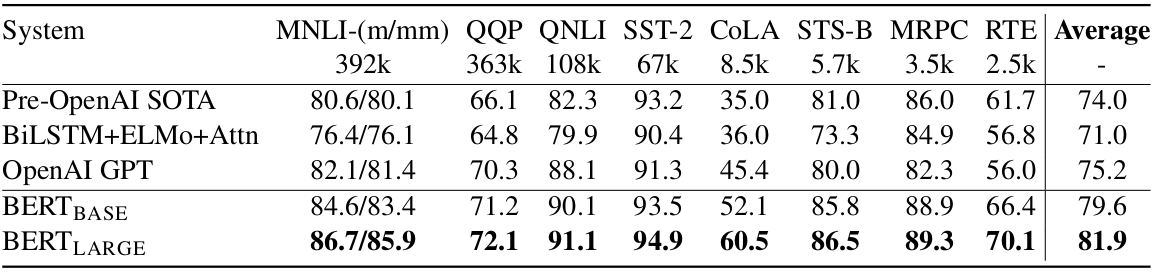


Table 1: GLUE Test results, scored by the GLUE evaluation server. The number below each task denotes the number of training examples. The “Average” column is slightly different than the official GLUE score, since we exclude the problematic WNLI set. OpenAI GPT = (L=12, H=768, A=12); BERT BASE = (L=12, H=768, A=12); BERT LARGE = (L=24, H=1024, A=16). BERT and OpenAI GPT are single-model, single task. All results obtained from

<https://gluebenchmark.com/leaderboard> and <https://blog.openai.com/language-unsupervised/>.

表1:GLUE测试结果，GLUE评估服务器得分。每个任务下面的数字表示训练样本的数量。"Average"列与官方的GLUE分数略有不同，因为我们排除了有问题的WNLI集合。OpenaiGPT(l12，h768，a12);BERTBASE(l12，h768，a12);BERTLARGE(l24，h1024，a16).Bert系统和OpenAIGPT系统都是单模块、单任务。所有结果来自上面链接。

### 4.1.1 GLUE Results

To fine-tune on GLUE, we represent the input sequence or sequence pair as described in Section 3, and use the final hidden vector C ∈ RH corresponding to the first input token ([CLS]) as the aggregate representation. This is demonstrated visually in Figure 3 (a) and (b). The only new parameters introduced during fine-tuning is a classification layer W ∈ R K×H , where K is the number of labels. We compute a standard classification loss with C and W , i.e., log(softmax(CW T )). 为了微调GLUE，我们表示第3节中描述的输入序列或序列对，并使用最终的隐藏向量C ∈ RH对第一个输入标记([CLS])作为聚合表示。这在图3(a)和(b)中首次演示。在微调过程中引入的唯一新的光栅是一个分类层w2rkh，其中k是标签的数目。我们用c和w计算标准的分类损失，即对数(softmax(CWt))

为了微调GLUE，我们如第3节所述表示输入序列或序列对，并使用对应于第一个输入标记（[CLS]）的最终隐藏向量C∈RH作为聚合表示。 这在图3（a）和（b）中视觉化展示。 在微调期间引入的唯一新参数是分类层W∈RK×H，其中K是标签的数量。 我们用C和W计算标准分类损失，即log（softmax（CWT））。

We use a batch size of 32 and 3 epochs over the data for all GLUE tasks. For each task, we ran fine-tunings with learning rates of 5e-5, 4e-5, 3e-5, and 2e-5 and selected the one that performed best on the Dev set. Additionally, for BERT LARGE we found that fine-tuning was sometimes unstable on small data sets (i.e., some runs would produce degenerate results), so we ran several random restarts and selected the model that performed best on the Dev set. With random restarts, we use the same pre-trained checkpoint but perform different fine-tuning data shuffling and classifier layer initialization. We note that the GLUE data set distribution does not include the Test labels, and we only made a single GLUE evaluation server submission for each BERTBASE and BERTLARGE .

