

SEU 知识表示

2019年10月30日 14:38

• 一、知识表示概念

• 什么是知识表示？

○ 知识图谱技术体系

应用层	语义搜索	智能问答	辅助决策			
服务层	知识存储	知识检索	知识推理	知识演化		
知识层	知识图谱	知识抽取	知识融合	知识众包	知识表示	知识建模
数据层	非结构化数据	结构化数据库	半结构化数据库	多媒体数据	多模态数据	

○ 知识表示

▪ 认知科学

关系到人类如何储存和处理资料

▪ 人工智能

主要目标为存储知识，让程序能够处理，达到人类的智慧。

目前这个领域仍然没有一个完美的答案。

□ 人工智能角度的知识表示：

- ◆ 人类如何表示知识
- ◆ 知识的本质是什么？我们如何表示它？
- ◆ 表示法应该只用在某个领域？或者通用于所有领域？
- ◆ 某种知识表达方案的表现能力或者说表现力如何？
- ◆ 表示法应该是描述性的，或者是程序性的？

▪ 知识表示是将世界的信息表示为符合机器处理的模式

用于模拟人对世界的认知和推理，以解决人工智能中的复杂任务

▪ 一种知识表示分类观点

- 基于非逻辑的知识表示
- 基于数理逻辑的知识表示
- 基于统计学习的分布式知识表示

▪ 知识表示的发展

语义网络	semantic network
产生式规则	production rules & expert system
框架系统	frame system
描述逻辑	description logic

语义网	Semantic web
统计表示	knowledge Graph representation learning

○ 语义网络

▪ 节点

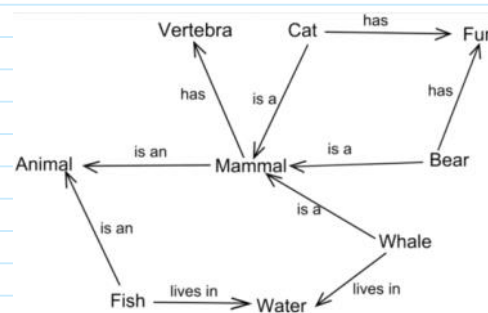
表示各种事物、概念、情况、属性、动作、状态等。每个节点可以带有若干属性

▪ 边

表示各种语义联系，指明他所链接的节点之间的某种语义关系

```
(defun *database* ()
  '((canary (is-a bird)
            (color yellow)
            (size small))
    (penguin (is-a bird)
             (movement swim))
    (bird (is-a vertebrate)
          (has-part wings)
          (reproduction egg-laying))))
```

语义网络的LISP表示例子



语义网络的图表示例子

▪ 语义网络优点

- 标识自然 易于理解 应用广泛
- 符合人类联想记忆
- 结构化知识表示

▪ 缺点

- 不严格：没有工人的逻辑基础
- 难有效处理：网络形式具有灵活的特点，但同时造成了处理和检索的低效率

○ 生产式系统 (production systems)

一个产生式系统的行为规则的集合。

一条规则包含前提 (IF) 和动作 (THEN) 两部分

动物分类的产生式系统

R1: 动物有毛 \rightarrow 哺乳类
 R2: 动物产奶 \rightarrow 哺乳类
 R3: 动物有羽毛 \rightarrow 鸟类
 R4: 动物会飞 \wedge 会下蛋 \rightarrow 鸟类
 R5: 哺乳类 \wedge 动物吃肉 \rightarrow 食肉类
 R6: 动物有犬齿 \wedge 有爪 \wedge 眼盯前方 \rightarrow 肉食类
 R7: 哺乳类 \wedge 有蹄 \rightarrow 蹄类动物
 R8: 哺乳类 \wedge 反刍 \rightarrow 蹄类
 R9: 食肉类 \wedge 黄褐色 \wedge 有斑点 \rightarrow 金钱豹
 R10: 食肉类 \wedge 黄褐色 \wedge 有黑色条纹 \rightarrow 虎
 R11: 蹄类 \wedge 长脖 \wedge 长腿 \wedge 有斑点 \rightarrow 长颈鹿
 R12: 蹄类 \wedge 有黑色条纹 \rightarrow 斑马
 R13: 鸟类 \wedge 长脖 \wedge 长腿 \wedge 不会飞 \rightarrow 鸵鸟
 R14: 鸟类 \wedge 会游泳 \wedge 黑白二色 \wedge 不会飞 \rightarrow 企鹅
 R15: 鸟类 \wedge 善飞 \rightarrow 信天翁

