# 理论分析

## 提出原因

传统的Embedding方法只能对每个词求出一个独立于上下文的向量表示，比如Word Embedding，这种方法无法解决多义词问题。

比如单词“bank”，当上下文中存在河流相关词时，它代表着河岸；当上下文与钱相关时，它代表着银行。

在不同的上下文中，多义词有着不同的语义，使用一个固定的向量表示显然不合适。

## 思想

1. 使用Word Embedding加上Contextual Embedding(上下文编码)作为词的向量表示
2. 每个词的Contextual Embedding都会根据整个句子进行计算，是一个动态的编码
3. 将词从静态表示(Word Embedding)变为了动态表示(Word Embedding+Contextual Embedding)

## 模型与算法

在NLP领域，一般都会先使用语言模型LM和大量的语义资料进行预训练，以得到词的向量表示。ELMO也一样，不过它使用了3个LM，并整合3个层次的Embedding向量作为词的最终的向量表示R。

### 3个语言模型

ELMO中使用了三个语言模型：Traditional LM、Forward LM、Backward LM

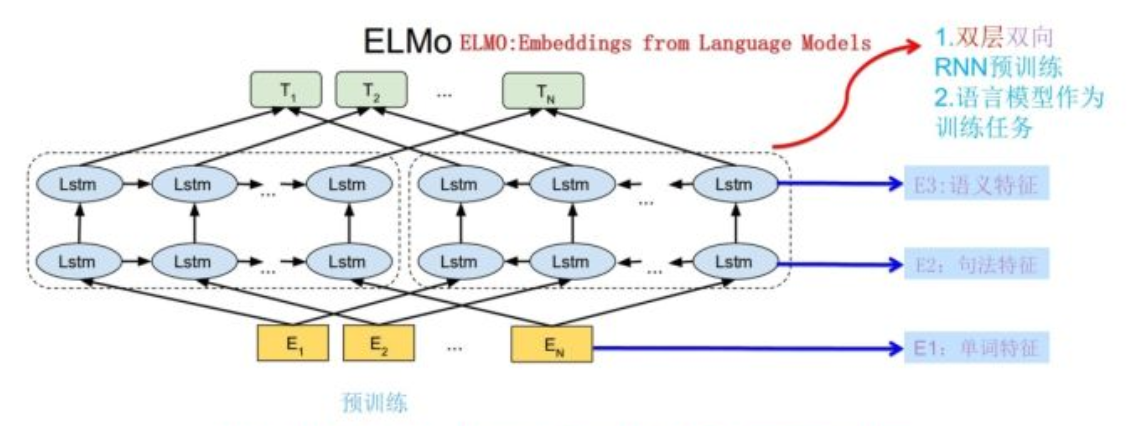


图 1.1ELMO的预训练模型

1. TLM就是传统的语言模型，为了先求得词的Word Embedding，用表示，就是上图中最下面的黄色向量。
2. FLM是一个多层前向LSTM网络，就是图1.1左侧的双层LSTM，用于从一个句子中的前k-1个词预测第k个词，公式如下：
3. BLM是一个多层后向LSTM网络，就是图1.1右侧的双层LSTM，用于从一个句子中的后n-k个词预测第k个词，公式如下：
4. FLM和BLM在本文中都是两层的，一起组成了BiLSTM
5. 每一层BiLSTM都会产生一个，一起用于计算词的最终向量表示。

### 计算公式

RNN是循环神经元，每次循环时都会产生一个中间变量，一般使用最后一个作为这个循环神经元的输出。

FLM有两层(也就是两个循环神经元)，第k个词在第一层的最终输出记为，第二层的最终输出记为。

BLM也一样，两层的输出分别为和

ELMO计算最终向量表示时会考虑到每一层BiLSTM的输出，如下：

其中：

1. 是词的Word Embedding
2. ，即通过融合FLM和BLM第一层的输出得到
3. ，即通过融合FLM和BLM第二层的输出得到
4. 如果FLM和BLM的层数更多，还会继续增加

论文中原始的表达公式为：

其中：

1. L为BiLSTM的层数
2. 是词的向量表示，猜测使用的是Word Embedding，也可能是one-hot，暂不确定

## 存在问题

1. LSTM网络提取特征能力较弱，不如Transformer
2. 双向特征拼接的方式不如一体化计算特征效果好

改进版本：BERT