如何生成 RDF 数据?

作者：xjtushilei
原文链接：https://ld246.com/article/1507896459396
来源网站：链滴

背景

知识图谱越来越火，技术人员开始习惯使用Sparql查询RDF数据。但是，我们如何构建一个RDF数据？

因为之前互联网上没有相关工具介绍的文章，所以，这里，我对常见的两种RDF构造工具做一个介绍。

相关知识

RDF

“资源描述框架（Resource Description Framework，RDF）”成为 W3C 推荐标准，与 XML 和 SAP 等 Web 标准并排。RDF 可以应用于处理特殊输入数据（如 CRM）的领域，已经广泛用于社会网和自助出版软件（如 LiveJournal 和 TypePad）。

关于RDF推荐去阮一峰的资源描述框架RDF 进行了解，因为其介绍的浅显易懂。

Sparql

比如说我们查询 mysql， 使用的是sql语言，那么我们查询RDf数据，使用的就是SPARQL。

学习这个语言的话，推荐直接去官方学习：https://www.w3.org/TR/rdf-sparql-query/

介绍列表

- Jena

Java 语言编写的，具有推理功能，适合开发和技术研究使用。

- rdflib

Python 语言编写的，但是功能简单，有基本的查询和构建功能。

Jena

Java 程序员将越来越多地得益于具有使用 RDF 模型的技能。在本文中，我将带您体验惠普实验室的放源代码 Jena Semantic Web Framework（请参阅 官网）的一些功能。您将了解如何创建和填充 RF 模型，如何将它们持久存储到数据库中，以及如何使用 RDQL 查询语言以程序方式查询这些模型。后，我将说明如何使用 Jena 的推理能力从本体推断模型知识。

对于专业人员来说，大家直接阅读官网就行，但是新手对官网很迷茫，不知道怎么看，所以将推荐看东西说一下。

1. 简单的RDF介绍和Sparql查询相结合： http://jena.apache.org/tutorials/rdf_api.html

2. https://github.com/apache/jena/tree/master/jena-core/src-examples 这里有十个例子，结博客把它看懂，基本操作就基本没问题了。

写triple

下面的函数分为SPO中O是否是Node节点来区分。在注释中举了两个例子。并且在主函数中给出了个测试方式。

```
import org.apache.jena.rdf.model.Model;
import org.apache.jena.rdf.model.ModelFactory;
import org.apache.jena.rdf.model.Property;
import org.apache.jena.rdf.model.RDFNode;
import org.apache.jena.rdf.model.Resource;

import config.Config;

public class WriteRdf {

    public static void main(String[] args) {
        // 函数示例
        Model model = ModelFactory.createDefaultModel();
        writeAnTripleWhenObjectIsNode(model, "数组", "分面", "二维数组");
        model.write(System.out, "N3");
    }

    /**
     * 例子:  <周杰伦> <年龄> '35'
     */
    public static void writeAnTripleWhenObjectIsNotNode(Model model, String s, String p, Stri
g o) {

        Property property = model.createProperty(p);
        model.createResource(s).addProperty(property, o);
    }

    /**
     * 例子:  <周杰伦> <妻子> <昆凌>
     */
    public static void writeAnTripleWhenObjectIsNode(Model model, String s, String p, String o
 {
        Resource resource = model.createResource(s);
        Property property = model.createProperty(p);
        RDFNode object = model.createResource(o);
        model.add(resource, property, object);
    }

}
```

导入RDF数据到Jena中

RDF数据格式有很多，但是jena内部的数据是以TDB形式存在的，需要把RDf这些文本文件转化为jen自己的模型。这里给出三种方法：

```
    public static void save1() {
        String directory = "C:\\RDFdata\\Database1\\";
        Dataset ds = TDBFactory.createDataset(directory);
        Model model = ds.getDefaultModel();//这里使用TDB的默认Model
```

```
        String source = "C:\\RDFdata\\source\\drugs.nq";
        TDBLoader.loadModel(model, source);
        ds.end();
        System.out.println("done");
    }

    public static void save2() {
        String directory = "C:\\RDFdata\\Database2\\";
        Dataset ds = TDBFactory.createDataset(directory);
        Model model = ds.getDefaultModel();//这里使用TDB的默认Model
        FileManager.get().readModel(model, "C:\\RDFdata\\source\\drugs.nq");
        ds.end();
        System.out.println("done");
    }

    public static void save3()  {
        Dataset ds = TDBFactory.createDataset("D:\\RDFdata\\1\\");
        String filename = "D://rdf.nt2";
        RDFDataMgr.read(ds, filename,Lang.N3);
        ds.close();
        System.out.println(" done");
    }
```

