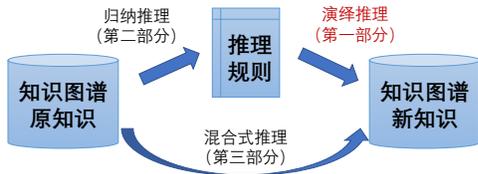


# 知识推理算法概括



# 目录

- 知识推理概述
- 演绎推理：推理具体事实
  - 经典逻辑推理 **命题逻辑**
  - 基于产生式规则的推理
  - 基于概率逻辑学习的推理
  - 自然语言演绎推理
- 归纳推理：学习推理规则
- 基于深度学习的知识推理方法
- 大语言模型下的推理方法
- 总结

# 基于产生式规则的推理

- 产生式系统
  - 一种**前向推理**系统，可以按照一定机制**执行规则**从而达到某些目标
- 典型应用：专家系统（如诊断传染性疾病的专家系统MYCIN）
- 产生式系统的组成
  - 事实集合 (Working Memory)
  - 产生式/规则集合
  - 推理引擎



本节内容参考书籍《知识图谱导论》 65

# 基于产生式规则的推理

- 事实集合/运行内存 (Working Memory, WM)
  - 事实的集合
  - 用于存储当前系统中的所有事实
- 事实
  - 描述对象，如(student name: Alice age:24)
  - 描述关系，如(olderThan John Alice)
- 产生式集合 (Production Memory, PM)
  - 产生式 (推理规则) 的集合
- 产生式
  - IF conditions THEN actions
  - conditions是由条件组成的集合，又称为LHS (Left Hand Side)
  - actions是由动作组成的序列，又称为RHS (Right Hand Side)

# 产生式规则： Left Hand Side

- LHS
  - 条件 (condition) 的集合，各条件之间是**且**的关系
  - 当LHS中的所有条件均被满足，则该规则触发
  - 每个条件形如(type attr<sub>1</sub>:spec<sub>1</sub> ... attr<sub>n</sub>:spec<sub>n</sub>)
    - spec<sub>i</sub>表示对attr<sub>i</sub>的约束，形式可取下面种类
      - 原子，如： Alice (person name:Alice)用于判断取值是否为Alice
      - 变量，如： x (person name:x)用于判断取值是否为x(如果未绑定常量，则视为自由变量)
      - 表达式，如[n+4] (person age:[n+4])用于判断取值是否等于 (需事先赋值)
      - 布尔测试，如{>10} (person age:{>10})用于判断取值是否满足条件
      - 约束的与、或、非操作

# 产生式规则： Right Hand Side

- RHS
  - 动作 (action)的序列，执行时依次执行
  - 动作的种类如下：
    - ADD pattern
      - 向事实集合WM中加入形如pattern的事实
    - REMOVE i
      - 从事实集合WM中移除当前规则第i个条件匹配的事实
    - MODIFY i (attr spec)
      - 对于当前规则第i个条件匹配的事实，将其对应于attr属性的值改为spec

# 产生式规则

## 产生式规则

- IF conditions THEN actions
- 示例1:
  - IF (Student 姓名:XYZ)
  - Then ADD (Person 姓:X)
  - 如果有个学生名为XYZ, 那么向事实集中加入一个事实, 表示有一个姓X的人。
- 示例2 (来自MYCIN专家系统, RULE 047):
  - IF (\$AND (NotDefinite Cntxt Ident) #病原体的鉴别名不确定, 且 (SAME Cntxt Site Blood) #病原体来自血液, 且 (SAME Cntxt Stain Gramneg) #病原体的染色是革兰氏阴性, 且 (SAME Cntxt Morph Rod) #病原体的形态是杆状的, 且 (SAME Cntxt Burnt) #病原体呈褐色,
  - Then (Cntxt Ident Pseudomonas Tally.4) #病原体的鉴别名是假单胞细菌, 可信度为0.4

69

# 推理引擎

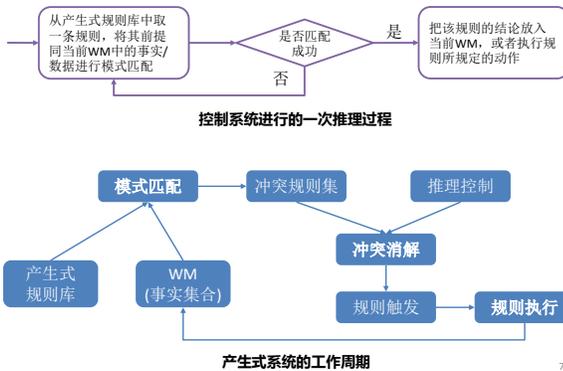
## 该步骤是产生式系统的核心, 用于控制系统的执行

- 模式匹配
  - 用规则的条件部分匹配事实集中的事实, 整个LHS都被满足的规则被触发, 并被加入议程(agenda)
- 冲突消解
  - 按一定的策略从被触发的多条规则中选择一条
- 规则执行
  - 执行被选择出来的规则的RHS, 从而对WM进行一定的操作

产生式系统=事实集合+产生式集合+推理引擎

70

# 产生式系统执行流程

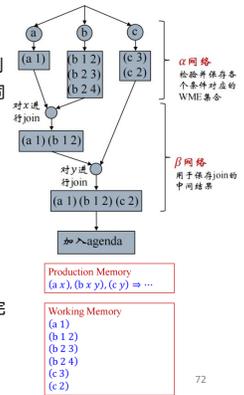


71

# 模式匹配: Rete算法

## Rete算法是产生式规则系统中常用的推理算法

- 核心思想: 将产生式规则中的LHS部分组织成别网络, 然后用分类的匹配项构造匹配网络, 同时缓存中间结果。
- 以空间换时间的做法
- Rete算法基本过程
  - $\alpha$ 网络用来检验和保存规则集中每条规则所对应的条件集合。
  - $\beta$ 网络用来保存Join计算的中间结果。
  - Working Memory中的所有事实首先与 $\alpha$ 网络中的元素进行匹配, 然后按照网络的结构形式完成Join操作, 将Join的中间结果保存于 $\beta$ 网络中, 最终结果加入agenda中。



72

# 冲突消解

- 解决冲突: 匹配成功的规则可能不止一条
  - 从被触发的多条规则中选择一条
    - 注: 在具体场景中, 被触发的多条规则可全被执行
  - 常见策略
    - 随机选择: 从被触发的规则中随机选择一条执行
    - 具体性 (specificity): 选最具体的规则
      - (student name: x)  $\rightarrow$  ...
      - (student name:x age:20)  $\rightarrow$  ...
    - 新近程度(recency): 选择最近没有被触发的规则

73

# 目录

- 知识推理概述
- 演绎推理: 推理具体事实
  - 经典逻辑推理
  - 基于产生式规则的推理
  - 基于概率逻辑学习的推理
  - 自然语言演绎推理
- 归纳推理: 学习推理规则
- 基于深度学习的知识推理方法
- 大语言模型下的推理方法
- 总结

74

# 马尔可夫逻辑网

- 马尔可夫逻辑网 (Markov Logic Network) [Richardson and Domingos, 2006] 是将概率图模型与一阶谓词逻辑相结合的一种统计关系学习模型，其核心思想是通过为规则绑定权重的方式将一阶谓词逻辑规则中的硬性约束 (hard constraints) 进行软化。
  - 1) 一阶谓词逻辑知识库可看作是在一个可能世界的集合上建立一系列硬性规则，即如果一个世界违反了其中的某一条规则，那么这个世界存在概率即为零。
  - 2) 马尔可夫逻辑网的基本思想是让那些硬性规则有所松弛，即如果一个世界违反了一条规则，那么这个世界存在的可能性将降低，但并非不可能。一个世界违反的规则越少，那么这个世界存在的可能性就越大。