▪ 优点

- 自然性
- 模块性
- 有效性

能表示确定性知识、不确定性知识、启发性知识、过程性知识等

- 清晰性

▪ 不足

- 效率不高：代价高

- 不能表达具有结构性的知识：不能把具有结构关系的事物之间的区别与联系表示出来

○ 框架系统 (frame systems)

人们对现实世界的各种事物的认知都是在以一种类似于框架的结构存储在记忆中

▪ 框架

框架是知识表示的基本单位

描述对象(事物 事件 概念等)属性的数据结构。一个框架由若干个槽 (slot) 结构组成
每个槽又可以分为若干侧面 (facet)

□ 槽

描述某一方面特性

□ 侧面

描述相应属性的一个方面，通常是一个属性支持

▪ E.G.关于男孩的框架

Slot	Value	Type
BOY	—	(This Frame)
ISA	Person	(parent frame)
SEX	Male	(instance value)
AGE	Under 12 yrs.	(procedural attachment - sets constraint)
HOME	A Place	(frame)
NUM_LEGS	Default = 2	(default, inherited from Person frame)

▪ E.G.关于猴子的框架

Slot	Value	Type
MONKEY	—	(This Frame)
ISA	Primate	(parent frame)
SEX	OneOf(Male,Female)	(procedural attachment)
AGE	an integer	(procedural attachment - sets constraint)
HABITAT	Default = Jungle	(default)
FAVORITE_FOOD	Default = Bananas	(default)
CLIMBS	Trees	—
BODY_TYPE	Default = Wiry	(default)
NUM_LEGS	Default = 2	(default)

▪ 优点

- 框架对于知识的描述非常完整和全面
- 基于框架的知识库质量非常高
- 框架允许数值计算 优于当时其他语言表示

▪ 不足

- 框架构建的成本比较高 对知识库的质量要求非常高
- 框架的表达形式不灵活，很难通其他形式的数据联合使用

○ 概念图

一阶逻辑的图接口 基于图的知识表示和推理模型

○ 描述逻辑 (description logic)

- 有断言、量化和公理的逻辑系统；
- 很多一阶逻辑是可靠和完备的演绎系统；

对于每个a，若a为哲学家，则a为学者

$$\forall a(\text{Phil}(a) \rightarrow \text{Schol}(a))$$

存在一些人可以在任何时间被愚弄

$$\exists x(\text{Person}(x) \wedge \forall y(\text{Time}(y) \rightarrow \text{Canfool}(x, y)))$$

▪ Horn子句

- 由很多个肯定的析取构成
- 一阶逻辑的子集
- Prolog语言的逻辑基础

在人工智能早期作为自动定理证明

Horn子句(Horn Clause)

	Disjunction form	Implication form
Definite clause	$\neg p \vee \neg q \vee \dots \vee \neg t \vee u$	$u \leftarrow p \wedge q \wedge \dots \wedge t$
Fact	u	u
Goal clause	$\neg p \vee \neg q \vee \dots \vee \neg t$	$false \leftarrow p \wedge q \wedge \dots \wedge t$

“人必有一死”的Horn子句

$$\forall X (human(X) \rightarrow mortal(X))$$

早期知识表示的两种路线的分析

- 非基于逻辑的知识表示形式
 - ◆ 语义网络和框架，表达形式灵活、易于理解、但没有明确语义
- 基于逻辑的知识表示形式
 - ◆ 一阶谓词逻辑和Horn子句，推理问题复杂度高，半判定或不可判定

描述逻辑的思想

描述逻辑克服了两种知识表示形式的不足并综合其长处

- 藐视逻辑作为知识表示语言，语义可化为一阶谓词逻辑的；克服语义网络和框架没有正式明确的语义缺点；
- 一阶谓词的推理问题是不可判定的，描述逻辑是一阶谓词逻辑的子集，在很大程度上改善了推理问题的可判定性和复杂性
- 描述逻辑继承语义网络和框架的特点，它的知识表示方式比一阶谓词更具有灵活性与可理解性

描述逻辑的特点

- 描述逻辑是一种知识表示语言，他以结构化的和易理解的形式来表示领域知识；
- 描述逻辑能很好的满足本体语言在语义、表达能力以及复杂性的要求：它具有正式的基于逻辑的语义和很强的表达能力，是一阶谓词的可判定子集
- 本体语言普遍将描述逻辑作为语言的逻辑基础

描述逻辑的发展

围绕“表达能力和推理能力的权衡”

