SEU 知识建模

2019年11月8日 20:50

- **一、本体（ontology）**

  本体：领域共享知识的描述方式，是语义Web、语义搜索、知识工程和很多人工智能应用的基础。

  什么是本体?

  本体是我们告诉计算机人类如何认识和理解世界万物的一种形式化描述方式。

  本体成为语义网的知识描述载体。语义网的研究热潮极大的促进了本体的研究。

- **二、知识建模方法**
- **本体工程**

  知识图谱中需要一个本体来形式化描述和界定他所描述的知识和事实的范围。

  本体工程是用工程化范围保证本体质量的方法学。

- **知识图谱本体VS数据库模式**

	知识图谱的Ontology	数据库的Schema
用途	• 共享知识和事实 • 交互、搜索、辅助AI	• 结构化数据组织和管理 • 查询
语法	逻辑为基础的形式化语法	ER图
语义	• 定义概念和关系 • 概念层次 • 丰富语义，支持推理 • 一致性，完备性	• 定义结构化数据 无概念层次 • 无(少)语义，不支持推理 • 主外键
规模	规模可以很大	规模一般较小

- **本体构建工程（手工）**
  - **Step 1：确定本体的领域和范围**

    明确基本问题：本体针对什么领域？ / 用途是什么？ / 描述什么信息? / 回答哪一类的问题？ / 谁将维护这个本体?

  - **Step 2：考虑重用现有的本体**

    收集和待开发本体相关的其他本体是有价值的。

    目前网络上已有一些本体库，从中可以获得很多现有的本体。

- **列出本体中重要的术语**

  这些术语能保证最终创建的本体不会偏离所感兴趣的领域

  这些术语大致表明建模过程中所感兴趣的事物、物所具有的属性和他们之间的关系等。

- **Step 4：定义类和类的继承**
  - 确保类的集成正确
  - 分析继承结构中的兄弟类
  - 引入新类的时机
  - 新类或属性值的取舍
  - 实例或类的取舍
  - 范围限制
  - 不相交的子类

  类的集成结构的定义可以采用自顶向下的方法，即从最大的概念开始，添加子类细化；

  也可以自底向上的方法，由最底层、最细的类定义开始，找到他们的父类；

  - **确保类的正确**
    - 类继承关系通常表示为“Is-a”、“Kind-of”，如果A是B的子类，则A的所有实例也是B的实例。
    - 类的继承关系是可传递的
    - 类代表着领域内的概念
    - 类的继承体系中要避免发生循环。如果A中有B，B中又有A。在发生类循环的时候，等于声明环路上所有的类都是等价的，这就造成了本体的冗余。
  - **分析继承结构中的兄弟类**
    - 同一类的全部直接子类之间称为兄弟类
    - 兄弟类的数目过多或者过少都不合适
  - **引入类的新时机**

  子类通常具有特有的且父类没有的属性，或者拥有不同于父类的限制条件，或者和父类相比较能作用于不同的关系之中。
    - 以上，当本体概念无法满足这些内容时，可以在继承体系中引入一个新类。
    - 没有必要为每一个限制条件创建新类，需要在创建新的类组织是否有用与创建更多的类之间进行平衡。
  - **新类或属性值的取舍**

    判断如何处理一些特殊区别点，到底是将他作为属性赋值，还是需要构建一个新类。
  - **实例或类的取舍**

    决定本体中一个特定的名词是一个类或一个实例，这取决于本体潜在应用。
  - **范围限制**

本体的定义应该尽可能完整，但并不是说要一定包含领域内的所有信息。

- **不相交的子类**
  很多系统要求明确说明一些类之间互不相交。若多个类之间不包含任何相同的实例，那么他们就是不相交的。

- **Step 5：定义属性和关系**
  定义了类还需要定义概念和概念之间的联系。
  这里所指的联系可以分为两种：
  - 一种是概念自身的属性，称为内在属性。即一个类具有内在属性，那么他所有的子类都继承了这种属性。
  - 另一种称为外在属性，同常用于链接概念间的实例
  - **逆属性/关系**
    一对属性或者关系互逆，当且仅当其定义域与值域互换。
  - **缺省属性值**
    类中的大部分实例都有着一个相同的属性值，那么就把这个赋值定义为这个属性的缺省值

