

Link Prediction 实验

211870287 丁旭

September 2023

1 论文摘要 abstract

知识图补全旨在解决带缺失三元组扩展 K 的问题。在本文中，我们提供了一种方法 GenKGC，该方法使用预训练的语言模型将知识图补全转换为序列到序列的生成任务。我们进一步引入了关系引导的演示和实体感知的分层解码，以实现更好的表示学习和快速推理。在三个数据集上的实验结果表明，我们的方法可以获得比基线更好或相当的性能，并且与先前使用预训练语言模型的方法相比，可以实现更快的推理速度。我们还发布了一个新的大规模中文知识图谱数据集 OpenBG500 用于研究目的

CCS 的概念计算方法 → 知识表示和推理。

关键字知识图谱补全; 一代; 变压器 ACM 参考格式:

2 introduction

知识图 (Knowledge Graphs, KGs) 将现实世界中的知识作为 < 主语、谓语、宾语 > 形式的事实三元组，缩写为 (s, p, o)，其中 s 和 o 表示实体，p 表示实体间的关系，这有利于广泛的知识密集型任务。知识图谱补全 (Knowledge graph completion, KGC) 旨在通过预测缺失三元组来补全知识图谱。在本文中，我们主要基于强大的预训练语言模型，针对 KGC 的链接预测任务。

大多数以前的 KG 补全方法，如 TransE[2]、ComplEx[11] 和 RotatE[9]，都是知识嵌入技术，它们将实体和关系嵌入到向量空间中，然后通过对这些向量利用预定义的评分函数来获得预测的三元组。最近，已经提出了一些文本编码方法 (例如 KG-BERT[14])，它们利用预训练的语言模型对三元组进

行编码，并输出每个候选对象的分数。显然，之前的大多数方法都利用了带有预定义评分函数的区分范式来进行知识嵌入。然而，这种判别策略在推理中需要对所有可能的三元组进行昂贵的评分，并且受到负抽样的不稳定性的影响。此外，那些密集的知识嵌入方法（如 TransE）忽略了实体和关系之间的细粒度交互，并且必须为大规模的现实世界知识图分配大量的内存占用。因此，为知识图补全寻找新的技术解决方案是很直观的。

在本文中，我们首先用序列到序列的生成对知识图补全建模，并提出了一种新的方法 GenKGC。具体来说，我们将实体和关系表示为输入序列，并利用预训练的语言模型来生成目标实体。遵循 GPT-3 朴素的“上下文学习”范式，其中模型可以通过连接与输入相关的选定样本来学习正确的输出答案，我们通过添加相同关系的三元组来提出关系引导演示。此外，在生成过程中，我们提出了实体感知的分层解码，以降低生成的时间复杂度。在两个数据集 WN18RR、FB15k-237 和新发布的大规模中文 KG 数据集 OpenBG500 上的实验结果证明了所提出方法的有效性。我们的工作贡献如下：

- 我们将链路预测转换为序列到序列生成，并提出了 GenKGC，在保持性能的同时减少了推理时间。
- 我们提出了关系导向的演示和实体感知的分层解码，可以更好地表示实体和关系，降低生成的时间复杂度。
- 我们报告了两个数据集的结果，并发布了一个新的大规模 KG 数据集 OpenBG500，用于研究目的。

3 问题描述

知识图谱将现实世界中的知识形式化为三元组格式：<subject, predicate, object>，即 <s, p, o>。其中 s 和 o 表示实体，p 表示实体间的关系。知识图谱补齐（KGC）是解决补齐知识图谱中缺失三元组的问题。很多先前知识图谱补齐的方法，例如：TransE、ComplEx、RotatE，都是将实体和关系嵌入到向量空间，并通过在向量上使用预定义的打分函数来获得预测的三元组。近来也出现了一些文本编码方法（如 KG-BERT），利用预训练语言模型编码三元组，再为每个候选输出分数。但是该方法需要在推理时为所有可能的三元组打分，因此代价很昂贵，并存在负样本导致的不稳定性问题。这些密集知识嵌入方法都忽略了实体和关系之间的细粒度交互，并不得

