Webanwendung zur Extraktion von Teildatensätzen aus DBpedia

Christian Ernst, Dominik Strohscheer, Hans Angermann
Till Nestler, Marvin Hofer, Robert Bielinski, Jonas Rebmann

Inhaltsverzeichnis

Arbeitsplan

Projektvision

Das DBpedia-Projekt sammelt auf Grundlage von Wikpedia-Artikeln strukturierte Daten. Die so gesammelten RDF-Datensätze können allerdings Inkonsistenzen aufweisen, zum Beispiel aufgrund von fehlerhaften Wikipedia-Artikelversionen oder wegen Problemen mit verwendeten Maßeinheiten.

Ziel unseres Projekts ist, ein Softwarepaket zu entwickeln, das mit Hilfe des SPARQL-Endpunkts von DBpedia und dem RDFUnit Testsystem diese Inkonsistenzen finden und möglicherweise beheben kann. Desweiteren soll ein Webinterface für diese Applikation entwickelt werden, mit dem das Filtern und Korrigieren der Datensätze gesteuert werden kann.

Voraussetzungen

Voraussetzungen für die planmäßige Fertigstellung dieser Softwarepakete sind

- Ein Webserver, an den Ruby mit allen benötigten Paketen (beziehungsweise gems) in den richtigen Versionen angebunden ist
- Versenden von SPARQL-Anfragen in Ruby
- Verarbeitung von RDF-Daten
- Verwendung von Webtechnologien mit serverseitigem Ruby

Designübersicht und Funktionalität

Designvorstellungen

Das Softwarepaket besteht aus der Webanwendung mit integriertem Webinterface und einem Server, von welchem die Webanwendung aus abrufbar sein muss. Auf der Weboberfläche soll ein Eingabefeld für die Abfrage von Datensätzen an den DBPedia-Endpoint geben. Die Daten sollen dann im Hintergrund durch die Software auf Inkonsistenzen und weitere Fehler überprüft werden. Der Benutzer kann dabei bestimmte Ressourcen nach Typ selektieren und sich die Datentypen anzeigen lassen. Als Ausgabe auf der Weboberfläche soll eine Tabelle mit den geprüften Datensätzen zur Verfügung gestellt werden. Zur besseren Auswertung wird es ein Feld für Statistiken geben und zur Weiterverarbeitung der Daten erhält das Webinterface eine Downloadfunktion für die gefilterten Datensätze in Form von HTML oder RDF.

Funktionale Anforderungen

/FA 10/

Das Softwareprodukt soll Daten über DBPedias SPARQL-Endpoint abrufen können.

/FA 20/

Das Softwareprodukt muss die erhaltenen Datensätze mittels RDFUnit auf Inkonsistenzen, Fehler und Datentypen überprüfen können.

/FA 30/

Das Softwareprodukt muss eine Weboberfläche besitzen, über die Benutzer SPARQL-Anfragen an DBPedia stellen können.

/FA 40/

Auf dem Webinterface sollen die vom Benutzer abgefragten Datensätze ohne Inkonsistenzen und Fehler dargestellt werden.

/FA 50/

Auf dem Webinterface müssen sich Ressourcen nach Typ selektieren und sich die Datentypen anzeigen lassen.

/FA 60/

Auf dem Webinterface sollen die Relationen und Datentypen der gefilterten Datensätze angezeigt werden.

/FA 70/

Auf dem Webinterface sollen dem Benutzer relevante Statistiken zu den Datensätzen angezeigt werden.

/FA 80/

Auf dem Webinterface müssen die überprüften konsistenten Datensätze als Download verfügbar sein.

/FA 90/

Das Softwareprodukt kann das Mapping zwischen gleichartigen Relationen ermöglichen.

Nicht-funktionale Anforderungen

/NFA 10/

Das Softwareprodukt sollte nicht 90% der Datensätze verwerfen, falls entsprechend viele Inkonsistenzen vorliegen.

/NFA 20/

Das Softwareprodukt soll nicht die Datensätze innerhalb der DBPedia verändern, sondern die Datensätze selektiv auf der Weboberfläche anzeigen.

/NFA 30/

Das Softwareprodukt soll für eine große Anzahl an Datensätzen ausgelegt sein.

Arbeitspakete

1. **Vorprojekt**, 20%

   Das Vorprojekt ist der Einstieg in die verwendeten Technologien. Mit Abschluss des Vorprojekts sollten alle Technologien gewählt worden sein und alle Teammitglieder Erfahrung mit deren Benutzung gesammelt haben.

   Möglicherweise können im Rahmen des Vorprojekts schon erste Bestandteile des Hauptprojekts entstehen.

2. **Verwaltung großer Datenmengen**, 10%

   Als nächster Schritt sollte sichergestellt werden, dass die Technologien, die im Vorprojekt an Beispieldaten demonstriert wurden, gut auf die große Anzahl an Datensätzen, die unser Produkt verarbeiten können soll, skaliert.

   Deshalb sollten schon in einem frühen Stadium des Projekts größere Datenmengen verarbeitet werden.

