

知识图谱基础知识

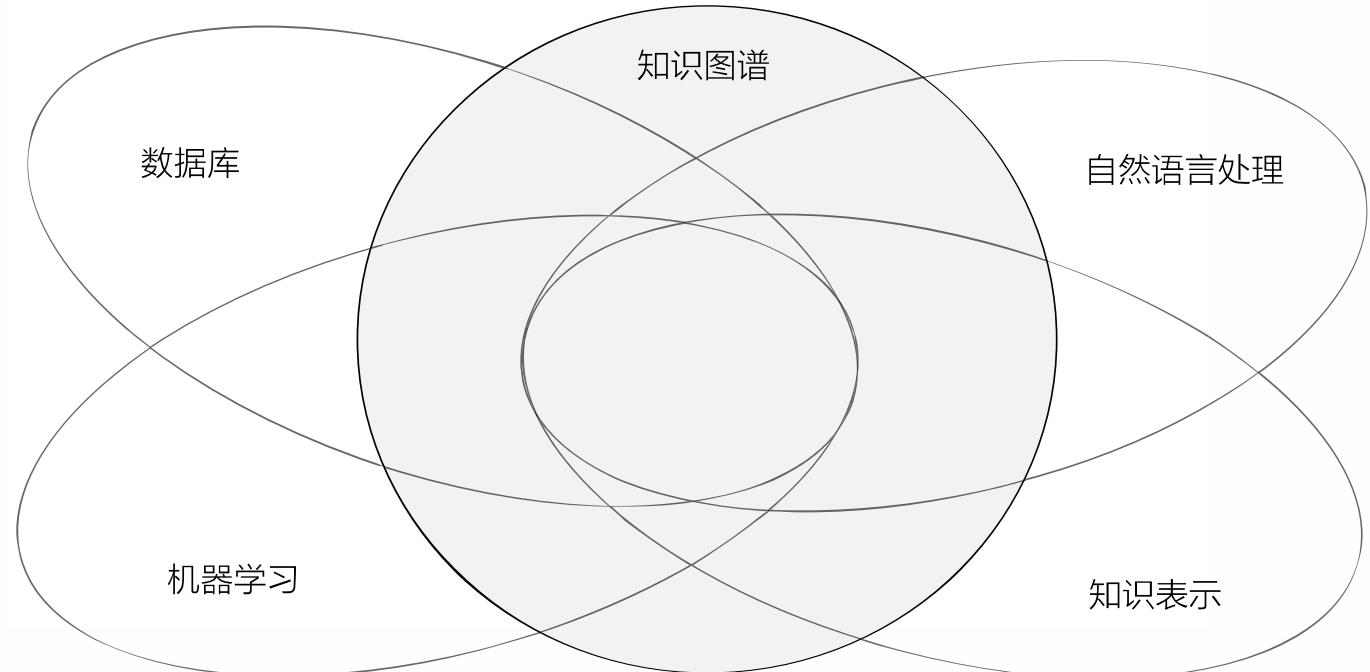
本章大纲

- 概述
- 知识表示
 - 基于图论的知识图谱表示
 - 基于数值的知识图谱表示
 - 其他相关知识表示
- 机器学习
 - 机器学习的基本概念
 - 深度学习概述
 - 卷积神经网络
 - 循环神经网络
 - 注意力机制
- 自然语言处理
 - 基本概念
 - 文本的向量化表示

概述

知识图谱相关计算机子学科

- 知识表示
- 机器学习
- 自然语言处理
- 数据库

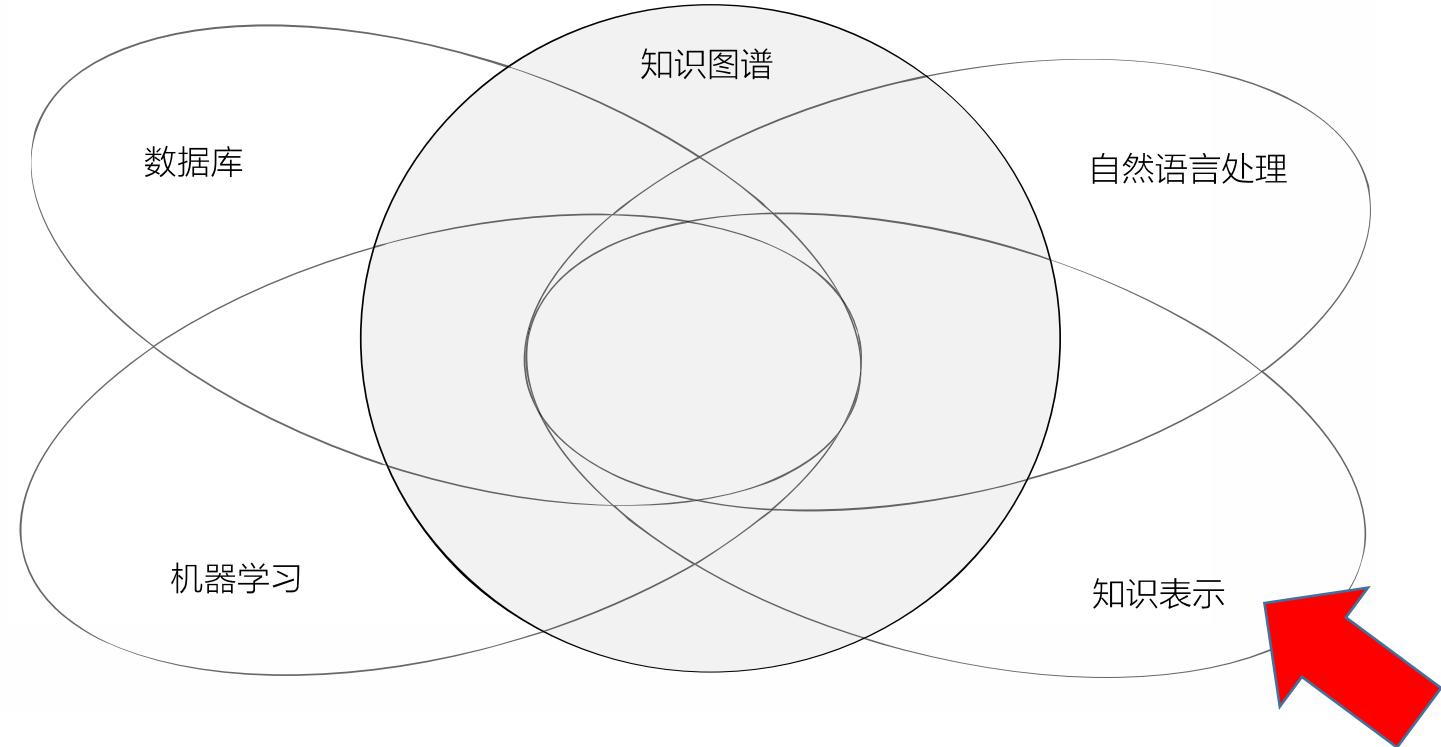


与知识图谱密切相关的计算机子学科

计算机相关学科-知识表示

• 知识表示

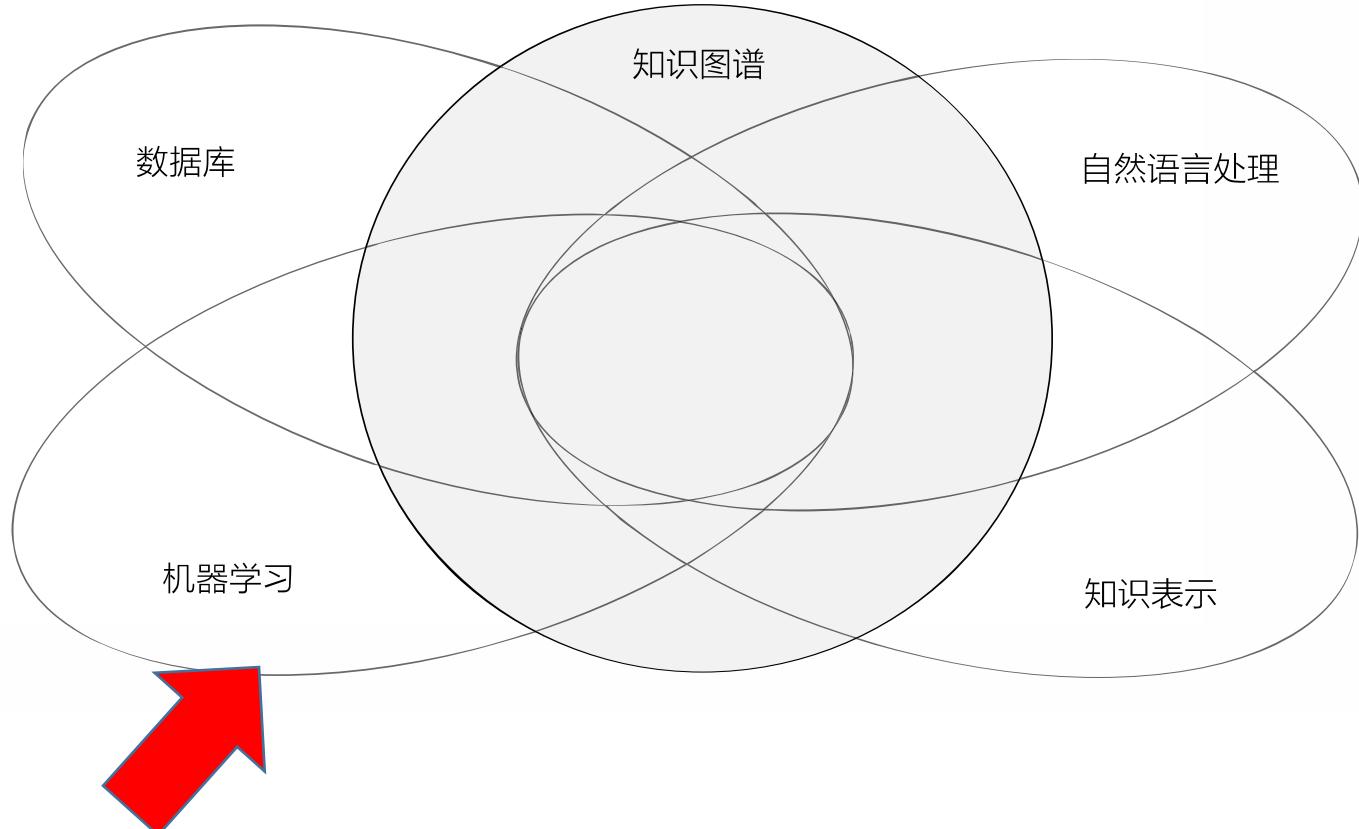
- 知识图谱的狭义概念是一种作为语义网络的知识表示。
- 知识表示除了知识图谱之外还有很多其他表示。澄清知识图谱与这些知识表示之间的联系与区别，是进一步深化对于知识图谱的理解的前提。



计算机相关学科-机器学习

• 机器学习

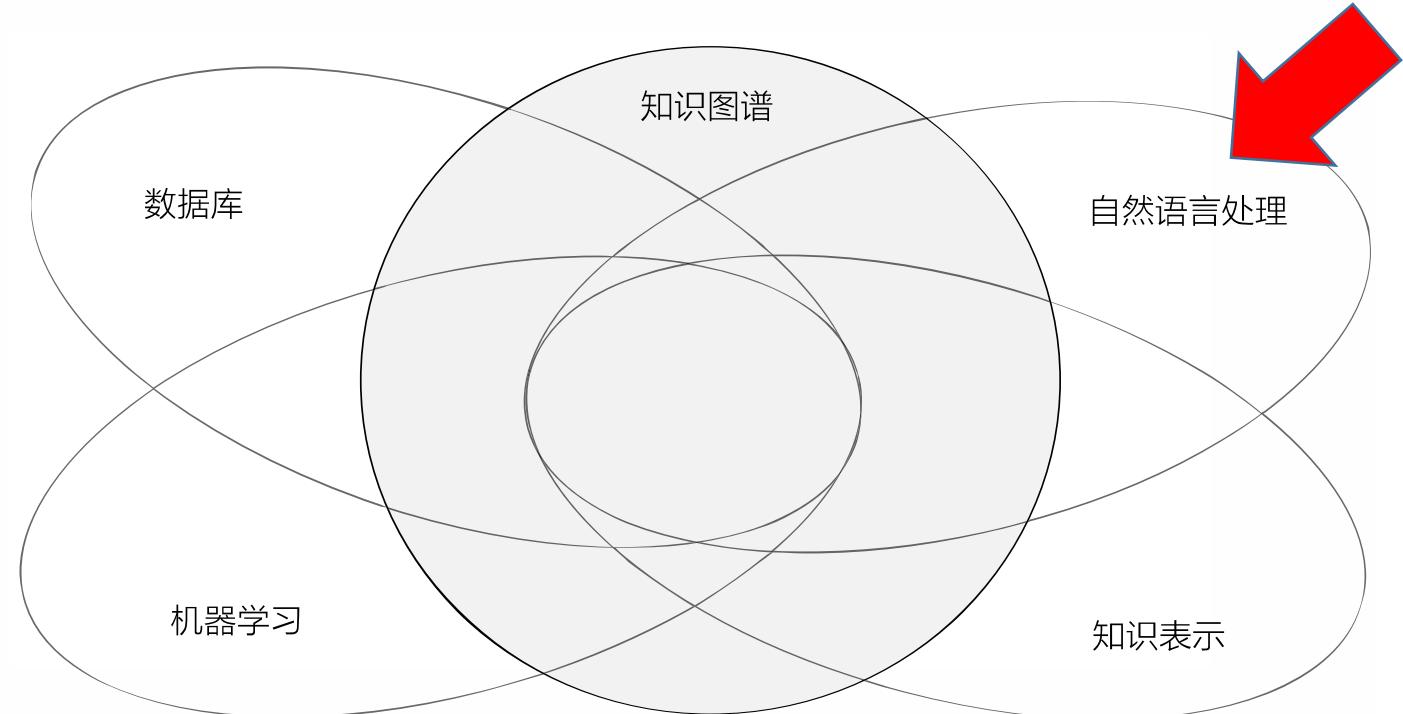
- 知识图谱应用的整个生命周期的多个环节都与机器学习有关。知识获取、知识应用、知识挖掘等众多环节构建机器学习模型。
- 从样本数据中习得有效的统计模式来解决问题是当前机器学习解决问题的主要思路。
- 知识图谱也可以广泛应用到信息检索、数据挖掘、可视化分析等计算机学科的各个领域。



计算机相关学科-自然语言处理

- 自然语言处理

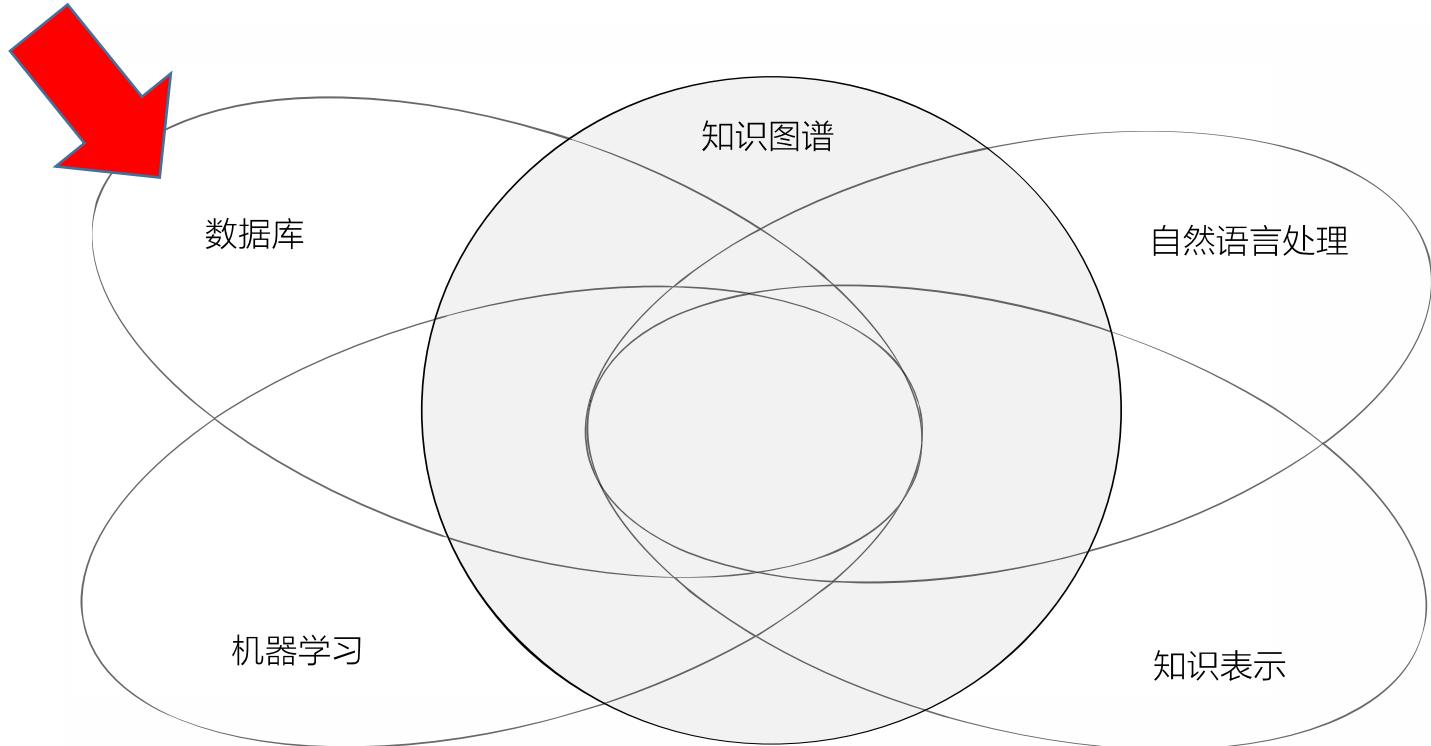
- 知识获取的一个重要途径是从自由文本进行抽取。而文本信息抽取是自然语言处理的核心问题之一。
- 知识图谱构建好之后通常可以用作支撑自然语言理解的背景知识。



