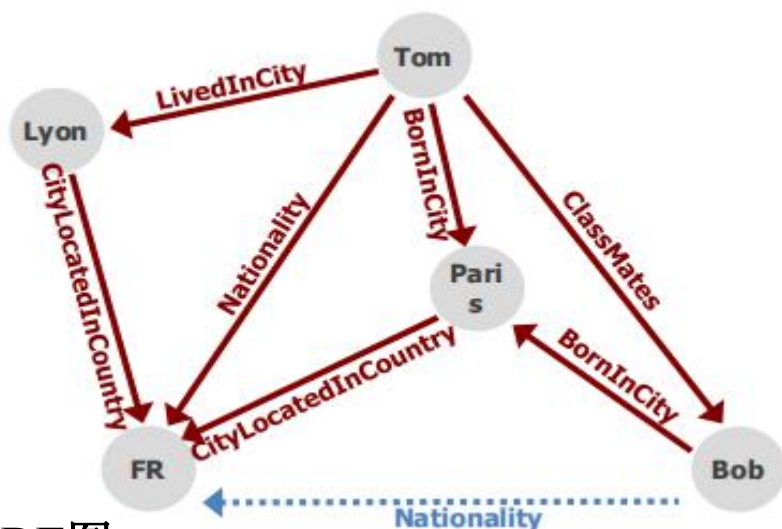


知识图谱及其推理

- 知识图谱概述
- 知识图谱中的知识表示
- 知识图谱的作用及构建
- 知识图谱中的知识推理

知识图谱中的知识推理

- ❑ 知识推理 (**knowledge inference**): 根据知识图谱中已有的知识（**事实或关系**），推断出新的、未知的知识。
- ❑ 知识图谱需要推理，主要体现在两种任务上：
 - **知识库补全**: 根据知识库中已有的知识预测出新知识。
 - **知识库问答**: 通过对自然问句解析，从知识库中寻找答案。



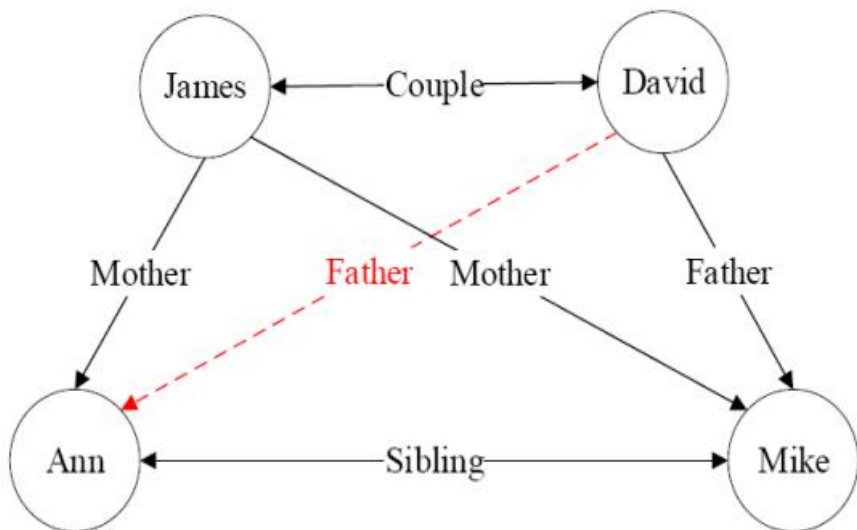
(Tom, BornInCity, Paris)
(Tom, LivedInCity, Lyon)
(Tom, Nationality, France)
(Tom, ClassMates, Bob)
(Paris, CityLocatedInCountry, France)
(Lyon, CityLocatedInCountry, France)
(Bob, BornInCity, Paris)
(Bob, Nationality, France)?

RDF图

三元组表

提高知识的完备性，扩大知识的覆盖面

知识图谱中的知识推理



一个简单的家庭关系知识图谱

问题：如何从知识图谱中推理得到

father(David, Ann)



$(\forall x)(\forall y)(\forall z)(Mother(z, y) \wedge Couple(x, z) \rightarrow Father(x, y))$

如果能够学习得到这条规则，该有多好？

（从具体例子中学习，这是归纳推理的范畴）

2020/3/16

□ 可利用一阶谓词来表达刻画知识图谱中节点之间存在的关系，如图中形如 **<James, Couple, David>** 的关系可用一阶逻辑的形式来描述，即 **Couple(James, David)**。

□ **Couple (x,y)** 是一阶谓词，**Couple** 是图中实体之间具有的关系，**x** 和 **y** 是谓词变量

□ 从图中已有关系可推知 **David** 是 **Ann** 的父亲，但这一关系在图中初始图(无红线)中并不存在，是需要推理的目标。

归纳推理：学习推理规则

- 从知识库中自动学习谓词逻辑规则

$$(\forall x)(\forall y)(\forall z)(Mother(z, y) \wedge Couple(x, z) \rightarrow Father(x, y))$$

- 规则学习方法

- 归纳逻辑程序设计 (**Inductive Logic Programming**)
- 关联规则挖掘 (**Association Rule Mining**)
- 路径排序算法 (**Path Ranking Algorithm**)
- 马尔可夫逻辑网和概率软逻辑中的结构学习 (**structure learning**)

归纳逻辑程序设计

- 归纳逻辑程序设计 (Inductive Logic Programming)
使用一阶谓词逻辑来进行知识表示，
通过从训练样例中归纳，完成推理任务。

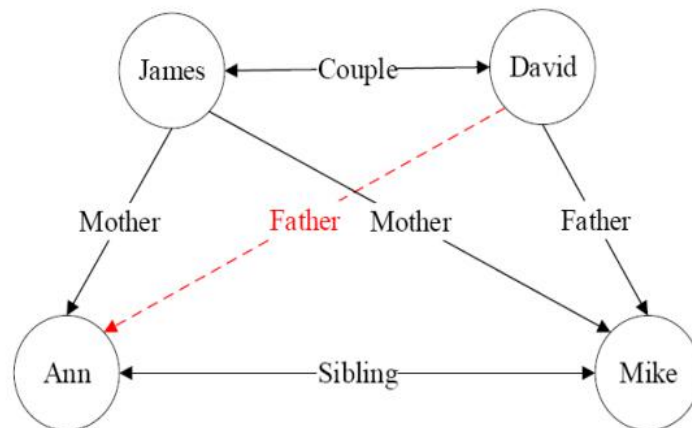
- 给定：

- 目标谓词：例如 **Father(x,y)**
- 目标谓词的正例集合
- 目标谓词的反例集合
- 背景知识

- 目标：

- 找到定义目标谓词的规则，使规则覆盖所有正例而不覆盖任何反例。

◆ ILP 的代表性方法：FOIL
(First Order Inductive Learner)
通过序贯覆盖实现规则推理。



目标谓词	Father(David, Mike)
训练样例	\neg Father(David, James)
集合	\neg Father(James, Ann)
	\neg Father(James, Mike)
	\neg Father(Ann, Mike)

