CS224n笔记[4]:自然语言中的依存分析（Dependency Parsing）

作者：郭必扬

什么是依存分析

自然语言处理任务中，有很重要的一块，就是分析语言的结构。语言的结构，一般可以有两种视角：

1. 组成关系（Constituency）
2. 依赖关系（Dependency）

前者，主要关心的是句子是怎么构成的，词怎么组成短语。所以研究Constituency，主要是研究忽略语义的“语法”结构（context-free grammars）。

后者，依赖关系，则主要关心的是句子中的每一个词，都依赖于哪个其他的词。

比如下面这个句子：

“瞧这个可爱的小傻瓜！”

- “傻瓜”，是“瞧”这个动作的对象，因此“傻瓜”是依赖于“瞧”的；
- “可爱的”、“小”都是修饰“傻瓜”的，因此，这两个形容词都是依赖于“傻瓜”的；
- “这个”，同样是指示“傻瓜”的，因此它也依赖于“傻瓜”。

这样，我们就清楚了这个句子中的所有依赖关系，画成依赖关系图则是这样：

依赖关系

注意，在图中我们增加了一个根节点“Root”，这是为了让“瞧”这个字也有依赖的对象。

当然，关系依存分析，还有很多的规则，里面比较复杂，我不太感兴趣，所以这里不多写了。

下面我们来介绍如何让机器自动地帮我们来分析句子的结构。

传统的基于转移的依存分析（Transition-based Parsing）

这里主要介绍Nivre在2003年提出的“Greedy Deterministic Transition-Based Parsing”方法，一度成为依存分析的标准方法。这里我简单地介绍一下它的工作原理。

我们构造一个三元组，分别是Stack、Buffer和一个Dependency Set。

- Stack最开始只存放一个Root节点；
- Buffer则装有我们需要解析的一个句子；
- Set中则保存我们分析出来的依赖关系，最开始是空的。

我们要做的事情，就是不断地把Buffer中的词往Stack中推，跟Stack中的词判断是否有依赖关系，有的话则输出到Set中，直到Buffer中的词全部推出，Stack中也仅剩一个Root，就分析完毕了。

下面，我通过一个十分简单的例子，来演示这个过程。 这次，我们分析的句子是：

分析的过程如下：

Stack	Buffer	Action	Dependency Set
Root	I love Wuhan	Shift	
Root I	love Wuhan	Shift	
Root I love	Wuhan	Left Arc (向左指的关系，把左边的词移除)	
Root love	Wuhan	Shift	I ← love
Root love Wuhan		Right Arc (向右指的关系，把右边的词移除)	I ← love, love → Wuhan
Root love		Right Arc (向右指的关系，把右边的词移除)	
Root			I ← love, love → Wuhan, Root → love

上面的过程怎么理解呢？ 比方从第二行，这个时候Stack中只有[Root,I]，不构成依赖关系，所以我们需要从Buffer中“进货”了，因此采取的Action是Shift（把Buffer中的首个词，移动到Stack中），于是就到了第三行。

第三行，我们的Stack变成了[Root,I,love]，其中I和Love构成了依赖关系，且是Love指向I，即“向左指”的依赖关系，因此我们将采取“Left Arc”的action，把被依赖的词（此时就是关系中的左边的词）给移除Stack，把这个关系给放入到Dependency Set中。

按照这样的方法，我们一直进行，不断地根据Stack和Buffer的情况，来从Shift、Left-arc、Right-arc三种动作中选择我们下一步应该怎么做，知道Stack中只剩一个Root，Buffer也空了，这个时候，分析就结束，我们就得到了最终的Dependency Set。

以上的过程，应该不难理解，但是相信大家此时一定会有疑问：

> 我怎么让机器去决定当前的Action呢？即机器怎么知道，Stack中是否构成了依赖关系？

在Nivre的年代，这里使用是机器学习的方法，需要做繁重的特征工程。这里的特征，往往有10^6 10^7个二值特征，即无数个指示条件作为特征，来训练模型，可以想象这么高纬度的

特征是十分稀疏的。因此，这种模型的95%左右的解析时间，都花费在计算特征上。这也是传统方法的最要问题。

神经依存分析（Neural Dependency Parsing）

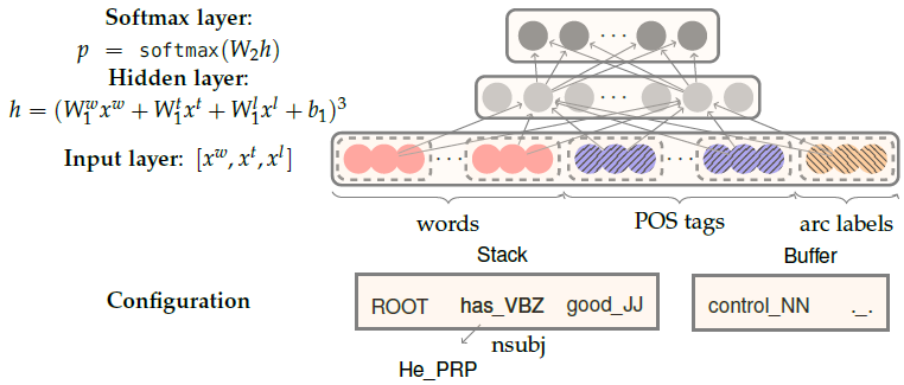
神经依存分析方法，是斯坦福团队2014年的研究成果，主要就是利用了神经网络的方法代替了传统机器学习方法、用低维分布式表示来代替传统方法的复杂的高维稀疏特征表示。而整个解析的过程，依然是根据之前的Transition-based方法。

首先明确，我们的预测任务，是**「根据当前的状态，即Stack、Buffer、Set的当前状态，来构建特征，然后预测出下一步的动作」**。

在神经依存分析中，我们的特征是怎么构建的呢？ 我们可以利用的信息包括词（word）、词性（pos tag）和依赖关系的标签（label）。我们对这三者，都进行低维分布式表示，即通过Embedding的方法，把离散的word、label、tag都转化成低维向量表示。

对于一个状态，我们可以选取stack、Buffer、set中的某些词和关系，构成一个集合，然后把他们所有的embedding向量都拼接起来，这样就构成了该状态的特征表示。

至于选择哪些词、关系，这个就是一个**「经验性」**的东西了，在斯坦福的论文中可以详细了解。 整个模型的网络结构也十分简洁：



A Fast and Accurate Dependency Parser using Neural Networks

对于Dependency Parsing的简单介绍就到此为止。依存分析，并不是我们NLP中最常见的任务之一，我们也很少看到直接将依存分析做应用的，我们更常见的是分类、实体识别、阅读理解、对话等任务。但是依存分析，作为自然语言处理的一项基础技术，试图让机器去理解语言的内部结构，理解了结构，NLU（Natural Language Understanding）才成为可能。

cs224n的Assignment3就是Neural Dependency Parsing的实现，代码见github：

https://github.com/beyondguo/CS224n-notes-and-codes/tree/master/assignment3