计算机相关学科-数据库

- 数据库

- 数据库技术实现知识图谱数据的系统化存储、组织、查询与检索，实现高效的知识图谱数据访问。
- 我们将放到第9-11章具体描述。



其他相关学科

- 语言学与认知学
 - 语言学是以人类语言为研究对象的学科。人类语言是知识的重要载体之一。知识图谱中表达的知识都是人类能够言说的知识。
 - 知识图谱中知识进一步可以支撑机器理解人类自然语言。人类是如何利用背景知识实现语言理解的？对这些问题的回答有助于实现基于知识图谱的机器语言认知。
 - 知识图谱中的知识是人类认知的结果。人类是如何认知这个世界进而形成知识的，对于我们理解知识图谱的内涵，指导知识图谱的建设有重要作用。

知识表示

知识表示的关键因素

- 知识必须经过合理的表示才能为计算机所处理。知识表示是对现实世界的一种抽象（abstract）表达。
- 评价知识表示的两个重要因素
 - 表达能力（expressiveness）：一个知识表示应该具有足够强的表达能力，以充分完整地表达特定领域或者问题所需的知识。
 - 计算效率（efficiency）：基于这一知识表示的计算求解过程也应有着足够高效的执行效率。

expressiveness



efficiency

符号表示 vs 数值表示

- 符号表示
 - 文字
 - “柏拉图”、“苏格拉底”、“老师”
 - 点、边等符号
 - 用关联图表示关系
 - 逻辑运算符号
 - $P \Rightarrow Q$ 表示两个命题之间的逻辑蕴涵关系
- 数值表示
 - 标量
 - 人的身高可以表达为“1.5m”
 - 向量
 - 文档的语义可以表示为词向量
 - 概率分布
 - 消费者购买商品的倾向可以表示为定义在商品集上的概率分布

知识表示分类

- 在实际应用中，根据学科背景的不同，人们发展了基于图论、逻辑学、概率论的各种知识表示。

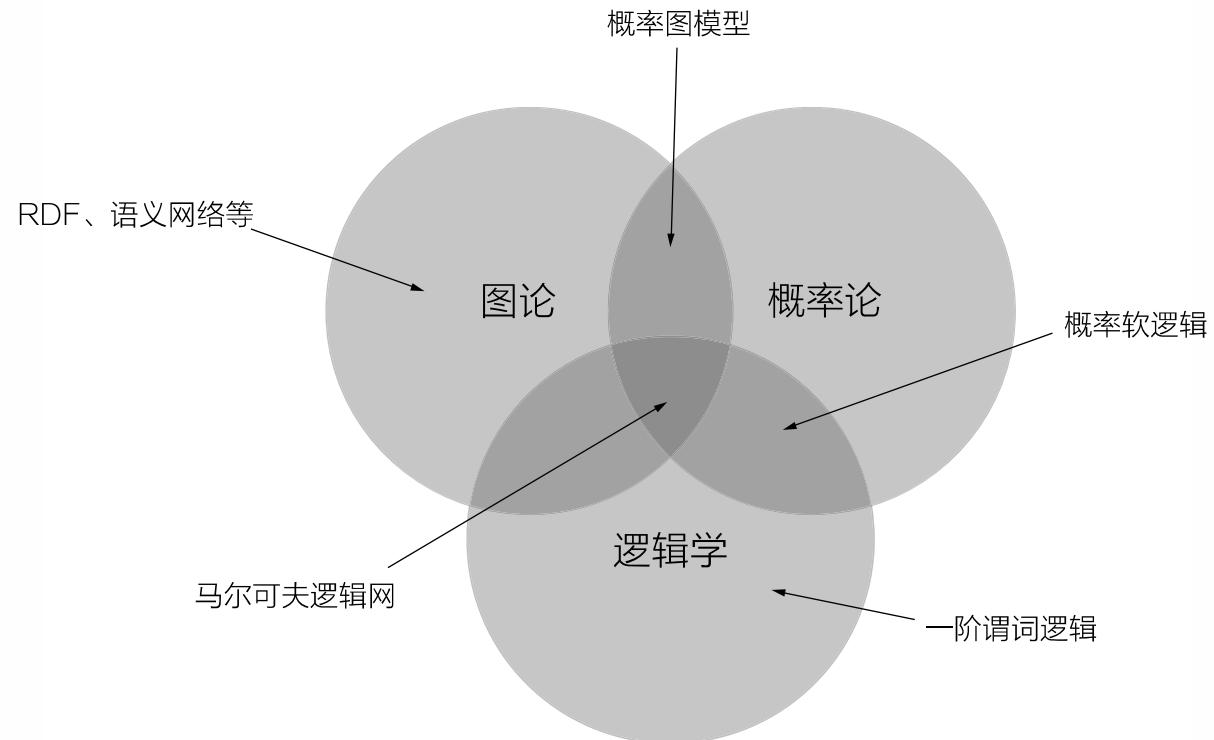


图 2-2 基于不同学科发展出来的知识表示

知识表示的趋势

- 知识表示的关键
 - 符号知识与数值表示的融合
 - 专家知识与统计模型的融合
- 知识表示从早期的单纯以符号表示为主，日益愈统计模型与数值表示融合
 - 侧重于表达专家的符号知识，以符号知识为主
 - 比如规则、逻辑，对概率图模型等鲜有提及
- 概率图模型
 - 随机变量的选择体现的是专家知识
 - 概率依赖关系的估计体现的是统计学习模型。
- 决策树
 - 专家给出决策的因素(特征)
 - 而机器自动从数据中学习决策时的权重

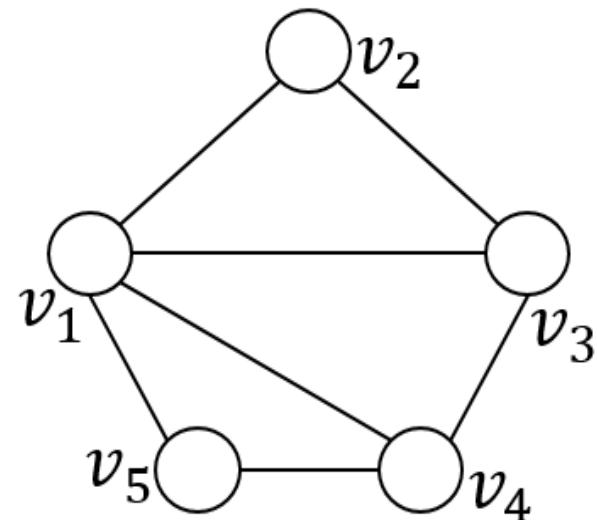
专家知识 + 统计模型

(专家给出决策框架) (机器决定关联强度或者给出概率估计)

基于图论的知识图谱表示

- $G = G(V, E)$,
 - 其中 V 表示顶点集, $E \subseteq V \times V$ 表示边的集合。

- 有向图、无向图
- 邻接表、邻接矩阵
- 度数、路径、可达…



0	1	1	1	1
1	0	1	0	0
1	1	0	1	0
1	0	1	0	1
1	0	0	1	0

基于三元组的知识图谱表示

- RDF (Resource Description Framework) 是用于描述现实中资源的 W3C 标准。
- 现实世界中每个概念、实体和事件都可以对应一个**资源**, 可以表示具体的事物也可以是抽象的概念, 以及属性;
- 每个资源都用**IRI** (Internationalized Resource Identifier, 国际化资源标识符) 进行标识;
- RDF 允许引入不包含任何IRI 标示的资源, 被称为**空白结点**或者**匿名资源**, 用于标示一种存在变量。空白结点不能用IRI 来全局处理, 所以为了区分不同的空白结点, RDF 解析器一般会为每个空白结点分配一个系统生成的内部名。

RDF图数据模型

资源描述框架(RDF)数据示例

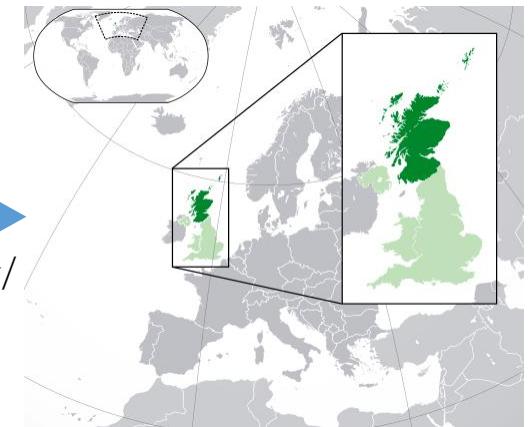
- RDF中任何实体都被称之为资源(Resource), 用一个统一的国际化标识符 (IRI, Internationalized Resource Identifier) 来唯一标识；
- 资源的属性可以被定义；
- 资源间关系可以被定义；

[http://dbpedia.org/resource/
University_of_Glasgow](http://dbpedia.org/resource/University_of_Glasgow)



[http://dbpedia.org/ontology/
located_in](http://dbpedia.org/ontology/located_in)

[http://dbpedia.org/resource/
Scotland](http://dbpedia.org/resource/Scotland)



RDF图数据模型

国际化资源标识符IRI

IRI 是一个用来标识资源的字符串，是数据集中资源的一个唯一的身份ID；当原始的IRI长度过长时，为了方便表达可以引入前缀（prefix）命名空间等方式来简化，以下是常见前缀

前缀	IRI
rdfs:	http://www.w3.org/2000/01/rdf-schema#
rdf:	http://www.w3.org/1999/02/22-rdf-syntax-ns#
xsd:	http://www.w3.org/2001/XMLSchema#
sfn:	http://www.w3.org/ns/sparql#
dbr:	http://dbpedia.org/resource/
dbo:	http://dbpedia.org/ontology/
dbp:	http://dbpedia.org/property/

RDF图数据模型

RDF三元组

每个资源的一个属性及属性值，或者它与其他资源的一条关系，都被表示成<主体，谓词，客体>的**三元组**形式，一个三元组又称为**陈述**

- 所谓**主体**，它是一个资源或者是一个空白节点；
- 所谓**属性/谓词**，是用来描述资源之间的语义关系，或者描述某个资源和属性值之间的关系；
- 所谓**客体**，它可以是一个资源，也可以是一个字面值，也可以是一个空白节点

RDF图数据模型

RDF数据集

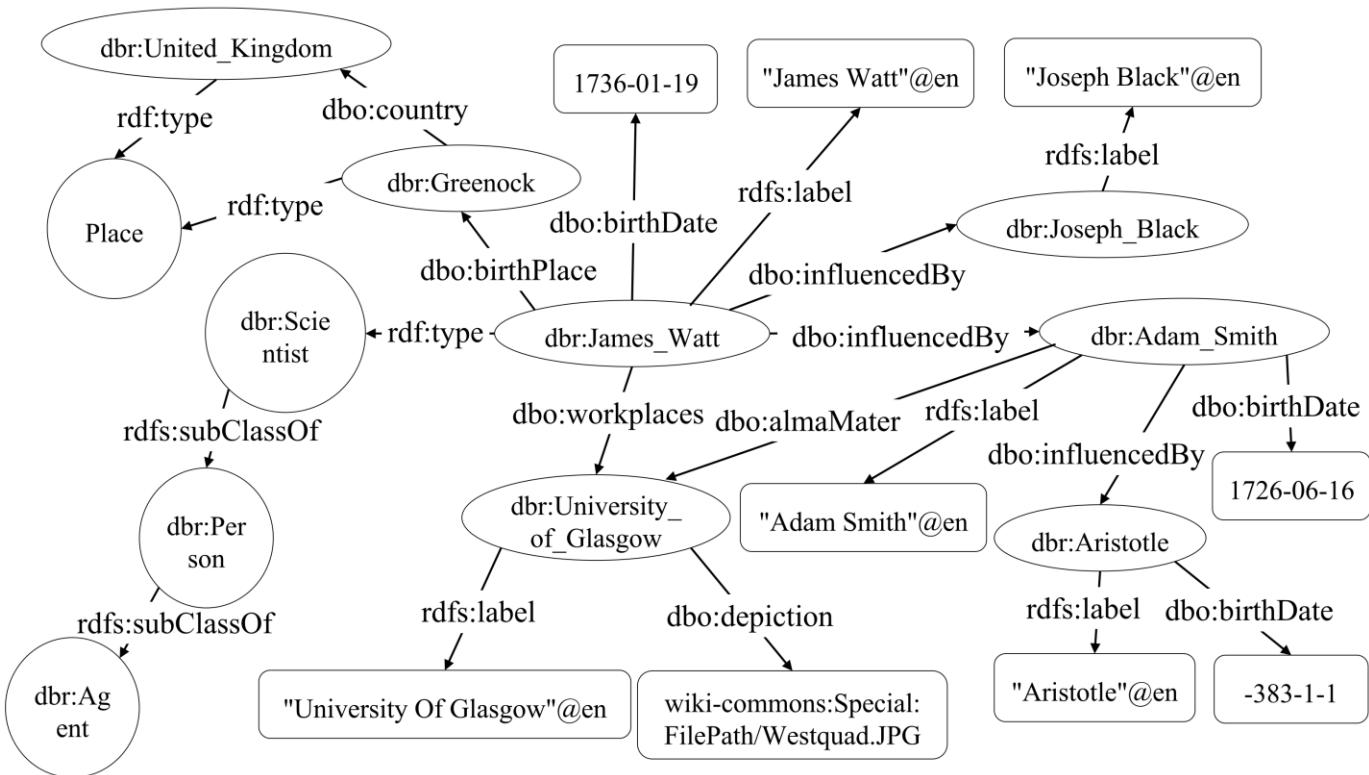
- 给定资源标识符集合 \mathcal{I} 、空白节点集合 \mathcal{B} 和字面值 \mathcal{L} ，一条三元组 t 是属于 $(\mathcal{I} \cup \mathcal{B}) \times \mathcal{I} \times (\mathcal{I} \cup \mathcal{B} \cup \mathcal{L})$ 的一个元素
- 一个RDF 数据集 \mathcal{T} 是 $(\mathcal{I} \cup \mathcal{B}) \times \mathcal{I} \times (\mathcal{I} \cup \mathcal{B} \cup \mathcal{L})$ 的一个子集

主体	属性	客体
dbr:James_Watt	rdfs:label	"James Watt"@en
dbr:James_Watt	dbo:birthDate	"1736-01-19"^^xsd:date
dbr:James_Watt	dbo:birthPlace	dbr:Greenock
dbr:James_Watt	rdf:type	dbo:Scientist
dbr:James_Watt	dbo:influencedBy	dbr:Joseph_Black
dbr:James_Watt	dbo:influencedBy	dbr:Adam_Smith
dbr:James_Watt	dbp:workplaces	dbr:University_of_Glasgow
dbr:Adam_Smith	rdfs:label	"Adam Smith"@en
dbr:Adam_Smith	dbo:birthDate	"1723-06-16"^^xsd:date
dbr:Adam_Smith	dbo:almaMater	dbr:University_of_Glasgow
dbr:Adam_Smith	dbo:influencedBy	dbr:Aristotle
dbr:Joseph_Black	rdfs:label	"Joseph Black"@en
dbr:University_of_Glasgow	name	"University Of Glasgow"@en
dbr:Aristotle	rdfs:label	"Aristotle"@en
dbr:Aristotle	dbo:birthDate	"-383-1-1"^^xsd:date
dbr:Greenock	dbo:country	dbr:United_Kingdom
dbr:Greenock	dbo:type	dbo:Place
dbr:United_Kingdom	dbo:type	dbo:Place
dbo:Scientist	dbo:subClassOf	dbo:Person
dbo:Person	dbo:subClassOf	dbo:Agent

