ELMO —— 《Deep contextualized word representations》论文解读

完成时间：2022.9.25 、2022.9.27、2022.10.3

论文发表时间：2018年

论文作者：Matthew E. Peters；Mark Neumann； Mohit Iyyer；Matt Gardner；Christopher Clark；Kenton Lee；Luke Zettlemoyer

Abstract

我们引入了一种新型的深度上下文单词表示，这种表示既建模了复杂的单词使用特征((例如，语法和语义))，也建模了这些表示**在不同的语境下的区别**(例如，对一词多义进行建模)。我们的词向量是深度双向语言模型(biLM)内部状态的学习函数，该模型是在大型文本语料库上预训练的。我们表明，这些表示可以很容易地添加到现有的模型中，并显著提高了六个具有挑战性的NLP问题的技术水平，包括问题回答、文本蕴涵和情感分析。我们还提出了一项分析，表明暴露**预训练网络的深层内部是至关重要的**，它允许下游模型混合不同类型的半监督信号。

1、Introduction

学习单词向量表示遇到的两个挑战：

1．单词使用的复杂特征（例如语法、语义）

2．这些用法如何在语言环境中变化（例如，一词多义）

本论文介绍了一种新型的深度上下文单词表示，直接解决上面两个挑战，同时可以很容易地集成到现有的模型中。

ELMO (Embeddings from Language Models) 表示是深度的，因为它们是biLM所有内部层的函数，这比仅使用顶部LSTM层显著提高了性能。评估表面，较高级的LSTM状态捕获了单词意义的上下文相关方面(例如，它们可以在不进行修改的情况下很好地执行监督的词义消歧任务)，而较低级的状态建模语法方面(例如，它们可以用于词性标记)。

2、Related work

之前提出的学习词向量的方法都只允许在每个词上有**一个**独立于上下文的表示。我们的方法还通过使用字符卷积从子词单元中获益，并且我们无缝地将多含义信息合并到下游任务中，而无需显式训练来预测预定义的含义类。

在本文中，我们充分利用了对丰富的单语数据的访问，并在大约3000万个句子的语料库上训练我们的biLM (Chelba et al, 2014)。我们还将这些方法推广到深度上下文表示中，我们证明这些方法在**一**系列不同NLP任务中都能很好地工作。

先前的研究也表明，不同层次的深层biRnns编码不同类型的信息。例如，在深度LSTM的**较低级别**引入多任务语法监督(例如词性标记)可以提高更高级别任务的总体性能，如依赖分析或CCG超级标记。在**一**个基于RNN的编码器-解码器机器翻译系统中，Belinkov等人(2017)表明，在2层LSTM编码器的第**一**层学习的表示比在第二层更好地预测POS标记。最后，用于编码单词上下文的LSTM的**顶层**已被证明可以**学习词义**的表示。我们表明，ELMo表示的修改语言模型目标也可以诱导类似的信号，这对于学习混合了这些不同类型的半监督的下游任务的模型非常有益。

Dai和Le(2015)和Ramachandran等人(2017)使用语言模型和序列自动编码器预训练编码器-解码器对，然后使用特定于任务的监督进行微调。相反，在用未标记的数据对biLM进行预训练之后，我们固定了权重并添加了额外的特定于任务的模型容量，允许我们利用大型、丰富和通用的biLM表示，以应对下游训练数据大小决定了更小的监督模型的情况。

3 ELMo: Embeddings from Language Models

与大多数广泛使用的单词嵌入不同，ELMo单词表示是**整个输入句子的函数**，如本节所述。它们是在**具有字符卷积的双层biLMs(第3.1节)**之上计算的，作为**内部网络状态的线性函数**(第3.2节)。这种设置允许我们进行半监督学习，在这种情况下，**biLM被大规模地预训练**(第3.4节)，并很容易被纳入到广泛的现有神经NLP体系结构中(第3.3节)。

3.1 Bidirectional language models

给定N个tokens序列（t_1，t_2，…，t_N），前向语言模型通过在给定历史（t_1、…、t_{k-1}）的条件下来建模token t_k的概率，从而来计算N个序列的概率：

$$p(t_1, t_2, \ldots, t_N) = \prod_{k=1}^{N} p(t_k \mid t_1, t_2, \ldots, t_{k-1}).$$