76

# 马尔可夫逻辑网

- 马尔可夫逻辑网 (Markov Logic Network) [Richardson and Domingos, 2006] 是将概率图模型与一阶谓词逻辑相结合的一种统计关系学习模型，其核心思想是通过为规则绑定权重的方式将一阶谓词逻辑规则中的硬性约束 (hard constraints) 进行软化。
  - 3) 为此，马尔可夫逻辑网给每条规则都加上一个特定的权重来反映其约束强度。规则的权重越大，其约束能力越强，即对于满足和不满足该规则的两个世界而言，它们之间的差异将越大。当规则的权重设置为无穷大时，其退化为硬性规则。

77

# 马尔可夫逻辑网

- 绑定权重的规则示例
  - 这些规则在现实世界中通常是真的，但不总是真的，并且它们有着不同的成立概率。

Table 1 Example of a first-order knowledge base and MLN. Fr(x, y) is short for Friends(), Sm(x) for Smokes(), and Ca() for Cancer()

English	First-order logic	Clausal form	Weight
Friends of friends are friends	$\forall x \forall y \forall z \text{ Fr}(x, y) \wedge \text{ Fr}(y, z) \Rightarrow \text{ Fr}(x, z)$	$\neg \text{ Fr}(x, y) \vee \neg \text{ Fr}(y, z) \vee \text{ Fr}(x, z)$	0.7
Friendless people smoke	$\forall x (\neg (\exists y \text{ Fr}(x, y)) \Rightarrow \text{ Sm}(x))$	$\text{ Fr}(x, g(x)) \vee \text{ Sm}(x)$	2.3
Smoking causes cancer	$\forall x \text{ Sm}(x) \Rightarrow \text{ Ca}(x)$	$\neg \text{ Sm}(x) \vee \text{ Ca}(x)$	1.5
If two people are friends, either both smoke or neither does	$\forall x \forall y \text{ Fr}(x, y) \Rightarrow (\text{ Sm}(x) \Leftrightarrow \text{ Sm}(y))$	$\neg \text{ Fr}(x, y) \vee \text{ Sm}(x) \vee \neg \text{ Sm}(y),$ $\neg \text{ Fr}(x, y) \vee \neg \text{ Sm}(x) \vee \text{ Sm}(y)$	1.1 1.1

78

# 马尔可夫逻辑网

- 马尔可夫逻辑网形式化定义

定义：马尔可夫逻辑网  $L$  是一个规则-权重对  $(f_i, w_i)$  的集合，其中  $f_i$  是一条标准的一阶谓词逻辑规则， $w_i$  是一个实数。马尔可夫逻辑网  $L$  与一个有限的常量集合  $C = \{c_1, c_2, \dots, c_{|C|}\}$  一起按照如下方式定义了一个马尔可夫网络（也叫马尔可夫随机场） $M_{L,C}$ ：

- $M_{L,C}$  为每一个可能的原子事实在  $L$  中都创建了一个二值节点。当原子事实为真时，该节点取值为1；否则该节点取值为0。
- $M_{L,C}$  为每个  $f_i$  的每一条可能的实例化规则在  $L$  中都设计了一维特征。当实例化规则为真时，该特征取值为1；否则该特征取值为0。该特征的权重即为规则  $f_i$  的权重  $w_i$ 。

79

# 马尔可夫逻辑网

- 从马尔可夫逻辑网的定义出发，很容易得到一个图结构：
  - 每个原子事实对应于图中的一个节点。
  - 当两个节点所表示的原子事实出现在同一个实例化规则之中时，这两个节点之间存在一条边。
  - 所有出现在同一个实例化规则之中的原子事实组成了一个团。

80

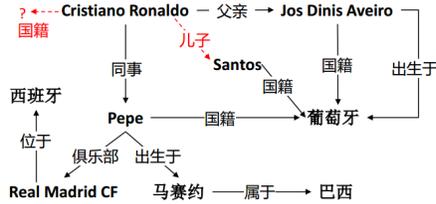
# 马尔可夫逻辑网研究任务

- 利用马尔可夫逻辑网对知识图谱进行建模后，我们可以：
  - 当规则及其权重已知时：推断知识图谱中任意未知事实成立的概率（马尔可夫随机场的推断问题）  
证据变量为知识图谱中的已知事实，问题变量为未知事实
  - 当规则已知但其权重未知时：自动学习每条规则的权重（马尔可夫随机场的参数学习）
  - 当规则及其权重均未知时：自动学习规则及其权重（马尔可夫随机场的结构学习），这实际上属于我们讨论过的归纳推理的范畴

81

# 概率逻辑推理

## 如何运用带权重规则集合进行有效的知识图谱推理?



5.0 父亲 (X,Y) ∧ 国籍 (X,Z) ⇒ 国籍 (Y,Z)

5.0 儿子 (X,Y) ∧ 国籍 (X,Z) ⇒ 国籍 (Y,Z)

2.0 同事 (X,Y) ∧ 俱乐部 (Y,Z) ∧ 位于 (Z,W) ⇒ 国籍 (X,W)

# 知识图谱中概率逻辑推理的问题

## 候选集过大

- 是某些链接推理问题的自然属性，如父亲 (X, Y)，所有的人员都可以作为候选
- 需要庞大的计算量，而大量的计算是无效的计算，是对计算资源的浪费
  - 推理算法本身复杂度高，一次推理的运行时间就很长
  - 总推理时间随着候选集规模的增加而增大
- 会引入大量噪声，其中一部分噪声甚至是无法用已有的规则集合区分的
  - 规则不全或规则之间的冲突
  - 概率模型本身的问题

# 知识图谱中概率逻辑推理的问题

## 候选集过大

- 需要庞大的计算量，而大量的计算是无效的计算，是对计算资源的浪费
- 会引入大量噪声，其中一部分噪声甚至是无法用已有的规则集合区分的

## 现有的解决方法

- 人工设定规则过滤
  - 例如父亲 (X, Y)，使用性别 (Y, 男) 可以过滤掉一半候选
  - 由于知识库中关系的多样性，不能人工列举这类规则
  - 由于真实世界的不确定性，置信度再高的规则也有例外
- 将一跳或两跳的实体直接作为候选
  - 约束太强，有很多例外
  - 统计表明，Freebase中使用这种方法选择出规模为100的候选集，包含正确答案的只有60%

# 基于表示学习的候选预选择方法

## 表示学习的优点:

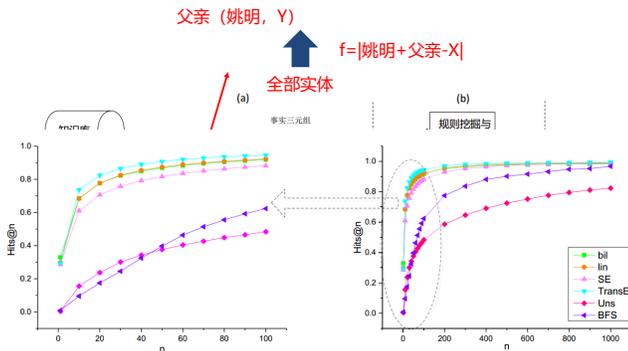
- 简单的向量计算，速度快，即使是大规模的候选集在短时间内部可计算完成
- 前N的准确率高，表示学习很容易把相关实体排到靠前的位置