我们对所有GLUE任务的数据使用32和3epoch的批处理大小。对于每个任务，我们进行了精细调整，学习率分别为5e-5、4e-5、3e-5和2e-5，并选择了在Dev集中表现最好的一个。此外，对于BERTLARGE我们发现微调有时是不稳定的小的数据集(例如，一些运行会产生去生成的结果)，所以我们运行了几个随机重启并选择了在Dev集上表现最好的模型。对于随机重启，我们使用相同的预先训练的检查点，但是执行不同的微调数据洗牌和初始化分类器层。我们注意到GLUE数据集发行版不包含Test标签，并且我们只为每个BERTBASE和BERTLARGE制作了一个GLUE评估服务器提交。

Results are presented in Table 1. Both BERTBASE and BERTLARGE outperform all existing systems on all tasks by a substantial margin, obtaining 4.4% and 6.7% respective average accuracy improvement over the state-of-the-art. Note that BERTBASE and OpenAI GPT are nearly identical in terms of model architecture outside of the attention masking. For the largest and most widely reported GLUE task, MNLI, BERT obtains a 4.7% absolute accuracy improvement over the state-of-the-art. On the official GLUE leader-board, 8 BERT LARGE obtains a score of 80.4, compared to the top leader board system, OpenAI GPT, which obtains 72.8 as of the date of writing. 结果见表1。Bertbase和BERTLARGE在所有任务中都比现有系统表现出色，分别获得了4.4%和6.7%的平均性能提升。请注意，就注意屏蔽之外的模型架构而言，BERTBASE和OpenAIGPT几乎是同一级别的。对于最大的和最广泛报道的GLUE任务，MNLI，BERT获得了4.7%的绝对精度提高超过最先进的水平。在GLUE官方的排行榜上，BERTLARGE获得了80.4分，OpenAIGPT在排行榜上名列前茅，截至写作之日，该排行榜获得72.8分。

It is interesting to observe that BERT LARGE significantly outperforms BERT BASE across all tasks, even those with very little training data. The effect of BERT model size is explored more thoroughly in Section 5.2. 值得注意的是，BERTLARGE筛选器在所有任务中都明显优于BERTBASE，即使是那些训练数据很少的任务。在第5.2节中对BERT模型的影响进行了更深入的探讨。

### 4.2 SQuAD v1.1

The Standford Question Answering Dataset (SQuAD) is a collection of 100k crowdsourced question/answer pairs (Rajpurkar et al., 2016). Given a question and a paragraph from Wikipedia containing the answer, the task is to predict the answer text span in the paragraph. For example: 斯坦福问题回答数据集(SQuAD)是一个100k众包问题/答案对的集合(Rajpurkaretal.，2016)。给出维基百科上的一个问题和一个段落包含答案的任务是在段落中预测下行文本跨度。例如:

* Input Question:

Where do water droplets collide with ice crystals to form precipitation? 水滴与冰晶碰撞在何处形成降水？

* Input Paragraph:

... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals within a cloud....降水形式是较小的水滴通过与其他雨滴或云中的冰晶碰撞而聚合。...

* Output Answer:

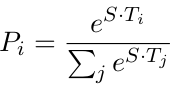
within a cloud 在一片云里

This type of span prediction task is quite different from the sequence classification tasks of GLUE, but we are able to adapt BERT to run on SQuAD in a straightforward manner. Just as with GLUE, we represent the input question and paragraph as a single packed sequence, with the question using the A embedding and the paragraph using the B embedding. The only new parameters learned during fine-tuning are a start vector S ∈ R H and an end vector E ∈ R H . Let the final hidden vector from BERT for the i th input token be denoted as T i ∈ R H . See Figure 3 (c) for a visualization. Then, the probability of word i being the start of the answer span is computed as a dot

product between T i and S followed by a softmax over all of the words in the paragraph:

这种类型的跨度预测任务与GLUE的序列分类任务完全不同，但我们能够以简单的方式调整BERT以在SQuAD上运行。 与GLUE一样，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。 在微调期间学习的唯一新参数是起始矢量S∈RH和结束矢量E∈RH. 令来自BERT的第i个输入令牌的最终隐藏向量表示为Ti∈RH. 有关可视化，请参见图3（c）。 然后，将单词i作为答案跨度的开始的概率计算为点

T i和S之间的乘积，然后是段落中所有单词的softmax：

、

The same formula is used for the end of the answer span, and the maximum scoring span is used as the prediction. The training objective is the log-likelihood of the correct start and end positions. 下部跨度末端采用相同的计算公式，最大计算跨度作为预测值。训练目标是正确的开始和结束位置的对数似然。

