

知识图谱 (Knowledge Graph)

吉建民

2025 年 4 月 6 日

Table of Contents

知识图谱是什么

知识图谱的表示、存储与查询

- RDF 方式

 - OWL 简介

- PG (Property Graph) 方式

- RDF 方式 vs. PG 方式

知识图谱的推理

- Graph-based Inference Algorithms

- Ontology-based inference Algorithms

 - 基于符号表示的推理

 - 基于嵌入学习的推理

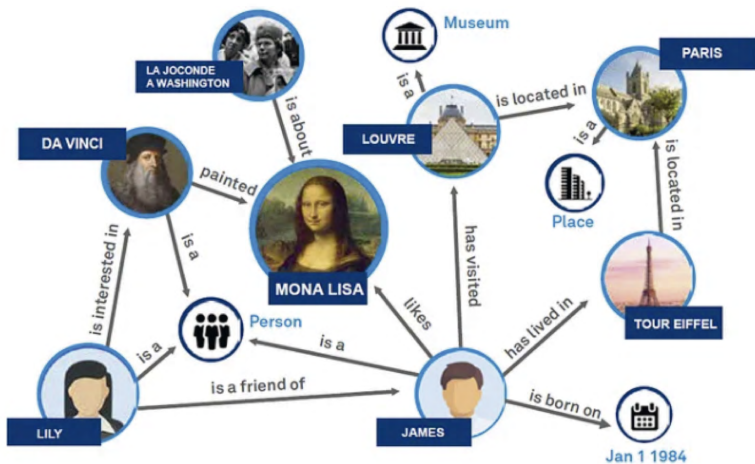
知识图谱的构建

知识图谱技术发展



知识图谱是什么

- 知识图谱就是由“实体-关系-实体”三元组、实体及其相关“属性-值”对，联结构成的网状知识结构



知识图谱的来源

- ▶ 知识图谱最早由 Google 于 2012 年 5 月发布，用于智能化搜索功能，对搜索结果进行系统的知识整理
- ▶ 知识图谱两大来源：
 - ▶ 知识表示与推理领域中语义网络 (Semantic Networks)、描述逻辑 (Description Logics)、概念图 (Conceptual Graphs)
 - ▶ 知识图谱可以从 KR 的角度，看作 RDF (Resource Description Framework) 三元组的知识库，采用 RDF 查询语言如 SPARQL 进行查询和推理
 - ▶ 数据库系统中网状数据库 (Network Databases) 和三元组存储 (Triple Stores)
 - ▶ 知识图谱也可以从图数据库 (Graph Database) 的角度，看作属性图 (Property Graph) 表示，在图数据库如 Neo4J 中存储，并用相应的查询语言 Cypher 进行查询

知识图谱的特点

- ▶ 知识图谱既可以从 RDF 的角度进行 KR 知识库的构建和形式化推理，也可以从属性图的角度进行高效存储和查询，并采用一系列学习方法进行知识图谱的构建、推理、融合和补全，甚至可以进行知识图谱嵌入 (KG Embedding)，将知识图谱知识转换为向量表示，进一步与深度学习技术融合
- ▶ 区别于大部分 KR 方法，知识图谱具有以下特点：
 - ▶ (1) **可大规模扩展 (Scalable)**，在保持高效查询和推理的条件下支持大规模知识库。例如，GDELT 全局知识图谱 (Global Knowledge Graph) 拥有 35 亿条知识，描述 3.64 亿个实体
 - ▶ (2) **自底向上构造 (Bottom-up construction)**，可以从底层数据中提取知识，自底向上的构造更符合问题需求的知识图谱，不过度依赖人工进行自顶向下的设计
 - ▶ (3) **支持多种混合方式构造 (Mixed models of construction)**，知识图谱既可以基于人工规则和数据库进行冷启动，也可以采用多种学习方法，灵活的从结构化数据、文本、图像等数据来源进行自动生成和扩展

知识图谱的应用

KG 为各类应用提供相应的知识库

- ▶ 语义搜索：Things, not Strings
- ▶ 知识问答：给万物都挂接一个背景知识库
- ▶ 推荐系统：KG 辅助的推荐系统，提升推荐结果的可解释性
- ▶ 大数据分析：KG 辅助的大数据分析
- ▶ 语言理解：KG 辅助的语言理解
- ▶ 视觉理解：KG 辅助的视觉理解
- ▶ 语义辅助设备互联

KG 应用举例：

- ▶ 阿里新零售知识图谱
- ▶ 中医药语义网络
- ▶ 通信网络大规模故障诊断知识图谱
- ▶ 金融知识图谱

Table of Contents

知识图谱是什么

知识图谱的表示、存储与查询

RDF 方式

OWL 简介

PG (Property Graph) 方式

RDF 方式 vs. PG 方式

知识图谱的推理

Graph-based Inference Algorithms

Ontology-based inference Algorithms

基于符号表示的推理

基于嵌入学习的推理

知识图谱的构建

知识图谱技术发展

知识图谱的表示

- ▶ 知识图谱的主流表示方式为两类：
 - ▶ **RDF 方式**：RDF 是 W3C 推动的语义数据交换标准与规范，有严格的语义逻辑基础（描述逻辑，Description Logic），支持推理，并兼容更负责的本体表示语言 OWL
 - ▶ **PG 方式**：属性图 (Property Graph) 是工业界最常用的图谱建模方法，属性图数据库充分利用图结构特点做了性能优化，实用度高，但不支持符号推理

Table of Contents

知识图谱是什么

知识图谱的表示、存储与查询

RDF 方式

OWL 简介

PG (Property Graph) 方式

RDF 方式 vs. PG 方式

知识图谱的推理

Graph-based Inference Algorithms

Ontology-based inference Algorithms

基于符号表示的推理

基于嵌入学习的推理

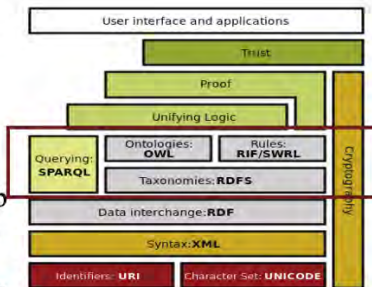
知识图谱的构建

知识图谱技术发展



RDF (Resource Description Framework)

- RDF is an **de facto standard** for Knowledge Graph (KG).
- RDF is a **language** for the conceptual modeling of information about web resources
- A **building block** of semantic web
- Make the information on the web and the interrelationships among them "**Machine Understandable**"



RDF (Resource Description Framework)

An RDF triple:



A set of such triples is called an RDF graph.

