Projektdokumentation im Modul Semantic Web

Anzahl der Publikationen pro Mitarbiter der HTWK-Leipzig

Marcel Kirbst

30. Juni 2014

Recherchefragestellung: Eine Liste aller Angestellten der HTWK-Leipzig, geordnet nach der Anzahl der bisherigen Publikationen im deutschsprachigen Raum.

1 Inhaltliche Interpretation der Fragestellung

Neben der Lehre stellt die Forschung und die Publikation deren Resultate eine wichtige Kernaufgabe der Professoren und wissenschaftlichen Mitarbeiter an Hochschulen dar. Die Aufgabe dieser Projektarbeit besteht daher zum einen darin, alle Mitarbeiter der HTWK-Leipzig zu ermitteln und in einer semantischen Datenbank zu erfassen.

Weiterhin sind Daten aller im deutschsprachigen Raum erschienen Publikationen aufzubereiten und einzubinden. Diese Daten stellt die Deutsche Nationalbibliothek kostenfrei zur Verfügung.

Die Umsetzung des Projektes erfolgt auf einer virtuellen Maschine unter Verwendung der Virtualisierungslösung VirtualBox der Firma Oracle. Auf der virtuellen Maschine wurde als Gastbetriebssystem die Linux-Distribution Kubuntu 14.04 von Canonical eingesetzt.

2 Relevante Datenquellen

An dieser Stelle erfolgt eine Auflistung aller relevanten Datenquellen und deren Beschreibung.

2.1 Mitarbeiterkatalog der HTWK-Leipzig

Die aktuellste, öffentlich zugängliche Auflistung aller Mitarbeiter der HTWK-Leipzig besteht aus dem HTWK-Telefonverzeichnis. Diese HTML-Seite lässt sich unter der Webadresse [1] abrufen. Eine aufbereitete Version dieser Daten im JSON-Format wird von Herrn Henri Knochenhauer, B.Sc. und Herrn Roy Meisser, B.Sc zur Verfügung gestellt und in diesem Projekt als Datenquelle genutzt.

Link	http://141.57.21.45:8080/info/staff
Datenformat	JSON
Schnittstelle	HTTP Rest-API
Lizenz	Daten: unbekannt. Datenquelle: GPL.

Die Daten dieser Datenquelle umfassen die eindeutige Mitarbeiteridentifikationsnummer, Nachname, Vornamen, akademischen Grad sowie die Fakultät jedes Mitarbeiters.

2.2 Bibliothekskatalog der Deutsch Nationalbibliothek

Die Deutsche Nationalbibliothek (DNB) sammelt alle deutschen Publikationen seit dem Jahr 1913. Darüber hinaus bietet sie umfangreiche Dienstleistungen für Bibliotheken und Wissenschaftler an. Mittels einer durch die DNB vergebenen Normdatensatz (GND)[2] lassen sich Werke, Personen oder Körperschaften identifizieren. Diese Identifizierung ermöglicht es Personen unterschiedlicher Quellen zu verlinken.

Link	http://datendienst.dnb.de/cgi-bin/mabit.pl? userID=opendata&pass=opendata&cmd=login
Datenformat	RDF
Schnittstelle	Linked Data, Dump, Rest-API
Lizenz	CC-SA

Die Daten lassen sich unter der angegebenen Webadresse direkt herunterladen. Für dieses Projekt wurde die Datei GND.rdf.gz herunter geladen, welche die gesammelten Normdaten zu den Autoren enthält und entpackt derzeit etwa 9 GByte groß ist Weiterhin wurde

[1] http://www.htwk-leipzig.de/de/hochschule/telefonverzeichnis/
[2] Gemeinsame Normdatei

die Datei DNBTitel.rdf.gz heruntergeladen. Diese Datei enthält die Namen und Autoren aller Werke der Deutschen Nationalbibliothek und ist entpackt derzeit etwa 12 GByte groß.

3 Extraktion relevanter Daten und import in einen Triplestore

Die Extraktion aller 3 Dateien erfolgt über eine für dieses Projekt erstellte JAVA-Anwendung, welche im Repository zu diesem Projekt hinterlegt ist. Das Repository kann unter folgender Webadresse abgerufen werden[3].

3.1 Extraktion der HTWK-Mitarbeiter

Die Daten der HTWK-Mitarbeiter liegen im JSON Format vor. Die exakte Struktur zeigt das folgende Listing auszugsweise.

```
[
  - {
      cuid: "597",
      name: "Siebeck, Andrea",
      degree: "Dipl.-Angl.",
      faculty: "AAA"
   },
  - {
  {
      cuid: "8",
      name: "Engel, Heike",
      degree: "Dipl.-Wirtschaftsinf.",
      faculty: "DF"
  }
]
```

Listing 1: Format der importierten HTWK-Mitarbeiterdaten

Die JAVA-Anwendung bezieht das JSON-Dokument direkt von der Webressource. Die JAVA-Anwendung generiert aus den importierten Daten eine RDF-Datei, welche nach folgendem RDF-Schema generiert wird.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
```

[3] https://github.com/mkirbst/semanticweb

```
@prefix : <https://raw.githubusercontent.com/mkirbst/semanticweb/master/scheme#>.