查询

首先获取到model

```
        Dataset ds = TDBFactory.createDataset("D:\\pdd\\");
        Model model = ds.getDefaultModel();
```

然后编写sparql语句，再进行查询

```
            String query111=
            "select (count(*) as ?num)" +
                    " Where"+
                    "{"+
                       "?s <http://kmap.xjtudlc.com/pdd_data/property/diagnoses_icd9>  ?o. "+
                    "}";
        QueryExecution qexec = QueryExecutionFactory.create(query111, model);
        System.out.println(qexec.getQuery().toString());
        ResultSet resultSet = qexec.execSelect();

        String rerult = ResultSetFormatter.asText(resultSet);
        System.out.println(rerult);
```

输出形式有很多，也可以遍历进行输出，并输出自己想要的结果。

比如，我们想获取某一个属性o，

```
          ResultSet resultSet = qexec.execSelect();
        String o="";
        List<QuerySolution> rerult2 = ResultSetFormatter.toList(resultSet);
```

```
System.out.println(rerult2.size());
for (QuerySolution querySolution : rerult2) {
    o=querySolution.get("o").toString();
    System.out.println(o);
}
```

其他

除了这些，jena的特斯就是进行推理，可以将查询图进行分析。这里我用的不多，就不进行展示了。

rdflib

RDFLib工作处理RDF是一个纯Python包。

RDFLib包含大多数处理RDF的东西,包括:

1. 解析和序列化 RDF/XML, N3, NTriples, N-Quads, Turtle, TriX, RDFa and Microdata.
2. 一个图模型
3. 存储模型.
4. SPARQL 1.1 实现 - 支持 SPARQL 1.1 查询和更新数据.

官方网站: https://rdflib.readthedocs.io

导入文件

```
# 导入nt文件
from rdflib import Graph

g = Graph()

'''
format: 'rdf/xml' 'xml', 'n3', 'nt', 'trix', 'rdfa'
'''
g.parse("icd10.nt", format="n3")

# 打印图的大小
print(len(g))  # prints 2

# 遍历所有三元组
import pprint

for stmt in g:
    pprint.pprint(stmt)
```

支持的文件格式有：’ rdf/xml’ ‘xml’ , ‘n3’ , ‘nt’ , ‘trix’ , ‘rdfa’

创建RDF文件

增加一个节点，并以turtle形式序列化输出

```
from rdflib import Graph
g = Graph()

g.add( (bob, RDF.type, FOAF.Person) )
g.add( (bob, FOAF.name, name) )
g.add( (bob, FOAF.knows, linda) )
g.add( (linda, RDF.type, FOAF.Person) )
g.add( (linda, FOAF.name, Literal('Linda') ) )

print g.serialize(format='turtle')
```

Sparql查询

```
# 利用SPARQL进行查询
qres = g.query("""
    SELECT *
    WHERE {
      ?subject ?predicate  ?object
    }
    LImit 10

""")

for row in qres:
    print(row)
```

对比

1. python版本的rdflib包 更简洁，语法简单，更容易理解
2. rdflib 的官方文档更加简洁，符合新手的认知能力，建议初学者使用python来进行最好!
3. Jena 文档首先需要吐槽，很不友好。
4. Jena的持久化做的比较好，在查询方式，序列化方式等方面做的更加专业。
5. Jena 提供了持久化的事务能力。
6. Jena提供了远程调用方案，可以解决数据共享问题。我们可以开放一个a SPARQL end-point 让其人通过http连连接，就像DBpedia提供的查询接口一样。
7. 总体来说，jena更专业一点，rdflib对新用户更友好，功能也够用。

PDD 与 我

我作为核心成员，发布了一个医疗的数据集PDD，在 http://kmap.xjtudlc.com/pdd/ 同时，我们最在整理最新的ICD10的数据集，与湘雅医院、腾讯的移动互联事业群（微信）在合作，准备利用知识谱进行疾病诊断与药物推荐等方面工作。