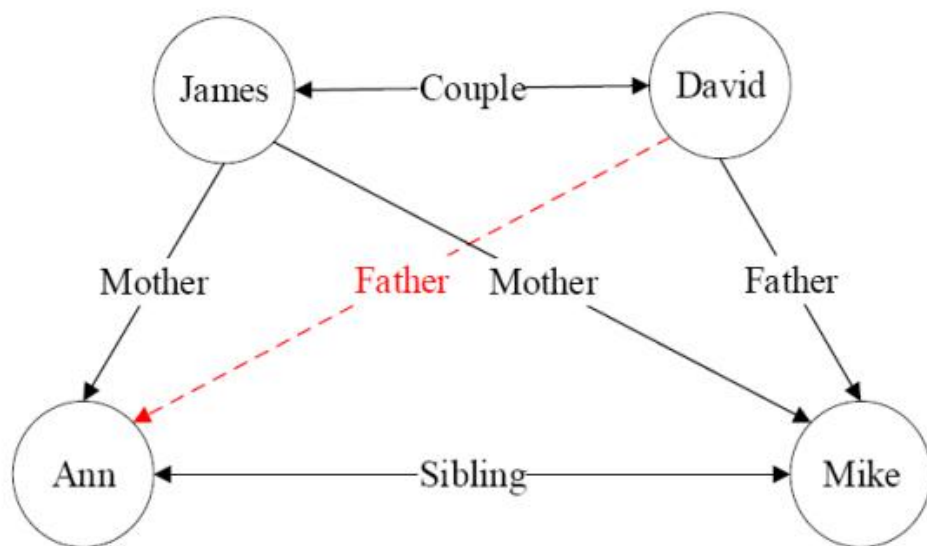
背景知识	Sibling(Ann, Mike)
样例集合	Couple(David, James)
	Mother(James, Ann)
	Mother(James, Mike)

$$(\forall x)(\forall y)(\forall z)(Mother(z, y) \wedge Couple(x, z) \rightarrow \text{Father}(x, y))$$

↑
前提约束谓词
(学习得到)

↑
目标谓词
(已知)

归纳逻辑程序设计



一个简单的家庭关系知识图谱

目标谓词	Father(David, Mike)
训练样例	\neg Father(David, James)
集合	\neg Father(James, Ann)
	\neg Father(James, Mike)
	\neg Father(Ann, Mike)

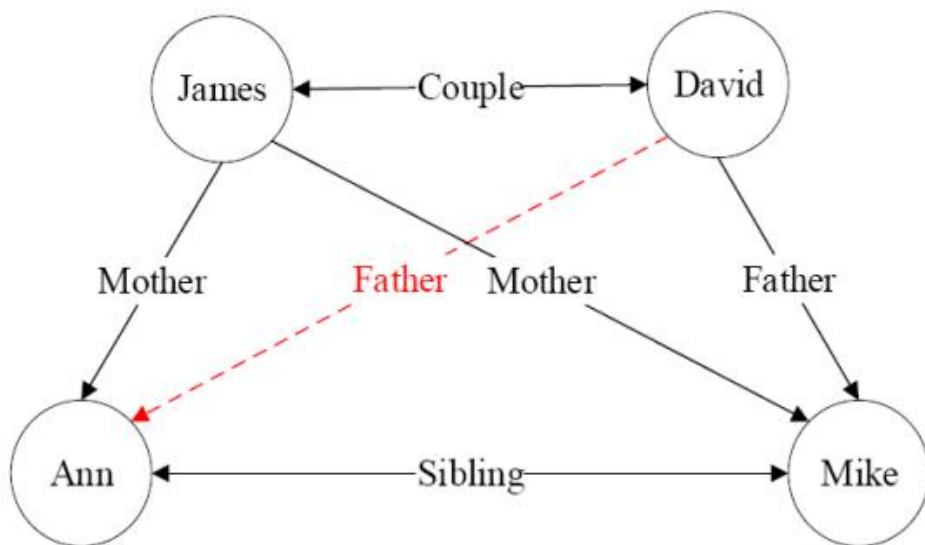
● 目标谓词: Father(x,y)

目标谓词只有一个正例
Father(David, Mike)

反例在知识图谱中一般不会显式给出，但可从知识图谱中构造出来。

只能在已知两个实体的关系且确定其关系与目标谓词相悖时，才能将这两个实体用于构建目标谓词的反例，而不能在不知两个实体是否满足目标谓词前提下将它们来构造目标谓词的反例

归纳逻辑程序设计



一个简单的家庭关系知识图谱

目标谓词	$\text{Father}(\text{David}, \text{Mike})$
训练样例	$\neg \text{Father}(\text{David}, \text{James})$
集合	$\neg \text{Father}(\text{James}, \text{Ann})$
	$\neg \text{Father}(\text{James}, \text{Mike})$
	$\neg \text{Father}(\text{Ann}, \text{Mike})$

背景知识	$\text{Sibling}(\text{Ann}, \text{Mike})$
样例集合	$\text{Couple}(\text{David}, \text{James})$
	$\text{Mother}(\text{James}, \text{Ann})$
	$\text{Mother}(\text{James}, \text{Mike})$

$$(\forall x)(\forall y)(\forall z)(\text{Mother}(z, y) \wedge \text{Couple}(x, z) \rightarrow \text{Father}(x, y))$$

↑
前提约束谓词
(学习得到)

↑
目标谓词
(已知)

推理思路：逐步给目标谓词添加前提约束谓词，直到所构成的推理规则不覆盖任何反例。

归纳逻辑程序设计 —— FOIL (First Order Inductive Learner)

$$(\forall x)(\forall y)(\forall z)(Mother(z, y) \wedge Couple(x, z) \rightarrow \boxed{Father(x, y)})$$



前提约束谓词
(学习得到)



目标谓词
(已知)

哪些谓词好呢？
可以作为目标
谓词的前提约
束谓词？

FOIL信息增益值计算方法如下：

$$FOIL_Gain = \widehat{m}_+ \cdot \left(\log_2 \frac{\widehat{m}_+}{\widehat{m}_+ + \widehat{m}_-} - \log_2 \frac{m_+}{m_+ + m_-} \right)$$

