

# 知识图谱表示学习研究综述

刘宾楚 等

**摘要:** 知识图谱的表示学习能够将图结构的知识图谱数据表示为连续, 低维度, 实值的向量形式, 为知识图谱的研究与应用提供了便利。本文首先通过介绍一些典型的表示学习方法类型, 向读者阐述了表示学习的主要思想。然而知识图谱中除了三元组信息 (fact), 还有许多其他可以利用的信息, 如实体类型, 文本描述信息, 复杂关系路径, 本文向读者介绍了这些方面典型的表示学习方法, 以使读者快速了解表示学习新进展。

**关键字:** 知识图谱, 表示学习, 综述, 翻译模型, 文本描述信息, 实体类型, 复杂关系路径

## 1 引论

近年来, 知识图谱方兴未艾, 知识图谱(knowledge graph)具有很强的语义处理能力, 借助于互联网快速发展产生的海量数据, 知识图谱这些数据以图亦称为网络连接起来, 构造出强大的知识网络, 成为人工智能领域一个重要的研究分支。

目前, 已经诞生了许多结构化的知识图谱, 如 WordNet, Freebase, DBpedia, YAGO等等。这些知识图谱成功的应用在现实生活的许多领域: 智能问答系统, 信息抽取, 语义推断等等。国内外的互联网和计算机领域的知名企业纷纷推出自己的知识图谱产品, 如谷歌, 微软, IBM, 百度, 搜狗等等。

构成知识图谱的最基本的单位是实体和实体之间的关系, 也即图中的点和边, 用三元组表示也就是(头实体, 关系, 尾实体), 这是知识图谱表示的通用方式, 符合W3C发布的RDF标准。这种表示方式显然具有图结构的特点, 使得人们对于知识图谱的研究和应用变得困难: 其计算效率因为图结构变得低下, 往往需要设计专门的图算法来进行知识图谱的研究和应用。因此将知识图谱中的图结构转换为向量也即知识图谱表示学习具有相当重要的研究价值和意义。

在2016年已经有清华大学刘知远等人对知识图谱表示学习做了综述[1], 然而时光荏苒, 表示学习领域发生了一些新变化, 本文将对表示学习领域新的研究成果做出展示, 首先阐述一系列经典模型, 并且指出了其若干不足之处, 接着阐述了若干这些不足所提出的一些模型, 并且针对每一种模型指出了其改进之处和应用。最后本文对每种类型的表示学习方法做出总结。

## 2 问题描述

### 2.1 知识图谱表示学习的定义

因为KG具有图结构的性质，为相关研究和应用带来不便，研究者们希望使用向量的数据形式表示知识图谱中的实体和关系，一些简单的表示方法，如独热表示（one-hot representation）[3]在文本数据中得到了很好的应用，但是对于图结构的知识图谱来说，它具有数据稀疏的问题，并且该模型假设实体间相互正交，也即实体间互无关系，显然不符合知识图谱的现实情况。为了更好的表示知识图谱中的数据，我们需要将其表示成：低维，实值，连续的向量表示，这也就是表示学习的目标和任务，而这一过程也可以看作将KG中的数据从原先空间中，投影或嵌入至另一个低维，连续，实值的空间中（如Fig.1所示）。因此KG的表示学习也叫做知识图谱嵌入（Knowledge Graph Embedding）。总而言之，知识图谱的表示学习就是将知识图谱中的实体和关系对应或者嵌入到低维空间的过程。

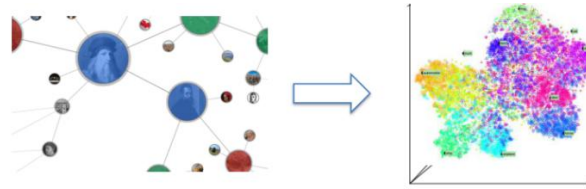


Fig. 1. 知识图谱表示学习图解

### 2.2 知识图谱表示学习的优点

通过知识图谱表示学习，将实体和关系投影到低维空间中，为知识图谱的研究与应用带来了一下的优点与便利

- 提高计算效率：知识图谱的图结构表示需要设计专门的图算法来研究和应用实体之间的语义和推理关系，计算复杂度比较高，但是通过表示学习RL得到数据的低维表示，可以得到更好的数学性质，高效的实现语义相似度计算等操作，明显的提高了计算效率
- 减轻数据稀疏的性质：知识图谱的图结构三元组表示本质上是基于独热表示的，这产生了严重的数据稀疏问题，而表示学习将数据投影嵌入到低维空间：低维空间中每个单独的一个维度都没有意义，但是将所有唯独组合成一个向量则能够表示一个实体或者是关系，因此这种表示也叫做分布式表示（distributed representation）。由此低维空间中的数据更加稠密，缓解了数据稀疏的问题。