We train for 3 epochs with a learning rate of 5e5 and a batch size of 32. At inference time, since the end prediction is not conditioned on the start, we add the constraint that the end must come after the start, but no other heuristics are used. The tokenized labeled span is aligned back to the original untokenized input for evaluation. 我们训练了3个时期，学习速度为5e-5，批量为32。在推理时，由于结束预测不受开始条件的限制，我们添加了结束必须在开始之后出现的约束，但是没有使用其他的试探法。将tok-enized标记span对齐到原来未标记的输入，以进行计算。

Results are presented in Table 2. SQuAD uses a highly rigorous testing procedure where the submitter must manually contact the SQuAD organizers to run their system on a hidden test set, so we only submitted our best system for testing. The result shown in the table is our first and only Test submission to SQuAD. We note that the top results from the SQuAD leaderboard do not have up-to-date public system descriptions available, and are allowed to use any public data when training their systems. We therefore use very modest data augmentation in our submitted system by jointly training on SQuAD and TriviaQA (Joshi et al., 2017). 结果见表2。Squad使用高度严格的测试程序，其中提交者必须手动联系SQuAD组织者，以便在一个隐藏的测试集上运行他们的系统，因此我们只提交了最好的系统进行测试。表中显示的结果是我们第一个也是唯一一个提交给SQuAD的测试。我们注意到排在前面的结果从队排行榜没有最新的公共系统说明可用，并允许使用任何公共数据时，他们的系统训练。因此，我们在我们提交的系统中使用非常适度的数据实现，通过联合对SQuAD和TriviaQA(Joshi等人，2017)进行训练。

Our best performing system outperforms the top leaderboard system by +1.5 F1 in ensembling and +1.3 F1 as a single system. In fact, our single BERT model outperforms the top ensemble system in terms of F1 score. If we fine-tune on only SQuAD (without TriviaQA) we lose 0.1-0.4 F1 and still outperform all existing systems by a wide margin. 我们最好的演奏系统表现优于排行榜系统由+1.5F1在自由组合和+1.3F1作为一个单一的系统。事实上，我们的单一BERT模型在F1得分方面优于顶部集成系统。如果我们只调整班(没有TriviaQA)我们失去了0.1-0.4F1和仍然超过所有现有的系统广泛的差距。

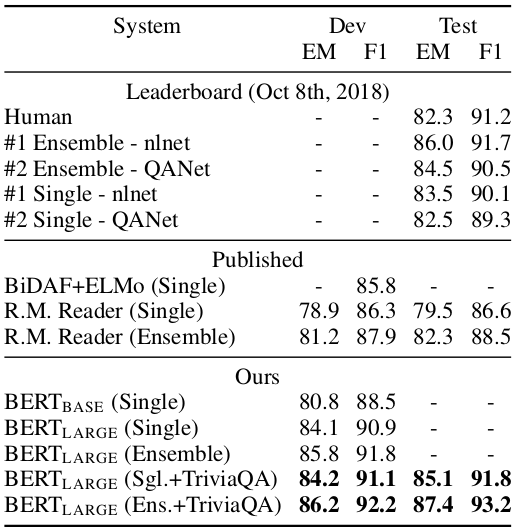


Table 2: SQuAD results. The BERT ensemble is 7x systems which use different pre-training checkpoints and fine-tuning seeds. 表2:小组成绩。Bert系统是使用不同的预训练检查点和微调种子的7x系统。

### 4.3 Named Entity Recognition

To evaluate performance on a token tagging task, we fine-tune BERT on the CoNLL 2003 Named Entity Recognition (NER) dataset. This dataset consists of 200k training words which have been annotated as Person , Organization , Location , Miscellaneous , or Other (non-named entity). 为了评估令牌标记任务的性能，我们在CoNLL2003命名实体识别(NER)数据集上微调BERT。这个数据集由20万个训练词组成，这些训练词被注释为Person、Organization、Location、Miscellaneous或Other(非命名实体)。

For fine-tuning, we feed the final hidden representation T i ∈ R H for to each token i into a classification layer over the NER label set. The predictions are not conditioned on the surrounding predictions (i.e., non-autoregressive and no CRF). To make this compatible with WordPiece tokenization, we feed each CoNLL-tokenized input word into our WordPiece tokenizer and use the hidden state corresponding to the first sub-token as input to the classifier. For example:为了微调，我们为每个令牌i提供最终的隐藏表示ti2rh到NER标签集上的分类层。预测的条件不是环绕预测(即，非自回归和没有CRF)。为了使它与WordPiece标记兼容，我们将每个经过conlltokenized化的输入字提供给WordPiece标记器，并使用与第一个标记器对应的隐藏状态子令牌作为分类器的输入。例如:

吉姆·汉森是一个木偶师



Where no prediction is made for X. Since the WordPiece tokenization boundaries are a known part of the input, this is done for both training and test. A visual representation is also given in Figure 3 (d). A cased WordPiece model is used for NER, whereas an uncased model is used for all other tasks. 在没有对x进行预测的地方。由于WordPiece标记的边界是已知的输入的一部分，因此既可用于培训，也可用于测试。图3(d)中也给出了一个可视化的表示。大小写WordPiece模型用于NER，而非大小写模型用于所有其他任务。

Results are presented in Table 3. BERT LARGE outperforms the existing SOTA, Cross-View Training with multi-task learning (Clark et al., 2018), by +0.2 on CoNLL-2003 NER Test. 结果见表3。Bertlarge超过了现有的SOTA，跨视图训练与多任务学习(克拉克等人，2018年)，在CoNLL-2003年NER测试+0.2。

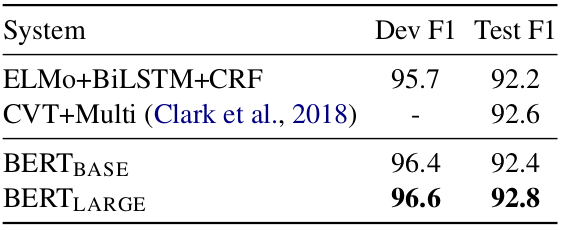


Table 3: CoNLL-2003 Named Entity Recognition results. The hyperparameters were selected using the Dev set, and the reported Dev and Test scores are averaged over 5 random restarts using those hyperparameters. 表3:CoNLL-2003年命名实体识别结果。使用Dev集合选择超参数，并且报告的Dev和Test分数通过使用这些hyperparame-ters获得了5个以上的随机重新开始。

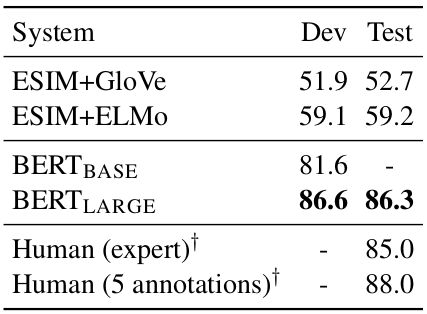


Table 4: SWAG Dev and Test accuracies. Test results were scored against the hidden labels by the SWAG authors. † Human performance is measure with 100 samples, as reported in the SWAG paper. 表4:SWAGDev和Test的精确度。测试结果由SWAGau-thors根据隐藏标签得分。正如SWAG文件中报告的那样，yHuman性能是用100个单位来度量的。

### 4.4 SWAG

The Situations With Adversarial Generations (SWAG) dataset contains 113k sentence-pair completion examples that evaluate grounded commonsense inference (Zellers et al., 2018). 使用敌对世代的情境(SWAG)数据集包含评价扎根常识推理的113k句对充实例子(Zellers等人，2018)。

Given a sentence from a video captioning dataset, the task is to decide among four choices the most plausible continuation. For example: 给出一个来自视频字幕数据集的句子，任务是在四个选项中选择最合理的继续。例如:

A girl is going across a set of monkey bars. She 一个女孩正在穿过一组单杠。她

(i) jumps up across the monkey bars. 跳过单杠

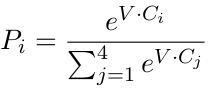
(ii) struggles onto the bars to grab her head. 挣扎在栏杆上，想抓住她的头

(iii) gets to the end and stands on a wooden plank. 走到尽头，站在木板上

(iv) jumps up and does a back flip. 跳起来做后空翻

Adapting BERT to the SWAG dataset is similar to the adaptation for GLUE. For each example, we construct four input sequences, which each contain the concatenation of the the given sentence (sentence A) and a possible continuation (sentence B). The only task-specific parameters we introduce

is a vector V ∈ RH , whose dot product with the final aggregate representation C i ∈ R H denotes a score for each choice i. The probability distribution is the softmax over the four choices: 适应BERT到SWAG数据集类似于适应GLUE。对于每个例子，我们构造四个输入序列，每个序列都包含给定句子(句a)和可能连续句(句b)的连接。我们引入的特定任务的唯一参数是向量v2RH，其点乘与最终的集合表示Ci2rh表示a每个选项的得分。概率分布是四个选项的最大值:



We fine-tune the model for 3 epochs with a learning rate of 2e-5 and a batch size of 16. Results are presented in Table 4. BERT LARGE outperforms the authors’ baseline ESIM+ELMo system by +27.1%. 我们对模型进行了3个时期的微调，学习速率为2e-5，批量大小为16。结果见表4。Bertlarge超出了作者的基线esim+elmosystem+27.1%。

## **5 Ablation Studies 消融实验（模型简化，控制变量，**验证结构有效性)

Although we have demonstrated extremely strong empirical results, the results presented so far have not isolated the specific contributions from each aspect of the BERT framework. In this section, we perform ablation experiments over a number of facets of BERT in order to better understand their relative importance. 虽然我们已经展示了非常强大的实证结果，但是迄今为止所展示的结果并没有从BERT框架的每个方面孤立出具体的贡献。在本节中，我们将在BERT的许多方面进行消融实验，以便更好地理解它们的相对重要性。

### **5.1 Effect of Pre-training Tasks** 预训练任务的影响

One of our core claims is that the deep bidirectionality of BERT, which is enabled by masked LM pre-training, is the single most important improvement of BERT compared to previous work. To give evidence for this claim, we evaluate two new models which use the exact same pre-training data, fine-tuning scheme and Transformer hyperparameters as BERT BASE : 我们的核心主张之一是BERT的深层的双向连接，这是遮罩LM预训练所能实现的，与以前的工作相比是BERT最重要的一项改进。为了证明这一说法，我们评估了两个新模型，它们使用了与BERTBASE完全相同的预训练数据、微调方案和Transformer超参数:

**1. No NSP**: A model which is trained using the “masked LM” (MLM) but without the “next sentence prediction” (NSP) task. 一个模型，训练使用"遮罩LM"(MLM)，但没有"下句预测"(NSP)的任务

**2. LTR & No NSP**: A model which is trained using a Left-to-Right (LTR) LM, rather than an MLM. In this case, we predict every in-put word and do not apply any masking. The left-only constraint was also applied at fine-tuning, because we found it is always worse to pre-train with left-only-context and fine-tune with bidirectional context. Additionally, this model was pre-trained without the NSP task. This is directly comparable to OpenAI GPT, but using our larger training dataset, our input representation, and our fine-tuning scheme. 使用从左到右的语言模型(LTR LM)进行训练的模型，而不是MLM。在这种情况下，我们预测每个输入单词并且不使用任何遮罩。左向约束也用于微调，因为我们发现把左向约束预训练的模型用于双向上下文进行微调效果总是更糟糕。此外，该模型在没有NSP任务的情况下进行了预训练。这与OpenAIGPT直接相似，但是使用了更大的训练数据集、输入表示和精细调优方案。

Results are presented in Table 5. We first examine the impact brought by the NSP task. We can see that removing NSP hurts performance significantly on QNLI, MNLI, and SQuAD. These results demonstrate that our pre-training method is critical in obtaining the strong empirical results presented previously.

结果见表5。我们首先检查NSP任务带来的影响。 我们可以看到，删除NSP会严重损害QNLI，MNLI和SQuAD的性能。 这些结果表明，我们的预训练方法对于获得先前提出的强有力的实证结果至关重要。

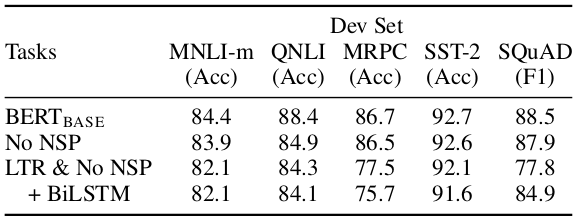


Table 5: Ablation over the pre-training tasks using the BERT BASE architecture. “No NSP” is trained without the next sentence prediction task. “LTR & No NSP” is trained as a left-to-right LM without the next sentence prediction, like OpenAI GPT. “+ BiLSTM” adds a randomly initialized BiLSTM on top of the “LTR + No NSP” model during fine-tuning. 表5:使用BERTBASE架构完成预训练的消融实验。"没有NSP"是训练没有下句预测任务。"LTR&NoNSP"被训练成一个没有下句预测的左到右LM，就像OpenAIGPT一样。"+BiLSTM"在微调期间在"LTR+NoNSP"模型的顶部添加了一个随机初始化的BiLSTM。

Next, we evaluate the impact of training bidirectional representations by comparing “No NSP” to “LTR & No NSP”. The LTR model performs worse than the MLM model on all tasks, with extremely large drops on MRPC and SQuAD. For SQuAD it is intuitively clear that an LTR model will perform very poorly at span and token prediction, since the token-level hidden states have no right-side context. For MRPC is unclear whether the poor performance is due to the small data size or the nature of the task, but we found this poor performance to be consistent across a full hyper-parameter sweep with many random restarts.