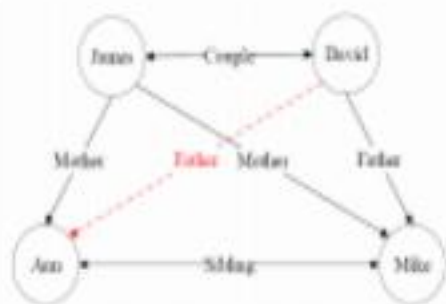
其中， \widehat{m}_+ 和 \widehat{m}_- 是增加前提约束谓词后所得新推理规则覆盖的正例和反例的数量， m_+ 和 m_- 是原推理规则所覆盖的正例和反例数量。

FOIL增益仅考虑正例的信息增益，并以新规则覆盖的正例数为权重

FOIL中信息增益值
(information gain)

归纳逻辑程序设计

$$(\forall x)(\forall y)(\forall z)(Mother(z,y) \wedge Couple(x,z) \rightarrow \boxed{Father(x,y)})$$



↑
前提约束谓词
(学习得到)

↑
目标谓词
(已知)

- $Mother(\cdot, \cdot)$
- $Sibling(\cdot, \cdot)$
- $Couple(\cdot, \cdot)$

依次将谓词加入到推理规则中作为前提约束谓词，并计算所得到新推理规则的FOIL增益值。基于计算所得FOIL增益值来选择最佳前提约束谓词。

归纳逻辑程序设计

推理规则		推理规则涵盖的正例和反例数		FOIL 信息增益值
目标谓词	前提约束谓词	正例	反例	信息增益值
$Father(x, y) \leftarrow$	空集	$m_+ = 1$	$m_- = 4$	$FOIL_Gain$
$Father(x, y) \leftarrow$	$Mother(x, y)$	$m_+ = 0$	$m_- = 2$	NA
	$Mother(x, z)$	$m_+ = 0$	$m_- = 2$	NA
	$Mother(y, x)$	$m_+ = 0$	$m_- = 1$	NA
	$Mother(y, z)$	$m_+ = 0$	$m_- = 1$	NA
	$Mother(z, x)$	$m_+ = 0$	$m_- = 1$	NA
	$Mother(z, y)$	$m_+ = 1$	$m_- = 3$	0.32
	$Sibling(x, y)$	$m_+ = 0$	$m_- = 1$	NA
	$Sibling(x, z)$	$m_+ = 0$	$m_- = 1$	NA
	$Sibling(y, x)$	$m_+ = 0$	$m_- = 0$	NA
	$Sibling(y, z)$	$m_+ = 0$	$m_- = 0$	NA
	$Sibling(z, x)$	$m_+ = 0$	$m_- = 0$	NA

给定目标谓词，此时推理规则只有目标谓词，因此推理规则所覆盖的正例和反例的样本数分别是训练样本中正例和反例的数量，即 1 和 4，因此， $m_+ = 1$ ， $m_- = 4$

背景知识 样例集合	Sibling(Ann, Mike) Couple(David, James) Mother(James, Ann) Mother(James, Mike)	目标谓词 训练样例 集合	Father(David, Mike) \neg Father(David, James) \neg Father(James, Ann) \neg Father(James, Mike) \neg Father(Ann, Mike)
--------------	---	--------------------	---

归纳逻辑程序设计

推理规则		推理规则涵盖的正例和反例数		FOIL 信息增益值
目标谓词	前提约束谓词	正例	反例	信息增益值
$Father(x, y) \leftarrow$	空集	$m_+ = 1$	$m_- = 4$	$FOIL_Gain$
$Father(x, y) \leftarrow$	$Mother(x, y)$	$\hat{m}_+ = 0$	$\hat{m}_- = 2$	NA
	$Mother(x, z)$	$\hat{m}_+ = 0$	$\hat{m}_- = 2$	NA
	$Mother(y, x)$	$\hat{m}_+ = 0$	$\hat{m}_- = 1$	NA
	$Mother(y, z)$	$\hat{m}_+ = 0$	$\hat{m}_- = 1$	NA
	$Mother(z, x)$	$\hat{m}_+ = 0$	$\hat{m}_- = 1$	NA
	$Mother(z, y)$	$\hat{m}_+ = 1$	$\hat{m}_- = 3$	0.32
	$Sibling(x, y)$	$\hat{m}_+ = 0$	$\hat{m}_- = 1$	NA
	$Sibling(x, z)$	$\hat{m}_+ = 0$	$\hat{m}_- = 1$	NA
	$Sibling(y, x)$	$\hat{m}_+ = 0$	$\hat{m}_- = 0$	NA
	$Sibling(y, z)$	$\hat{m}_+ = 0$	$\hat{m}_- = 0$	NA
	$Sibling(z, x)$	$\hat{m}_+ = 0$	$\hat{m}_- = 0$	NA
	$Sibling(z, y)$	$\hat{m}_+ = 1$	$\hat{m}_- = 2$	0.74

将 $Mother(x, y)$ 作为前提约束谓词加入，可得到推理规则
 $Mother(x, y) \rightarrow Father(x, y)$

在背景知识中， $Mother(x, y)$ 有两个实例

- $Mother(\text{James}, \text{Ann})$ $\hat{m}_+ = 0, \hat{m}_- = 1$
- $Mother(\text{James}, \text{Mike})$

对于 $Mother(\text{James}, \text{Ann})$ 这一实例， $x = \text{James}$, $y = \text{Ann}$ ，将 x 和 y 代入 $Father(x, y)$ 得到 $Father(\text{James}, \text{Ann})$ ，可知在训练样本中 $Father(\text{James}, \text{Ann})$ 是一个反例

背景知识 样例集合	Sibling(Ann, Mike)	目标谓词 训练样例 集合	Father(David, Mike)
	Couple(David, James)		$\neg Father(\text{David}, \text{James})$
	Mother(James, Ann)		$\neg Father(\text{James}, \text{Ann})$
	Mother(James, Mike)		$\neg Father(\text{James}, \text{Mike})$
			$\neg Father(\text{Ann}, \text{Mike})$