- KL-one
- CLASSIC
- Tableau
 - ◆ FaCT作为代表Tableau算法提供了完备的推理手段
- 描述逻辑的扩展

描述逻辑的组成

- 概念
- 关系
- 实例
- 公理

▪ 描述逻辑ALC的语法

描述逻辑ALC是最小的命题封闭的描述逻辑

定义 3.1: ALC 语法. 令 N_C 和 N_R 是可数的不相交的原子概念集和原子关系集。ALC 的概念描述(简称概念)递归定义如下:

- 1) 任意原子概念 $A \in N_C$ 是 ALC 概念;
- 2) 令 C 和 D 是 ALC 概念, R 是 ALC 的原子关系, 即 $R \in N_R$, 则表达式 $\neg C$ (补)、 $C \sqcup D$ (并)、 $C \sqcap D$ (交)、 $\exists R.C$ (存在约束) 和 $\forall R.C$ (全称约束) 是 ALC 概念。

▪ 描述逻辑ALC的语义

概念解释为一元谓词; 关系解释为二元谓词。

定义 3.2: ALC 语义. ALC 解释是一个二元对 $I = (\Delta^I, \cdot^I)$, 其中 Δ^I 是代表论域的非空集合, \cdot^I 是解释函数, 它将所有 $A \in N_C$ 映射为 Δ^I 的子集, 所有 $R \in N_R$ 映射为 $\Delta^I \times \Delta^I$ 的子集, 分别称为原子概念 A 和原子关系 R 的解释。对于定义 3.1 中由构造子构成的概念, 解释函数 \cdot^I 定义它们的解释为:

$(\neg C)^I$	=	Δ^I / C^I
$(C \sqcap D)^I$	=	$C^I \cap D^I$
$(C \sqcup D)^I$	=	$C^I \cup D^I$
$(\forall R.C)^I$	=	$\{a \in \Delta^I \mid \forall b((a,b) \in R^I \rightarrow b \in C^I)\}$
$(\exists R.C)^I$	=	$\{a \in \Delta^I \mid \exists b, (a,b) \in R^I \wedge b \in C^I\}$

▪ 描述逻辑ALC的知识库形式

□ TBox

表示属于公理 (概念间蕴含和等价)

□ ABox

表示断言事实 (陈述个体-概念, 个体间-关系)

TBox [[edit](#)]

- $\mathcal{I} \models C \sqsubseteq D$ if and only if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
- $\mathcal{I} \models \mathcal{T}$ if and only if $\mathcal{I} \models \Phi$ for every $\Phi \in \mathcal{T}$

ABox [[edit](#)]

- $\mathcal{I} \models a : C$ if and only if $a^{\mathcal{I}} \in C^{\mathcal{I}}$
- $\mathcal{I} \models (a, b) : R$ if and only if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$
- $\mathcal{I} \models \mathcal{A}$ if and only if $\mathcal{I} \models \phi$ for every $\phi \in \mathcal{A}$

▪ 描述逻辑ALC的推理

□ 概念推理问题

- ◆ 概念满足 (Satisfiability)

- ◆ 概念蕴含 (Subsumption)
- ◆ 概念等同 (Equivalence)
- ◆ 概念脱节 (Disjointness)

□ 后三个问题都可以转换为概念满足问题

概念满足问题可以转换为其他三个问题

C 蕴含于 D 等价于 $C \sqcap \neg D$ 不可满足;
 D 等同 C 等价于 $C \sqcap \neg D$ 和 $D \sqcap \neg C$ 不可满足;
 D 与 C 脱节等价于 $C \sqcap D$ 不可满足。

C 可满足等价于 C 不蕴含于 \perp , 等价于 C 不等同于 \perp , 等价于 C 和 \top 不脱节。

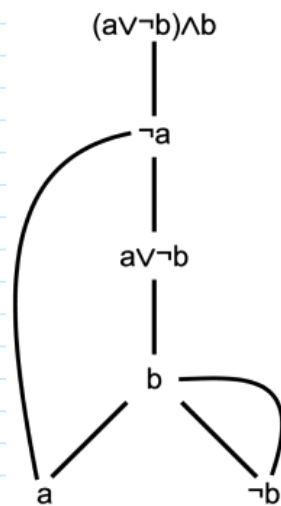
□ 个体推理问题:

- ◆ 一致性 consistency
- ◆ 实例检查 instance checking
- ◆ 检索问题 retrieval problem

□ **Tableau算法原理:**

利用描述逻辑中的转化规则, 不断将原始问题展开为树结构, 直至能判断该命题成立或者不可满足为止

Tableau是一个深度优先的算法 他总是先扩展最左分支, 在完成态如果存在一个无冲突分支, 则表示推理结论一致。



\sqcap -rule	if: $(C_1 \sqcap C_2) \in L_G(a)$, for some a and $\{C_1, C_2\} \not\subseteq L_G(a)$. then: $S' = S \cup \{G'\} \setminus \{G\}$, where $G' = \langle V_G, E_G, L_{G'} \rangle$. $L_{G'} = L_G$ except $L_{G'}(a) = L_G(a) \cup \{C_1, C_2\}$.
\sqcup -rule	if: $(C_1 \sqcup C_2) \in L_G(a)$, for some a and $\{C_1, C_2\} \cap L_G(a) = \emptyset$. then: $S' = S \cup \{G_1, G_2\} \setminus \{G\}$, where $G_k = \langle V_G, E_G, L_{G_k} \rangle$. $L_{G_k} = L_G$ except $L_{G_k}(a) = L_G(a) \cup \{C_k\}$ for $k \in \{1, 2\}$.
\exists -rule	if: $\exists R \cdot C \in L_G(a_1)$, for some a_1 , and there is no $a_2 \in V_G$, such that both $R \in L_G(\langle a_1, a_2 \rangle)$ and $C \in L_G(a_2)$. then: $S' = S \cup \{G'\} \setminus \{G\}$, where $G' = \langle V_G \cup \{a_2\}, E_G \cup \{\langle a_1, a_2 \rangle\}, L_{G'} \rangle$. $L_{G'} = L_G$ except $L_{G'}(a_2) = \{C\}$, $L_{G'}(\langle a_1, a_2 \rangle) = \{R\}$.
\forall -rule	if: $\forall R \cdot C \in L_G(a_1)$, for some a_1 and there is $a_2 \in V_G$, such that $R \in L_G(\langle a_1, a_2 \rangle)$, but not $C \in L_G(a_2)$. then: $S' = S \cup \{G'\} \setminus \{G\}$, where $G' = \langle V_G, E_G, L_{G'} \rangle$. $L_{G'} = L_G$ except $L_{G'}(a_2) = L_{G'}(a_2) \cup \{C\}$.

▪ E.G.

沃日 这个例子真的是笑死我了23333

Unfold and normalise $vegan \sqcap \neg vegetarian$

$person \sqcap \forall eats.plant \sqcap (\neg person \sqcup \exists eats.(\neg plant \sqcap \neg dairy))$

Initialise T to L(x) to contain:

$person \sqcap \forall eats.plant \sqcap (\neg person \sqcup \exists eats.(\neg plant \sqcap \neg dairy))$

Apply \sqcap -rule and add to L(x):

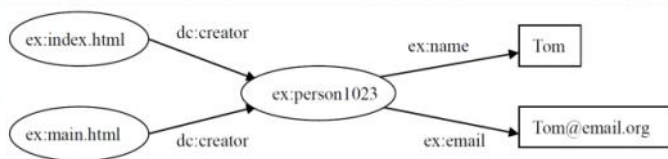
$\{person, \forall eats.plant, \neg person \sqcup \exists eats.(\neg plant \sqcap \neg dairy)\}$

○ **RDF (Resource description framework)**

最低限度的约束，灵活的描述信息，可用于Web

- 图表示：采用了基于三元组声明的图模型
- 唯一标识：基于URI的可扩展集
- 信息交换：基于XML的序列化语法编码
- 逻辑基础：形式化的语义和可证明的推论
- 开放：允许任何人发表任何资源的声明

RDF声明都表示为三元组的形式

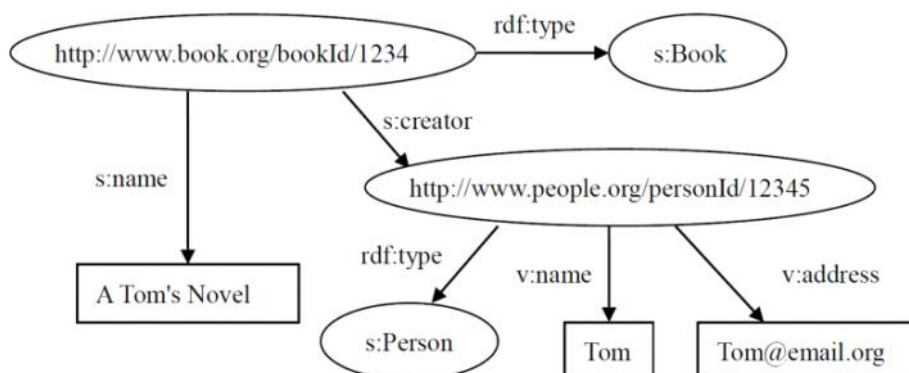


▪ **RDF Schema**

RDFS用于定义和描述词汇集

- 类: rdfs: Class
- 类层次: rdfs: subClassOf
- 实例定义: rdfs: type
- 属性定义: rdfs: range, rdfs: domain
- 属性层次: rdfs: SubPropertyOf