接下来，我们通过比较"NoNSP"和"LTR&NoNSP"来评价双向表征训练的效果。LTR 模型在所有任务上的表现都比MLM模型差，MRPC和SQuAD上的表现非常差。对于SQuAD来说，LTR模型在span和token预测上的表现非常差，因为令牌级的隐藏状态没有右侧上下文。对于MRPC，尚不清楚性能不佳是由于数据量小还是任务的性质，但我们发现这种不良性能在完整的超参数扫描中具有很多随机重启的一致性。

In order make a good faith attempt at strengthening the LTR system, we tried adding a randomly initialized BiLSTM on top of it for fine-tuning. This does significantly improve results on SQuAD, but the results are still far worse than the pre-trained bidirectional models. It also hurts performance on all four GLUE tasks.

为了加强LTR系统，我们尝试在它上面添加一个随即初始化的BiLSTM来进行微调。这确实显著改善了SQuAD的结果，但是结果仍然远远不如预训练的双向模型。它还会影响所有四个GLUE任务的性能。

We recognize that it would also be possible to train separate LTR and RTL models and represent each token as the concatenation of the two models, as ELMo does. However: (a) this is twice as expensive as a single bidirectional model; (b) this is non-intuitive for tasks like QA, since the RTL model would not be able to condition the answer on the question; (c) this it is strictly less powerful than a deep bidirectional model, since a deep bidirectional model could choose to use either left or right context.

我们认识到，还可以训练分开的LTR和RTL模型，并将每个标记表示为两个模型的连接，就像ELMo那样。然而:(a)这是单一双向模型开销的两倍;(b)对于QA这样的任务来说，这是非直观的，因为RTL模型不能将答案置于问题之上;(c)严格来说，它不如双向深度模型强大，因为一个深度的双向的模型可以选择使用左上下文或右上下文。

### **5.2 Effect of Model Size** 模型尺寸的影响

In this section, we explore the effect of model size on fine-tuning task accuracy. We trained a number of BERT models with a differing number of layers, hidden units, and attention heads, while otherwise using the same hyperparameters and training procedure as described previously.

在这一部分中，我们将探讨模型大小对于微调任务准确率的影响。我们训练了一些BERT模型，这些模型具有不同数量的图层、隐藏单元和注意头(head)，同时使用了前面描述的相同的超参数和训练方法。

Results on selected GLUE tasks are shown in Table 6. In this table, we report the average Dev Set accuracy from 5 random restarts of fine-tuning. We can see that larger models lead to a strict accuracy improvement across all four datasets, even for MRPC which only has 3,600 labeled training examples, and is substantially different from the pre-training tasks. It is also perhaps surprising that we are able to achieve such significant improvements on top of models which are already quite large relative to the existing literature.For example, the largest Transformer explored in Vaswani et al. (2017) is (L=6, H=1024, A=16) with 100M parameters for the encoder, and the largest Transformer we have found in the literature is (L=64, H=512, A=2) with 235M parameters (Al-Rfou et al., 2018). By contrast, BERTBASE contains 110M parameters and BERT LARGE contains 340M parameters.

选定的GLUE任务的结果如表6所示。在此表中，我们报告了5次随机重启微调的平均开发集准确率。我们可以看到，较大的模型导致所有四个数据集准确率的绝对提高，即使对于仅有3,600个标记训练样例的MRPC，并且与预训练任务有很大不同。 同样令人惊讶的是，我们能够在相对于现有文献已经相当大的模型之上实现这种显著的改进。例如，Vaswani等人研究的largest Transformer explored（2017）是（L = 6，H = 1024，A = 16），编码器有100M参数，我们在文献中找到的最大Transformer是（L = 64，H = 512，A = 2），235M 参数（Al-Rfou等，2018）。 相比之下，BERTBASE包含110M参数，BERTLARGE包含340M参数。