不为大规模知识图谱分配庞大的内存，因此很有必要为 KGC 提出新的解决方法。

本文提出了一种新方法-GenKGC,即将知识图谱补齐任务转换为 seq2seq 的生成任务。具体来说，就是将实体和关系作为输入序列，利用预训练语言模型生成目标实体。为了更好的表示学习和快速推理，本文进一步引入了关系引导示例和实体感知层次解码两个策略。并且，在三个数据集上的实验结果表明，本文的方法能获得比之前使用预训练语言模型更好的性能。为了研究需求，本文也发布了一个新的大规模中文知识图谱数据集 OpenBG500。

4 输入、输出、模型算法描述（附框架图；有多个的挑 1 个主要实现）

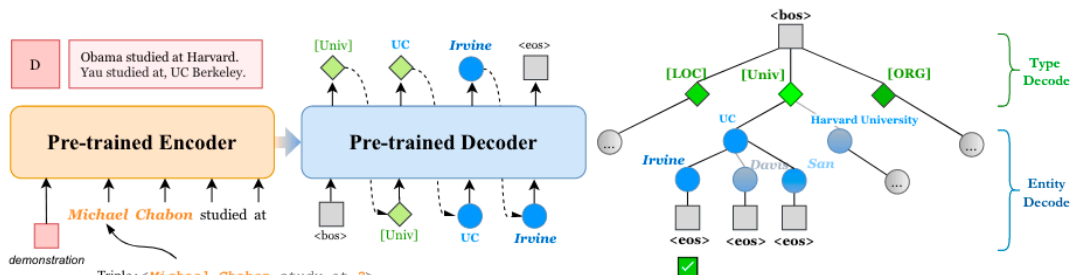


Figure 1: Architecture of GenKGC. We augment the input text of entity and relation with demonstrations, and introduce entity-aware hierarchical decoding for fast inference.

图 1:

本论文主要研究了将知识图谱补全任务从判别式方法转化为生成式方法。给定一个不完整的知识图谱三元组集合，任务是预测缺失的实体或关系。

输入：是一个不完整的知识图谱三元组集合

输出：是对缺失实体或关系的生成预测。

模型采用了基于生成式 Transformer 的方法来完成知识图谱补全任务。Link Prediction as Seq2Seq Generation——将链接预测任务形式化为 seq2seq 的生成任务

知识图谱由实体类别和实体描述定义为：其中是实体集，是关系集，是

三元组集，是实体类别，是实体描述。对于每一个三元组，分别表示头、尾实体。同时，对于每个实体都存在对应的实体类别和文本描述。实体链接将补齐知识图谱里缺失三元组当作：在给定头实体和关系的情况下，预测尾实体。

而本文将实体和关系都视为 tokens 序列，就是用纯文本代替嵌入来表示实体，以此弥补知识图谱中三元组和预训练语言模型之间的差距。具体来说就是，对于给定的缺失尾实体的三元组，先分别获得，然后将它们拼接起来。如图 1 所示，将实体描述”Michael Chabon” 和关系描述”studies at” 拼接成”Michael Chabon studies at” 作为输入序列，而输出的目标实体序列就是”UC, Irvine”。

Relation-guided Demonstration 关系引导示例

-tuning 启发，本文在编码器侧使用关系引导示例，以关系为引导，从训练集中采样一些示例，这些示例包含相同的关系。最后输入序列可以形式化为：

$$x = \langle \text{bos} \rangle \text{demonstration}(r_j) \langle \text{sep} \rangle d_{e_i} d_{r_j} \langle \text{sep} \rangle$$

