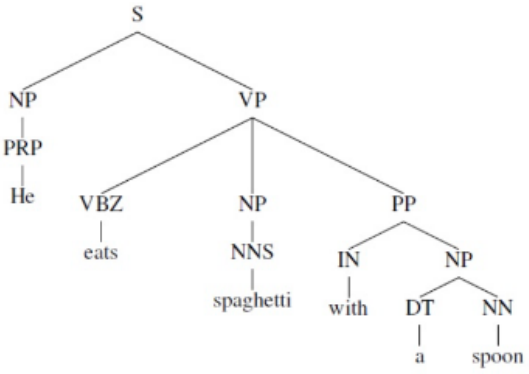
Lecture18 Constituency Parsing and Tree Recursive Neural Networks

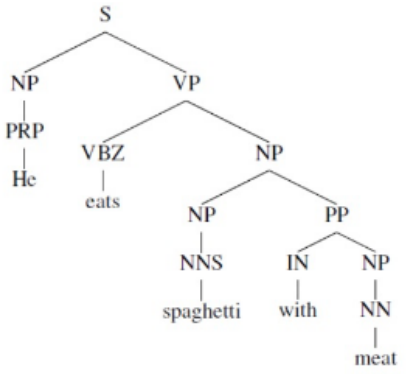
---

- The **snowboarder** is leaping over a mogul
- **A person on a snowboard** jumps into the air

“snowboarder”在语义上相当于“A person on a snowboard”，但字长不同。人类如何理解其中的含义？可能唯一的答案是组合原则，通过组件的含义将它们组合成更大的组件。

语言结构





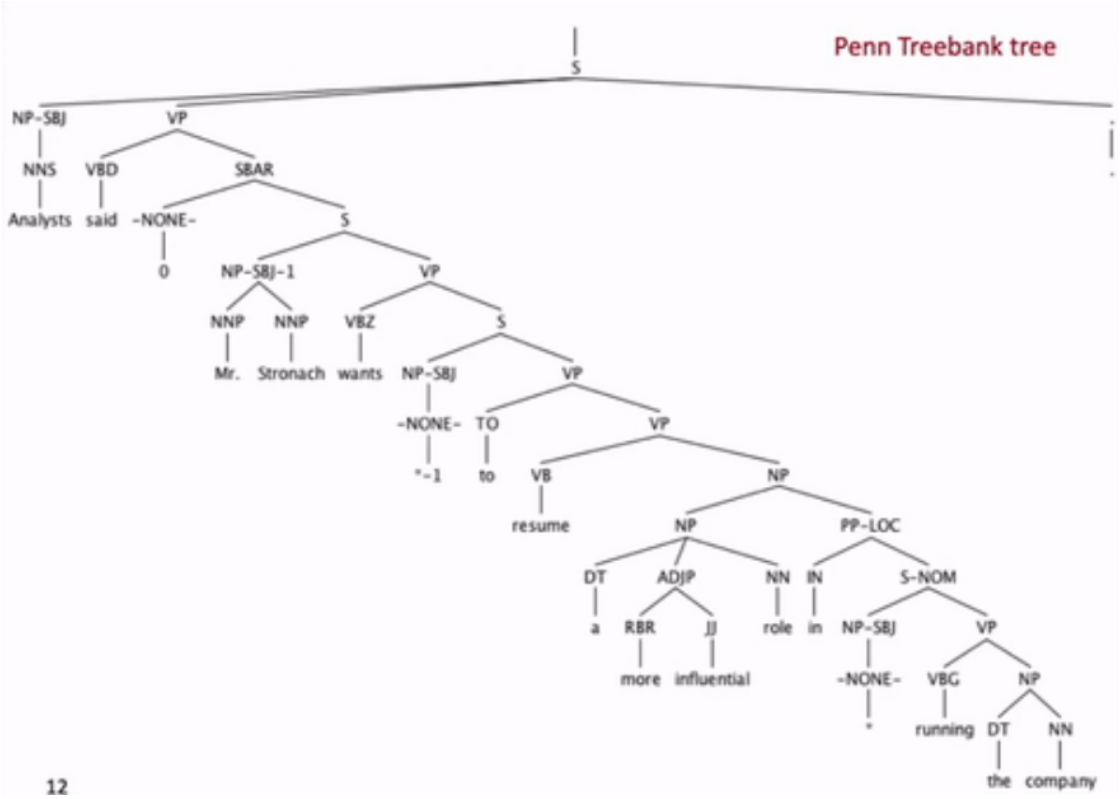
递归地描述语言

[The person standing next to [the man from [the company that purchased [the firm that you used to work at]]]]

整体是一个大名词短语，但里面有一个名词短语[the man from [the company that purchased [the firm that you used to work at]]]，其又是另一个大名词短语，其中又有晓得名词短语[the company that purchased [the firm that you used to work at]]……最内部的小名词短语[you]。因此个体名词也是名词短语

