Informationspraxis Bd. 1, Nr. 2 (2015)

DOI: http://dx.doi.org/10.11588/ip.2015.2.23784

Christian HAUSCHKE, Elena LIVENTSOVA

Erstellung wiederverwendbarer RDF-Geodaten mit Google Refine

Zusammenfassung

Das Forschungsinformationssystem VIVO bietet als Linked-Data-basiertes System die Möglichkeit, RDF-Daten aus anderen Quellen wiederzuverwenden. In der Praxis kann man dabei auf Konvertierungsprobleme stoßen, weil relevante Daten häufig nur in tabellarischem Format vor, z.B. als CSV-Datei, vorliegen. Zur Datenkonvertierung existieren verschiedene Werkzeuge, viele dieser Werkzeuge erfordern jedoch entweder spezielle technische Umgebungen (oft Linux-Systeme) oder sie sind in der Bedienung sehr anspruchsvoll. Im Artikel wird ein Workflow für die Konvertierung von Daten aus GeoNames für VIVO mit Google Refine beschrieben.

Schlüsselwörter

Linked Open Data, Geographische Daten, Datenpublikation, GeoNames, VIVO

Creating resuable RDF-Geodata with Google Refine

Abstract

The research information system VIVO can, as a linked data based system, re-use RDF data from different sources. In practice, one can encounter data conversion issues. Often, data is only available as a CSV file. There are several tools to convert this data. However, many of these tools require either specific technical environments (often Linux systems), or they are very challenging to use. In this article, a workflow for converting data from GeoNames using Google Refine for instant use in VIVO is described.

Keywords

Linked open data, geographical data, data publication, GeoNames, VIVO

Inhaltsverzeichnis

1 Ausgangslage und Zielsetzung

VIVO (http://vivoweb.org) enthält vorinstalliert einen geringen Grundbestand an Geodaten. Dabei handelt es sich ausschließlich um Informationen über Staaten und die Bundesstaaten der USA. Die Informationen über Staaten stammen aus der Geopolitical-Ontologie und haben je Entität einen entsprechenden Uniform Resource Identifier (URI).

Beispiel: http://aims.fao.org/aos/geopolitical.owl#Germany

Unser Ziel war die hierarchisch korrekte Verknüpfung der vorliegenden Daten mit weiteren Entitäten zu Städten und deutschen Bundesländern. Als Datenquelle wurde GeoNames ausgewählt, da die dort zur Verfügung gestellten Daten umfangreich, gut gepflegt, hierarchisch miteinander verknüpft und in Linked-Data-Kontexten schon gebräuchlich sind. Die ebenfalls evaluierten Daten aus DBPedia wurden aufgrund der heterogenen Qualität als Datenquelle verworfen. Zukünftig könnte Wikidata eine interessante Quelle sein. Die Orte und Länder in GeoNames sind über eine GeoNames-URI identifiziert.

Beispiel: http://sws.geonames.org/6557470 (für die Stadt Nordhorn)

2 Google Refine

Für die Aufbereitung der Daten wurde Google Refine (GR, https://code.google.com/p/google-refine/) verwendet. Dabei handelt es sich um ein Werkzeug zur Bereinigung von Datensätzen.

Die höchstzulässige Dateigröße ist abhängig von der verwendeten Hardware, insbesondere vom zur Verfügung stehenden RAM. Während (Larsson 2013) eine Grenze bei 512MB vermutet, kann man GR mehr RAM zur Verfügung stellen und dann deutlich größere Dateien öffnen. Carmen Wong hat die technischen Spezifikationen zu (Wong u.a. 2015) auf Anfrage per Mail folgendermaßen angegeben:

> „*For our data, we had 16,783,855 records and 16 variables (columns). The original file was a comma delimited text document with a size of 906,318 KB and this was imported into OpenRefine for standardisation. To import this file, the maximum heap size as well as the memory allocated to run OpenRefine were increased in accordance to the instructions accessible on the website https://github.com/OpenRefine/OpenRefine/wiki/FAQ:-Allocate-More-Memory. Our Windows 7 Professional computer has a 64-bit Intel Xeon CPU E5-1620v2 @ 3.70GHz processor with 32GB installed memory.*“

GR ist eine Open-Source-Software, die seit 2011 als OpenRefine (http://openrefine.org/) weitergeführt wird. GR ist besonders geeignet für die Bearbeitung tabellarischer Daten. Dafür stehen verschiedene Werkzeuge zur Verfügung, unter anderem:

- Mittels Facetten können Zeilen mit bestimmten Zeichenketten gefunden und gezielt bearbeitet werden. Beispiel: In einer großen Adressdatenbank könnte man sich alle Personen heraussuchen, für die in der Spalte „Stadt“ der Heimatort „Berlin“ angegeben ist.
- Mit Expressions (Ausdrücken) können Daten mit (regulären) Ausdrücken und verschiedenen Befehlen bearbeitet werden, zum Beispiel Parsen von XML- oder JSON-Dateien, aber auch komplexe Suchen-Ersetzen-Befehle. Die Verwendung der Google Expression Language (GREL) macht GR zu einem mächtigen Werkzeug. Für die meisten Anwendungsfälle sind in der Dokumentation Beispiel-Ausdrücke verfügbar, die auf die eigenen Bedürfnisse angepasst werden können.
- Eigene Daten können mit anderen angereichert werden. Dies kann entweder geschehen über das Auslesen von Webseiten oder anderen Dateien, wie im Abschnitt „Daten einreichern“ beschrieben, oder über *Reconciliation*. Damit wird ein Verfahren beschrieben, mittels dessen sich eigene Daten mit Fremddaten matchen und mappen lassen (Vgl. Verborgh & Wilde 2013:65–80).
- Datenbereinigung, zum Beispiel Entfernung von Leerzeichen am Anfang und Ende einer

Zeichenkette, Deduplizierung oder Normalisierung von Werten.

Die Benutzung wird in der Dokumentation (2015) und in (Verborgh & Wilde 2013) erläutert. Um Daten als RDF (Resource Description Framework) zu exportieren, ist zusätzlich zu GR die Installation der RDF-Extension für GR (http://refine.deri.ie/) notwendig.

Da eine funktionierende GR-Installation auf dem verwendeten Rechner lief, wurde für dieses Projekt nicht extra OpenRefine installiert. Unterschiede in der Bedienung und in der Funktionsweise zu Google Refine sind marginal, die RDF-Extension funktioniert auch mit OpenRefine.

3 Datenquellen

Informationen über Staaten sollten zwecks Kompatibilität mit den in VIVO vorhandenen Daten aus der geopolitical-Ontologie stammen. Informationen über die Bundesländer wurden manuell erstellt.

Als Quelldaten für Informationen über Städte wurden die Datensätze DE.txt und cities15000.txt aus dem GeoNames-Downloadverzeichnis (http://download.geonames.org/export/dump/) ausgewählt. Die enthaltenen Daten sind in den begleitenden Informationen (http://download.geonames.org/export/dump/readme.txt) dokumentiert. Die Dateien wurden in je ein GR-Projekt importiert.

3.1 Beispiel: Städte in Deutschland

3.1.1 Filtern der Daten

Anhand des *feature codes*, der Klassifizierung von Objekttypen auf GeoNames, wurden alle für unser Forschungsinformationssystem irrelevanten Objekte aus dem Projekt entfernt. Darunter befanden sich unter anderem Bahnhöfe (feature code RSTN), Kanäle (CNL) oder Parks (PRK). Beibehalten wurden die administrativen Einheiten des Typs ADM4 (feature code für “fourth-order administrative division; a subdivision of a third-order administrative division”). Für solche Arbeitsschritte bietet GR sogenannte Filter für Texte und numerische Werte. Mit einem Textfilter wurden die ADM4-Objekte im Ausklappmenü der Spalte “feature code” (s. Abb. 1) ausgewählt:

→ *Im Textfilter ”ADM4” eingeben*

→ *include “false”*

→ *erste Spalte “All”*

→ *Edit rows*

→ *Remove all matching rows*

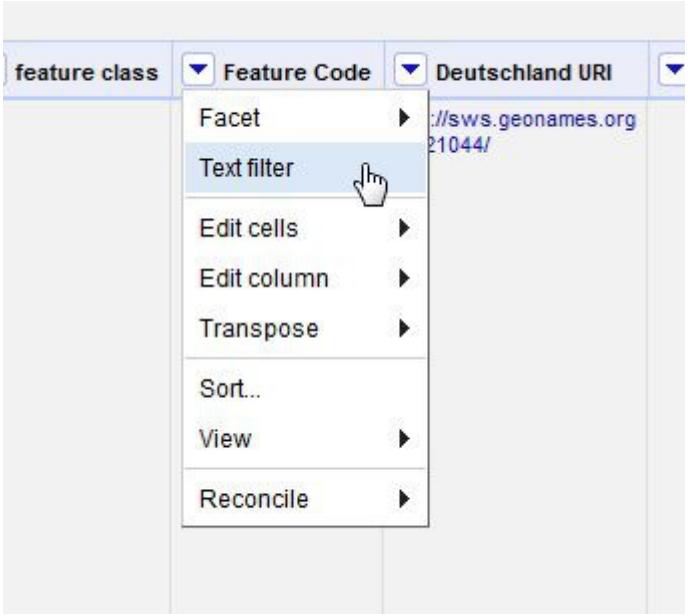

Abb. 1: Filter- und Bearbeitungsmöglichkeiten in Google Refine

Über einen weiteren Filtervorgang wurden dann Städte mit weniger als 1000 Einwohnern ausgeschlossen. Dazu wird aus der Spalte “Population” eine neue Spalte “Size” generiert:

→ *Edit column*

→ *Add column based on this column (Name: „Size“)*

→ *”value>999” in dem Feld “Expression” statt “value” eingeben.*

Es entsteht eine neue Spalte „Size“ mit den Werten true für die Städte ab 1000 Einwohnern und false für die Städte unter 1000 Einwohnern. In dieser Spalte wenden wir die Funktion Text facet an. Die mit false markierten Datensätze sind zu löschen. Dazu werden die false-Werte durch “include” ausgewählt. Dann in der ersten Spalte „All“:

→ *Edit rows*

→ *Remove all matching rows*

3.1.2 Anreichern der Daten

Zusätzliche Informationen, die in der heruntergeladenen Datei nicht enthalten waren, lassen sich aus dem RDF des jeweiligen Objektes entnehmen. Dafür wird der URL der Quelle benötigt, in diesem Fall also die RDF-Beschreibungs-URI in GeoNames:

Beispiel: http://sws.geonames.org/291074/about.rdf

Der Aufbau aller URIs der geographischen Objekte in GeoNames ist identisch. Die Nummer ist die GeoNames-ID, die wir in der heruntergeladenen Datei schon haben. Diese mit dieser ID aufgebaute RDF-Beschreibungs-URI wird später auch als Individual-URI ins VIVO übernommen. Zur Erstellung einer neuen Spalte mit den GeoNames-URIs sind folgende Schritte nötig:

In der Spalte “GeoNames-ID”:

→ *Edit column*

→ *Add column based on this column…*

→ *GREL-Ausdruck in dem Feld “Expression” anstatt “value” eingeben: “http://sws.geonames.org/”+value+”/about.rdf”*

→ *Neue Spalte benennen, z.B. “* GeoNames-*URI”.*

Die Informationen zu Orten im RDF-Format konnten über die „fetch“-Funktion des GRs abgerufen und in die Tabelle eingetragen werden:

Im Ausklappmenü der Spalte “GeoNames-URI”:

→ *Edit column*

→ *Add column by fetching URLs…*

→ *Spalte benennen, z.B. “GeoNames-RDF”*

Je nach Anzahl der Datensätze und der Reaktionszeiten des Servers kann dieser Prozess einige Stunden in Anspruch nehmen.

Nachdem wir in der neuen Spalte “GeoNames-RDF” das vollständige RDF/XML für jeden Ort haben, kann man daraus Informationen extrahieren: z.B. GeoNames-URI für das Bundesland, in dem jeweilige der Ort liegt (Kode “ADM1” in der GeoNames-Datenbank).

Die Herangehensweise, um Zeichenketten in GR aus Daten einer Spalte zu extrahieren, sieht wie folgt aus:

In der Spalte mit den HTML- oder XML-Daten (in unserem Projekt “Geonames-RDF”):

→ *Edit column*

→ *Add column based on this column…*

→ *GREL-Ausdruck in dem Feld “Expression” anstatt “value” eingeben.*

Nun kann man – wie in den folgenden Beispielen – GREL-Ausdrücke zum Extrahieren von der gewünschten Informationen eingeben. An der Stelle wird der Befehl *parseHtml()* verwendet, der auch für Parsing von XML geeignet ist.[1] Die genaue Beschreibung der Befehlsanwendung ist in der GREL-Dokumentation (unter anderem z. B.: https://github.com/OpenRefine/OpenRefine/wiki/GREL-Other-Functions) zu finden.[2]

- Um Bundesland-GeoNamesURL zu extrahieren: *value.parseHtml().select("gn|parentADM1[rdf:resource],[rdf:resource]")[0].htmlAttr('rdf:resource')*
- Um den Wikipedia-Link zu extrahieren: *value.parseHtml().select("gn|wikipediaArticle[rdf:resource]")[0].htmlAttr('rdf:resource')*
- Um den DBpedia-Link zu extrahieren: *value.parseHtml().select("rdfs|seeAlso[rdf:resource]")[0].htmlAttr('rdf:resource')*

1 Vgl. http://stackoverflow.com/questions/15893426/parsing-xml-using-google-refine

2 Noch mehr GREL-Ausdrücke für XML-Parsing sind z.B. auch unter https://github.com/OpenRefine/OpenRefine/wiki/Recipes beschrieben.