## 2.3 常用符号定义

在本节，我们将要定义几种在本文中常用到的符号。首先将知识图谱KG表示为 $KG = (E, R, S)$ ，其中 $E = \{e_1, e_2, e_3, \dots, e_{|E|}\}$ 表示知识图谱中实体的集合，其中 $|E|$ 表示实体的个数，类似的R和S表示知识图谱中关系和三元组的集合，其中三元组的集合也就是图中边的集合。本文将三元组表示为 $(h, r, t)$ ，分别表示头实体，关系，尾实体，比如 $(SEU, isLocatedIn, NanJing)$ ，头实体和尾实体在低维空间中的向量表示分别为 $l_h, l_t$ 。

## 2.4 模型评价标准

得到模型后，大多数论文都采用链接预测的效果来描述模型的好坏。假设整个知识库中一共有 $n$ 个实体，那么评价过程如下：

- 将一个正确的三元组 $a$ 中的头实体或者尾实体，依次替换为整个知识库中的所有其它实体，也就是会产生 $n$ 个三元组。
- 分别对上述 $n$ 个三元组计算其损失函数值。
- 对上述 $n$ 个三元组对应的损失函数值进行升序排序。
- 记录三元组 $a$ 的损失函数值排序后的序号。
- 对所有的正确的三元组重复上述过程。
- 对每个正确三元组的损失函数值排序后的序号求平均，得到的值我们称为Mean Rank。
- 计算正确三元组的损失函数值排序后的序号小于10的比例，得到的值我们称为Hits@10。

那么Mean Rank和Hits@10就是常用的两个表示学习模型的评价指标。显然Mean Rank反映了一个平均水平，其数值越小则模型越优秀，而Hits@10则是越大越好。

# 3 传统知识图谱表示学习方法

本节介绍的一些知识图谱表示学习方法都只用到了知识图谱中所包含的三元组信息，也即（头实体，关系，尾实体）信息，因此将这种方法成为传统知识图谱表示学习方法。

## 3.1 张量神经网络模型(neural tensor network, NTN)

张量神经网络模型(neural tensor network, NTN)的基本思想是：用双线性张量取代传统神经网络中的线性变换层，在不同纬度下将头实体、尾实体向量联系起来。其基本思想如下图所示：

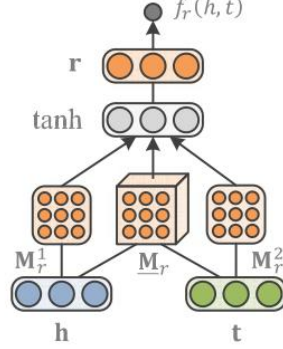


Fig. 2. NTN 模型

张量神经网络模型为每个三元组 $(h, r, t)$ 定义了损失函数去判断实体之间存在关系 $r$ 的可能性:

$$f(h, t) = u_r^T (l_h M_r l_t + M_{r,1} l_h + M_{r,2} l_t + b_r) \quad (1)$$

其中 $u_r^T$ 是一个与关系相关的线性层,  $g()$ 是  $\tanh$  函数,  $M_r$  是一个三阶张量,  $M_{r,1}$ 和 $M_{r,2}$ 是关系 $r$ 有关的投影矩阵。值得注意的是, NTN中的实体向量是该实体中所有单词向量的平均值, 这样能够使得实体中的单词数量远小于实体数量, 可以充分利用单词向量构建实体表示, 降低实体表示学习的稀疏性问题, 增强不同实体的语义联系。

由于张量神经网络模型引入了张量操作, 虽然能更精确地刻画实体和关系的复杂语义联系, 但是计算复杂性非常高, 需要大量三元组样例才能得到充分学习, 张量神经网络在大规模稀疏知识图谱上的效果较差。

### 3.2 DeepWalk及其拓展方法

#### Deepwalk

Deepwalk是一种在线学习算法, 它可以构建有用的增量结果, 并且可以并行地并行化。不同于传统意义上的高纬度降低到低纬度, 而是将一个复杂的结构降低到低纬度。或者说可以理解为, 将网络中的拓扑结构, 嵌入到一个低维向量中, 每个节点的低维向量, 从某种程度上反应了该节点在网络中的连接情况。

Deepwalk 将网络图谱嵌入到一个低维的向量中。它使用了 Skip-gram 的方法进行网络中节点的表示学习。随机游走随机均匀地选取网络节点, 并生成固定长度的随机游走序列, 将此序列类比为自然语言中的句子 (节点序列=句子, 序列中的节点=句子中的单词)。下图是Deepwalk的步骤图示:

