基于知识图谱嵌入的药物重定位研究

经验分享

卢艳峰

中国药科大学理学院

前沿讲座课，2024 年 5 月 15 日

个人介绍

本科年级	17 级信管
本科成绩	4/115
硕士专业	21 级生物与医药（医药大数据与人工智能）
硕士导师	侯凤贞
研究方向	知识图谱嵌入和大语言模型在医药领域的应用
荣誉奖项	自 2019 年连续 5 年获得校一等奖学金
工作	大模型工程师
爱好	玩游戏、看动漫、写博客
邮箱	luyanfeng_nlp@qq.com
博客	https://www.luyf-lemon-love.space/

1 课题背景
- 疾病带给人类的挑战
- 药物重定位
- 知识图谱
- 知识图谱嵌入

2 研究现状

3 知识图谱嵌入工具包

4 数据处理

5 药物重定位

6 读研和求职的体会

疾病带给人类的挑战

- 2020 年版《中国药典》包含 5911 种药物 [1]：2711 种中药、2712 种化学药物和 153 种生物制品。

疾病带给人类的挑战

- 2020 年版《中国药典》包含 5911 种药物 [1]：2711 种中药、2712 种化学药物和 153 种生物制品。
- 人类仍面临着一些疾病挑战，如艾滋病、抑郁症和帕金森病等，它们尚未被完全攻克。

疾病带给人类的挑战

- 2020 年版《中国药典》包含 5911 种药物 [1]：2711 种中药、2712 种化学药物和 153 种生物制品。
- 人类仍面临着一些疾病挑战，如艾滋病、抑郁症和帕金森病等，它们尚未被完全攻克。
- 除此之外，人类也有应对突发性传染病和治疗罕见病的需求。

疾病带给人类的挑战

- 2020 年版《中国药典》包含 5911 种药物 [1]：2711 种中药、2712 种化学药物和 153 种生物制品。
- 人类仍面临着一些疾病挑战，如艾滋病、抑郁症和帕金森病等，它们尚未被完全攻克。
- 除此之外，人类也有应对突发性传染病和治疗罕见病的需求。
- 在当前的新药研发领域，开发一种全新的药物是一个既耗时又耗资巨大的过程。

药物重定位

- 药物重定位主要是对已获得批准并广泛用于临床治疗的药物，进行新的适应症或新用途的探索。

药物重定位

- 药物重定位主要是对已获得批准并广泛用于临床治疗的药物，进行新的适应症或新用途的探索。
- 药物重定位的主要优势在于其较低的研发成本和时间效率。

药物重定位

- 药物重定位主要是对已获得批准并广泛用于临床治疗的药物，进行新的适应症或新用途的探索。
- 药物重定位的主要优势在于其较低的研发成本和时间效率。
- 在众多药物重定位方法中，知识图谱（knowledge graph, KG）已经成为实现药物重定位目标的关键技术之一。

知识图谱

- KG 是一种基于拓扑结构图存储知识的数据库。

知识图谱

- KG 是一种基于拓扑结构图存储知识的数据库。
- 知识中的具体事物和抽象概念在 KG 中被表示为实体，实体之间的联系被表示为关系，进而知识被表示成格式为（头实体，关系，尾实体）的三元组。

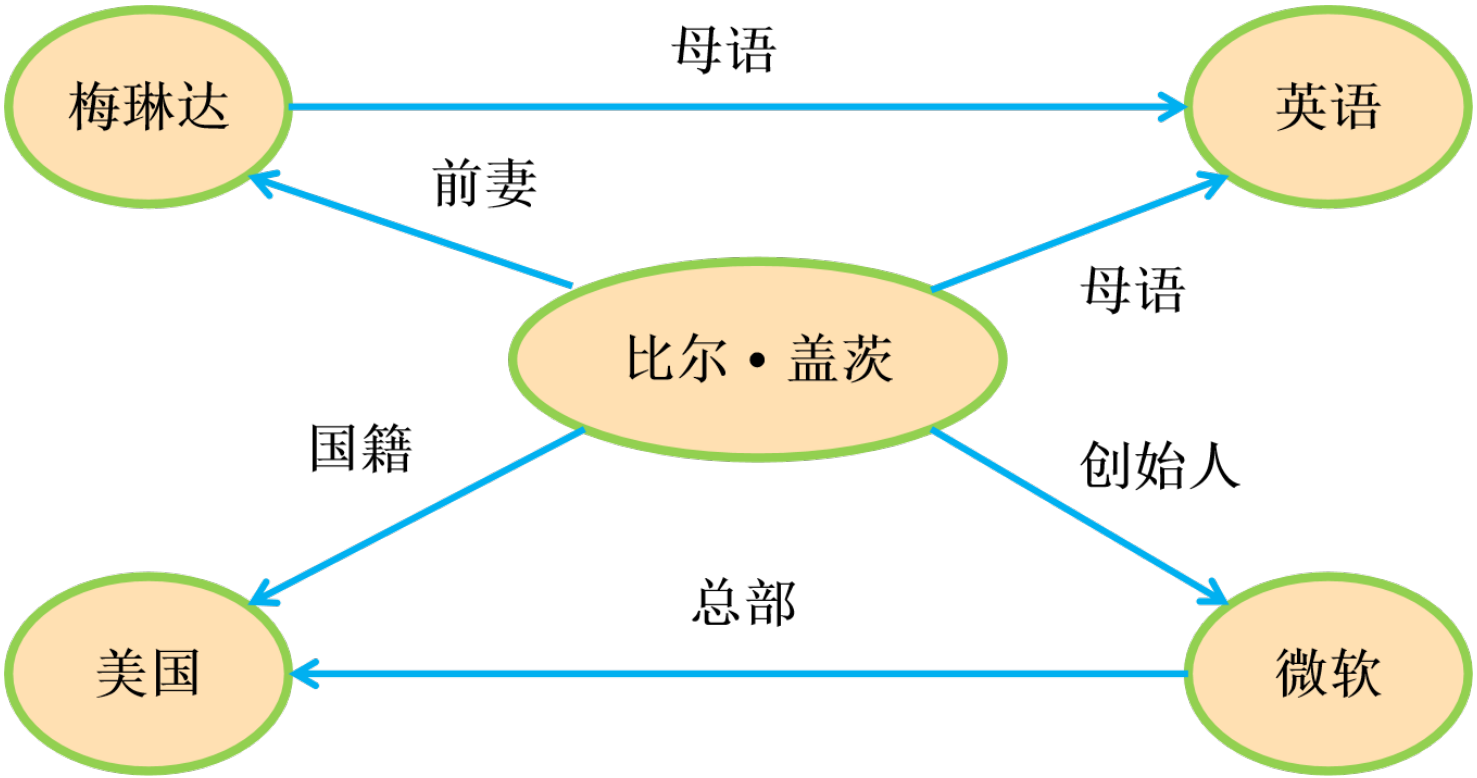
知识图谱

- KG 是一种基于拓扑结构图存储知识的数据库。
- 知识中的具体事物和抽象概念在 KG 中被表示为实体，实体之间的联系被表示为关系，进而知识被表示成格式为（头实体，关系，尾实体）的三元组。
- （江苏省，省会，南京市）三元组表示了“江苏省的省会是南京”这一事实。

知识图谱

- KG 是一种基于拓扑结构图存储知识的数据库。
- 知识中的具体事物和抽象概念在 KG 中被表示为实体，实体之间的联系被表示为关系，进而知识被表示成格式为（头实体，关系，尾实体）的三元组。
- （江苏省，省会，南京市）三元组表示了“江苏省的省会是南京”这一事实。
- KG 是一个由大量三元组组成的有向图结构，图中的节点表示实体，边表示实体间的关系。

知识图谱



知识图谱嵌入

- 大数据背景下，KG 的规模和复杂性也不断增长。

知识图谱嵌入

- 大数据背景下，KG 的规模和复杂性也不断增长。
- 药物重定位知识图谱（drug repurposing knowledge graph, DRKG）包含了 97,238 个实体以及 5,874,261 个三元组。