3. **Finden von Inkonsistenzen**, 25%

   Die eigentliche Aufgabe des Produkts, die Analyse der Datensätze hinsichtlich möglicher Inkonsistenzen sollte in diesem Stadium des Projekts implementiert werden. Damit wäre die Kernanwendung benutzbar.

   Spätestens ab diesem Schritt sollte das Produkt ausgiebig getestet werden.

4. **Entwicklung des Webinterfaces**, 15%

   Da das Filterprogramm auch unabhängig vom Webinterface nutzbar sein sollte, kann in einem relativ fortgeschrittenen Stadium mit der Entwicklung des Webinterfaces begonnen werden.

5. **Testen und Fertigstellen**, 20%

   Um die Qualität des Endprodukts zu gewährleisten sollte ein Arbeitspaket nur für die Fehlerprüfung der Software eingeplant werden.

6. **Erweiterte Funktionalität**, 10%

   Falls das Produkt fertig ist und allen Qualitätsansprüchen genügt können am Ende optionale Funktionen ergänzt oder die Benutzerfreundlichkeit weiter verbessert werden.

Vorprojekt

Das Vorprojekt sollte zeigen, dass alle Voraussetzungen gegeben sind und trotzdem deutlich weniger umfangreich sein als das Hauptprojekt.

Wünschenswerte Bestandteile des Vorprojekts sind:

- Abfrage von beispielhaften Datensätzen über den SPARQL-Endpunkt von DBpedia
- Prüfen der Datensätze mit Hilfe von RDFUnit
- Einleiten der Abfrage durch ein Webinterface
- Anzeigen von Datensätzen in einem Webinterface

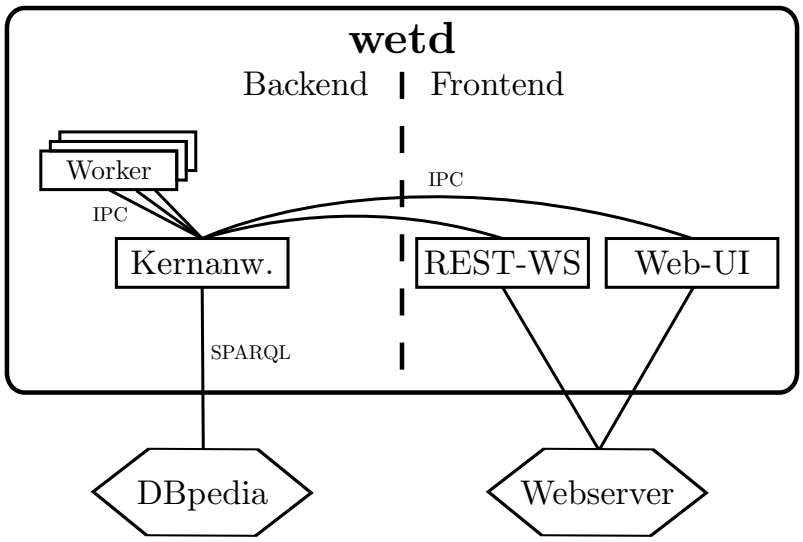

Abbildung 1: Mögliche Aufteilung in Prozesse.

Glossar

Abfragesprache

Eine Abfragesprache ist eine Sprache zur Suche nach Information innerhalb einer Datenbank. Das Ergebnis einer solchen Abfrage (Query) ist eine Teilmenge der zugrundeliegenden Informationen (Filterung).

CI

Continuous integration (CI) bezeichnet das automatische Erstellen von Softwarepaketen auf einem Server bzw. das Automatische durchführen von Unittests.

Verbreitete CI Systeme sind zum Beispiel Jenkins und Travis CI.

Datenbanksystem

System zur dauerhaften und effizienten Speicherung und Verwaltung großer Datenmengen.

DBpedia

DBpedia ist ein Crowd-Sourced-Community-Projekt, welches sich mit der Sammlung strukturierter Daten im RDF Format aus der Web-Enzyklopädie Wikipedia beschäftigt. Es werden komplexe Anfragen an Wikipedia und die Verlinkung verschiedenster Datensätze zu Wikipedia ermöglicht.[1] Der DBpedia RDF-Datensatz wird mit Hilfe des SPARQL-Endpunktes von OpenLink Vituoso zugänglich gemacht.

FA

Funktionale Anforderungen, d.h., Funktionen, die das Softwareprodukt konkret ausführen können muss. Bspw. “suche auf der Weboberfläche nach einem Begriff”

Inkonsistente Daten

Daten gelten als inkonsistent, wenn aus der Gültigkeit dieser Daten ein Zustand folgt der widersprüchlich ist.

Ein Film kann beispielsweise keine zwei verschiedenen Laufzeiten haben.

IPC

Interprozesskommunikation (IPC), ist eine Sammelbezeichnung für sämtliche Funktionen die ein Betriebssystem zur Verfügung stellt um den Datenaustausch zwischen verschiedenen darauf ausgeführten Prozessen zu ermöglichen.

