

Beijing Forest Studio  
北京理工大学信息系统及安全对抗实验中心



# 基于元学习的知识图谱补全 技术

硕士研究生 蔡成成

导师：潘丽敏

2021年06月27日

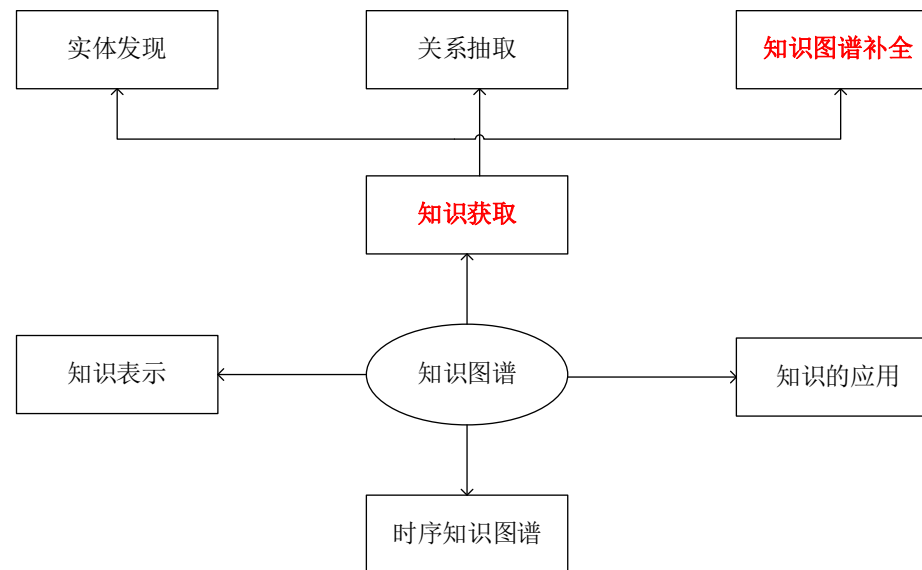
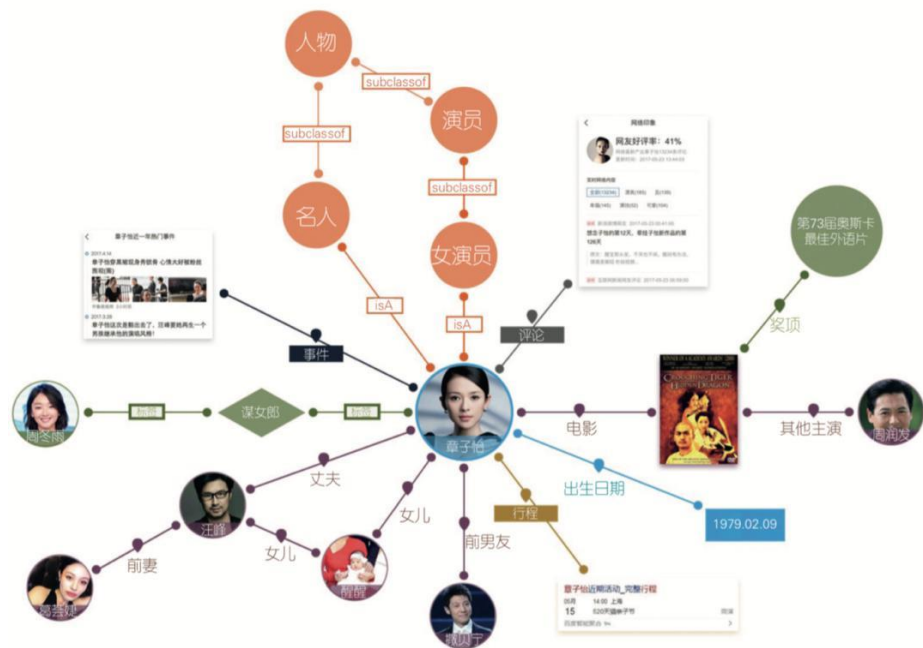
- 背景简介
- 基本概念
- 算法原理
- 应用总结
- 参考文献

- 预期收获
  - 1.了解知识图谱补全任务
  - 2.了解元学习思想及基本概念
  - 3.了解基于元学习的知识图谱补全技术



## 背景介绍

- 知识图谱 ( Knowledge Graph )
  - 本质：语义网络 ( Semantic Network ) 的**知识库**
  - 简单理解：**多关系图**
  - 应用：智能搜索、深度问答、社交网络…