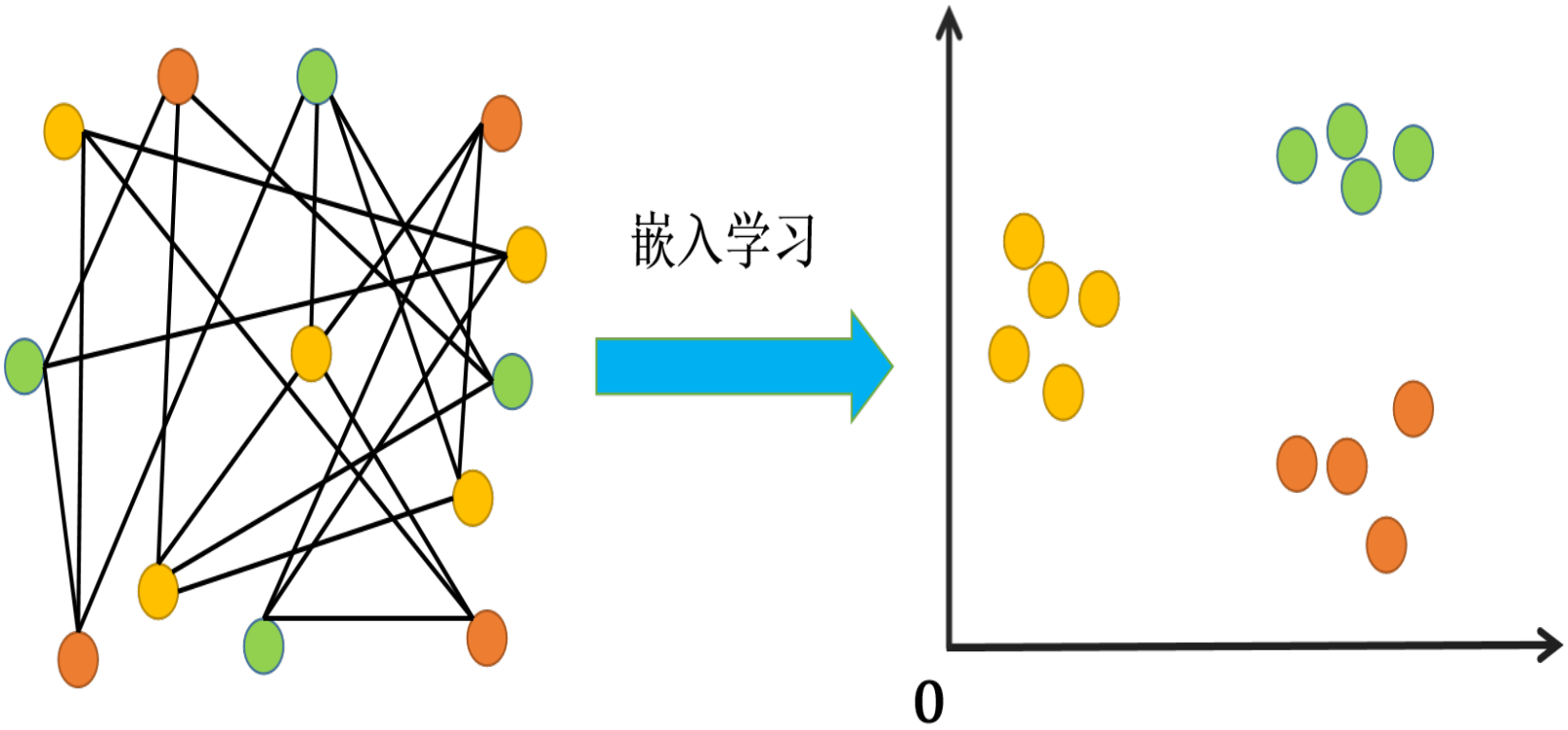
知识图谱嵌入

- 大数据背景下，KG 的规模和复杂性也不断增长。
- 药物重定位知识图谱（drug repurposing knowledge graph, DRKG）包含了 97,238 个实体以及 5,874,261 个三元组。
- 知识表示学习（knowledge representation learning，KRL）旨在通过机器学习和深度学习算法，来增强处理 KG 的能力，以及提升其在不同应用场景中的表现。

知识图谱嵌入

- 大数据背景下，KG 的规模和复杂性也不断增长。
- 药物重定位知识图谱（drug repurposing knowledge graph, DRKG）包含了 97,238 个实体以及 5,874,261 个三元组。
- 知识表示学习（knowledge representation learning，KRL）旨在通过机器学习和深度学习算法，来增强处理 KG 的能力，以及提升其在不同应用场景中的表现。
- 在近年来的 KRL 领域，知识图谱嵌入（knowledge graph embedding, KGE）技术已经迅速成为主流的研究方法。

