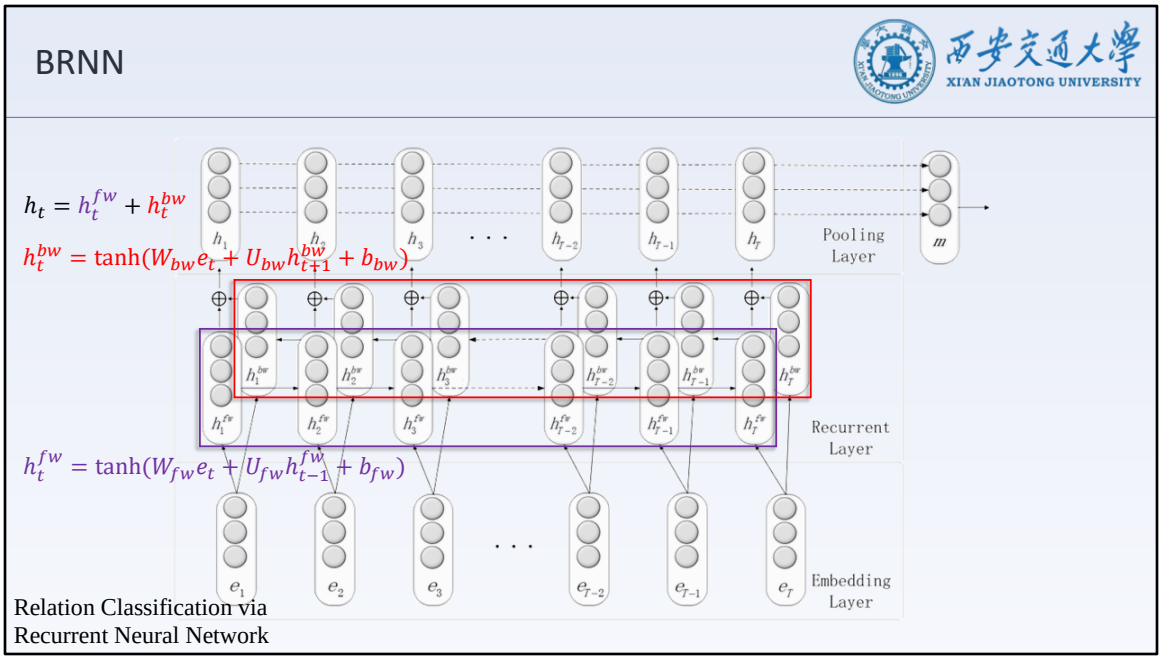
西安交通大学
XIAN JIAOTONG UNIVERSITY

工作汇报

刘卓

2022.05.13



双向循环神经网络
后文介绍的模型是以这类为主体的
实现的文章大都是针对文本分类任务进行的，它们基本上都是通过双向网络来提取输入文字的特征，然后跟一个全连接层进行分类
比如说15年的双向RNN
它在词嵌入之后，搭了两个RNN层，一个沿着输入句子正向走，一个沿着输入句子逆向走，公式就是这里对应颜色的
然后简单把对应时间步的隐藏状态相加，得到最终的隐藏状态(原论文中的图我感觉有点儿问题，魔改了下)
原文中说的隐藏状态相加，就是对应元素直接相加，还可以相乘、求平均等，但代码里默认的模式是拼接
提取出两个方向的特征并综合之后，进行最大池化，然后再通过MLP分类
这里用最大池化，文章中说是要体现出句子里关键词对于分类的作用，是针对它的任务设置的，作者对比了结果，这样相比直接把各状态累加会好很多
但实际上我猜可能结合attention会更好一些，后边有一篇用了attention的双向LSTM分类文章，它就是使用attention代替了max pooling

BRNN

西安交通大学 XIAN JIAOTONG UNIVERSITY

双向原因：

A potential problem of the one-directional forward RNN is that the information of future words are not fully utilized when predicting the semantic meaning in the middle of a sentence.