Abb. 2: RDF/XML mit zu extrahierenden Daten

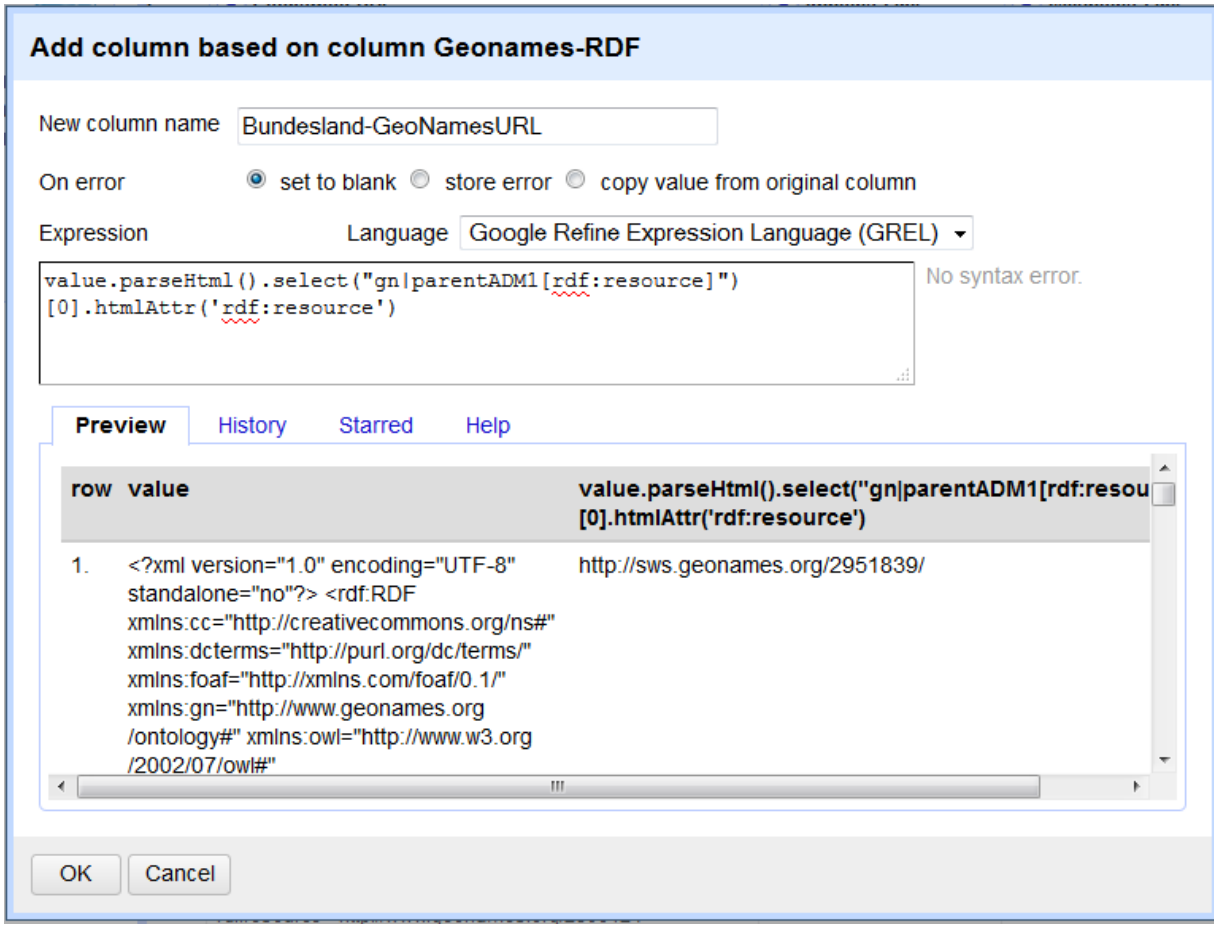

Abb. 3: Beispiel für Daten-Extraktion aus RDF/XML

Zur Reinigung der Daten sollten manchmal Inhalte in Zellen mit der Funktion Transform geändert werden, zum Beispiel, um aus der Spalte "Name (dt)" die Zeichenkette “, Kreisstadt” zu entfernen:

In der Spalte "Name (dt)":

→ *Edit cells*

→ *Transform...*

→ *im Feld “Expression” anstatt “value” eingeben: value.replace(", Kreisstadt", "")*

In der VIVO-Otologie gibt es die Möglichkeit die Koordinaten des Ortes über Geo-URI auszudrücken. Geodaten können laut RFC 5870 (Internet Engineering Task Force 2010) als URI in der Form "geo:latitude,longitude" angegeben werden. Die Werte der geographischen Breite und Länge sind im GR-Projekt vorhanden. Zur Entstehung einer entsprechender Schreibweise, sollen die Inhalte der Spalten latitude und longitude verknüpft und mit der

Zeichenkette “geo:” eingeleitet werden:

In der Spalte "latitude":

→ *Edit column*

→ *Add column based on this column*

→ *GREL-Ausdruck im Feld “Expression”: "geo:"+value+","+cells["longitude"].value*

3.1.3 Export der Daten

Der Aufbau einer RDF-Datei in Turtle-Serialisierung erfolgte durch den „RDF-Skeleton-Editor“ der oben erwähnten RDF-Extension. Die abzubildenden Entitäten und ihre Beziehungen zueinander wurden vorher wie in Abb. 4 modelliert.