RDF图数据模型

RDF图

- 三元组的主体和客体就是RDF图中的一系列节点
 - 一个谓词的资源标识符在同一张图里可能充当节点，也可能充当边，



RDF图数据模型

RDF字面值

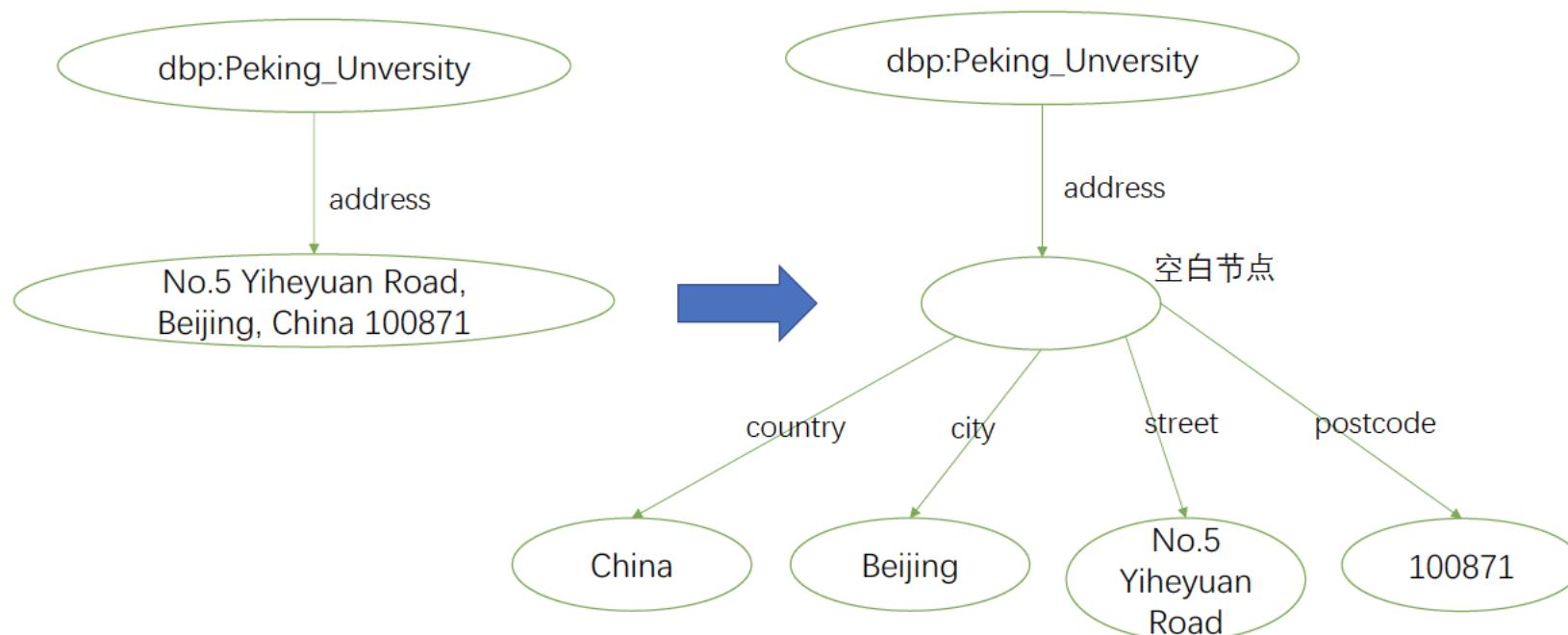
在RDF 的定义中， 字面值只会出现在RDF 三元组的客体中。有两种表达方式：

- 朴素文本(plain literals)，也就是普通意义上的字符串
- 类别化文本(typed literals)，可以指定某个字面值的数据类型，参考XML 语法中数据类型定义的[19]，包括xsd:integer, xsd:decimal 等

RDF图数据模型

RDF空白节点

空白节点(Blank Node) 是指没有统一资源标识符 (IRI) 同时在知识图谱数据集外部不需要直接访问的节点
空白节点的引入可以更加方便地表达多元关系和结构化的数据值



RDF图数据模型

RDF Schema

RDF Schema (简称RDFS)，用来表达实体与类别，以及类别之间、以及属性与属性之间、属性的定义域、值域之间的关系

RDF 预定义了一些核心概念和核心属性，这些概念并不提供某个具体领域专用的类别和属性，但是RDFS 为定义某个领域的本体概念提供了基础

RDF图数据模型

RDF Schema

- **核心类**
 - rdfs:Class: 所有类的类
 - rdfs:Resource: 所有资源的类
 - rdfs:Literal: 所有字面值的类
 - rdfs:Property: 所有属性的类
- **核心属性**
 - rdf:type: 连接一个资源和它属于的类别
 - rdfs:subClassOf: 连接一个类别和它的父类
 - rdfs:subPropertyOf: 连接一个属性和它的子属性
 - rdfs:domain: 定义一个属性的作用域（它的主语的类别）

RDF图数据模型

RDF Schema核心类和核心属性举例

三元组	含义
dbp:James_Watt rdf:type dbo:British_Scientist	实体“瓦特”是类别“英国科学家”之一
dbo:British_Scientist rdfs:subclass dbo:Scientist	类别“英国科学家”属于类别“科学家”
dbo:birthPlace rdf:type rdfs:Property	birthPlace是一个属性
dbo:birthPlace rdfs:domain dbo:Person	birthPlace的主体一定是类别“人”的实体
dbo:birthPlace rdfs:range dbo:location	birthPlace的客体一定是类别“地址”的实体

属性图模型

属性图模型简介

- 属性图模型是一种不同于RDF三元组的一种图数据模型
- 这个模型由点来表示现实世界中的实体，由边来表示实体与实体之间的关系。同时，点和边上都可以通过键值对的形式被关联上任意数量的属性和属性值
- 在这种图模型中，关系被提到了一个和实体本身一样重要的程度
- 从形式化的角度来看，属性图模型包含三种元素组成：**值、图和表**。

属性图模型

属性图模型符号列表

值类型	表示单个值的符号	表示值集合的符号
属性键	k	\mathcal{K}
点标识符	n	\mathcal{N}
边标识符	r	\mathcal{R}
点标签	l	\mathcal{L}
关系标签	t	\mathcal{T}
函数	f	\mathcal{F}
值	v	\mathcal{V}

属性图模型

属性图模型组成元素——值

- 所谓**值**，就是属性图数据模型中涉及的各种数据类型，包括以下这些值类型
 - **标识符**，包括 \mathcal{N} 中的点标识符、 \mathcal{R} 中的边标识符；
 - **基本数据类型**，包括整数、字符串、布尔值与空值；
 - **链表**，即给定 m 个值 v_1, v_2, \dots, v_m ，链表 $[v_1, v_2, \dots, v_m]$ 也是值；
 - **映射表**，即给定 m 个不同的属性键 k_1, k_2, \dots, k_m 和 m 个值 v_1, v_2, \dots, v_m ，映射表 $\{k_1: v_1, k_2: v_2, \dots, k_m: v_m\}$ 也是值；
 - **路径**，即给定 m 个点 n_1, n_2, \dots, n_m 和 $m - 1$ 条边 r_1, r_2, \dots, r_{m-1} ，路径 $\text{path}(n_1, r_1, n_2, \dots, n_{m-1}, r_{m-1}, n_m)$ 也是值，路径经常会被简写成 $n_1r_1n_2\dots n_{m-1}r_{m-1}n_m$

属性图模型

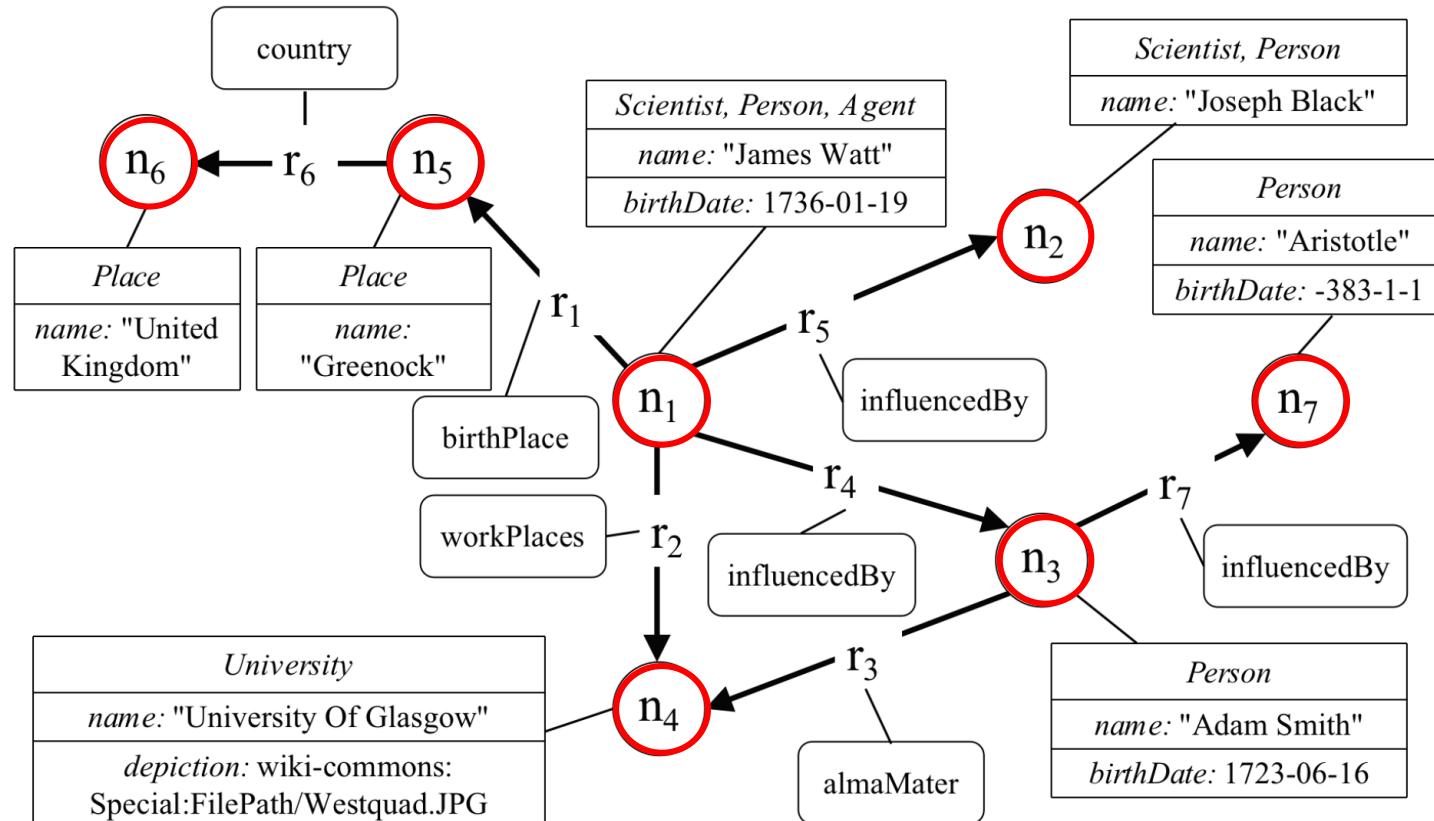
属性图模型组成元素——图

- 属性图模型中的图定义七元组 $G = \langle N, R, \text{src}, \text{tgt}, \iota, \lambda, \tau \rangle$ ，
 - N 是点标识符集合 \mathcal{N} 的子集，表示是图 G 中的点集；
 - R 是边标识符集合 \mathcal{R} 的子集，表示是图 G 中的边集；
 - $\text{src} : R \rightarrow N$ 是一个函数，表示将一条边映射到其的起点；
 - $\text{tgt} : R \rightarrow N$ 是一个函数，表示将一条边映射到其的终点；
 - $\iota : (N \cup R) \times \mathcal{K} \rightarrow V$ 是一个函数，表示将一个点或者一条边上一个属性键映射到一个值；
 - $\lambda : N \rightarrow 2^L$ 是一个函数，表示将一个点映射到一个有限的点标签集合；
 - $\tau : R \rightarrow T$ 是一个函数，表示将一条边映射到一个边类型；

属性图模型

属性图示例

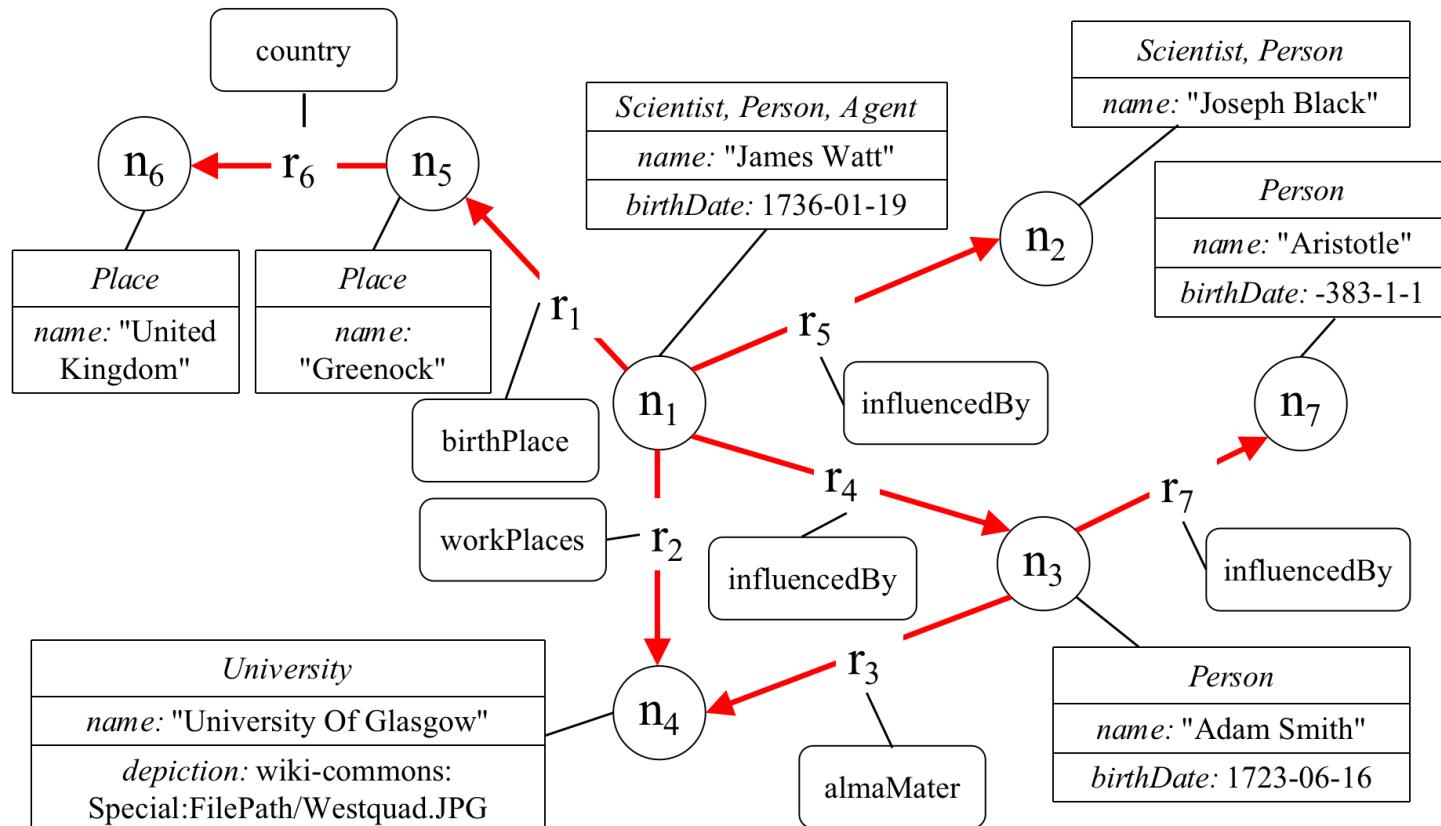
$N = \{n_1, n_2, n_3, n_4, n_5, n_6, n_7\}$ 为点集



属性图模型

属性图示例

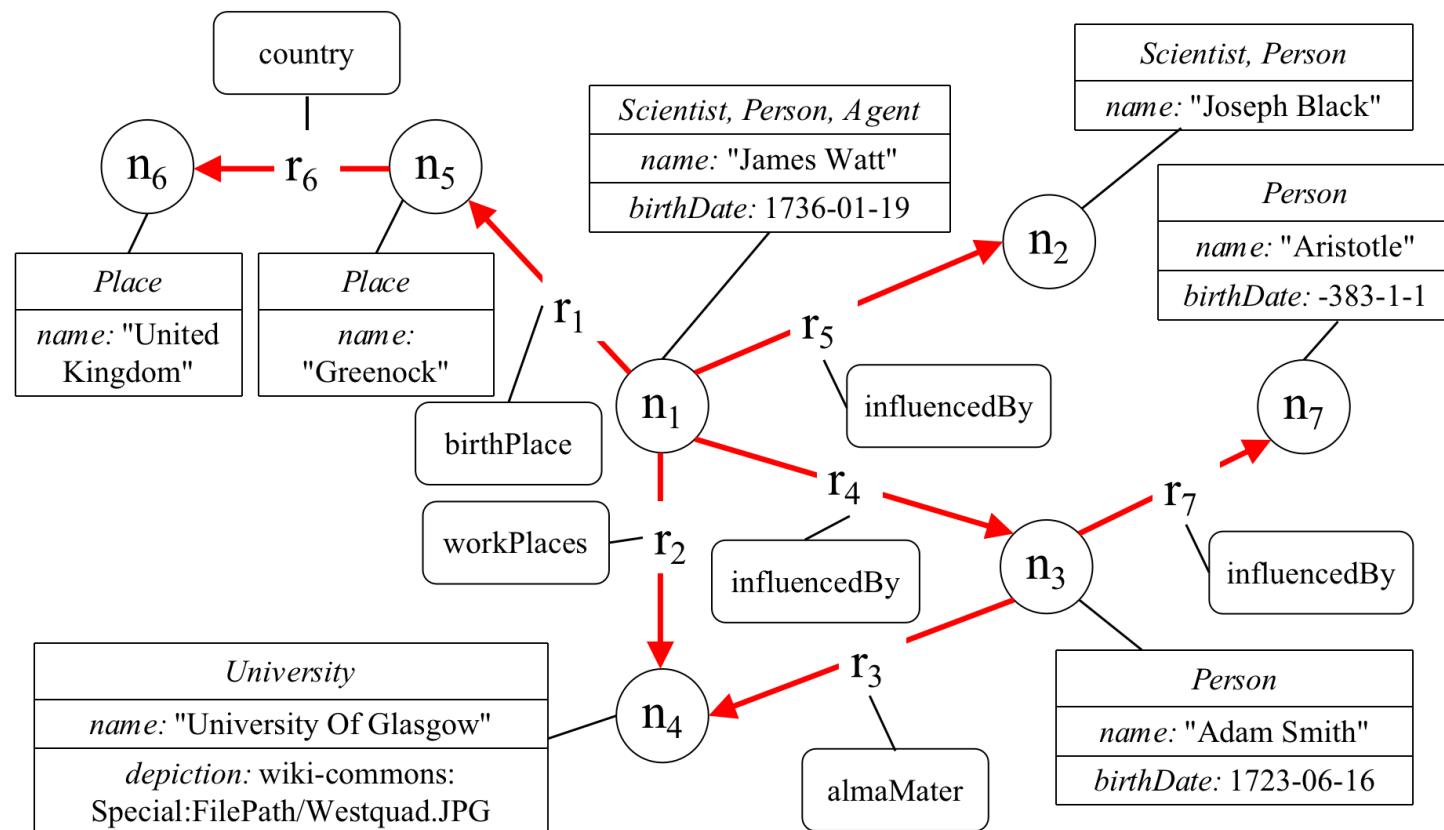
$R = \{r_1, r_2, r_3, r_4, r_5, r_6, r_7\}$ 为边集