知识图谱嵌入



- 研究内容

1 课题背景

2 研究现状
- 知识图谱嵌入模型
- 链接预测
- 药物重定位
- 基线数据集
- 知识图谱嵌入工具包
- 知识图谱嵌入模型的表现

3 知识图谱嵌入工具包

4 数据处理

5 药物重定位

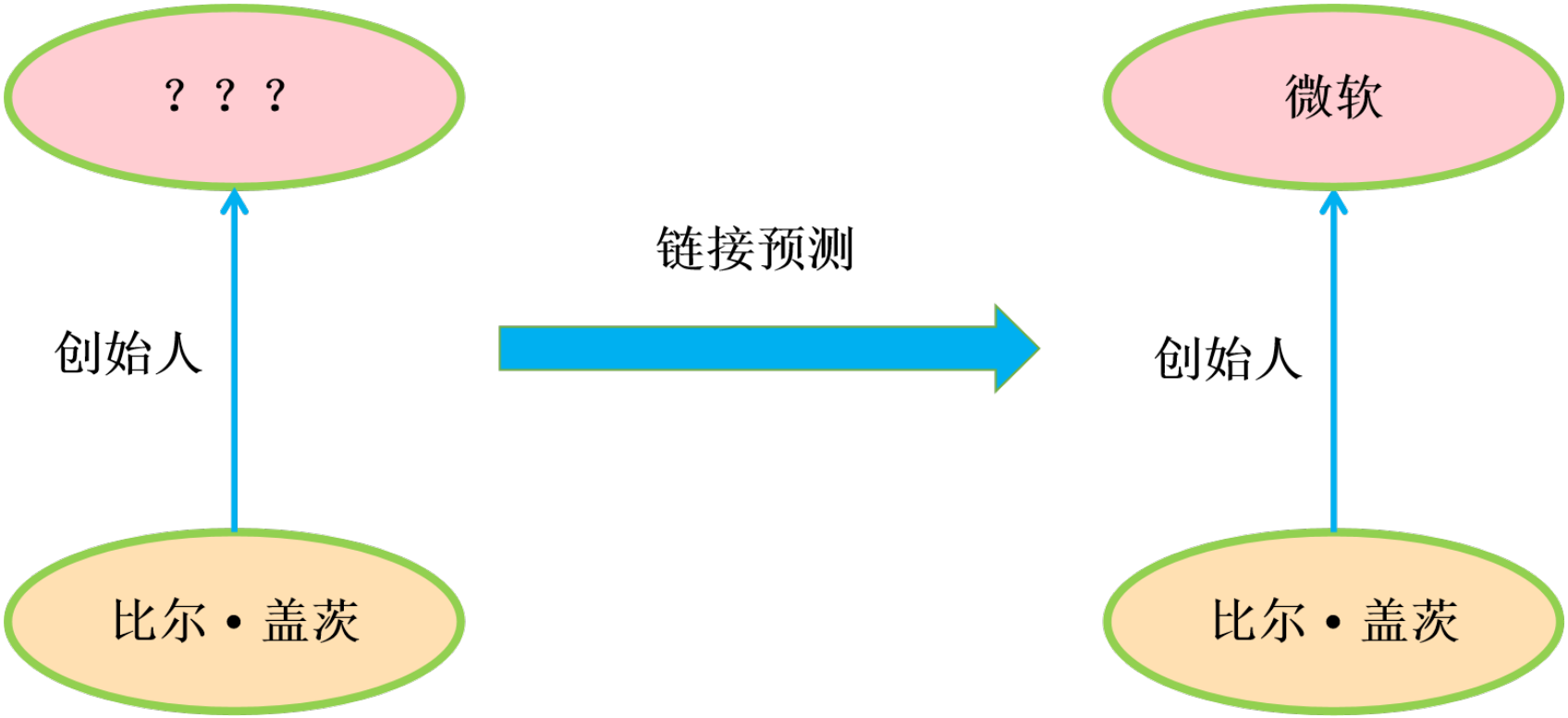
6 读研和求职的体会

知识图谱嵌入模型

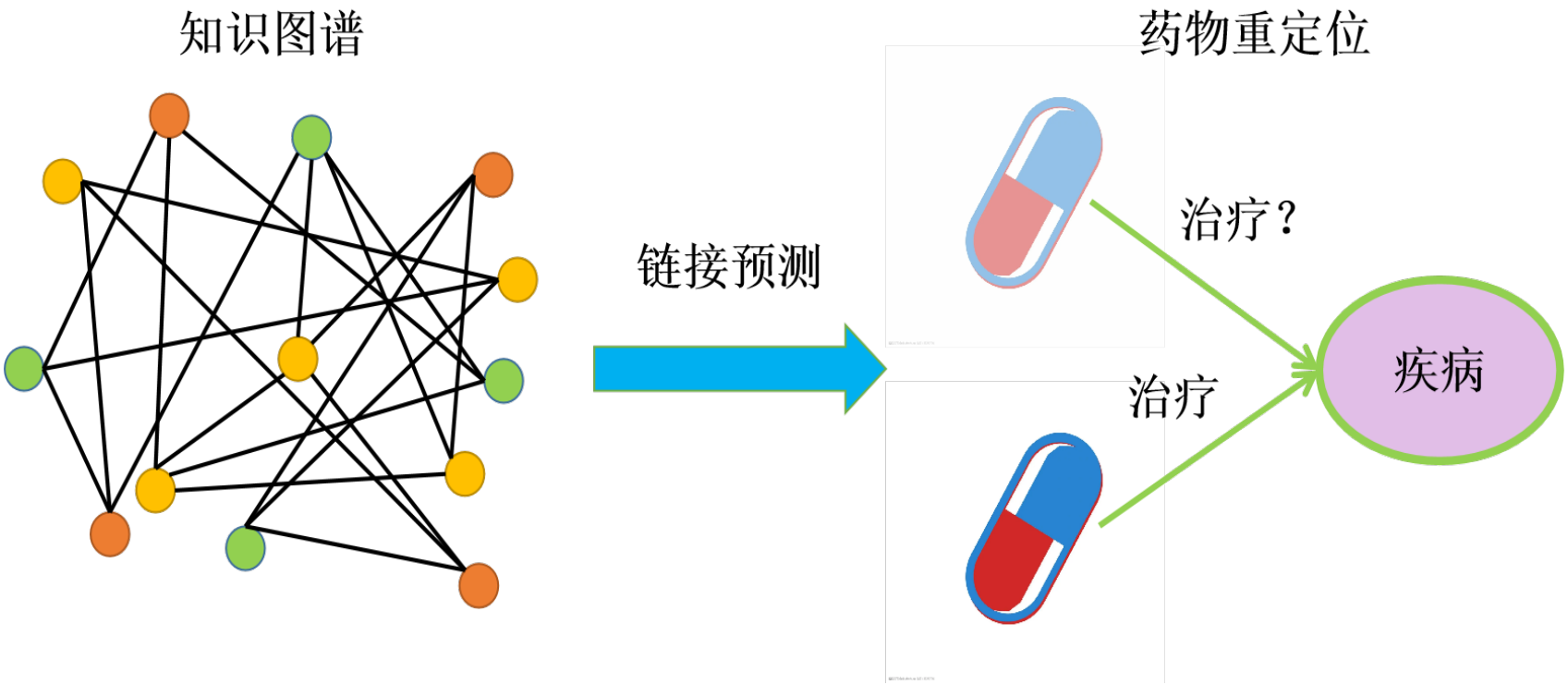
KGE 的核心在于将 KG 中的实体和关系映射到低维的连续向量空间，通过这种方式，能够实现对实体及其相互关系的高效编码。

- 平移模型：TransE[2]、TransH[3]、TransR[4]、TransD[5] 和 RotatE[6]
- 语义匹配模型：RESCAL[7]、DistMult[8]、HolE[9]、ComplEx[10]、Analogy[11] 和 SimplE[12]
- 图神经网络模型：RGCN[13] 和 CompGCN[14]

链接预测



药物重定位



基线数据集

表 1: KGE 常用基线数据集

名称	实体	关系	训练集	验证集	测试集
WN18[2]	40,943	18	141,442	5,000	5,000
WN18RR[15]	40,943	11	86,835	3,034	3,134
FB15K[2]	14,951	1,345	483,142	50,000	59,071
FB15k-237[16]	14,541	237	272,115	17,535	20,466

问题

- 第一个需要研究并解决的问题???

为什么需要 KGE 工具包

- 早期模型附带的源代码已经无法使用 [2, 3, 7]。

为什么需要 KGE 工具包

- 早期模型附带的源代码已经无法使用 [2, 3, 7]。
- 算法提出时的计算条件与如今不同 [2, 10]，不能武断地通过原始论文报告的指标判断模型的好坏。

为什么需要 KGE 工具包

- 早期模型附带的源代码已经无法使用 [2, 3, 7]。
- 算法提出时的计算条件与如今不同 [2, 10]，不能武断地通过原始论文报告的指标判断模型的好坏。
- 大多论文附带的源代码主要集中于评估模型在上述基线数据集的表现，经常是极其耗时的 [17]。

KGE 常用工具包

表 2: 不同 KGE 工具包的比较[1]

名称	模型	并行	超参数优化	最后一次维护日期
OpenKE[17]	11	否	否	2020 年 03 月 13 日
PyKEEN[18]	40	否	是	2024 年 03 月 06 日
DGL-KE[19]	6	是	否	2022 年 02 月 12 日
Pykg2vec[20]	25	否	是	2021 年 03 月 11 日
CogKGE[21]	16	否	否	2022 年 05 月 28 日
NeuralKG[22]	21	否	是	2022 年 12 月 08 日

[1]最后一次维护日期定义为**倒数第 3 次**在 GitHub 上提交源代码的日期，不包括 README 和文档等非源代码文件的更新（**截止至 2024 年 3 月 11 日**）。

KGE 模型的表现

表 3: 不同 KGE 模型在 FB15K-237 上的表现[2]

模型	模型提出时间	Hits@10
TransE[2]	2013	0.51
TransH[3]	2014	0.50
TransR[4]	2015	0.51
TransD[5]	2015	0.49
DistMult[8]	2015	0.48
ComplEx[10]	2016	0.40
SimplE[12]	2018	0.29
RotatE[6]	2019	0.53
RGCN[13]	2017	0.43
CompGCN[14]	2020	0.52

[2]TransR 和 TransD 的结果来自 OpenKE[17] 代码仓库发布的结果，剩余的模型的结果都来在 NeuralKG[22] 代码仓库发布的结果（**截止至 2024 年 3 月 11 日**）。所有的结果都用**四舍五入**方法保留**两位小数**。

研究内容

- 由于不同 KGE 模型的表现并没有太明显的差异，因此，需要系统地评估各个模型在相应 KG 上性能，进而选择合适的 KGE 模型用于药物重定位。
- 考虑上述工具包的缺陷，本研究基于 OpenKE[17] 开发了一个全新的 KGE 工具包，用于训练 KGE 模型进行药物重定位研究。
- 由于疾病种类众多，因此本研究选择了阿尔茨海默病（Alzheimer's disease，AD）作为药物重定位的研究对象。
- 考虑到时间关系，本次仅仅分享课题的一部分内容。

1 课题背景

2 研究现状

3 知识图谱嵌入工具包
- 工具包的优势
- 评分函数
- 平移模型
- 语义匹配模型
- 图神经网络模型
- 损失函数
- 负采样
- 评估指标

4 数据处理

5 药物重定位

6 读研和求职的体会

pybind11-OpenKE

相比于 OpenKE[17]，pybind11-OpenKE[3]主要具有以下优势:

- 实现了 RGCN[13] 和 CompGCN[14]，填补了 OpenKE 在图神经网络模型方面的空白
- 优化了跨平台兼容性，现在可以在多种操作系统上无缝运行
- 引入了自动化超参数搜索功能，简化了模型调优过程
- 支持多 GPU 并行训练，大幅提升了模型训练效率
- 集成了Weights & Biases，以便于更有效地追踪和记录实验日志
- 纠正了SimplE[12] 实现中的错误
- 对HolE[9] 进行了重构，使其与更高版本的 PyTorch 兼容
- 新增了早停止功能，以避免过拟合并节省训练时间
- 提供了直接通过pip安装的选项，简化了程序的安装过程
- 增加了文档，使得用户更容易学习和使用该工具

[3] https://pybind11-openke.readthedocs.io/zh-cn/latest/

评分函数

- KGE 模型首先会将实体和关系映射到稠密的向量空间，然后利用 KGE 模型相应的模型假设对头实体、关系和尾实体的嵌入向量进行运算，最终得到 KGE 模型对待评估三元组的得分，这个得分能够度量三元组的成立的概率。
- 可以把 KGE 模型看作一个对三元组进行评分的函数，这个评分函数一般能够清晰地表达了 KGE 模型的结构和数学假设。