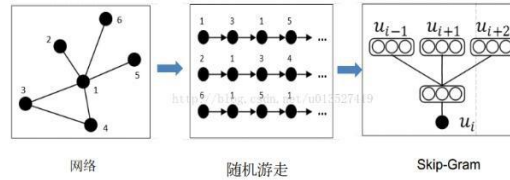


Fig. 3. Deepwalk 图解

### LINE.

LINE是一种基于简单神经网络并适用于大规模图结构的网络表示学习方法，该方法提出了一级相似度与二级邻近度的概念来进行节点之间的概率建模，并用到了 KL 散度去度量概率分布之间的距离参数，并且使用SGD优化算法。

- 一级相似度：直接相连的结点较为相近，这和原本的邻接矩阵的建模比较相似，需考虑到边关系仍旧是比较稀疏的；
- 二级邻近度：两个节点的邻居节点有重叠，即两个有共同邻居节点的节点对也是相近的。可以看到，二级相似度对一级相似度进行了相应的补充。这两种相似度都对节点之间的概率进行了计算，越是相近的点算出的概率越大。

LINE 主要是用于单关系网络结构，即当两个节点存在多条关系无法处理。当关系边存在语义特征或是权重时，LINE 无法去做出有效的相近性的判断。LINE、DeepWalk 都是属于网络表示方法（network embedding），都较多的用在单关系网络中。

### 3.3 翻译模型

在只利用到三元组信息的表示学习方法中，TransE及其衍生方法最负盛名，这是一种典型的翻译距离模型，将关系视为从头实体到尾实体的翻译。

#### TransE

TransE模型[2]将头实体和尾实体都视为低维空间中的向量 $h, t$ ，而关系则被视为翻译向量 $r$ ，那么头实体和尾实体也就近似的通过翻译向量 $r$ 联系起来，也即： $h + r \cong t$ ，如下图所示：

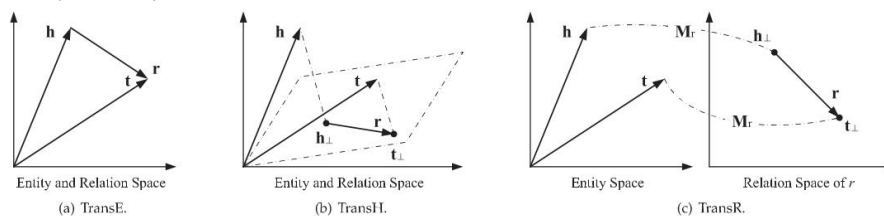


Fig. 2. TransE, TransH, TransR模型图示

TransE模型通过定义损失函数(1)，并且最小化损失函数使得达到将关系视为头实体和尾实体之间的翻译向量。

$$f_r(h,t) = |l_h + l_r - l_t|_{L1/L2} \quad (1)$$

TransE模型满足词向量空间中的平移不变现象，也即若在知识图谱中存在两个三元组 $(l_{h1}, l_{r2}, l_{t2}), (l_{h1}, l_{r2}, l_{t2})$ 满足如下关系： $l_{r1} \cong l_{r2}$ ，那么通过TransE训练得出的向量有： $l_{t2} - l_{h2} \cong l_{t1} - l_{h1}$ ，例如：男人和女人之间的距离约等于国王和王后之间的距离。此外TransE模型还有简单有效的特点，算法思想十分简单，易于理解，模型参数较少，因此计算复杂度不高。

但是其简单的特点也带来了问题，知识图谱中的关系可以大致分为四种：1-1，1-N，N-1，N-N，而TransE对与1-1关系有较好的效果，但是面对其他三种复杂关系，效果较差。如果一个关系是N-1关系，那么在TransE中有 $l_{h1} \cong l_{h2} \dots \cong l_{hn}$ ，对于其他两种复杂关系也有同样的效果，例如(东南大学，校长，张广军)，(东南大学，校长，顾冠群)，两个尾实体的向量相似，但实际上，两个尾实体除了作为东南大学校长这一方面比较相似，其他方面差别可能非常大。为了解决上述问题zhen等人提出了TransH模型。

### TransH

Zhen等人在论文Knowledge Graph Embedding by Translating on Hyperplanes中提出transH模型[4]，通过上面的例子分析我们可以得出以下结论：