- **Step 6：定义属性的限制**
  进一步定义属性的一些限制，包括属性的基数、属性值的类型，以及属性的定义域和值域

- **Step 7：创建实例**
  确定与个体最接近的类，然后添加个体进去作为该类的一个实例，同时要为实例的属性赋值。

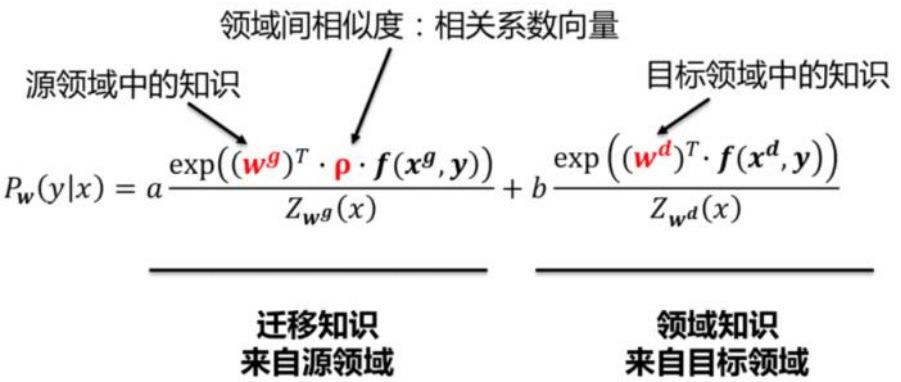
- 其他的要求：
  - 在本体中规定合理的命名规则并严格遵循
  - 命名规则一旦制定就要严格遵循
  - 命名规则要考虑系统对大小写的敏感性、单复数、前后缀和分隔符的使用问题
  - 对于概念名要尽量避免使用缩写

- **本体学习 （自动）**
  - **方法一：基于规则的本体学习**
    - 人工写模板规则抽取本体
    - 优点：利用专家知识写抽取模板
    - 缺点：规则不足、规则冲突、不好扩展
  - **方法二：基于机器学习的本体学习**
    - 将本体学习转化为机器学习中的分类或序列标注问题
    - 选取特征和学习模型，进行训练，学习得到本体
    - 优点：效率高，自动化
    - 缺点：学习模型通用性和学习效果之间存在矛盾

- 1、提出了一种基于最大熵的迁移学习模型
  结合迁移学习框架提出了基于最大熵的迁移学习模型，解决了传统大熵模型的领域依赖性
- 2、提出了一种机遇迁移学习的通用本体学习模型
  通过知识迁移，模型预测出结果F的值平均提升了0.03，平均达到0.86；
  错误率平均降低了0.03，平均达到了0.11
- 3、提出了一种基于视觉树的本体生成算法
  自上而下的对is-a进行寻找，F值平均到达了0.9022
  自下而上的使用集合划分思想对subclass-of关系进行寻找，F值平均达到0.7224

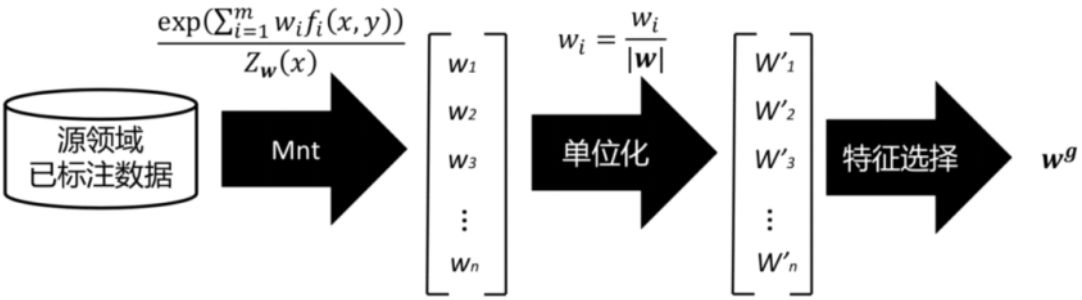
- **基于最大熵的迁移学习模型**

模型核心思想：



参数估计：

w^g ：



ρ ：

刻画相关性，协方差：$Cov(X,Y) = E[(X-\mu)(Y-\varphi)]$

相关系数：$\rho_{X,Y} = \frac{Cov(X,Y)}{\sqrt{D(X)D(Y)}}$

标准化因子 ，方差：$D()$

相关系数的性质：

- $|\rho_{X,Y}| \leq 1$
- $\rho_{X,Y}>0$: X,Y线性正相关
- $\rho_{X,Y}<0$: X,Y线性负相关
- $\rho_{X,Y}=0$: X,Y线性无关
- $|\rho_{X,Y}|$越大，线性相关的程度越大

$\boldsymbol{w^d}$：对于模型的变化

$$\boldsymbol{w^d} = (w_1^d, w_2^d, \dots, w_n^d)^T \longrightarrow \boldsymbol{w^d + \delta} = (w_1^d + \delta_1, w_2^d + \delta_2, \dots, w + \delta_n)^T$$

