序列标注

序列标注问题包括自然语言处理中的分词，词性标注，命名实体识别，关键词抽取，词义角色标注等等。我们只要在做序列标注时给定特定的标签集合，就可以进行序列标注。

　　序列标注问题是NLP中最常见的问题，因为绝大多数NLP问题都可以转化为序列标注问题，虽然很多NLP任务看上去大不相同，但是如果转化为序列标注问题后其实面临的都是同一个问题。所谓“序列标注”，就是说对于一个一维线性输入序列：

https://images2018.cnblogs.com/blog/1335117/201807/1335117-20180725205710497-99721212.png

　　给线性序列中的每个元素打上标签集合中的某个标签：

https://images2018.cnblogs.com/blog/1335117/201807/1335117-20180725205744384-394040534.png

所以，其本质上是对线性序列中每个元素根据上下文内容进行分类的问题。一般情况下，对于NLP任务来说，线性序列就是输入的文本，往往可以把一个汉字看做线性序列的一个元素，而不同任务其标签集合代表的含义可能不太相同，但是相同的问题都是：如何根据汉字的上下文给汉字打上一个合适的标签（无论是分词，还是词性标注，或者是命名实体识别，道理都是想通的）。

**序列标注问题之中文分词**

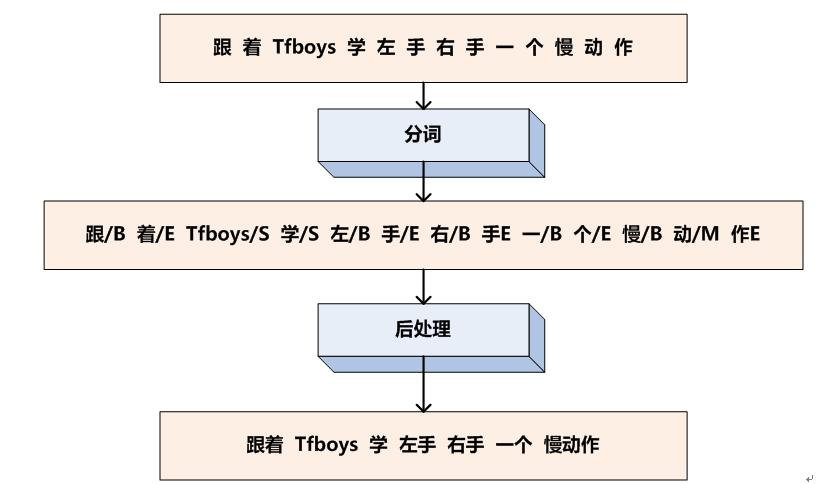
　　以中文分词任务来说明序列标注的过程。假设现在输入句子“跟着TFboys学左手右手一个慢动作”，我们的任务是正确地把这个句子进行分词。首先，把句子看做是一系列单字组成的线性输入序列，即：

https://images2018.cnblogs.com/blog/1335117/201807/1335117-20180725210026436-92172388.png

　　序列标注的任务就是给每个汉字打上一个标签，对于分词任务来说，我们可以定义标签集合为（jieba分词中的标签集合也是这样的）：

https://images2018.cnblogs.com/blog/1335117/201807/1335117-20180725210047308-899941442.png

　　其中B代表这个汉字是词汇的开始字符，M代表这个汉字是词汇的中间字符，E代表这个汉字是词汇的结束字符，而S代表单字词。



　　有了这四个标签就可以对中文进行分词了。这时你看到了，中文分词转换为对汉字的序列标注问题，假设我们已经训练好了序列标注模型，那么分别给每个汉字打上标签集合中的某个标签，这就算是分词结束了，因为这种形式不方便人来查看，所以可以增加一个后处理步骤，把B开头，后面跟着M的汉字拼接在一起，直到碰见E标签为止，这样就等于分出了一个单词，而打上S标签的汉字就可以看做是一个单字词。于是我们的例子就通过序列标注，被分词成如下形式：

https://images2018.cnblogs.com/blog/1335117/201807/1335117-20180725210218111-1941846063.png

　　在这里我们可以采用双向LSTM来处理该类问题，双向会关注上下文的信息。

　　在NLP中最直观的处理问题的方式就是要把问题转换为序列标注问题，思考问题的思维方式也就转换为序列标注思维，这个思维很重要，决定你能否真的处理好NLP问题。

**序列标注之命名实体识别（NER）**

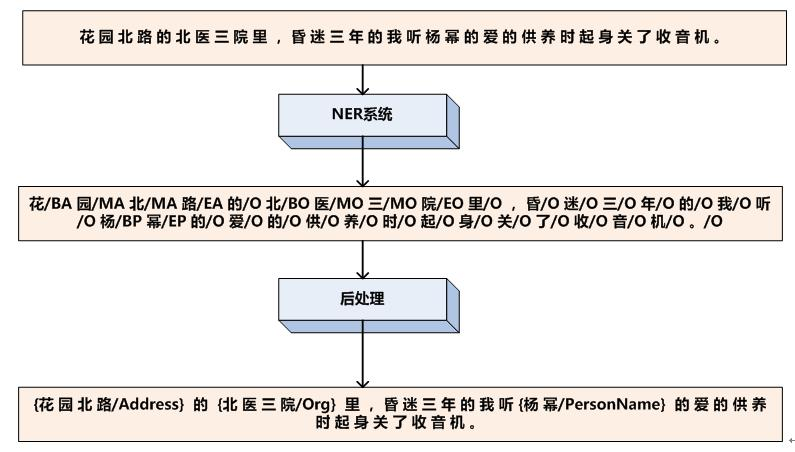
　　我们再来看看命名实体识别问题中的序列标注，命名实体识别任务是识别句子中出现的实体，通常识别人名、地名、机构名这三类实体。现在的问题是：假设输入中文句子

https://images2018.cnblogs.com/blog/1335117/201807/1335117-20180725210731375-1237953271.png