最新的神经语言模型计算**一**个与上下文无关的token表示$\mathbf{x}_k^{LM}$(通过token embedding或字符上的CNN)，然后将其穿过前向LSTM的L层。在每个位置k，每个LSTM层输出**一**个上下文相关的表示$\overrightarrow{\mathbf{h}}_{k,j}^{LM}$，其中j = 1, …, L。顶层LSTM输出$\overrightarrow{\mathbf{h}}_{k,L}^{LM}$，用于预测下**一**个令牌$t_{k+1}$与Softmax层。

反向LM与正向LM类似，不同的是它反向运行序列，在给定未来上下文的情况下预测前**一**个标记：

$$p(t_1, t_2, \ldots, t_N) = \prod_{k=1}^{N} p(t_k \mid t_{k+1}, t_{k+2}, \ldots, t_N).$$

它可以用类似于正向LM的方式实现，每个反向LSTM层j 在 L层深度模型中产生给定(t_{k+1}, …, t_N)的t_k的表示 $\overleftarrow{\mathbf{h}}_{k,j}^{LM}$。

biLM结合了前向和反向LM。我们的公式共同最大化了正向和反向的对数似然函数：

$$\sum_{k=1}^{N} \Big(\log p(t_k \mid t_1, \ldots, t_{k-1}; \Theta_x, \overrightarrow{\Theta}_{LSTM}, \Theta_s) + \log p(t_k \mid t_{k+1}, \ldots, t_N; \Theta_x, \overleftarrow{\Theta}_{LSTM}, \Theta_s) \Big)$$

我们在正向和反向绑定token表示(Θ_x)和Softmax层(Θ_s)的参数，同时在每个方向为LSTM维护单独的参数。总的来说，这个公式类似于Peters等人(2017)的方法，**不同的是我们在方向之间共享了一些权重**，而不是使用完全独立的参数。在下一节中，我们将与前面的工作不同，介绍一种学习单词表示的新方法，它是biLM层的线性组合。

3.2 ELMo

ELMo是biLM中间层表示的特定于任务的组合。对于每一个token t_k，L层 biLM计算一组2L+1表示，

$$\begin{aligned} R_k &= \{\mathbf{x}_k^{LM}, \overrightarrow{\mathbf{h}}_{k,j}^{LM}, \overleftarrow{\mathbf{h}}_{k,j}^{LM} \mid j = 1, \ldots, L\} \\ &= \{\mathbf{h}_{k,j}^{LM} \mid j = 0, \ldots, L\}, \end{aligned}$$

对于每一个biLSTM层，其中 $\mathbf{h}_{k,0}^{LM}$ 是token层和$\mathbf{h}_{k,j}^{LM} = [\overrightarrow{\mathbf{h}}_{k,j}^{LM}; \overleftarrow{\mathbf{h}}_{k,j}^{LM}]$。为了包含在下游模型中，ELMo将R中的所有层折叠成一个单一向量，$\mathbf{ELMo}_k = E(R_k; \mathbf{\Theta}_e)$。在最简单的情况下，ELMo只选择顶层，$E(R_k) = \mathbf{h}_{k,L}^{LM}$。更一般地，我们计算所有biLM层的任务特定权重：

$$\mathbf{ELMo}_k^{task} = E(R_k; \Theta^{task}) = \gamma^{task} \sum_{j=0}^{L} s_j^{task} \mathbf{h}_{k,j}^{LM} \tag{1}$$

在(1)中，s^{task}是softmax归一化权值，标量参数γ^{task}可以使得任务模型缩放整个ELMo向量。γ在辅助优化过程中具有实际重要性(详见补充材料)。**考虑到每个biLM层的激活有不同的分布**，在某些情况下也有助于在加权前对每个biLM层进行 层归一化 。

正则化参数γ的选择很重要，若值较大，例如γ=1，则相当于将加权函数简化为层的简单平均值；而较小的值(如γ = 0.001)则可以有效的学习层的权重变化。

3.3 Using biLMs for supervised NLP tasks

给定一个预先训练的biLM和目标NLP任务的监督架构，使用biLM改进任务模型是一个简单的过程。我们只需**运行biLM并记录每个单词的所有层表示。**然后，我们让最终任务模型学习这些表示的线性组合，如下所述。

5043 字

首先考虑没有biLM的监督模型的最低层。大多数受监督的NLP模型在最低层共享公共体系结构，允许我们以一致的、统一的方式添加ELMo。给定一个token序列(t_1，…， t_N)，使用预先训练的单词嵌入和可选的基于字符的表示，为每个token位置形成一个上下文**独立**的token表示X_k是标准的。然后，该模型通常使用双向RNNs、CNNs或前馈网络 形成一个上下文**相关**的表示h_k。