属性图模型

属性图示例

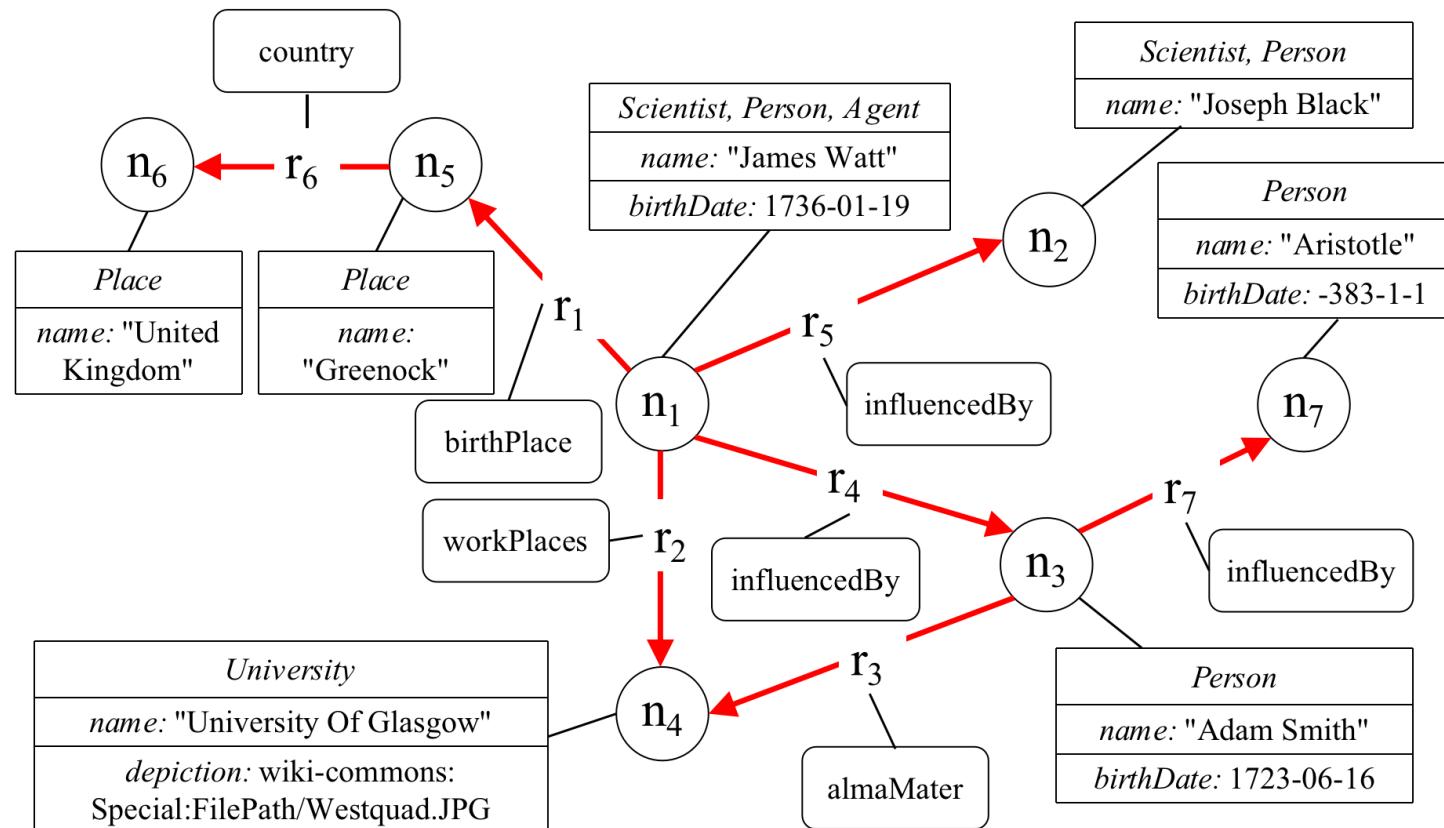
$\text{src} = \{r_1 \rightarrow n_1, r_2 \rightarrow n_1, r_3 \rightarrow n_3, r_4 \rightarrow n_1, r_5 \rightarrow n_1, r_6 \rightarrow n_5, r_7 \rightarrow n_3\}$ 用以将各条边映射到其的起点



属性图模型

属性图示例

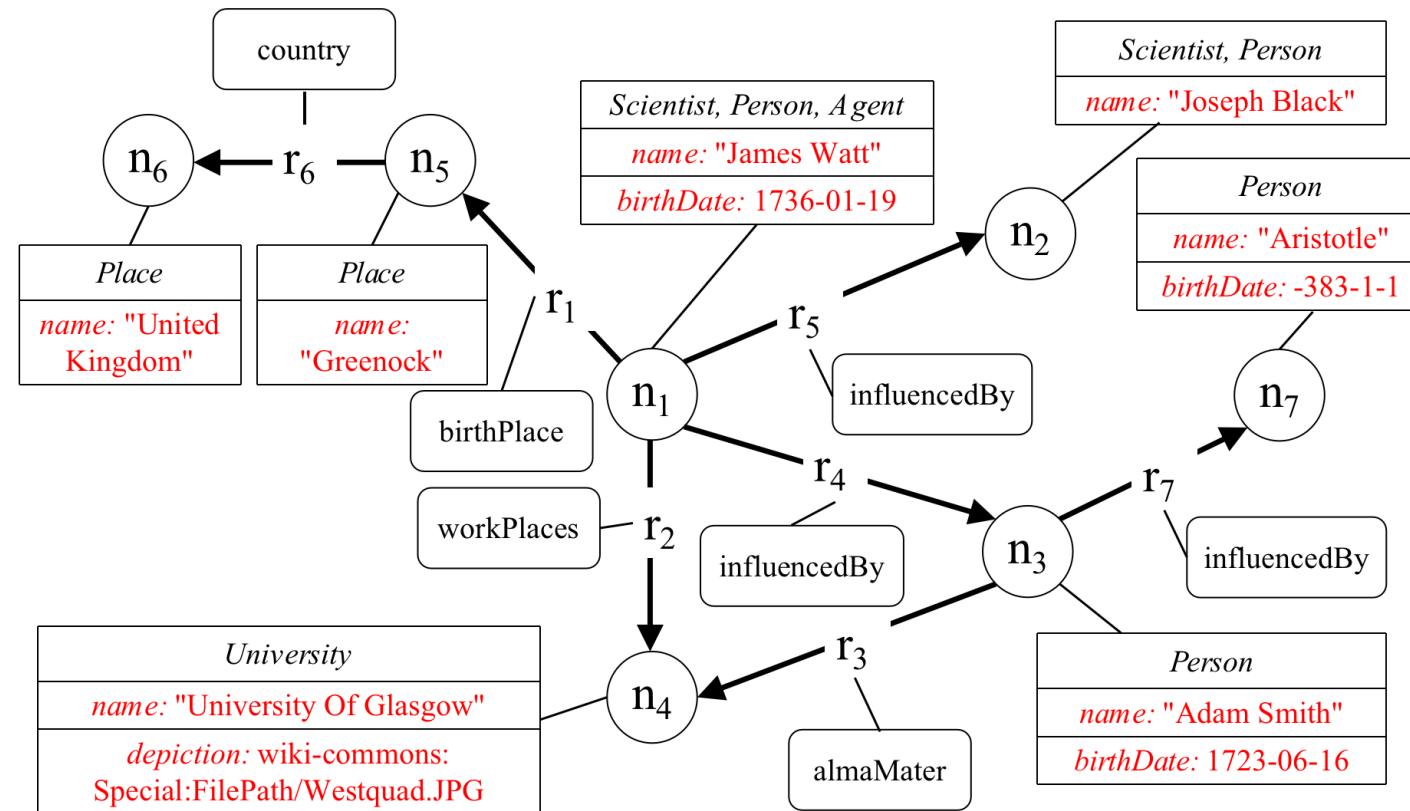
$\text{tgt} = \{r_1 \rightarrow n_5, r_2 \rightarrow n_4, r_3 \rightarrow n_4, r_4 \rightarrow n_3, r_5 \rightarrow n_2, r_6 \rightarrow n_6, r_7 \rightarrow n_7\}$ 用以将各条边映射到其的起点



属性图模型

属性图示例

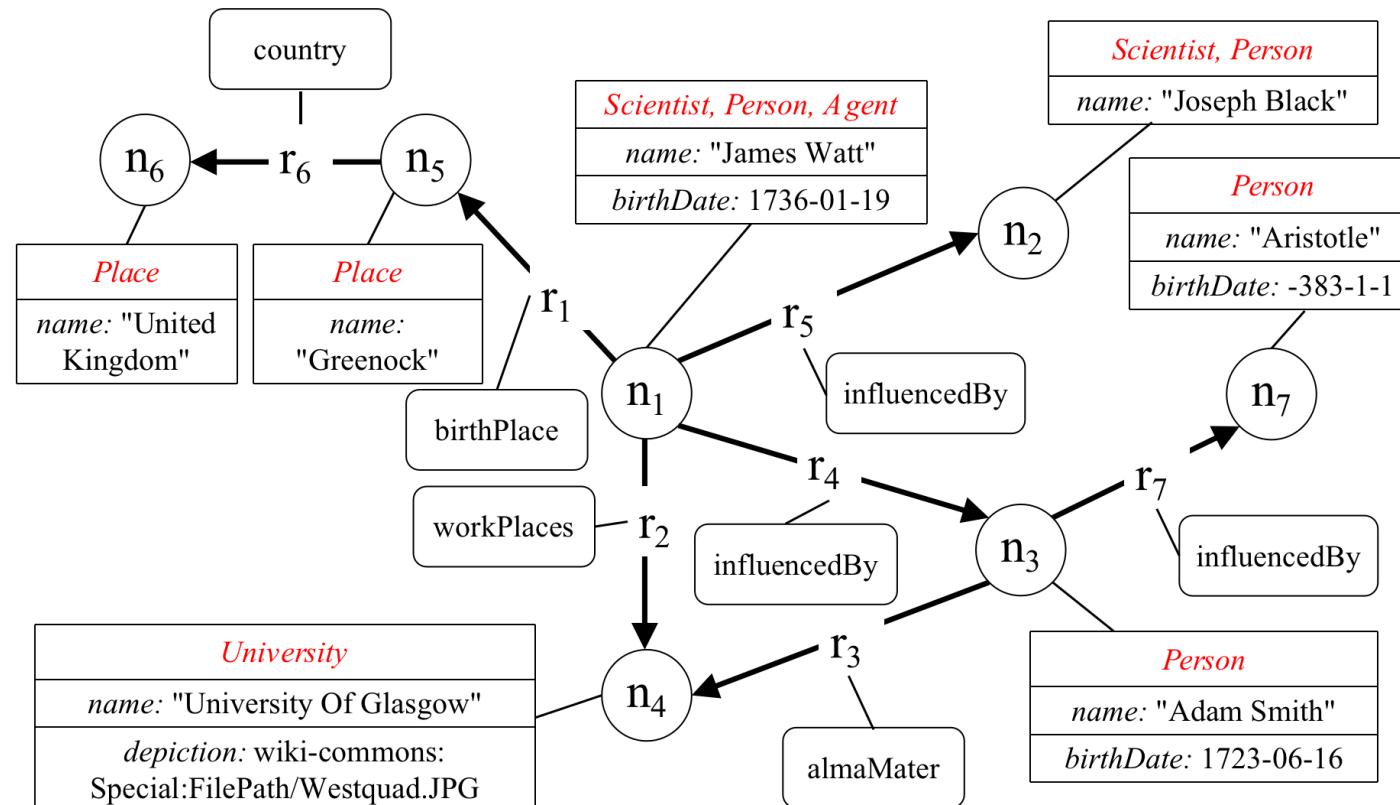
$\iota(n_1, \text{name}) = \text{"James Watt"}$, $\iota(n_1, \text{birthDate}) = 1736-01-19$, $\iota(n_2, \text{name}) = \text{"Joseph Black"}$,
....., $\iota(n_7, \text{birthDate}) = -383-1-1$



属性图模型

属性图示例

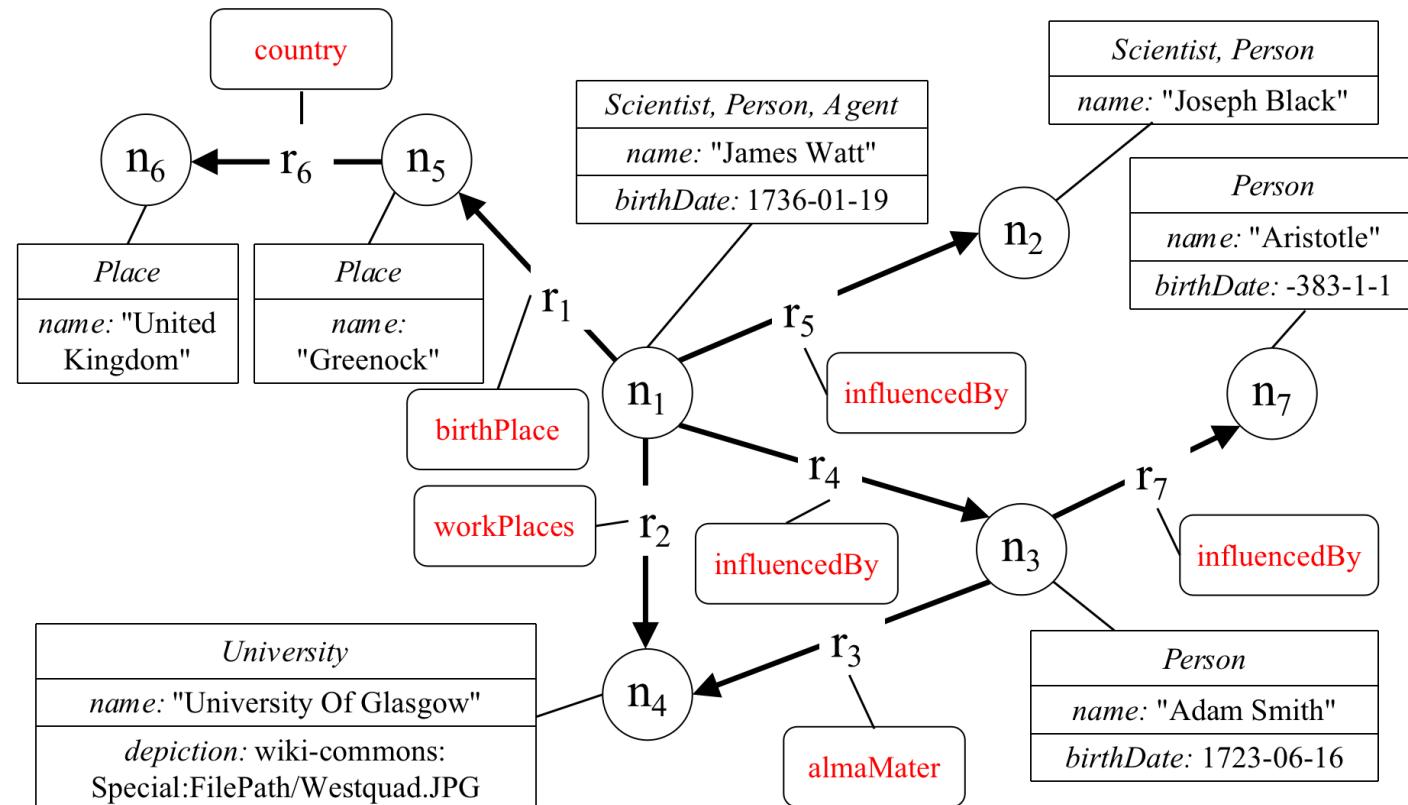
$\lambda(n_1) = \{\text{Scientist, Person, Agent}\}$, $\lambda(n_2) = \{\text{Scientist, Person}\}$, $\lambda(n_3) = \{\text{Person}\}$, $\lambda(n_4) = \{\text{University}\}$, $\lambda(n_5) = \{\text{Place}\}$, $\lambda(n_6) = \{\text{Place}\}$, $\lambda(n_7) = \{\text{Person}\}$