TransE

受 word2vec[23] 学习到的词向量平移现象的启发，TransE[2] 提出了将关系的嵌入向量 $\mathbf{r}$ 建模为头实体嵌入向量 $\mathbf{h}$ 到尾实体嵌入向量 $\mathbf{t}$ 的平移的假设，即当三元组（h，r，t）成立时，TransE 假设 $\mathbf{h}+\mathbf{r} \simeq \mathbf{t}$，尾实体的嵌入向量 $\mathbf{t}$ 应该在向量空间中与头实体和关系的嵌入向量的和 $\mathbf{h}+\mathbf{r}$ 距离最近，如果三元组不成立，两者的距离应该足够远。式(1)表示了 TransE 的评分函数：

$$f(\mathbf{h}, \mathbf{r}, \mathbf{t}) = \| \mathbf{h}+\mathbf{r}-\mathbf{t} \|_{\ell_1/\ell_2} \tag{1}$$

TransE 假设实体嵌入向量和关系嵌入向量同属于一个向量空间，并且同一实体作为头实体和尾实体时，嵌入向量是相同的。

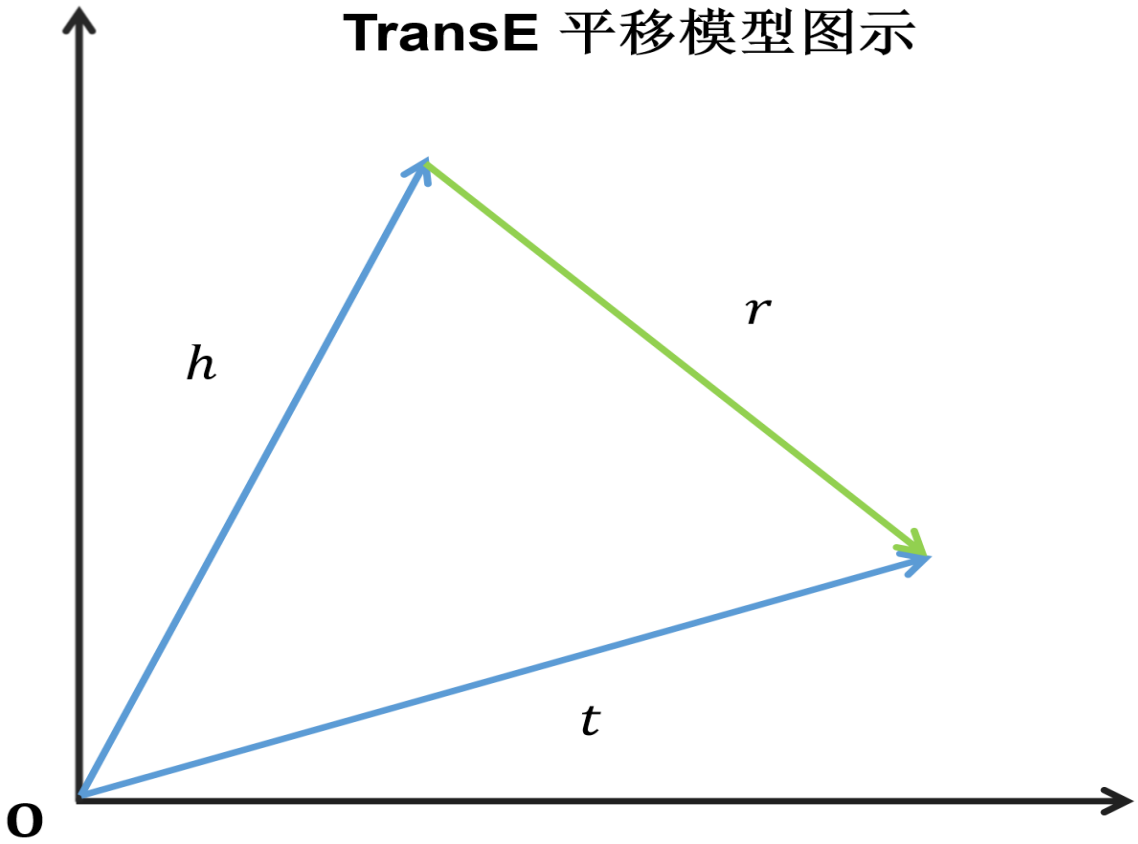
TransE

TransE 使用 ℓ_1 范数和 ℓ_2 范数作为距离函数度量两个向量的远近，因此，式(1)计算的分值越小表示三元组成立的概率越大。因此，需要将其转换为分值越大三元组成立概率越大。一般有 2 种转换方式，TransE 转换后的评分函数如式(2)和(3)所示：

$$f(\mathbf{h},\mathbf{r},\mathbf{t}) = -\parallel \mathbf{h}+\mathbf{r}-\mathbf{t} \parallel_{\ell_1/\ell_2} \tag{2}$$

$$f(\mathbf{h},\mathbf{r},\mathbf{t}) = \gamma-\parallel \mathbf{h}+\mathbf{r}-\mathbf{t} \parallel_{\ell_1/\ell_2} \tag{3}$$

TransE



TransH

TransH[3] 发表于 2014 年，是一个将关系建模为实体低维向量在超平面上的平移操作的模型。式(4)表示了 TransH 的评分函数：

$$f(\mathbf{h},\mathbf{r},\mathbf{t}) = \|(\mathbf{h} - \mathbf{r_w}^\top \mathbf{h}\mathbf{r_w}) + \mathbf{r_d} - (\mathbf{t} - \mathbf{r_w}^\top \mathbf{t}\mathbf{r_w})\|_{\ell_1/\ell_2} \qquad (4)$$

每一个关系 $\mathbf{r}$ 被 2 个向量表示：超平面的的法向量 $\mathbf{r_w}$ 和超平面上的平移向量 $\mathbf{r_d}$。

TransR

TransR[4] 发表于 2015 年，是一个为实体和关系嵌入向量分别构建了独立的向量空间，将实体向量投影到特定的关系向量空间进行平移操作的模型。式(5)表示了 TransR 的评分函数：

$$f(\mathbf{h},\mathbf{r},\mathbf{t}) = \|\mathbf{h}\mathbf{M}_\mathbf{r} + \mathbf{r} - \mathbf{t}\mathbf{M}_\mathbf{r}\|_{\ell_1/\ell_2} \tag{5}$$

为了避免过拟合，实体和关系的嵌入向量初始化为 TransE 的结果，关系矩阵 $\mathbf{M}_\mathbf{r}$ 初始为单位矩阵。实体和关系嵌入向量的维度不需要相同。

TransD

TransD[5] 发表于 2015 年，是 TransR 的改进版，为实体和关系分别定义了两个向量。第一个向量表示实体或关系的意义；另一个向量（投影向量）表示如何将实体嵌入向量投影到关系向量空间，投影向量被用来构建映射矩阵。因此，每个实体-关系对有独一无二的映射矩阵。式(6)表示了 TransD 的评分函数：

$$f(\mathbf{h},\mathbf{r},\mathbf{t}) = \|(\mathbf{r}_p\mathbf{h}_p^T + \mathbf{I})\mathbf{h} + \mathbf{r} - (\mathbf{r}_p\mathbf{t}_p^T + \mathbf{I})\mathbf{t}\|_{\ell_1/\ell_2} \tag{6}$$

为了加速收敛和避免过拟合，实体和关系的嵌入向量初始化为 TransE 的结果。实体和关系嵌入向量的维度不需要相同。

RotatE

RotatE[6] 发表于 2019 年，受欧拉恒等式 $e^{i\theta} = \cos\theta + i\sin\theta$ 的启发，发现复数与复指数 $e^{i\theta}$ 相乘相当于复数对应的向量旋转 θ 角度，$\theta > 0$ 表示逆时针旋转，$\theta < 0$ 表示顺时针旋转。类似 TransE 的模型假设，RotatE 将关系嵌入向量建模为复数向量空间内的头实体嵌入向量到尾实体嵌入向量的旋转。式(7)表示了 RotatE 的评分函数：

$$f(\mathbf{h}, \mathbf{r}, \mathbf{t}) = \gamma - \|\mathbf{h} \circ \mathbf{r} - \mathbf{t}\|_{\ell_2} \tag{7}$$

γ 是一个间隔超参数，$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{C}^n$ 是复数向量，$\circ$ 表示哈达玛积。理论和实验证明，RotatE 能够很好建模对称关系、非对称关系、逆关系和组合关系这 4 种关系，因此，RotatE 也是第一个成功同时建模上面 4 种关系的模型。

RESCAL

RESCAL[7] 发表于 2011 年，为了充分地建模 KG，为每一个关系都创建了一个关系矩阵，每一个关系矩阵分别表示 KG 中该关系对应的三元组集合中头实体和尾实体的交互。式(8)表示了 RESCAL 的评分函数：

$$f(\mathbf{h}, \mathbf{r}, \mathbf{t}) = \mathbf{h}^{\top} \mathbf{M}_r \mathbf{t} \tag{8}$$