There can be three kinds of nodes:

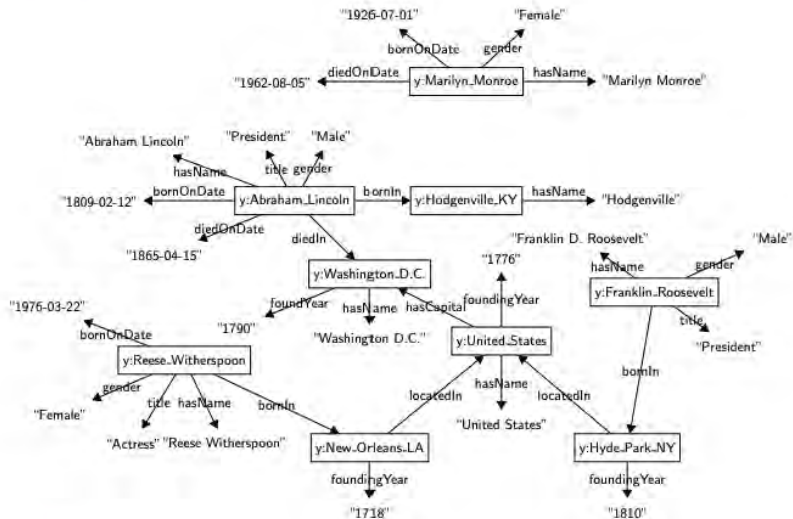
- ▶ An **IRI** is an Internationalized Resource Identifier used to uniquely identify resources on the web.
- ▶ A **literal** is a value of a certain data type, for example, string, integer, etc.
- ▶ A **blank node** is a node that does not have an identifier, and is similar to an anonymous or an existential variable.

例：RDF 表达

```
@prefix foaf: <http://xmlns.com/foaf/0.1/>  
@prefix ex: <http://example.org/>  
@prefix exstaff: <http://example.org/staff/>  
@prefix exterms: <http://example.org/terms/>
```

```
ex:art foaf:knows ex:bob  
ex:art foaf:knows ex:bea  
ex:bob foaf:knows ex:cal  
ex:bea foaf:age 23  
exstaff:85740 exterms:address _:art_address  
_:art_address exterms:street "1501 Grant Avenue"  
_:art_address exterms:city "Bedord"  
_:art_address exterms:state "Massachusetts"  
_:art_address exterms:zip "01730"
```

RDF Graph



RDF Dataset

- ▶ An **RDF vocabulary** is a collection of IRIs intended for use in RDF graphs.
- ▶ **RDF graphs** are atemporal in the sense that they provide a static snapshot of data.
- ▶ An **RDF dataset** is a collection of RDF graphs and comprises exactly one default graph that can be empty and does not need to have a name, and one or more named graphs.

RDF Data Volumes are Growing

- ▶ LOD consists of 1255 datasets with (estimated) >100B triples



>84B triples



2B triples



9.5B triples



1.1 billion statements

SPARQL Query Language

- ▶ SPARQL contains capabilities for **querying required and optional graph patterns** along with their conjunctions and disjunctions.
- ▶ SPARQL also supports **extensible value testing and constraining queries** by source RDF graph. The results of SPARQL queries can be sets or RDF graphs.
- ▶ The query consists of two parts:
 - ▶ the **SELECT clause** identifies the variables to appear in the query results, and
 - ▶ the **WHERE clause** provides the graph pattern to match against the data graph.

例：SPARQL 查询

```
SELECT ?person
WHERE
<http://example.org/art> <http://xmlns.com/foaf/0.1/knows> ?person
```

?person1
<http://example.org/bob>
<http://example.org/bea>

```
SELECT ?person ?person1
```



```
WHERE
<http://example.org/art> <http://xmlns.com/foaf/0.1/knows> ?person
?person <http://xmlns.com/foaf/0.1/knows> ?person1
```

Above query returns the following result set on our data graph.

?person1	?person2
<http://example.org/bob>	<http://example.org/cal>
<http://example.org/bob>	<http://example.org/cam>
<http://example.org/bea>	<http://example.org/coe>
<http://example.org/bea>	<http://example.org/cory>



例：SPARQL 查询

```
@prefix dc:
<http://purl.org/dc/elements/1.1/> .
@prefix : <http://example.org/book/> .
@prefix ns: <http://example.org/ns#> .
```

```
:book1 dc:title "SPARQL Tutorial" .
:book1 ns:price 42 .
:book2 dc:title "The Semantic Web" .
:book2 ns:price 23 .
```