至于为什么要用双向的，左边这是原文中的话，它说在预测句子中间的语义时，未来单词的信息没有得到充分利用

我的理解：无论BRNN，BLSTM，还是ELMo，GPT，BERT，它们这些核心网络的目的都是要对词向量进行改造，把它调整成针对特定任务的词向量

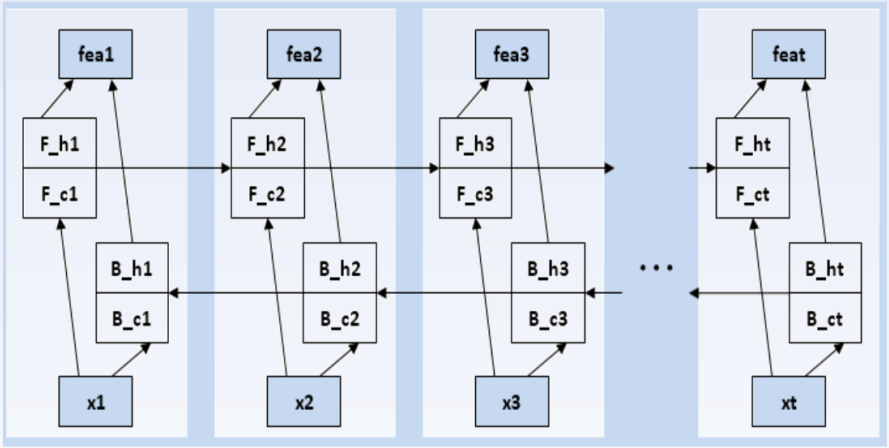
按常规想法，词义肯定是要放到上下文中进行理解的，也就必须结合前后文的信息，而在单向RNN中，在RNN单元产生信息时，没有考虑到之后的信息，这样提取出的特征是不全面的，于是就诞生了双向融合上下文信息的想法

之前说的BRNN是通过堆叠2个RNN实现双向的，之后的BLSTM、ELMo都是基于这种思想

而BERT的双向则是通过对mask掩码的优化实现的

BLSTM

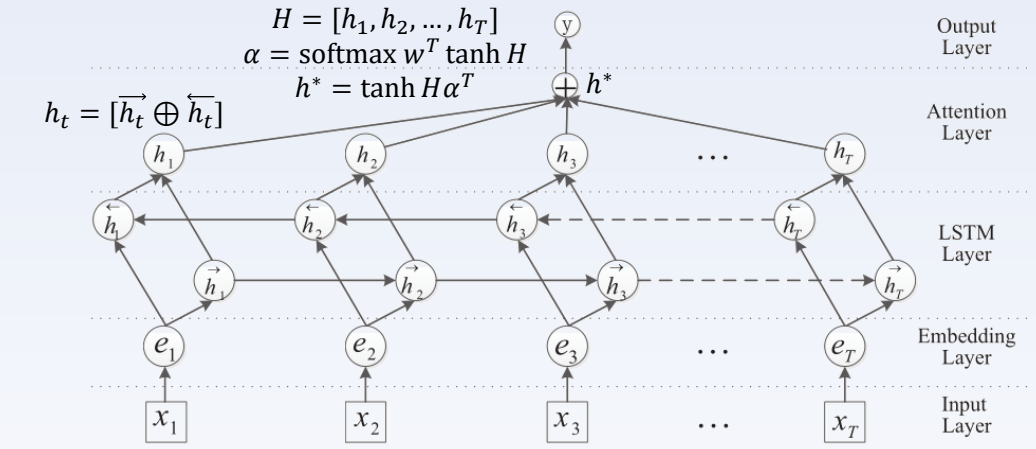
西安交通大學 XIAN JIAOTONG UNIVERSITY



Bidirectional Long Short-Term Memory Networks for Relation Classification

双向LSTM结构基本上和双向RNN是一样的，就是把RNN换成了LSTM

BLSTM + Attention



Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification

这是一篇引入注意力的，它和刚才的BLSTM不同之处就是用注意力代替了最大池化，就从注意力层开始看

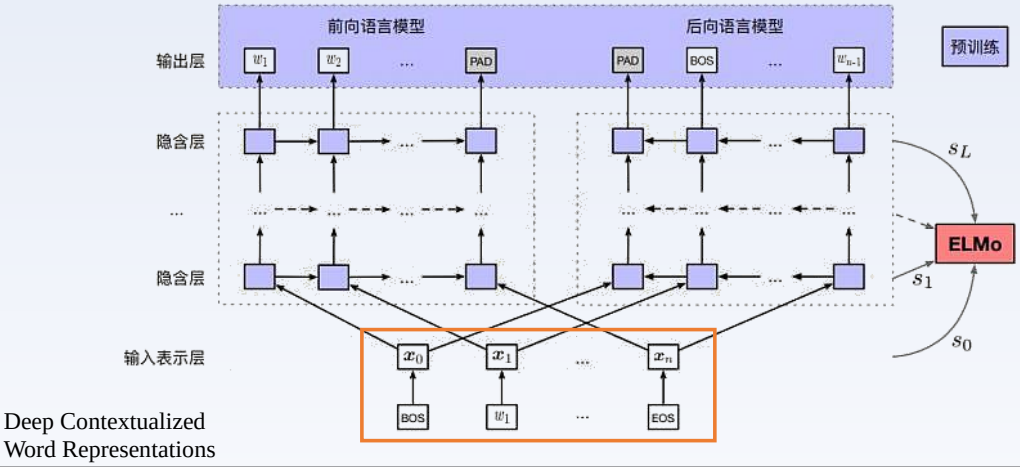
原文中是通过直接对前后向LSTM的隐藏状态相加得到注意力层输入的，源代码默认的也是add模式，实际也可以用拼接模式