RESCAL 也是一个较早的将头实体嵌入向量和尾实体嵌入向量进行统一的模型，即同一个实体，头实体和尾实体的嵌入向量应该相同。

DistMult

DistMult[8] 发表于 2015 年，通过直接将 RESCAL 的关系矩阵替换为对角矩阵，大幅度地降低模型的复杂度，达到与 TransE 相似的时间复杂度和空间复杂度，因此，可以很容易地扩展到大型的 KG 上。式(9)表示了 DistMult 的评分函数：

$$f(\mathbf{h},\mathbf{r},\mathbf{t}) =< \mathbf{h},\mathbf{r},\mathbf{t} >= \sum_{i=1}^{n} \mathbf{h}_i\mathbf{r}_i\mathbf{t}_i \tag{9}$$

DistMult 由于将关系矩阵变为对角矩阵，导致其只能建模对称关系，因此，这也后续语义匹配模型对 DistMult 的改进方向。

HolE

HolE[9] 发表于 2016 年，全息嵌入（HolE）利用循环相关算子来计算实体和关系之间的交互。通过快速傅里叶变换实现循环相关算子，可以加速计算，进而评分函数可以表示为如下形式：

$$f(\mathbf{h},\mathbf{r},\mathbf{t}) = \mathbf{r}^{\top}(\mathcal{F}^{-1}(\overline{\mathcal{F}(\mathbf{h})} \circ \mathcal{F}(\mathbf{t}))) \tag{10}$$

其中 $\mathcal{F}(\cdot)$ 和 $\mathcal{F}^{-1}(\cdot)$ 分别表示快速傅里叶变换及其逆变换，$\overline{\mathbf{x}}$ 表示复数共轭，$\circ$ 表示哈达玛积。

HolE 虽然通过循环相关算子大大地压缩了 RESCAL 的参数量和计算复杂度，但由于快速傅里叶变换的时间复杂度是准线性的 O(NlogN)，相比于 TransE 和 DistMult 的线性时间复杂度，还是有一定的差距。

ComplEx

ComplEx 于 2016 年发表，通过将实体和关系嵌入向量引入到复数域，使得向量间的乘法能够很好地建模非对称关系和逆关系，大大地提升了 DistMult 的表现力。式(11)表示了 ComplEx 的评分函数：

$$
\begin{aligned}
f(\mathbf{h},\mathbf{r},\mathbf{t}) &= \mathrm{Re}(<\mathbf{h},\mathbf{r},\bar{\mathbf{t}}>) \\
&= <\mathrm{Re}(\mathbf{h}),\mathrm{Re}(\mathbf{r}),\mathrm{Re}(\mathbf{t})> \\
&\quad + <\mathrm{Re}(\mathbf{h}),\mathrm{Im}(\mathbf{r}),\mathrm{Im}(\mathbf{t})> \\
&\quad + <\mathrm{Im}(\mathbf{h}),\mathrm{Re}(\mathbf{r}),\mathrm{Im}(\mathbf{t})> \\
&\quad - <\mathrm{Im}(\mathbf{h}),\mathrm{Im}(\mathbf{r}),\mathrm{Re}(\mathbf{t})>
\end{aligned}
\tag{11}
$$

已经有研究表明 HolE 其实是 ComplEx 的一个特例，它们的评分函数是相似的。不过 ComplEx 是线性时间复杂度，因此，从消耗的时间角度，ComplEx 可能是更优的。

ANALOGY

ANALOGY[11] 发表于 2017 年，通过显式地建模实体和关系嵌入向量的类比性质，统一了 DistMult、HolE 和 ComplEx。经过复杂的推导，发现 ANALOGY 的评分函数其实是 DistMult 和 ComplEx 的评分函数的和：

$$f(\mathbf{h},\mathbf{r},\mathbf{t}) = < \mathbf{h_d},\mathbf{r_d},\mathbf{t_d} > + \mathrm{Re}(< \mathbf{h_c},\mathbf{r_c},\overline{\mathbf{t_c}} >) \tag{12}$$

$\mathbf{h_d},\mathbf{r_d},\mathbf{t_d}$ 是 DistMult 部分对应的头实体、关系和尾实体的嵌入向量，$\mathbf{h_c},\mathbf{r_c},\mathbf{t_c}$ 是 ComplEx 部分对应的头实体、关系和尾实体的嵌入向量。

SimplE

SimplE[12] 发表于 2018 年，它为每一实体定义了两个嵌入向量：$\mathbf{h_e}$ 和 $\mathbf{t_e}$ 分别表示该实体在头实体和尾实体时的嵌入向量；也为每一个关系定义两个嵌入向量：$\mathbf{r}$ 和 $\mathbf{r^{-1}}$ 分别表示该关系和其逆关系的嵌入向量。其评分函数为式(13)：

$$f(\mathbf{h},\mathbf{r},\mathbf{t}) = \frac{1}{2}(< \mathbf{h_h},\mathbf{r},\mathbf{t_t} > + < \mathbf{h_t},\mathbf{r^{-1}},\mathbf{t_h} >) \tag{13}$$

SimplE 与上面介绍过的平移模型和语义匹配模型都不同，它并没有统一头实体和尾实体的嵌入向量，即当一个实体处在头实体和尾实体时，对应的嵌入向量是不同。相比于 DistMult，SimplE 虽然能够显式地建模非对称关系，但是会使得实体的嵌入向量不能得到充分训练数据，因此，需要显式地建模逆关系来弥补这一缺陷。

RGCN

RGCN[13] 发表于 2017 年，本质是一个编码器。在链接预测时，RGCN 将会生成实体的潜在特征表示；然后利用 DistMult 生成三元组的得分。可以从这里获得更详细的信息。

CompGCN

CompGCN[14] 发表于 2020 年，这是一种在图卷积网络中整合多关系信息的新框架，它利用知识图谱嵌入技术中的各种组合操作，将实体和关系共同嵌入到图中。可以从这里获得更详细的信息。

MarginLoss

MarginLoss 来自于 TransE，也是 TransH、TransR、TransD、DistMult 和 HolE 原始实现中使用的损失函数。该损失函数假设正负三元组的分值存在一个间隔。式(14)表示 MarginLoss 计算公式：

$$\mathcal{L} = \sum_{i=1}^{n} \frac{1}{n} \sum_{(h,r,t)\in T} \sum_{(h',r,t')\in T'} [\gamma + f(\mathbf{h},\mathbf{r},\mathbf{t}) - f(\mathbf{h}',\mathbf{r},\mathbf{t}')]_+ \qquad (14)$$

pybind11-OpenKE 遵循 TransE 的最初的设置，正三元组的分值小于负样本的分值。

SoftplusLoss

SoftplusLoss 来自于 ComplEx，也是 ANALOGY 和 SimplE 的损失函数。SoftplusLoss 是一种对数似然的损失函数，该损失函数不需要间隔参数，减少了需要调整的超参数，进而能够在一定程度提高了 KGE 模型的表现。式(15)表示 SoftplusLoss 计算公式：

$$\begin{aligned}\mathcal{L} = & \frac{1}{2}\left(\sum_{(h,r,t)\in\mathbf{T}} \log(1+\exp(-f(\mathbf{h},\mathbf{r},\mathbf{t}))) \right. \\ & \left. + \sum_{i=1}^{n}\frac{1}{n}\sum_{(h',r,t')\in\mathbf{T}'} \log(1+\exp(f(\mathbf{h}',\mathbf{r},\mathbf{t}'))) \right)\end{aligned} \tag{15}$$

SoftplusLoss 要求正三元组分值大于负三元组分值。

SigmoidLoss

SigmoidLoss 来自于 RotatE，是一种基于 sigmoid 函数的损失函数，该损失函数也不需要间隔超参数，因此，类似 SoftplusLoss，SigmoidLoss 也能在一定程度改善 KGE 模型的表现。式(16)表示 SigmoidLoss 计算公式：

$$\begin{aligned}\mathcal{L} = \frac{1}{2}\Bigg(& \sum_{(h,r,t)\in\mathbf{T}} -\log\sigma(f(\mathbf{h},\mathbf{r},\mathbf{t})) \\ & -\sum_{i=1}^{n}\frac{1}{n}\sum_{(h',r,t')\in\mathbf{T}'}\log\sigma(-f(\mathbf{h}',\mathbf{r},\mathbf{t}'))\Bigg)\end{aligned} \tag{16}$$

在式(16)中，σ 是 sigmoid 函数。SigmoidLoss 也要求正三元组分值应该大于负三元组分值。

RGCNLoss

RGCNLoss 来自于 RGCN，是一种基于二元交叉熵的损失函数。不同于 MarginLoss，该损失函数也要求正三元组分值应该大于负三元组分值。式(17)表示 RGCNLoss 计算公式：

$$\mathcal{L} = -\frac{\mathbf{1}}{(n+\mathbf{1})} \sum_{(h,r,t)\in \mathbf{T}\cup\mathbf{T}'} y \log \sigma(f(\mathbf{h},\mathbf{r},\mathbf{t})) + (\mathbf{1}-y)\log(\mathbf{1}-\sigma(f(\mathbf{h},\mathbf{r},\mathbf{t}))) \tag{17}$$