但是说语言是递归的，如果加入一些限制，在某种意义上不是真正的递归，因为他不会无穷大。

Constituency Parsing 或称 短语结构语法

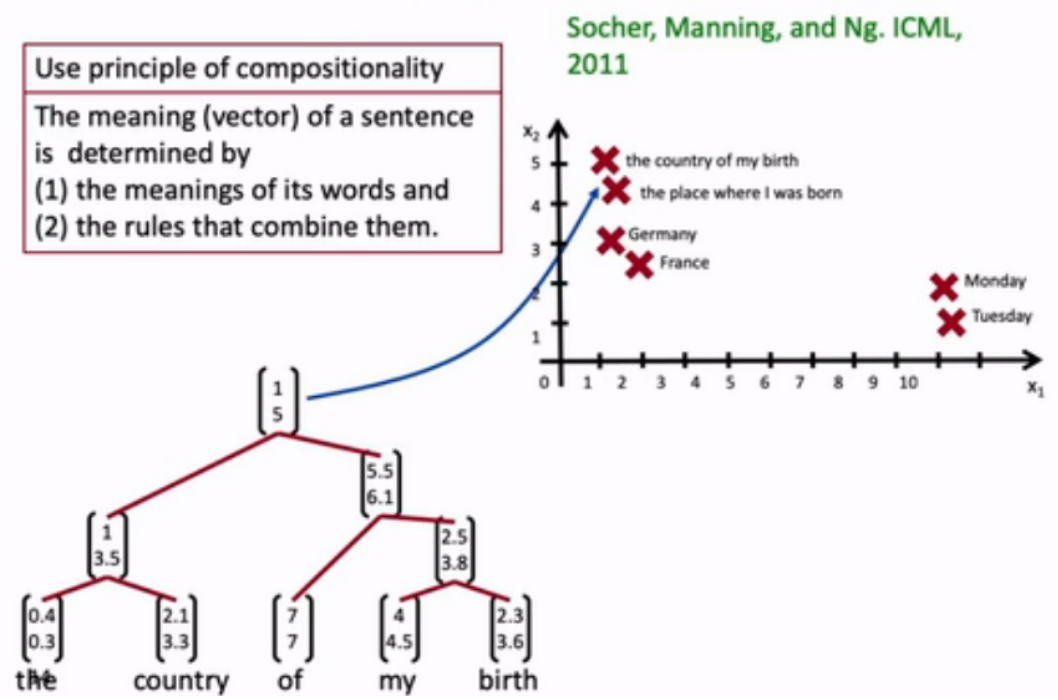


这是一个原始的Penn Treebank tree，包含短语结构语法和额外的注释。

那么我们想要作些什么来解决成分的语义相似性，不仅需要单词向量空间，还希望能够采取更大的成分比如名词短语并给使他们具有意义。就需要有一种方法以组合的方式计算任何短语的含义，将这些短语映射到相同的向量空间