## 表示学习的缺点:

- 没有应用到显式的逻辑规则，因此推理的结果不够精确

Embedding Model	Hits@1(%)	Hits@10(%)
Uns[4]	0.384	15.573
SE[5]	28.633	61.026
SME-lin[3]	29.807	68.386
SME-bil[3]	32.911	68.506
TransE[4]	29.401	73.71

# 基于表示学习的候选预选择方法

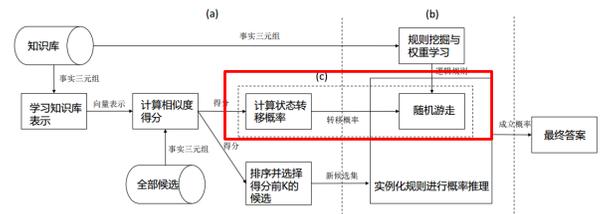


Large-scale knowledge base completion: Inferring via grounding network sampling over selected instances, CIKM 2015. 87

# 基于表示学习的候选预选择

## 仍存在的问题

- 小规模候选集内依然存在噪声
- 表示学习方法计算的候选得分在后续推理中没有被利用



## 基于表示学习的候选预选择方法

### 解决方法

- 将表示学习方法计算的得分，作为先验加入到后续推理过程中，以获得更准确的结果
  - 因为在新的候选集中，候选打分不同，得分高的候选更可能是正确答案，因此候选的得分中含有一定先验信息
  - 将候选得分转化成随机游走的状态转移概率，令得分高的候选有更高的概率采样到规则，相比其他候选更具优势
    - 例如：国籍（姚明，中国）：0.8
    - 国籍（姚明，美国）：0.2
    - 前者比后者采样到规则的概率高了4倍
- 改写随机游走的状态转移概率

$$Q(u, v) = \begin{cases} \frac{(1-\gamma)f(score_v)}{2Z}, & v \in N_{+rand}(u) \\ \frac{1-\gamma}{2Z}, & v \in N_{-rand}(u) \end{cases}$$

- 加入表示学习先验

$$f(score_v) = \lambda \cdot \frac{score_v}{|score|_{max}} + 1$$

91

## 概率软逻辑

- 概率软逻辑 (Probabilistic Soft Logic) [Kimmig et al., 2012] 是马尔可夫逻辑网的进一步延伸，其最大优点是允许原子事实的真值可以在连续的 [0,1] 区间内任意取值，而不像马尔可夫逻辑网那样只能取 {0,1} 中的离散值。
  - 马尔可夫逻辑网：给一阶谓词逻辑加入了出色的不确定性处理能力，通过建模不确定性规则，能够容忍知识库中存在的不完整性和矛盾性问题。
  - 概率软逻辑：进一步增强了马尔可夫逻辑网的不确定性处理能力，能够同时建模不确定性的规则和事实。并且连续真值的引入使得推理从原本的离散优化问题简化为连续优化问题，大大提升了推理效率。

91

## 目录

- 知识推理概述
- 演绎推理：推理具体事实
  - 经典逻辑推理
  - 基于产生式规则的推理
  - 基于概率逻辑学习的推理
  - 自然语言演绎推理
- 归纳推理：学习推理规则
- 基于深度学习的知识推理方法
- 大语言模型下的推理方法
- 总结

93

## 基于表示学习的候选预选择方法

### 实验

- 数据集：FB15K
- 评价指标：Hits@N和AP@N
- 对比方法：1.单独表示学习方法；2.没有进行候选选择的MLN

方法	Hits@1(AP@1)	Hits@10	AP@2	AP@3	AP@5	AP@7	AP@10
Uns	0.00384	0.15573	0.01305	0.01747	0.02128	0.02336	0.02571
SME-bil	0.32911	0.68506	0.32119	0.3208	0.32655	0.32951	0.33348
a SME-lin	0.29807	0.68386	0.29136	0.29011	0.29426	0.29805	0.30102
SE	0.28633	0.61026	0.28015	0.27996	0.28194	0.28480	0.28737
TransE	0.29401	0.73710	0.30263	0.30835	0.31691	0.32099	0.32529
b MLN	0.32540	0.51917	0.30472	0.29668	0.28869	0.28519	0.28232
Uns-INS	0.32899	0.39388	0.30518	0.29349	0.28302	0.28102	0.28288
SME-bil-INS	0.69212	0.83464	0.67224	0.66194	0.65233	0.65051	0.65483
c SME-lin-INS	0.68143	0.82478	0.65615	0.64645	0.63599	0.63568	0.64042
SE-INS	0.67333	0.79938	0.65196	0.64158	0.63029	0.62823	0.63236
TransE-INS	<b>0.69303</b>	<b>0.84894</b>	<b>0.67540</b>	<b>0.66566</b>	<b>0.65781</b>	<b>0.65550</b>	<b>0.66114</b>
d TransE-INS-ES4.5	<b>0.71692</b>	<b>0.86633</b>	<b>0.70146</b>	<b>0.69320</b>	<b>0.68386</b>	<b>0.68100</b>	<b>0.68557</b>

92

## 概率软逻辑研究任务

- 利用概率软逻辑对知识图谱进行建模后，我们可以：
  - 当规则及其权重已知时：推断知识图谱中任意未知事实成立的概率（马尔可夫随机场的推断问题）
    - 证据变量为知识图谱中的已知事实，问题变量为未知事实
  - 当规则已知但其权重未知时：自动学习每条规则的权重（马尔可夫随机场的参数学习）
  - 当规则及其权重均未知时：自动学习规则及其权重（马尔可夫随机场的结构学习），这实际上属于我们讨论过的归纳推理的范畴

92

## 自然语言推理：文本蕴含

### RTE: Recognizing Text Entailment

- 文本蕴含定义为一对文本之间的有向推理关系，其中蕴含前件记作 T (Text)，蕴含后件记作 H (Hypothesis)。
- 如果人们依据自己的理解认为H的语义能够由T的语义推理得出，那么称T蕴含H，记作 T → H

前件(T)	如果你帮助需要的人，上帝就会报答你。	标签
后件(H)	拿钱给穷人可以得到好的结果。	蕴含(entailment)
	拿钱给穷人不会有好结果。	矛盾(contradiction)
	拿钱给穷人会让你成为更好的人。	中立(neutral)

- 典型定义 (Classical Definition)
  - 在T为真的所有可能世界中H都为真，则前件T蕴含后件H
- 应用定义 (Applied Definition)
  - 如果人读了T之后能推断H为真，则前件T蕴含后件H
- 数学定义 (Mathematical Definition)
  - 如果  $P(h \text{ is true} | t) > P(h \text{ is true})$ ，则前件T蕴含后件H

94

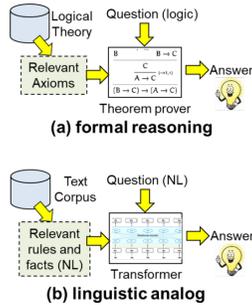
# 自然语言演绎推理: Soft Reasoners over Language

- 使用自然语言模仿“三段论”的知识推理，针对普通、否定等不同层级的推理能力，使用Transformer模型进行模拟。
- (二分类的)准确率能达到95%左右。

*(Input Facts:)* Alan is blue. Alan is rough. Alan is young. Bob is big. Bob is round. Charlie is big. Charlie is blue. Charlie is green. Dave is green. Dave is rough.

*(Input Rules:)* Big people are rough. If someone is young and round then they are kind. If someone is round and big then they are blue. All rough people are green.

Q1: Bob is green. True/false? [Answer: T]  
 Q2: Bob is kind. True/false? [F]  
 Q3: Dave is blue. True/false? [F]



Transformers as Soft Reasoners over Language, IJCAI 2020.