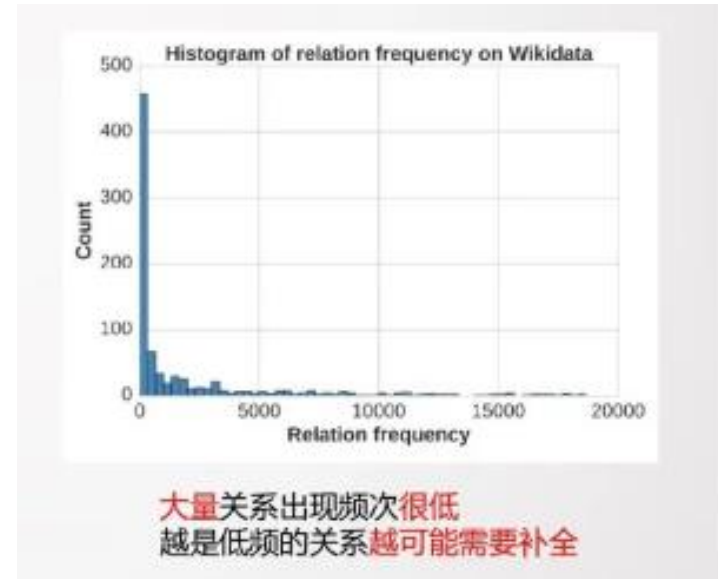
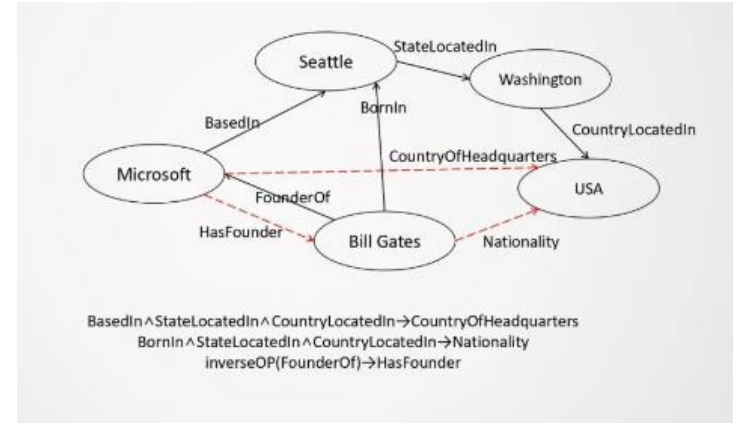
- 知识图谱补全

- 必要性

- 知识图谱普遍存在**不完备**的问题
    - **长尾数据**问题

- 难点与挑战

- 如何更好的建模知识的结构和推理规则?
    - 如何查找路径?
    - 强化学习可用吗?
    - 如何建模逻辑规则?
    - 如何解决长尾数据问题? few-shot learning?



- 知识图谱补全

- 任务定义

- 给定知识图谱  $G = \{E, R, F\}$ ，任务是预测出当前知识图谱中缺失的三元组  $F = \{(h, r, t) \mid (h, r, t) \notin F, r \in R\}$ ，即  $(h, r, ?)$  or  $(h, ?, r)$

- 两个子任务

- 封闭域的知识图谱补全：限制要补全的三元组中的实体都在E中
    - 开放域的知识图谱补全：不限制要补全的实体一定在E中
    - 后面分享中除非特别提到，否则主要指封闭域的知识图谱补全

- 知识图谱补全
  - 技术归纳
    - 基于知识表示
      - TransE、TransH、TransR
    - 基于路径查找
      - PRA ( Path Ranking Algorithm )
    - 基于推理规则
      - 推理规则与embedding结合
      - 神经网络模型与传统的推理模型结合
    - 基于强化学习
      - DeepPath
    - 基于元学习
      - 实现小样本学习