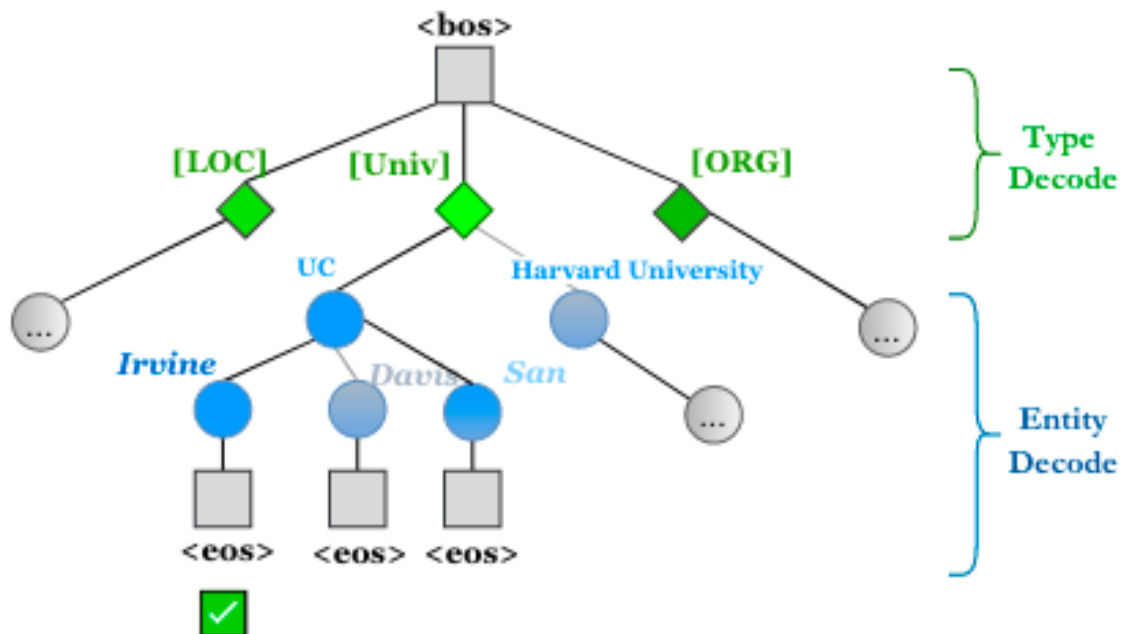
Table 1: Inference and training efficiency comparison. $|d|$ is the length of the entity description. $|\mathcal{E}|$ is the numbers of all unique entities in the KG. k is the negative samples for KG-BERT and the beam size for our GenKGC . The time under RTX 3090 is used to estimate the speed of training and inference given a single (entity,relation) pair on OpenBG500.

For One Triple	Method	Complexity	Time under RTX 3090
Training	TransE	$O(k + 1)$	0.08ms
	KG-BERT	$O(d ^2 \times (k + 1))$	72ms
	GenKGC	$O(d ^2)$	2.35ms
Inference	TransE	$O(\mathcal{E})$	0.02s
	KG-BERT	$O(d ^2 \times \mathcal{E})$	10100s
	GenKGC	$O(d ^2 \times d ^k)$	0.96s

在平凡解码设定下，需要遍历中的所有实体，并通过打分函数来排序。但这是非常耗时的，如图 1 所示。而本文的方法使用波束搜索，从中取得 top-k 个实体。具体来说，对于，GenKGC 通过以下公式计算每个的分数并排序：

$$p_{\theta}(y \mid x) = \prod_{i=1}^{|c|} p_{\theta}(z_i \mid z_{<i}, x) \prod_{i=|c|+1}^N p_{\theta}(y_i \mid y_{<i}, x), \quad (1)$$

z 是类别的个 token 集, y 是文本表示的 N 个 token 集。



KG 中蕴含着丰富的语义信息, 例如实体类型, 直觉是对解码进行约束能加速推理。本文在预训练语言模型词汇表中添加一些特殊 token 来表示类型, 以此来约束解码。为确保生成的实体在实体候选集中, 本文构建了一颗前缀树, 来解码实体名称, 具体如图 1 所示。例如: "UC" 后面只能跟随 "Irvine"、"Davis"、"San" 之一, 极大地缩小了解码遍历空间。

