2.1 基于特征的无监督方法。

使用神经网络或者非神经网络的方法来学习单词的可应用表示是一个非常活跃的领域。预训练的词嵌入模型是现代nlp系统中的一个完整部分，它在嵌入学习上从头提供了一个显著的提升。为了预训练词向量，使用了从左到右的语言模型目标函数，和目标函数在左右的上下文中识别正确和错误的单词。 这些方法已推广到粗粒度，例如句子嵌入和段落嵌入。为了训练句子的向量表示，已经有过三类方法产生目标函数：1、给出前一个句子的向量表示后对下一个候选句子排序。2、生成下一个候选句子。3、自编码去噪目标函数。

elmo和它的前辈沿着不同的维度来研究传统的词嵌入模型。他们在一个从左到右和从右到左的语言模型中抽取上下文敏感的特征。每个单词的上下文表示向量是从左到右和从右到左的向量表示的拼接 。当将上下文单词嵌入应用到现有特定任务框架中时，elmo取得了最优的结果，问答，情感分析，命名实体识别等任务。kim sang使用LSTM基于左右上下文来预测当前单词的任务来学习上下文表示. 和elmo一样,这些模型都是基于特征的,而不是深度双向的.fedus的研究表明,完形填空任务可以提高文本生成模型的鲁棒性.

2.2 无监督微调方法

和基于特征的方法一样,在这个方向上的第一个工作仅仅使用未标注的数据训练词向量参数.近期,用于产生“上下文表示”的句子或文档编码器已经在大量未标注文本上训练过了,然后微调用于监督下游任务.这些方法的优势在于需要从头开始训练的参数很少.至少部分是归因于这个优势,OpenAI GPT在很多句子级别的任务中曾经实现了最优的结果,使用glue基准.从左到右的语言模型和自编码模型用于预训练这样的模型.

2.3 从监督数据上迁移学习

已经有很多研究在大型有监督数据集上完成了有效的迁移任务,例如自然语言推理和机器翻译.计算机视觉也说明了从大型数据集上进行迁移学习是很重要的.其中被证明有效的方法是微调从imagenet上预训练好的模型.

3 Bert

我们在这章介绍bert和它的详细细节实现.在我们的框架中有两步:预训练和微调.在预训练过程中,模型是在不同的任务中训练未标注数据.对于微调,bert模型是使用预训练参数进行初始化,而且所有的参数都在下游的标注数据上进行微调的.每个下游任务有不同的微调模型.图1的问答例子充当这一节的运行例子.

bert的一个显著特点是它统一了不同的任务.bert在预训练任务的结构和最终下游结构的差异非常小.模型结构：bert的模型结构是多层双向transformer编码器。transformer是基于vaswani提出的最初实现版本。由于transformer的使用已经变的非常普遍，而且 我们的实现几乎和原始版本一致。我们将忽略到冗长的背景描述，而让读者看vaswani的论文或者这篇博客。在这个工作中，我们将层数记为L，隐藏层大小为H，self-atttention头部个数为A。我们首先报告两个模型的大小**。bert\_base(L=12，H=768，A=12，总参数量为110M),**bert\_largee(L=24,H=1024,A=16,total paraameterrs=340M).

为了方便对比bert选择和GPT相同大小的模型。Bert的transformer使用双向self-attention的结构，相比之下GPT的transformer使用约束的self-attentionon,每个token只能和它左边的上下文关联。

输入输出表示

为了使**bert能处理大量的下游任务**，我们的输入表示可以编码一个句子或者两个句子（用于QA任务 ）。 在这个任务中，“句子”可以是由任意字符组成的连续文本而不用是一个真实的语言句子。一个句子对应bert中的输入token句子，它可能是一个句子也可能是两个句子打包在一起的。

我们使用wordpiece词嵌入，有3万个token词汇。每个句子的**第一个token永远是一个特殊分类token[cls]**。这个token的最后隐藏层可以收集整个句子的信息用于分类任务。“句子对”被打包起来输入到单个句子中。我们通过两个步骤来区分句子，首先，我们使用一个特殊符号[sep]来区分它们，然后，我们对每个token添加了一个学习到的embedding来表明它是属于A还是B句子。如图1所示，我们将输入embedding记为E，特殊token[cls]的最后隐藏层作为C，第i个输入token的最后隐藏向量记为Ti。对于一个给定的token，它的输入embedding是其对应token embedingg，分区embeding和位置embedding相加。这种构建如图2所示。