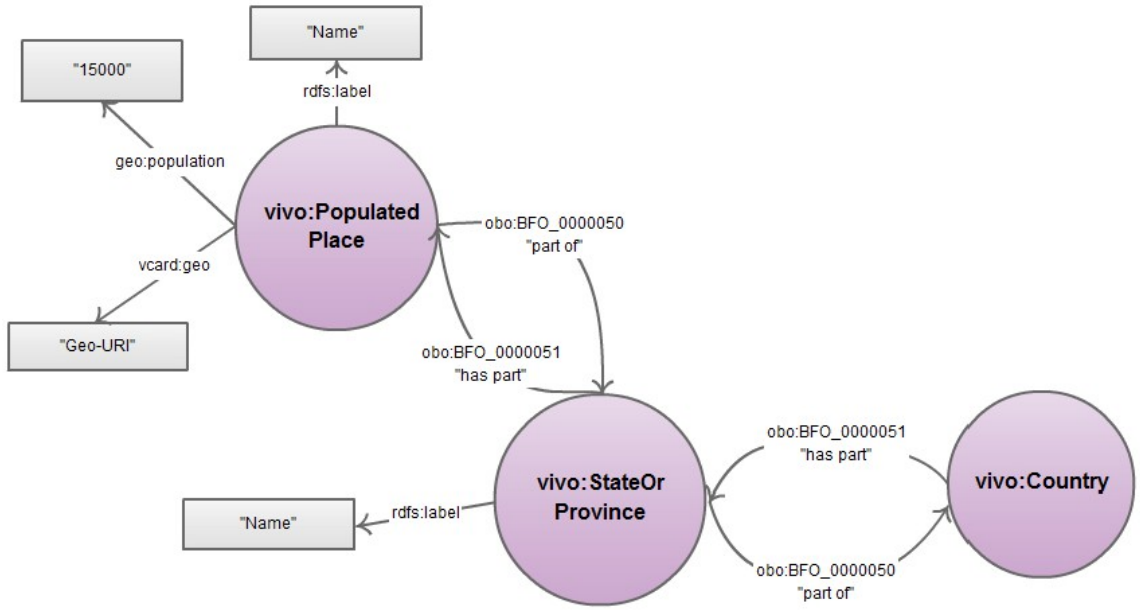

Abb. 4: Modell für GermanyPopulatedPlaces.ttl

Die Base URI wird zu “http://sws.geonames.org/” geändert. Zusätzlich zu den schon vorhandenen Ontologien (*Available Prefixes*) wurden die Geopolitische FAO-Ontologie (Prefix “geo”), VIVO-Ontologie (Prefix “vivo”), vCARD („vcard“) und OBO („obo“) hinzugefügt.

→ *add prefix (S. Abb. 5)*

→ *prefix eingeben*

→ *Advanced*

Als root node gilt die Spalte “GeoNames-ID”:

→ *“Row Index” anklicken*

→ *Use content from cell … “GeoNames-ID”*

→ *The cell's content is used … as an URI*

Der URI für den Ort hat dadurch im Export die gewünschte Form wie im Beispiel http://sws.geonames.org/2806124. Durch das Betätigen der “*add rdf:type*” - Schaltfläche kann man dem Wurzelknoten die Klasse *vivo:PopulatedPlace* zuweisen. Über die Funktion “*add property*” kann man die Beziehungen zu anderen Entitäten und Daten gestalten.