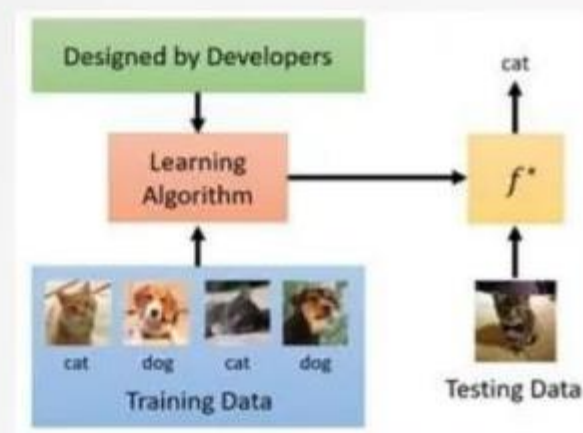




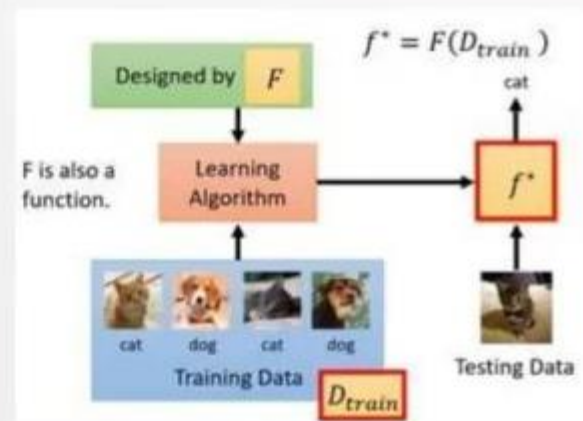
## 基本概念

- 元学习 (Meta Learning)
  - 传统的机器学习：一种“从**数据到标签**的映射”；
  - 元学习：“一种更高阶的映射”，即从**数据到函数**的映射；
  - 元学习可以用较少的数据去完成学习任务，是实现**小样本学习**的重要方法；
  - 两种方法
    - 基于度量的方法
    - 基于优化的方法

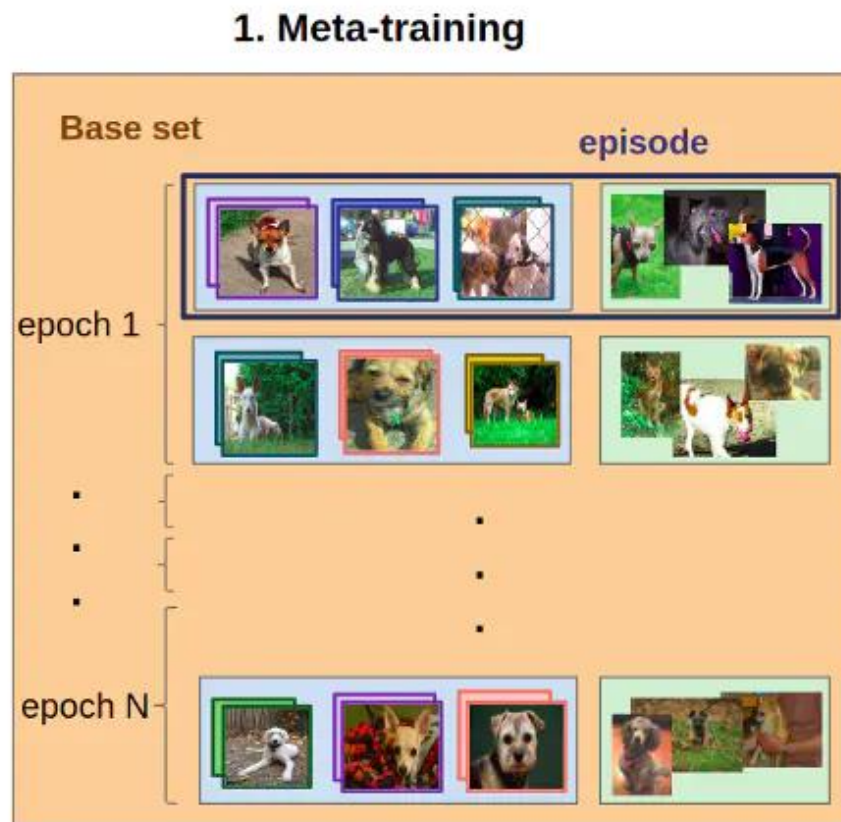
传统机器学习



元学习



- 元学习的训练
  - 以task为单位
- 基本概念
  - Episode
  - support set
  - query set
  - N-way K-shot
  - epoch
  - meta training set
  - meta testing set



- 基于元学习的知识图谱补全
  - 给定一个关系 $r$ 和他的少量正样例实体对，设计一个机器学习模型，使之可以针对每个新的头节点 $h$ ，排序所有的尾节点候选实体 $t$ ，排名第一的为真实尾节点实体
- 元学习设置
  - 任务集:  $\mathcal{T}_{train} = \{\mathcal{T}_i\}_{i=1}^M$   $\mathcal{T}_{test} = \{\mathcal{T}_j\}_{j=1}^N$
  - 每个任务的训练数据:  $D_r = \{P_r^{train}, P_r^{test}\}$ ,  $r \in \mathcal{R}$ 
    - 其中  $P_r^{train}$ :  $r$ 关系的少量正样例实体对
    - 其中  $P_r^{test}$ :  $r$ 关系的正样例和负样例实体对

Training	
Task #1 (CountryCapital)	
Support	(China, CountryCapital, Beijing)
Query	(France, CountryCapital, Paris)
Task #2 (CEOof)	
Support	(Satya Nadella, CEOof, Microsoft)
Query	(Jack Dorsey, CEOof, Twitter)
Testing	
Task #1 (OfficialLanguage)	
Support	(Japan, OfficialLanguage, Japanese)
Query	(Spain, OfficialLanguage, Spanish)

Table 1: The training and testing examples of 1-shot link prediction in KGs.