3.1预训练bert

与peters和radford不同，我们没有使用传统的从左到右或者从右到左的语言模型来预训练bert.我们使用两个无监督任务来训练bert，这节会详细介绍。

任务#1：masked LM（掩码语言模型）

直觉上来讲，我们可以相信，深度双向模型比左到右的模型或者左到右和右到左的拼接模型要更强。而不幸的是，标准的条件语言模型从左到右或者从右到左被训练，这是因为双向的条件会让每个单词间接“看到”它自己，这个模型会简单地在多层语境中预测目标单词。

**为了****训练深度双向表示**，我们简单地对输入token按一定比例随机掩码，然后预测这些掩码的token。我们将这个过程称作“掩码语言模型”（MLM），尽管这在一些文献中常被称为“完型填空”任务。在这种情况下，掩码tokens的最终隐藏层向量连接上softmax输出层对整个词汇表做分类，就和标准的语言模型一样。在我们所有实验中，我们使用WordPiece上每个句子15%的tokens进行掩码。与去噪自编码器相反，我们仅仅预测掩码单词而不是预测整个输入。

尽管这种方式可以让我们获得双向的预训练模型，它的缺点是会产生预训练和微调过程中的不匹配，因为[mask]token在微调过程不存在。为了缓和这个问题，我们不总是用“mask”单词token作为实际[MASK]token。训练数据生成器随机选择了15%的token来进行预测。如果第i个token被选中了，那我们将以80%的概率替换成[mask]token，10%的概率替换成任意token，10%的概率保持不变。然后Ti会用于预测原来的token使用交叉熵（softmax）损失函数。我们在附录中比较这个过程的不同变种。

任务#2：预测下一个句子（NSP）

很多重要的下游任务例如问答任务（QA）和自然语言推理（NLI）都是基于理解两个句子之间的联系，而这不是直接通过语言模型捕捉到的。为了**训练模型****理解句子间的联系**，我们将预测下一个句子任务作为二分类任务并且从任意的单语语料上进行预训练。具体来讲，对于预训练样本中的句子A和句子B，50%的B是A的下一个句子（标签是IsNext），50%的B是从语料中任选的句子（标签是NotNext）。正如我们在

图1中展示的，C是用于下一个句子预测任务（NSP）。尽管这个做法很简单，我们在5.1节展示了这种任务的预训练对**QA和NLI**是非常有利的。

这个NSP任务的目标函数和jernite提出的表征学习目标函数很相似。然而以往的研究中，仅仅只有句子嵌入向量被迁移到下游任务，而bert将所有参数都用于初始化下游任务的模型参数。

预训练数据

我们的预训练过程很大比例上遵循了语言模型预训练方面已有的文献。对于预训练语料我们使用BooksCorpus(800M词语)和英语维基百科。对于维基百科，我们仅仅抽取文本段落，而忽略列表，表格和文章头部。注意语料要使用文档级的而不能用打乱的句子级别语料例如Billion Word Benchmark，因为需要抽取长的连续句子。

3.2 微调Bert

微调过程很简单，只需替换输入和输出，因为transformer中的自适应机制让bert能对很多下游任务建模，无论下游是涉及到单个文本还是一对文本。对于涉及到一对文本的应用，传统的方法是在应用双向交叉注意力之前独立编码句子对，如parikh的论文。然而bert使用自注意力机制统一了这两个过程，直接编码拼接起来的文本对，bert的自注意力可以很有效的包含两个句子的双向交叉注意力。对于每个任务，我们**简单地将任务特定的输入和输出输入到bert中然后微调所有的参数**。在输入中，预训练的句子A和句子B和以下任务中的输入相同：1 、段落中的一对句子2、“假设前提对”中的句子。3、问答任务中的“问题-通道”4、退化成“文本-空集”对在文本分类和句子标注任务中。

**在输出中，token的表示可以被连接到输出层用于token级别的任务，句子标注和问答任务。而且[CLS]token可以连接输出层做分类任务**，例如推理和句子分析任务。和预训练相比，微调的成本低的多。这篇文章中的所有结果都可以复现，只要使用一个云TPU跑一个小时或者一个GPU跑几个小时，前提是从相同的预训练模型开始训练。我们在第四节描述了特定任务的细节，更多的细节在附录中找到。

4 实验

在这一节，我们展示bert在11项nlp任务中的微调结果。

4.1 glue

通用语言理解衡量基准（GLUE）是不同的自然语言理解任务的基准。对于GLUE数据集的具体描述可以在附录B.1中找到。