我们介绍了一种新的深度语境词表示方法，它可以同时对词所使用的复杂词性（例如语法和语义）进行建模。而且它建模的方式会随着语境不同而不同（即，多义性建模）。我们词向量是从深层双向语言模型的内部状态学习到的，这个语言模型在大型的语料库上预训练。我们将展示这种词向量能很容易地加入到已有的模型中，并且能够显著地提高6项最NLP任务的最优（SOTA）表现。这6项NLP任务包括问答，文本推理和文本分析。我们同样分析说明了暴露预训练网络的深度内部向量是非常关键的，这一步使得下游模型能混合不同类型的半监督标志。

我们的向量表示与传统类型的词向量不同之处在于，每个token的映射向量都是整个输入句子的函数。我们使用双向LSTM在大型语料上训练语言模型，然后取出BILSTM内部的向量作为词向量。也是这个原因，我们称它们为ELMos表示（语言模型上的embedding）。和以前学习语境词向量的方法不同，elmos的向量更深，因为它是BILM所有内部层的函数。更具体来讲，我们对每种不同的下游任务学习了每个输入单词上向量的线性组合，这相比直接使用LSTM最顶层向量显著提高了模型表现。

以这种方式结合内部状态产生了语义非常丰富的词向量表示。使用内在评估标准，我们展示了更高层LSTM状态更擅长捕捉与语境相关的单词含义（例，它们无需修改就能在有监督的单词语义消歧任务上表现很好）而更低层的状态更擅长捕捉语法相关的含义（它们可以用于做词性标注）。同时利用所有这些

向量是很有帮助的，它会使得学习到的模型挑选对不同下游任务最有利的半监督类型。

扩展的实验说明了elmo展示在实践中表现非常好。我们首先说明它可以很轻易地加入到6种现有语言理解问题中。elmo向量的加入能显著地提高每种任务的最优表现，包括相关误差减少20%。在任务上对比，elmo表现优于cove，cove是使用机器学习翻译自编码来计算语境表示的。最终在对elmo和CoVe分析后显示LSTM深层信息的表现比仅仅使用顶层LSTM更优。我们的模型和代码已公开，我们希望elmo能在其他nlp任务中也获得相似的提升。

3 elmo：语言模型中的词向量

和广泛使用的词向量不同，elmo词向量是整个输入句子的映射，这节将详细描述。它们在biLM的最上面两层进行计算使用字符卷积，作为内部网络状态的线性函数。这一步骤使得我们能做半监督学习，这里BiLM是在大规模语料上预训练的，并且很轻易地合并了很多现有的NLP神经网络结构。

3.1 双向语言模型

给定N个tokens的序列，(t1,t2，...,tN)，一个前向的语言模型计算序列的概率通过对tokens tk建模，在给定t1，...,tk。一个反向的LM和前向的LM相似，除了它在序列的运行顺序上是相反的，给出未来的语境来预测以前的单词。 这可以通过和前向LM相同的方法来实现，在L层的深度模型上第j个LSTM层给定(tk+1,...,tN)生成的tk向量记为h.一个双向语言模型合并了前向和后向语言模型。我们的公式联合最大化前向和后向的log似然函数。我们对前向和后向的token展现层和softmax层都绑定了参数，前向和后向在LSTM层的参数不一样。总体来讲，这个公式和peters的方法很像，除了我们在两个方向上共享了参数而不是使用独立的参数。在下一节，我们将与前面的方法区分开，使用一种新的方法来学习词向量表示，这是一种BiLM层的线性组合。