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
SELECT ?title
WHERE { ?x dc:title ?title
        FILTER regex(?title, "^SPARQL")
      }
```

title
"SPARQL Tutorial"

RDF Dataset 存储

- ▶ 三元组表：最简单的存储，（关系数据库）直接存储三元组
- ▶ 属性表存储：基于关系数据库实现，以实体类型为中心，把属于同一个实体类型的属性组织为一个表，即属性表进行存储
- ▶ 垂直划分表（二元表）：基于关系数据库实现，对三元组按属性分组，为每个属性在关系数据库中建立一个包含（Subject、Object）两列的表，由于一个知识图谱中属性数量是有限的，表的总体数量是可控的
- ▶ 全索引结构：性能最好的存储方式，仅维护一张三元组表，但增加了多个方面的优化手段：
 - ▶ 1. 建立 Mapping Table，将所有的字符串首先映射到唯一的数字 ID，从而压缩存储空间
 - ▶ 2. 建立 6 种索引：SPO, SOP, PSO, POS, OPS, OSP，即分别建立 Subject-Predicate-Object 等 6 个方面的全索引，从而覆盖多个维度的图查询需求
 - ▶ 3. 三元组基于字符串排序，并利用 clustered B+ tree 来组织索引以进一步优化索引检索的效率

RDF Dataset 存储

存储方案		优点	缺点	代表性系统
基于 关系	三元组表	① 存储结构简单	① 大量自连接操作开销巨大	3store
	水平表	① 知识图谱的邻接表,存储方案简单	① 可能超出所允许的表中列数目的上限 ② 表中可能存在大量空值 ③ 无法表示一对多联系或多值属性 ④ 谓语的增加、修改或删除成本高	DLDB
	属性表	① 克服了三元组表的自连接问题 ② 解决了水平表中列数目过多的问题	① 真实知识图谱需建立的关系表数量可能超过上限 ② 由于知识图谱的灵活性,表中可能存在大量空值 ③ 无法表示一对多联系或多值属性	Jena
	垂直划分	① 解决了空值问题 ② 解决了多值问题 ③ 能够快速执行不同谓语句表的连接查询	① 真实知识图谱需维护大量谓语句表 ② 复杂知识图谱查询需执行的表连接操作 ③ 数据更新维护代价大	SW-Store
	六重索引	① 每种三元组模式查询均可直接使用对应索引快速查找 ② 通过不同索引表之间的连接操作直接加速知识图谱上的连接查询	① 需要花费6倍的存储空间开销和数据更新维护代价 ② 复杂知识图谱查询会产生大量索引表连接查询操作	RDF-3X Hexastore

网络本体语言 OWL (Web Ontology Language)

- ▶ OWL 建立在 RDF 之上，增加更丰富的表达
- ▶ OWL 是为处理网络的上信息
- ▶ OWL 被设计为便于被计算机理解
- ▶ OWL 并不便于被人阅读
- ▶ OWL 通过 XML 描述
- ▶ OWL 可以映射到描述逻辑 (description logic)，是一阶逻辑的子集，可判定，具有自动推理机，如 FaCT 和 RACER

OWL 的三个子语言

- ▶ OWL Lite
 - ▶ OWL DL 的一个子集，良好的可计算性
 - ▶ 提供给那些只需要一个分类层次和简单约束的用户
- ▶ OWL DL
 - ▶ 基于描述逻辑，与一部分 RDF 不兼容，旨在支持已有的描述逻辑商业处理（business segment）和具有良好计算性质的推理系统
 - ▶ 包括了 OWL 语言的所有成分，但有一定的限制，如类型的分离
 - ▶ 保证计算的完全性，可判定性，计算复杂度较高，暂时没有完全的推理机实现
- ▶ OWL Full
 - ▶ 最强的表达能力，不可判定
 - ▶ 支持那些需要尽管没有可计算性保证，但有最强的表达能力的用户

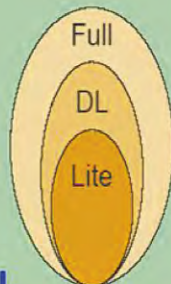
在表达能力和推理能力上，每个子语言都是前面的语言的扩展

Language Layers

■ OWL Light

- (sub)classes, individuals
- (sub)properties, domain, range
- conjunction
- (in)equality
- cardinality 0/1
- datatypes
- inverse, transitive, symmetric
- hasValue
- someValuesFrom
- allValuesFrom

} **RDF Schema**



■ OWL DL

- Negation
- Disjunction
- Full Cardinality
- Enumerated types

■ OWL Full

- Allow meta-classes etc

OWL: constructors

Constructor	Abbreviation	Example
intersectionOf	$C_1 \wedge \dots \wedge C_n$	Human $\wedge$ Male
unionOf	$C_1 \vee \dots \vee C_n$	Doctor $\vee$ Lawyer
complementOf	$\neg C$	$\neg$ Male
oneOf	$\{x_1 \dots x_n\}$	{john, mary}
toClass	$\forall P.C$	$\forall$ hasChild.Doctor
hasClass	$\exists P.C$	$\exists$ hasChild.Lawyer
hasValue	$\exists P.\{x\}$	$\exists$ citizenOf.{USA}
minCardinalityQ	$\geq n P.C$	≥ 2 hasChild.Lawyer
maxCardinalityQ	$\leq n P.C$	≤ 1 hasChild.Male
cardinalityQ	$= n P.C$	$= 1$ hasParent.Female

+ XML Schema datatypes:

- int, string, real, etc

OWL: Axioms

Axiom	Abbreviation	Example
subClassOf	$C_1 \sqsubseteq C_2$	Human $\sqsubseteq$ Animal $\wedge$ Biped