归纳逻辑程序设计

推理规则		推理规则涵盖的正例和反例数		FOIL 信息增益值
目标谓词	前提约束谓词	正例	反例	信息增益值
$Father(x, y) \leftarrow$	空集	$m_+ = 1$	$m_- = 4$	$FOIL_Gain$
$Father(x, y) \leftarrow$	$Mother(x, y)$	$m_+ = 0$	$m_- = 2$	NA
	$Mother(x, z)$	$m_+ = 0$	$m_- = 2$	NA
	$Mother(y, x)$	$m_+ = 0$	$m_- = 1$	NA
	$Mother(y, z)$	$m_+ = 0$	$m_- = 1$	NA
	$Mother(z, x)$	$m_+ = 0$	$m_- = 1$	NA
	$Mother(z, y)$	$m_+ = 1$	$m_- = 3$	0.32
	$Sibling(x, y)$	$m_+ = 0$	$m_- = 1$	NA
	$Sibling(x, z)$	$m_+ = 0$	$m_- = 1$	NA
	$Sibling(y, x)$	$m_+ = 0$	$m_- = 0$	NA
	$Sibling(y, z)$	$m_+ = 0$	$m_- = 0$	NA
	$Sibling(z, x)$	$m_+ = 0$	$m_- = 0$	NA
	$Sibling(z, y)$	$m_+ = 1$	$m_- = 2$	0.74

背景知识 样例集合	Sibling(Ann, Mike)	目标谓词 训练样例 集合	Father(David, Mike)
	Couple(David, James)		$\neg Father(David, James)$
	Mother(James, Ann)		$\neg Father(James, Ann)$
	Mother(James, Mike)		$\neg Father(James, Mike)$
			$\neg Father(Ann, Mike)$

将 $Mother(x, y)$ 作为前提约束谓词加入，可得到推理规则
 $Mother(x, y) \rightarrow Father(x, y)$

在背景知识中， $Mother(x, y)$ 有两个实例

- $Mother(James, Ann)$ $\hat{m}_+ = 0, \hat{m}_- = 1$
- $Mother(James, Mike)$ $\hat{m}_+ = 0, \hat{m}_- = 1$

对于 $Mother(James, Mike)$ 这一实例， $x = James, y = Mike$ ，将 x 和 y 代入 $Father(x, y)$ 得到
 $Father(James, Mike)$ ，可知在训练样本中 $Father(James, Mike)$ 是一个反例

归纳逻辑程序设计

推理规则		推理规则涵盖的正例和反例数		FOIL 信息增益值
目标谓词	前提约束谓词	正例	反例	信息增益值
$Father(x, y) \leftarrow$	空集	$m_+ = 1$	$m_- = 4$	$FOIL_Gain$
$Father(x, y) \leftarrow$	$Mother(x, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 2$	NA
	$Mother(x, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 2$	NA
	$Mother(y, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Mother(y, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Mother(z, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Mother(z, y)$	$\widehat{m}_+ = 1$	$\widehat{m}_- = 3$	0.32
	$Sibling(x, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Sibling(x, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Sibling(y, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Sibling(y, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Sibling(z, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Sibling(z, y)$	$\widehat{m}_+ = 1$	$\widehat{m}_- = 2$	0.74

$Mother(x, y) \rightarrow Father(x, y)$

覆盖正例和反例数量分别为0和2，即 $\widehat{m}_+ = 0$ ， $\widehat{m}_- = 2$

由于 $\widehat{m}_+ = 0$ ，代入 $FOIL_Gain$ 公式时会出现负无穷的情况，此时 $FOIL_Gain$ 记为NA (Not Available)

背景知识 样例集合	Sibling(Ann, Mike)	目标谓词 训练样例 集合	Father(David, Mike)
	Couple(David, James)		$\neg Father(David, James)$
	Mother(James, Ann)		$\neg Father(James, Ann)$
	Mother(James, Mike)		$\neg Father(James, Mike)$
			$\neg Father(Ann, Mike)$

FOIL信息增益值计算方法如下：

$$FOIL_Gain = \widehat{m}_+ \cdot \left(\log_2 \frac{\widehat{m}_+}{\widehat{m}_+ + \widehat{m}_-} - \log_2 \frac{m_+}{m_+ + m_-} \right)$$

其中， \widehat{m}_+ 和 \widehat{m}_- 是增加前提约束谓词后所得新推理规则覆盖的正例和反例的数量， m_+ 和 m_- 是原推理规则所覆盖的正例和反例数量。

归纳逻辑程序设计

推理规则		推理规则涵盖的正例和反例数		FOIL 信息增益值
目标谓词	前提约束谓词	正例	反例	信息增益值
$Father(x, y) \leftarrow$	空集	$m_+ = 1$	$m_- = 4$	$FOIL_Gain$

背景知识 样例集合	Sibling(Ann, Mike)	目标谓词 训练样例 集合	Father(David, Mike)
	Couple(David, James)		$\neg Father(David, James)$
	Mother(James, Ann)		$\neg Father(James, Ann)$
	Mother(James, Mike)		$\neg Father(James, Mike)$
			$\neg Father(Ann, Mike)$

$Father(x, y) \leftarrow$	$Sibling(x, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Sibling(y, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Sibling(y, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Sibling(z, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Sibling(z, y)$	$\widehat{m}_+ = 1$	$\widehat{m}_- = 2$	0.74
	$Couple(x, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Couple(x, z)$	$\widehat{m}_+ = 1$	$\widehat{m}_- = 1$	1.32
	$Couple(y, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Couple(y, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Couple(z, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 2$	NA
	$Couple(z, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA

如果将 $Couple(x, z)$ 作为前提约束谓词加入, 可得到如下推理规则
 $Couple(x, z) \rightarrow Father(x, y)$

在背景知识中, $Couple(x, z)$ 只有一个实例 $Couple(David, James)$, 即 $x=David, z=James$, 将其代入 $Father(x, y)$ 得到 $Father(David, y)$ 。

在训练样本中存在正例 $Father(David, Mike)$ 以及反例 $\neg Father(David, James)$, 即 $Couple(x, z) \rightarrow Father(x, y)$ 覆盖正例和反例数量分别为1和1。信息增益值为:

$$\begin{aligned}
 & \widehat{m}_+ \cdot \left(\log_2 \frac{\widehat{m}_+}{\widehat{m}_+ + \widehat{m}_-} - \log_2 \frac{m_+}{m_+ + m_-} \right) \\
 &= 1 \cdot \left(\log_2 \frac{1}{1+1} - \log_2 \frac{1}{1+4} \right) \\
 &= 1.32
 \end{aligned}$$