3.2 elmo

elmo是一个BiLM中间层表示的特定组合任务。对于每个token tk，一个L层的BiLM计算了2L+1个表示，表示集合为Rk。（每个LSTM层的输出计算两个，token词向量一个)为了兼容下游的模型，elmo将R中所有层压缩到一个向量中，elmok = E（Rk）。最简单的情况，elmo仅仅是顶层的输出，就和tagLM和CoVe一样。更加普遍的，我们计算所有BiLM层对特定任务的权重：其中S是softmax归一化后的权重而尺度参数R允许任务模型放缩整个elmo向量。为了补救最优化过程R在实践中是非常重要的（详情可以看超参数部分）。考虑到每个BiLM层的激活函数有不同的分布，在一些情况下使用对每个biLM层在加权之前使用LN标准化是有用的。

3.3 对有监督NLP任务使用biLM

给定一个预训练好的biLM和对一个目标NLP任务上的有监督结构，很容易使用BiLM来提升任务模型的性能。我们简单的运行了BiLM并且记录了每个单词所有层的表现。然后，我们让下游任务模型学习所有表示的线性组合，和上面描述的一样。首先考虑没有BiLM的有监督模型的最底层。大多数有监督NLP模型

在最底层都有相同的结构，这允许我们用一种统一的方法将elmo加入进去。给定一个tokens序列(t1,...,tN)，常规的做法是对每个token构建一个内容无关的token表示xk，xk是基于字符或者词

语的预训练词向量模型。然后，这个模型组成了一个内容敏感的表示hk，特别是使用双向RNNs，CNNs或者前向神经网络。

为了将elmo加入到有监督模型中，我们首先冻结biLM上的权重，然后拼接elmo向量elmok和xk，然后将elmo提升后的表示[xk,elmok]输入到任务级别的RNN中。对于一些任务，例如SNLII, SQUAD,我们观察到将elmos向量加入到具体任务RNN输出中会有提升，即使用[hk,elmok]来替换hk（RNN的输出）。同时有监督模型的剩余部分保持不变，这个增加部分也可以加入在更加复杂的神经模型中。例如，第四节的SNLI模型，在BiLSTMs模型后跟着双向注意力层，或者共参分辨率模型其中聚类模型是在BiLSTM层之上的。最后，我们发现适量的添加dropout是有效的，而且有时需要加入二范数来正则化elmo的权重。这会减轻elmo权重的偏差并且输出更接近所有层的平均值。

3.4 预训练双向语言模型结构

这篇文章中的预训练双向语言模型和joze和kim的很相似，修改部分是为支持联合训练两个方向的模型和在LSTM层之间添加残差结构。我们在这个任务中主要关注大规模BiLMs，和peters一样，他强调要使用大规模语料来训练仅前向的biLMs模型。 为了平衡所有语言模型的复杂性带来的模型大小和算力需求，对于下游任务仅仅维持一个纯粹的字符级输入展示，我们对于joz的单个最好模型CNN-BIG-LSTM的所有嵌入层和隐藏层减少了一半的参数。最终的模型使用了L=2BILSTM层有4096个节点和512维的映射和一个残差结构连接从第一层到第二层。内容不敏感内心的展现使用2048字符n-gram残差卷积算子紧跟着两个high way层，和一个线性映射到512维的表示。作为结果，biLM对每个输入token提供了三层展示，包括那些训练集外的由于纯粹的字符输入。与之相反，传统的单词embedding方式仅仅提供了固定词表中对于token的一层展示。 在1B benchmark数据集上训练了10个epoch后，平均的前向和后向复杂度到了39.7，和前向CNN-BIG-LSTM的30对比有很大提升。更普遍来讲，我们发现前向和后向的困惑度（perplexity)大致相等，其中后向的值略低一点。

一旦预训练后，biLM可以计算任何任务的表示。在一些情况下，在目标领域中微调数据biLM模型可以带来困惑度的显著下降和性能大幅提升。这可以被看成是一种biLM的领域迁移。结果上，在大多数情况下我们使用微调后的biLM在下游任务使用。细节部分看附录材料。

命名实体抽取

2017年最好的方法是使用预训练的单词向量，基于字符的CNN表示，两个BILSTM层和一个条件随机场损失，和collobert一样。在这个基础上，我们的方法从90%提升到92%。