sameClassAs	$C_1 \doteq C_2$	Man $\doteq$ Human $\wedge$ Male
subPropertyOf	$P_1 \sqsubseteq P_2$	hasDaughter $\sqsubseteq$ hasChild
samePropertyAs	$P_1 \doteq P_2$	cost $\doteq$ price
sameIndividualAs	$x_1 \doteq x_2$	President_Bush $\doteq$ G_W_Bush
disjointWith	$C_1 \sqsubseteq \neg C_2$	Male $\sqsubseteq \neg$ Female
differentIndividualFrom	$\{x_1\} \sqsubseteq \neg\{x_2\}$	{john} $\sqsubseteq \neg$ {peter}
inverseOf	$P_1 \doteq P_2^-$	hasChild $\doteq$ hasParent $^-$
transitiveProperty	$P^+ \sqsubseteq P$	ancestor $^+$ $\sqsubseteq$ ancestor
uniqueProperty	Thing $\sqsubseteq \leq 1P$	Thing $\sqsubseteq \leq 1$ hasMother
UnambiguousProperty	Thing $\sqsubseteq \leq 1P^-$	Thing $\sqsubseteq \leq 1$ isMotherOf $^-$

 Axioms (mostly) reducible to subClass/PropertyOf

描述逻辑与 OWL 关系

语言构造成份	SHOQ(D)语法	OIL关键字	DAML+OIL关键字	OWL关键字
原子概念	A	class-def	Class	Class
关系	R	slot-def	rdfs:Property	rdfs:Property
概念合取	$C \wedge D$	and	intersectionOf	intersectionOf
概念析取	$C \vee D$	or	unionOf	unionOf
概念取反	$\neg C$	not	complementOf	complementOf
存在约束	$\exists R.C$	domain	hasClass	someValuesFrom
值约束	$\forall R.C$	range	toClass	allValuesFrom
全概念	$\top$	top	Thing	Thing
空概念	$\perp$	Bottom	Nothing	Nothing
最小基数约束	$(\geq n \text{ S.C})$	min-cardinality	minCardinalityQ	minCardinality
最大基数约束	$(\leq n \text{ S.C})$	max-cardinality	maxCardinalityQ	maxCardinality
命名	O	has-Value	hasValue	hasValue
数据类型	D	Value-type	rdf:datatype	rdf:datatype

Table of Contents

知识图谱是什么

知识图谱的表示、存储与查询

RDF 方式

OWL 简介

PG (Property Graph) 方式

RDF 方式 vs. PG 方式

知识图谱的推理

Graph-based Inference Algorithms

Ontology-based inference Algorithms

基于符号表示的推理

基于嵌入学习的推理

知识图谱的构建

知识图谱技术发展



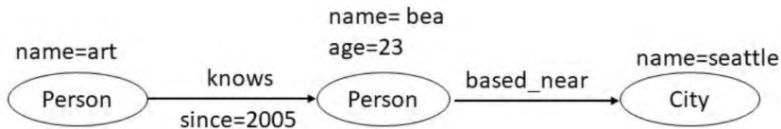
图数据库

图数据库建模的好处：

- ▶ 关系模型中关系被隐含定义，图数据库中关系被显式描述
- ▶ 图数据库可以利用图的结构特征建索引
- ▶ 表达更自然：图是自然的描述事物关系的方式，更加接近于人类对客观事物的记忆方式
- ▶ 易于扩展：图模型更易于适应变化，可以方便的扩展或删减
- ▶ 更善于表达复杂关联：图模型易于表达复杂关联逻辑的查询
- ▶ 多跳优化：图模型在处理多跳查询上，有性能优势

属性图 (Property Graph)

- ▶ 属性图是图数据库 Neo4J 实现的图结构表示模型，在工业界有广泛应用
- ▶ 属性图是由节点 (Node)、关系 (relationship)、属性 (property) 组成的
 - ▶ 节点：含有一个标签 (Label) 和一组属性
 - ▶ 属性：是 key-value 对，key 为 string，value 可以为任意数据类型
 - ▶ 关系：为两个节点之间的有向边，有一个标签，也可以有一组属性



Cypher Query Language

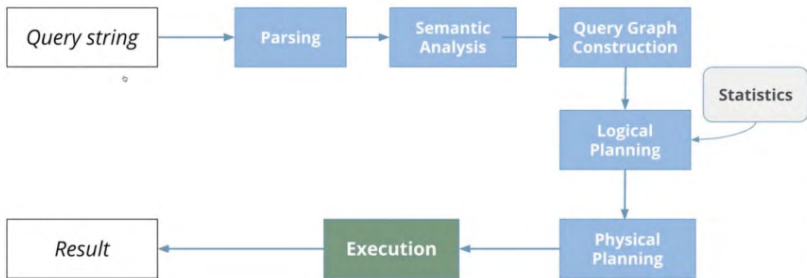
- ▶ Cypher are being considered for adoption into an ISO standard for a graph query language.
- ▶ In addition to querying, Cypher can also be used to create, update and delete data from a graph database.
- ▶ The query consists of two parts:
 - ▶ the **MATCH** clause specifies a graph pattern that should match against the data graph and
 - ▶ the **RETURN** clause specifies what should the query return.
- ▶ The graph pattern is specified in an ASCII notation for graphs: each node is written in parentheses, and each edge is written as an arrow. Both node and relation specifications include their respective types, and any additional properties that should be matched.

