

Improved Adaptive Attentional Network for Few-shot Knowledge Graph Reasoning

姓名: 肖文璇 学号: 2018202163

问题描述

提供support entity pair 和 background KG, 推断query问题: 给出头实体利用通过support pair得到的 relation信息, 得到其对应的正确尾实体, 如:

(Satya Nadella, CEO of, Microsoft)

(Tim Cook, CEO of, Apple)



(Jack Dorsey, CEO of, ?)

(Sundar Pichai, CEO of, Google)

数据集描述

Nell数据集

其中Train训练集和Test测试集的形式为, 每个关系选取前5条已知support pair:

```
"concept:leaguecoaches": [{"concept:sportsleague:mlb", "concept:leaguecoaches",
"concept:coach:joe_torre"}, [{"concept:sportsleague:mlb", "concept:leaguecoaches",
"concept:personmexico:curtis_granderson"}, [{"concept:sportsleague:mlb",
"concept:leaguecoaches", "concept:coach:seth_smith"}, [{"concept:sportsleague:mlb",
"concept:leaguecoaches", "concept:coach:john_mcgraw"}, [{"concept:sportsleague:mlb",
"concept:leaguecoaches", "concept:personmexico:jack_morris"]
```

Query形式为, 给定头实体和关系, 尾实体空缺:

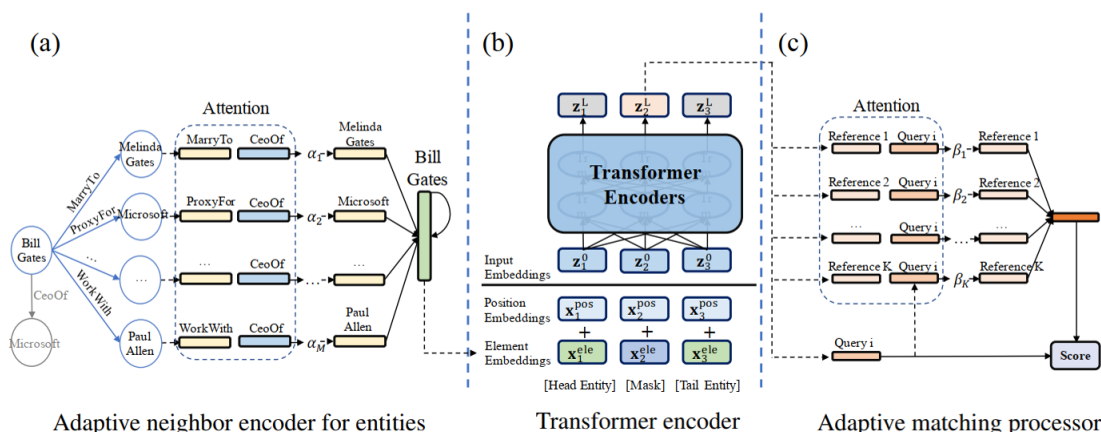
```
"concept:sportsgamesport":
[["concept:sportgame:n1973_world_series", "concept:sportsgamesport"],
["concept:sportgame:n1982_world_series", "concept:sportsgamesport"],
["concept:sportgame:n1959_world_series", "concept:sportsgamesport"],
["concept:sportgame:n1991_world_series", "concept:sportsgamesport"],
["concept:sportgame:n1968_world_series", "concept:sportsgamesport"]]
```

Candidate为猜测每个尾实体时提供的候选尾实体列表, 形式如下:

```
"concept:sportsgamesport":
["concept:sport:rafting", "concept:sport:quality_pool", "concept:sport:tennis",
"concept:sport:basket_ball", "concept:sport:management_services", "concept:sport:water_ski",
"concept:sport:skiing", "concept:sport:kickball", "concept:sport:contact_sports",
"concept:sport:snow_sports", "concept:sport:kayak", "concept:sport:fitness",
"concept:sport:table_tennis", "concept:sport:kayaking", "concept:sport:initiatives", "concept:sport:ski"]
```

模型说明

模型主要参考了 Adaptive Attentional Network for Few-Shot Knowledge Graph Completion（见参考文献），分为 Adaptive neighbor encoder，Transformer Encoder和 Matching Processor三部分。



Neighbour Encoder

Neighbour Encoder为每个Entity学习可调整的向量表示，根据不同的Query问题，识别实体在这个Query扮演的角色，以此得到不同的表示结果。具体为，先得到每个Entity的Neighbour Embedding，选择知识图谱中实体h所有邻居relation中和Query中的relation最接近的relation，分辨出实体h在query中可能扮演的角色。如果越接近，则说明这个邻居关系和我们要求的关系越相似，要越注意这个关系，具体评判两个关系相似度的函数如下：