在知识图谱的各种关系，尤其是复杂关系中，每个实体都可能与多种关系匹配，也即每个实体都由不同的属性方面组成，不同的关系只关注其中一种属性方面。

这也是TransE模型无法处理复杂关系的原因，而TransH模型首先将实体投影至关系所在的超平面上，再通过翻译模型，训练得出向量，如图Fig.2所示。损失函数如下：

$$l_{hr} = l_h - w_r^T l_h w_r, \quad l_{tr} = l_t - w_t^T l_t w_r \quad (2)$$

$$f_r(h,t) = |l_{hr} + l_r - l_{tr}|_{L1/L2} \quad (3)$$

其中 $w_r$ 为关系的投影矩阵。

### TransR

TransR[5]认为TransH仅仅将实体投影至不同的超平面之上限制了其表示能力，TransR将每个实体投影至不同的语义空间中，将实体的不同属性方面投影至不同的语义空间，如图Fig.2所示；其公式化描述如下：

$$l_{hr} = l_h M_r, \quad l_{tr} = l_t M_r \quad (4)$$

$$f_r(h,t) = |l_{hr} + l_r - l_{tr}|_{L1/L2} \quad (5)$$

### TransD

TransD[6]修正了TransR的若干缺点：

- TransR没有区分头实体和尾实体的投影矩阵，也即认为同一个关系联系的两个实体的属性方面相同，与实际情况不同，如上例(东南大学，校长，张广军)，头实体和尾实体的类型显然不同，TransD将投影矩阵分解为两个向量，分别将头尾实体投影至语义空间。
- 通过将矩阵分解为两个向量，TransD缓解了TransR由于投影矩阵带来的参数过多，计算复杂度大的问题。
- TransD把投影过程同关系与头尾实体联系起来，而TransR的投影操作只与关系相关

我们将两个投影向量定义为  $w_h, w_t$ ，那么投影矩阵则为

$$M_r^1 = l_r w_h^T + I, \quad M_r^2 = l_r w_t^T + I \quad (6)$$

接下来通过投影矩阵将实体投影至语义空间之中：

$$l_{hr} = l_h M_r^1, \quad l_{tr} = l_t M_r^2 \quad (7)$$

那么通过最小化损失函数，得到学习结果：

$$f_r(h, t) = |l_{hr} + l_r - l_{tr}|_{L1/L2} \quad (8)$$

### KG2E

KG2E[7]模型认为知识图谱中的实体和关系有一定的不确定性，因此应该使用正态分布来描述每一个实体和关系，正态分布的均值向量即为在语义空间中的位置向量，而协方差矩阵则为其不确定性程度，也即：

$$h \sim N(\mu_h, \Sigma_h), \quad t \sim N(\mu_t, \Sigma_t), \quad r \sim N(\mu_r, \Sigma_r) \quad (9)$$

其中  $\mu_h, \mu_t, \mu_r$ ，分别表示头实体，尾实体，关系的均值向量， $\Sigma_h, \Sigma_t, \Sigma_r$ ，分别表示三者的协方差矩阵。KG2E也利用了翻译距离的思想，KG2E最小化两个随机向量的距离， $h - t \sim N(\mu_h - \mu_t, \Sigma_h + \Sigma_t)$  和  $r \sim N(\mu_r, \Sigma_r)$ 。KG2E使用了两种不同的距离度量，第一种是KL散度（Kull-back-Leibler divergence），也叫信息增益（information gain），并且定义了如下的损失函数：

$$f_r(h, t) = - \int N_x(\mu_h - \mu_t, \Sigma_h + \Sigma_t) \ln \frac{N_x(\mu_h - \mu_t, \Sigma_h + \Sigma_t)}{N_x(\mu_r, \Sigma_r)} dx - \text{tr}(\Sigma_r^{-1}(\Sigma_h + \Sigma_t)) - \mu^T \Sigma_r^{-1} \mu - \ln \frac{\det(\Sigma_r)}{\det(\Sigma_h + \Sigma_t)} \quad (10)$$

另一种距离度量则是概率内积，并且定义了如下损失函数：

$$f_r(h, t) = - \int N_x(\mu_h - \mu_t, \Sigma_h + \Sigma_t) \cdot \ln N_x(\mu_r, \Sigma_r) dx - \mu^T \Sigma_r^{-1} \mu - \ln \det(\Sigma) \quad (11)$$

### 其他距离翻译模型

还有许多距离翻译模型拓展了TransE的设计方法，为了解决实体和关系的异质性和不平衡性，提出了TranSparse[8]模型，TransA[9]模型将距离度量由L1/L2距离更改为马氏距离（Mahalanobis distance），TransF[9]模型则将 $l_h + r \cong l_t$ 的条件放宽至 $l_t$ 同 $l_h + r$ 方向一致，且 $l_h$ 同 $l_t - r$ 方向一致。此前曾提到，每个实体都是由若干个属性方面构成，TransG[10]采用高斯混合模型的思想，对于实体的每一个属性方面，也即语义使用一个高斯分布刻画。