```
MATCH (p1:Person {name:art}) -[:knows {since: Y}]-> (p2: Person)
WHERE Y <= 2010
RETURN p2
```



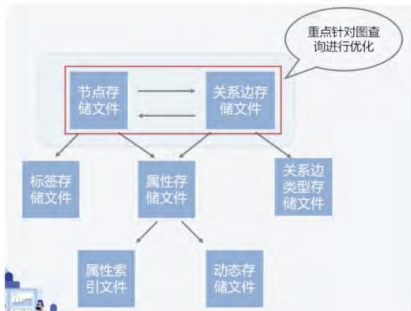
Neo4j Cypher Query Engine

Neo4j Cypher Query Engine



图数据库的实现

- ▶ 采用免索引邻接 (Index-free adjacency) 构建的图数据库引擎: Neo4j
- ▶ 采用免索引邻接的数据库为每一个节点维护了一组指向其相邻节点的引用, 这组引用本质上是相邻节点的微索引 (Micro Index)
- ▶ 这种微索引比起全局索引在处理图遍历查询时非常廉价, 其查询复杂度与数据集整体大小无关, 仅正比于相邻子图的大小



属性数据的存储处理

- ▶ 图数据库中存在大量属性，这些属性的检索与图遍历的计算是分开的，这是为了让节点之间的图遍历能不受大量属性数据的影响
- ▶ 节点和关系的存储记录都包含指向它们的第一个属性 ID 的指针，属性记录也是固定大小，便于之间通过 ID 计算获得存储位置
- ▶ 每个属性记录包含多个属性块，以及属性链中下一个属性的 ID
- ▶ 每个属性记录包含属性类型以及属性索引文件，属性索引文件存储属性名称
- ▶ 对于每一个属性值，记录包含一个指向动态存储记录的指针（大属性值）或内联值（小属性值）



Table of Contents

知识图谱是什么

知识图谱的表示、存储与查询

RDF 方式

OWL 简介

PG (Property Graph) 方式

RDF 方式 vs. PG 方式

知识图谱的推理

Graph-based Inference Algorithms

Ontology-based inference Algorithms

基于符号表示的推理

基于嵌入学习的推理

知识图谱的构建

知识图谱技术发展

RDF 与 Property Graphs

- ▶ RDF 支持更复杂的推理：RDF Schema, Web Ontology (OWL)
- ▶ RDF 与 PG 主要区别：
 - ▶ PG 支持 edge properties, RDF 为了支持 edge properties 需要用其扩展 reification
 - ▶ PG 不需要 IRIs
 - ▶ PG 不支持 blank nodes
- ▶ RDF 与 PG 相似处：两者中的数据可以相互转换
- ▶ 不足：一些现实中关系不能仅由二元关系自然的表达

RDF 图模型和属性图模型的比较

数据模型特性		RDF 图模型	属性图模型
结构	标准化程度	已由 W3C 制定了标准化的语法和语义 ^[23]	尚未形成工业标准
	数学模型	3-均匀有向标签超图	有向标签属性图
	表达力	RDF 图模型强于属性图模型	属性图模型弱于 RDF 图模型
	边属性表达	通过额外方法,如“具体化”	内置支持
	概念层本体定义	RDFS ^[26] 、OWL ^[27]	不支持
	串行化格式	XML ^[32] 、JSON ^[33] 、N-Triples ^[34] 、Turtle ^[35] 等	CSV
操作	查询代数	SPARQL 代数 ^[36]	无
	查询语言	SPARQL ^[36]	Cypher ^[37] 、Gremlin ^[38] 、PGQL ^[39] 、G-CORE ^[30]
约束	约束语言	RDF Shapes 约束语言(SHACL) ^[11]	无

知识图谱查询语言的比较

语法/语义/特性		SPARQL	Cypher	Gremlin	PGQL	G-CORE
图模式匹配查询	语法	CGP	CGP	CGP(无可选) ¹	CGP	CGP
	语义	子图同态、包 ²	无重复边、包 ²	子图同态、包 ²	子图同构 ³ 、包 ²	子图同态、包 ²
导航式查询	语法	RPQ 超集 (增加反向边和属性集上的否定)	RPQ 子集 (*只能作用在单边)	RPQ 超集 (增加通过表达式比较属性值)	RPQ 超集 (增加比较路径上的顶点和边)	RPQ 超集 (增加复杂路径表达式)
	语义	任意路径、集合 ⁴	无重复边 ⁵ 、包 ²	任意路径 ⁶ 、包 ²	最短路径 ⁷ 、包 ⁸	最短路径 ⁹ 、包 ²
分析型查询		聚合函数	聚合函数	聚合函数、PageRank、PeerPressure 聚类	聚合函数	聚合函数
查询可组合性		否	是	是	否	是
数据更新语言 DML		CRUD ¹⁰	CRUD	无	无	CR
数据定义语言 DDL		无	有	无	无	无
实现系统		Jena、RDF4J、gStore、Virtuoso 等	Neo4j、AgensGraph 等	TinkerTop 等	Oracle PGX	无

知识图谱存储的选择

- ▶ 知识图谱存储方式的选择需要综合考虑性能、动态扩展、实施成本等多方面综合因素
- ▶ 区分原生图存储和非原生图存储：原生图存储在复杂关联查询和图计算方面有性能优势，非原生图存储兼容已有工具集通常学习和协调成本会低
- ▶ 区分 RDF 图存储和属性图存储：RDF 存储一般支持推理，属性图存储通常具有更好的图分析性能优势
- ▶ 在大规模处理情况下，需要考虑与底层大数据存储引擎和上层图计算引擎集成需求

关于图模型与图数据库

- ▶ 图模型是更加接近于人脑认知和自然语言的数据模型，图数据库是处理复杂的、半结构化、多维度的、紧密关联数据的最好技术。图数据库技术用于知识图谱项目是当前主流方式
- ▶ 图数据库有它弱处，假如你的应用场景不包含大量的关联查询，对于简单查询，传统关系模型和 NoSQL 数据库目前在性能方面更有优势
- ▶ RDF 作为一个知识图谱表示框架的参考标准，向上对接 OWL 等更丰富的语义表示和推理能力，向下对接简化后的属性图模型以及图计算引擎，是值得重视的知识图谱表示框架

Table of Contents

知识图谱是什么

知识图谱的表示、存储与查询

RDF 方式

OWL 简介

PG (Property Graph) 方式

RDF 方式 vs. PG 方式

知识图谱的推理

Graph-based Inference Algorithms

Ontology-based inference Algorithms

基于符号表示的推理

基于嵌入学习的推理

知识图谱的构建

知识图谱技术发展



- ▶ Knowledge Graph Retrieval
 - ▶ Query languages: SPARQL and Cypher
- ▶ Knowledge Graph Inference
