

ELMo算法介绍

原创 M 没啥深度 2018-09-17

这篇介绍一下ELMo算法(<https://arxiv.org/pdf/1802.05365.pdf>)。按说应该加入前面的《关于句子embedding的一些工作简介》系列，但是严格来讲，只能说它通过自己产生的word embedding来影响了句子embedding, 所以干脆另写一篇吧。

Introduction

作者认为好的词表征模型应该同时兼顾两个问题：一是词语用法在语义和语法上的复杂特点；二是随着语言环境的改变，这些用法也应该随之改变。作者提出了 *deep contextualized word representation* 方法来解决以上两个问题。这种算法的特点是：每一个词语的表征都是整个输入语句的函数。具体做法就是先在大语料上以language model为目标训练出bidirectional LSTM模型，然后利用LSTM产生词语的表征。ELMo 故而得名(Embeddings from Language Models)。为了应用在下流的NLP任务中，一般先利用下游任务的语料库(注意这里忽略掉label)进行language model的微调,这种微调相当于一种domain transfer; 然后才利用label的信息进行supervised learning。

ELMo表征是“深”的，就是说它们是biLM的所有层的内部表征的函数。这样做的好处是能够产生丰富的词语表征。高层的LSTM的状态可以捕捉词语意义中和语境相关的那方面的特征(比如可以用来做语义的消歧)，而低层的LSTM可以找到语法方面的特征(比如可以做词性标注)。如果把它们结合在一起，在下流的NLP任务中会体现优势。

Bidirectional language models

ELMo 顾名思义是从Language Models得来的embeddings，确切的说是来自于Bidirectional language models。具体可以表示为：

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

和

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

作为语言模型可能有不同的表达方法，最经典的方法是利用多层的LSTM，ELMo的语言模型也采取了这种方式。所以这个Bidirectional LM由stacked bidirectional LSTM来表示。

ELMO

对于每一个token，一个L层的biLM要计算出共 $2L + 1$ 个表征：

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

这里的x是输入的token层的表示，h是每层LSTM的输出，k代表输入序列的位置，j代表层数。第二个等式是generalized的表示：j=0代表输入层，这时h=x。

在下游的任务中，ELMo把所有层的R压缩在一起形成一个单独的vector。(在最简单的情况下，可以只保留最后一层。)

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

式子里的系数都是在训练中产生。具体来讲如何使用ELMo产生的表征呢？对于一个supervised NLP任务，可以分以下三步：

1. 产生 pre-trained biLM 模型。模型由两层 bi-LSTM 组成，之间用 residual connection连接起来。
2. 在任务语料上(注意是语料，忽略label)fine tuning上一步得到的biLM模型。可以把这一步看为biLM的domain transfer。
3. 利用ELMo的word embedding来对任务进行训练。通常的做法是把它们作为输入加到已有的执行目标任务的模型中，一般能够明显的提高原模型的表现。

印象中太深的NLP方面的模型基本没有，这和Computer Vision领域非常不一样。当然这也是所解决问题的本质决定：Image的特征提取在人脑里就是从低阶到高阶的过程，深层网络有助于高级特征的实现。对于语言来讲很难定义这样的过程，这篇文章的两层biLM加residual connection的架构比较少见(Google的transformer是多层网络+residual connection另一个例子)。文章认为低层和高层的LSTM功能有差异：低层能够提取语法方面的信息；高层擅于捕捉语义特征。

Evaluation and Analysis

效果

先看一下在 QA, Textual entailment, Semantic role labeling, Coreference resolution, NER, 和 Sentiment analysis上的表现。

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 \pm 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 \pm 0.19	90.15	92.22 \pm 0.10	2.06 / 2.1%
SST-5	McCann et al. (2017)	53.7	51.4	54.7 \pm 0.5	3.3 / 6.8%

和state of art比基本上每个任务都有明显的改善。表中的OUR BASELINE在论文中有详细介绍，它指的是作者选定的某些已有的模型。ELMo+BASELINE指的是作者把ELMo的 word representation作为输入提供给选定的模型。这可以清楚的比较在使用和不使用ELMo词嵌入时的效果。

多层和最后一层

公式(1)用各层表征的叠加来代表相应位置的向量，作者在下表中比较了仅仅使用最后一层的效果。

Task	Baseline	Last Only	All layers	
			$\lambda=1$	$\lambda=0.001$
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

显然多层的叠加效果好于仅使用最后的一层。最后一列里代表的是网络参数 regularization的大小。结果说明合适的regularization有好处。

存在于输入层和输出层


其实ELMo不仅可以作为下游模型的输入，也可以直接提供给下游模型的输出层。

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	84.9

上表说明有时候同时提供给下游模型的输入和输出层效果更好。

biLM捕捉到的词语信息

ELMo提高了模型的效果，这说明它产生的word vectors捕捉到其他的word vectors没有的信息。直觉上来讲，biLM一定能够根据context区别词语的用法。下表比较了Glove和biLM在play这个多义词上的解释。

Source		Nearest Neighbors
GloVe	play	playing, game, games, played, players, plays, player, Play, football, multiplayer
biLM	Chico Ruiz made a spectacular <u>play</u> 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 <u>play</u> .
	Olivia De Havilland signed to do a Broadway <u>play</u> for Garson {...}	{...} they were actors who had been handed fat roles in a successful <u>play</u> , and had talent enough to fill the roles competently , with nice understatement .  没啥深度

对于Glove来说，play的近义词一股脑包含了不同的语法上的用法：动词(playing, played), 名词(players, game)。但是biLM能够同时区分语法和语义：第一个例子中的play名词，表示击球，第二个例子中play也是名词，表示表演。显然biLM能够在表示词语嵌入时考虑到context的信息。

总结

ELMo在处理很多NLP下游任务中表现非常优异。但是我想这跟它集中在产生更好的词语级别的embedding是有关系的。过去介绍到的一些其他的算法，比如Quick thoughts也是利用了语言模型作为句子的encoder;还有InferSent使用biLSTM作为encoder。和ELMo相比，它们都显得“野心”太大：它们为下游的NLP任务提供了句子embedding的解决方案：即直接利用它们的pretrained encoder，最终的预测无非是在上面加上softmax的classifier。对比而言ELMo要单纯很多，它只提供了word级别的解决方案：利用它的pretrained biLM来产生word embedding,然后提供给下游的已有模型。这里的下游模型往往是sequence model，其效果已经在相应的NLP任务上得到验证。这时有新的兼具语法语义及环境特征的word embedding的加持，难怪效果会更好。更不要说，ELMo还在任务语料库上小心翼翼的再进行过一轮微调，更是保证了对新domain的adaptation。

喜欢此内容的人还喜欢

浙江之声记者汪婷：见证浙江法院司法为民、探索创新的精彩时刻 | 2020记者看法院

最高人民法院