## 4 传统知识图谱表示学习改进方法

上文所介绍的翻译模型方法都有一个共同的特点：所用到的实体，关系都存在于知识图谱中，然而现有的知识图谱并不能完全包括世界上所存在的知识，在知识图谱中除了实体和关系之外还有一些其他能够利用的信息，如：

- 实体的类型
- 两个不直接相连的实体之间的路径即复杂关系路径
- 实体和关系标签之中的文本关系

在表示学习方法中利用上述方法都能够提升表示学习的性能，在下面的章节中将要介绍若干典型的利用到上述信息的表示学习方法。

### 4.1 利用到实体类型的表示学习方法

在大多数知识图谱都记录了实体的语义类型，比如：东南大学的类型就是学校/大学，张广军的类型则是人/校长，并且通常在知识图谱中也用三元组来表示，如：（东南大学，是，学校），因此表示学习显然可以利用这些类型信息。

#### SSE

semantically smooth embedding模型（SSE）[11]通过将设置约束条件：与有相同类型的实体在语义空间中距离相近。有两种学习算法可以得到满足这个条件的投影。

第一种是拉普拉斯特征映射（Laplacian eigenmaps），这是一种降维算法，其直观思想是希望属于同一类的每一个实体在学习后的空间中尽可能的靠近。定义如下的光滑度：

$$R_1 = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \|e_i - e_j\|_2^2 w_{ij}^1 \quad (12)$$

其中 $e_i, e_j$ 表示实体在投影空间中的向量， $w_{ij}^1$ 有两个取值，取1表示两个实体属于同一个类别，取0则表示属于不同的类别，通过最小化 $R_1$ 可以得到符合要求的投影向量。

第二种方法则将每一个实体向量用同一个类别的其他实体向量线性组合表示，也即每一个实体向量都等于同一类别中其他向量的线性相加，也即如下光滑度函数：



$$R_2 = \sum_{i=1}^n \left\| e_i - \sum_{e_j \in N_{e_i}} w_{ij}^2 e_j \right\|_2^2 \quad (13)$$

其中 $e_{j \in N_{e_i}}$ 表示同 $e_i$ 属于同一个类型的 $e_j$ ,通过最小化 $R_2$ ,我们将每一个实体的向量线性化重构为其同类型的实体向量的线性组合.

然后将 $R_1, R_2$ 合并起来作为正则化项加入投影任务之中,[11]证明了SSE相比于直接投影的方法,具有更好的效果.

### TKRL

SSE的一个重要假设:实体中的语义类别没有从属关系,这显然不符合下现实情况,比如:东南大学这一实体的类型为学校/大学,显然具有两层从属关系.TKRL[12](type-embodied knowledge representation learning)模型能够处理有层次(hierarchical)的实体关系和实体类别的多标签性质.

TKRL首先将头实体 $l_h$ 和尾实体 $l_t$ 使用和类别相关的投影矩阵,投影至低维,实值,连续的投影空间之中,然后将关系 $l_r$ ,作为从头实体到尾实体的翻译,可见TKRL也是一种距离翻译模型.

$$f_r(h, t) = -|M_{rh}l_h + l_r - M_{rt}l_t|_{L1} \quad (14)$$

其中 $M_{rh}, M_{rt}$ 是分别对应于头实体和尾实体的两个投影矩阵,是实体所有可能属于的类别的投影矩阵的加权平均值,也即:

$$M_{rh} = \frac{\sum_{i=1}^{n_h} a_i M_{ci}}{\sum_{i=1}^{n_h} a_i}, a_i = \begin{cases} 1, c_i \in C_{rh} \\ 0, \text{其他} \end{cases} \quad (15)$$

其中 $n_h$ 代表头实体所属的类别的个数,  $M_{ci}$ 表示类别 $c_i$ 的投影矩阵,  $C_{rh}$ 表示头实体所属的类别的集合.

实验证明TKRL能够在实体关系预测,三元组分类等任务具有很好效果,但是由于其引入了层次化的类别信息,其参数空间较大,计算复杂度较高.