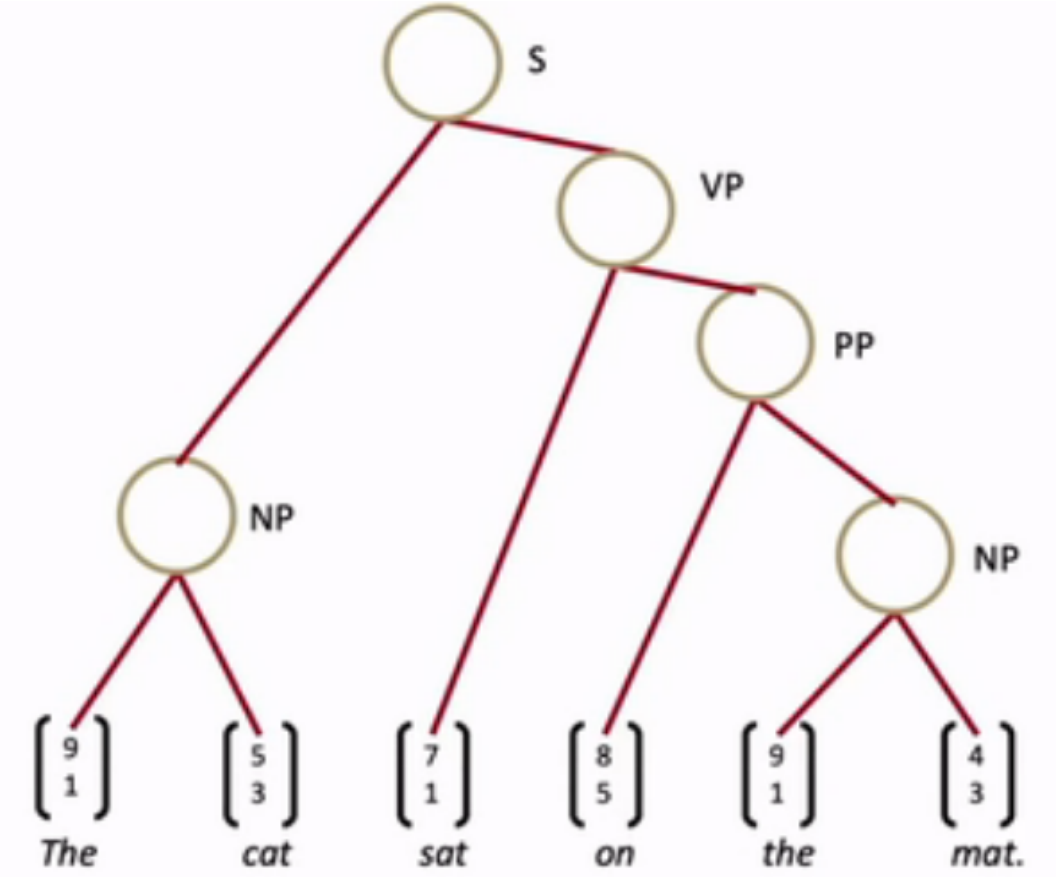
怎样将短语映射到向量空间？

1. 组合原则。如果想要理解短语的含义，①要通过了解其词义来②制定结合这些含义的规则

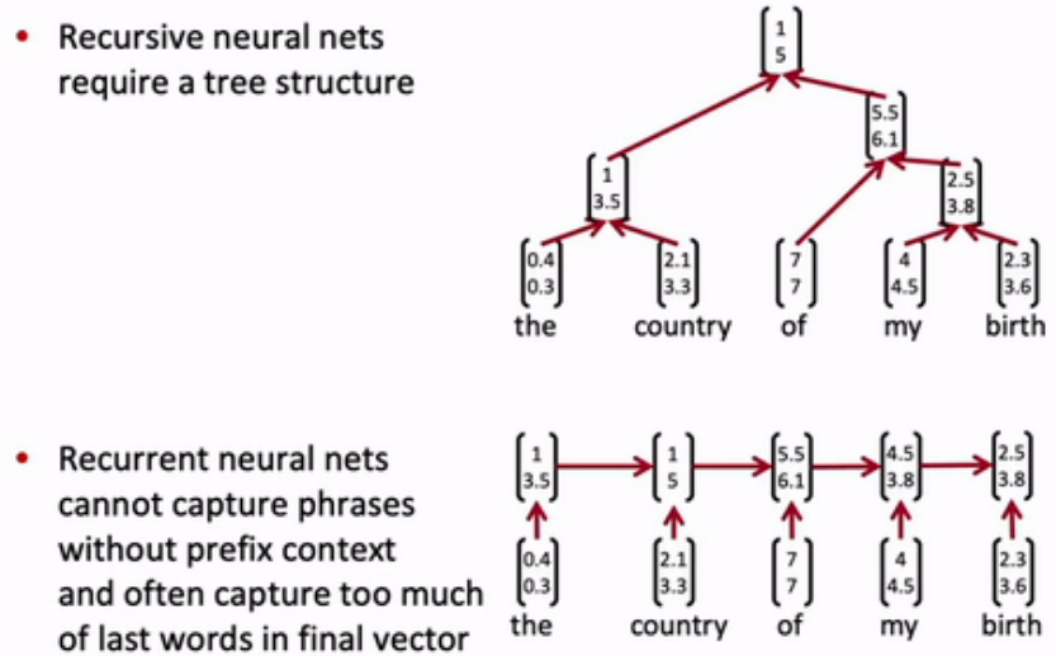


怎样建立模型来做到？

首先有了单词向量，加上已经解析了句子的正确结构，然后要需要计算含义来弄清楚句子的意思



递归语言结构 VS RNNs

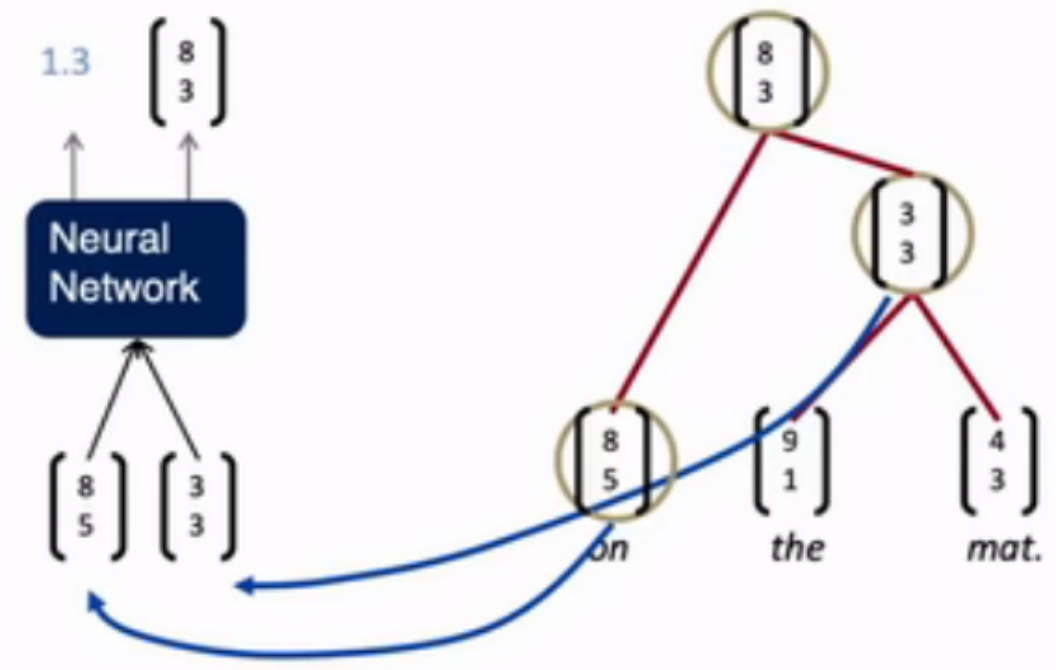


- 树递归神经网络需要一个具有树形结构的句子或者任何类型的短语，我们就能知道它的组成部分是什么，然后为其中对句法结构敏感的短语计算出含义表示
- RNN。只需要运行序列模型，只对整个序列有一个含义表示