<https://blog.csdn.net/fadeG>

- 基于元学习的知识图谱补全
  - 度量指标
    - top-k hit ratio (Hits@k)
      - 正确答案预测排名不超过n的比率（越大越好）
    - mean reciprocal rank (MRR)
      - 所有正确答案的预测排名的倒数的均值t（越大越好）

- 自编码器 (Auto Encoder)

- 一种特殊的神经网络，用于**特征提取**和数据降维；
- 隐含层的映射充当编码器 $h_{\theta}$ ，输出层的映射充当解码器 $g_{\theta'}$ ，得到重构后的向量，是对输入向量的近似；
- 编码器和解码器同时训练，训练的目标是最小化重构误差，即重构向量与原始向量之间的误差最小化，目标函数为：

$$\min \frac{1}{2l} \sum_{i=1}^l \|x_i - g_{\theta'}(h_{\theta}(x_i))\|_2^2$$



## 算法原理

T	知识图谱封闭空间三元组预测( $h, r, ?$ )
I	知识图谱中三元组 (每个task下的Reference set)
P	1.关系感知的异构邻居编码 2.循环自编码 (LSTM) 聚合few-shot实体对交互 3.实体对相似性匹配
O	候选节点相似性排序

P	1.异构邻居信息和few-shot交互信息的忽略导致模型性能不佳
C	给定的知识图谱
D	1.如何提取节点异构邻居信息, 编码节点向量 2.如何提取利于few-shot实体对交互信息, 编码关系向量
L	AAAI2020



- 模型框架

- 模块一：异构邻居编码模块

- 一个关系可感知的异构邻居编码器，

给定头节点h的关系邻居集合为：

$$\mathcal{N}_h = \{(r_i, t_i) | (h, r_i, t_i) \in G'\}$$

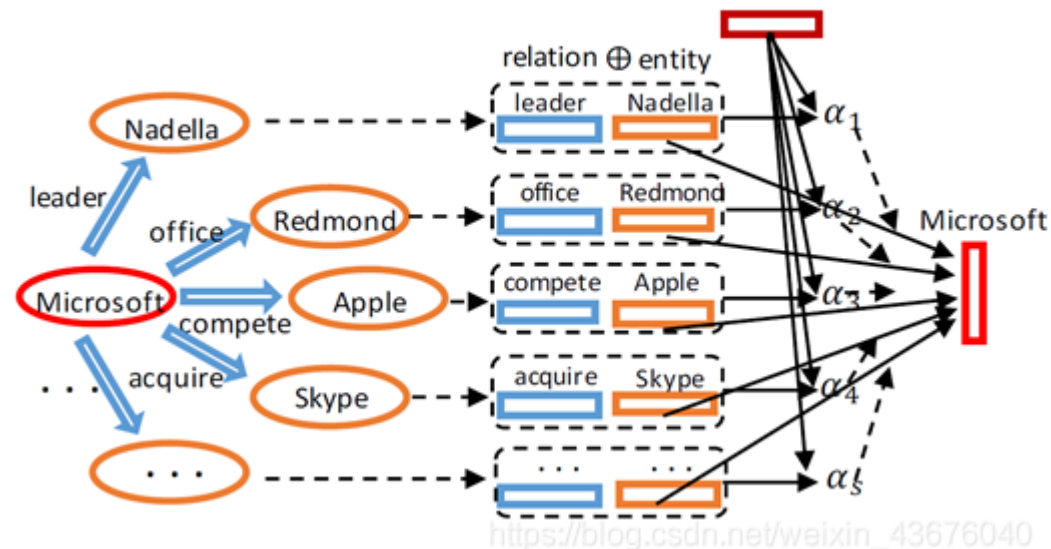
- 不同邻居（不同r）对头节点h的权重系数：

$$\alpha_i = \frac{\exp \{u_{rt}^T (\mathcal{W}_{rt}(e_{r_i} \oplus e_{t_i}) + b_{rt})\}}{\sum_j \exp \{u_{rt}^T (\mathcal{W}_{rt}(e_{r_j} \oplus e_{t_j}) + b_{rt})\}}$$

- 头节点h的异构邻居编码：

$$f_\theta(h) = \sigma \left( \sum_i \alpha_i e_{t_i} \right)$$

(b)



[https://blog.csdn.net/weixin\\_43676040](https://blog.csdn.net/weixin_43676040)