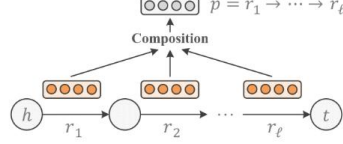
## 4.2 复杂关系路径建模

本文介绍的第二种可以利用的信息是复杂关系(multi-hop relationships)的关系路径,在连通图中,任意两个实体之间的关系,实际上同他们之间的路径有非常大的关系.比如以下两个三元组(NENU, nearBy, JLU), (JLU, isLocatedIn, Changchun), 那么NENU同Changchun之间的关系也就是nearBy和isLocatedIn这两个关系的语义组合.那么利用到multi-hop关系的表示学习方法的一个关键点就是:如何将路径上的关系的语义组合表示成向量.

### PTransE

如上文所述,利用复杂关系路径的一个关键是:如何将复杂关系的语义组合表示成向量,并且同头尾实体处在同一个语义空间之中.一个直接的想法是将复杂关

系路径上的关系组合起来,作为目标关系向量. PTransE[13]也利用了这种思想,如下图所示:



**Fig. 3.** Path representations are semantic compositions of their relation embeddings.

典型的组合方式有:相加(addition),相乘(multiplication),循环神经网络(RNN),给定一条复杂路径上的所有关系 $r_h \rightarrow \dots \rightarrow r_t$ , 有目标关系 $p = r_h \rightarrow \dots \rightarrow r_t$ ,那么则有以下三种组合方式:

$$\text{addition: } p = r_h + \dots + r_t \quad (16)$$

$$\text{multiplication: } p = r_h \cdot \dots \cdot r_t \quad (17)$$

$$\text{RNN: } c_i = f(W[c_{i-1}; r_i]) \quad (18)$$

其中,  $c_i$ 为第 $i$ 个关系的累积路径向量,  $W$ 是所有关系共享的合成矩阵,  $[c_{i-1}; r_i]$ 表示 $c_{i-1}$ 至 $r_i$ 的级联(concatenation),  $f$ 则是非线性函数. 通过令 $c_1 = r_1$ ,并从左到右地递归遍历, 最终可以得到 $p = c_l$ .

对于每一个在知识图谱中的三元组, PTransE定义了如下的损失函数:

$$f_p(h, t) = \frac{1}{Z} \sum_{p \in P(h, t)} R(p|h, t) \cdot \varphi(p, r) \quad (19)$$

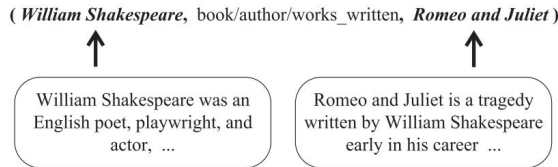
其中 $P(h, t)$ 是所有头实体 $h$ 和尾实体 $t$ 之间的路径的集合,  $R(p|h, t)$ 则表示给定首尾实体, 路径的可靠性(reliability),  $Z = \sum_{p \in P(h, t)} R(p|h, t)$ 为规范化因子,  $\varphi(p, r)$ 是同路径, 关系相关的损失函数, 其公式定义如下:

$$\varphi(p, r) = \sum_r \max(0, \gamma + \|p - r\|_{L1} - \|p - r'\|_{L1}) \quad (20)$$

实验证明, PTransE在实体预测方面相比于TransE有着35%左右的提升[13], PTransE在知识图谱补全方面也有着更好的效果.

### 4.3 融合文本信息

本节将讨论如何将实体的文字描述信息同表示学习方法融合, 事实上, 在大多数知识图谱的实体标签中, 都有一些简单的描述, 比如在Freebase中实体就有一些简要的描述:



**Fig. 4.** Freebase中实体描述信息的例子

实体中的文本描述信息能够提供许多额外信息,对于知识图谱补全,和实体类型分类等任务有着很大的提升作用.

#### DKRL.

DKRL[14] (Representation Learning of Knowledge Graphs with Entity Descriptions)模型能够将文本描述信息利用到表示学习之中, 其损失函数如下:

$$f(h,r,t) = f_s(h,r,t) + f_D(h,r,t) \quad (21)$$

可见DKRL将表示学习的过程划分为两个部分,前半部分利用知识图谱中存在的结构化的信息, 也即传统的表示学习部分,DKRL采用TransE模型处理这一部分的表示学习过程, 也即公式前半部分等于公式 (1)

第二部分是对于知识图谱中实体描述信息的处理,其损失函数如下所示:

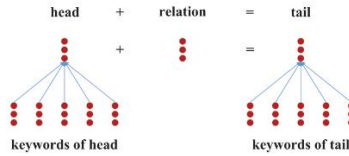
$$f(h,r,t) = |l_{hd} + l_r - l_{td}|_{L1/L2} + |l_{hd} + l_r - l_{ts}|_{L1/L2} + |l_{hs} + l_r - l_{td}|_{L1/L2} \quad (22)$$

