DKRL 阅读记录

全称：Description-Embodied Knowledge Representation Learning

思路：在基于 triples 的基础上，结合每个实体各自对应的 description 来提升 embeddings 的学习——结合 Structure-based Representations 和 Description-based Representations 两种信息来源来优化 KG 中 embeddings 的学习

数学公式

得分函数

Score function:

$$E = E_S + E_D \tag{1}$$

其中,

$$E_S = \|h_s + r - t_s\| \tag{2}$$

$$E_D = \|h_d + r - t_d\| + \|h_d + r - t_s\| + \|h_s + r - t_s\| \tag{3}$$

h_s, r, t_s 是传统的 transE 等仅基于 Structure-based Representations 的 RL 模型学习出来的实体/关系 embeddings，而 h_d, t_d 则是通过编码各个实体对应的 description（即Description-based Representations）所得到 embeddings

目标函数

Goal function

$$L = \sum_{(h,r,t)\in T} \sum_{(h',r',t')\in T'} \max\left(0, \gamma + d(h+r,t) - d(h'+r',t')\right) \tag{4}$$

其中

$$T' = \{(h',r,t)|h' \in E\} \cup \{(h,r,t')|t' \in E\} \cup \{(h,r',t)|r' \in R\}$$

需要注意到的是，由于对于每个 h 和 t 均有两种 embeddings 表示，式（4）中的 entity，可以是两种中的任意一种。例如，可以是

$$\begin{aligned} &\max\left(0, \gamma + d(h_d + r, t_d) - d(h'_d + r', t'_d)\right) \quad + \\ &\max\left(0, \gamma + d(h_s + r, t_s) - d(h'_s + r', t'_s)\right) \quad + \\ &\max\left(0, \gamma + d(h_d + r, t_s) - d(h'_d + r', t'_s)\right) \quad + \\ &\max\left(0, \gamma + d(h_s + r, t_d) - d(h'_s + r', t'_d)\right) \end{aligned} \tag{5}$$

通过查阅代码，发现它的实现也是用式（5）的形式

上述原文引用：

> Since there are two types of representations for both h and t, entities in the margin-based score function could either be structure-based representations or description-based representations.

模型架构

论文中采用了两种模型来对 description 进行编码，依次为 CBOW 和 CNN；而对于 Structure-based Representations 的编码则是采用了传统的 transE

CBOW（词袋模型)

模型架构：

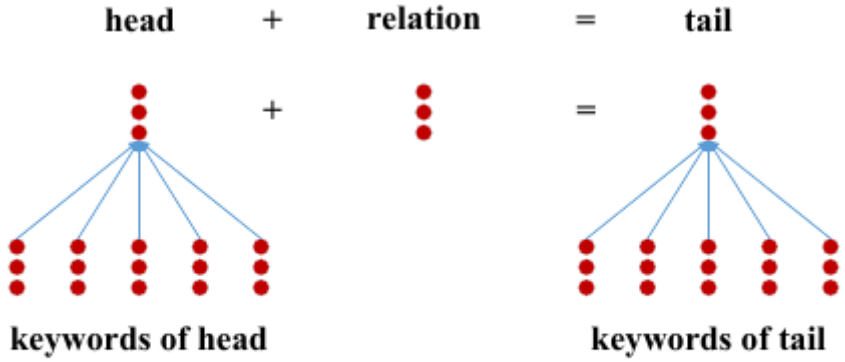

Figure 2: The CBOW Encoder

词袋模型的数学公式：$p(w|Context(w))$，其背后的意图为：通过给定单词 w（在本论文中为实体）上下文（本论文中即为实体对应的description）的前提下，出现该单词 w 的概率值。词袋模型所做的便是对每个单词 w 都计算对应的概率值，得到所有单词的联合概率，然后采用最大似然的方法来最大化这个联合概率值，具体的可参考博客链接：https://blog.csdn.net/u014595019/article/details/51943428 (https://blog.csdn.net/u014595019/article/details/51943428) 和 https://blog.csdn.net/itplus/article/details/37969979 (https://blog.csdn.net/itplus/article/details/37969979) 。总之，该模型的输入就是一段文本，输出为这段文本中各个实体对应的编码 embeddings（尚未确定，待后续看CBOW代码实现时再确认），论文就是利用所得到的这些编码作为模型的输入。