- 模型框架

- 模块二：Reference Set聚合模块

- 功能：对参考集实体对嵌入进行聚合
    - 实现：循环自编码聚合器，

$$\mathcal{E}_{h_k, t_k} = [f_{\theta}(h_k) \oplus f_{\theta}(t_k)]$$

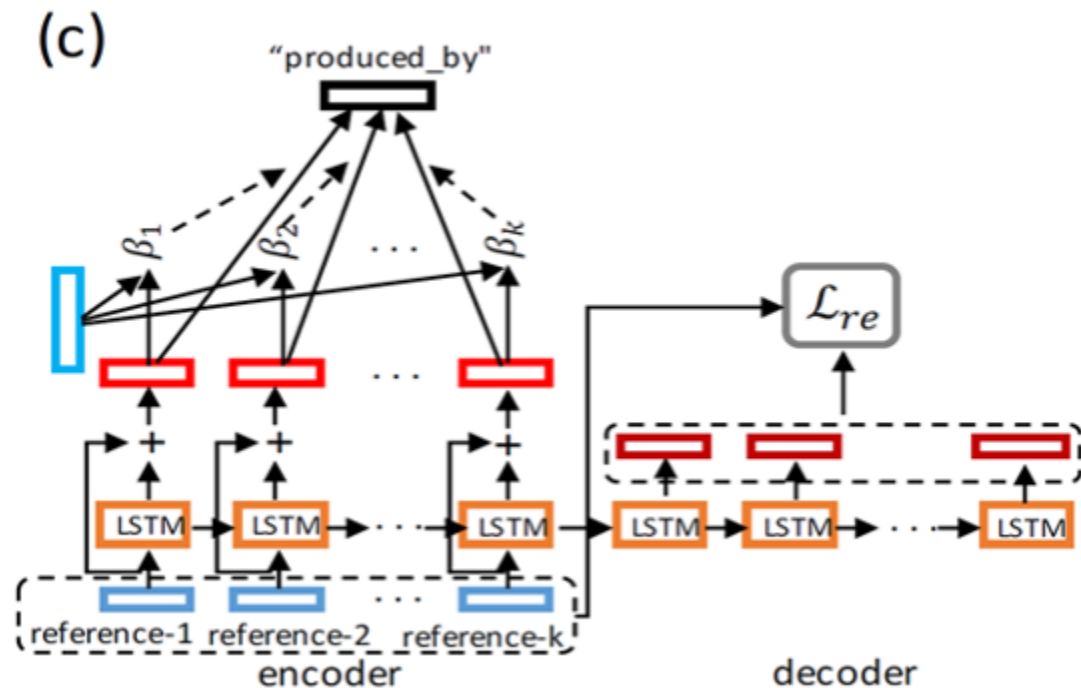
$$\mathcal{E}_{h_1, t_1} \rightarrow m_1 \rightarrow \dots \rightarrow m_K \rightarrow d_K \rightarrow \dots \rightarrow d_1$$