# 自然语言演绎推理任务描述

- 数据样例: (context, statement, answer)
  - context是事实和规则集合，形式为(fact\*, rule\*)
  - statement是问题，也就是要给需要证明的陈述句 (declarative sentence 陈述句)
  - answer为T(rue)或F(false)，分别表示在封闭世界 (CWA) 假设下，能否被context推理得到。

事实、规则、问题陈述都由自然语言表达 (目前为根据模板自动生成的英文)。

*(Input Facts:)* Alan is blue. Alan is rough. Alan is young. Bob is big. Bob is round. Charlie is big. Charlie is blue. Charlie is green. Dave is green. Dave is rough.

*(Input Rules:)* Big people are rough. If someone is young and round then they are kind. If someone is round and big then they are blue. All rough people are green.

Q1: Bob is green. True/false? [Answer: T]  
 Q2: Bob is kind. True/false? [F]  
 Q3: Dave is blue. True/false? [F]

# 数据集构建

- 目标: (context, statement, answer)集合
- 基本思路:
  - 生成逻辑上的小理论(theory)
    - 包含事实和规则(facts+rules)
  - 执行前向链接推理得到蕴含结果
  - 从蕴含结果中得到问题
    - 被证明的结果答案为True
    - 未被证明结果答案为False
- 构造了5个数据集，表征不同推理深度
- 最大推理深度分别为D=0, D≤1, D≤2, D≤3 和 D≤5
  - D=0表示直接从facts中检索得到
  - D≤5为DMax包含≤5的各种深度，可以用于测试模型的泛化能力

Facts and rules (you can provide your own):

Metals conduct electricity.  
 Insulators do not conduct electricity.  
 If something is made of iron then it is metal.  
 Nails are made of iron.

Is it true?

Nails conduct electricity.

# 实验

- 模型
  - RoBERTa-large (在RACE数据集上精调)
    - 也对比了BERT、ESIM (LSTM) 等模型
  - 输入格式: [CLS] context [SEP] statement [SEP]
  - 输出格式: 利用[CLS]计算logit score
  - 使用交叉熵Loss训练模型
- 评价使用准确率 (随机猜测为50%)

# Can RoBERTa Answer Reasoning Questions?

Training	Num Q	Mod0 D = 0	Mod1 D ≤ 1	Mod2 D ≤ 2	Mod3 D ≤ 3	MMax DMax
Test (own)	~ 20000	100	99.8	99.5	99.3	99.2
Test (DMax)	20192	53.5	63.5	83.9	98.9	99.2
Depth=0	6299	100	100	100	100	100
Depth=1	4434	57.9	99.0	98.8	98.5	98.4
Depth=2	2915	34.3	36.8	98.8	98.8	98.4
Depth=3	2396	20.4	23.1	71.1	98.5	98.8
Depth=4	2134	10.2	11.4	43.4	98.8	99.2
Depth=5	2003	11.2	12.3	37.2	97.6	99.8

Out-of-distribution tests (reasoning depth unseen in training)

- 模型能够取得良好性能
- Mod0不能回答更难的推理问题
- Mod3具备良好/不错的推理泛化能力

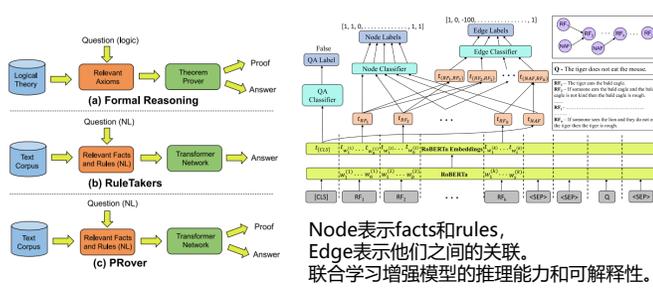
# 基于其他基础模型的实验结果

Training	Mod0 D = 0	Mod1 D ≤ 1	Mod2 D ≤ 2	Mod3 D ≤ 3	MMax DMax
Test (own):					
RoBERTa	100	99.8	99.5	99.3	99.2
BERT	100	99.3	98.2	97.0	96.9
ESIM	100	90.3	87.8	84.2	80.0
DECOMP	72.5	68.2	58.6	57.8	64.1
Test (DMax):					
RoBERTa	53.5	63.5	83.9	98.9	
BERT	53.5	64.1	90.6	95.3	
ESIM	53.5	66.4	73.2	79.6	
DECOMP	56.5	58.1	56.4	57.4	

(Includes questions at depths unseen during training)

- 其他模型效果也不差 (除了DECOMP)
  - DECOMP: 一个NLI模型。the decomposable attention model (DECOMP) on our data [Parikh et al., 2016]

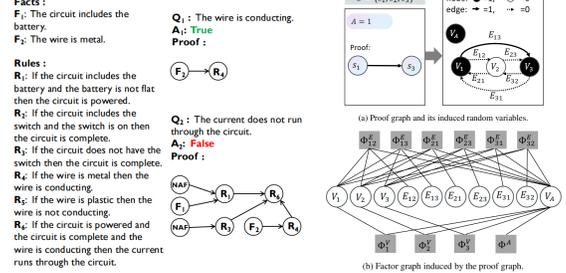
## 后续工作：关注推理过程和可解释性



PProver: Proof generation for interpretable reasoning over rules, EMNLP 2020.

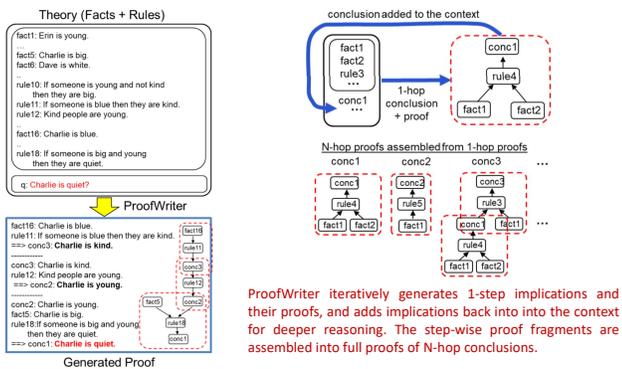
## 后续工作：建模推理过程图

### 基于概率图推理同时得到答案和证明过程



Probabilistic Graph Reasoning for Natural Proof Generation, ACL 2021.

## 后续工作：逐步推理，得到完整证明过程



ProofWriter: Generating Implications, Proofs, and Abductive Statements over Natural Language, ACL 2021. 104

## 后续工作：显式选择规则、事实并进行推理

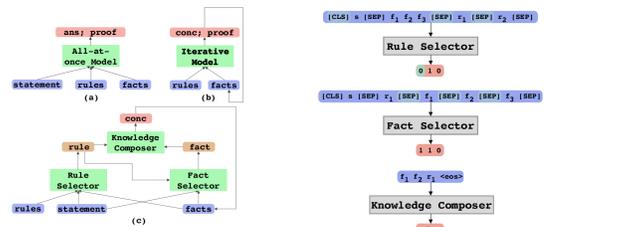


Figure 2: Reasoning process in different models. (a): ProofWriter ("All") directly output the entailment prediction and proof graph for given input. (b): ProofWriter ("Iter") iteratively generates the one-step intermediate conclusions and their proofs. (c): FAIRR selects a rule, then a fact, and finally combines them to generate an intermediate inference. Note that the proof is implicitly determined by the selection steps. Please refer to Section 3.1 for details.