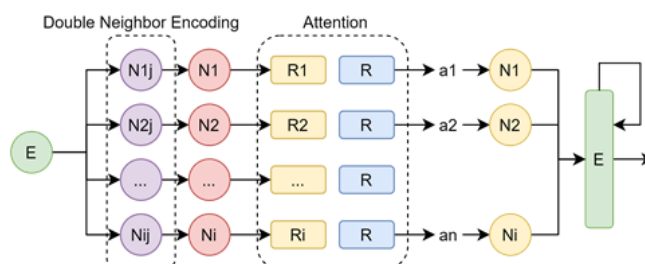
$$\psi(r, r_{nbr}) = \mathbf{r}^T \mathbf{W} \mathbf{r}_{nbr} + b$$

如果实体h的某个neighbor在该metric函数得分越高，它对h的neighbour embedding影响越大，在表示h的时候权重就越大。

最终实体h的表示方法由TransE预训练的Embedding和我们得到的Neighbour Embedding相结合，即为

$$f(h) = \sigma(\mathbf{W}_1 \mathbf{h} + \mathbf{W}_2 \mathbf{c}_{nbr})$$

在原模型的基础上，我们的改进模型添加了二级邻居的信息，进行Double Neighbor Encoding。原模型只使用了实体h一级邻居的信息，而我们加入了二级邻居的信息为h的一级邻居编码，再利用这个优化后的编码进一步得到h的Neighbor Embedding，从而能通过更多邻居信息得到更优更准确的Neighbor Embedding。

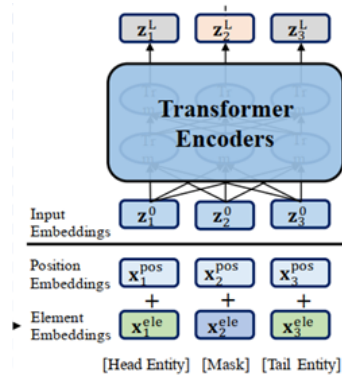


Transformer Encoder

有了每个实体h的表示，接下来Transformer Encoder用于学习实体对(h,t)之间关系r的表示，为后续预测Query答案做准备。在Transformer中，我们的输入表示为Z，Z由下面的公式得到，

$$\mathbf{z}_i^0 = \mathbf{x}_i^{\text{ele}} + \mathbf{x}_i^{\text{pos}}$$

其中 $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$ ，其中 $\mathbf{x}_1, \mathbf{x}_3$ 是头/尾实体， \mathbf{x}_2 是task relation，其中 $\mathbf{x}_i^{\text{ele}}$ 是从Neighbour Encoder得到的表示方法， $\mathbf{x}_i^{\text{pos}}$ 与 \mathbf{x}_i 在关系中的先后位置有关

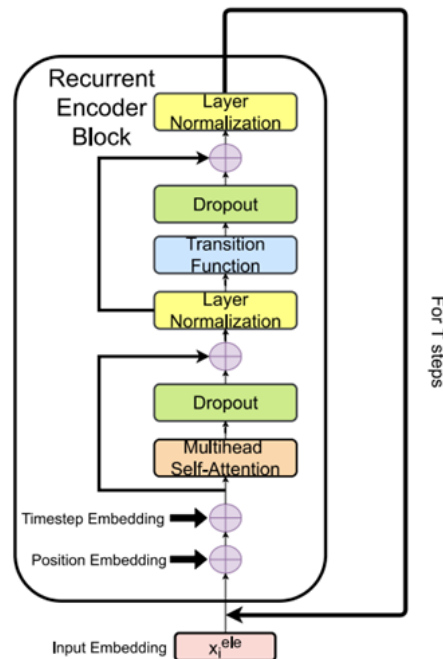


经过Transformer Encoder得到的Z表示为

$$\mathbf{z}_i^l = \text{Transformer}(\mathbf{z}_i^{l-1}), l = 1, 2, \dots, L.$$

最终得到的Z2就是训练后的实体关系的表示，这种表示能分辨出更细粒度的语义。

原模型使用了基本的Transformer模型，我们在原模型的基础上进行了Transformer的优化，使用了Universal Transformer Encoder，参考了[Universal transformers]（见参考文献）。原来的Transformer固定层数限制了其表达能力。我们引入了时间步step，实现了循环递归，除了第一次是原始信息作为输入，之后都是由前一个step的输出作为后一个的输入。为了让不固定层数的transformer适应没有见过的层数，我们使用Transition函数，即全连接神经网络，共享每层的网络权重。



此外除了Position embedding外，新引入了TimeStep embedding来建模不同层的偏置，编码公式如下：

$$P_{i,2j}^t = \sin(i/10000^{2j/d}) + \sin(t/10000^{2j/d})$$

$$P_{i,2j+1}^t = \cos(i/10000^{2j/d}) + \cos(t/10000^{2j/d}).$$

在递归环节，我们引入Adaptive Computation Time (ACT) 自适应计算时间机制。因为有些实体的语意比其他的更模糊。他们需要进行多次的计算，经过更多的层来获得一个较好的表示。我们的模型利用ACT机制，动态调整每个位置所需的计算steps。当某个位置停止计算后，直接copy它的隐状态到下一步。当所有位置都停止计算后，整个过程才停止，从而得到更优的关系表示。