▪ **E.G. 关于一本书的RDF图**



▪ **RDF的不足**

- **值域的定义:**

defs中通过rdfs: range定义了属性的值域, 该值域是全局性的

□ **类 属性 个体的等价性:**

rdfs中无法声明两个或多个类、属性和个体是等价还是不等价还是不等的

□ **不想交类的定义:**

只能声明子类关系, 无法声明这两个类是不想交的

□ **类的布尔结合定义:**

通过类的并 交 补的声明实现对某些类的结合, 从而构建新的类。

□ **基数约束**

对某些属性值可能或必须的取值范围进行约束

□ **关于属性特性的描述:**

声明属性的某些特性, 以及一个属性是另一个属性的逆属性

▪ **OWL的设计思想**

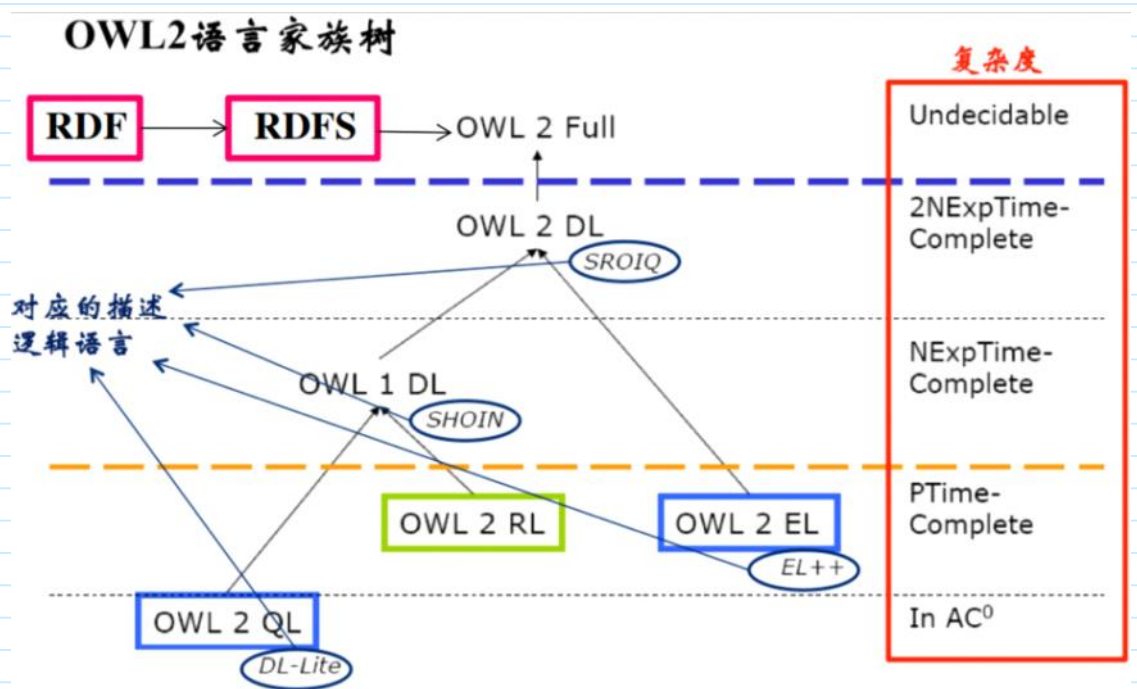
□ 扩展RDFS 语法规则采用RDFS/XML 语义严格遵循描述逻辑

□ OWL提供OWL DL 和OWL Lite两种子语言

□ 提供OWL Full 可完全兼容RDFS

▪ **OWL2**

OWL2定义了OWL的子语言, 通过限制语法使用, 使得子语言能够更方便的实现和应用



▪ **JSON-LD (JSON for Linked Data)**

□ JSON-Ld提供了一种规范的术语（本体）表示, 使得对术语的理解跨越了猜测的过程, 使得机器能处理和理解

一个JSON描述片段

```
{  
  "name": "Manu Sporny",  
}
```

一个JSON描述片段

```
{
  "name": "Manu Sporny",
  "homepage": "http://manu.sporny.org/",
  "image": "http://manu.sporny.org/images/manu.png"
}
```