属性图模型

属性图示例

$\tau(r_1) = \text{birthPlace}$, $\tau(r_2) = \text{workPlaces}$, $\tau(r_3) = \text{almaMaster}$, $\tau(r_4) = \tau(r_5) = \tau(r_7) = \text{influencedBy}$, $\tau(r_6) = \text{country}$



属性图模型

属性图模型组成元素——表

- 在属性图模型中，用来定义Cypher 查询语言结果的元素就是**表 (table)**
- 一张表是由多个**记录**所组成的。给定若干表中的列名 $\mathcal{A} = \{a_1, a_2, \dots, a_n\}$ ，一个的记录是一个从 \mathcal{A} 到值集合 \mathcal{V} 的部分函数
- 一个**记录**常被被记为一个包含列名域的元组 $u = (a_1 : v_1, a_2 : v_2, \dots, a_n : v_n)$ ，其中 a_1, a_2, \dots, a_n 为列名，而 v_1, v_2, \dots, v_n 为值
- 我们将一个记录所涉及的列名集合记为 $\text{dom}(u)$ ，并称之为 u 的**定义域**，它是 \mathcal{A} 的子集

属性图模型

属性图模型组成元素——表

- 给定两个定义域完全不相交的记录元组 $u_1 = (a_1 : v_1, a_2 : v_2, \dots, a_n : v_n)$ 和 $u_2 = (a'_1 : v'_1, a'_2 : v'_2, \dots, a'_{n'} : v'_{n'})$, u_1 和 u_2 的连接表示为 :

$$(u_1, u_2) = (a_1 : v_1, a_2 : v_2, \dots, a_n : v_n, a'_1 : v'_1, a'_2 : v'_2, \dots, a'_{n'} : v'_{n'})$$

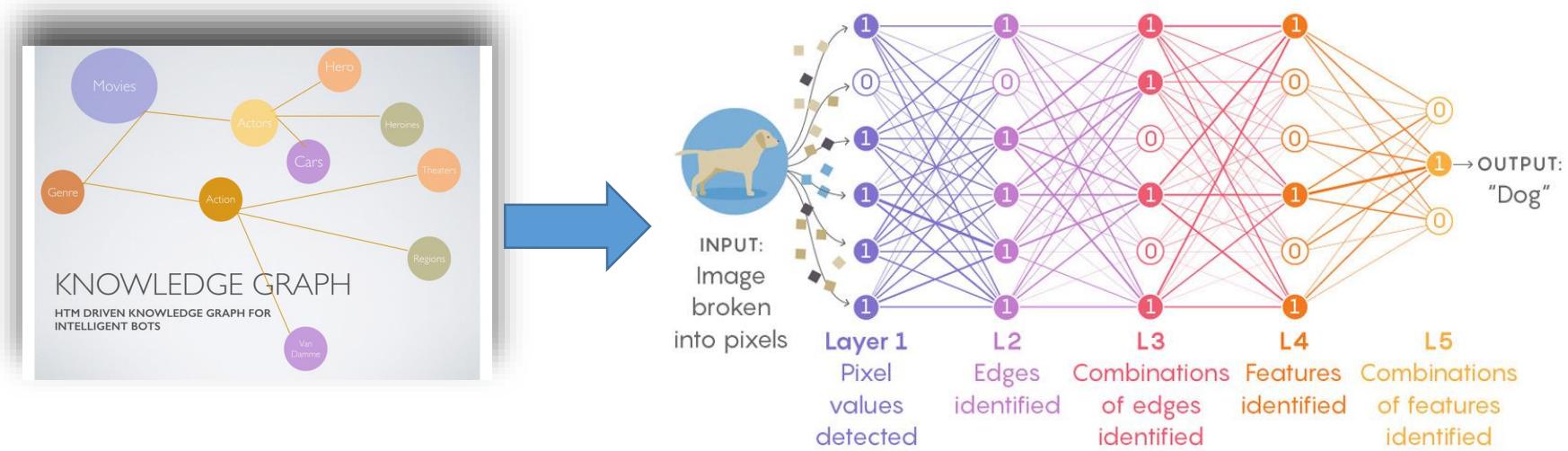
属性图模型

属性图模型组成元素——表

- 在记录的基础上，一张在列名集合 \mathcal{A} 上的表T 被定义为一个记录的包，这个包中每个记录u 的定义域是都是 \mathcal{A} ；
- 一张没有记录的空表被表示为()；
- 两张表 T_1 和 T_2 的全连接记为 $T_1 \times T_2$, 也就是将 T_1 中任意一个记录和 T_2 中任意一个记录进行连接所形成的记录集合。

基于数值的知识图谱表示-Motivation

- 随着深度学习模型应用的日益广泛，如何将知识图谱作为背景知识融合进入到深度模型中？



将知识图谱中的点与边表达成数值化的向量

基于数值的知识图谱表示-Definition

- 知识图谱的表示学习旨在将知识图谱中的元素(包括实体、属性、概念等)表示为**低维稠密实值向量**
- 知识图谱的两种表示各有其适用场景
 - 向量化表示是面向**机器**处理的
 - 符号化表示是面向**人**的理解的。
 - 相对于向量化表示，符号知识易于理解，可以实现**符号推理**。

知识图谱表示学习

- 学习实体和关系的向量化表示的关键是合理定义知识图谱中关于事实（三元组 $\langle h, r, t \rangle$ ）的损失函数 $f_r(\mathbf{h}, \mathbf{t})$

$$\operatorname{argmin} \sum_{r \in KB} f_r(\mathbf{h}, \mathbf{t})$$

- 其中 \mathbf{h} 和 \mathbf{t} 是三元组的两个实体 h 和 t 的向量化表示。
- 通常情况下，当事实 $\langle h, r, t \rangle$ 成立时，期望最小化 $f_r(\mathbf{h}, \mathbf{t})$ 。
- 基于数值表示的两种思路：
 - 基于距离的模型
 - 基于翻译的模型

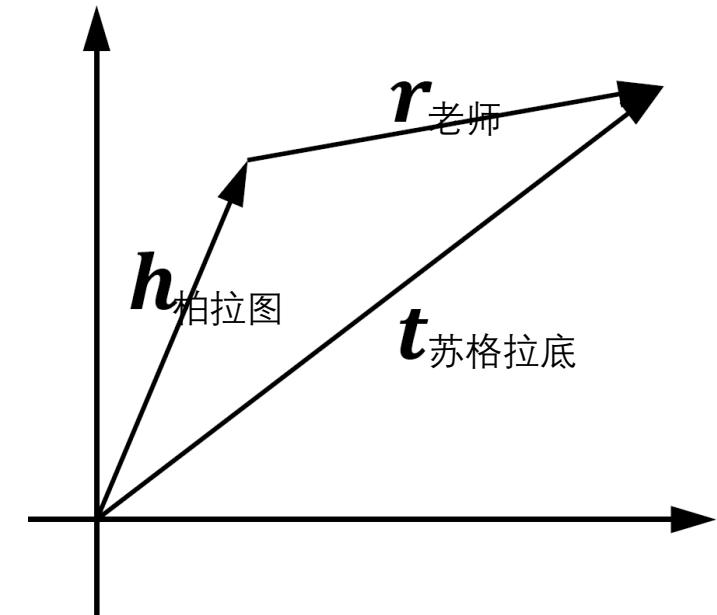
基于距离的模型

- 其代表性工作是SE模型。
- 基本思想是当两个实体属于同一个三元组 $\langle h, r, t \rangle$ 时，它们的向量表示在投影后的空间中也应该彼此靠近。
- 因此，损失函数定义为向量投影后的距离：

$$f_r(\mathbf{h}, \mathbf{t}) = \|\mathbf{W}_{r,1}\mathbf{h} - \mathbf{W}_{r,2}\mathbf{t}\|_{l_1}$$

基于翻译的模型-TransE

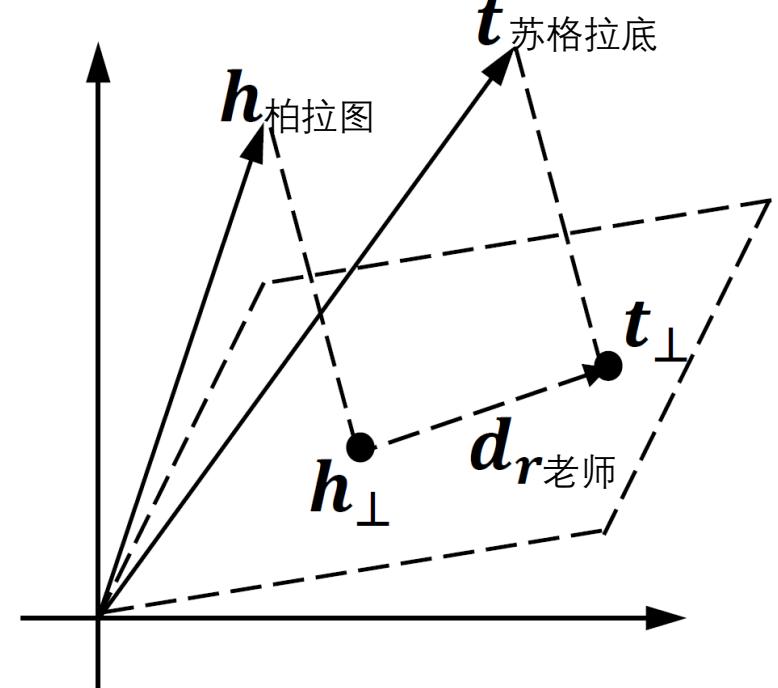
- TransE认为每个三元组实例 $\langle h, r, t \rangle$ 可以看做是头实体 h 到尾实体 t 利用关系 r 所进行的翻译。也就是， $h + r \approx t$ 。
- 基于这个思想的损失函数为：
 - $f_r(\mathbf{h}, \mathbf{t}) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{l_1/l_2}$
- Hinge loss
$$L = \sum_{(h,r,t) \in S} \sum_{(h',r,t') \in S'} [\gamma + f_r(\mathbf{h}, \mathbf{t}) - f_r(\mathbf{h}', \mathbf{t}')]_+$$
- 缺点：
 - 不适用于自反、多对一、一对多型关系
 - 如： \langle 张三，性别，男 \rangle 、 \langle 李四，性别，男 \rangle 。



基于翻译的模型-TransH

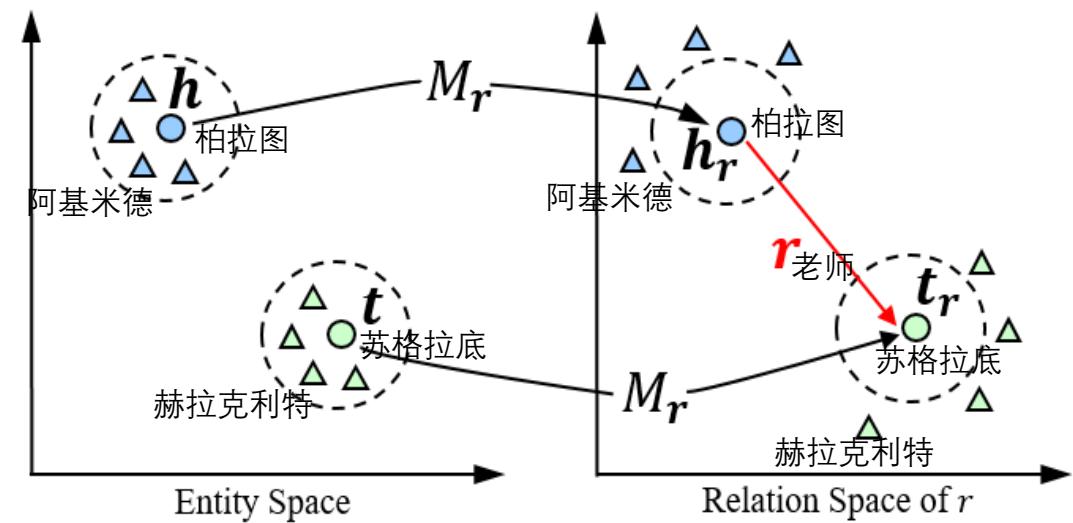
- TransE模型中 $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ 假设太强，导致在自反、一对多、多对一等关系下实体向量学习的错误。
- TransH模型放宽了 $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ 这一假设，只要求头尾实体在关系相对应的超平面上的投影彼此接近即可，即 $\mathbf{h}_\perp + \mathbf{d}_r \approx \mathbf{t}_\perp$ 。
- 头尾实体映射向量：
$$\mathbf{h}_\perp = \mathbf{h} - \mathbf{W}_r^T \mathbf{h} \mathbf{W}_r \quad \mathbf{t}_\perp = \mathbf{t} - \mathbf{W}_r^T \mathbf{t} \mathbf{W}_r$$
- 基于这个思想的损失函数为：

$$f_r(\mathbf{h}, \mathbf{t}) = \|(\mathbf{h} - \mathbf{W}_r^T \mathbf{h} \mathbf{W}_r) + \mathbf{d}_r - (\mathbf{t} - \mathbf{W}_r^T \mathbf{t} \mathbf{W}_r)\|_{l_1/l_2}$$



基于翻译的模型-TransR

- TransE、TransH中实体和关系都在**相同的空间**中进行表示，无法区分两个语义相近的实体在某些特定方面(关系)上的不同
 - 比如：**<马克思，民族，犹太民族>**，**<恩格斯，民族，德意志民族>**
- TransR仅要求头尾实体在**关系空间中的投影**彼此接近即可
- 头尾实体映射后的向量：
$$\mathbf{h}_r = M_r \mathbf{h} \quad \mathbf{t}_r = M_r \mathbf{t}$$
- 损失函数为：
$$f_r(\mathbf{h}, \mathbf{t}) = \|\mathbf{h}_r + \mathbf{r} - \mathbf{t}_r\|_{l_1/l_2}$$



基于翻译的模型-TransD

- TransR存在几个问题。
 - 1、计算复杂度高。
 - TransR利用 $\mathbf{h}_r = \mathbf{M}_r \mathbf{h}$ 和 $\mathbf{t}_r = \mathbf{M}_r \mathbf{t}$ 对头尾实体进行映射。矩阵计算仍然开销较大。
 - 2、没有考虑头尾实体的差异。
 - <柏拉图, 出生地, 希腊>
 - “柏拉图” 和 “希腊” 显然是不同类型的实体, 对 “柏拉图” 和 “希腊” 使用相同的映射显然是不合适的。
 - 3、映射应该考虑实体。
 - 实体的映射不仅仅和关系有关, 映射函数和实体和关系同时相关。

基于翻译的模型-TransD

- TransD用向量运算取代矩阵映射
- 映射函数同时考虑实体与关系：

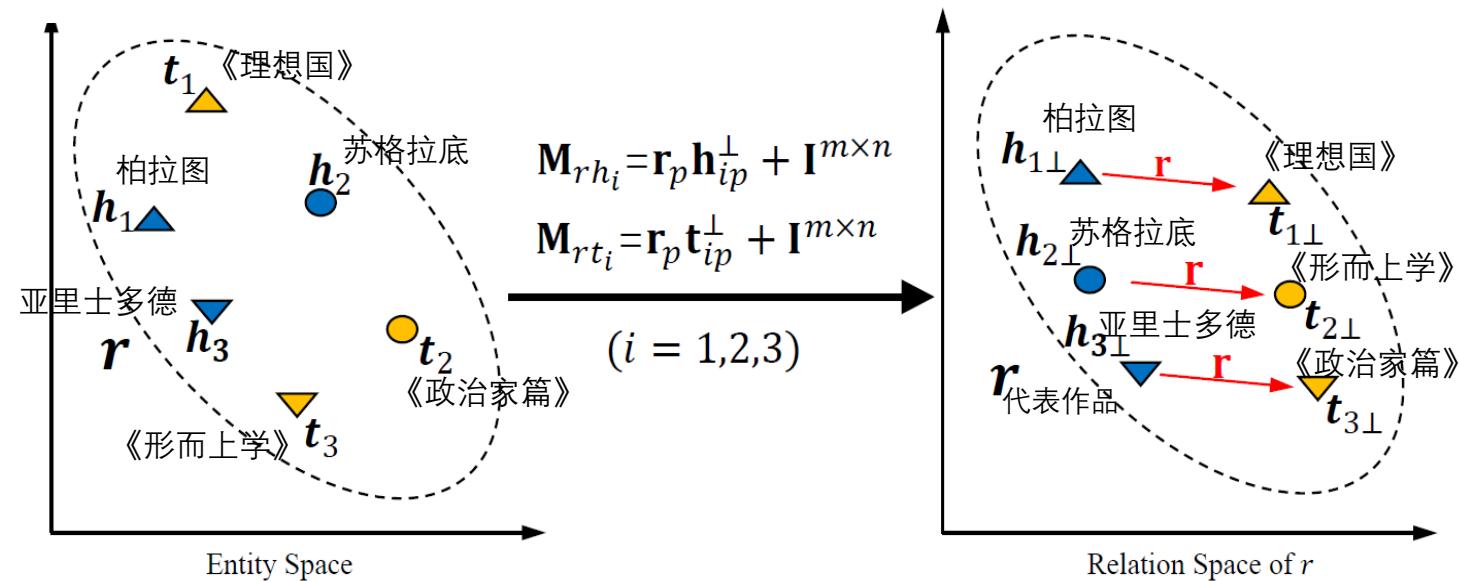
$$\mathbf{M}_{rh} = \mathbf{r}_p \mathbf{h}_p^T + \mathbf{I}^{m \times n}, \quad \mathbf{M}_{rt} = \mathbf{r}_p \mathbf{t}_p^T + \mathbf{I}^{m \times n}$$