类似于普通的 seq2seq 模型, 本文使用标准的 seq2seq 目标函数来优化 GenKGC:

$$\mathcal{L} = -\log p_{\theta}(y \mid x) \quad (2)$$

5 评价指标及其计算公式

本论文采用了以下评价指标对知识图谱补全任务进行评估：

Hit@K:

”Hit” 表示一个样本是否被正确地命中或预测出来。”@K” 表示我们关注命中前 K 个结果的情况。

Hit@1:

在测试集上对模型进行评估时，计算每个样本是否在模型的预测结果中命中了正确的答案。如果某个样本的正确答案是模型的第一个预测结果，那么该样本被认为在”Hit@1” 下被正确预测。

Hit@3:

类似于”Hit@1”，但是在这种情况下，我们关注模型的前三个预测结果。如果某个样本的正确答案在模型的前三个预测结果中，那么该样本被认为在”Hit@3” 下被正确预测。

Hit@10:

类似于”Hit@1” 和”Hit@3”，但是在这种情况下，我们关注模型的前十个预测结果。如果某个样本的正确答案在模型的前十个预测结果中，那么该样本被认为在”Hit@10” 下被正确预测。这些指标用于衡量模型对于给定任务的准确性和预测能力。在测试集上进行评估时，通过计算在不同命中率下的准确预测数量，我们可以了解模型的性能表现。具体来说，”Hit@1”、”Hit@3” 和”Hit@10” 的值越高，表示模型的预测准确率越好。

6 对比方法及这些对比方法的引用论文出处

对比结果：

1、GenKGC 在所有数据集上都取得了比 KG-BERT 更好的结果，并保持着更快的推理速度。

2、TransE 等基于转移的方法（将实体、关系视为相同空间中的向量），虽然在一些指标上表现更好，但却面临着内存问题。在 OpenBG500 上，TransE 需要 260M 的参数来存储所有的实体和关系，而 GenKGC 使用的预训练语言模型（BART-base）只需要 110M 参数。这一问题在实体变得更多时，将更加严重。

本论文与以下对比方法进行了实验比较，对比方法的引用论文出处如下：

Table 2: Experiment results on WN18RR, FB15k-237 and OpenBG500. \diamond Resulting numbers are reported by [8], we reproduce the model result on OpenBG500 and take other results from the original papers.

Method	WN18RR			FB15k-237			OpenBG500		
	Hits@1	Hits@3	Hits@10	Hits@1	Hits@3	Hits@10	Hits@1	Hits@3	Hits@10
<i>Graph embedding approach</i>									
TransE [2] \diamond	0.043	0.441	0.532	0.198	0.376	0.441	0.207	0.340	0.513
DistMult [13] \diamond	0.412	0.470	0.504	0.199	0.301	0.446	0.049	0.088	0.216
ComplEx [11] \diamond	0.409	0.469	0.530	0.194	0.297	0.450	0.053	0.120	0.266
RotatE [9]	0.428	0.492	0.571	0.241	0.375	0.533	-	-	-
TuckER [1]	0.443	0.482	0.526	0.226	0.394	0.544	-	-	-
ATTH [4]	0.443	0.499	0.486	0.252	0.384	0.549	-	-	-
<i>Textual encoding approach</i>									
KG-BERT [14]	0.041	0.302	0.524	-	-	0.420	0.023	0.049	0.241
StAR [12]	0.243	0.491	0.709	0.205	0.322	0.482	-	-	-
GenKGC	0.287	0.403	0.535	0.192	0.355	0.439	0.203	0.280	0.351

图 2:

TransE: 一种常用的基于距离的知识图谱补全模型, 它利用欧几里得距离或曼哈顿距离来度量不同实体和关系之间的相似度。

论文: Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data.

ComplEx: 一种基于矩阵分解的知识图谱补全模型, 它使用复数向量来表示实体和关系, 并通过点积运算来计算它们之间的相似度。

论文: Trouillon, T., Welbl, J., Riedel, S., Gaussier, E., Bouchard, G. (2016). Complex embeddings for simple link prediction.