- 1) 规则选择器：分类任务，对规则r1,r2进行分类，如果选择了CLS则停止；
- 2) 事实选择器：分类任务，对事实集合进行多分类；
- 3) 知识组合器：基于T5的生成模型。

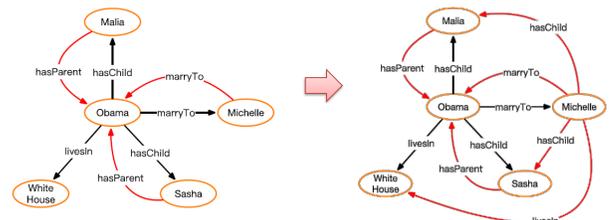
FaIRR: Faithful and Robust Deductive Reasoning over Natural Language, ACL 2022.

## 目录

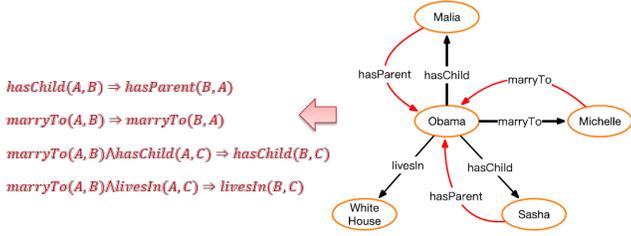
- 知识推理概述
- 演绎推理：推理具体事实
- 归纳推理：学习推理规则
  - 归纳推理概述
  - 归纳逻辑程序设计
  - 路径排序算法 (PRA算法)
  - 关联规则挖掘算法 (AMIE算法)
- 基于深度学习的知识推理方法
- 大语言模型下的推理方法
- 总结

## 演绎推理：应用推理规则

$hasChild(A, B) \Rightarrow hasParent(B, A)$   
 $marryTo(A, B) \Rightarrow marryTo(B, A)$   
 $marryTo(A, B) \wedge hasChild(A, C) \Rightarrow hasChild(B, C)$   
 $marryTo(A, B) \wedge livesIn(A, C) \Rightarrow livesIn(B, C)$



# 归纳推理：学习推理规则



$hasChild(A, B) \Rightarrow hasParent(B, A)$   
 $marryTo(A, B) \Rightarrow marryTo(B, A)$   
 $marryTo(A, B) \wedge hasChild(A, C) \Rightarrow hasChild(B, C)$   
 $marryTo(A, B) \wedge livesIn(A, C) \Rightarrow livesIn(B, C)$

# 推理规则

- 基于规则的推理：精准+可解释
  - 规则学习 (Rule Induction)：自动化的规则获取
- 一阶谓词逻辑规则
  - $\forall x, y \text{ Female}(x) \wedge \text{ParentOf}(y, x) \Rightarrow \text{DaughterOf}(x, y)$
  - $\forall x, y \text{ CapitalOf}(x, y) \Rightarrow \text{LocatedIn}(x, y)$
  - 谓词：CapitalOf, LocatedIn, ParentOf等
  - 个体变量：x, y
  - 逻辑符号：与、或、非、蕴含 ( $\Rightarrow$ )，表示“若...，则...”的语义
  - 全称量词： $\forall$ ，表示“对任意的”、“凡”、“都”等语义
  - 存在量词： $\exists$ ，表示“至少存在一个”、“有”等语义
  - 规则体 (body)：CapitalOf(x, y)，表示该规则的前提
  - 规则头 (head)：LocatedIn(x, y)，表示该规则的结论
  - 规则实例：CapitalOf(Beijing, China)  $\Rightarrow$  LocatedIn(Beijing, China)

# 推理规则概述

- 规则：包含规则主体 (body) 和规则头 (head) 两部分
    - Rule: body  $\rightarrow$  head     CapitalOf(x, y)  $\Rightarrow$  LocatedIn(x, y)
    - head部分都是一个二元的原子，body则由1个或多个一元或二元原子组成 (body是一个合取范式)
    - 一般规则：body部分包含肯定部分和否定部分
      - 城市(x, y)  $\wedge$   $\neg$ 沿海(x)  $\rightarrow$  内陆城市(x, y)
    - 霍恩规则 (horn rules)：body部分只包含肯定部分
    - 路径规则： $r_1(e_1, e_2) \wedge r_2(e_2, e_3) \dots r_n(e_n, e_{n+1}) \rightarrow r_0(e_1, e_{n+1})$
- 路径规则  $\in$  霍恩规则  $\in$  一般规则

# 目录

- 知识推理概述
- 演绎推理：推理具体事实
- 归纳推理：学习推理规则
  - 归纳推理概述
  - 归纳逻辑程序设计
  - 路径排序算法 (PRA算法)
  - 关联规则挖掘算法 (AMIE算法)
- 基于深度学习的知识推理方法
- 大语言模型下的推理方法
- 总结

# 归纳逻辑程序设计

- 归纳逻辑程序设计 (Inductive Logic Programming, ILP) 使用一阶谓词逻辑来进行知识表示，通过修改和扩充逻辑表达式来完成对数据的归纳。

给定：

- 目标谓词 P
- P 的正例集合 E<sup>+</sup>
- P 的反例集合 E<sup>-</sup>
- 背景知识

目标：

- 找到定义目标谓词 P 的规则，使其覆盖所有正例而不覆盖任何反例

背景知识	Female(Ann)
	Female(Mary)
	Female(Eve)
	ParentOf(Ann, Mary)
	ParentOf(Eve, Tom)
样例	DaughterOf(Mary, Ann)
	DaughterOf(Mary, Bob)
	$\neg$ DaughterOf(Tom, Ann)
	$\neg$ DaughterOf(Tom, Bob)
	$\neg$ DaughterOf(Tom, Eve)
	$\neg$ DaughterOf(Mary, Eve)

DaughterOf(X, Y)  $\leftarrow$  Female(X)  $\wedge$  ParentOf(Y, X)

# FOIL算法

- FOIL (First Order Inductive Learner) [Quinlan, 1990] 利用序贯覆盖实现规则的学习，其基本流程为：
  - ① 从空规则 “P  $\leftarrow$ ” 开始，将目标谓词作为规则头
  - ② 逐一将其他谓词加入规则体进行考察，按预定标准评估规则的优劣并选取最优规则
  - ③ 将该规则覆盖的训练样例去除，以剩下的训练样例组成训练集重复上述过程

# FOIL算法

- FOIL (First Order Inductive Learner) [Quinlan, 1990] 利用**序贯覆盖实现**规则的学习，其基本流程为：
  - 从空规则 “ $P \leftarrow$ ” 开始，将目标谓词作为规则头
  - 逐一将其他谓词加入规则体进行考察，按预定标准评估规则的优劣并选取最优规则
  - 将该规则覆盖的训练样例去除，以剩下的训练样例组成训练集重复上述过程

从一般到特殊：逐步向规则体添加约束，直至其不覆盖任何反例

$$P \leftarrow L_1 \wedge L_2 \wedge \dots$$

114

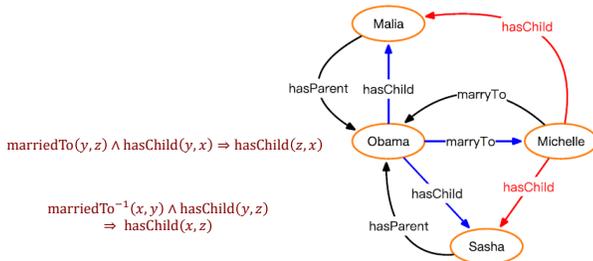
# 目录

- 知识推理概述
- 演绎推理：推理具体事实
- 归纳推理：学习推理规则
  - 归纳推理概述
  - 归纳逻辑程序设计
  - 路径排序算法 (PRA算法)
  - 关联规则挖掘算法 (AMIE算法)
- 基于深度学习的知识推理方法
- 大语言模型下的推理方法
- 总结