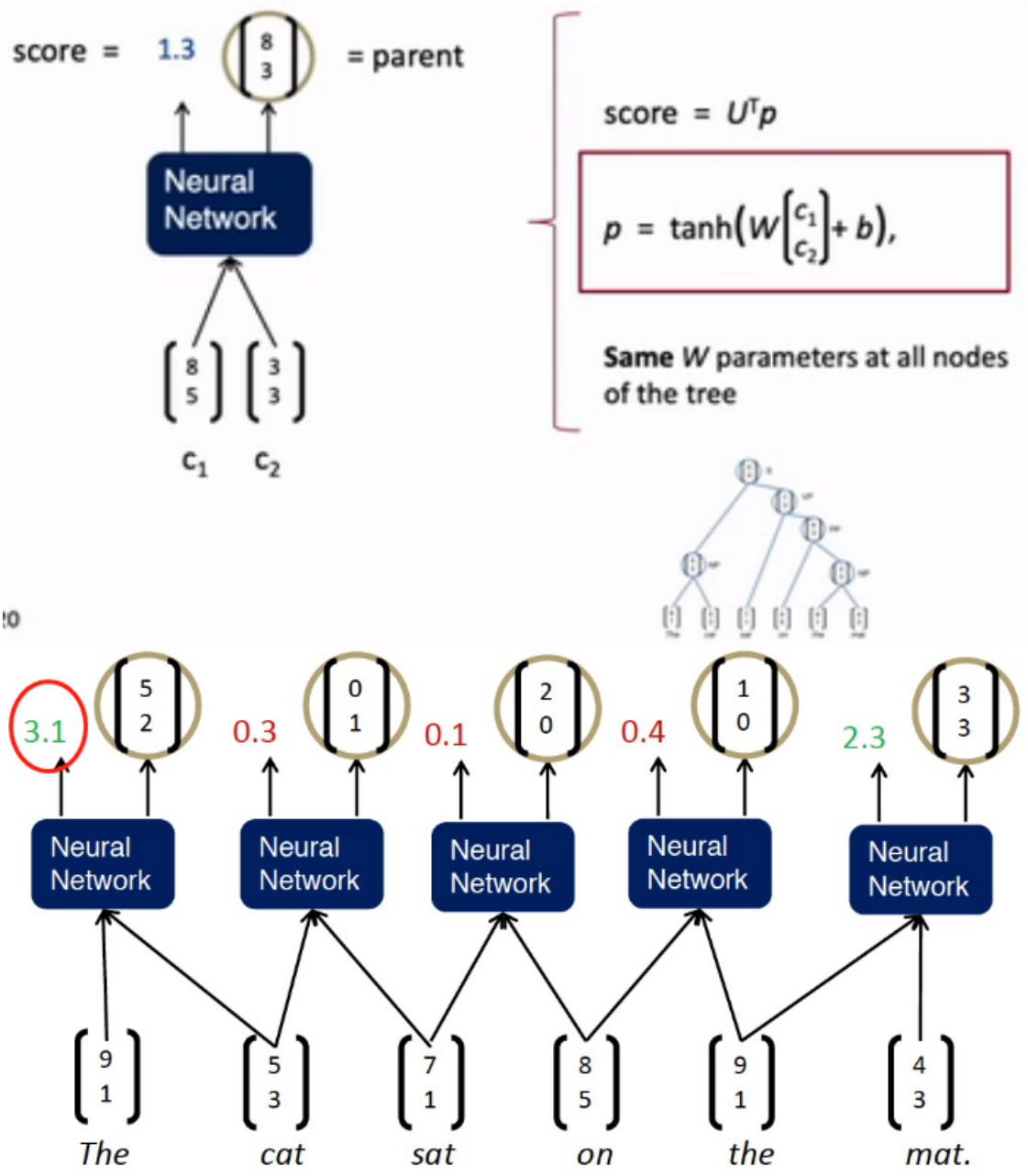
HOW?