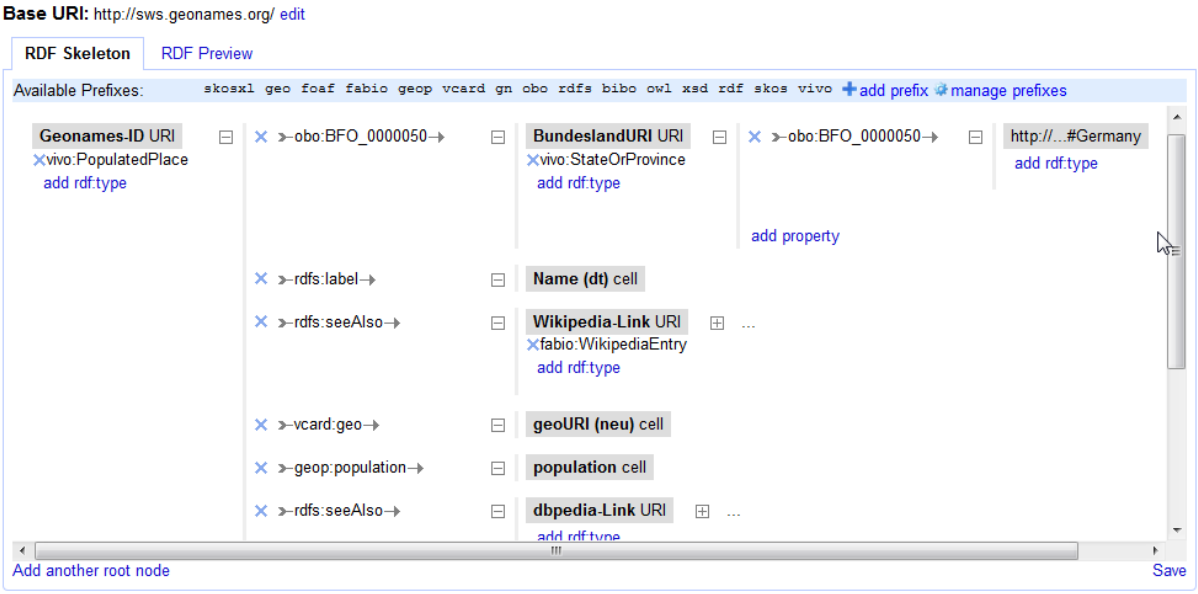

Abb. 5: RDF-Skeleton (Ausschnitt) für GermanyPopulatedPlaces.ttl

Abschließend wurden die Daten als Turtle-Datei *GermanyPopulatedPlaces.ttl* exportiert.

```
<http://sws.geonames.org/2806124> a vivo:PopulatedPlace .

<http://sws.geonames.org/2951839/> a vivo:StateOrProvince ;
        obo:BFO_0000050 geop:Germany .

<http://sws.geonames.org/2806124> obo:BFO_0000050 <http://sws.geonames.org/2951839/> ;
        rdfs:label "Wörnitz" .

<http://en.wikipedia.org/wiki/W%C3%B6rnitz> a fabio:WikipediaEntry .

<http://sws.geonames.org/2806124> rdfs:seeAlso <http://en.wikipedia.org/wiki/W%C3%B6rnitz> ;
        vcard:geo "geo:49.25,10.25" ;
        geop:population "1701" ;
        rdfs:seeAlso <http://dbpedia.org/resource/W%C3%B6rnitz> .

<http://sws.geonames.org/2806236> a vivo:PopulatedPlace ;
        obo:BFO_0000050 <http://sws.geonames.org/2951839/> ;
        rdfs:label "Wonneberg" .

<http://en.wikipedia.org/wiki/Wonneberg> a fabio:WikipediaEntry .
```

Abb. 6: Ausschnitt aus GermanyPopulatedPlaces.ttl

3.2 Beispiel: Städte weltweit

Für weitere Städte (alle Städte weltweit ab einer Einwohnerzahl von 50.000 Einwohnern) wurde das zweite, aus der Datei cities15000.txt generierte GR-Projekt geöffnet. Die Filterung und Anreicherung erfolgte analog zum oben beschriebenen Vorgang:

- Städte mit weniger als 50.000 Einwohnern wurden entfernt
- Deutschen Städten wurden ausgeschlossen (ISO2-Kürzel “DE”)
- Pro Geo-URI wurde eine Entität erstellt
- Aus den RDF-Daten wurde der URI des Landes geholt.

Um die Verknüpfung zu den schon im VIVO vorhandenen Ländern zu realisieren, wurde ein drittes GR-Projekt “ISO-geopolitical” mit den ISO2-Kürzeln der Länder und deren geopolitical-URIs aus dem VIVO erstellt. Dazu wurden die Daten aus der geopolitical-Ontologie über den OpenLink Virtuoso SPARQL Query Editor (http://demo.openlinksw.com/sparql/?default-graph-uri=http%3A%2F%2Faims.fao.org%2Faos%2Fgeopolitical.owl&qtxt=select%20%3Fs%20%28str%28%3Fobject%29%20as%20%3Flabel%29%0Afrom%20%3Chttp%3A%2F%2Fwww.fao.org%2Fcountryprofiles%2Fgeoinfo%2Fgeopolitical%2Fdata%2Fself_governing%3E%0Awhere%20{%3Fs%20%3Chttp%3A%2F%2Fwww.w3.org%2F1999%2F02%2F22-rdf-syntax-ns%23type%3E%20%3Chttp%3A%2F%2Faims.fao.org%2Faos%2Fgeopolitical.owl%23self_governing%3E.%20%3Fs%20%3Chttp%3A%2F%2Faims.fao.org%2Faos%2Fgeopolitical.owl%23codeISO2%3E%20%3Fobject.%20}&should-sponge=soft&format=text

%2Fcsv&timeout=50000) abgefragt, als CSV-Datei gespeichert und als eigenes GR-Projekt „ISO-geopolitical“ geöffnet (s. Abb. 7).

Abb. 7: SPARQL-Query für ISO2-Kürzel der Länder

Anhand der ISO2-Abkürzungen wurden die geopolitical-URIs aus dem Projekt “ISO-geopolitical” geholt. In der Spalte ISO2 des Projektes “WorldPopulatedPlaces”:

→ *Edit column*

→ *Add column based on this column*

→ *GREL-Ausdruck: cell.cross("ISO-geopolitical", "ISO2").cells["geopoliticalURI"].value[0]*

Darauf aufbauend wurden die gewünschten Informationen und Verknüpfungen nach dem Datenmodell in Abb. 8 als RDF-Skelett modelliert und schließlich exportiert.