:Employee a rdfs:Class .
:Publications a rdfs:Class .

:Int a rdfs:Datatype; owl:onDatatype  xsd:integer; xsd:minInclusive 1 .
:String a rdfs:Datatype; owl:onDatatype  xsd:string .

:cuid a rdf:Property; rdfs:domain :Employee; rdfs:range :Int .
:lastname a rdf:Property; rdfs:domain :Employee; rdfs:range :String .
:firstname a rdf:Property; rdfs:domain :Employee; rdfs:range :String .
:degree a rdf:Property; rdfs:domain :Employee; rdfs:range :String .
:dnbautorid a rdf:Property; rdfs:domain :Employee; rdfs:range :Int .
:birth a rdf:Property; rdfs:domain :Employee; rdfs:range :Int .

:dnbpubid a rdf:Property; rdfs:domain :Publication; rdfs:range :Int .
:dnbautorid a rdf:Property; rdfs:domain :Publication; rdfs:range :Int .
:title a rdf:Property; rdfs:domain :Publication; rdfs:range :String .
:pubdate a rdf:Property; rdfs:domain :Publication; rdfs:range :Int .
```

Listing 2: RDF-Schema der exportierten HTWK-Mitarbeiterdaten

Da diese Datei im Format N3 vorliegt, wurde diese mit einem RDF-Translator [4] in das XML-Format überführt.

Die vom Java-Programm generierte RDF-Datei kann nun direkt über die Webschnittstelle in OntoWiki importiert werden und hat folgendes Format.

```
<rdf:RDF
  xmlns:semweb="https://raw.githubusercontent.com/mkirbst/semanticweb/master/scheme#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
>

  <semweb:Employee rdf:about="https://raw.githubusercontent.com/mkirbst/semanticweb/master/htwkstaff.json#cuid597">
    <semweb:cuid>597</semweb:cuid>
    <semweb:lastname>Siebeck</semweb:lastname>
    <semweb:firstname> Andrea</semweb:firstname>
    <semweb:degree>Dipl.-Angl.</semweb:degree>
    <semweb:faculty>AAA</semweb:faculty>
    <semweb:dnbautorid>0</semweb:dnbautorid>
    <semweb:birth>0</semweb:birth>
  </semweb:Employee>
  <semweb:Employee rdf:about="https://raw.githubusercontent.com/mkirbst/semanticweb/master/htwkstaff.json#cuid8">
    <semweb:cuid>8</semweb:cuid>
    <semweb:lastname>Engel</semweb:lastname>
    <semweb:firstname> Heike</semweb:firstname>
    <semweb:degree>Dipl.-Wirtschaftsinf.</semweb:degree>
    <semweb:faculty>DF</semweb:faculty>
    <semweb:dnbautorid>0</semweb:dnbautorid>
    <semweb:birth>0</semweb:birth>
  </semweb:Employee>
```

[4] http://rdf-translator.appspot.com/

```
24 </rdf:RDF>
```

Listing 3: RDF-Format der exportierten HTWK-Mitarbeiterdaten

3.2 Extraktion der Daten der Deutschen Nationalbibliothek

Unter der Webadresse [5] lassen sich die benötigten Daten direkt im RDF-Format herunterladen. Die heruntergeladenen Dateien sind jedoch mit 9 und 12 GByte so groß, das diese nicht direkt über die Webschnittstelle des OntoWiki importiert werden können.

Aus diesem Grund filtert die JAVA-Anwendung aus den Datensätzen nur diese heraus, deren Autor den Vor- und Nachname von HTWK-Mitarbeitern enthält. Um diese Operationen effizient durchführen zu können, werden in der JAVA-Anwendung Hashmaps erzeugt, die als Schlüssel die betreffenden RDF-Tags so wie Vor- und Nachname als Hashwert enthalten. Durch diese Implementierungsvariante werden die Vorteile der Hashmap-Datenstruktur genutzt. Außerdem werden so wenig wie möglich rechenintensive Stringmanipulationen durchgeführt.