115

# 知识图谱中规则与关系路径

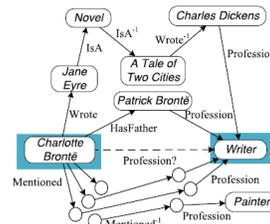
- 知识图谱中包含的仅仅是实体间的二元关系，因此规则与知识图谱中的关系路径存在对应关系。



116

# 路径排序算法

- PRA (Path Ranking Algorithm) [Lao et al., 2011] 以实体间的路径作为特征，来学习目标关系的分类器。



$$score(s, t) = \sum_{\pi_j \in \mathcal{P}_t} \theta_j P(s \rightarrow t; \pi_j)$$

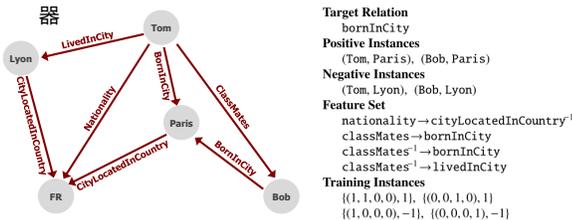
从实体  $s$  出发，通过关系路径  $\pi_j$  到达实体  $t$  的概率也可以是布尔值，表示实体  $s$  和实体  $t$  之间是否存在路径  $\pi_j$  还可以是两实体间路径出现的次数、频率等

Random Walk Inference and Learning in a large scale Knowledge Base, EMNLP 2011.

117

# 路径排序算法

- PRA工作流程
  - 特征抽取：生成并选择路径特征集合
  - 特征计算：计算每个训练样例的特征值
  - 分类器训练：根据训练样例，为每个目标关系训练一个分类器



118

# 路径排序算法

- PRA工作流程
  - 特征抽取：
    - 随机游走，广度优先搜索，深度优先搜索
  - 特征计算：
    - 随机游走概率，布尔值（出现/不出现），出现频次/频率
  - 分类器训练：
    - 单任务学习：为每个关系单独训练一个二分类分类器
    - 多任务学习：将不同关系进行联合学习，同时训练它们的分类器

119

# 路径排序算法

- PRA规则学习：根据分类器权重自动挖掘并筛选可靠规则
- PRA在Freebase上挖掘出的规则

**Target Relation**  
/organization/place\_founded

**Top 5 Features**  
/location/contains<sup>-1</sup>  
/organization/headquarter\_citytown  
/organization/headquarter\_citytown→/location/contains<sup>-1</sup>  
/organization/headquarter\_stateprovince→/location/contains  
/organization/headquarter\_citytown→/bibs\_location/state

120

# 目录

- 知识推理概述
- 演绎推理：推理具体事实
- 归纳推理：学习推理规则
  - 归纳推理概述
  - 归纳逻辑程序设计
  - 路径排序算法 (PRA算法)
  - 关联规则挖掘算法 (AMIE算法)
- 基于深度学习的知识推理方法
- 大语言模型下的推理方法
- 总结

121

# 规则学习评估方法

CapitalOf(x,y) ⇒ LocatedIn(x,y)

- 规则实例化
  - 将规则中的变量替换成知识图谱中的实体后的结果
    - CapitalOf(北京,中国) ⇒ LocatedIn(北京,中国)
- 支持度 (support)
  - 支持度指同时满足规则主体和规则头的实例个数
- 置信度 (confidence)
  - $confidence(rule) = \frac{support(rule)}{\#body(rule)}$
- 规则头置信度 (head Coverage)
  - $HC(rule) = \frac{support(rule)}{\#head(rule)}$
- 三个评估方法没有必然关联，通常会结合这三者来综合衡量规则的质量

122

# AMIE：不完备知识库中的关联规则挖掘

- AMIE (Association Rule Mining under Incomplete Evidence) [Galárraga et al., 2013] 支持从不完备的知识库中，挖掘**闭式 (closed)** 规则（即路径规则）。
  - 两个谓词共享一个变量或实体，则称其为连通的。
  - 规则中任意两个谓词可通过连通关系的传递性相连，则称该规则为连通的。
  - 规则是连通的并且其中的变量都至少出现两次，则称其为闭式 (closed) 逻辑规则。

diedIn(x,y) ⇒ ∃z : wasBornIn(x,z) ✗

diedIn(x,y) ⇒ wasBornIn(x,y) ✓

hasChild(p,c) ∧ isCitizenOf(p,s) ⇒ isCitizenOf(c,s) ✓

123

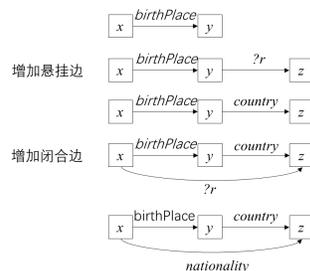
# AMIE：不完备知识库中的关联规则挖掘

- AMIE依次学习预测每种关系的规则。对于每种关系，从规则体为空的规则开始，通过三种操作扩展规则体部分，保留支持度大于阈值的候选（闭式）规则。
  - **添加悬挂边 (Add Dangling Atom)**: 悬挂边是指边的一端是一个未出现过的**变量**，而另一端（变量或常量）是在规则中出现过的。
  - **添加实例边 (Add Instantiated Atom)**: 实例边与悬挂边类似，边的一端也是在规则中出现过的变量或常量，但另一端是未出现过的**常量**，也就是知识库中的实体。
  - **添加闭合边 (Add Closing Atom)**: 闭合边则是连接两个已经存在于规则中的元素（变量或常量）的边。

124

# AMIE：不完备知识库中的关联规则挖掘

- AMIE工作流程示意



birthPlace(x,y) ∧ country(y,z) ⇒ nationality(x,z)

125

# AMIE: 不完备知识库中的关联规则挖掘

## AMIE规则评估

- 支持度 (support): 同时符合规则体和规则头的实例数目

$$supp(\vec{B} \Rightarrow r(x, y)) := \#(x, y) : \exists z_1, \dots, z_m : \vec{B} \wedge r(x, y)$$

- 置信度 (confidence): 支持度除以仅符合规则体的实例数目

$$conf(\vec{B} \Rightarrow r(x, y)) := \frac{supp(\vec{B} \Rightarrow r(x, y))}{\#(x, y) : \exists z_1, \dots, z_m : \vec{B}}$$



126

# AMIE: 不完备知识库中的关联规则挖掘

## AMIE规则评估

- 支持度 (support): 同时符合规则体和规则头的实例数目

$$supp(\vec{B} \Rightarrow r(x, y)) := \#(x, y) : \exists z_1, \dots, z_m : \vec{B} \wedge r(x, y)$$

- 置信度 (confidence): 支持度除以仅符合规则体的实例数目

$$conf(\vec{B} \Rightarrow r(x, y)) := \frac{supp(\vec{B} \Rightarrow r(x, y))}{\#(x, y) : \exists z_1, \dots, z_m : \vec{B}}$$

封闭世界假设 (Closed World Assumption): 知识库中不存在的事实都是错误的



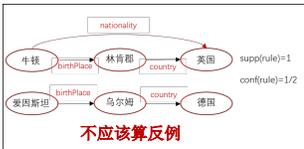
127

# AMIE: 不完备知识库中的关联规则挖掘

## AMIE规则评估

- 置信度 (confidence): 支持度除以仅符合规则体的实例数目

- 部分完整性假设 (Partial Completeness Assumption): 在知识库中, 如果存在某个实体 x 的关系 r 属性, 则知识库中包含了 x 的所有关系 r 属性