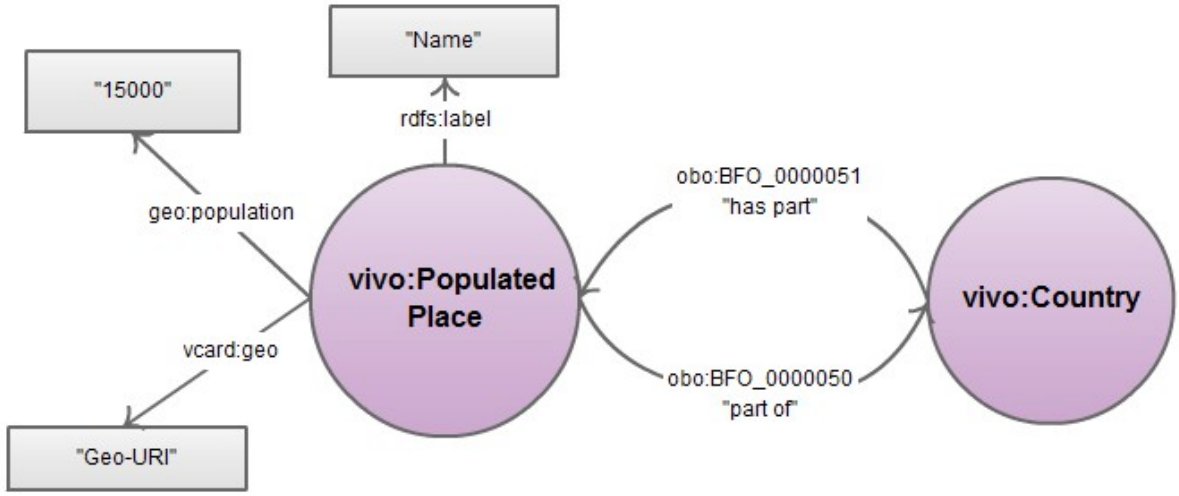

Abb. 8: Datenmodell für Stadt-Staat-Verknüpfung

4 Veröffentlichung der erzeugten Daten

Um auch Dritten nicht nur die Nachnutzung, sondern auch die Erweiterung oder Korrektur an den produzierten Datensätzen zu erlauben, wurden die Daten auf GitHub veröffentlicht und dokumentiert und anschließend auf Wunsch der VIVO-Community durch eine Publikation des GitHub-Repositories (Liventsova & Hauschke 2014) zitierbar gemacht.

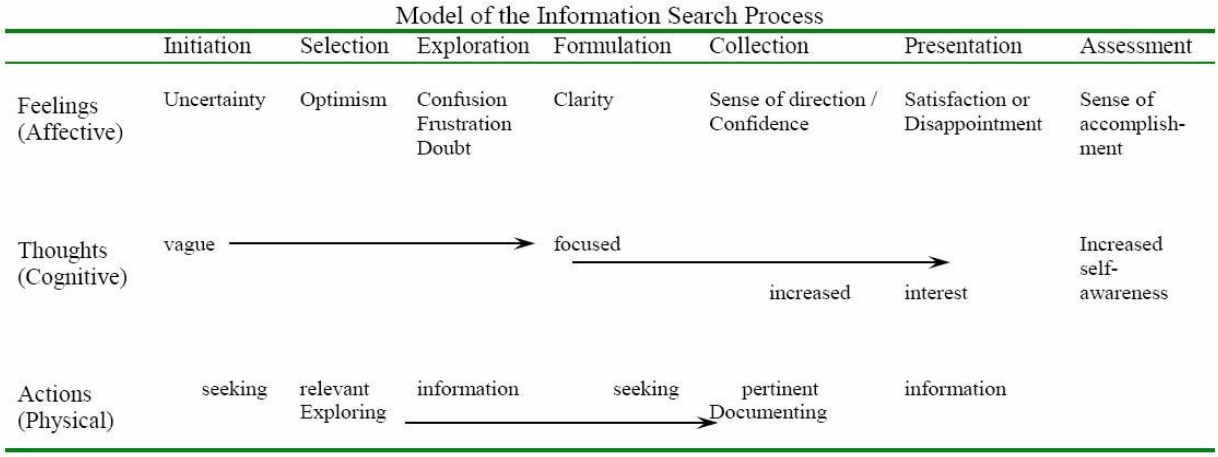

Abb. 9: Das Modell Information Search Process (ISP) (nach Kuhlthau 2013:95)

Die besondere Leistung des ISPs ist die die Ergänzung eines linearen Informationsprozesses um affektive und kognitive Faktoren, die einzelnen Arbeitsschritten zugeordnet sind. Das Modell vernachlässigt jedoch, dass Such- und Lernprozesse rekursiv sind bzw. sich Gefühle und Gedanken zu einzelnen Schritte und dem gesamten Suchprozess asynchron verhalten können. Dennoch findet der ISP eine breite fachliche Rezeption und Akzeptanz, die sich u.a. in einer

Weiterentwicklung des ISPs mit rekursiven und asynchronen Zyklen im Information Seeking-Modell von Marchionini findet (Marchionini 1997) bzw. eine Bestätigung in der Übertragung auf das Suchverhalten in Gruppen bei Shah et al. zeigt (Shah & González-Ibáñez 2010).