ConvE: 使用卷积神经网络 (CNN) 来学习实体和关系之间的嵌入, 然后将它们连接起来以预测缺失的三元组。

论文: Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S. (2018). Convolutional 2d knowledge graph embeddings.

RotatE: 一种利用复数旋转来计算实体和关系之间相似度的知识图谱补全模型。

论文: Sun, Z., et al. "Rotate: Knowledge graph embedding by relational rotation in complex space." ICLR 2019.

7 结果

7.1 实验结果

1. 运行代码./scripts/openbg.sh

```
root@LAPTOP-EKXMC19R: ~/PromptKG/research/GenKGC
(global seed set to 7)
tokenize the prefix tree: 0% | 0/269658 [00:00<?, 71t/s]
tokenize the prefix tree: 0% | 1/269658 [00:00<8:39:15, 8.66t/s]
tokenize the prefix tree: 100% | 269658/269658 [00:28<00:00, 9426.01t/s]
*****max output length : 128*****
total 0 cannot be tokenized
wandb: (1) Create a W&B account
wandb: (2) Use an existing W&B account
wandb: (3) Don't visualize my results
wandb: Enter your choice: 3
wandb: You chose "Don't visualize my results"
wandb: WARNING resume will be ignored since W&B syncing is set to "offline". Starting a new run with run id afqmeut.
wandb: Tracking run with wandb version 0.15.12
wandb: W&B syncing is set to "offline" in this directory.
wandb: Run "wandb online" or set "WANDB_MODE=online" to enable cloud syncing.
Using native 16bit precision.
GPU available: True, used: True
TPU available: False, using: 0 TPU cores
TPU available: False, using: 0 IPUs
Delete entities without text name.: 100% | 5000/5000 [00:00<00:00, 1118361.77it/s]
total entity not in text : 0
max number of filter entities : 18713 2182
convert text to examples: 100% | 5000/5000 [00:00<00:00, 280641.80it/s]
Delete entities without text name.: 100% | 5000/5000 [00:01<00:00, 2675.38it/s]
total entity not in text : 0
max number of filter entities : 18713 2182
convert text to examples: 100% | 5000/5000 [00:00<00:00, 239450.12it/s]
convert text to examples: 100% | 5000/5000 [00:01<00:00, 2669.81it/s]
LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES: [0]
/root/anaconda3/envs/genkgc/lib/python3.8/site-packages/torch/utils/data/dataloader.py:474: UserWarning: This DataLoader will create 32 worker processes in total. Our suggested max number of worker in current system is 16, which is smaller than what this DataLoader is going to create. Please be aware that excessive worker creation might get DataLoader running slow or even freeze, lower the worker number to avoid potential slowness/freeze if necessary.
  warnings.warn_create_warning_msg(
  Name      Type      Params
0  model     BartKGC
1  loss_fn   CrossEntropyLoss  140 M
-----
140 M      Trainable params
0          Non-trainable params
140 M      Total params
560.778    Total estimated model params size (MB)
Validation sanity check: 0% | 0/2 [00:00<?, ?it/s]
origin input : 无核白葡萄干系列
model output : 无核白葡萄干系列
model constrained output : 无核白葡萄干
无核红葡萄干
无核绿葡萄干
无核葡萄干
无核白
true label : 葡萄干
*****
(global seed set to 7)
/root/anaconda3/envs/genkgc/lib/python3.8/site-packages/pytorch_lightning/utilities/data.py:71: UserWarning: Your 'IterableDataset' has '__len__' defined. In combination with multi-process data loading (when num
```