```
199 Map<String, emp>hEmps = new HashMap<String, emp>();
```

Listing 4: Auszug JAVA-Anwendung: Hashmap zur schnelleren Filterung der Normdaten

```
233 //DNB-GND.rdf name format as hasmap key value
234 hEmps.put("<rdf:li>"+name+"</rdf:li>", tmpemp);
```

Listing 5: Auszug JAVA-Anwendung: Einfügen von Objekten in die Hashmap

```
135  int hits = 0;
136  String begintag = "<rdf:Description";
137  String endtag = "</rdf:Description";
138  String nametag = "<gndo:preferredNameForThePerson>";
139  StringBuilder t3 = new StringBuilder();
140  String ts3 = "";
141  boolean found = false;
142
143  BufferedReader br = new BufferedReader(new FileReader(gndfilepath));
144  String line;
145
146  while ((line = br.readLine()) != null) {
147   String linetrimmed = line.trim(); // performance: trim only once
148
149   //linetrimmed begins with begintag OR nametag OR endtag OR other
150   if(linetrimmed.startsWith(begintag))
151   {
152    t3 = new StringBuilder();
153    t3.append(line+"\n");
```

[5] http://datendienst.dnb.de/cgi-bin/mabit.pl?userID=opendata&pass=opendata&cmd=login

```
154     found = false;
155   } else if (linetrimmed.startsWith(nametag))
156   {
157    t3.append(line+"\n");
158    if(hEmps.containsKey(linetrimmed)) {  //hEmps == hashmap containing htwk ↩
           employee objects
159     found = true;
160     hits++;
161    }
162   } else if (linetrimmed.startsWith(endtag)) {
163    t3.append(line+"\n\n");
164    if(found == true)  {
165     System.out.println(hits + " " + t3.toString());
166     writer.write(t3.toString());
167    }
168   } else {
169    t3.append(line+"\n");
170   }
171  }
```

Listing 6: Auszug JAVA-Anwendung: RDF-Parser

Das resultierende JAVA-Programm verarbeitet die gesamten Datensätze beider Dateien innerhalb einer Laufzeit von circa 2 Minuten. Die zu Grunde liegende Hardware ist eine Laptop der Marke HP EliteBook 8470w (CPU:Intel(R) Core(TM) i7-3740QM CPU @ 2.70GHz, RAM 16 GB, Primärspeicher: Micron SSD 250 GB).

Die resultierenden beiden Dateien haben exakt das gleiche RDF-Format wie die Eingabedateien, jedoch bereinigt um alle Datensätze, deren Autoren nicht einen Namen eines HTWK-Mitarbeiters tragen. Diese Dateien lassen sich nun über die Webschnittstelle von OntoWiki importieren.

4 Verlinkung von Ressourcen

Mit den 3 verwendeten Datenquellen lassen sich die Daten auf 2 Arten verknüpfen um die gewünschten Ergebnisse zu erhalten. Die Daten der Deutschen Nationalbibliothek sehen für jeden Autor eine eindeutige Identifikationsnummer vor. Als problematisch haben sich im Verlauf des Projektes dabei 2 Umstände erwiesen.

Die Deutsche Nationalbibliothek weist darauf hin das sich die für das Projekt verwendeten Daten noch in einem Migrationsprozess befinden und noch nicht vollständig konsistent sind. Daraus resultiert, dass manche Autoren mehrfach mit unterschiedlichen Publikationen im Datenbestand vorhanden sind.

Das zweite Problem besteht darin, dass von der Datenquelle HTWK-Telefonbuch nur Nachname und Vornamen für die Verlinkung einbezogen werden können. Weitere hilfreiche Attribute wie beispielsweise das Geburtsjahr sind in dieser Datenquelle nicht hin-

terlegt. Für ein wirklich tragfähiges Ergebnis des Projekts sollten folglich konsistente Datenquellen heran gezogen werden.

5 Anfrage an die Forschungswissensbasis

5.1 SPARQL-Anfrage

```
PREFIX ns3: <http://purl.org/ontology/bibo/>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ns2: <http://rdvocab.info/>
PREFIX dct: <http://purl.org/dc/terms/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX ns1: <http://iflastandards.info/ns/isbd/elements/>
PREFIX ns0: <http://id.loc.gov/vocabulary/relators/>
PREFIX semweb: <https://raw.githubusercontent.com/mkirbst/semanticweb/master/↩
    scheme#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX marcRole: <http://id.loc.gov/vocabulary/relators/>

select ?person ?lastname ?firstname (count(?document) as ?count)
where {
  ?person rdf:type semweb:Employee;
          semweb:lastname ?lastname;
          semweb:firstname ?firstname .

  ?document rdf:type ns3:Document;
            marcRole:edt ?editor .

  Filter (
   #?editor = concat(?lastname, ", ", ?firstname)
    strstarts(concat(?editor, ""), ?lastname)
    AND strends(concat(?editor, ""), ?firstname)
  )
  }
LIMIT 200
```

Listing 7: SPARQL-Abfrage

5.2 Ergebnis der Anfrage

6 Interpretation und Zusammenfassung

Es handelt sich hier um ein Muster, das Dokument ist daher unvollständig...