为了将ELMo添加到监督模型中，我们首先冻结biLM的权重，然后用x_k连接ELMo向量 $\mathbf{ELMo}_k^{task}$，并将ELMo增强表示$[\mathbf{x}_k; \mathbf{ELMo}_k^{task}]$传递到任务RNN中。对于某些任务，我们观察到进一步的改进，通过在任务RNN的输出中加入ELMO，引入另一组输出比线性权值，并将h_k替换为 $[\mathbf{h}_k; \mathbf{ELMo}_k^{task}]$.。由于监督模型的其余部分保持不变，这些添加可以在更复杂的神经模型上下文中发生。例如，参见第4节中的SNLI实验，其中bi-attention 跟随biLSTMs，或者或在biLSTM之上分层聚类模型的共指分辨实验。

最后，我们发现在ELMo中添加适量的dropout是有益的，在某些情况下，通过在损失中添加 $\lambda\|\mathbf{w}\|_2^2$ 来正则化ELMo权重。这对ELMo权值施加了一个归纳偏差，以接近所有biLM层的平均值。

3.4 Pre-trained bidirectional language model architecture

本文预训练的bilm与Kim等人(2015)的体系结构相似，但经过修改，支持两个方向的联合训练，并在LSTM层之间增加了残余连接。在这项工作中，我们将重点放在大规模的biLMs上，因为Peters等人(2017)强调了在仅前向的LMs和大规模训练上使用biLMs的重要性。

为了平衡整个语言模型的困惑度与模型大小和下游任务的计算需求，同时保持纯粹基于字符的输入表示，我们从Józefowicz等人的单一最佳模型CNN-BIG-LSTM中**减半所有嵌入和隐藏维度**。最终模型使用带有4096个单元和512维投影的L = 2 biLSTM层和从第一层到第二层的残差连接。上下文不相关的类型表示使用2048个字符n-gram卷积滤波器，然后是两个高速公路层和向下到512表示的线性投影。因此，**biLM为每个输入token提供了三层表示**，包括由于纯字符输入而在训练集之外的表示。相比之下，传统的词嵌入方法只为固定词汇表中的token提供一层表示。

一旦经过预先训练，biLM就可以为任何任务计算表示。在某些情况下，**对领域特定数据的biLM进行微调会显著降低复杂性，并提高下游任务性能。**这可以看作是biLM的一种域传输。因此，在大多数情况下，我们在下游任务中使用经过微调的biLM。详情见补充材料。

4 Evaluation

TASK	PREVIOUS SOTA		OUR BASELINE	ELMO + BASELINE	INCREASE (ABSOLUTE/ RELATIVE)
SQuAD	Liu et al. (2017)	84.4	81.1	85.8	4.7 / 24.9%
SNLI	Chen et al. (2017)	88.6	88.0	88.7 ± 0.17	0.7 / 5.8%
SRL	He et al. (2017)	81.7	81.4	84.6	3.2 / 17.2%
Coref	Lee et al. (2017)	67.2	67.2	70.4	3.2 / 9.8%
NER	Peters et al. (2017)	91.93 ± 0.19	90.15	92.22 ± 0.10	2.06 / 21%
SST-5	McCann et al. (2017)	53.7	51.4	54.7 ± 0.5	3.3 / 6.8%

The Stanford Question Answering Dataset **(SQuAD)**。**斯坦福问题回答数据集**(SQuAD) 包含10万多个来自人群的问题-答案对，其中答案是给定维基百科段落的span。我们的baseline模型在双向注意力流模型的基础上增加了一个自注意力层，并简化了一些池化操作，并将LSTM替换为门控循环单元(GRU)。在将ELMo添加到基线模型之后，测试集F1从81.1%提高到85.8%，提高了4.7%。

文本蕴含。文本蕴涵是在给定一个“前提”的情况下，判断一个“假设”是否正确的任务。**斯坦福自然语言推理(SNLI)**语料库提供了大约550K个假设/前提对。我们的baseline是Chen等人(2017)的ESIM序列模型，它使用biLSTM对前提和假设进行编码，然后是矩阵注意力层、局部推理层、另一个biLSTM推理组合层，最后在输出层之前进行池化操作。总的来说，将ELMo添加到ESIM模型中，在五个随机种子中平均提高了0.7%的准确性。