归纳逻辑程序设计

推理规则		推理规则涵盖的正例和反例数		FOIL 信息增益值
目标谓词	前提约束谓词	正例	反例	信息增益值
$Father(x, y) \leftarrow$	空集	$m_+ = 1$	$m_- = 4$	$FOIL_Gain$
$Father(x, y) \leftarrow$	$Mother(x, y)$	$\hat{m}_+ = 0$	$\hat{m}_- = 2$	NA
	$Mother(x, z)$	$\hat{m}_+ = 0$	$\hat{m}_- = 2$	NA
	$Mother(y, x)$	$\hat{m}_+ = 0$	$\hat{m}_- = 1$	NA
	$Mother(y, z)$	$\hat{m}_+ = 0$	$\hat{m}_- = 1$	NA
	$Mother(z, x)$	$\hat{m}_+ = 0$	$\hat{m}_- = 1$	NA
	$Mother(z, y)$	$\hat{m}_+ = 1$	$\hat{m}_- = 3$	0.32
	$Sibling(x, y)$	$\hat{m}_+ = 0$	$\hat{m}_- = 1$	NA
	$Sibling(x, z)$	$\hat{m}_+ = 0$	$\hat{m}_- = 1$	NA
	$Sibling(y, x)$	$\hat{m}_+ = 0$	$\hat{m}_- = 0$	NA
	$Sibling(y, z)$	$\hat{m}_+ = 0$	$\hat{m}_- = 0$	NA
	$Sibling(z, x)$	$\hat{m}_+ = 0$	$\hat{m}_- = 0$	NA
	$Sibling(z, y)$	$\hat{m}_+ = 1$	$\hat{m}_- = 2$	0.74
	$Couple(x, y)$	$\hat{m}_+ = 0$	$\hat{m}_- = 1$	NA
	$Couple(x, z)$	$\hat{m}_+ = 1$	$\hat{m}_- = 1$	1.32
	$Couple(y, x)$	$\hat{m}_+ = 0$	$\hat{m}_- = 0$	NA
	$Couple(y, z)$	$\hat{m}_+ = 0$	$\hat{m}_- = 0$	NA
	$Couple(z, x)$	$\hat{m}_+ = 0$	$\hat{m}_- = 2$	NA
	$Couple(z, y)$	$\hat{m}_+ = 0$	$\hat{m}_- = 1$	NA

- $Couple(x, z)$ 加入后信息增益最大
- 将 $Couple(x, z)$ 加入推理规则，得到 $Couple(x, z) \rightarrow Father(x, y)$ 新推理规则
- 将训练样例中与该推理规则不符的样例去掉。
这里不符指当 $Couple(x, z)$ 中 x 取值为 David 时，与 $Father(David, _)$ 或 $\neg Father(David, _)$ 无法匹配的实例。
- 训练样本集中只有正例 $Father(David, Mike)$ 和负例 $\neg Father(David, James)$ 两个实例

Back-ground knowledge	Sibling(Ann, Mike) Couple(David, James) Mother(James, Ann) Mother(James, Mike)
Positive and negative samples	Father(David, Mike) $\neg Father(David, James)$ $\neg Father(James, Ann)$ $\neg Father(James, Mike)$ $\neg Father(Ann, Mike)$

归纳逻辑程序设计

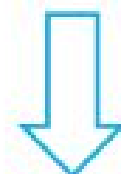
推理规则		推理规则涵盖的 正例和反例数		FOIL 信息增益值
现有规则	拟加入前提 约束谓词	正例	反例	信息增益值
$Father(x, y) \leftarrow Couple(x, z)$		$m_+ = 1$	$m_- = 1$	1.32
	$\wedge Mother(x, y)$	$\overline{m}_+ = 0$	$\overline{m}_- = 0$	NA
	$\wedge Mother(x, z)$	$\overline{m}_+ = 0$	$\overline{m}_- = 0$	NA
	$\wedge Mother(y, x)$	$\overline{m}_+ = 0$	$\overline{m}_- = 0$	NA
	$\wedge Mother(y, z)$	$\overline{m}_+ = 0$	$\overline{m}_- = 0$	NA
	$\wedge Mother(z, x)$	$\overline{m}_+ = 0$	$\overline{m}_- = 0$	NA
	$\wedge Mother(z, y)$	$\overline{m}_+ = 1$	$\overline{m}_- = 0$	1
	Back-ground knowledge	Sibling(Ann, Mike)		NA
		Couple(David, James)		NA
		Mother(James, Ann)		NA
		Mother(James, Mike)		NA
				NA
	Positive and negative samples	Father(David, Mike)		NA
		$\neg Father(David, James)$		0
		$\neg Father(James, Ann)$		NA
		$\neg Father(James, Mike)$		NA
		$\neg Father(Ann, Mike)$		NA

- $Mother(z, y)$ 加入信息增益最大
- 将 $Mother(z, y)$ 加入，得到新推理规则

$Mother(z, y) \wedge Couple(x, z)$

$\rightarrow Father(x, y)$

- 当 $x=David$ 、 $y=Mike$ 、 $z=James$ 时，该推理规则覆盖训练样本集中正例 $Father(David, Mike)$ 且不覆盖任意反例，因此算法学习结束。



已知：

$Mother(James, Ann)$

$Couple(David, James)$

于是： $Father(David, Ann)$

归纳逻辑程序设计

$$(\forall x)(\forall y)(\forall z)(Mother(z, y) \wedge Couple(x, z) \rightarrow Father(x, y))$$



前提约束谓词
(学习得到)



目标谓词
(已知)

推理手段: *positive examples + negative examples + background knowledge examples* \Rightarrow **hypothesis**

背景知识 样例 集合	Sibling(Ann, Mike)	目标谓词 训练样例 集合	Father(David, Mike)
	Couple(David, James)		\neg Father(David, James)
	Mother(James, Ann)		\neg Father(James, Ann)
	Mother(James, Mike)		\neg Father(James, Mike)
			\neg Father(Ann, Mike)