在式(17)中，σ 表示 sigmoid 函数，y 是三元组的标签，1 表示正三元组，0 表示负三元组。

CompGCNLoss

CompGCNLoss 来自于 CompGCN，是带有标签平滑的标准二元交叉熵损失函数。首先，需要对三元组的标签 y_o 进行平滑，计算公式如式(18)所示：

$$y = (1 - m)y_o + \frac{1}{|\mathbf{E}|} \tag{18}$$

在式(18)中，m 是平滑率，在 CompGCN 附带的源代码中默认为 0.1，$|\mathbf{E}|$ 表示 KG 实体的总数。在对正三元组和负三元组的标签平滑后，损失函数计算如式(19)所示：

$$\begin{aligned}\mathcal{L} = -\frac{\mathbf{1}}{(n+\mathbf{1})} \sum_{(h,r,t)\in\mathbf{T}\cup\mathbf{T}'} & y \log f(\mathbf{h},\mathbf{r},\mathbf{t}) \\ & + (\mathbf{1}-y)\log(\mathbf{1}-f(\mathbf{h},\mathbf{r},\mathbf{t}))\end{aligned} \tag{19}$$

UniSampler

UniSampler 称为均匀分布训练集采样器，用于平移模型和语义匹配模型。即模型训练过程中，通过交替地替换头实体和尾实体来构建负三元组。负三元组的构造方法如式(20)所示：

$$\mathbf{T}^{'} = \{(h^{'}, r, t) \mid h^{'} \in \mathbf{E}\} \cup \{(h, r, t^{'} \mid t^{'} \in \mathbf{E}\} \tag{20}$$

可以看出负三元组集合是由训练集中的三元组不同时随机替换头实体或尾实体得到的。除此之外，ComplEx 的原论文表明在 KGE 模型训练过程中，为每一个正三元组构造更多的负三元组能够提升模型的训练效果。

pybind11-OpenKE 保证了构造的负三元组一定没有在训练集中出现过。???

BernSampler

BernSampler 来源于 TransH。因为 KG 是稀疏的，即未被完全补全的，KG 中蕴含着很多未知的正三元组。对于 1 对多的关系，如果通过替换尾实体构造负三元组，该负三元组会比替换头实体构造出的负三元组，更容易是正三元组。下面介绍 BernSampler 负采样的步骤：

1. 对于给定关系 r 的所有三元组，首先从训练集中得到两种统计信息：计算每个头实体的尾实体的平均个数，记为 tph；计算每个尾实体的头实体的平均个数，记为 hpt。
2. 对于训练集中的三元组（h，r，t），应该以 tph/(tph+hpt) 概率替换头实体来构造负三元组；以 hpt/(tph+hpt) 概率替换尾实体来构造负三元组。

自我对抗负采样

自我对抗负采样来自于 RotatE。随着 KGE 模型训练的进行，越来越多的负三元组明显是假的，不能够为训练提供有效信息。因此，RotatE 提出了一种称为自我对抗负采样的数据增强方法，该方法能够根据当前嵌入模型对负三元组进行采样，即从式(21)表示的分布中采样负三元组：

$$p_r(h'_j, t'_j \mid \{(h, r, t)\}) = \frac{\exp af(h'_j, r, t'_j)}{\sum_i \exp af(h'_i, r, t'_i)} \tag{21}$$

式(21)中，a 表示采样温度。

自我对抗负采样

由于采样过程可能成本过高，因此，可以直接将式(21)中特定负三元组的概率用作它对损失值的贡献。以 RotatE 的 SigmoidLoss 损失函数为例，SigmoidLoss 损失函数最终的计算公式为式(22)：

$$\mathcal{L} = \frac{1}{2}\left(\sum_{(h,r,t)\in \mathbf{T}} -\log\sigma(f(\mathbf{h},\mathbf{r},\mathbf{t})) - \sum_{(h',r,t')\in \mathbf{T}'} \sum_{i=1}^{n} p_r(h_i', t_i') \log\sigma(-f(\mathbf{h}', \mathbf{r}, \mathbf{t}'))) \right) \tag{22}$$

从式(22)中可以发现自我对抗负采样其实是对负三元组的损失值进行了加权平均。实验表明自我对抗负采样能够很好提升模型的训练效果。

评估指标

- KGE 模型可以通过链接预测技术预测 KG 中缺失的三元组，即给定（h，r，?）预测缺失的尾实体 t，或者给定（?，r，t）预测缺失的头实体 h。可以通过链接预测给出正确实体的排名。
- 对于每一个测试三元组，头实体依次被实体集中的每一个实体替换。这些损坏的三元组首先被 KGE 模型计算得分，然后按照得分排序。因此得到了正确实体的排名。上述过程通过替换尾实体被重复。

评估指标

- 因为一些损坏的三元组可能已经存在 KG 中，在这种情形下，这些损坏三元组可能比测试集中的三元组排名更靠前，但不应该被认为是错误的，因为两个三元组都是正确的。
- 因此，pybind11-OpenKE 删除了那些出现在训练集、验证集或测试集中的损坏的三元组，这确保了这样的损坏三元组不会参与排名。

评估指标

常使用 3 种经典指标来评估 KGE 模型链接预测的性能：正确实体的平均排名（mean rank，MR），正确实体的平均倒数排名（mean reciprocal rank，MRR）和正确实体的前 N 的比例即前 N 命中率 Hits@N（N = 1，3，10）。具体的计算方法分别为式(23)、(24)和(25)：

$$\mathrm{MR} = \frac{1}{2 \mid \mathbf{S} \mid} \sum_{(h,r,t)\in \mathbf{S}} \mathrm{rank}_h + \mathrm{rank}_t \tag{23}$$

$$\mathrm{MRR} = \frac{1}{2 \mid \mathbf{S} \mid} \sum_{(h,r,t)\in \mathbf{S}} \frac{1}{\mathrm{rank}_h} + \frac{1}{\mathrm{rank}_t} \tag{24}$$

$$\mathrm{Hits@N} = \frac{1}{2 \mid \mathbf{S} \mid} \sum_{(h,r,t)\in \mathrm{S}} \mathrm{I}[\mathrm{rank}_h \leq N] + \mathrm{I}[\mathrm{rank}_t \leq N] \tag{25}$$

评估指标

- 对于大型的 KG，上述的评估非常耗时，因此，可以在评估时进行类型限制进而快速评估。
- 具体来说，首先统计整个 KG 中某一个关系出现过的头实体集合和尾实体集合，然后在评估时，仅仅用该关系的头实体集合中的实体替换得到正确头实体的排名，注意在替换时如果出现 KG 中已经存在的正确三元组，将其丢弃不让其参与排名。类似地，对于计算尾实体的排名，通过该关系的尾实体集合中的实体进行替换。
- 在这种设置下，计算得到的上述 3 种指标，称为 MR_TYPE、MRR_TYPE 和 Hits@N_TYPE。

评估指标

相比于 OpenKE，pybind11-OpenKE 还能够让用户自己通过列表指定 Hits@N 和 Hits@N_TYPE 的 N 值，默认值为[1，3，10]。

1 课题背景

2 研究现状

3 知识图谱嵌入工具包

4 数据处理
- 数据集
- 数据预处理

5 药物重定位

6 读研和求职的体会

数据集

- DRKG[4]是一个涉及基因、药物、疾病、生物过程、副作用和症状的综合生物 KG，包括来自 DrugBank、Hetionet、GNBR、STRING、IntAct 和 DGIdb 等 6 个现有数据库的信息，以及从 COVID-19 爆发早期发表的 COVID-19 出版物（截至 2020 年 3 月 22 日）中收集的数据（后文标记为 bioarx 数据库）。
- 它有属于 13 种实体类型的 97,238 个实体；以及属于 107 种关系类型的 5,874,261 个三元组。
- DRKG 使用“实体类型::ID”的格式表示一个实体，如“Disease::MESH:D000544”；使用“数据源名:: 关系名:: 头实体类型: 尾实体类型”的格式表示关系，如“DRUGBANK::treats::Compound:Disease”。

[4] https://github.com/gnn4dr/DRKG/

治疗关系

表 4: DRKG 的治疗关系

关系名	实际含义
DRUGBANK::treats::Compound:Disease	治疗
GNBR::T::Compound:Disease	治疗，治疗（包括检查）
Hetionet::CtD::Compound:Disease	治疗

