

资源描述框架对比

RDF | 知识图谱的基石

资源描述框架(Resource Description Framework), 是一个数据模型, 提供了一个统一的标准用于描述实体/资源

- 表达能力有限
 - 无法区分类和对象
 - 无法定义和描述类的关系/属性
 - RDF是对具体事物的描述, 缺乏抽象能力, 无法对同一个类别的事物进行定义和描述

序列化方法

- RDF/XML: 用XML的格式来表示RDF数据
- **N-Triples**: 用多个三元组来表示RDF数据集, 是最直观表示方法。在文件中, 每一行表示一个三元组, 方便机器解析和处理。开放领域知识图谱DBpedia通常是用这种格式来发布数据的。
- **Turtle**: 使用得最多的一种RDF序列化方式。比RDF/XML紧凑, 且可读性比N-Triples好。
- RDFa: “The Resource Description Framework in Attributes”, 是HTML5的一个扩展, 在不改变任何显示效果的情况下, 让网站构建者能够在页面中标记实体, 像人物、地点、时间、评论等等
- JSON-LD: “JSON for Linking Data”, 用键值对的方式来存储RDF数据

N-Triples实例

```
<http://www.kg.com/person/1> <http://www.kg.com/ontology/chineseName> "罗纳尔多·
路易斯·纳萨里奥·德·利马"^^string.
<http://www.kg.com/person/1> <http://www.kg.com/ontology/career> "足球运动
员"^^string.
<http://www.kg.com/person/1> <http://www.kg.com/ontology/fullName> "Ronaldo
Luís Nazário de Lima"^^string.
<http://www.kg.com/person/1> <http://www.kg.com/ontology/birthDate> "1976-09-
18"^^date.
<http://www.kg.com/person/1> <http://www.kg.com/ontology/height> "180"^^int.
<http://www.kg.com/person/1> <http://www.kg.com/ontology/weight> "98"^^int.
<http://www.kg.com/person/1> <http://www.kg.com/ontology/nationality> "巴
西"^^string.
<http://www.kg.com/person/1> <http://www.kg.com/ontology/hasBirthPlace>
<http://www.kg.com/place/10086>.
<http://www.kg.com/place/10086> <http://www.kg.com/ontology/address> "里约热内
卢"^^string.
<http://www.kg.com/place/10086> <http://www.kg.com/ontology/coordinate>
"-22.908333, -43.196389"^^string.
```

Turtle实例

```
@prefix person: <http://www.kg.com/person/> .
@prefix place: <http://www.kg.com/place/> .
@prefix : <http://www.kg.com/ontology/> .

person:1 :chineseName "罗纳尔多·路易斯·纳萨里奥·德·利马"^^string.
person:1 :career "足球运动员"^^string.
person:1 :fullName "Ronaldo Luís Nazário de Lima"^^string.
person:1 :birthDate "1976-09-18"^^date.
person:1 :height "180"^^int.
person:1 :weight "98"^^int.
person:1 :nationality "巴西"^^string.
person:1 :hasBirthPlace place:10086.
place:10086 :address "里约热内卢"^^string.
place:10086 :coordinate "-22.908333, -43.196389"^^string.
```

RDFS

“Resource Description Framework Schema”，是最基础的模式语言。

- 在概念、抽象层面对RDF数据进行定义
- 定义类，及每个类包含的属性

常用词汇

常用词汇	描述
rdfs:Class	定义类
rdfs:domain	表示该属性属于哪个类别
rdfs:range	描述该属性的取值类型
rdfs:subClassOf	描述该类的父类
rdfs:subProperty	描述该属性的父属性

RDFS实例

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix : <http://www.kg.com/ontology/> .

### 这里我们用词汇rdfs:Class定义了“人”和“地点”这两个类。
:Person rdf:type rdfs:Class.
:Place rdf:type rdfs:Class.
```

rdfs当中不区分数据属性和对象属性，词汇rdf:Property定义了属性，即RDF的“边”。

```
:chineseName rdf:type rdf:Property;
    rdfs:domain :Person;
    rdfs:range xsd:string .

:career rdf:type rdf:Property;
    rdfs:domain :Person;
    rdfs:range xsd:string .

:fullName rdf:type rdf:Property;
    rdfs:domain :Person;
    rdfs:range xsd:string .

:birthDate rdf:type rdf:Property;
    rdfs:domain :Person;
    rdfs:range xsd:date .

:height rdf:type rdf:Property;
    rdfs:domain :Person;
    rdfs:range xsd:int .