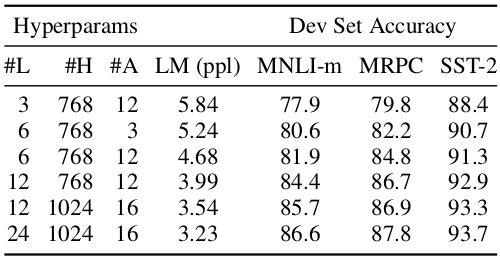
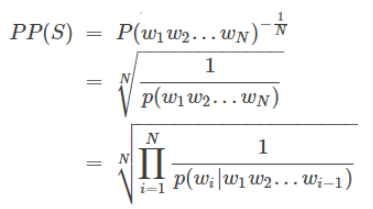


Table 6: Ablation over BERT model size. #L = the number of layers; #H = hidden size; #A = number of attention heads. “LM (ppl)” is the masked LM perplexity of held-out training data. 表6：BERT模型尺寸的消融实验。 #L =层数; #H =隐藏的大小; #A =关注头数。 “LM（ppl）”是保持训练数据的 遮罩LM困惑度。

Dev Set Accuracy 验证集准确率。

注释：PPL是用在自然语言处理领域（NLP）中，衡量语言模型好坏的指标。它主要是根据每个词来估计一句话出现的概率，并用句子长度作normalize，公式为



S代表sentence，N是句子长度，p(wi)是第i个词的概率。第一个词就是 p(w1|w0)，而w0是START，表示句子的起始，是个占位符。还有人说，Perplexity可以认为是average branch factor（平均分支系数），即预测下一个词时可以有多少种选择。别人在作报告时说模型的PPL下降到90，可以直观地理解为，在模型生成一句话时下一个词有90个合理选择，可选词数越少，我们大致认为模型越准确。这样也能解释，为什么PPL越小，模型越好。

It has been known for many years that increasing the model size will lead to continual improvements on large-scale tasks such as machine translation and language modeling, which is demonstrated by the LM perplexity of held-out training data shown in Table 6. However, we believe that this is the first work to demonstrate that scaling to extreme model sizes also leads to large improvements on very small scale tasks, provided that the model has been sufficiently pre-trained.

多年来人们已经知道，增加模型大小将导致机器转换和语言建模等大规模任务的不断改进，表6中所示训练数据的LM的Perplexity展示了这一规律。然而，我们认为，这是第一次证明向极端模型尺寸伸缩也会导致对非常小规模任务的大规模改进，前提是该模型已经得到充分的预先训练。

### **5.3 Effect of Number of Training Steps** 训练步数的影响

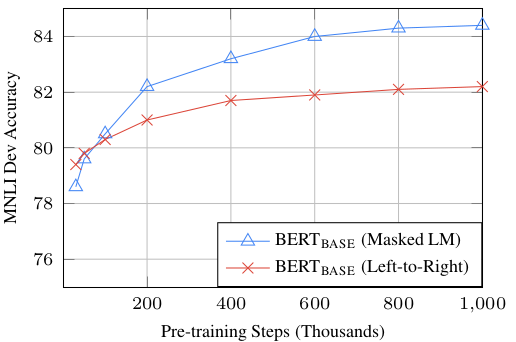


Figure 4: Ablation over number of training steps. This shows the MNLI accuracy after fine-tuning, starting from model parameters that have been pre-trained for k steps. The x-axis is the value of k. 图4:训练步骤的消融次数。这表明MNLI精调后的准确性，从模型参数，已经预先训练的k步骤。X轴是k的值。

Figure 4 presents MNLI Dev accuracy after fine-tuning from a checkpoint that has been pre-trained for k steps. This allows us to answer the following questions: 图4显示了MNLI开发精度从检查点微调后，已经为k步骤预先训练的。这使我们能够回答下列问题:

1. Question: Does BERT really need such a large amount of pre-training (128,000 words/batch \* 1,000,000 steps) to achieve high fine-tuning accuracy? 提问:BERT真的需要如此大量的前期训练(128,000字/批\*100,000步)来达到高微调精度吗？

Answer: Yes, BERT BASE achieves almost 1.0% additional accuracy on MNLI when trained on 1M steps compared to 500k steps. 答案:是的，与500k步骤相比，BERTBASE在MNLI上达到了几乎1.0%的额外准确率。

1. Question: Does MLM pre-training converge slower than LTR pre-training, since only 15% of words are predicted in each batch rather than every word? 问题:传销预训练聚合慢于LTR预训练，因为只有15%的字预测在每批比每个字