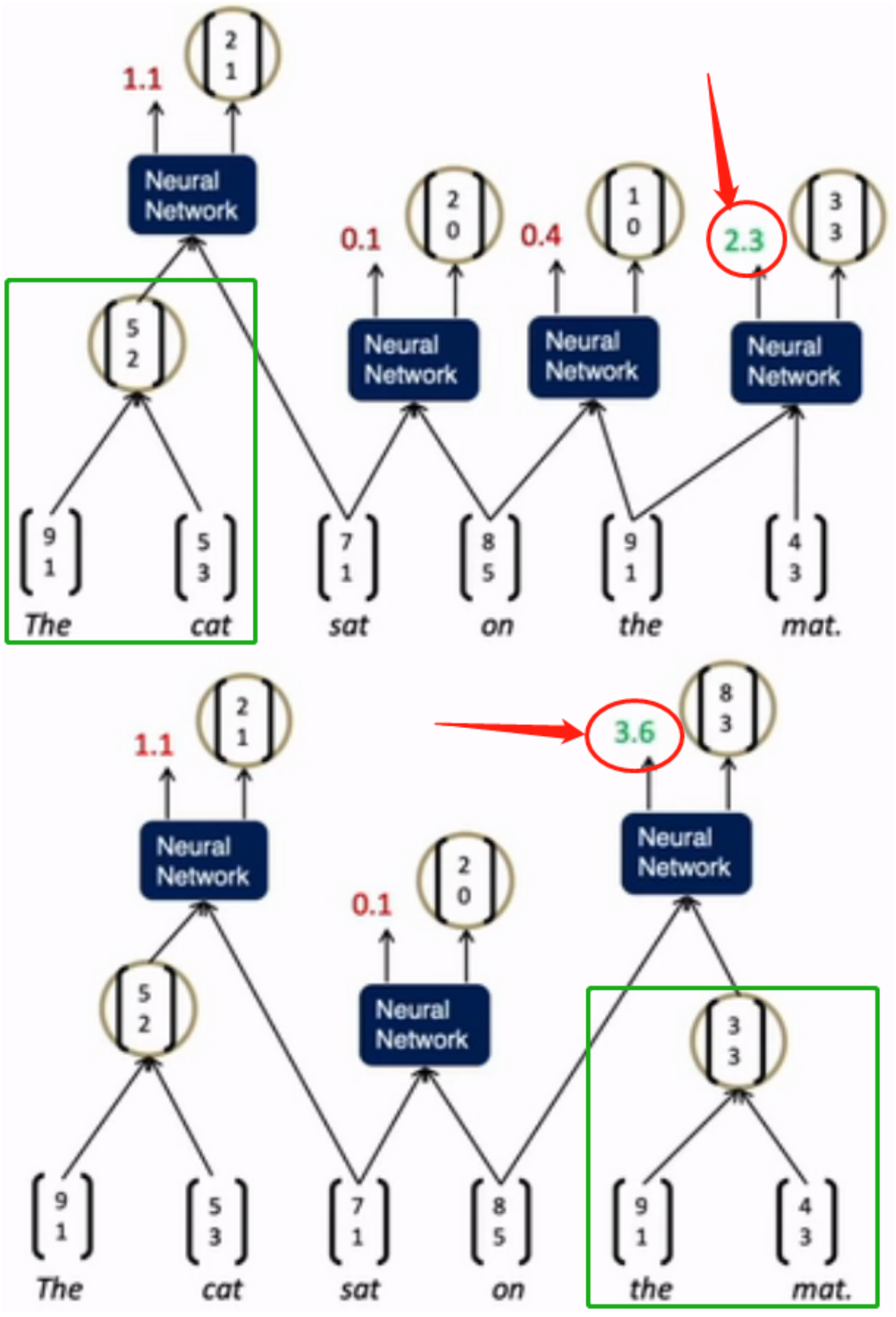
如果我们自上而下的工作，那么我们在底层有单词向量，所以我们想要递归地计算更大成分的含义。

已经有了on和mat的含义表示，将他们feed into神经网络，我们可以从中获得①分数，将用于解析的内容是否构成解析树的一部分。和②一个含义组合表示

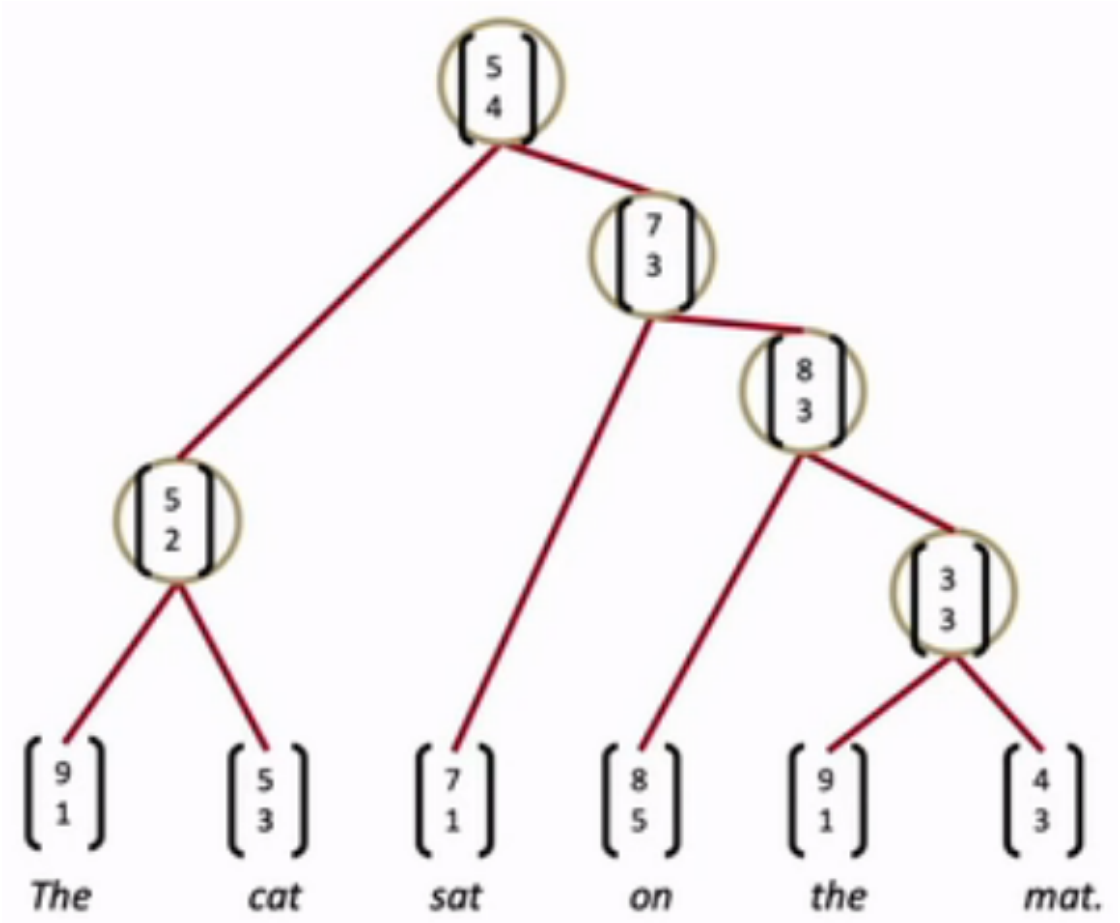


FIRST MODEL(greedily)