$$m_k = \text{RNN}_{\text{encoder}}(\mathcal{E}_{h_k, t_k}, m_{k-1})$$

$$d_{k-1} = \text{RNN}_{\text{decoder}}(d_k)$$

- 自编码聚合器的训练（损失函数）：

$$\mathcal{L}_{re}(R_r) = \sum_k \|d_k - \mathcal{E}_{h_k, t_k}\|_2^2$$

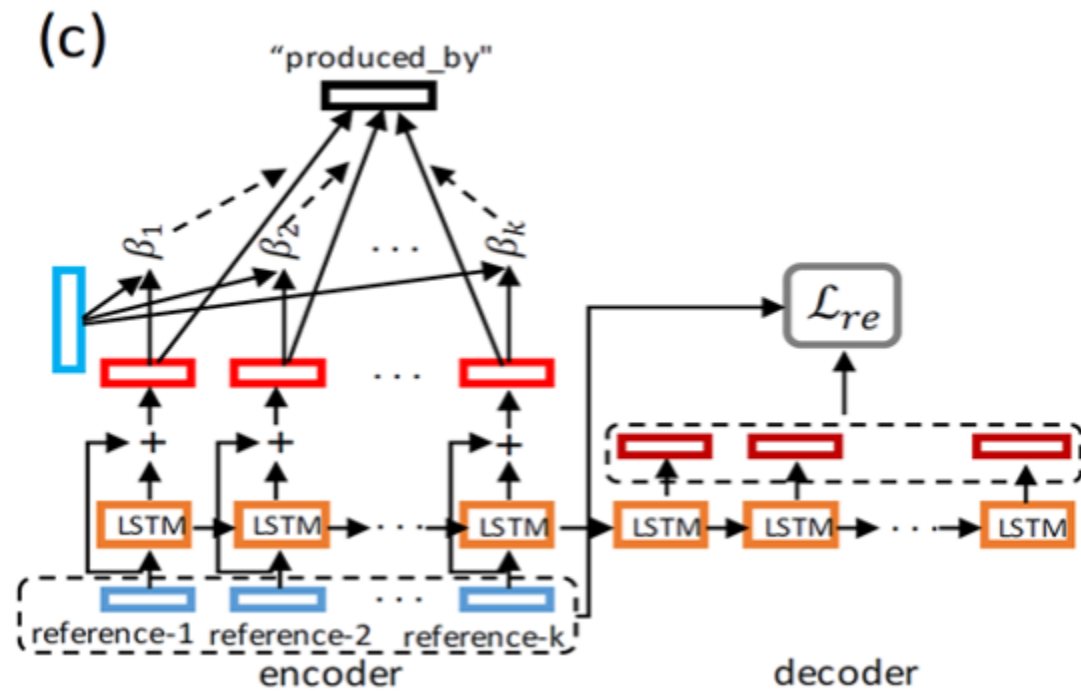


- 模型框架
  - 模块二：Reference Set聚合模块
    - 聚合结果

$$m'_k = m_k + \mathcal{E}_{h_k, t_k}$$

$$\beta_k = \frac{\exp \{ u_R^T (\mathcal{W}_R m'_k + b_R) \}}{\sum_{k'} \exp \{ u_R^T (\mathcal{W}_R m'_{k'} + b_R) \}}$$

$$f_\epsilon(R_r) = \sum_k \beta_k m'_k$$



- 模型框架

- 模块三: Query Set和Reference Set匹配模块

- 基于参考集  $R_r$  匹配每一个查询集实体对  $(h_l, t_l) \in Q_r$

$$f_\epsilon(R_r) \quad \mathcal{E}_{h_l, t_l} = [f_\theta(h_l) \oplus f_\theta(t_l)]$$

- 一个自循环处理器去执行多步匹配

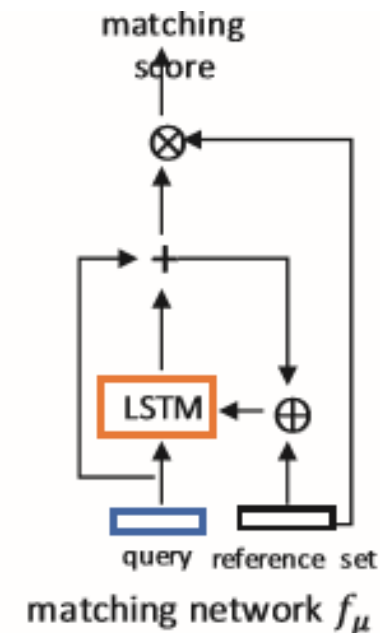
$$g'_t, c_t = \text{RNN}_{\text{match}}(\mathcal{E}_{h_l, t_l}, [g_{t-1} \oplus f_\epsilon(R_r)], c_{t-1})$$