 - ▶ Drawing conclusions that are not explicit in the knowledge graph
- ▶ Knowledge Graph Inference
 - ▶ **Graph-based inference** algorithms: The graph algorithms are applicable to any graph-structured data, and support operations such as finding minimum paths between nodes in a graph, identifying salient nodes in a graph, etc.
 - ▶ **Ontology-based inference** algorithms: The ontology-based algorithms operate on the structure of the graph but take its semantics into account, for example, traversing specific paths, or concluding new connections based on background domain knowledge.

Table of Contents

知识图谱是什么

知识图谱的表示、存储与查询

RDF 方式

OWL 简介

PG (Property Graph) 方式

RDF 方式 vs. PG 方式

知识图谱的推理

Graph-based Inference Algorithms

Ontology-based inference Algorithms

基于符号表示的推理

基于嵌入学习的推理

知识图谱的构建

知识图谱技术发展



Graph-based Inference Algorithms

- ▶ **Path finding** involves finding a path between two or more nodes in a graph that satisfies certain properties.
- ▶ **Centrality detection** is about understanding which nodes are important in a graph. Different methods used to define the meaning of importance lead to many variations in the centrality detection algorithms.
- ▶ **Community detection** is about identifying a group of nodes in a graph that satisfies some criteria of being in a community. Community detection is useful for studying emergent behaviors in graphs that may otherwise not be noticed.

Table of Contents

知识图谱是什么

知识图谱的表示、存储与查询

RDF 方式

OWL 简介

PG (Property Graph) 方式

RDF 方式 vs. PG 方式

知识图谱的推理

Graph-based Inference Algorithms

Ontology-based inference Algorithms

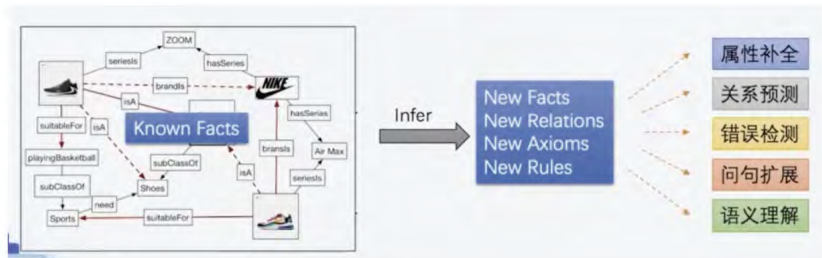
基于符号表示的推理

基于嵌入学习的推理

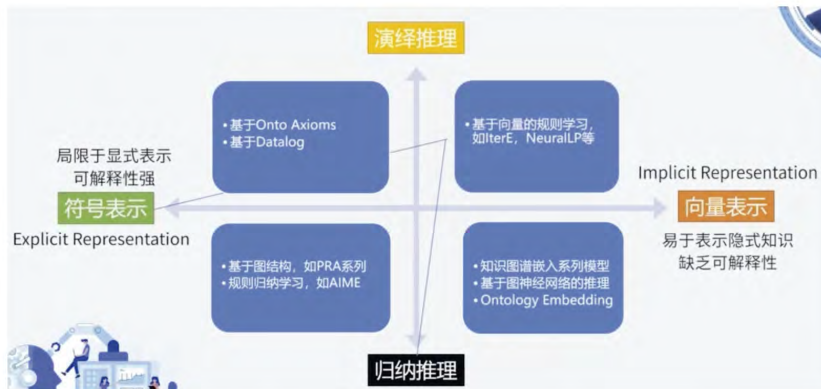
知识图谱的构建

知识图谱技术发展

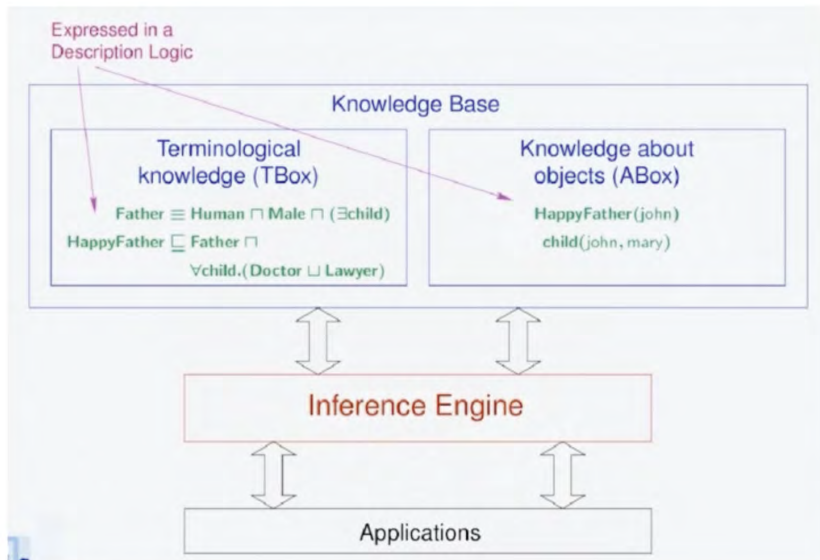
在知识图谱上实现推理



知识图谱推理方法分类

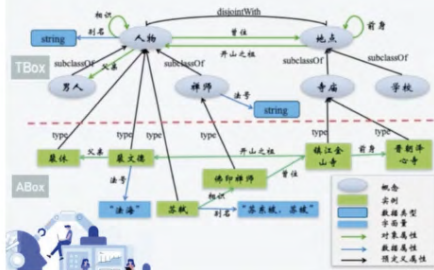


Reasoning in Description Logic



Taxonomic Inference

TBox vs ABox



● Tbox语言

➢ 定义: 引入概念以及关系的名称

● 例如, **Mother, Person, has_child**

➢ 包含: 声明包含关系的公理

● 例如, **Mother $\sqsubseteq$ has_child. Person**

概念和概念、关系和概念之间可以组成更复杂的概念。

● Abox语言

➢ 概念断言——表示一个对象是否属于某个概念

● 例如, **Mother(Alice), Person(Bob)**

➢ 关系断言——表示两个对象是否满足特定关系

● 例如, **has_child(Alice, Bob)**

Tableaux 算法

- ▶ OWL 本体上实现的各种推理都可以用 Tableaux 算法来实现
- ▶ Tableaux 算法的基本思想是通过一系列规则构建 Abox, 以检测知识库的可满足性
- ▶ Tableaux 算法将概念包含、实例检测等推理都转化为可满足性检测问题来实现
- ▶ Tableaux 算法检查可满足性的基本思想类似于一阶逻辑的归结反驳

Tableaux 算法

● Tableaux 运算规则 (以主要DL算子举例)

初始情况下, $\emptyset$ 是原始的Abox, 迭代运用如下规则:

$\sqcap^+$ -规则: 若 $C \sqcap D(x) \in \emptyset$, 且 $C(x), D(x) \notin \emptyset$, 则 $\emptyset := \emptyset \cup \{C(x), D(x)\}$;

$\sqcap^-$ -规则: 若 $C(x), D(x) \in \emptyset$, 且 $C \sqcap D(x) \notin \emptyset$, 则 $\emptyset := \emptyset \cup \{C \sqcap D(x)\}$;

$\exists$ -规则: 若 $\exists R.C(x) \in \emptyset$, 且 $R(x,y), C(y) \notin \emptyset$, 则 $\emptyset := \emptyset \cup \{R(x,y), C(y)\}$,