score = 1.3
= parent
Neural Network
c_1 c_2
score = $U^T p$
$p = \tanh\left(W\begin{bmatrix}c_1\\c_2\end{bmatrix} + b\right),$
Same W parameters at all nodes of the tree
3.1
0.3
0.1
0.4
2.3
Neural Network
The
cat
sat
on
the
mat.

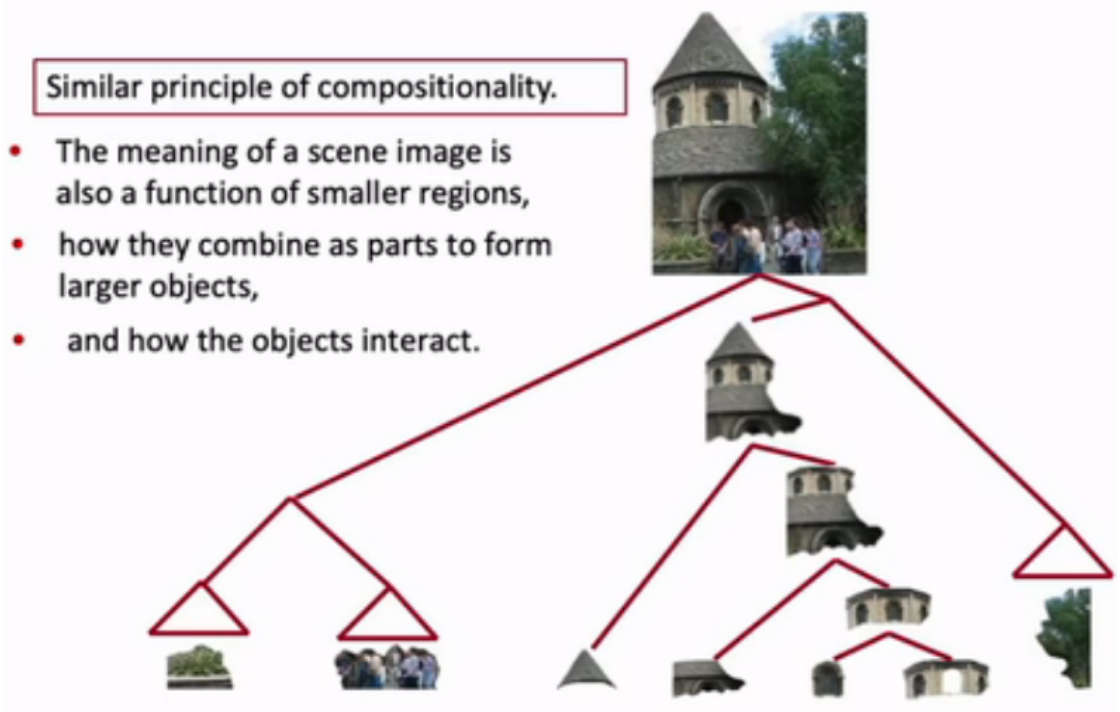

1.1
Neural Network
0.1
0.4
2.3
Neural Network
Neural Network
Neural Network
The
cat
sat
on
the
mat.
1.1
3.6
Neural Network
0.1
Neural Network
Neural Network
The
cat
sat
on
the
mat.

.....