给定目标谓词，FOIL算法从实例（正例、反例、背景样例）出发，不断测试所得到推理规则是否还包含反例，一旦不包含负例，则学习结束，展示了“**归纳学习**”能力。

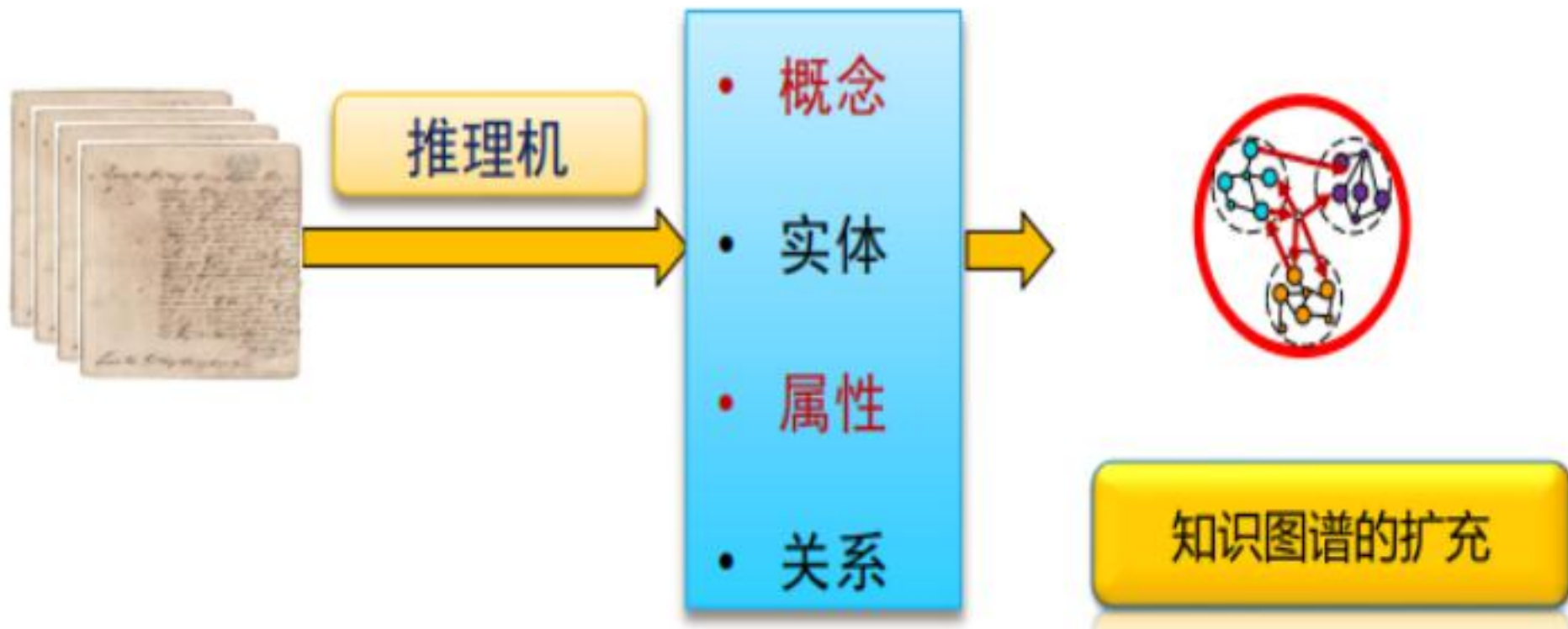
归纳逻辑程序设计

- ❑ **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：

FOIL算法	
输入：	目标谓词 P ， P 的训练样例（正例集合 E^+ 和反例集合 E^- ），其他背景知识
输出：	推导得到目标谓词 P 的推理规则
1	将目标谓词作为所学习推理规则的结论
2	将其他谓词逐一作为前提约束谓词加入推理规则，计算所得到推理规则的FOIL信息增益值，选取最优前提约束谓词以生成新推理规则，并将训练样例集合中与该推理规则不符的样例去掉
3	重复2过程，直到所得到的推理规则不覆盖任意反例

$$\text{FOIL_Gain} = \hat{m}_+ \cdot \left(\log_2 \frac{\hat{m}_+}{\hat{m}_+ + \hat{m}_-} - \log_2 \frac{m_+}{m_+ + m_-} \right)$$

知识图谱中的知识推理



知识图谱中的知识推理

■ 归纳推理：学习推理规则

➤ 归纳逻辑程序设计

➤ 关联规则挖掘

➤ 路径排序算法（见吴飞教授的MOOC）

■ 演绎推理：推理具体事实

➤ 马尔可夫逻辑网

➤ 概率软逻辑

■ 基于分布式表示的推理

➤ 表示学习技术

➤ 张量分解技术

符号推理：在知识图谱中的实体和关系符号上直接进行推理。

<https://www.icourse163.org/learn/ZJU-1003377027?tid=1450309465#/learn/content?type=detail&id=1214618868>

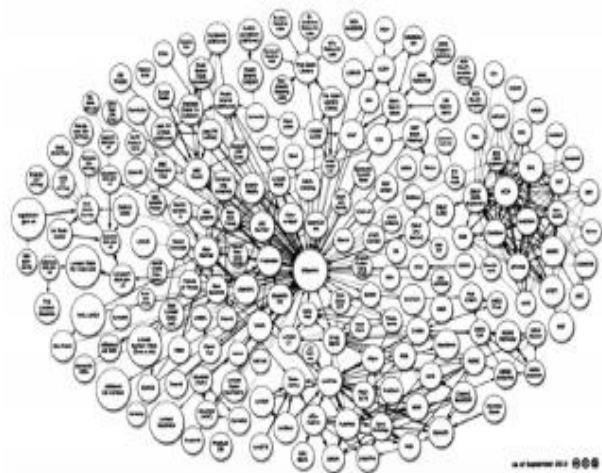
数值推理：使用数值计算，尤其是向量矩阵计算的方法，捕捉知识图谱上隐式的关联，模拟推理的进行。

- 实体关系表示（向量/矩阵/张量）
- 打分函数定义（距离函数/相似度函数）
- 表示学习（开放世界假设/封闭世界假设）

基于分布式表示的推理

基于离散符号的知识表示

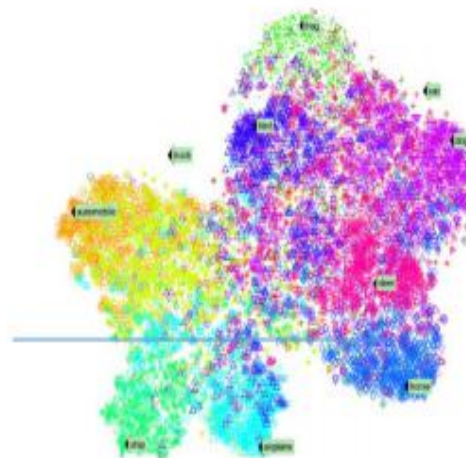
RDF, OWL, 各种Rule Language等



VS

基于连续向量的知识表示

Tensor, 各种Embedding, 神经网络表示等



显式知识、强逻辑约束、易于解释、推理不易扩展

隐式知识、弱逻辑约束、不易解释、对接神经网络

回顾：词嵌入->知识图谱嵌入

- Word embedding -> knowledge graph embedding
- 为知识图谱中的每个实体和关系都学习一个向量表示
- 将离散的符号表示转换为连续的向量表示

$C(\text{king}) - C(\text{queen}) \approx C(\text{man}) - C(\text{woman})$
或 $C(\text{king}) - C(\text{man}) + C(\text{woman}) \approx C(\text{queen})$
其中， $C(w)$ 就是word2vec学习到的词向量表示