而它文中所说的注意力，是按这里的公式定义的，看上去稍微有点儿自注意力的意思，也是用自身去算自己的权重

ELMo

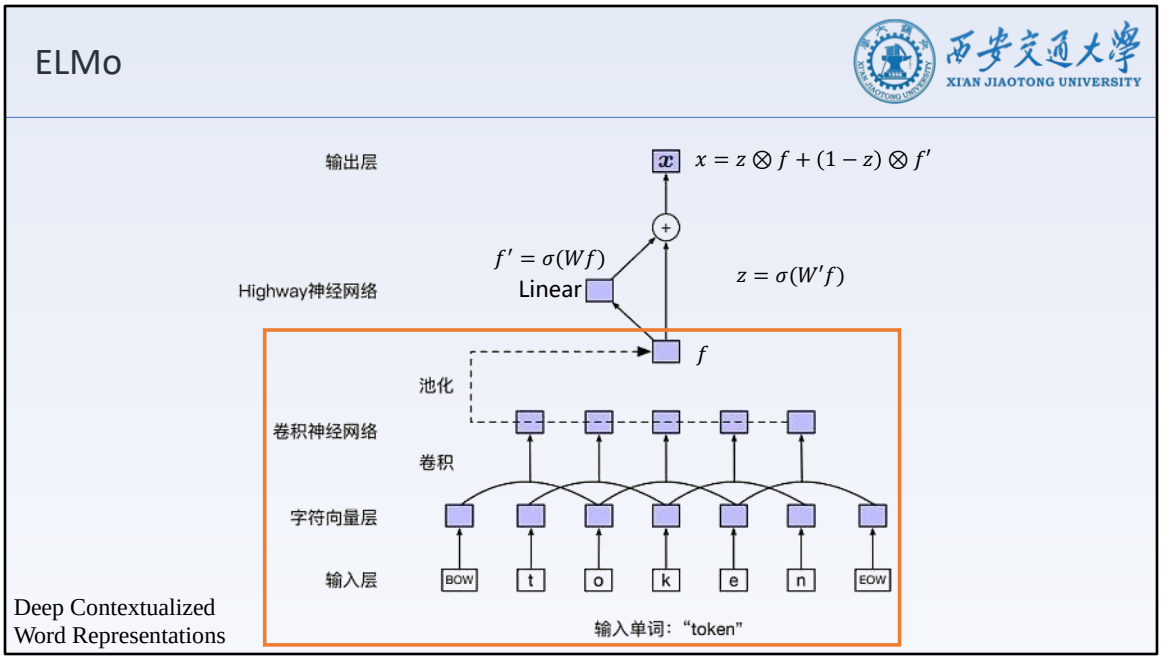
西安交通大學 XIAN JIAOTONG UNIVERSITY

Embeddings from Language Models



Deep Contextualized
Word Representations

ELMo，它和GPT、BERT都是一个类型，作为预训练模型，微调之后提取输入文本特征，承接不同的下游任务
这类模型也可以看作一种词向量表示，起到的作用和word2vec+特征提取网络的作用是一样的
先前所用的词嵌入，比如说word2vec、glove这种，都是每一个词对应一个词向量，用的时候就去查表
但如果这个词是个多义词，需要根据上下文来判断语义，那这种方式肯定就不大合适了，比如说bank可以是河岸，也可以是银行，这两个对应的词向量理论上不能是一样的
整体上来说，ELMo分为两个阶段，首先还是要训练词向量，这部分通过双向LSTM来捕捉一个词在上下文之间的关系，得到了多组词向量的表示
然后是根据下游任务的训练数据，动态调整一组权重，用这个权重对这多组词向量进行加权，得到最终的词向量表示