$$g_t = g'_t + \mathcal{E}_{h_l, t_l}$$

- T步之后的隐藏层  $g_T$  作为最终的查询实体对向量

$$\mathcal{E}_{h_l, t_l} = g_T$$

- 使用  $\mathcal{E}_{h_l, t_l}$  和  $f_\epsilon(R_r)$  的**内积**作为相似性得分



- 实验结果

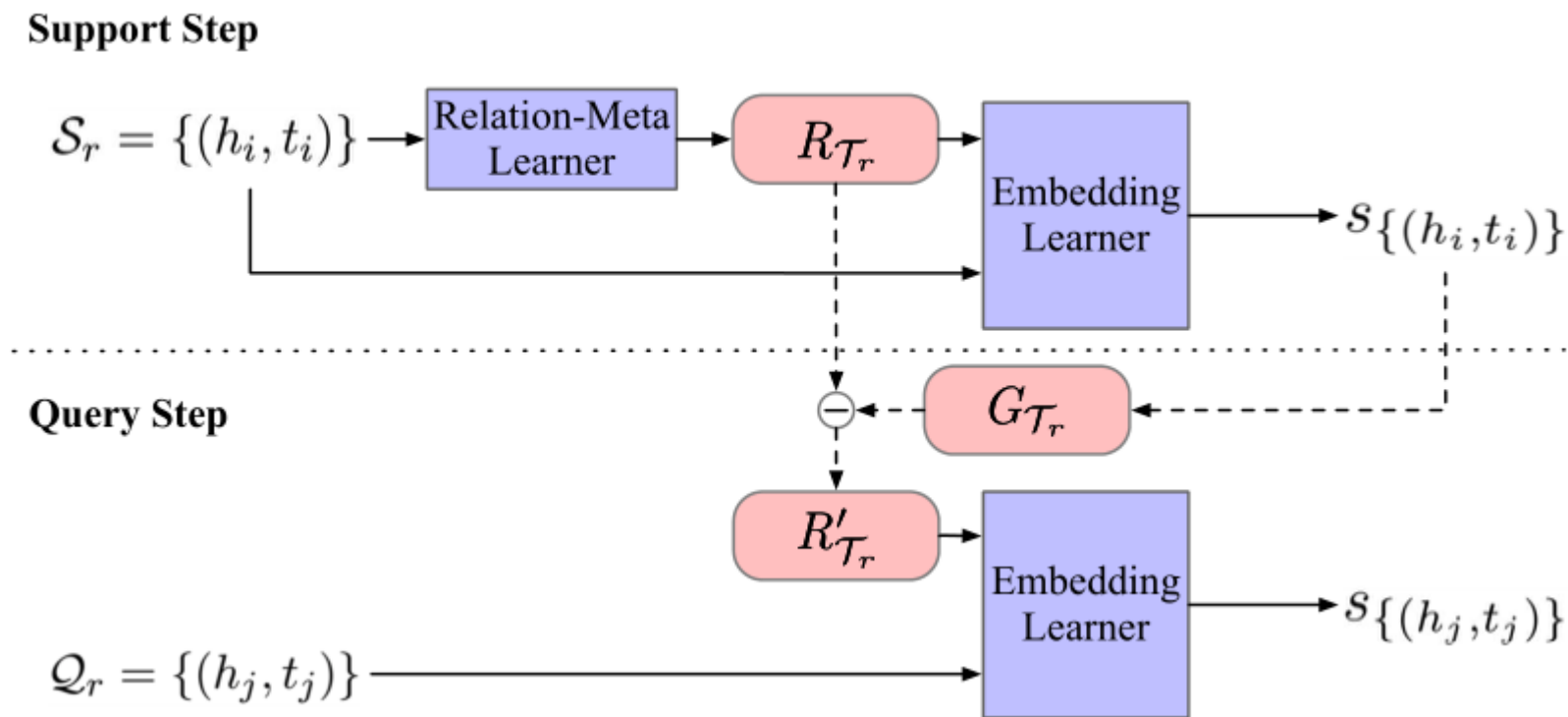
Model	Data: NELL				Data: Wiki			
	Hits@1	Hits@5	Hits@10	MRR	Hits@1	Hits@5	Hits@10	MRR
RESCAL	.069/.141	.160/.313	.204/.383	.119/.223	.259/.057	.297/.090	.309/.126	.279/.081
TransE	.056/.119	.112/.256	.189/.320	.104/.193	.186/.069	.352/.134	.431/.176	.273/.111
DistMult	.066/.164	.123/.306	.178/.375	.109/.231	.271/.069	.419/.156	.459/.195	.339/.112
ComplEx	.049/.129	.092/.223	.112/.273	.079/.185	.226/.085	.315/.117	.397/.145	.282/.106
GMatching (MaxP)	<u>.244/.198</u>	<u>.418/.370</u>	<u>.524/.464</u>	<u>.331/.279</u>	<u>.313/.095</u>	<u>.402/.235</u>	<u>.468/.324</u>	<u>.346/.171</u>
GMatching (MeanP)	<u>.257/.186</u>	<u>.455/.360</u>	<u>.542/.453</u>	<u>.341/.267</u>	<u>.290/.128</u>	<u>407/.274</u>	<u>.484/.350</u>	<u>.352/.203</u>
GMatching (Max)	<u>.179/.152</u>	<u>.391/.335</u>	<u>.476/.445</u>	<u>.273/.241</u>	<u>.279/.135</u>	<u>.396/.284</u>	<u>.477/.374</u>	<u>.342/.214</u>
FSRL (Ours)	<b>.345/.211</b>	<b>.502/.433</b>	<b>.570/.507</b>	<b>.421/.318</b>	<b>.338/.155</b>	<b>.430/.327</b>	<b>.486/.406</b>	<b>.390/.241</b>

T	知识图谱封闭空间三元组预测(h, r, ?)
I	知识图谱中三元组 (每个task中的support set)
P	1. 根据support set中的实体对生成 <b>关系元</b> ; 2. 通过实体嵌入和关系元计算support set实体对真值; 3. 计算 <b>梯度元</b> , 并更新关系元; 4. 通过实体嵌入和关系元计算query set实体对真值;
O	候选节点相似性排序

P	基于少量样本的知识图谱补全如何用support set传递最关键信息至query set
C	给定的知识图谱
D	关键信息 (关系元和梯度元) 的提取
L	EMNLP2019

- 两个思考
  - 支持集和查询集之间最可转移和最常见的信息是什么？
  - 如何通过只观察一项任务中的几个实例来加快学习速度？
- 两个思路
  - 关系
  - 梯度
- 解决方案
  - 关系元：表示支持集和查询集中共有的头实体和尾实体的关系信息。提取每种关系的关系元，表示为向量，并将其传输到查询集
  - 梯度元：支持集中关系元的损失梯度。在关系元传递至查询集之前，要通过梯度元更新