AD 实体

表 5: DRKG 的 AD 实体

实体名	中文含义
Disease::DOID:10652	阿尔茨海默病
Disease::MESH:C536599	无神经纤维缠结的阿尔茨海默病
Disease::MESH:D000544	阿尔茨海默病

药物集合

- 选择 DRKG 中提供的药物重定位候选药物集合，作为 AD 药物重定位候选药物集合，一共8,104种药物。这些药物是 DrugBank 中 FDA 批准的药物，但是排除了分子量 <250 的药物，因为分子量 <250 的药物很多实际上是补充剂。
- 遍历 DRKG 中的 5,874,261 个三元组，寻找利用表4的 3 种关系治疗表5的 3 种 AD 实体的药物，一共找到了 126 种药物。这些药物将在 AD 药物重定位的结果中被排除。

1 课题背景

2 研究现状

3 知识图谱嵌入工具包

4 数据处理

5 药物重定位
- 候选模型
- 经典评估
- 嵌入评估
- 药物重定位的结果

6 读研和求职的体会

候选模型

考虑图神经网络的复杂度和表现，本研究没有将 RGCN 和 CompGCN 纳入到 AD 的药物重定位候选模型列表中。

KGE 模型测试集上评估结果

模型	MRR	MR	HITS @1	HITS @3	HITS @10	HITS @30	HITS @50	HITS @100	HITS @200
TransE	0.221	713.995	0.083	0.306	0.442	0.579	0.634	0.703	0.767
TransH	0.226	**704.754**	0.084	0.314	0.452	0.588	0.643	0.710	0.772
TransR	0.180	1336.794	0.077	0.240	0.333	0.449	0.507	0.588	0.675
TransD	0.218	735.788	0.086	0.297	0.430	0.566	0.621	0.691	0.757
RotatE	0.357	717.016	0.200	0.472	0.591	0.689	0.728	0.778	0.824
RESCAL	0.288	708.244	0.168	0.360	0.486	0.600	0.649	0.713	0.776
DistMult	0.258	841.95	0.180	0.293	0.399	0.500	0.549	0.623	0.707
HolE	0.350	1018.915	0.240	0.409	0.553	0.668	0.709	0.757	0.801
ComplEx	**0.397**	770.605	0.241	**0.510**	**0.647**	**0.739**	**0.770**	**0.808**	**0.843**
ANALOGY	0.339	1245.968	**0.252**	0.375	0.512	0.625	0.669	0.722	0.770
SimplE	0.311	2052.869	0.199	0.372	0.511	0.632	0.678	0.732	0.779

注：**加粗**的是最优结果，下划线的是次优结果

KGE 模型测试集上评估结果（类型限制）

模型	MRR	MR	HITS @1	HITS @3	HITS @10	HITS @30	HITS @50	HITS @100	HITS @200
TransE	0.229	408.083	0.088	0.316	0.452	0.588	0.644	0.715	0.779
TransH	0.233	413.395	0.089	0.323	0.462	0.596	0.652	0.719	0.780
TransR	0.181	572.828	0.078	0.240	0.334	0.451	0.509	0.591	0.679
TransD	0.226	439.560	0.091	0.307	0.441	0.575	0.631	0.701	0.767
RotatE	0.362	338.742	0.203	0.478	0.596	0.695	0.735	0.784	0.830
RESCAL	0.290	368.725	0.169	0.362	0.489	0.603	0.653	0.719	0.783
DistMult	0.269	378.393	0.188	0.306	0.414	0.515	0.565	0.640	0.724
HolE	0.359	364.303	0.247	0.418	0.565	0.681	0.722	0.772	0.816
ComplEx	**0.401**	**301.772**	0.244	**0.515**	**0.652**	**0.743**	**0.775**	**0.813**	**0.849**
ANALOGY	0.350	415.729	**0.261**	0.387	0.527	0.642	0.687	0.743	0.792
SimplE	0.322	433.608	0.208	0.384	0.525	0.647	0.694	0.749	0.798

注：**加粗**的是最优结果，下划线的是次优结果

平移模型嵌入评估

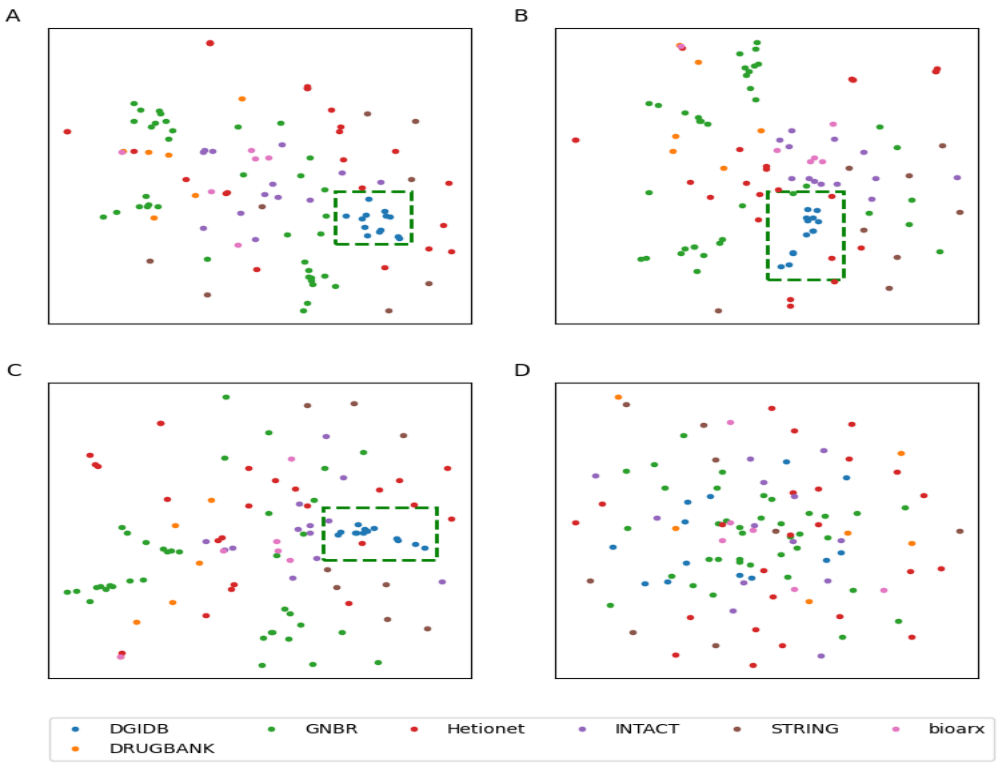


图 1: 4 种平移模型关系嵌入向量在 2D 向量空间的可视化图[5]

[5]A：TransE；B：TransH；C：TransD；D：RotatE

语义匹配模型嵌入评估

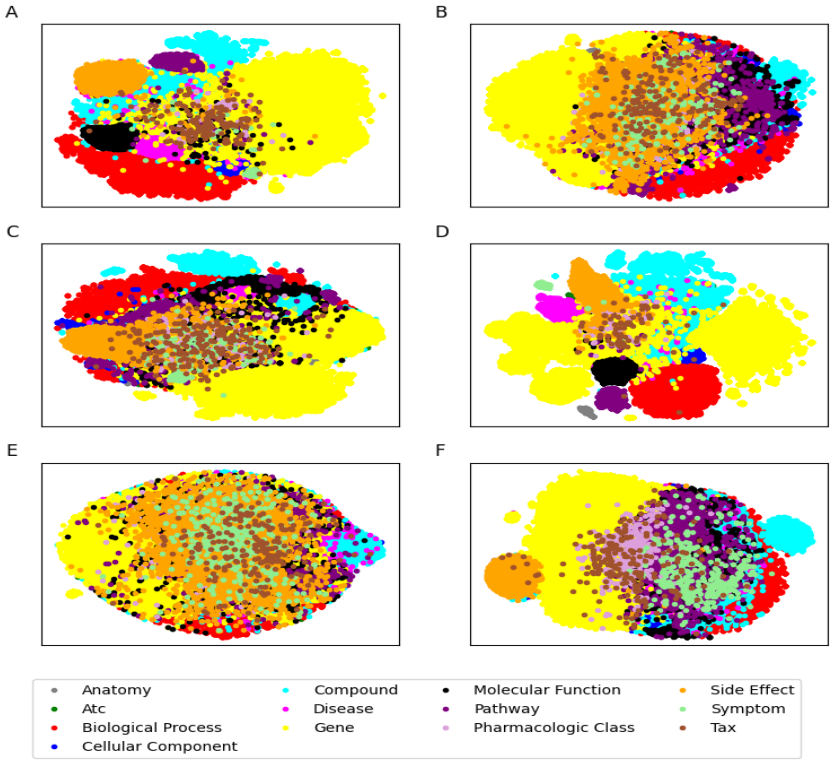


图 2: 6 种语义匹配模型实体嵌入向量在 2D 向量空间的可视化图[6]