- 映射之后的头尾实体：

$$\mathbf{h}_\perp = \mathbf{M}_{rh} \mathbf{h}, \quad \mathbf{t}_\perp = \mathbf{M}_{rt} \mathbf{t}$$

- 损失函数为：

$$f_r(\mathbf{h} + \mathbf{t}) = \|\mathbf{h}_\perp + \mathbf{r} - \mathbf{t}_\perp\|_{l_1/l_2}$$



其他相关知识表示

- 谓词逻辑 (Predicate Logic)
- 产生式规则 (Production Rule)
- 框架 (Frame)
- 树形知识表示
- 概率图模型 (Probabilistic Graphical Model)
- 马尔可夫链 (Markov Chain)

谓词逻辑 (Predicate Logic)

- 谓词 $P(x_1, x_2, \dots, x_n)$
 - BirthPlace(亚理士多德, 波伊提乌)表达了“亚里士多德生于波伊提乌”
- 否定(\neg)、析取(\vee)、合取(\wedge)、蕴含(\rightarrow)
 - \neg BirthPlace(亚理士多德, 波伊提乌)
 - BirthPlace(亚理士多德, 波伊提乌) \vee BirthPlace(亚理士多德, 色雷斯)
 - BirthPlace(亚理士多德, 波伊提乌) \wedge InfluencedBy(亚理士多德, 柏拉图)
 - BirthPlace(亚理士多德, 斯塔基拉) \rightarrow BirthPlace(亚理士多德, 色雷斯)
- 全称量词($\forall x$)和存在量词($\exists x$)
 - $\forall x$ Man(x) \rightarrow Male(x) \vee Female(x)
 - $\forall x \exists y$ Man(x) \rightarrow Father(y, x)

产生式规则 (Production Rule)

- 产生式规则是种形如“条件-动作”的规则，基本形式为：

if <condition> then <conclusion>

- R1: IF 病人发烧 AND 咳嗽 THEN 病人得了病毒性感冒
- R2: IF 病人具有长期吸烟史 THEN 该病人罹患肺部疾病
- R3: IF 病人得了病毒性感冒 THEN 给予抗病毒治疗

表达动作

(0.9) 加入不确定性

- 产生式规则优缺点

- 一种自然的、清晰的、可扩展的知识表示，擅长表达具有因果关系的过程性知识。
- 在实际应用也会碰到，规则应用时的规则冲突、规则失配等难题。

框架 (Frame)

- 框架表示是以框架理论为基础发展起来的一种结构化的知识表示

框架理论认为，人类对现实世界中各类事物的认知都是以框架的结构存储在记忆中的。当人面临新的情境时，会从记忆中找出一个合适的框架，并根据实际情况对这一框架的细节进行加工、修改和补充，形成对新情景的认识并存入人脑中。

```
<框架名>
<槽名 1>: : <侧面 11>: <值 111, 值 112, …, 值 11k1>
...
    <侧面 1m>: <值 1m1, 值 1m2, …, 值 1mkm>
<槽名 2>: <侧面 21>: <值 211, 值 212, …, 值 21k1>
...
    <侧面 2m>: <值 2m1, 值 2m2, …, 值 2mkm>
<约束>: <约束 1>
...
    <约束 m>
```

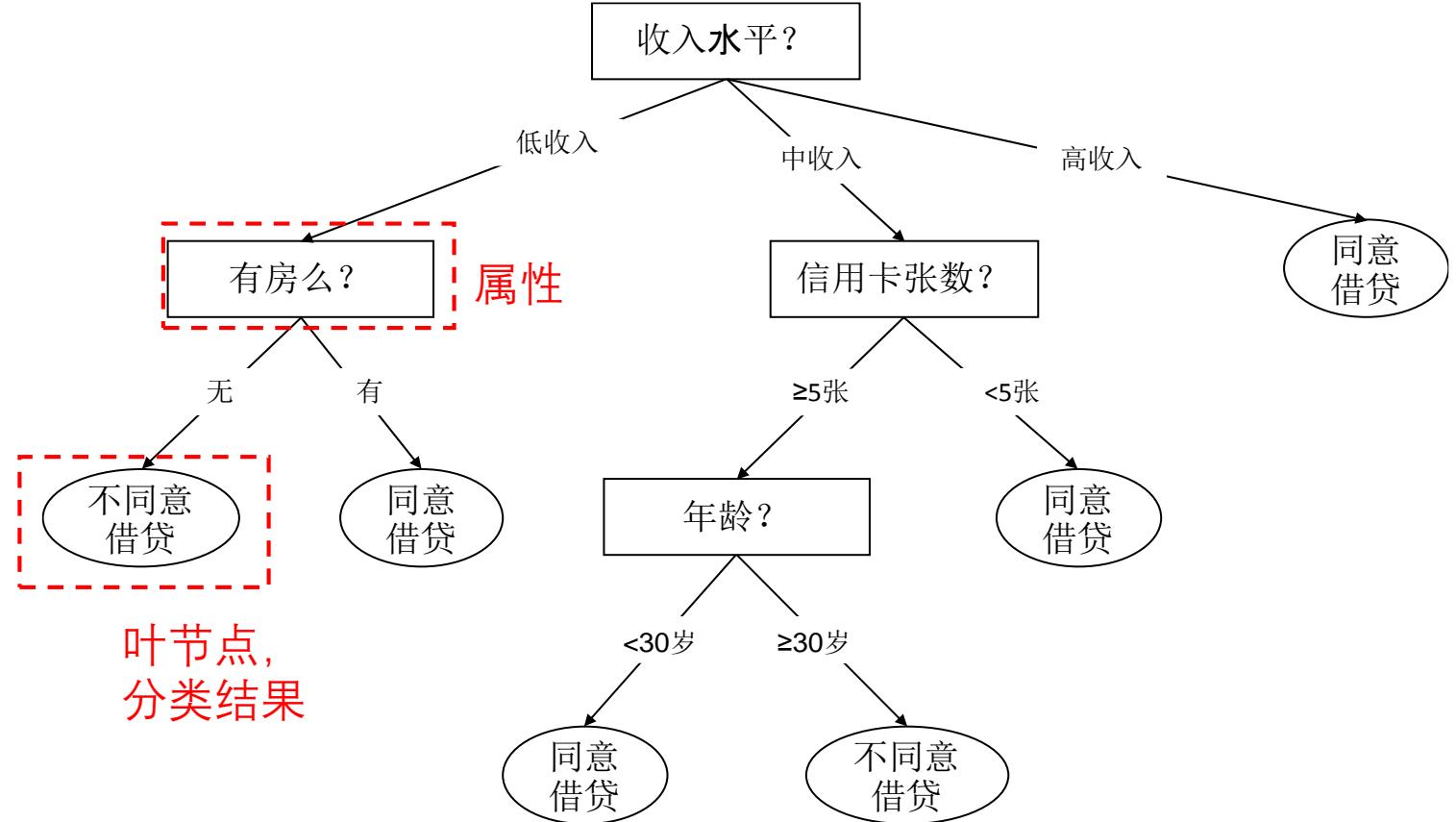
```
框架名: <哲学家>
类属: <人物>
工作: 范围: (教学, 科研)
        缺省: 哲学
性别: (男, 女)
类型: (<唯物主义哲学家>, <唯心主义哲学家>)
```

哲学家框架示例

一般的框架表示

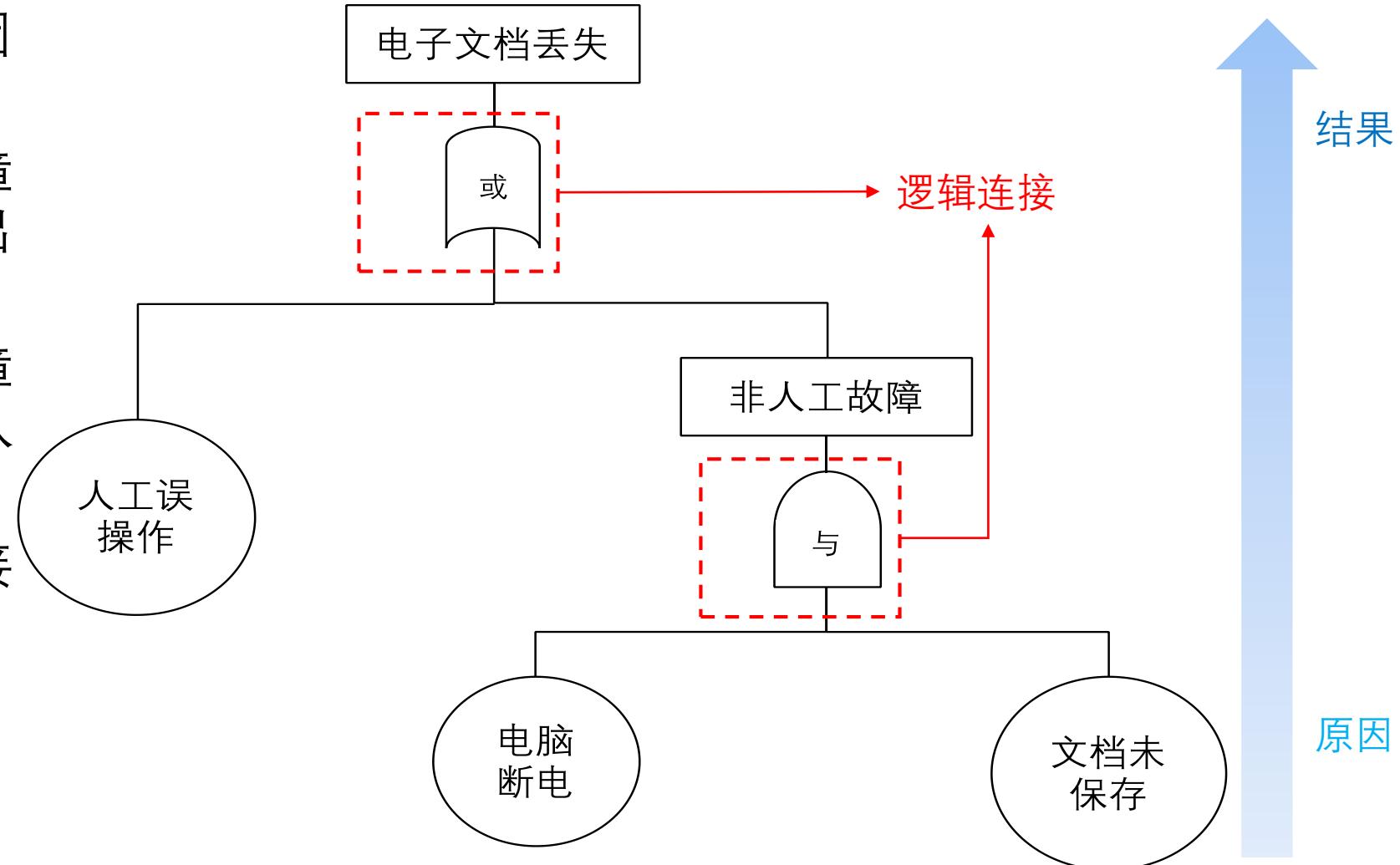
树形知识表示-决策树

- 是一种用于分类的树形结构
- 一棵决策树由根节点、若干中间节点和若干叶节点组成
- 从根节点到叶节点的每一条路径，就代表了一种分类方案



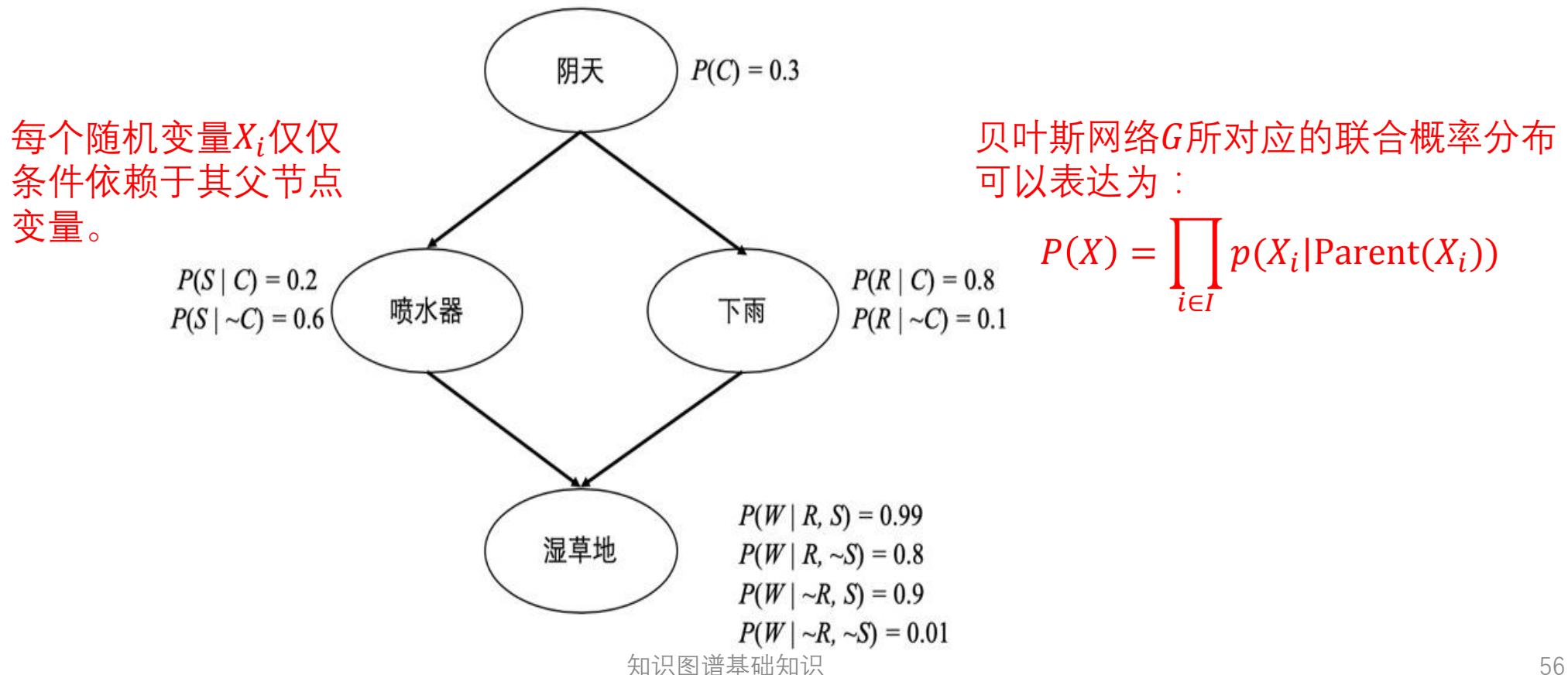
树形知识表示-故障树

- 一种树形的逻辑因果关系图
- 父节点是产生故障的结果，也称输出事件
- 子节点是产生故障的原因，也称输入事件
- 利用逻辑符号连接子节点和父节点



概率图模型 (PGM: Probabilistic Graphical Model)

- 贝叶斯网络，也称信念网络或者是有向无环图模型。
- 基于随机变量之间的**条件独立性**(conditional independence)对一组随机变量的联合分布的一种精简表示



PGM的学习和推理过程

- 学习：如何从数据中习得最优的贝叶斯网络结构
- 推理：给定贝叶斯网络和其中一些随机变量的取值设置，推断其他随机变量的分布
 - 给定因推断果(Causal Reasoning)、给定果推断因(Evidential Reasoning)，以及给定某个结果推断不同原因之间的相关作用(Intercausal Reasoning)
- 推理的问题模型：
 - Evidence: $E=e$
 - Query: a subset of variables Y
 - Task: compute $P(Y|E=e)$

PGM的应用

- 广泛应用于表达领域的决策过程
- 知识表示的角度来看，贝叶斯网络有如下优点
 - 能够准确表达决策过程中的**不确定性**。
 - 能够有效地将专家的**先验知识与数据驱动的学习方法进行融合**。
 - 专家知识：依赖专家指定贝叶斯网络所表达的随机变量
 - 数据驱动：利用数据驱动的方法学习网络的结构以及条件依赖 的概率分布值