图 3: 开始运行

2. 运行结果

```
root@LAPTOP-EKKMC19R:~/PromptKG/research/GenKGC#
Egin input : 0%无核白葡萄干系列
model output : 无核白葡萄干系列
model constrained output : 无核白葡萄干
无核红葡萄干
无核绿葡萄干
无核葡萄干
无核白
true label : 葡萄干
*****
Epoch 2: 2%
Egin input : 2%周志异细分市场
model output : 周志异细分市场
model constrained output : 交通工具
图书馆
神话传说
图
图
true label : 教材
*****
Epoch 2: 4%
Egin input : 4%纽曼 MC16 车载麦克风全民唱歌手机 k 歌麦克风唱歌吧话筒蓝牙无线车载 ktv 汽车通用车内唱歌神器直播 1 声卡 直播细分市场
model output : 纽曼 MC16 车载麦克风全民唱歌手机 k 歌麦克风唱歌吧话筒蓝牙无线车载 ktv 汽车通用车内唱歌神器直播 1 声卡 直播细分市场
model constrained output : 北京工艺美术出版社
北京工业大学出版社
北京电子工业出版社
北京大学出版社
美国国家地理
true label : 直播
*****
Epoch 2: 6%
Egin input : 6%针织套头衫服装款式细节
model output : 针织套头衫服装款式细节
model constrained output : 针织套头衫
针织套头衫
针织连衣裙
针织套头衫
针织套头衫
true label : 字母
*****
Epoch 2: 8%
Egin input : 8%牛仔裤裤门襟
model output : 牛仔裤裤门襟
model constrained output : 牛仔裤
牛仔裤
牛仔衣裤
西裤
西裤
true label : 拉链
*****
Epoch 2: 10%
Egin input : 10%t 恤服装款式
model output : t 恤服装款式
model constrained output : t 恤
t 恤
t 恤
西裤套裤
西裤套裤
西裤外套
true label : 宽松
*****
Epoch 2: 12%
Egin input : 12%女黑 t 恤长
model output : 女黑 t 恤长
model constrained output : 女黑 t 恤长
```

图 4: 运行过程

```
wandb: Run history:
wandb: Eval/hits1
wandb: Eval/hits10
wandb: Eval/hits20
wandb: Eval/hits3
wandb: epoch
wandb: trainer/global_step
wandb:
wandb: Run summary:
wandb: Eval/hits1 0.2032
wandb: Eval/hits10 0.3684
wandb: Eval/hits20 0.4844
wandb: Eval/hits3 0.2804
wandb: epoch 8
wandb: trainer/global_step 8
wandb:
wandb: You can sync this run to the cloud by running:
wandb: wandb sync /root/PromptKG/research/GenKGC/wandb/offline-run-20231013_170516-afqmwuek
wandb: Find logs at: ./wandb/offline-run-20231013_170516-afqmwuek/logs
(genkgc) root@LAPTOP-EKKMC19R:~/PromptKG/research/GenKGC#
```

图 5: 运行结束

7.2 论文

对不同的解码策略分析样例，对于没有使用层次解码的 GenKGC，本文使用常规的 beam search。从表 4 可以看出 GenKGC 生成更好的实体结果，而未使用层次解码的 GenKGC 更早地停留在正确但不准确的回答上，本文认为这是预训练语言模型自带的偏差造成的。而本文提出的实体层次编码能够约束解码过程并缓解由预训练语言模型带来的偏差。

Table 4: We list a query and first five entities with their probability predicted by GenKGC w/o entity-aware decoding, and its reranking with GenKGC.

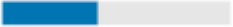
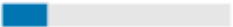

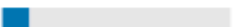
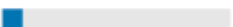

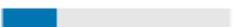



Query:(?, student, Michael Chabon)		
Rank	GenKGC w/o hierarchical decoding	Probability
1	University of California	
2	University of California, Irvine	
3	University of California, San Francisco	
4	University of California, Davis	
5	University of California, Santa Cruz	
Rank	GenKGC	Probability
1	University of California, Irvine	
2	University of California, San Francisco	
3	University of California, Davis	
4	University of California, Santa Cruz	
5	University of Calgary	

图 6: 论文结果

在本文中，GenKGC，可以在使用预训练模型进行链路预测时达到可比较的结果，同时减少了推理和训练成本。在三个基准数据集上的实验结果证明了 GenKGC 的方法的有效性，特别是在推理时间上。