Instead：Beam search with chart

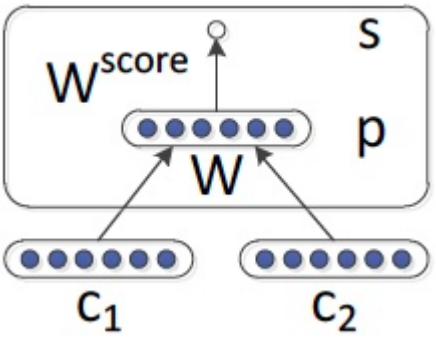
将这种方法用于视觉：



怎样建立神经网络来做到？

Backpropagation Through Structure. Goller & Küchler (1996)具体细节略

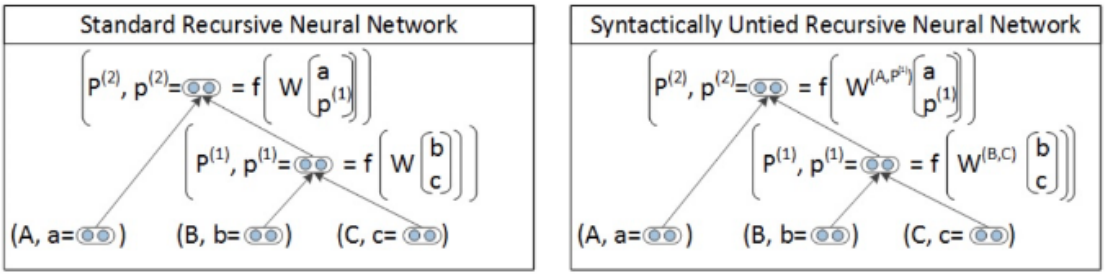
Simple TreeRNN



如果只是连接并乘以权重，实际上并没有模拟这两个向量之间的相互作用；且在上一个model中只有一个权重向量，用于整合短语的含义是行不通的。

Syntactically-Untied RNN**

Socher, Bauer, Manning, Ng 2013
用于无上下文风格选区解析
使用常规的概率无上下文语法为句子生成可能的树结构



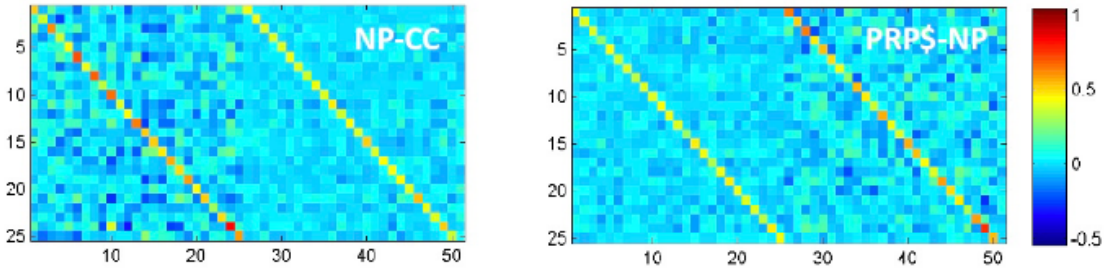
对于每个节点和句子都有一个类别，我们可以使用对应不同类别的矩阵组合起来，例如将类别 B 和类别 C 的矩阵组合起来作为本次计算的权重矩阵，所以这个权重矩阵是更符合句子结构的

Compositional Vector Grammars

PCFG（Probabilistic Context-free Grammar）和树递归神经网络的组合
不仅有提供正确的解析树的解析器，还计算节点的意义表示。还可以了解到这些模型正在

$$W^{(\cdot\cdot)} = 0.5[I_{n\times n} I_{n\times n} 0_{n\times 1}] + \epsilon$$

学习的权重矩阵
当我们加载这些矩阵时，将它们初始化为一对对角矩阵（因为有两个子节点），会给我们提供默认的平均语义，直到学习到了不同的东西

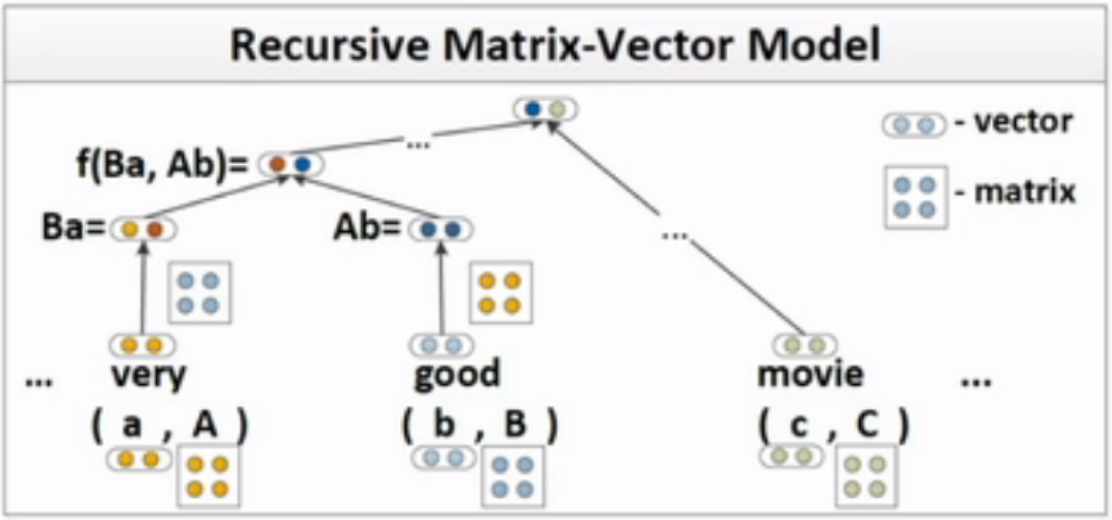


在某种程度上他学习到一些有趣的东西——取出某一子节点的语义，可以看到右边对角线

上的红色和橙色和其他领域的深蓝色和绿色。当训练这个模型时，它在学习一个短语的哪个子节点是重要的。如：
左图：“the cat and”。大多数语义必须在“the cat”中找到，并没有太多语义在“and”中。
右图：“her tabby cat”。大多数含义在“tabby cat”。
所以模型实际上在学习句子的重要语义。
这个模型能够很好的捕捉短语和句子的含义。