其中 $l_{hd}$ 是由描述信息得到的头实体的向量表示,  $l_{hs}$ 表示由结构信息得到的头实体的向量,  $l_{td}$ ,  $l_{ts}$ 也有类似的含义, 公式 (22) 将结构信息和描述信息得到的实体向量混合起来训练翻译模型, 且关系向量由损失函数的三个部分所共享. 实验证明, 这种混合方式能够相互促进两种不同的表示方法. 为了得到由描述信息生成的向量, DKRL采用了两种编码器: 连续词袋编码器, 卷积神经网络编码器

#### 连续词袋编码器

该编码器采用连续词袋模型 (CBOW) 的思想, 并且假设相类似的实体具有相类似的文本描述信息. 对于每个实体, 该编码器首先选取前 $n$ 个关键词作为输入 (使用TF-IDF等经典文本特征对关键词排序), 令 $e_d$ 表示描述信息得到的实体向量,  $x_i$ 表示实体描述信息所对应的第 $i$ 个关键词, 其组合方式如下:

$$e_d = \sum_{i=1}^n x_i \quad (23)$$



**Fig. 5.** The CBOWEncoder

可以看出, 连续词袋编码器忽略了词与词之间的顺序关系, 损失了一部分文本描述的语义信息。

### CNN编码器

CNN编码器克服了CBOW编码器忽略了词序信息，容易受关键词提取质量的影响，CNN编码器能够利用词序中隐藏的文本信息。其架构如图所示：

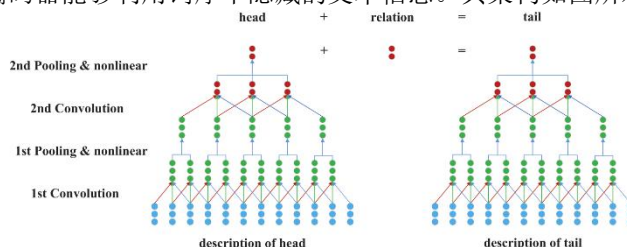


Fig. 6. The Convolutional Neural Network Encoder

第一层为输入层，首先是去除停用词，再将所有的实体名短语视作单词，最后将得到的词向量作为输入（可以使用word2vec在Wikipedia上的训练结果作为输入），第二，第四层为卷积层，通过卷积降低了向量维度，向量个数，提高了向量的稠密程度。在第三，第五层内，进行了pooling操作，以降低参数空间，过滤噪音。

通过CBOW或者CNN编码器，可以得到由描述信息所生成的实体的向量，再最小化公式（22）的损失函数，最后将结构化信息和描述信息结合起来，得到DKRL所学习的实体嵌入向量。

实验证明：不论采用哪种编码器的DKRL算法，在知识图谱补全和实体分类这些任务上，较单纯利用到结构信息的方法，都有较大的提升；并且CBOW编码器具有简单的特点，但是CNN编码器的效果更好[14]。

## 5 知识图谱表示学习的应用

本节将阐述知识图谱表示学习在各种知识图谱研究与开发任务中的应用。表示学习将知识图谱中的实体和关系表示为低维，连续，实值的形式。根据实际应用时采用的数据，可以将知识图谱表示学习应用分为知识图谱内部应用（In-KG Applications）和知识图谱外部应用（Out-KG Applications）。知识图谱内部应用（In-KG Applications）是指所使用到的数据也即实体，关系，都是表示学习训练后得到的数据，都是在现有知识图谱中存在的数据；而知识图谱外部应用（Out-KG Applications）则突破了这一限制。

### 5.1 知识图谱内部应用（In-KG Applications）

#### 链接预测 (Link Prediction)

给定一个事实（fact）中的关系和任意一个实体，预测另一个实体的任务被称为链接预测。也即若给定 $(r, t)$ 则需要预测 $h$ ，若给定 $(h, r)$ 则需要预测 $t$ 。比如：给定 $(SEU, isLocatedIn)$ ，则是预测东南大学的地址，其结果可能是

*Nanjing*。若给定(*isLocatedIn,Nanjing*)，则是预测地址为南京的一切实体，其结果有若干个。

### 三元组分类 (Triple Classification)

给定一个未知的三元组，判断其是否符合事实的任务被称作三元组分类，也即给定一个未知三元组，将其分为True或False的任务。假定三元组(*Nanjin,isLocatedIn,China*)若判断其为True，则(*Nanjin,isLocatedIn,USA*)和(*Nanjin,isNotLocatedIn,China*)则被判断为False。这一任务也可以视为某种类型的知识图谱补全。