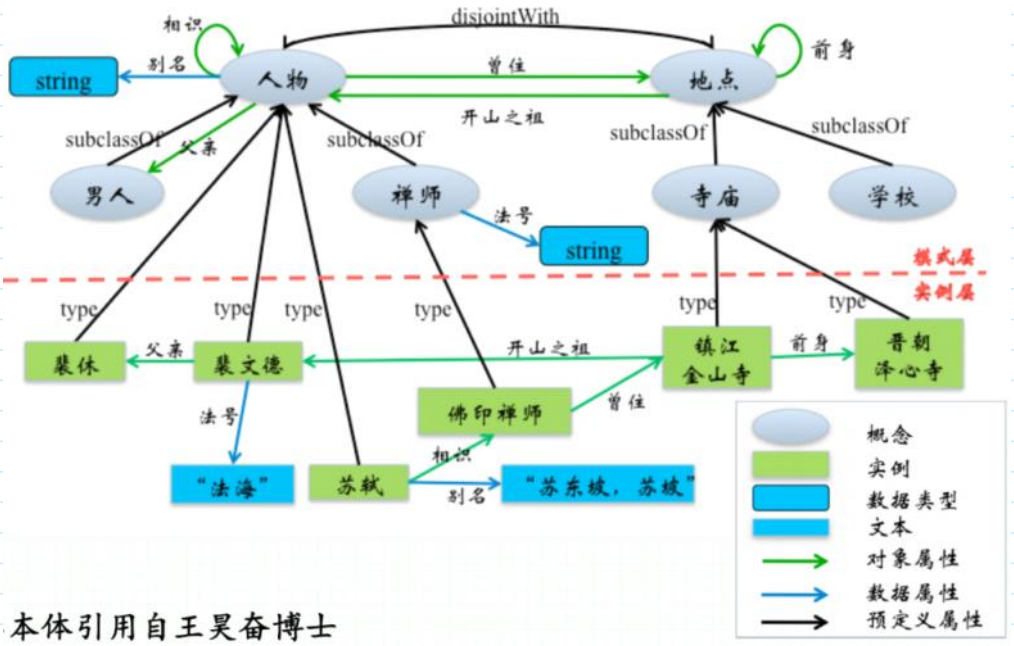
似然函数的变化为：其中使用了$\log(a+b) \geq \log(2\sqrt{ab}), a>0, b>0$做不式放缩

$$L(\boldsymbol{w^d} + \boldsymbol{\delta}) - L(\boldsymbol{w^d}) = \sum_{x,y} \tilde{P}(x,y)\, P_{\boldsymbol{w^d}+\boldsymbol{\delta}}(y|x) - \sum_{x,y} \tilde{P}(x,y)\, P_{\boldsymbol{w^d}}(y|x)$$

$$\geq \sum_{x,y} \tilde{P}(x,y) log \left[2 \sqrt{a \frac{exp((\boldsymbol{w^g})^T \cdot \boldsymbol{\rho} \cdot \boldsymbol{f}(\boldsymbol{x^g}, \boldsymbol{y}))}{Z_{\boldsymbol{w^g}}(x)} * b \frac{exp((\boldsymbol{w^d} + \boldsymbol{\delta})^T \cdot \boldsymbol{f}(\boldsymbol{x^d}, \boldsymbol{y}))}{Z_{\boldsymbol{w^d}+\boldsymbol{\delta}}(x)}} \right]$$

$$- \sum_{x,y} \tilde{P}(x,y)\, P_{\boldsymbol{w^d}}(y|x)$$

- 基于**Protege（建模工具）**的知识建模实例



本体引用自王昊奋博士

- **Protege功能**
  - **类建模**:
    提供图形化界面来建模类（领域概念）和他们的属性及关系。
  - **实例编辑**
    Protege自动昌盛交互式的形式，全用户或领域专家进行有效实例编辑成为可能
  - **模型处理**
    提供插件库，可以定义语义、解答询问以及定义逻辑行为
  - **模型交换**
    最终的模型（类和实例）能以各种各样的格式被装载和保存

- **领域知识建模的工业实践**
  - Step 1：知识背景
  - Step 2：知识重用
  - Step 3：本体设计
  - Step 4：领域专家优化
  - Step 5：本体实现
- **知识图谱中的知识建模总结**
  - 知识图谱中的包含层和实例层，本体通常手工构建，实例通常自动化抽取
  - 构建本体的目的是为了确定知识图谱能描述的知识
  - 不一定需要本体的形式化表示
  - 不一定需要Protege的专业的建模软件
  - 不一定要把本体存储在数据库，可以在程序中直接使用
  - 本体是给知识图谱实施者使用，通过它来确定知识抽取范围、推理规则、构造查询等