　　我们要识别出里面包含的人名、地名和机构名。如果以序列标注的角度看这个问题，我们首先得把输入序列看成一个个汉字组成的线性序列，然后我们要定义标签集合，标签集合如下（在这里的标签用什么代表不重要，重要的是它代表的含义）：

https://images2018.cnblogs.com/blog/1335117/201807/1335117-20180725210924636-717053426.png

　　其中，BA代表这个汉字是地址首字，MA代表这个汉字是地址中间字，EA代表这个汉字是地址的尾字；BO代表这个汉字是机构名的首字，MO代表这个汉字是机构名称的中间字，EO代表这个汉字是机构名的尾字；BP代表这个汉字是人名首字，MP代表这个汉字是人名中间字，EP代表这个汉字是人名尾字，而O代表这个汉字不属于命名实体。



　　有了输入汉字序列，也有了标签集合，那么剩下的问题是训练出一个序列标注ML系统，能够对每一个汉字进行分类，假设我们已经学好了这个系统，那么就给输入句子中每个汉字打上标签集合中的标签，于是命名实体就被识别出来了，为了便于人查看，增加一个后处理步骤，把人名、地名、机构名都明确标识出来即可。

　　除了上面的分词和命名实体标注，很多其他的NLP问题同样可以转换为序列标注问题，比如词性标注、CHUNK识别、句法分析、语义角色识别、关键词抽取等。

　　传统解决序列标注问题的方法包括HMM/MaxEnt/CRF等，很明显RNN很快会取代CRF的主流地位，成为解决序列标注问题的标准解决方案，那么如果使用RNN来解决各种NLP基础及应用问题，我们又该如何处理呢，下面我们就归纳一下使用RNN解决序列标注问题的一般优化思路。

　　对于分词、词性标注（POS）、命名实体识别（NER）这种前后依赖不会太远的问题，可以用RNN或者BiRNN处理就可以了。而对于具有长依赖的问题，可以使用LSTM、RLSTM、GRU等来处理。关于GRU和LSTM两者的性能差不多，不过对于样本数量较少时，有限考虑使用GRU（模型结构较LSTM更简单）。此外神经网络在训练的过程中容易过拟合，可以在训练过程中加入Dropout或者L1/L2正则来避免过拟合。

**CRF和LSTM在序列标注上的优劣**

**LSTM：**像RNN、LSTM、BILSTM这些模型，它们在序列建模上很强大，它们能够capture长远的上下文信息，此外还具备神经网络拟合非线性的能力，这些都是crf无法超越的地方，对于t时刻来说，输出层yt受到隐层ht（包含上下文信息）和输入层xt（当前的输入）的影响，但是yt和其他时刻的yt`是相互独立的，感觉像是一种point wise，对当前t时刻来说，我们希望找到一个概率最大的yt，但其他时刻的yt`对当前yt没有影响，如果yt之间存在较强的依赖关系的话（例如，形容词后面一般接名词，存在一定的约束），LSTM无法对这些约束进行建模，LSTM模型的性能将受到限制。

**CRF：**它不像LSTM等模型，能够考虑长远的上下文信息，它更多考虑的是整个句子的局部特征的线性加权组合（通过特征模版去扫描整个句子）。关键的一点是，CRF的模型为p(y | x, w)，注意这里y和x都是序列，它有点像list wise，优化的是一个序列y = (y1, y2, …, yn)，而不是某个时刻的yt，即找到一个概率最高的序列y = (y1, y2, …, yn)使得p(y1, y2, …, yn| x, w)最高，它计算的是一种联合概率，优化的是整个序列（最终目标），而不是将每个时刻的最优拼接起来，在这一点上CRF要优于LSTM。

**HMM：**CRF不管是在实践还是理论上都要优于HMM，HMM模型的参数主要是“初始的状态分布”，“状态之间的概率转移矩阵”，“状态到观测的概率转移矩阵”，这些信息在CRF中都可以有，例如：在特征模版中考虑h(y1), f(yi-1, yi), g(yi, xi)等特征。

**CRF与LSTM：**从数据规模来说，在数据规模较小时，CRF的试验效果要略优于BILSTM，当数据规模较大时，BILSTM的效果应该会超过CRF。从场景来说，如果需要识别的任务不需要太依赖长久的信息，此时RNN等模型只会增加额外的复杂度，此时可以考虑类似科大讯飞**FSMN**（一种基于窗口考虑上下文信息的“前馈”网络）。

**CNN＋BILSTM＋CRF：**这是目前学术界比较流行的做法，BILSTM＋CRF是为了结合以上两个模型的优点，CNN主要是处理英文的情况，英文单词是由更细粒度的字母组成，这些字母潜藏着一些特征（例如：前缀后缀特征），通过CNN的卷积操作提取这些特征，在中文中可能并不适用（中文单字无法分解，除非是基于分词后），这里简单举一个例子，例如词性标注场景，单词football与basketball被标为名词的概率较高， 这里后缀ball就是类似这种特征。

 参考文章：

[使用RNN解决NLP中序列标注问题的通用优化思路](https://blog.csdn.net/malefactor/article/details/50725480)

[使用深度学习进行中文自然语言处理之序列标注](https://www.jianshu.com/p/7e233ef57cb6)

[CRF和LSTM 模型在序列标注上的优劣？](https://www.zhihu.com/question/46688107/answer/117448674)