其中, y 是新加进来的个体;

$\forall$ -规则: 若 $\forall R.C(x), R(x,y) \in \emptyset$, 且 $C(y) \notin \emptyset$, 则 $\emptyset := \emptyset \cup \{C(y)\}$;

$\sqsubseteq$ -规则: 若 $C(x) \in \emptyset, C \sqsubseteq D$, 且 $D(x) \notin \emptyset$, 则 $\emptyset := \emptyset \cup \{D(x)\}$;

$\perp$ -规则: 若 $\perp(x) \in \emptyset$, 则拒绝 $\emptyset$;

Tableaux 算法

● Tableaux 运算规则举例

给定如下本体，检测是否可满足？

$\text{Man} \sqcap \text{Women} \sqsubseteq \perp$
 $\text{Man}(\text{Allen})$
 $\text{Woman}(\text{Allen})$

如果 $\text{Woman}(\text{Allen})$ 在初始情况已存在于原始本体，那么推导出该本体不可满足！

Allen 不在 Woman 中



OWL 本体推理工具

工具名称	支持本体语言	编程语言	算法
FaCT++	OWL DL	C++	tableau-based
Racer	OWL DL	Common Lisp	tableau-based
Pellet	OWL DL	Java	tableau-based
HermiT	OWL 2 Profiles	Java	hypertableau

Rule-based Inference

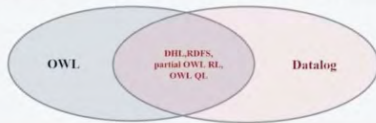
基于 Datalog 的符号推理

本体推理的局限:

- 主要实现了基于本体概念描述的推理,无法支持规则型知识的推理
- 用户无法定义自己的推理过程

引入规则推理

- 可以根据特定的场景定制规则,以实现用户自定义的推理过程
- Datalog语言可以结合本体推理和规则推理
- Datalog是面向知识库和数据库设计的逻辑语言,表达能力与OWL相当,支持递归,同时便于撰写规则,实现规则推理



Datalog 推理举例

● Datalog 推理举例

规则集

$\text{path}(x, y) :- \text{edge}(x, y).$

$\text{path}(x, y) :- \text{path}(x, z), \text{path}(z, y).$

事实集

$\text{edge}(a, b).$

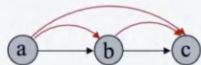
$\text{edge}(b, c).$

结果集

$\text{path}(a, b).$

$\text{path}(b, c).$

$\text{path}(a, c).$



→ edge

→ path

Datalog 推理工具举例

工具名称	支持本体语言	实现编程语言	支持编程语言
KAON2	OWL DL/SWRL	Java	Java
RDFox	OWL 2 RL	C++	Java/C++/Python

知识图谱嵌入 (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
- ▶ 将推理转换为向量或矩阵之间的计算过程

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



知识图谱嵌入模型：预测问题与推理评价



关系推理的评价指标

- Hit@n: 所有预测样本中排名在n以内的比例, n常用的取值为1, 3, 10
- MR: Mean Rank 所有预测样本的平均排名
- MRR: Mean Reciprocal Rank 先对所有预测样本的排名求倒数, 然后求平均

知识图谱嵌入推理模型

类别	方法	核心创新点
知识图谱嵌入	DURA	在基于张量分解的KGC模型中，通过增加对偶的基于距离的KGC模型作为正则项显著克服了模型过拟合的问题。
	Interstellar	提出了一种递归网络结构在路径信息中搜索长距离和短距离的信息提升嵌入表示学习结果。
	QuatE	在超复数空间中进行知识图谱嵌入表示学习。
	Platt/Iso	结合普拉特定标和等渗回归方法，解决了KGE方法中负样本不可用情景下的预测概率校准问题。
	RotatE	在复数空间中用旋转来模拟实体之间存在的关系
	TransRHS	将关系层次结构融入到KGE中
	wRAN	利用对抗学习实现低资源知识图谱推理
	MetaR	利用元学习实现少样本知识图谱推理
规则学习	AnyBURL	基于路径的规则离散的规则挖掘方法
	DRUM	通过将规则置信度和低秩张量联系完成可微规则学习。
	Neural-Num-LP	在可微规则学习框架下进行数值类规则的学习。
	IterE	基于迭代学习同时学习规则和Embedding
多跳推理	RLH	提出了层次化的强化学习用于知识图谱多跳推理
	DPMPN	提出了基于神经网络的知识图谱推理方法，不仅可以生成基于子图的解释，还可以在图上完成信息传播。
	BetaE	提出了一种用Beta分布对知识图谱中实体和逻辑查询进行建模的概率嵌入方法，完成复杂的多跳逻辑推理。
可微查询	DPMPN	将知识图谱查询语句表示为向量空间中的一个盒子，能更加高效地完成知识图谱逻辑查询。

符号逻辑与表示学习的融合

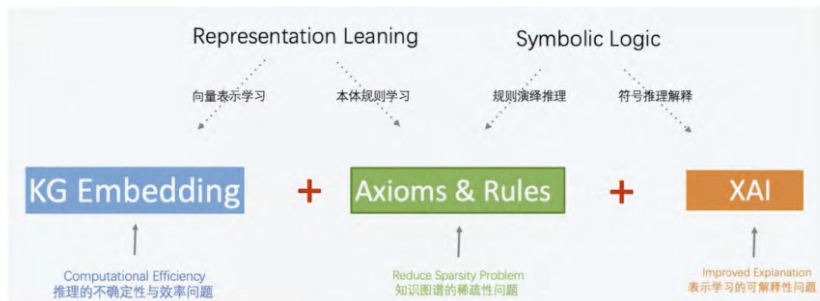


Table of Contents

知识图谱是什么

知识图谱的表示、存储与查询

RDF 方式

OWL 简介

PG (Property Graph) 方式

RDF 方式 vs. PG 方式

知识图谱的推理

Graph-based Inference Algorithms

Ontology-based inference Algorithms

基于符号表示的推理

基于嵌入学习的推理

知识图谱的构建

知识图谱技术发展

知识图谱工程



知识图谱工程

从不同来源、不同结构的数据中进行知识提取，形成知识存入到知识图谱



Creating a Knowledge Graph

