**评估比较中文命名实体识别**

计算机科学与技术 计硕2004 贾星薪 2001752

# 概念

## 命名实体识别

命名实体识别（NER）是在自然语言处理中的一个经典问题，其应用也极为广泛。传统的公认比较好的处理算法是条件随机场（CRF），它是一种判别式概率模型，是随机场的一种，常用于标注或分析序列资料，如自然语言文字或是生物序列。简单是说在NER中应用是，给定一系列的特征去预测每个词的标签。

要对特征进行提取，并且希望提取的特征是有效，如何对特征的数量进行控制，这都是进行实体识别时要面对的问题，关于特征提取大都是根据经验得出的，而采用深度学习的方法可以免去人工提取特征的问题。

## Bi-LSTM

长短期记忆网络（[LSTM](https://baike.baidu.com/item/LSTM/17541102" \t "_blank)，Long Short-Term Memory）是一种时间循环神经网络，是为了解决一般的[RNN](https://baike.baidu.com/item/RNN/5707183" \t "_blank)（[循环神经网络](https://baike.baidu.com/item/%E5%BE%AA%E7%8E%AF%E7%A5%9E%E7%BB%8F%E7%BD%91%E7%BB%9C/23199490)）存在的长期依赖问题而专门设计出来的，所有的RNN都具有一种重复神经网络模块的链式形式。，在理论上，RNN绝对可以处理这样的长期依赖问题。人们可以仔细挑选参数来解决这类问题中的最初级形式，但在实践中，RNN却不能够成功学习到这些知识。因此，LSTM就是为了解决长期依赖问题而生的，LSTM通过刻意的设计来避免长期依赖问题。LSTM中一个重要的概念就是“门”，它控制信息通过的量，实质上就是一个σ函数。

前向的LSTM与后向的LSTM结合成BiLSTM，BLSTM 可以获取到整个序列的信息，在命名实体识别任务中可以充分利用输入序列的上下文信息，从而可以更加准确的判断一个词语是否为一个实体。在处理序列数据时，BLSTM 比一般的 LSTM增加了一个反向计算的过程，此过程可以利用到序列的下文信息，最后把正向和反向两个方向计算的值同时输出到输出层，这样通过两个方向获取到了一个序列的全部信息。

## 条件随机场(CRF)

条件随机场(CRF)是给定一组输入序列条件下另一组输出序列的条件概率分布模型，在自然语言处理中得到了广泛应用。对于命名实体识别这一类的序列标注问题，通常采用的是线性条件随机场（linear-CRF）。

# 实现

采用pytorch实现双向循环神经网络（Bi-RNN）和条件随机场（CRF）在NER上的应用。把数据放入神经网络中进行运行后输出我们想要的结果，其本质都是通过神经网络进行学习提取特征，根据特征获得结果。

简单的LSTM的优点是能够通过双向的设置学习到观测序列（输入的字）之间的依赖，在训练过程中，LSTM能够根据目标（比如识别实体）自动提取观测序列的特征，但是缺点是无法学习到状态序列（输出的标注）之间的关系，要知道，在命名实体识别任务中，标注之间是有一定的关系的，比如B类标注（表示某实体的开头）后面不会再接一个B类标注，所以LSTM在解决NER这类序列标注任务时，虽然可以省去很繁杂的特征工程，但是也存在无法学习到标注上下文的缺点。

相反，CRF的优点就是能对隐含状态建模，学习状态序列的特点，但它的缺点是需要手动提取序列特征。所以一般的做法是，在LSTM后面再加一层CRF，以获得两者的优点。

因此采用Bi-LSTM加一层CRF的代码实现.

## 数据获得

数据集可以采用自定义标注的数据或者采用公开数据集。数据集用的是论文ACL 2018[Chinese NER using Lattice LSTM](https://github.com/jiesutd/LatticeLSTM)中收集的简历数据，数据的格式如下，它的每一行由一个字及其对应的标注组成，标注集采用BIOES（B表示实体开头，E表示实体结尾，I表示在实体内部，O表示非实体），句子之间用一个空行隔开。该数据集就位于项目目录下的ResumeNER文件夹里。

## **将词转换成神经网络的输入数**据

在神经网络中只能使用数字作为输入，因此我们需要将分词转化为词向量，通常采用gensim word2vec、glove等工具进行。采用庞大的数据集进行训练，获得高效可使用的训练模型。这样就可以解决面对未出现过的词也能有较好的识别转换效果。或者采用谷歌训练好的词向量，其中包含了上千万个原料库。

## LSTM单元编码

获取该句子的向量后，便将其放入LSTM的的输入层（论文中也多称为input layer或者embedding layer），每个输入神经元对应一个字的词向量，正向传播则从第一个字“马”开始，随着时间推移一直到“巴”。  
  每个时刻t对于的字xt​通过前向传播和后向传播并拼接得到ht​，其次得到y^​t​，该值即为当前时刻t对应的7个标签中每个标签预测的概率。例如对于“马”字， y^​t​=[0.031,0.305,0.219,0.015,0.129,0.133,0.168] ，最大的值为0.305，对应于下标1，即标签“B-PER”。