语义角色标记(SRL)为句子的谓词-参数结构建模，通常被描述为回答“谁对谁做了什么”。He将SRL建模为BIO标记问题，并使用了纵横交错的8层深度biLSTM。如表1所示，当将ELMo添加到He的重新实现时，单一模型测试集F1从81.4%上升到84.6%，这是OntoNotes基准测试的最新水平，甚至比之前的最佳集成结果提高了1.2%。

> BIO标注就是联合标注的一种，具体地B、I、O 分别表示Begin Inner Other。
>
> - 进一步地来说，B-X表示元素是X类型并且位于片段的起始位置，I-X表示元素是X类型并且位于元素片段的中间，O则表示元素不属于X类型。

Coreference resolution —— 共指解析实验。论文里简称为**Coref**。

> - Coreference resolution (共指解析)是自然语言处理(nlp)中的一个基本任务，目的在于自动识别表示**同一个实体**的名词短语或代词，并将他们归类。
> - 实体出现在自然语言文本中的时候可能会有不同的形式(or名字)。例如文章中出现奥巴马这个词，它有的时候是美国总统，有的时候是奥巴马，甚至有的时候是一个简单的代词他。当这些名词短语或代词出现在一起时，我们根据我们已有的知识或者是上下文信息都清楚地知道它们指代的是同一个实体，那么怎么让计算机自动识别这些指向同一个实体的名词短语或代词呢？这就是coreference resolution要完成的工作。
> - 实体(entity)：知识库中完整定义的，唯一存在的条目，在coreference resolution这个任务中，每一个实体都可以看作是指代它的名词短语或代词构成的集合(巴拉克 - 奥巴马={美国总统, 奥巴马, 第44任美国总统, 他})。

5、Analysis

本节提供消融分析来验证我们的主要声明，并阐明ELMo表示的一些有趣方面。

- 第5.1节表明，**在下游任务中使用深度上下文表示**比以前只使用顶层的工作提高了性能，而不管它们是由biLM还是MT编码器产生的，而且ELMo表示提供了最佳的总体性能。
- 第5.3节探讨了在biLMs中捕获的**不同类型的上下文信息**，并使用两个内在评估来表明**语法信息在较低的层中更好地表示，而语义信息在较高的层中捕获**，这与MT编码器一致。它还表明，我们的biLM始终提供比CoVe更丰富的表示。
- 此外，我们分析了ELMo在任务模型中包含的位置(第5.2节)、训练集大小(第5.4节)的敏感性，并在任务中可视化ELMo学习权重(第5.5节)。

5.1 Alternate layer weighting schemes

$$\mathbf{ELMo}_k^{task} = E(R_k; \Theta^{task}) = \gamma^{task} \sum_{j=0}^{L} s_j^{task} \mathbf{h}_{k,j}^{LM} \tag{1}$$

- 正则化参数γ的选择很重要，若值较大，例如γ=1，则相当于将加权函数简化为层的简单平均值；而较小的值(如γ = 0.001)则可以有效的学习层的权重变化。

Task	Baseline	Last Only	All layers	
			λ=1	λ=0.001
SQuAD	80.8	84.7	85.0	**85.2**
SNLI	88.1	89.1	89.3	**89.5**
SRL	81.6	84.1	84.6	**84.8**

- 包含来自最后一层的上下文表示可以提高baseline性能，而与仅使用最后一层相比，包含来自所有层的表示可以提高总体性能。
- 若允许任务模型学习单个层的权重，相比所有层相加求平均又提高了0.2% (λ=1 vs λ=0.001)。在ELMo的大多数情况下，小λ是首选，尽管对于具有较小训练集的NER任务，结果对λ不敏感(未显示)。

> 最好的组合就是：使用**小λ**，并考虑**所有层**的上下文表示。

5.2 Where to include ELMo?

Task	Input Only	Input & Output	Output Only
SQuAD	85.1	**85.6**	84.8
SNLI	88.9	**89.5**	88.7
SRL	**84.7**	84.3	80.9

如表3所示，在SNLI和SQuAD的输入和输出层都包含ELMo比仅输入层提高，但对于SRL(和共指解析，图中未显示)，当它仅包含在输入层时性能最高。**一个可能的解释是SNLI和SQuAD架构都**

在biRNN之后使用注意力层，所以在这一层引入ELMo允许模型直接关注biLM的内部表示。在SRL的例子中，特定于任务的上下文表示可能比来自biLM的上下文表示更重要。