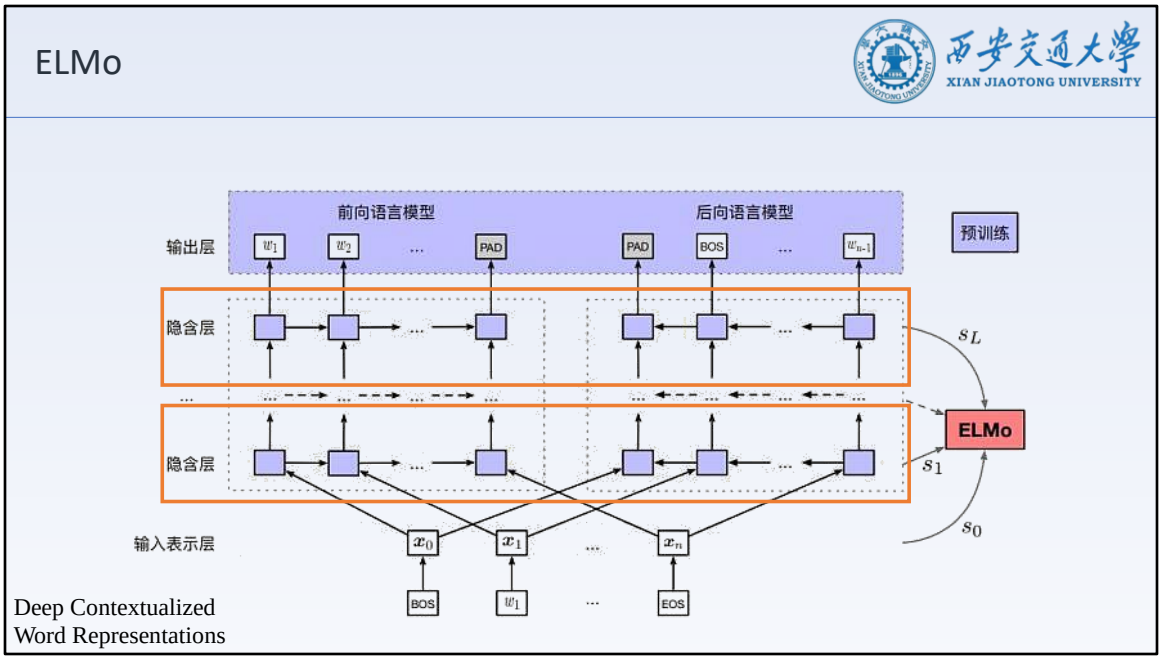
一阶段的词嵌入不是简单的word2vec这种查表，而是参考自Exploring the Limits of Language Modeling与Character-Aware Neural Language Models这两篇论文的字符级卷积神经网络来表示的

把输入的单词拆成字符，字符映射到随机初始化的一组字符向量，对这些字符对应的向量进行一维卷积与池化

之后再经过一个Highway神经网络，这个东西类似于残差，同样是为了避免梯度出问题，不过多了一个门控

最终由多个字符向量得出了一个词向量

通过这种方式构造的词向量，很明显就和上下文完全无关了，而是和词形有很大关系，就比如说book和books，它们的词向量就会比较接近，所以这种词向量能够很好地体现单词维度的特征



然后就是双向LSTM层，如果我们称之前介绍过的双向LSTM称作1个BLSTM层，那ELMo就是由多个BLSTM层堆叠成的，论文里用的是2个
图中左边的是前向网络，右边的是后向网络
之前也说了，这个网络设立的整体目标是已知前边n-1个词，让你用双向LSTM去推测第n个词，就比如说输入一个句子<BOS>I have a dream<EOS>，那前向网络最终输出的目标就是I have a dream<EOS><PAD>，也就是说我们期望第1个时间步的前向LSTM单元输出的是I，第2个输出的是have，以此类推，而后向对应的目标句就是<PAD><BOS>I have a dream，然后交叉熵算损失反向传播

ELMo

对第k个token：

				softmax				
2 BLSTM	语义级	$h_{k,2}^{LM} = \{\overrightarrow{h}_{k,2}^{LM}, \overleftarrow{h}_{k,2}^{LM}\}$	×	s_2				
1 BLSTM	句法级	$h_{k,1}^{LM} = \{\overrightarrow{h}_{k,1}^{LM}, \overleftarrow{h}_{k,1}^{LM}\}$	×	s_1	+	× γ	$ELMo_k$	
Embedding	单词级	$h_{k,0}^{LM} = \{e_k, e_k\}$	×	s_0	+			

Deep Contextualized
Word Representations

以上是一阶段，而二阶段接入具体下游任务的时候，作者认为，比较底层的BLSTM能提取出输入句子的句法信息，比较高层的BLSTM则对应的是语义信息
作者把这一阶段训练的词向量定义为单词级、句法级、语义级这三个层级信息的结合，以文章中2层BLSTM，即L=2为例
单词级就是嵌入层的词向量，句法级就是1层BLSTM，语义级就是2层BLSTM，把下游任务的句子输入到已经预训练好的ELMo网络里，得到这3个层级的输出，用一组可训练的权重对它们进行加权求和，再乘以一个缩放系数γ，就能得到这个词在具体上下文中的表示，而这个上下文环境正是通过双向LSTM层实现的