5 Fazit

Die Aufbereitung und die Konversion von tabellarischen Daten als RDF ist auch ohne umfangreiche technische Vorkenntnisse mit Google/Open Refine möglich. Die in diesem Projekt verwendeten GREL-Ausdrücke sind meist aus der GR-Dokumentation übernommene und für unsere Zwecke angepasste Recipes („Rezepte“).

Für darüber hinausgehende Fragen ist eine aktive Experten-Community auf Stackoverflow.com bei speziellen Problemstellungen gezielt ansprechbar. Auf eine Fragestellung im Rahmen des hier beschriebenen Projekts gestellte Frage (http://stackoverflow.com/q/22014417/1784133) wurde innerhalb von drei Stunden konkret und korrekt geantwortet.

Für automatisierte und wiederkehrende Datenkonversionen sind Skriptsprachen wie Python oder kommandazeilenbasierte Programme wie Catmandu (http://librecat.org/Catmandu/) sicherlich komfortabler. Zu einem Vergleich von Catmandu und LODRefine, der Kombination von OpenRefine und der auch für das in diesem Artikel beschriebene Projekt verwendeten RDF-Extension siehe (Harlow 2015) .GR empfiehlt sich besonders für nicht wiederkehrende Aufgaben, für die Arbeit mit heterogenen und anzureichernden Daten und vor allem auch als Alternative für Einsteiger in die Linked-Data-Produktion. Das GR sich allerdings auch halb-automatisch betreiben lässt, ist u.a. von (Silbermann u.a. 2013) beschrieben. Weitere Informationen und Anregungen für die Arbeit mit GR oder OpenRefine liefern u.a. (Hawksey 2015), (van Hooland & Verborgh 2015:94–107), (Qasmi 2014) oder (Schelper 2015).

Literatur

Harlow, Christina 2015. Data Munging Tools in Preparation for RDF: Catmandu and LODRefine. *Code4Lib*(30). Online im Internet: URL: http://journal.code4lib.org/articles/11013.

Hawksey, Martin 2015. OpenRefine(ing) and visualizing library data, in Engard, Nicole C. & Sauers, Michael P. (Hg.): *More library mashups: Exploring new ways to deliver library data*. London: Facet, 43–58.

Internet Engineering Task Force (IETF) 2010. *A Uniform Resource Identifier for Geographic Locations ('geo' URI)*. URL: http://tools.ietf.org/html/rfc5870.

Larsson, Per 2013. *Evaluation of Open Source Data Cleaning Tools: Open Refine and Data Wrangler*. URL: http://courses.cs.washington.edu/courses/cse544/13sp/final-projects/p12-plarsson.pdf.

Liventsova, Elena & Hauschke, Christian 2014. *geodata: Populated places for VIVO*. (Zenodo). URL: http://dx.doi.org/10.5281/zenodo.13101.

OpenRefine: A free, open source, powerful tool for working with messy data 2015. URL: http://openrefine.org/ [Stand 2015-07-09].

Qasmi, Muhammad A. 2014. RDF Quality Extension for OpenRefine. Masterarbeit. Rheinische Friedrich-Wilhelms-Universität Bonn. URL: http://eis-bonn.github.io/Theses/2014/Muhammad_Ali_Qasmi/thesis.pdf [Stand 2015-10-20].

Schelper, Katja 2015. Open Refine, in Blümel, Ina (Hg.): *VIVO-Handbuch*. URL: http://handbuch.io/w/VIVO-Handbuch/Open_Refine.

Silbermann, Jascha, u.a. 2013. RefPrimeCouch--a reference gene primer CouchApp. *Database : the journal of biological databases and curation*. Online im Internet: URL: http://dx.doi.org/10.1093/database/bat081.

van Hooland, Seth & Verborgh, Ruben 2015. *Linked Data for Libraries, Archives and Museums: How to clean, link and publish your metadata*. London: Facet Publishing.

Verborgh, Ruben & Wilde, Max de 2013. *Using OpenRefine: The essential OpenRefine guide that takes you from data analysis and error fixing to linking your dataset to the web*. Birmingham: Packt. (Community experience distilled).

Wong, Carmen K., u.a. 2015. Standardisation of the FAERS database: a systematic approach to manually recoding drug name variants. *Pharmacoepidemiology and drug safety* 24(7), 731–737. URL: http://dx.doi.org/10.1002/pds.3805

AutorInnen

Christian HAUSCHKE
Bibliothek der Hochschule Hannover
Ricklinger Stadtweg 118
D-30459 Hannover
http://hs-hannover.de/bibl
christian.hauschke@hs-hannover.de

Elena LIVENTSOVA
Niedersächsische Staats- und Universitätsbibliothek Göttingen
Papendiek 14
D-37073 Göttingen
http://www.sub.uni-goettingen.de
liventsova@sub.uni-goettingen.de