Matching Processor

在Matching Processor中，我们要得到每个候选答案的得分，并选取最高分的答案作为我们的答案，即比较Query $q_r(h, r, t)$ 和我们已给的K个support集合 S_r ，如果两者越相似就越说明 t 更可能是我们问题 (h, r) 的答案。因此我们要将K个support组成的集合 S_r 合成一个表示。为了获得更好的匹配结果，模型并不是对 S_r 形成固定的一个表示，而是根据Query的要求动态调整 S_r 的表示，具体操作为先使用点积来比较我们目标query和我们给的参考三元组 S_{rk} ：

$$\delta(q_r, s_{rk}) = \mathbf{q}_r \cdot \mathbf{s}_{rk}$$

然后使用Attention机制，根据不同的query调整我们 S_r 的表示

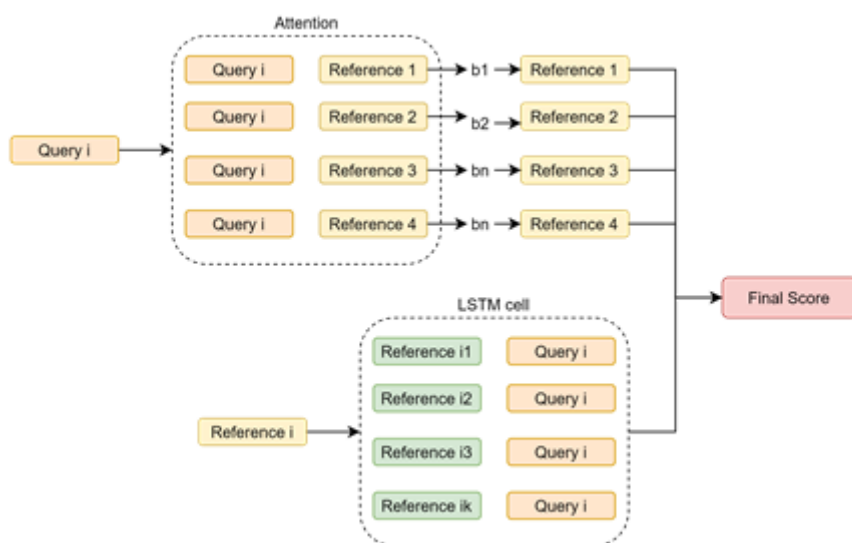
$$g(S_r) = \sum_{s_{rk} \in S_r} \beta_k \mathbf{s}_{rk}$$

$$\beta_k = \frac{\exp(\delta(q_r, s_{rk}))}{\sum_{s_{rj} \in S_r} \exp(\delta(q_r, s_{rj}))}$$

最终比较Query $q_r(h, r, t)$ 和 S_r ，相似度得分越高的qr就越可能是最后的答案：

$$\phi(q_r, S_r) = \mathbf{q}_r \cdot g(S_r).$$

我们对评分环节也进行了一定的优化，原模型计算了Query i和所有已给support的相似度，我们还添加了每个support与query的相似性评分，并将其反映到了在Matching Processor的最终评分上，通过双重比较，可能获得更优的匹配结果。



结果展示

原模型在测试集上的结果为：

HITS 10	HITS 5	HITS 1	MRR
0.428	0.364	0.200	0.279

我们的模型在原模型的基础上有略微的改进：

HITS 10	HITS 5	HITS 1	MRR
0.443	0.375	0.207	0.293

总结与反思

本次课题先复现了模型 Adaptive Attentional Network for Few-Shot Knowledge Graph Completion，并再原模型上进行了一些改进，包括为neighbor Encoder整合了二级邻居的信息获得更优的entity向量表示，通过自适应层数的Universal Transformer Encoder获得更优的entity pair关系表示，在匹配结果中，分别计算了每个support与query的相似性评分，并将其反映到了在Matching Processor的最终评分上，用以优化匹配的准确性，在最终结果上取得了一定的提升。

当然，也有一些不足之处，出于由于设备性能不足，我们在neighbor encoder取二级邻居时，没有加入所有的二级邻居，只取了部分的邻居，之后可以考虑加入完整邻居信息。

参考文献

[Sheng et al.2020] Sheng J, Guo S, Chen Z, et al. Adaptive Attentional Network for Few-Shot Knowledge Graph Completion[J]. arXiv preprint arXiv:2010.09638, 2020.

[Dehghani et al.2018] Dehghani M, Gouws S, Vinyals O, et al. Universal transformers[J]. arXiv preprint arXiv:1807.03819, 2018.