128

# AMIE: 不完备知识库中的关联规则挖掘

## AMIE规则评估

- 支持度 (support): 同时符合规则体和规则头的实例数目

$$supp(\vec{B} \Rightarrow r(x, y)) := \#(x, y) : \exists z_1, \dots, z_m : \vec{B} \wedge r(x, y)$$

- 置信度 (confidence): 支持度除以仅符合规则体的实例数目

$$conf(\vec{B} \Rightarrow r(x, y)) := \frac{supp(\vec{B} \Rightarrow r(x, y))}{\#(x, y) : \exists z_1, \dots, z_m : \vec{B}}$$

- PCA置信度 (PCA confidence):

$$pcaconf(\vec{B} \Rightarrow r(x, y)) := \frac{supp(\vec{B} \Rightarrow r(x, y))}{\#(x, y) : \exists z_1, \dots, z_m, y' : \vec{B} \wedge r(x, y')}$$

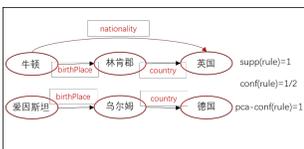


129

# AMIE: 不完备知识库中的关联规则挖掘

$$pcaconf(\vec{B} \Rightarrow r(x, y)) := \frac{supp(\vec{B} \Rightarrow r(x, y))}{\#(x, y) : \exists z_1, \dots, z_m, y' : \vec{B} \wedge r(x, y')}$$

只要存在一个不是y的才可算数



130

# AMIE: 不完备知识库中的关联规则挖掘

## AMIE在YAGO上挖掘出的规则

Rule	Precision in the unknown region	Std. Confidence	PCA Confidence
?b <isMarriedTo> ?a => ?a <isMarriedTo> ?b	100.00%	53.01%	91.79%
?a <isIn> ?c ?c <isLocatedIn> ?b => ?a <isPoliticianOf> ?b	13.33%	2.70%	85.30%
?a <isVessel> ?c ?c <isLocatedIn> ?b => ?a <isPoliticianOf> ?b	13.33%	2.47%	83.33%
?a <isLocatedIn> ?c ?c <hasOfficialLanguage> ?b => ?a <hasOfficialLanguage> ?b	70.83%	0.07%	83.33%
?a <isMarriedTo> ?a ?c <isVessel> ?b => ?a <isVessel> ?b	79.31%	33.71%	69.68%
?c <isMarriedTo> ?a ?c <isVessel> ?b => ?a <isVessel> ?b	60.71%	22.81%	68.42%
?c <hasOfficialLanguage> ?b ?c <isLocatedIn> ?a => ?a <hasOfficialLanguage> ?b	30.00%	14.29%	64.29%
?a <created> ?b ?a <produced> ?b => ?a <directed> ?b	0.00%	49.83%	58.68%
?a <isMarriedTo> ?a ?b <hasChild> ?b => ?a <hasChild> ?b	34.48%	57.57%	57.57%
?a <wasBornIn> ?c ?c <isLocatedIn> ?b => ?a <isCitizenOf> ?b	3.33%	3.75%	57.42%
?a <wasBornIn> ?c ?c <isLocatedIn> ?b => ?a <isPoliticianOf> ?b	0.00%	0.65%	57.18%
?a <isMarriedTo> ?c ?c <hasChild> ?b => ?a <hasChild> ?b	33.33%	56.11%	56.11%
?c <isLocatedIn> ?b ?a <isLocatedIn> ?c => ?a <isCitizenOf> ?b	3.45%	0.40%	54.24%
?a <isCitizenOf> ?b => ?a <isVessel> ?b	100.00%	4.50%	52.36%
?a <influences> ?c ?c <isCitizenOf> ?b => ?a <isCitizenOf> ?b	3.33%	10.56%	51.96%
?a <isVessel> ?c ?c <isLocatedIn> ?b => ?a <isCitizenOf> ?b	0.00%	3.91%	48.13%
?a <isIn> ?c ?c <isLocatedIn> ?b => ?a <isCitizenOf> ?b	0.00%	4.51%	48.13%
?a <actedIn> ?b ?a <created> ?b => ?a <directed> ?b	3.45%	38.29%	45.23%
?a <directed> ?b => ?a <isVessel> ?b	84.00%	1.21%	43.41%
?a <hasChild> ?c ?b <hasChild> ?c => ?a <isMarriedTo> ?b	70.00%	17.32%	40.98%

131

[http://resources.mpi-inf.mpg.de/yago-naga/amie/data/yago2/amie\\_yago2\\_2.html](http://resources.mpi-inf.mpg.de/yago-naga/amie/data/yago2/amie_yago2_2.html)

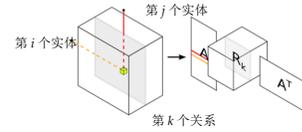
# 目录

- 知识推理概述
- 演绎推理：推理具体事实
- 归纳推理：学习推理规则
- 基于深度学习的知识推理方法
  - 基于表示学习的知识图谱推理
  - 基于强化学习的知识图谱推理
- 大语言模型下的推理方法
- 总结

132

# 知识图谱的数值表示（向量化表示）

- 把知识库中的实体和关系表示为低维空间的对象（向量）及其它们的操作（空间转换）；
- 该表示能够蕴涵其在知识库中的性质，即具有类似上下文的对象，在低维空间中更接近。

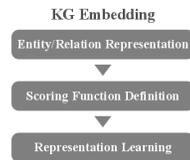


事实集合 → 学习实体和关系的低维表示 → 事实推理  
 表示学习技术：从原始数据中学习概念的潜在表示

133

# 分布式知识表示

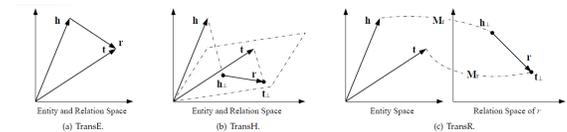
- 分布式知识表示 (Knowledge Graph Embedding) 的核心思想是将符号化的实体和关系在低维连续向量空间进行表示，在简化计算的同时最大程度保留原始的图结构。
- 基本步骤：
  - 实体关系表示：定义实体和关系在向量空间中的表示形式（向量/矩阵/张量）。
  - 打分函数定义：定义打分函数，衡量每个三元组成立的可能性。
  - 表示学习：构造优化问题，学习实体和关系的低维连续向量表示。



134

# 位移距离模型

- 代表性方法：TransE及其变种
  - 目标：头尾实体表示之差与关系表示一致
  - head entity + relation = tail entity
  - China - Beijing = France - Paris = capital-of
  - Beijing + capital-of = China
  - Paris + capital-of = France



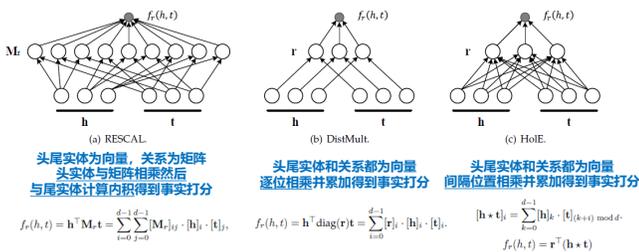
$$f_r(h, t) = -\|h + r - t\|_{1/2}, \quad h_{\perp} = h - w_{\perp}^T h w_{\perp}, \quad t_{\perp} = t - w_{\perp}^T t w_{\perp}, \quad h_{\perp} = M_{\perp}^T h, \quad t_{\perp} = M_{\perp}^T t$$

$$f_r(h, t) = -\|h_{\perp} + r - t_{\perp}\|_2, \quad f_r(h, t) = -\|h_{\perp} + r - t_{\perp}\|_2$$

135

# 语义匹配模型

- 代表性方法：RESCAL及其变种
  - 直接根据三元组头尾实体和关系的表示定义计算函数
  - matching(relation, composition(head, tail))



136

# 融合多元化信息的分布式知识表示

- 上述分布式知识表示方法仅用到了知识图谱中的三元组信息，还有多种其他类型的信息也被证实能够提升分布式知识表示的效果。
  - 实体类别
  - 关系路径
  - 实体描述文本
  - 逻辑规则
  - ...