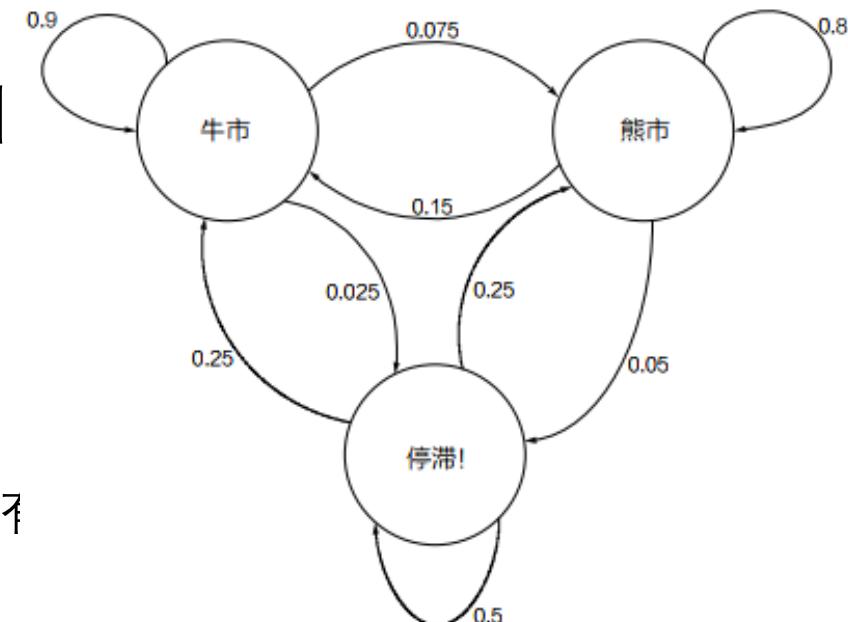
马尔可夫链 (Markov Chain, MC)

- 马尔可夫链是一种满足马尔可夫性质的离散随机变量集合。
 - 马尔可夫性(Markov Property), 是指某个随机变量序列的下一个状态仅仅和当前的状态有关, 而与之前的状态没有关系。

$$P(X_{t+1}|X_t, \dots, X_1) = P(X_{t+1}|X_t)$$

- 马尔可夫链可以表达为边上带概率的有向图

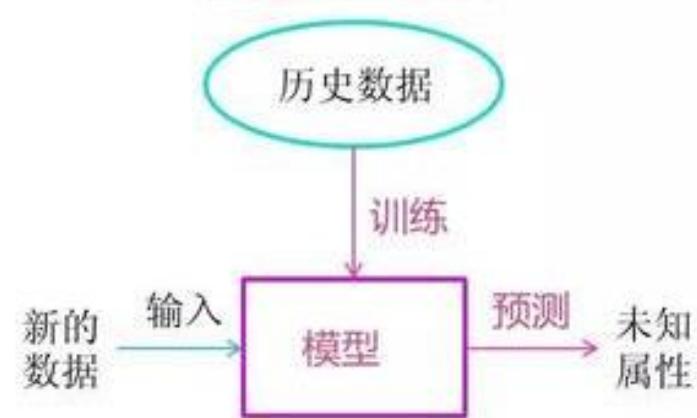
- 顶点集合 : 状态 S
- 有向边 $s_i \rightarrow s_j$: 从状态 s_i 转移到状态 s_j 的概率
 - 表示为 $\Pr(X_{t+1} = s_j | X_t = s_i)$, 称作转移概率。
- $\Pr(X_{t+1} | X_t = s_i)$
 - 当系统在 t 时刻处于状态 s_i , 下一时刻系统状态在所有可能状态上的概率分布。



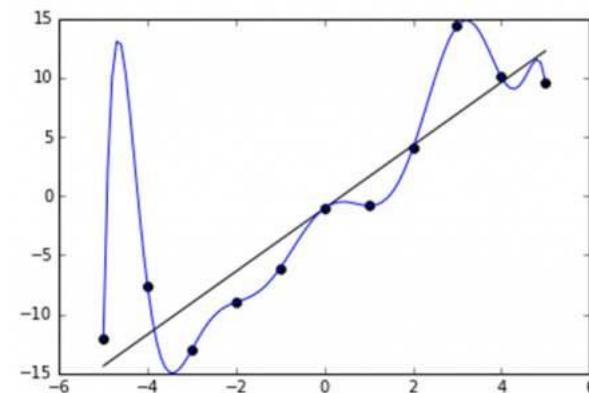
机器学习

机器学习

- 机器学习 (Machine Learning, ML)
 - 是一种从观测数据 (样本) 中寻找规律，并利用学到的规律 (模型) 对未知或无法观测的数据进行预测的方法
- 基本过程
 - 从数据中进行学习，得到某种模式、知识或规律作为模型，这个模型进一步用于未观测到的同类数据上作出归类或者预测。
- 机器学习是数据驱动的问题求解方式的代表
 - 随着大数据的日积月累，从大数据中发现统计规律，进而利用这种统计规律解决实际问题有着越来越多的应用需求。
- 机器的理论基础
 - 统计学习，旨在通过数据拟合从样本中学习统计规律



机器学习的基本过程



数据拟合，注意避免过拟合

典型ML任务：手写数字识别

- 是个典型有监督分类问题
 - 样本：一个数据与相应类标签的二元组 (x_i, y_i) ，其中 x_i 是数据， y_i 为相应的数字分类
 - 每一个样本中观测数据的均需要得到合理的表示，这种表示称为特征（feature）。
- 基本任务
 - 学习一个最优手写体预测的函数，其中 X 为所有可能的手写体数字， $Y=\{0-9\}$ 的数字标签集。
 - 函数 $F(x)$ ：有监督学习所要学习的模型（model）



图 2-14 手写数字识别示例^[44]

学习的泛化能力

- 关键问题

- 如何确保习得的模型具有较好的泛化能力。也就是说从训练集上习得的模型在未见的测试集上也能取得较好的预测结果。

$$\hat{f} = \arg \min_{f \in F} L(Y, f(X))$$

- 机器学习的三个关键要素

1. 模型选择，也就是F的确定
2. 优化准则，也就是L的确定
3. 优化方法，也就是优化问题求解过程

机器学习的一般模型函数和损失函数都是可导的。从而，可以使用梯度下降法来寻找较优的参数使得在训练集上的损失函数值较小。比如Adagrad, Adam, Rmsprop

例：线性模型指的是形如 $Ax+B$ 这样的函数所组成的函数族，而训练即寻找最优的参数A, B使得 $f(x) = Ax + B$ 在一个在训练集上计算的指标最优

例：给定训练集 $\{(x_i, y_i)\}_n$, 常见的损失函数是均方差函数, $L(Y, f(X)) = \frac{1}{n} \sum_i (\hat{y}_i - y_i)^2$;

学习的基本方式

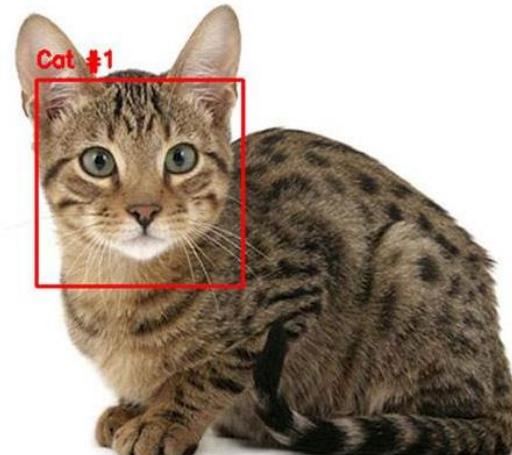
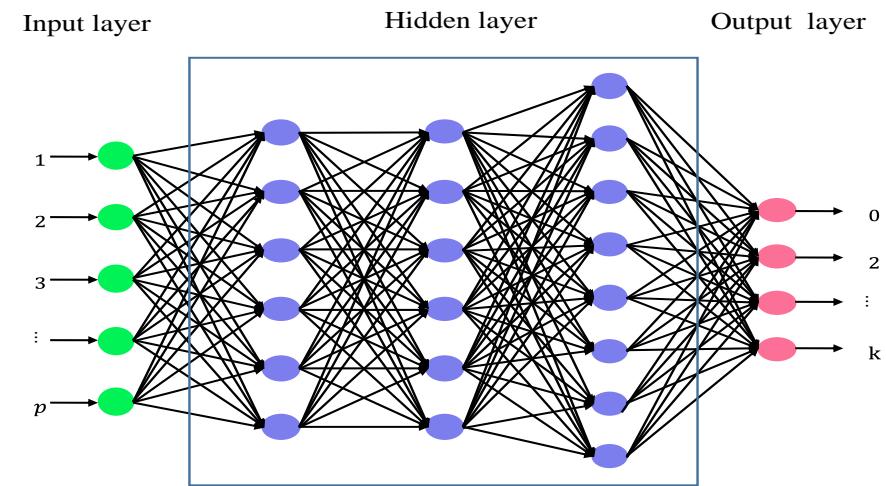
- **有监督学习**(supervised learning)
 - 包括分类 (classification) 和回归 (regression) 两类问题
- **无监督学习**(unsupervised learning)
 - 从无标签样本中学习模式
 - 包括聚类(clustering)、分布密度估计 (density estimation) 以及维度约简(dimentionality reduction)
- **半监督学习**(semi-supervised learning)
 - 介于监督学习（所有数据都有标注）和无监督学习（没有任何标注数据）之间的一种学习方式
 - 通常包含少量的标注数据和大量的未标注数据，它通过假设未标注数据的分布能够揭示标注数据之间的联系，利用未标注数据加强标注数据的学习能力
- **弱监督学习**(weakly supervised learning)
 - 利用大量容易获得的弱标注数据提升机器学习的性能

传统机器学习的局限性

- 特征主要靠专家定义，代价高；只能定义一些显性特征，难以表达隐性特征
 - Example: Logistic regression: $h_{\theta}(x) = \frac{1}{1+e^{-\theta^T x}}$
 - 模型函数简单，可直接推导最大似然得到目标函数，并带入训练数据得到参数
- 模型通常是一些较为简单的函数形式（比如如线性函数），难以表达复杂函数形式
 - Example: 2-layer MLP: $h_{\theta}(x) = W_2 f(W_1 x + b_1) + b_2$
 - 模型函数复杂，随着层数增长更加复杂，难以直接推导最大似然，需要通过后向传播进行梯度下降

深度学习概述

- 深度学习：深度神经网络的一类机器学习模型。
- 深度神经网络是在传统浅层神经网络基础上引入了更多的中间层，因而具有较深层次的神经网络模型。
- 深度神经网络由输入层-中间层-输出层的结构构成
 - 中间层（隐藏层）完成了**隐式特征**提取
 - 降低了专家特征定义的代价
 - 捕捉隐式特征
 - 引入多个中间层的深度神经网络可以表达复杂的非线性函数映射



深度神经网络的建模能力

- 深度学习 (deep learning) 即使用多个简单函数的复合作为机器学习的模型
 - 而其需要训练的参数为每个简单函数中的参数
- 常见的作为深度学习中的简单函数是带激活函数的全连接层, 即 $F_i(x) = \sigma(Ax + B)$, 这里的 σ 可以选择tanh或sigmoid等多种非线性函数
 - 以它为基本函数复合而成的深度学习模型即为传统的多层神经网络
- 通常需要针对不同的任务选择不同的函数进行组合以获得较好效果

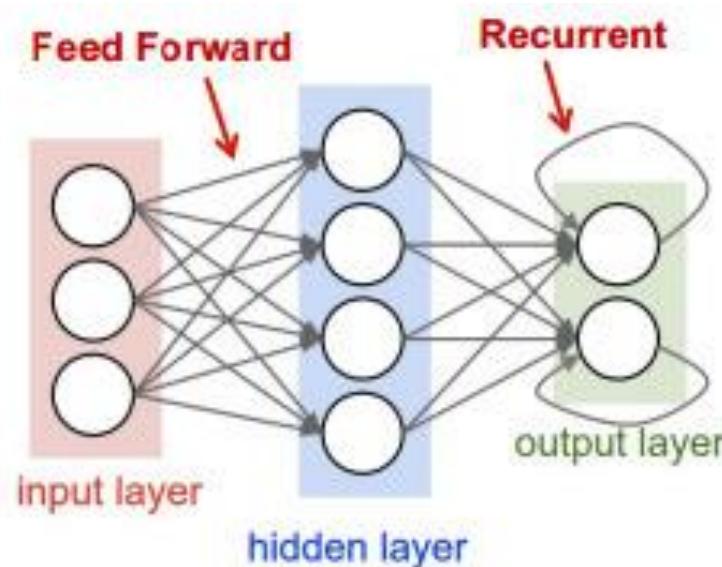
$$y = F(x) = F_L(\dots F_3(F_2(F_1(x))))$$

Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1 + e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
TanH		$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2 + 1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parametric Rectified Linear Unit (PReLU) [2]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU) [3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1 + e^x)$	$f'(x) = \frac{1}{1 + e^{-x}}$

深度学习中的激活函数, 决定了神经元的输出形式

典型DL模型：前馈神经网络(feedforward neural network)

- 也称之为多层感知器(multilayer perceptron)
- 层与层之间的神经元采取**全连接**方式
 - VS 循环神经网络，输出是否可作为输入
- 通常使用反向传播算法进行参数学习。



(c) Feedforward versus feedback (recurrent) networks

https://static.leiphone.com/uploads/new/article/740_740/201704/58defb01bfe8f.png?imageMogr2/format/jpg/quality/90

卷积神经网络

- 全连接神经网络模型的两个弊端
 - 参数太多，从而更容易产生过拟合
 - 比如对一个 512×512 的3通道图像来说，仅仅一层全连接层就拥有上百万参数。
 - 没有使用局部不变性
 - 图像相关的任务一般具有局部不变性。例如，一张狗的照片向右平移5个像素并不会影响人类判断图中是一只狗
- 卷积神经网络（Convolutional Neural Network, CNN）是一类用来引入了卷积操作来代替全连接层的矩阵运算的神经网络模型
 - 降低参数复杂性：**参数共享，稀疏互联**
 - 在图像处理的任务上获得了巨大成功：**平移不变性**
- CNN网络结构通常由输入、卷积函数层、池化层以及全连接层构成

卷积神经网络

- CNN的根本特征：卷积层的引入

- 卷积层使用一个相对原始数据来说规模很小的卷积核（convolution kernel，有时又称为过滤器）作为参数，将输入数据和卷积核进行卷积运算
- 对输入数据的每个区域都进行相同的卷积运算

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Input Data

1	0	1
0	1	0
1	0	1

Kernel

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Image

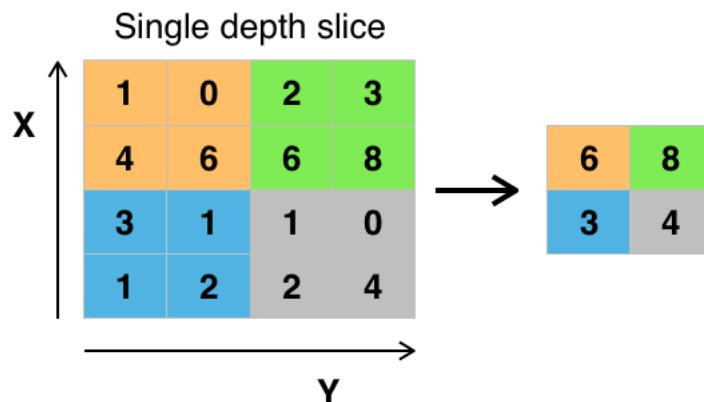
4		

Convolved Feature

https://pic2.zhimg.com/v2-a7ecf6026ad52ef15308b75426ab915f_b.jpg

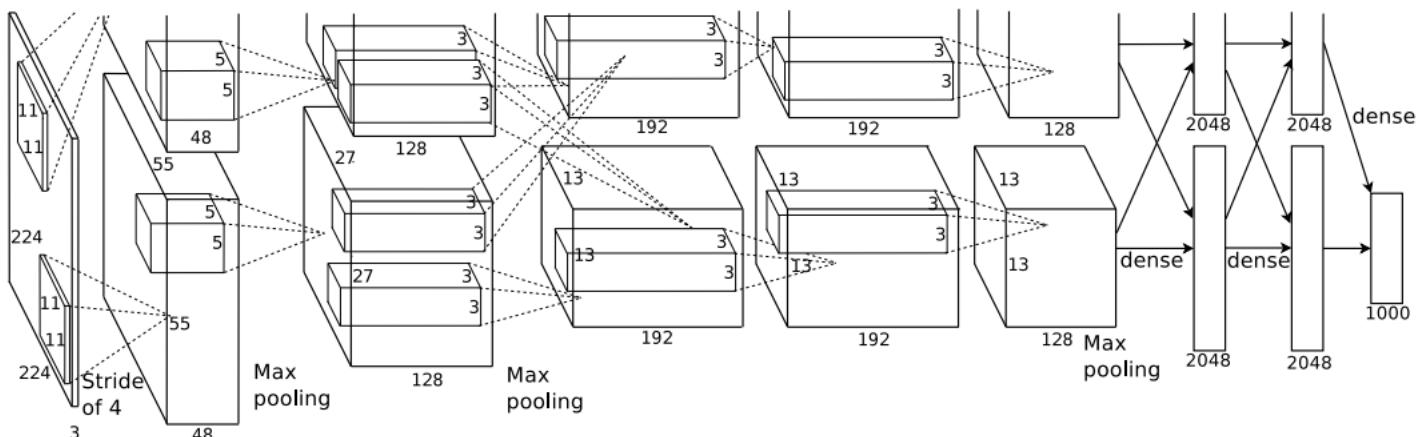
卷积神经网络

- CNN的卷积层之后，往往还跟着一个**池化层**。池化层使用池化操作进一步降低表示的复杂性
 - 池化函数有多种实现方式。如最大池化操作，给出相邻区域内的最大值



步幅为2，池化窗口为2的最大池化

https://upload.wikimedia.org/wikipedia/commons/e/e9/Max_pooling.png

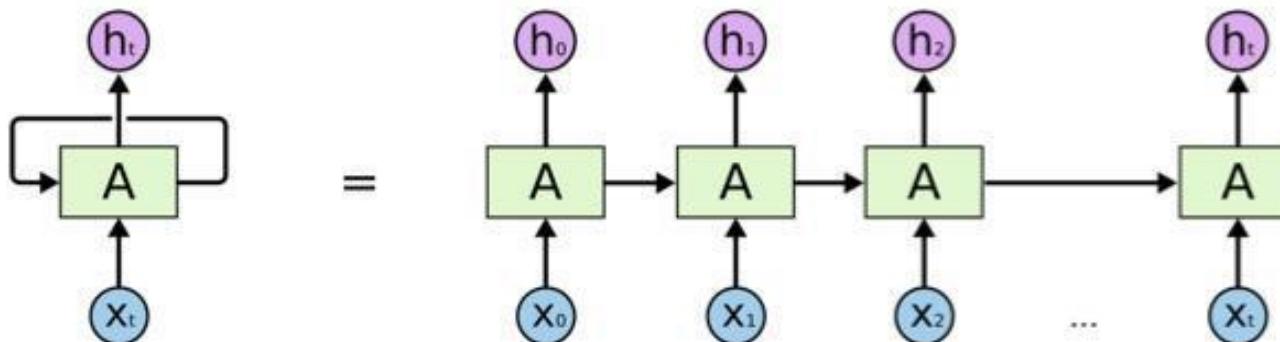


AlexNet网络结构

A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012.