- 算法原理图





- Relation-Meta Learner

- L层全连接网络。对于每一个支持集的实体对  $\{(h_i, t_i) \in \mathcal{S}_r\}$  :

$$\mathbf{x}^0 = \mathbf{h}_i \oplus \mathbf{t}_i$$

$$\mathbf{x}^l = \sigma(\mathbf{W}^l \mathbf{x}^{l-1} + b^l)$$

$$R_{(h_i, t_i)} = \mathbf{W}^L \mathbf{x}^{L-1} + b^L$$

- 平均当前任务中所有实体对计算所得的关系，作为当前任务的最终关系元:

$$R_{\mathcal{T}_r} = \frac{\sum_{i=1}^K R_{(h_i, t_i)}}{K}$$

- Embedding Learner

- 真值评分函数

$$s(h_i, t_i) = \|\mathbf{h}_i + R_{\mathcal{T}_r} - \mathbf{t}_i\|$$

- 损失函数

$$L(\mathcal{S}_r) = \sum_{(h_i, t_i) \in \mathcal{S}_r} [\gamma + s(h_i, t_i) - s(h_i, t'_i)]_+$$

- 梯度元

$$G_{\mathcal{T}_r} = \nabla_{R_{\mathcal{T}_r}} L(\mathcal{S}_r)$$

- 更新关系元

$$R'_{\mathcal{T}_r} = R_{\mathcal{T}_r} - \beta G_{\mathcal{T}_r}$$

- 应用于查询集

$$s(h_j, t_j) = \|\mathbf{h}_j + R'_{\mathcal{T}_r} - \mathbf{t}_j\|$$

- 最终目标函数

$$L(\mathcal{Q}_r) = \sum_{(h_j, t_j) \in \mathcal{Q}_r} [\gamma + s(h_j, t_j) - s(h_j, t'_j)]_+$$

## • 实验结果

	MRR		Hits@10		Hits@5		Hits@1	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
<b>NELL-One</b>								
GMatching_RESCAL	<u>.188</u>	–	.305	–	.243	–	<u>.133</u>	–
GMatching_TransE	.171	–	.255	–	.210	–	.122	–
GMatching_DistMult	.171	–	.301	–	.221	–	.114	–
GMatching_ComplEx	.185	<u>.201</u>	<u>.313</u>	<u>.311</u>	<u>.260</u>	<u>.264</u>	.119	<u>.143</u>
GMatching_Random	.151	–	.252	–	.186	–	.103	–
MetaR (BG:Pre-Train)	.164	.209	.331	.355	.238	.280	.093	.141
MetaR (BG:In-Train)	<b>.250</b>	<b>.261</b>	<b>.401</b>	<b>.437</b>	<b>.336</b>	<b>.350</b>	<b>.170</b>	<b>.168</b>
<b>Wiki-One</b>								
GMatching_RESCAL	.139	–	.305	–	.228	–	.061	–
GMatching_TransE	.219	–	.328	–	.269	–	.163	–
GMatching_DistMult	<u>.222</u>	–	<u>.340</u>	–	.271	–	<u>.164</u>	–
GMatching_ComplEx	.200	–	.336	–	<u>.272</u>	–	.120	–
GMatching_Random	.198	–	.299	–	.260	–	.133	–
MetaR (BG:Pre-Train)	<b>.314</b>	<b>.323</b>	<b>.404</b>	<b>.418</b>	<b>.375</b>	<b>.385</b>	<b>.266</b>	<b>.270</b>
MetaR (BG:In-Train)	.193	.221	.280	.302	.233	.264	.152	.178



# 总结

- 总结

- 元学习算法致力于解决知识图谱补全中长尾关系的问题，让模型在少量训练数据的情况下有快速适配的能力
- 基于度量的方法
  - 直接对训练数据和测试数据之间的关联进行建模。
- 基于优化的方法
  - 直接寻找预测一个实体对间是否含有某个关系所需要的核心信息

- 未来方向

- 持续强化知识推理的可解释性

- [1] Chen M, Zhang W, Zhang W, et al. Meta relational learning for few-shot link prediction in knowledge graphs[J]. arXiv preprint arXiv:1909.01515, 2019.
- [2] Zhang C, Yao H, Huang C, et al. Few-shot knowledge graph completion[C]// Proceedings of the AAAI Conference on Artificial Intelligence. 2020, 34(03): 3041-3048.
- [3] Xiong W, Yu M, Chang S, et al. One-shot relational learning for knowledge graphs[J]. arXiv preprint arXiv:1808.09040, 2018.

知人者智，自知者明。胜人者有力，自胜者强。知足者富。强行者有志。不失其所者久。死而不亡者，寿。

# 谢谢！