- ▶ **冷启动**: When we are working with **structured and semi structured data sources**, we have to perform **schema mapping task** (i.e., relating the schema in the input source with the schema of the knowledge graph) and **record linkage task** (i.e., relating new instances with the pre-existing instances in the knowledge graph).
- ▶ **多来源自动生成和扩展**: When we are working with the **unstructured sources**, we have to solve the **information extraction problems of entity extraction and relation extraction**.
- ▶ An RDF model needs to adopt a **scheme for IRIs** which is not necessary for property graphs. In a property graph model, we need to decide whether a value should be represented as a **property or as a node**, while this distinction is unnecessary in an RDF model.

Design of an RDF Graph

Linked Data Principles:

1. Use URIs as names for things.
 - ▶ Identify the items of interest, referred to as resources
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL).
 - ▶ Dereferencing of a non-information object returns RDF data
4. Include links to other URIs, so that they can discover more things.

Design of a Property Graph

- ▶ Choosing nodes, labels and properties
 - ▶ Nodes usually represent entities in a domain
 - ▶ Property should not change with time
- ▶ Introduce relationships
 - ▶ When efficient access is required
- ▶ Introduce relationship properties
 - ▶ Time varying relationships
- ▶ Handle non-binary relationships
 - ▶ 1. Create an object representing the relationship; 2. Create objects for each argument of the relationship; 3. Introduce relationships to connect them.

知识抽取

从文本、视觉等不同来源不同结构的数据中进行知识抽取，现在用多模态大模型极大的简化了这个过程

- ▶ 实体识别：基于模版和规则，基于序列标注的方法，隐马尔科夫模型 (HMM)，条件随机场 (CRF)，深度学习方法 (BiLSTM + CRF)，基于预训练语言模型
- ▶ 实体关系抽取：基于模版的方法 (触发词匹配，依存句法匹配)，基于监督学习的方法 (SVM, CNN, RNN, BiLSTM, GNN)，基于预训练语言模型，半监督学习 (远程监督，强化学习)
- ▶ 属性补全：抽取式属性补全、生成式属性补全
- ▶ 概念抽取 (实体的抽象为概念, isA)：基于模版抽取，基于百科抽取 (从半结构化数据中获取)，基于机器学习
- ▶ 事件识别与抽取 (什么人/组织，什么时间，在什么地方，做了什么事)：事件发现和分类，事件要素抽取。模式匹配方法，机器学习方法，弱监督



Table of Contents

知识图谱是什么

知识图谱的表示、存储与查询

RDF 方式

OWL 简介

PG (Property Graph) 方式

RDF 方式 vs. PG 方式

知识图谱的推理

Graph-based Inference Algorithms

Ontology-based inference Algorithms

基于符号表示的推理

基于嵌入学习的推理

知识图谱的构建

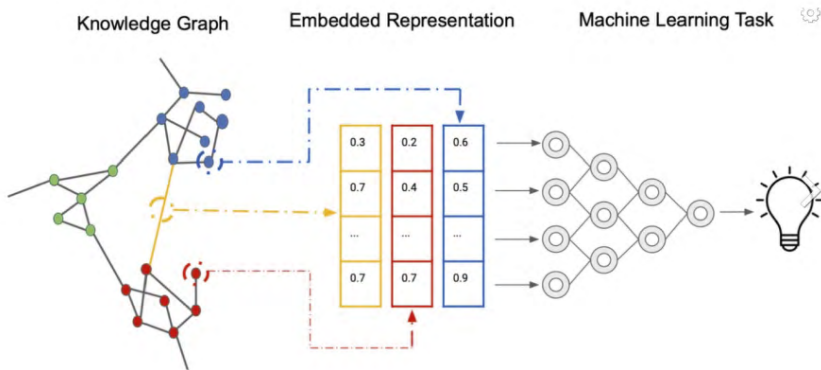
知识图谱技术发展



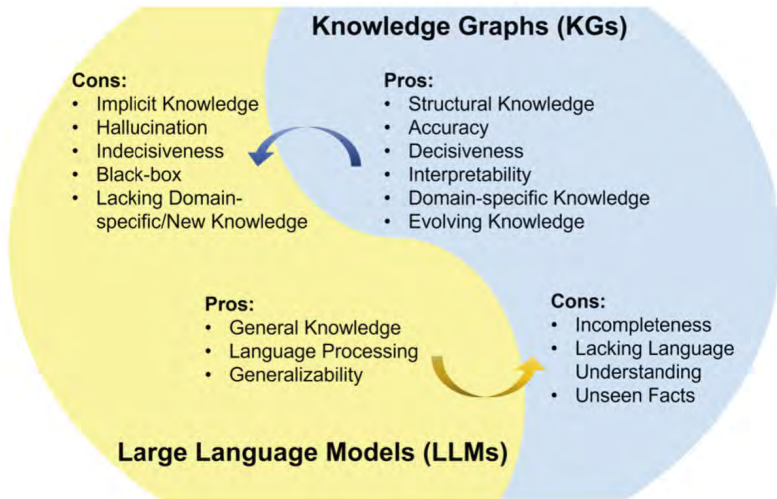
知识图谱技术发展

- ▶ 多模态知识图谱 (IMGpedia、MMKG、ImageGraph、Richpedia): 模态知识互补、模态实体消歧、模态语义搜索、利用多模态数据补全知识图谱、利用知识图谱增强多模态任务
- ▶ 事理知识图谱 (Event Logic Knowledge Graph): 事理逻辑知识库, 描述事件之间的演化规律和模式, 回答 “how” 和 “why” 的问题
- ▶ Knowledge Graph Embedding: 推理和知识的向量化, Knowledge representation learning
- ▶ 知识图谱与大模型:
 - ▶ KG-enhanced LLMs
 - ▶ LLM-augmented KGs
 - ▶ Synergized LLMs + KGs

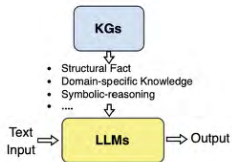
Knowledge Graph Embedding



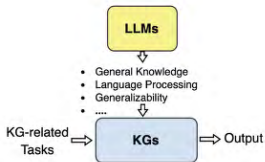
知识图谱与大模型



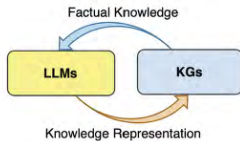
知识图谱与大模型



a. KG-enhanced LLMs

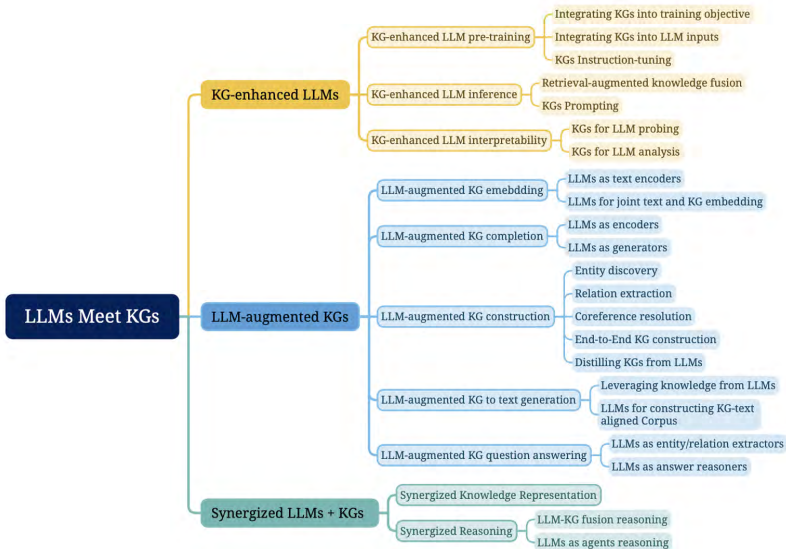


b. LLM-augmented KGs



c. Synergized LLMs + KGs

知识图谱与大模型



荐