由于词袋模型忽略了description中实体出现的相对顺序，故而论文认为这会丢失掉一些重要的信息，故而提出另一个将相对顺序考虑入内的模型——CNN

CNN（卷积神经网络）

模型架构：

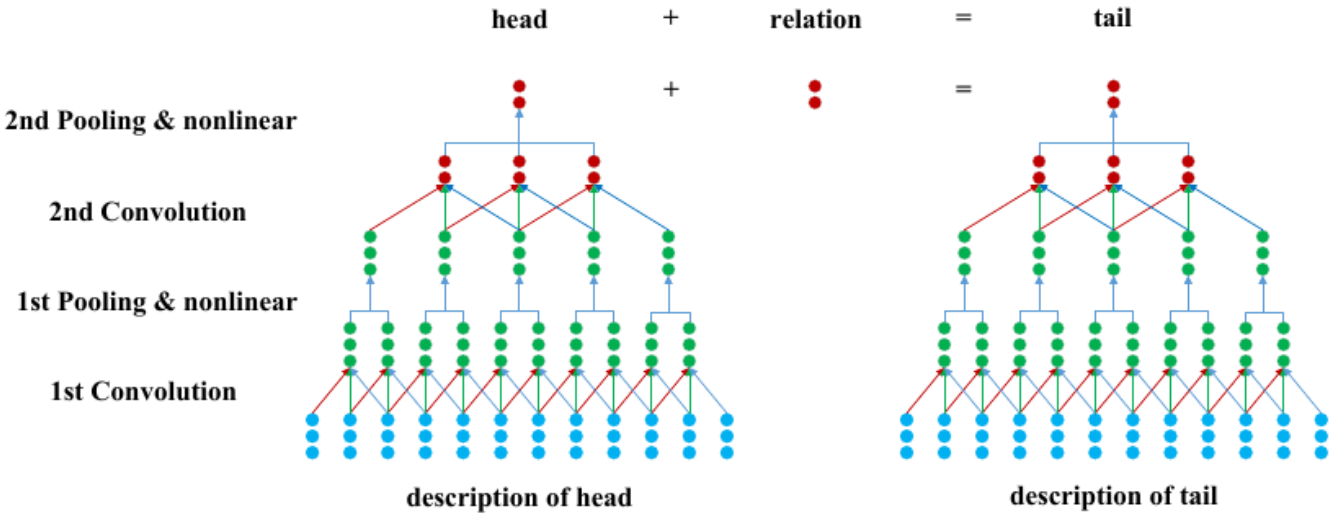

Figure 3: The Convolutional Neural Network Encoder

该模型有5层：`Input layer => CONV layter => Max-Pool layer => CONV layter => Mean layer(Output layer)`。模型以实体的整段经过预处理后的 description 作为输入，输出则是这个实体的 description-based 表示。数据的预处理如下：

> In preprocessing we first remove all stop words from raw texts, then we mark all phrases in descriptions (we simply select all entity names in training set as phrases) and consider those phrases to be words. Afterwards, each word is represented by a word embedding as the input of convolution layer. In our experiments, we use the word embeddings trained on Wikipedia by word2vec (Mikolov et al. 2013) as inputs for the CNN Encoder.

卷积和pool的细节查阅论文

困惑

目前不清楚它是如何将 CBOW 和 CNN 模型融入到公式（1）中的，在将模型的输出作为公式（1）的输入后，在公式（1）的优化过程中会不会将信息“反馈”给模型？同时，CNN是如何优化的，即它要对齐的标准是什么？

目前我个人的理解是存在“反馈”机制，模型对齐的标准其实就来自于“反馈”的信息——即将模型编码得到的embedding直接代入到公式（4）中，将公式（4）的输出作为模型的 loss function（即公式（4）就是模型的loss function），优化的目的就是通过调整模型的权重值来最小化公式（4）。