## CRF解码

HMM模型中存在两个假设，一是输出观察值之间严格独立，二是状态转移过程中当前状态只与前一状态有关。也就是说，在命名实体识别的场景下，HMM认为观测到的句子中的每个字都是相互独立的，而且当前时刻的标注只与前一时刻的标注相关。但实际上，命名实体识别往往需要更多的特征，比如词性，词的上下文等等，同时当前时刻的标注应该与前一时刻以及后一时刻的标注都相关联。由于这两个假设的存在，显然HMM模型在解决命名实体识别的问题上是存在缺陷的。条件随机场通过引入自定义的特征函数，不仅可以表达观测之间的依赖，还可表示当前观测与前后多个状态之间的复杂依赖，可以有效克服HMM模型面临的问题。

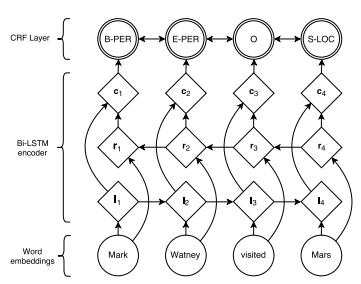
在CRF中要解决的问题之一是解码问题，对于y^​t​的结果不一定完全符合输出规则，因此需要将其按照输出规则进行解码。输出规则则体现在CRF中的超参数和参数。例如对于t=5时刻，字为“州”，对应的y^​t​=[0.085,0.113,0.153,0.220,0.207,0.108,0.114]，可知最大的值对应下标表示的标签为“B-LOC”，虽然成功的预测了其属于地区这一类实体，但很显然应该是“I-LOC”。因此将该输出概率向量做下列计算：

P(yt​∣y^​t​)=Z(x)1​exp(λk​tk​(y^​t−1​,yt​,x,t)+μl​sl​(yt​,x,t))

然后对其他词按照该式子进行计算，通过维特比算法求出最大值，即对应的序列中，“州”字的概率向量可能变为：

y^t=[0.085,0.113,0.153,0.207,0.220,0.108,0.114]。

CRF模型通过对输出标签二元组进行建模，使用动态规划算法找出得分最高的路径作为最优路径y^\*进行序列标注。整体基于字符的BiSLTM-CRF中文命名实体识别模型框架如图 :



# 使用pytorch实现命名实体识别

使用pytorch实现命名实体识别主要分为三个部分，Main.py主要解决控制台参数和数据获取，以及模型调用;数据处理,包括对数据编号，获取词向量以及标注信息等操作;字符串处理，对已生成的标注序列进行实体提取；Bi\_lstm\_crf.py文件，编写的神经网络模型，主要完成训练任务。

这里想主要解释一些该模型的python代码含义：在该类中首先进行初始化，传入几个重要的参数，例如batch\_size：批大小，一次训练的样本数量；epoch：使用全部训练样本训练的次数；embeddinds:词嵌入矩阵（字数量\*特征数量；hidden\_dim：隐藏维度。完成初始化后，进入train函数，将传入的数据集按长度进行排序，之后进行训练并计算损失，总是保存性能最好的那个模型。test(self, word\_lists, tag\_lists, word2id, tag2id)：返回最佳模型在测试集上的预测结果。

在models文件夹中，util.py编写了程序运行过程中对于词处理的工具类，word2features实现抽取单个字的特征；sent2features抽取序列特征；tensorized(batch, maps)：传入训练的样本以及映射信息，其中利用torch.ones将信息转换为张量，并返回张量信息以及batch中各个元素的长度；sort\_by\_lengths(word\_lists, tag\_lists)：对训练数据按照长度进行排序；def cal\_lstm\_crf\_loss(crf\_scores, targets, tag2id)：计算BiLSTM-CRF的模型的损失；indexed(targets, tagset\_size, start\_id):将targets中的数转化为在[T\*T]大小序列中的索引,T是标注的种类。

Bilstm\_crf.py文件中主要包括：init初始化函数；train对数据集进行排序操作以及调用train\_step函数进行模型的训练，在每轮训练完成后记录其loss值，与上一轮训练进行比较，总是保留在验证集上性能最好的那一个，来获得最优的模型，验证性能时调用validate函数；train\_step主要写了训练的具体过程，准备数据时调工具类中写好的tensorized(batch, maps)，调用model函数进行训练，并返回评估分数，计算loss更新损失值；在test函数中因为要采用crf算法，获取到forward函数的返回值作为crf的得分，根据维特比算法进行解码，分析各个标记的分数，最后返回解码结果。

整个项目中不仅实现了使用Bi-LSTM+CRF来解决中文命名实体识别问题，还分别实现了采用CRF以及Bi-LSTM来分别单独解决中文实体识别的方法，并对这三种方法进行评估比较，比较其召回率、准确率以及F1分数。发现两者结合之后的模型效果更好。

# 4.总结

本代码只是基于pytorch下实现的Bi-LSTM，CRF以及Bi-LSTM+CRF的中文命名实体识别，还分别分析比较了Bi-LSTM情况下的命名实体识别效率并进行了评估。原理并不是很复杂，重在看代码的过程对自然语言翻译这个领域进行接触和学习，对本堂课所涉及知识有更深入的理解，在实践中体会深度学习在这个领域的强大功能，在效率的提升方面有着较好的表现。