### 实体分类(Entity Classification)

实体分类的任务是给定一个实体，判断其语义类别。比如给定实体*SEU*则其语义类别则是*school/university*。实际上，在知识图谱中，若实体的类型都已经由事实(fact)给出，也即对于每个实体存在一个三元组指出其类型，比如(*Nanjin,isA,City*)。那么实体分类的任务则可以视为链接预测(Link Prediction)的任务，即给定(*Nanjin,isA*)，预测其尾实体。实体分类也是一种知识图谱补全的任务。

### 实体解析(Entity Resolution)

在一些知识图谱中，一些实体实际上等价，都表示现实世界中的同一个事物，实体解析的任务则是判断两个实体是否等价，是否指向同一个事物。实体解析的任务是找出并且消除这样的重复实体。

## 5.2 知识图谱外部应用 (Out-KG Applications)

### 关系抽取(Relation Extraction)

在一段文本中，若两个实体已经检测出来，则关系抽取的任务是从文本中抽取。如给出文本“东南大学在南京”，已有实体*SEU*和实体*Nanjing*，则关系抽取的任务是从文本中抽取关系*isLocatedIn*。

### 知识问答(Question Answering)

给定一个有自然语言给出的问题，知识问答的任务是通过知识图谱中的一个或多个事实(fact)，给出正确的答案。

## 6 总结

本文首先通过介绍一系列经典模型，向读者展示了表示学习的主要思想，接着指出表示学习发展的三个新方向：融合实体类型，文本描述信息，复杂关系路径，并通过典型模型向读者介绍其主要想法，以使其了解表示学习领域发展新方向。

## References

- [1] 刘知远, 孙茂松, 林衍恺, 谢若冰, “知识表示学习研究进展,” 计算机研究与发展, vol. 53, no. 2, p. 247/261, 2016.
- [2] A. Bordes, N. Usunier, J. Weston, and O. Yakhnenko, “Translating Embeddings for Modeling Multi-Relational Data,” *Adv. NIPS*, vol. 26, pp. 2787–2795, 2013.
- [3] J. Turian, L. Ratinov, Y. Bengio, and J. Turian, “Word Representations: A Simple and General Method for Semi-supervised Learning,” in *ACL*, 2010, no. July, pp. 384–394.
- [4] Z. Wang, J. Zhang, J. Feng, and Z. Chen, “Knowledge Graph Embedding by Translating on Hyperplanes,” *AAAI Conf. Artif. Intell.*, pp. 1112–1119, 2014.
- [5] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu, “Learning entity and relation embeddings for knowledge graph completion,” *AAAI*, pp. 2181–2187, 2015.
- [6] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, “Knowledge Graph Embedding via Dynamic Mapping Matrix,” *Proc. 53rd Annu. Meet. Assoc. Comput. Linguist. 7th Int. Jt. Conf. Nat. Lang. Process. (Volume 1 Long Pap.)*, pp. 687–696, 2015.
- [7] S. He, K. Liu, G. Ji, and J. Zhao, “Learning to Represent Knowledge Graphs with Gaussian Embedding,” in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management - CIKM '15*, 2015, pp. 623–632.
- [8] G. Ji, K. Liu, S. He, and J. Zhao, “Knowledge Graph Completion with Adaptive Sparse Transfer Matrix,” *Proc. 30th Conf. Artif. Intell. (AAAI 2016)*, pp. 985–991, 2016.
- [9] H. Xiao, M. Huang, Y. Hao, and X. Zhu, “TransA: An Adaptive Approach for Knowledge Graph Embedding,” *Aaai*, 2015.
- [10] H. Xiao, M. Huang, Y. Hao, and X. Zhu, “TransG : A Generative Mixture Model for Knowledge Graph Embedding,” *aclweb.org*, 2015.
- [11] S. Guo, Q. Wang, B. Wang, L. Wang, and L. Guo, “Semantically Smooth Knowledge Graph Embedding,” *Proc. ACL-15*, pp. 84–94, 2015.
- [12] R. Xie, Z. Liu, and M. Sun, “Representation Learning of Knowledge Graphs with Hierarchical Types,” in *IJCAI*, 2016.
- [13] Y. Lin, Z. Liu, H. Luan, M. Sun, S. Rao, and S. Liu, “Modeling Relation Paths for Representation Learning of Knowledge Bases,” *ijcai*, 2015.
- [14] R. Xie, Z. Liu, J. Jia, H. Luan, and M. Sun, “Representation Learning of Knowledge Graphs with Entity Descriptions,” *AAAI*, 2016.