KG Embedding/知识图谱嵌入

=

Entity Vector & Relation Vectors/实体向量和关系向量

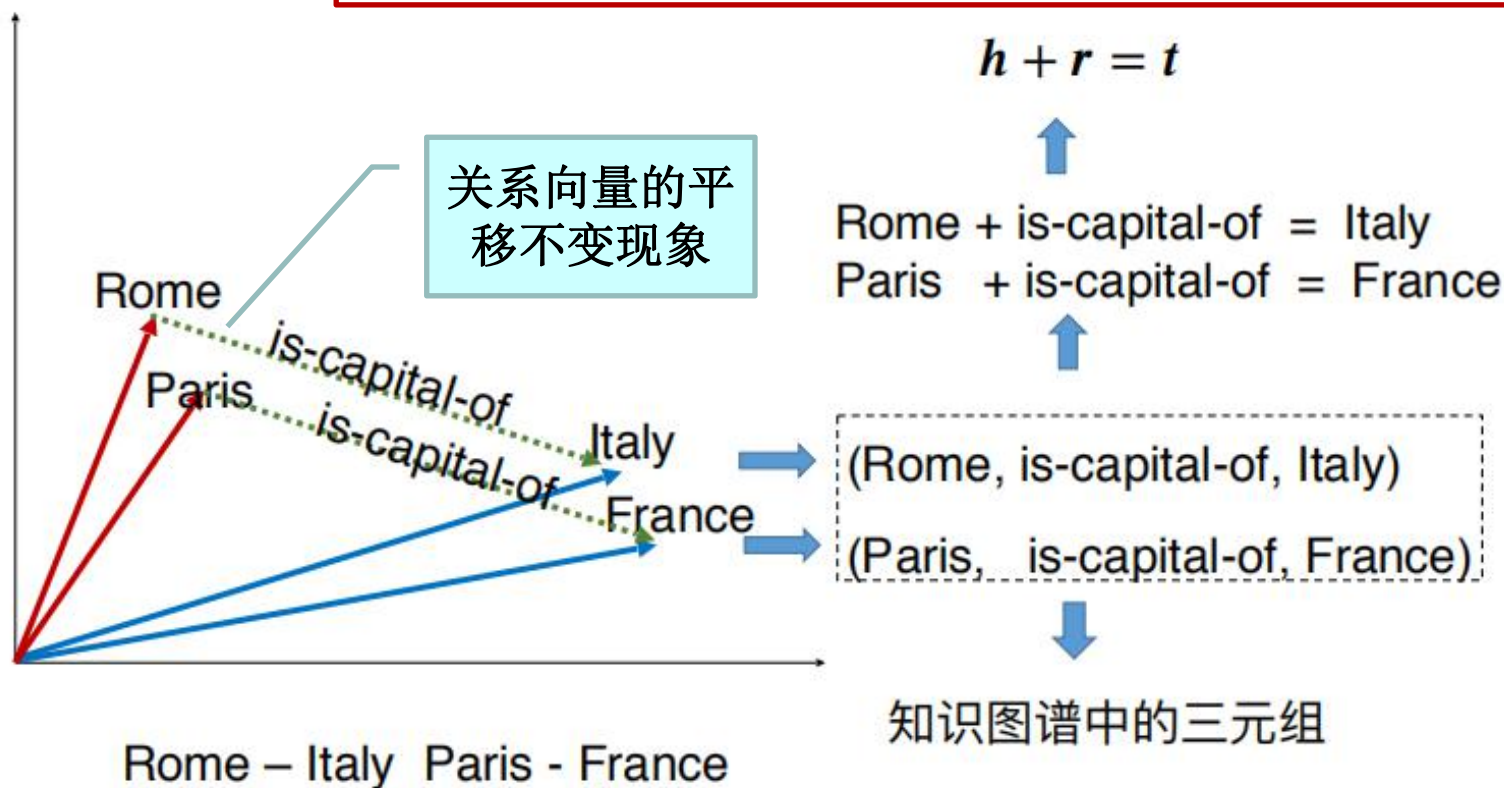
=

Distributed Representations/知识图谱的分布式表示

知识图谱嵌入模型TransE

- TransE(Translating Embeddings for Modeling Multi-relational Data. NIPS 2013)

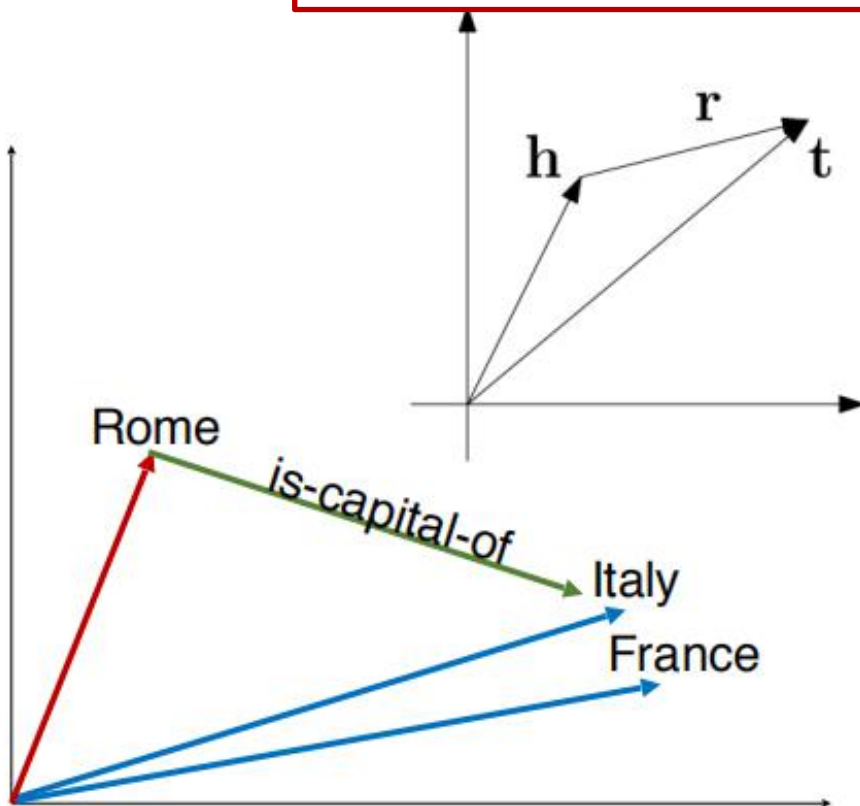
关系向量 r 看作为头实体向量 h 和尾实体向量 t 之间的平移
也可以将 r ,看作从 h 到 t , 的翻译, 因此**TransE**也被称为翻译模型



知识图谱嵌入模型TransE

● TransE

关系向量 r 看作为头实体向量 h 和尾实体向量 t 之间的平移
也可以将 r 看作从 h 到 t 的翻译，因此TransE也被称为翻译模型



对每个三元组(h,r,t)的优化目标:

每个三元组的目标函数:

$$f_r(h, t) = \|h + r - t\|_{L_1/L_2}$$

知识图谱嵌入模型TransE

● TransE 损失函数

在模型的训练过程中，transE采用最大间隔方法，最小化目标函数：

$$L = \sum_{(h,r,t) \in S} \sum_{(h',r,t') \in S'} \max \left(0, \underset{\substack{\uparrow \\ \text{(h,r,t)的score}}}{f_r(h,t)} + \underset{\substack{\uparrow \\ \text{间隔}}}{\gamma} - \underset{\substack{\uparrow \\ \text{(h,r,t) 的负样本的score}}}{f_r(h',t')} \right).$$

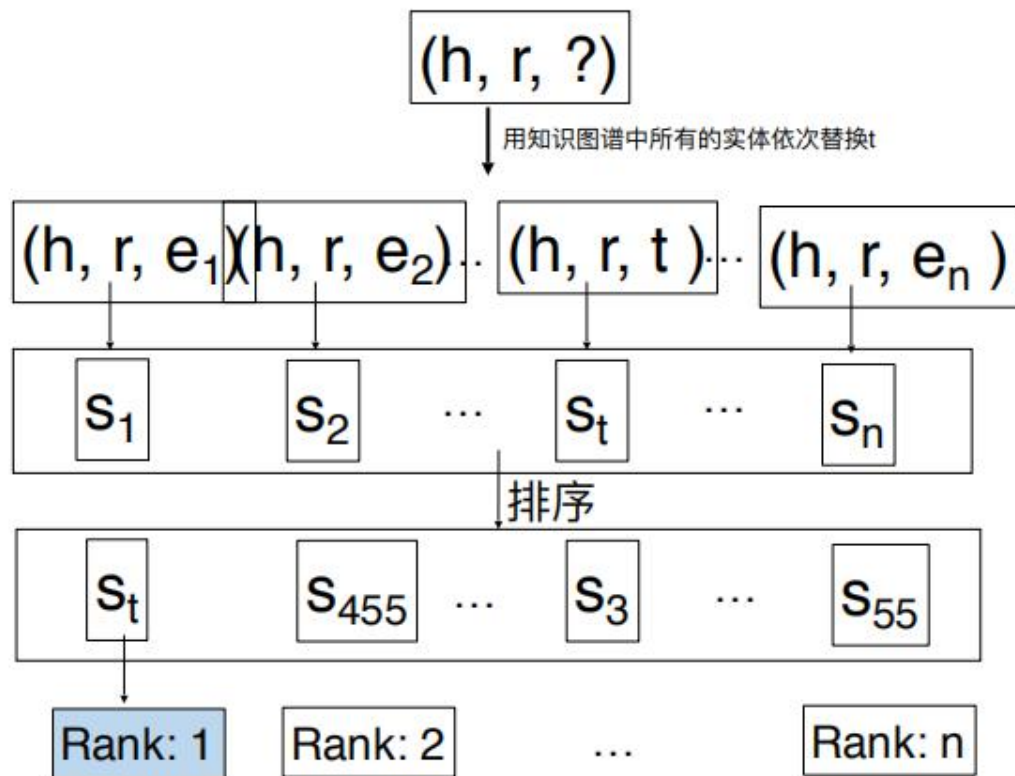
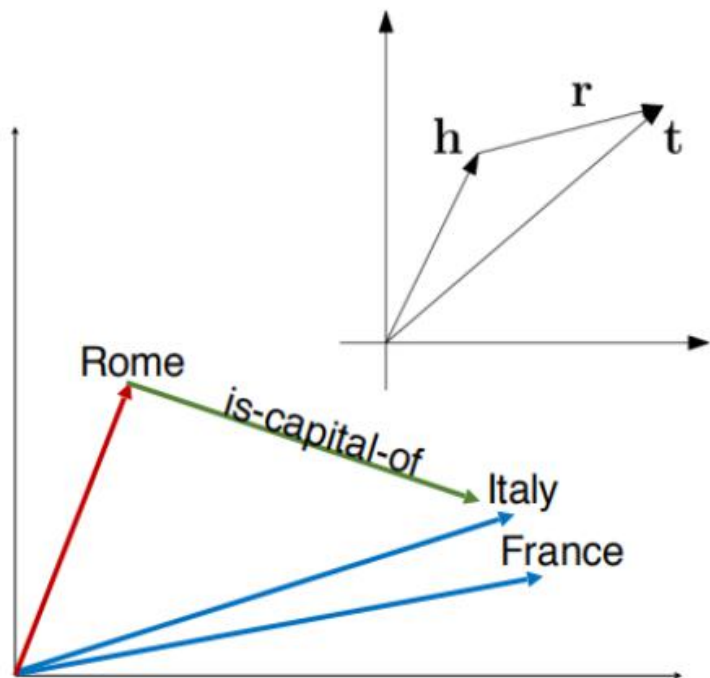
S 是知识库中的三元组即训练集，

S' 是负采样的三元组，通过替换 h 或 t 所得，是人为随机生成的

关于模型的参数：设一共有 $|E|$ 个实体和 $|R|$ 个关系，每个实体/关系的向量长度为 d 维，因此，一共有 $(|E| + |R|) * d$ 个参数。

知识图谱嵌入模型TransE

- 测试三元组 (h, r, t)
- 尾实体预测 $(h, r, ?)$
- 头实体预测 $(?, r, t)$



更加细分的知识图谱技术要素

知识图谱技术点



针对单关系问题的知识图谱问答系统

➤ 下载知识图谱文件:

- 英文问答任务: 谷歌的Freebase知识图谱 (<http://developers.google.com/freebase>)
- 中文问答任务: NLPCC的中文知识图谱 (<http://pan.baidu.com/s/1dEYcQXz>)

➤ 下载问答数据集:

- 英文问答任务: SimpleQuestions数据集 (<https://research.fb.com/downloads/babi/>)
- 中文问答任务: NLPCC-KBQA数据集

➤ 实现问答方法: 实体链接、关系分类

语义相似度建模工具包: DSSM

(<http://www.microsoft.com/en-us/research/project/dssm/>)



不断发展的知识图谱技术

Noisy Data



Unstructured Data



Structured Data



Multimedia



IoT Sensors



Crowds

趋势一：从离散的符号知识表示到连续的向量知识表示的发展变化

趋势二：融合强化学习、对抗学习等实现弱监督、少样本的关系发现与语义融合

趋势三：在连续的向量空间实现大规模的可微分推理

趋势四：将知识图谱作为先验知识融入深度学习模型，提升模型可解释性。

Quality Knowledge