[6]A：RESCAL；B：DistMult；C：HolE；D：ComplEx；E：ANALOGY；F：SimplE

候选药物列表大小

综合 KGE 的经典评估和嵌入评估结果，本研究决定使用 RotatE 和 ComplEx 分别代表平移模型和语义匹配模型作为 AD 药物重定位的最终模型。两个模型都选择前50的药物作为候选药物列表。

RotatE 药物重定位的结果

RotatE 确定了28种有可能用于 AD 治疗的药物，其中的21种药物，即脱氢表雄酮、谷胱甘肽、雌二醇、环腺苷酸、葡萄糖、顺铂、视黄酸、睾酮、辣椒素、亚麻酸、他莫昔芬、奎尼丁、槲皮素、腺苷、孕酮、紫杉醇、腺苷三磷酸、硝苯地平、大黄素、二十碳五烯酸、奥曲肽已被前人的研究证实对于 AD 有潜在的治疗作用。

ComplEx 药物重定位的结果

ComplEx 确定了10种有可能用于 AD 治疗的药物，其中的7种药物，即脱氢表雄酮、二十碳五烯酸、谷胱甘肽、托法替尼、辅酶 Q10、维生素 D3、睾酮已被前人的研究证实对于 AD 有潜在的治疗作用。

RotatE 和 ComplEx 药物重定位结果的交集

通过对 RotatE 和 ComplEx 的上述候选药物集合取交集，得到5种药物，分别是二十碳五烯酸、谷胱甘肽、胆固醇、睾酮、脱氢表雄酮，除了胆固醇外，所有的药物都已被前人的研究证实对于 AD 有潜在的治疗作用，这也说明了利用 KGE 模型进行药物重定位的正确性和有效性。

1 课题背景
2 研究现状
3 知识图谱嵌入工具包
4 数据处理
5 药物重定位
6 读研和求职的体会

读研的体会

- 读研比较累。
- 导师只会在必要的时候进行指导，毕竟雏鹰总有一天要独自翱翔。
- 读研后，薪资会比本科要高的，但是也有五分之一的人要考公务员。
- 读研主要是为了获得一种可以通过短时间看论文独立解决实际问题的能力。

读研的体会

- 想要做成一件事，一定要有信心、决心和耐心，缺一不可。
- 基础（数学和代码能力）比大部分人想的还要重要。
- 具备独立阅读的能力。
- 如何想成为一名算法工程师或者程序员，一定要对自己研究的领域或者计算机有持续的热爱。

求职的体会

拿到了 3.5 个 offer，都是大模型算法工程师。

- 4 月 2 日（星期二），导师下午修改完简历后，开始投简历。
- 4 月 4 日（星期四），清明节。
- 4 月 6 日（星期六），第一次面试。
- 4 月 17 日（星期三），最后一次面试。
- 一到周四，这一周就没面试了。总共应该就面了七八天。

求职的体会

拿到了 3.5 个 offer，都是大模型算法工程师。

- 天津：13k，14 薪，单休，领导器重，独角兽。
- 北京：20k，14 薪，双休，要加班，报销打车费，独角兽。
- 南京：17k，14 薪，双休，朝九晚六，事业单位，有人才公寓，六险一金，40 多张 A100。
- 合肥：给 offer 太晚了，当时已经签约了，联想的上市子公司，也是朝九晚六，双休，感觉应该是 15k-16k。

求职的体会

我是三无人员（无实习、无竞赛和无顶会），甚至六级也没过。

- 感觉只要不想去大厂，211 的牌子，独角兽还是随便去的。
- 二线城市一般会给 11k-13k，只有一线城市（北京、上海和深圳）才会给 15k 以上。
- 南京软件行业外包多，想留在南京是很难的。
- 对于应届生，企业还是很喜欢要的，主要喜欢看基础和潜力。
- 自信、乐观、真诚和事少是面试最有力的武器。
- 无牵无挂、想去哪就去哪，最容易找到工作，你们应该懂我的意思吧！QWQ

参考文献 I

[1] 兰奋, 洪小栩, 宋宗华, et al. 《中国药典》2020 年版基本概况和主要特点[J/OL]. 中国药品标准, 2020, 21(03): 185-188. http://dx.doi.org/10.19778/j.chp.2020.03.001.

[2] Bordes A, Usunier N, Garcia-Duran A, et al. Translating embeddings for modeling multi-relational data[J]. Advances in neural information processing systems, 2013, 26.

[3] Wang Z, Zhang J, Feng J, et al. Knowledge graph embedding by translating on hyperplanes[C] // Proceedings of the AAAI conference on artificial intelligence: Vol 28, [S.l.], 2014.

[4] Lin Y, Liu Z, Sun M, et al. Learning entity and relation embeddings for knowledge graph completion[C] // Proceedings of the AAAI conference on artificial intelligence: Vol 29, [S.l.], 2015.

参考文献 II

[5] Ji G, He S, Xu L, et al. Knowledge graph embedding via dynamic mapping matrix[C] // Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers), [S.I.], 2015: 687-696.

[6] Sun Z, Deng Z-H, Nie J-Y, et al. Rotate: Knowledge graph embedding by relational rotation in complex space[J]. arXiv preprint arXiv:1902.10197, 2019.

[7] Nickel M, Tresp V, Kriegel H-P, et al. A three-way model for collective learning on multi-relational data.[C] // Icml: Vol 11, [S.I.], 2011: 3104482-3104584.

[8] Yang B, Yih W-t, He X, et al. Embedding entities and relations for learning and inference in knowledge bases[J]. arXiv preprint arXiv:1412.6575, 2014.

参考文献 III

[9] Nickel M, Rosasco L, Poggio T. Holographic embeddings of knowledge graphs[C] // Proceedings of the AAAI conference on artificial intelligence : Vol 30, [S.l.], 2016.

[10] Trouillon T, Welbl J, Riedel S, et al. Complex embeddings for simple link prediction[C] // International conference on machine learning, [S.l.], 2016 : 2071-2080.

[11] Liu H, Wu Y, Yang Y. Analogical inference for multi-relational embeddings[C] // International conference on machine learning, [S.l.], 2017 : 2168-2178.

[12] Kazemi S M, Poole D. Simple embedding for link prediction in knowledge graphs[J]. Advances in neural information processing systems, 2018, 31.

参考文献 IV

[13] Schlichtkrull M, Kipf T N, Bloem P, et al. Modeling relational data with graph convolutional networks[C] // The semantic web: 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, proceedings 15, [S.l.], 2018 : 593-607.

[14] Vashishth S, Sanyal S, Nitin V, et al. Composition-based multi-relational graph convolutional networks[J]. arXiv preprint arXiv:1911.03082, 2019.

[15] Convolutional 2D Knowledge Graph Embeddings, Dettmers T, Minervini P, Stenetorp P, et al. .

[16] Toutanova K, Chen D. Observed versus latent features for knowledge base and text inference[C] // Proceedings of the 3rd workshop on continuous vector space models and their compositionality, [S.l.], 2015 : 57-66.

参考文献 V

[17] Han X, Cao S, Lv X, et al. Openke: An open toolkit for knowledge embedding[C] // Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations, [S.l.], 2018 : 139-144.

[18] Ali M, Berrendorf M, Hoyt C T, et al. PyKEEN 1.0: a python library for training and evaluating knowledge graph embeddings[J]. Journal of Machine Learning Research, 2021, 22(82) : 1-6.

[19] Zheng D, Song X, Ma C, et al. Dgl-ke: Training knowledge graph embeddings at scale[C] // Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval, [S.l.], 2020 : 739-748.

参考文献 VI

[20] Yu S-Y, Chhetri S R, Canedo A, et al. Pykg2vec: A python library for knowledge graph embedding[J]. Journal of Machine Learning Research, 2021, 22(16): 1-6.

[21] Jin Z, Men T, Yuan H, et al. CogKGE: A knowledge graph embedding toolkit and benchmark for representing multi-source and heterogeneous knowledge[C] // Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, [S.l.], 2022: 166-173.

[22] Zhang W, Chen X, Yao Z, et al. NeuralKG: an open source library for diverse representation learning of knowledge graphs[C] // Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, [S.l.], 2022: 3323-3328.

参考文献 VII

[23] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality[J]. Advances in neural information processing systems, 2013, 26.

Thanks for your attention!

愿诸君在未来的日子里都能够心想事成！

www.luyf-lemon-love.space/about/presentation/240515.pdf