一个JSON-LD描述片段

```
{
  "http://schema.org/name": "Manu Sporny",
  "http://schema.org/url": { "@id": "http://manu.sporny.org/" }, "http://schema.org/image":
  { "@id": "http://manu.sporny.org/images/manu.png" }
}
```

知识图谱统计表示学习

传统表示:

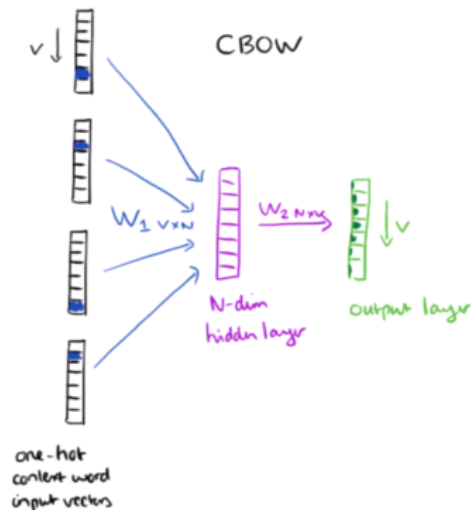
- 基于符号的三元组表示, 不能捕获实体间的语义关系 (显式 + 隐式)
- 不好直接利用各种机器学习模型进行分析和挖掘

统计学习表示:

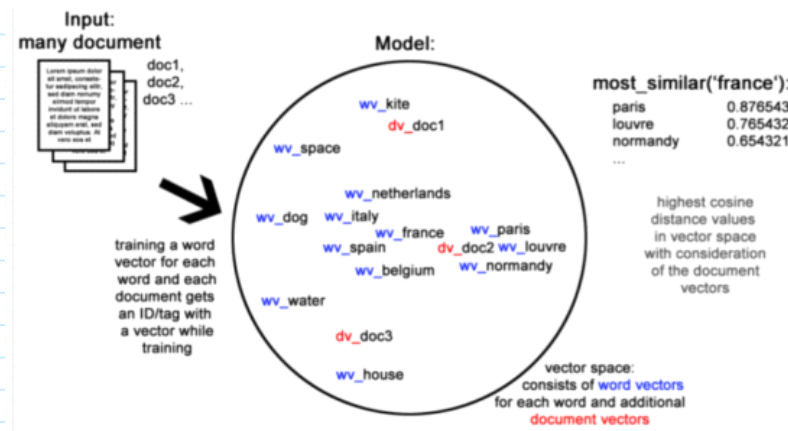
- 给出一种统计上的分布式表示形式, 能捕获实体间的语义关系, 特别是隐式关系
- 表示形式为向量, 能被各种机器学习模型直接使用

自然语言中的表示学习

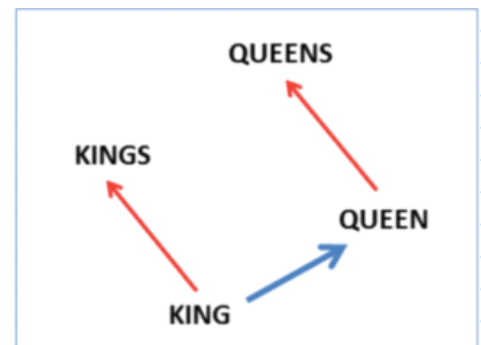
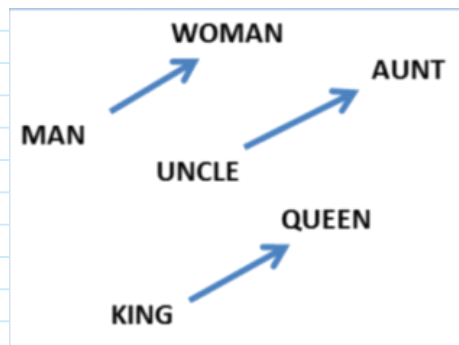
Word2Vec



Doc2Vec



- 表示结果有统一的语义空间 语义可计算



▪ 知识图谱表示学习原理

- 将三元组<h,r,t>看成h通过r翻译到t的过程
- 优化目标
 - ◆ 势能函数

$$f(h, r, t) = ||h + r - t||_2$$
 - ◆ 目标函数