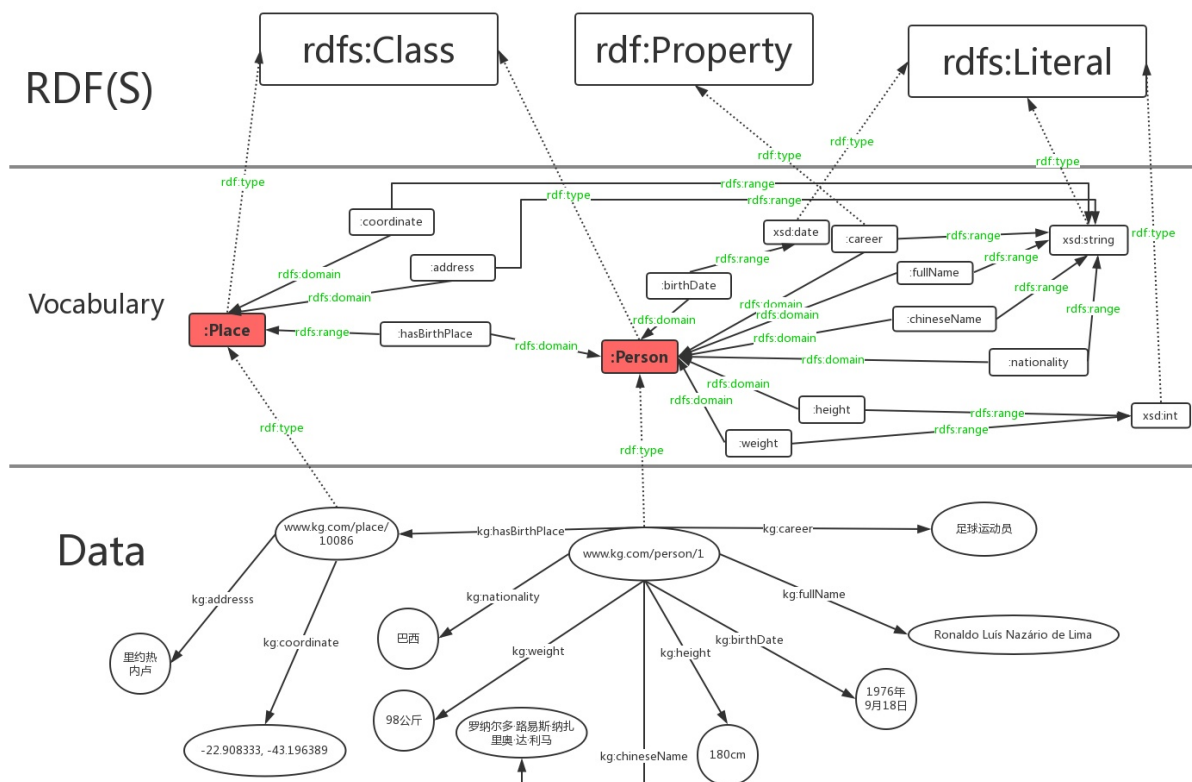
:weight rdf:type rdf:Property;
    rdfs:domain :Person;
    rdfs:range xsd:int .

:nationality rdf:type rdf:Property;
    rdfs:domain :Person;
    rdfs:range xsd:string .

:hasBirthPlace rdf:type rdf:Property;
    rdfs:domain :Person;
    rdfs:range :Place .

:address rdf:type rdf:Property;
    rdfs:domain :Place;
    rdfs:range xsd:string .

:coordinate rdf:type rdf:Property;
    rdfs:domain :Place;
    rdfs:range xsd:string .
```



自底向上：

- **Data层**：对我们研究的图谱的具体描述
- **Vocabulary层**：自定义的词汇（类别，属性）
- **RDFS层**：预定义词汇

OWL

“Web Ontology Language”

- 提供快速、灵活的数据建模能力
- 高效的自动推理

常用词汇

属性

常用词汇	描述
<code>owl:DatatypeProperty</code>	定义数据属性
<code>owl:ObjectProperty</code>	定义对象属性

属性特征

常用词汇	描述
owl:TransitiveProperty	表示该属性具有 传递性 。 例如，我们定义“位于”是具有传递性的属性，若A位于B，B位于C，那么A肯定位于C
owl:SymmetricProperty	表示该属性具有 对称性 。 例如，我们定义“认识”是具有对称性的属性，若A认识B，那么B肯定认识A
owl:FunctionalProperty	表示该属性取值的 唯一性 。 例如，我们定义“母亲”是具有唯一性的属性，若A的母亲是B，在其他地方我们得知A的母亲是C，那么B和C指的是同一个人
owl:inverseOf	定义某个属性的 相反关系 。 例如，定义“父母”的相反关系是“子女”，若A是B的父母，那么B肯定是A的子女

本体映射

常用词汇	描述
owl:equivalentClass	表示某个类和另一个类是相同的
owl:equivalentProperty	表示某个属性和另一个属性是相同的
owl:sameAs	表示两个实体是同一个实体

OWL实例

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix : <http://www.kg.com/ontology/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .

### 这里我们用词汇owl:Class定义了“人”和“地点”这两个类。
:Person rdf:type owl:Class.
:Place rdf:type owl:Class.

### owl区分数据属性和对象属性（对象属性表示实体和实体之间的关系）。词汇
owl:DatatypeProperty定义了数据属性，owl:ObjectProperty定义了对对象属性。
:chineseName rdf:type owl:DatatypeProperty;
    rdfs:domain :Person;
    rdfs:range xsd:string .

:career rdf:type owl:DatatypeProperty;
    rdfs:domain :Person;
    rdfs:range xsd:string .
```

```
:fullName rdf:type owl:DatatypeProperty;
    rdfs:domain :Person;
    rdfs:range xsd:string .

:birthDate rdf:type owl:DatatypeProperty;
    rdfs:domain :Person;
    rdfs:range xsd:date .

:height rdf:type owl:DatatypeProperty;
    rdfs:domain :Person;
    rdfs:range xsd:int .

:weight rdf:type owl:DatatypeProperty;
    rdfs:domain :Person;
    rdfs:range xsd:int .

:nationality rdf:type owl:DatatypeProperty;
    rdfs:domain :Person;
    rdfs:range xsd:string .

:hasBirthPlace rdf:type owl:ObjectProperty;
    rdfs:domain :Person;
    rdfs:range :Place .

:address rdf:type owl:DatatypeProperty;
    rdfs:domain :Place;
    rdfs:range xsd:string .

:coordinate rdf:type owl:DatatypeProperty;
    rdfs:domain :Place;
    rdfs:range xsd:string .
```