Answer: The MLM model does converge slightly slower than the LTR model. However, in terms of absolute accuracy the MLM model begins to outperform the LTR model almost immediately. 答:传销模式确实收敛比LTR模式稍慢。无论如何，在绝对精度方面的传销模型开始超过几乎立即LTR模型

### 5.4 Feature-based Approach with BERT

All of the BERT results presented so far have used the fine-tuning approach, where a simple classification layer is added to the pre-trained model, and all parameters are jointly fine-tuned on a down-stream task. However, the feature-based approach, where fixed features are extracted from the pre-trained model, has certain advantages. First, not all NLP tasks can be easily be represented by a Transformer encoder architecture, and therefore require a task-specific model architecture to be added. Second, there are major computational benefits to being able to pre-compute an expensive representation of the training data once and then run many experiments with less expensive models on top of this representation.

到目前为止，所有的BERT结果都使用了微调方法，其中一个简单的分类层添加到预先训练的模型，所有的参数都在下行任务中联合进行了微调。然而，基于特征的方法，即从预先训练的模型中提取固定特征，具有一定的优势。首先，并非所有的NLP任务都可以轻松地由Transformer编码器体系结构表示，因此需要添加特定于任务的模型体系结构。其次，一旦能够预先计算训练数据昂贵的表示，然后在这种表示之上用较便宜的模型进行许多实验，这对计算有很大的好处。

In this section we evaluate how well BERT performs in the feature-based approach by generating ELMo-like pre-trained contextual representations on the CoNLL-2003 NER task. To do this, we use the same input representation as in Section 4.3, but use the activations from one or more layers without fine-tuning any parameters of BERT. These contextual embeddings are used as input to a randomly initialized two-layer 768-dimensional BiLSTM before the classification layer.

在本节中，我们通过在CoNLL-2003NER任务上生成与elmo类似的经过预先训练的上下文表示，来评估基于特征的方法中BERT表示形式的好坏。为此，我们使用与第4.3节中相同的输入表示，但使用一个或多个层的激活，并对BERT的任何参数进行微调。这些上下文关系的嵌入用作分类层之前的ran-domly初始化的两层768-dimensionBiL-STM的输入。

Results are shown in Table 7. The best performing method is to concatenate the token representations from the top four hidden layers of the pre-trained Transformer, which is only 0.3 F1 behind fine-tuning the entire model. This demonstrates that BERT is effective for both the fine-tuning and feature-based approaches.

结果如表7所示。最好的执行方法是连接来自预先训练过的Transformer的顶层四个隐藏层的token表示，后面只需要微调整整个模型0.3F1。这表明BERT对于微调和基于特征的方法都是有效的。

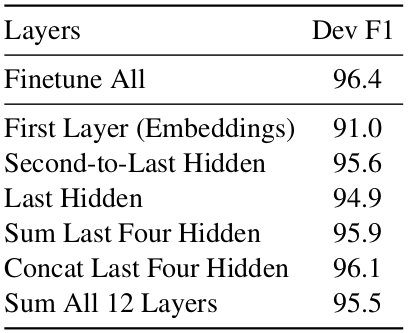


Table 7: Ablation using BERT with a feature-based approach on CoNLL-2003 NER. The activations from the specified layers are combined and fed into a two-layer BiLSTM, without backpropagation to BERT.

表7:基于CoNLL-2003NER的BERT特征层消融方法。来自指定层的激活被合并到一个两层的BiLSTM中，而不需要对BERT进行反向传播。

## **6 Conclusion** 结论

Recent empirical improvements due to transfer learning with language models have demonstrated that rich, unsupervised pre-training is an integral part of many language understanding systems. In particular, these results enable even low-resource tasks to benefit from very deep unidirectional architectures. Our major contribution is further generalizing these findings to deep bidirectional architectures, allowing the same pre-trained model to successfully tackle a broad set of NLP tasks.

由于语言模型的迁移学习，最近的经验改进表明，丰富的，无监督的预训是许多语言理解系统的一个组成部分。特别是，这些结果甚至使低资源任务能够从非常深的单向自由切割中获益。我们的主要贡献是将这些发现进一步发展为深层的双向扩展，允许相同的预先训练的模型成功地处理广泛的NLP任务。

While the empirical results are strong, in some cases surpassing human performance, important future work is to investigate the linguistic phenomena that may or may not be captured by BERT.

虽然实证结果是强有力的，在某些情况下超越了人类的表现，未来的重要工作是调查语言现象，可能或可能不捕捉BERT。