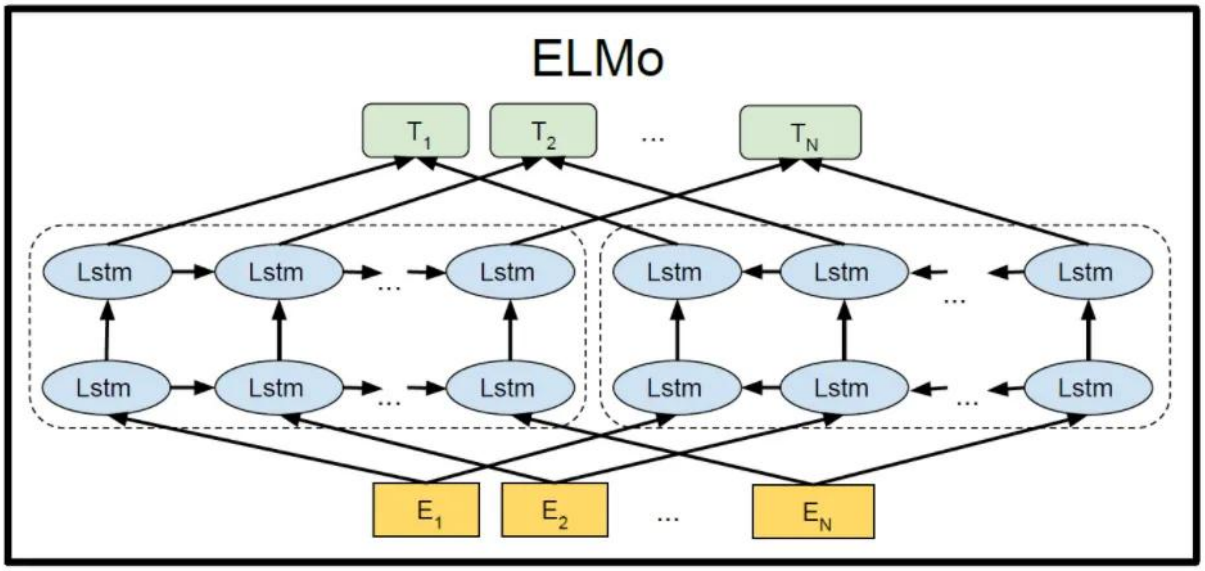
5.3 What information is captured by the biLM’s representations? —— biLM捕获了什么信息

由于添加ELMo比单独的词向量提高了任务性能，因此biLM的上下文表示必须编码对NLP任务通常有用的信息，而这些信息没有被词向量捕获。直观地说，biLM必须使用上下文消除单词的含义的歧义。以“play”为例，这是一个高度多义词。

	Source	Nearest Neighbors
GloVe	play	playing, game, games, played, players, plays, player, Play, football, multiplayer
biLM	Chico Ruiz made a spectacular play on Alusik ’s grounder {...}	Kieffer , the only junior in the group , was commended for his ability to hit in the clutch , as well as his all-round excellent play .
	Olivia De Havilland signed to do a Broadway play for Garson {...}	{...} they were actors who had been handed fat roles in a successful play , and had talent enough to fill the roles competently , with nice understatement .

- 上表的顶部列出了使用GloVe向量的“play”的最近邻居。它们分布在几个词性中(例如，作为动词的“playing”、“playing”和作为名词的“player”、“game”)，但大部分集中在与运动相关的“play”的意义上。
- 相比之下，下面两行显示SemCor数据集(见下文)在源句中使用“play”的biLM的上下文表示的最近邻居句。在这些情况下，biLM能够消除源句中的词性和词义歧义。
- 这些观察结果可以通过与Belinkov et al.（2017）类似的上下文表征的内在评估进行量化。为了分离由biLM编码的信息，使用该表示直接对细粒度词义消歧(WSD)任务和词性标记任务进行预测。使用这种方法，还可以与CoVe进行比较，并跨每个单独的层进行比较。

总结



- 第一层是单词特征（从下到上）
- 第二层是句法特征
- 第三层是语义特征

E2指向左边第二个lstm，会接收E1传递的左边的lstm，即上文信息。

E2指向右边第二个lstm，会接收E2传递的右边的lstm，即下文信息。

然后上下文信息再分别经过LSTM得到，再把得到的「上文信息 + 下文信息」传递给T。

> 本质上，就是词向量+两层上下文信息。即，ELMO不只是训练一个 Q 矩阵，我还可以把这个词的上下文信息融入到这个 Q 矩阵中。

论文原文

Deep contextualized word representations.pdf (416 KB)

5043 字

aadee4f5bb3d.png&title=ELMO%20%E2%80%94%E2%80%94%20%E3%80%8ADeep%20contextua