Compositionality Through Recursive Matrix-Vector Spaces

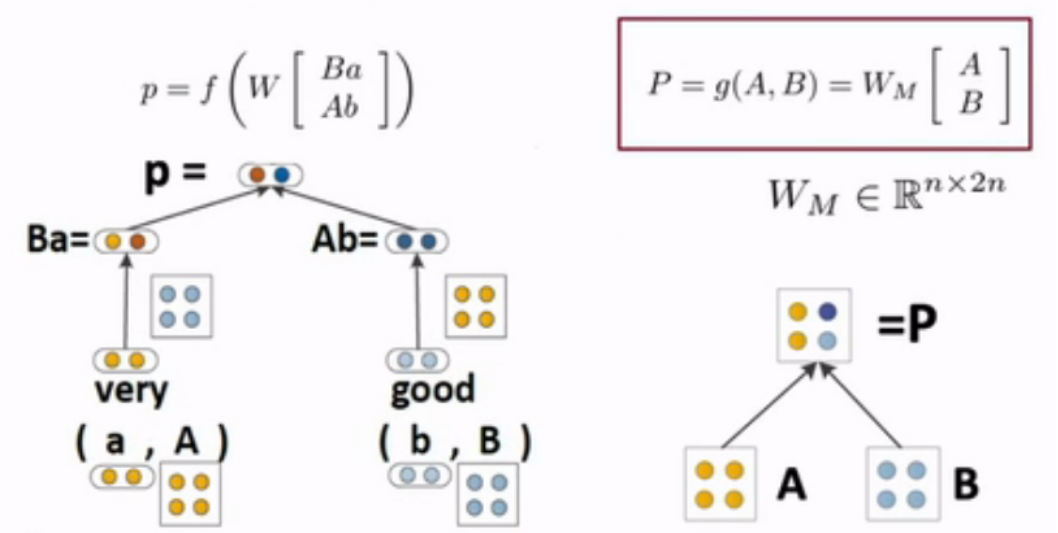
修饰语或者运算符的单词例如“very good”里的very，没有什么含义，类似于一种操作增加了very后面词的规模。如果想要捕获这类语义，不能仅通过之前的简单方法，而应该抓住“good”的含义，并在某些方面修正使产生“very good”的含义。



上图中，每个词都有一个向量含义与一个矩阵含义，然后开始构建短语，例如“very good”，它们也具有vector meaning和matrix meaning。

- 首先计算vector meaning。结合每个词的matrix和vector meaning，“good”的matrix meaning乘上“very”的vector meaning；“very”的matrix meaning乘上“good”的vector meaning。然后通过神经网络层将它们结合起来f(Ba,Ab)。.....最终得到整个短语的vector meaning。

- matrix meaning。



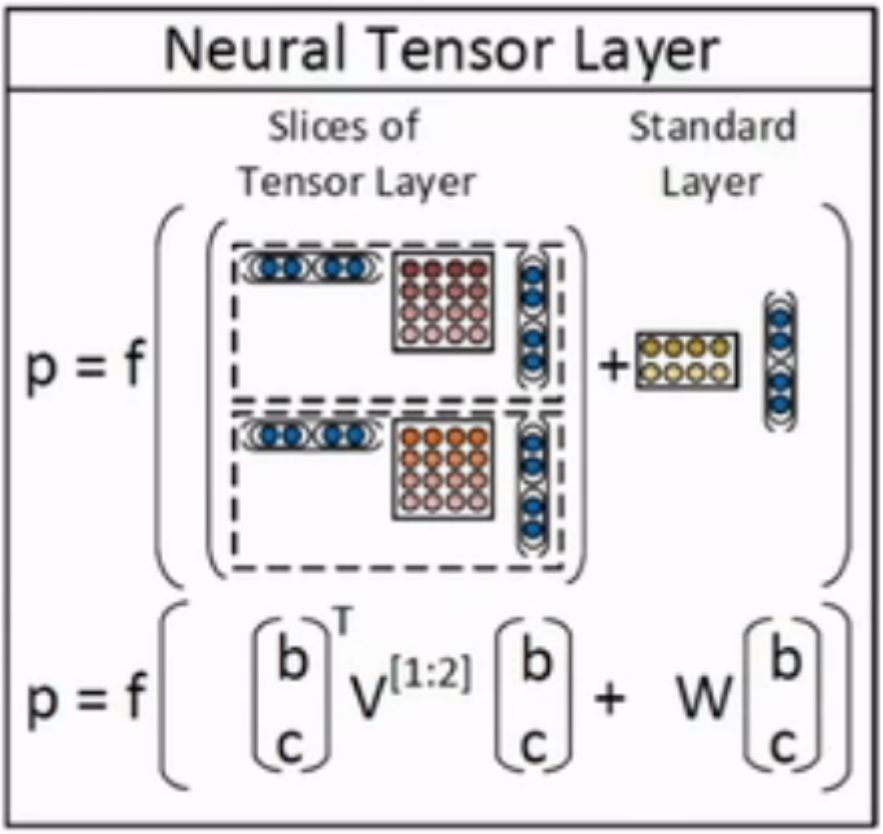
简单连接两个子节点的matrix meaning，再乘以一个权重矩阵得到父节点的matrix meaning

Recursive Neural Tensor Network

用于情感分析的例子：

With this cast, and this subject matter, the movie should have been funnier and more entertaining.

首先建立了一个Sentiment Treebank来让人们对情感进行评价句子中的每一个短语是积极的还是消极的。
接下来就是建立更好的模型来解决这个问题——Recursive Neural Tensor Network



RNTN 可以捕捉类似 X but Y 的结构，在
双重否定的句子上的表现也比其他模型好。

以上这些模型要做什么取决于句子的结构，每个句子都有不同的结构，因此没有办法批量计算一组句子并进行相同的计算。

在其他领域的应用： QCD-Aware Recursive Neural Networks for Jet Physics

Gilles Louppe, Kyunghun Cho, Cyril Becot, Kyle Cranmer (2017)
通过构建树递归神经网络编码解码器来进行编程语言之间的翻译。