<https://github.com/xinguoxia/KGE>

137

# 目录

- 知识推理概述
- 演绎推理：推理具体事实
- 归纳推理：学习推理规则
- 基于深度学习的知识推理方法
  - 基于表示学习的知识图谱推理
  - 基于强化学习的知识图谱推理
- 大语言模型下的推理方法
- 总结

138

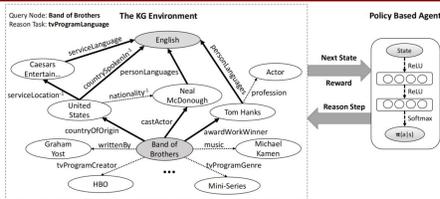
# DeepPath: 基于强化学习的推理

- PRA算法的升级版，利用强化学习 (RL) 学习路径 (paths)
- 将知识图谱推理简化为一个“事实判断”任务
- 事实判断即确定一个三元组 (h, r, t) 是否成立
  - 寻找一条能链接已知头实体h和尾实体的路径，并将此问题建模为序列决策问题。
  - 训练基于策略梯度的强化学习方法搜索路径。
  - 需要事先编码KG，使用TransE等传统方法预训练KG

DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning, EMNLP 2017.

139

## DeepPath模型示意图



- 状态表示:  $s_t = (e_t, e_{target} - e_t)$ ,  $e_t$ 表示agent当前所处结点
- 动作空间: 给定源实体 $e_s$ , agent使用策略网络从所有关系中选择最可能的关系进行探索。

### Reward (激励函数设计):

- 1) 是否准确达到 $e_{target}$ :  $r_{GLOBAL} = \begin{cases} +1, & \text{if the path reaches } e_{target} \\ -1, & \text{otherwise} \end{cases}$
  - 2) 路径效率:  $r_{EFFICIENCY} = \frac{1}{length(p)}$
  - 3) 路径多样性:  $r_{DIVERSITY} = -\frac{1}{|F|} \sum_{i=1}^{|F|} \cos(p, p_i)$
- $p$ 为路径关系embedding之和

140

## DeepPath效果: 关系预测

- 对于每个三元组，使用PRA得到TopK结果，来得到训练和测试样本的负样本 (即替换掉真实三元组中的t为t')。
- 判断真实的(h,r,t)是否得分最高

Tasks	FB15K-237				NELL-995				
	PRA	RL	TransE	TransR	PRA	RL	TransE	TransR	
teamSports	0.987	0.955	0.896	0.784	athletePlaysForTeam	0.547	0.750	0.627	0.673
birthPlace	0.441	0.531	0.403	0.417	athletePlaysInLeague	0.841	0.960	0.773	0.912
personNationality	0.846	0.823	0.641	0.720	athleteHomeStadium	0.859	0.890	0.718	0.722
filmDirector	0.349	0.441	0.386	0.399	athletePlaysSport	0.474	0.957	0.876	0.963
filmWrittenBy	0.601	0.457	0.563	0.605	teamPlaysSports	0.791	0.738	0.761	0.814
filmLanguage	0.663	0.670	0.642	0.641	orgHeadquarterCity	0.811	0.790	0.620	0.657
tvLanguage	0.960	0.969	0.804	0.906	worksFor	0.681	0.711	0.677	0.692
capitalOf	0.829	0.783	0.554	0.493	bomLocation	0.668	0.757	0.712	0.812
organizationFounded	0.281	0.309	0.390	0.339	personLeadsOrg	0.700	0.795	0.751	0.772
musicianOrigin	0.426	0.514	0.361	0.379	orgHiredPerson	0.599	0.742	0.719	0.737
...	...	...	...	...	...	...	...	...	...
Overall	0.541	0.572	0.532	0.540	...	0.675	0.796	0.737	0.789

141

## DeepPath学习得到的路径样例

Relation	Reasoning Path
filmCountry	filmReleaseRegion → locationContains <sup>-1</sup> featureFilmLocation → locationContains <sup>-1</sup> actorFilm <sup>-1</sup> → personNationality
personNationality	placeOfBirth → locationContains <sup>-1</sup> peoplePlaceLived → locationContains <sup>-1</sup> peopleMarriage → locationOfCeremony → locationContains <sup>-1</sup>
tvProgramLanguage	tvCountryOfOrigin → countryOfficialLanguage tvCountryOfOrigin → filmReleaseRegion <sup>-1</sup> → filmLanguage tvCastActor → filmLanguage
personBornInLocation	personBornInCity graduatedUniversity → graduatedSchool <sup>-1</sup> → personBornInCity personBornInCity → atLocation <sup>-1</sup> → atLocation
athletePlaysForTeam	athleteHomeStadium → teamHomeStadium <sup>-1</sup> athletePlaysSport → teamPlaysSport <sup>-1</sup> athleteLedSportsTeam
personLeadsOrganization	worksFor organizationTerminatedPerson <sup>-1</sup> mutualProxyFor <sup>-1</sup>

142

## 基于强化学习的查询问答

- 把知识推理定义为查询问题任务
  - <Colin Kaepernick, PlaysInLeague, ?>
  - 无法“预知”答案对应的尾实体 (更符合真实场景)
  - 需要尽可能避免遍历大规模知识图谱 (影响算法效率)



Go for a Walk and Arrive at the Answer: Reasoning Over Paths in Knowledge Bases using Reinforcement Learning, ICLR 2018.

143

## 推理模式

人类进行推理时，往往呈现出以下两个特点：

- 考虑尽可能多的因素，全局推理。实际中，考虑的影响因素越全面，就越有可能得到正确的推理结论。



- 人的智能表现在可以利用潜在的推理模式，而这些推理模式难以穷举。有些事物之间虽然没有呈现出显式的联系，但有可能存在尚未探知的隐式规律。

144

## 符号推理 VS. 数值推理

- 之前介绍的几种归纳和演绎推理方法都属于**符号推理**的范畴，即在知识图谱中的实体和关系符号上直接进行推理。
- 与符号推理相对的就是**数值推理**，即使用数值计算，尤其是向量矩阵计算的方法，捕捉知识图谱上隐式的关联，模拟推理的进行。基于分布式知识表示的推理就是典型的数值推理方法。
  - 捕捉实体和关系之间的隐式关联
  - 分布式空间映射对特征间的复杂关系进行了解耦，减少了维数灾难问题 (curse of dimensionality)
  - 使符号数据可以直接参与运算且计算速度非常快

145

## 符号推理与分布式表示推理的比较

	符号表示	分布式表示
表示方法	离散符号	连续的数值
推理范围	局部	全局
推理的精确性	高	较低
效率	小规模高，大规模低	高
是否易被人理解	容易	困难
与其他系统结合的难易程度	难	易
跨领域	难 (需专家设计种子规则)	易

知识推理：融合之道  
融合符号表示和分布式表示的知识推理

146