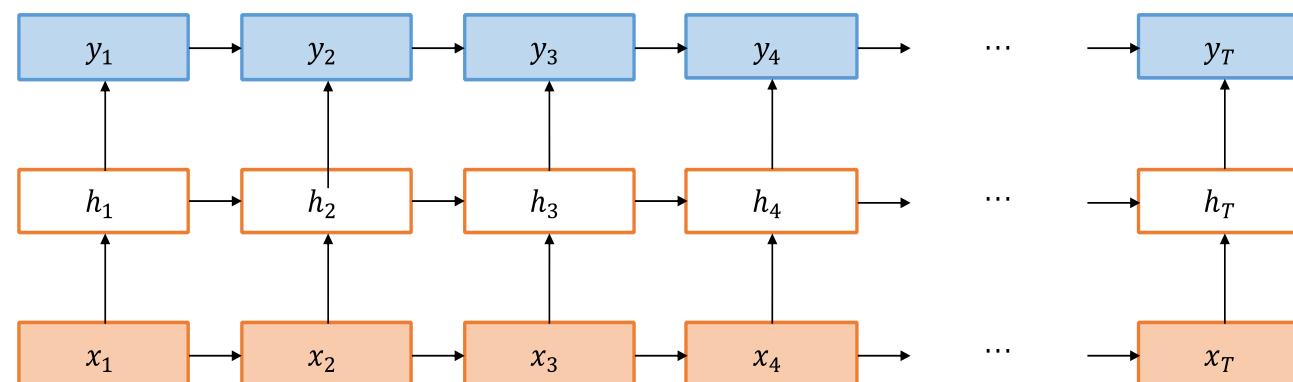
循环神经网络

- 循环神经网络 (Recurrent Neural Network, RNN) 是一类专用于处理**序列数据**的神经网络模型
 - 数据的t时刻状态往往决定于其前序状态
 - RNN实用化根本原因也在于参数共享、稀疏连接
- RNN被广泛应用于自然语言等可以建模为**序列数据处理**中



循环神经网络

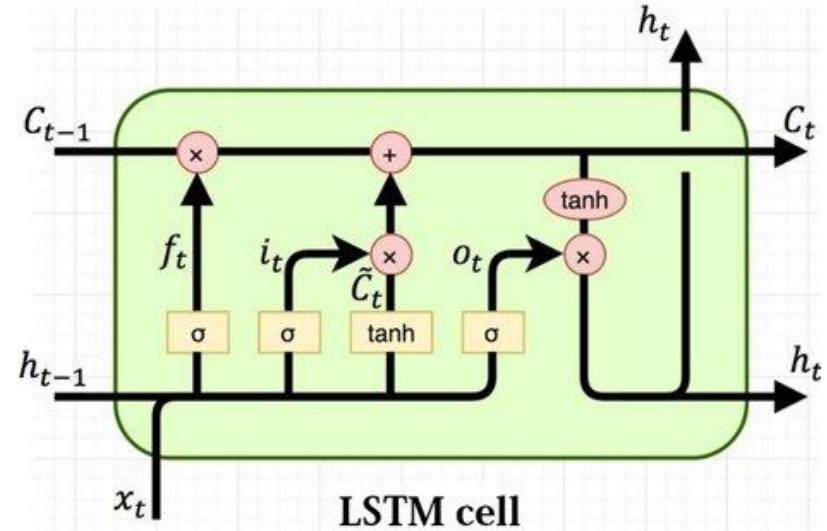
- 给定一个输入序列 $x_{1:T} = (x_1, x_2, \dots, x_t, \dots, x_T)$, RNN通过下面公式更新带反馈边的隐藏层的活性值
 - $h_t = f(h_{t-1}, x_t)$ 。
 - 隐藏单元 h_{i-1} 被称为记忆或状态, 它是由序列的前*i* – 1个元素生成的。在处理第*i*个输入单元 x_i 后, 同时会产生第*i*时刻的记忆 h_i , 和下一个元素一起作为下一时刻的输入
- RNN不仅处理当前元素, 还考慮到序列前面的元素。



RNN变种

- 双向循环神经网络

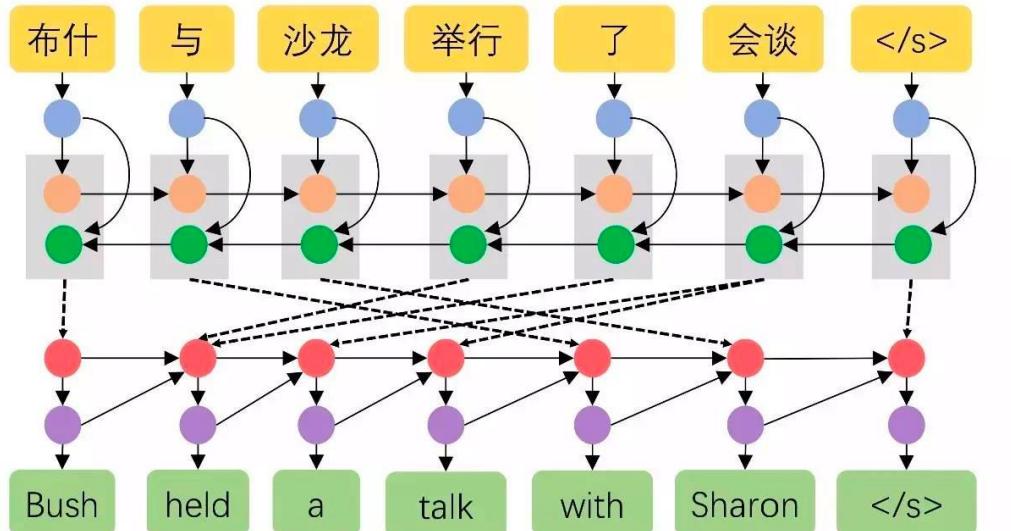
- 在很多实际应用中，序列中的某个数据不仅与其前序数据相关，也与后续数据有关
- 在单向循环神经网络的基础上引入额外的循环层实现序列数据中的反向影响



注意力机制

- 注意力
 - 人类在处理大量信息时是有**选择性**的，在关注一些信息的同时忽略其他信息
 - **许多任务中，输出往往只和输入数据中的部分相关，而其他输入数据可以忽略**
 - Eg, 阅读理解、机器翻译
- 注意力机制 (Attention)
 - 在深度神经网络中引入注意力机制，以使得模型在每一阶段的输出（例如，翻译任务中生成句子的每一个词）只需要从输入中的某些片段计算产生，而不需要处理整个输入

- 利用注意力机制动态计算源语言端相关上下文



(Bahdanau et al., 2015) 30

自然语言处理

基本概念

- NLP可以在词汇、语句、段落以及篇章等不同粒度处理自然语言
 - 自然语言中提及的词汇对应到人类认知的实体与概念
 - “亚理斯多德” 指代作为哲学家的实体
 - “哲学家” 指代特定人群的概念
- 文本作为知识图谱的数据来源，需要对语句做各种处理
 - 词法分析(lexical analysis)
 - 语法分析(syntactic analysis)
 - 语义分析(semantic analysis)
 - 语用分析(pragmatic analysis)
- NLP可以是以上层面进行，其中前三者文本与知识图谱密切相关

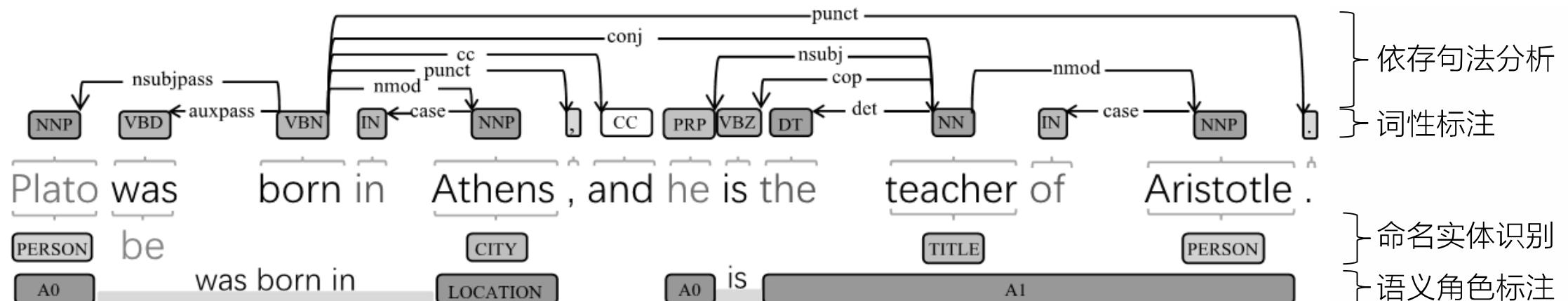
NLP 常见操作	NLP 常见操作
断句(sentence segmentation)	依存句法(dependency parsing)
分词(tokenization)	命名实体识别(named entity recognition)
词性标注(part-of-speech tagging)	共指消解(conference resolution)
词形还原(lemmatization)	语义角色标注(semantic role labeling)
识别停用词(identifying stop-words)	...

基本概念

- 断句
 - 一般通过标点符号即可实现
- 分词
 - 指对文本进行词汇的切割
 - 比如，“小明吃苹果”，分词之后为，“小明”、“吃”、“苹果”
 - 常见的中文分词工具有jieba、SnowNLP和NLPIR等
- 词性标注
 - 给句子中每个词标记一个词性
- 词形还原
 - 将某一单词还原至原型，包括将名词的单复数、be动词以及动词的过去时态和现在进行时等还原成相应原形
- 识别停用词
 - 单词“the”，“a”，“an”，“of”等是文本中的常见高频词，对于句子某些处理与分析任务往往起到噪声作用，被称为停用词

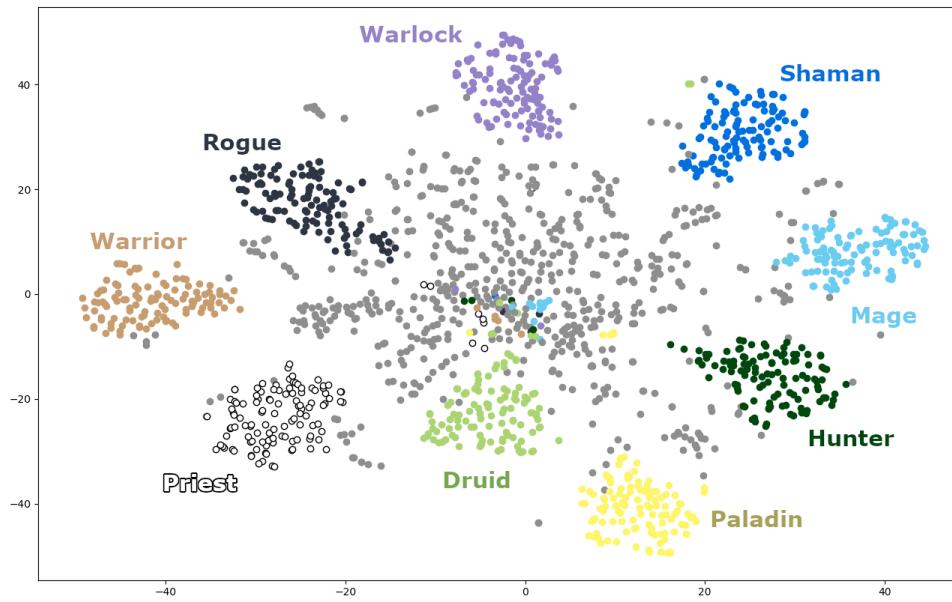
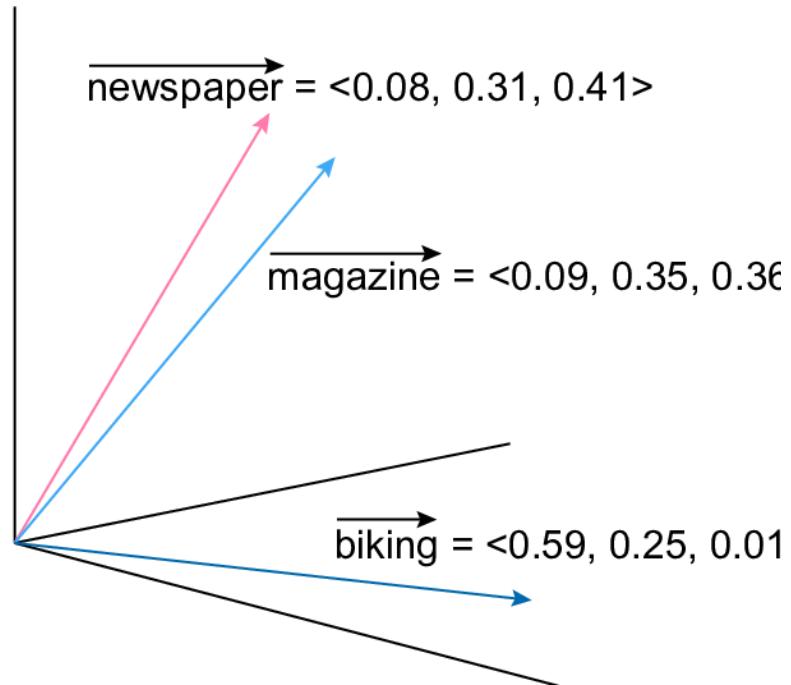
基本概念

- 依存句法分析
 - 识别句子的语法结构。通过分析句中各语言成分之间的依存关系。其分析结果是一颗依赖关系树
 - 句法结构的识别是对自然语言浅层理解的一种重要形式
- 语义分析
 - 命名实体识别：识别句子的词序列中具有特定意义的实体，并将其区分为人名、机构名、日期、地名、时间和职位等类别的任务
 - 共指消解：指代词共指现象，如it/he/she等，又比如缩写简称，复旦大学通常称为复旦
 - 语义角色标注：找出谓语的相应语义角色成分，包括核心角色成分（施事者、受事者等）和附属语义角色成分（如时间、目的、地点、原因等）



文本的向量化表示

- 文本表示是指将文字表示成计算机能够运算的数字或向量
 - 一般称之为词嵌入 (Word Embedding)，即将文本中的词嵌入到文本空间，用一个向量进行表示
- 文本表示分为**离散表示**和**分布式表示**两大类型



文本的向量化表示

- 离散表示
 - One-hot表示
 - 在语料库中，为每个字/词编码一个索引，根据索引进行one-hot表示，具有稀疏性
 - 词袋表示：词袋表示常用于文本表示，直接将文本中单词的one-hot向量进行相加
 - N-gram表示：与词袋模型原理相似。将相邻N个单词进行索引编码。比如，Bi-gram是将相邻两个单词进行索引
 - 离散表示并不能表示词语间的语义关系，比如同义、反义或类比关系

One-hot表示

Rome	=	[1, 0, 0, 0, 0, 0, ..., 0]
Paris	=	[0, 1, 0, 0, 0, 0, ..., 0]
Italy	=	[0, 0, 1, 0, 0, 0, ..., 0]
France	=	[0, 0, 0, 1, 0, 0, ..., 0]

文本的向量化表示

- 分布式假设(Distributional Hypothesis)
 - 两个语义相似的词通常具有相类似的上下文
 - 分布式表示
 - 基于词计数(word counting)
 - 根据大型文本语料库中特定字词与临近字词共同出现的统计值表达词向量
 - 基于语言模型
 - 将词看作待学习的参数，根据学习的词向量建立自然语言的预测模型，往往伴随着语言模型的学习过程
 - word2vec的两个模型：CBOW和skip-gram模型
- What is **tezgüino**?
- A bottle of **tezgüino** is on the table.
 - Everybody likes **tezgüino**.
 - **Tezgüino** makes you drunk.
 - We make **tezgüino** out of corn.
- The contexts in which a word appears tell us a lot about what it means