周报\_2018\_9\_23

1. DAG-Structured Long Short-Term Memory for Semantic Compositionality（2016）

在一般的处理序列的模型中，都有一个较强的假设，学习的representation是完全由原子组件构成的，然而在人类的语言以及其他的一些模型中，非组合性是一个基本的现象。如何同时考虑序列模型中的组合性与非组合性是该论文所解决的问题。

该论文提出了一个有向无环图结构的LSTM。引入了non-compositional内存块的概念，使得LSTM能够同时考虑序列的组合性以及非组合性。

1. DeepWalk: Online Learning of Social Representations（2014）

该论文的主要贡献是用来建模自然语言的技术，可以被用来在建模网络中的关系。即是将节点构成的序列看作是类似与语言模型中的序列，然后进行建模。（将walks序列看做是特定语言中的短句或者短语）

该论文对中心论文而言，主要是提供了random walk的采样方法，即是从图中采集特定两个节点之间的路径。

1. Semantic Proximity Search on Heterogeneous Graph by Proximity Embedding（2017）

（该论文与“中心”论文的作者相同）

该论文给出了语义邻近搜索的较为清晰的说明：在graph中，给一个邻近类和一个指定类型的查询节点，语义邻近搜索旨在输出与查询节点相同类型的节点列表，该列表是根据与查询节点的相似程度定义的。语义邻近搜索的难点也在于如何测量异构图中不同种类对象的距离。

该论文的创新点在于它是direct proximity embedding，即是它的embedding是基于两个节点的路径所得到的。 它生成的embedding向量不仅仅是一个节点的向量，而是f(q,v)，即是q与v的邻近程度的向量。

该论文的实现流程大致是：

1. 在graph中使用random walk进行路径采样；（采样后每条路径的长度是相同的，即路径是与q节点相距为L的所有路径）
2. 并在所采样的路径中进行子采样，取出(q,v)的子路径；
3. 使用LSTM进行序列的嵌入；
4. 因为对每条路径都会生成一个embedding 向量，因此需要对这些embedding向量进行权衡，引入了a参数，a越大越倾向于路径长度越短的路径；
5. 最后使用以上生成的embedding向量计算proximity scores；最终的目标是使得与q越相近的节点的分数要越大。

该论文的实现与“中心”论文的实现几乎完全一样。唯一的区别在于“中心”论文中考虑的是交互路径的嵌入，而该论文并没有考虑交互路径。