IRI

Generalisierung von URIs, die den UNICODE Zeichensatz unterstützt.[2]

Linked Data

Linked Data beschreibt eine Methode strukturierte Daten zu veröffentlichen, sodass diese untereinander verlinkt werden können und so durch semantischer Anfragen nützlicher werden.

Literal

Literale sind atomare Werte die Werte direkt Darstellen. Literale beinhalten keine weiteren Verweise oder Weiterleitungen. Im Kontext von RDF kommen Literale nur als Objekte vor, da diese keinen neuen Tripel bilden können.

Mapping

Mapping beschreibt die Formatierung eines Datensatzes auf ein anderes formales Schema.

Metadaten

Metadaten enhalten Informationen über die Merkmale anderer Daten, aber nicht diese Daten selbst.

NFA

Nichtfunktionale Anforderungen, d.h. Qualitäten, die das Softwareprodukt erfüllen muss. Bspw. “Komme in endlicher Zeit zu einem Ergebnis”

Ontologie

Eine Ontologie ist ein Netzwerk aus Begrifflichkeiten und den zwischen ihnen bestehenden Beziehungen. Sie dient dem Datenaustausch zwischen Anwendungen. Eine formale Sprache zur Beschreibung von Ontologien ist zum Beispiel das RDF-Schema.

RDF

Resource Description Framework (kurz RDF) gilt als grundlegender Baustein des Semantic Webs zur Beschreibung von Ressourcen und deren Metadaten bestehend aus Tripeln der Form (Subjekt, Prädikat, Objekt).

RDFUnit

Ein Unittest-System für RDF-Daten das prüft ob diese einer Ontologie entsprechen[3].

Ruby

Ruby ist eine interpretierte Programmierpsrache die viele Paradigmen unterstützt allerdings mehr auf Objektorientierung ausgelegt ist.[4]

Ruby sollte zur Entwicklung unseres Web-Backends[5] sowie der Kernanwendung geeignet sein[6]. Ruby funktioniert gut mit verbreiteten CI-Systemen wie Travis CI[7] und verfügt über ein Unittest Framework[8].

Ruby eignet sich auch für die Entwicklung des Webinterfaces da es einige Web Application Frameworks, zb. Ruby on Rails, bereitstellt, mit denen man Webanwendungen erstellen bzw. Ruby Code in HTML Dokumente integrieren kann siehe eRuby[9]. Die bietet den Vorteil sich bei Webanwendungen nur mit einer Programmierpsrache auseinander zu setzten.

Große Library welche, zb. bereits schon entwickelte Pakete für den Umgang mit SPARQL und RDF liefert.

Ruby on Rails

Ist ein Web Application Framework von Ruby[10].

Es bezieht sich sehr stark auf das MVC Konzept welches die Entwicklung von Modell, View und Controller trennbar machen kann und somit eine leichte Änderbarkeit der einzelnen Schichten möglich macht.

Benutzt Bundler als Paketverwaltung um die breite Library von Ruby einfach zu integrieren und die Pakete auf dem aktuellen Stand zu halten. Desweiteren Pakete die nur zur Entwicklung gebraucht werden von dem fertigen publizierten Framework zu trennen (zb. Entwickulngsdatenbank und Datenbank des Webservers).

Benutzt Scaffolding und erleichtert somit den Aufbau von Webapplikationen.

Semantic Web

Das Semantic Web macht Daten zwischen Rechnern einfacher austauschbar und einfacher verwertbar. Verknüpfungen zwischen Objekten die dem Menschen durch den Kontext klar werden, werden im Semantic Web auch den Maschinen zugänglich gemacht.

SPARQL

SPARQL (SPARQL Protocol and RDF Query Language) ist eine graphenbasierte Abfragesprache für RDF.[11]

Tripel-Store

Ein Tripel-Store ist eine Datenbank in der die Daten in Form von Tripeln (Subjekt,Prädikat,Objekt) gespeichert sind und für Abfragen zur Verfügung stehen.

Unittests

Unittests sind Testprogramme die kleine Einheiten (‘units’, zum Beispiel einzelne Klassen) des Projekts auf ihre Formale richtigkeit mithilfe von Testfällen prüfen.

Je nach wahl der Programmiersprache können Unittest-Werkzeuge entweder teil der Sprache selbst oder ihrer Standandbibliothek sein oder als externes Unittest-framework verfügbar sein.

URI

Ein Uniform Resource Identifier (kurz URI) ist eine Zeichenfolge die eine abstrakte oder physische Ressource identifiziert.[12] Zum Beispiel gehören dazu Dateien oder Webseiten.

URL

Ein Uniform Resource Locator (kurz URL) ist eine Unterart der URIs und identifiziert und lokalisiert eine Ressource.[13]

Webanwendung

Eine Webanwendung ist ein Anwendungsprogramm, das in einem Webbrowser angezeigt und bedient wird. Sie liegt auf einem Webserver, auf den über ein Netzwerk zugegriffen werden kann und dann von einem Klienten geladen wird.

Wetd16

Der Name des Projekts und der Projektgruppe, die sich mit dieser Thematik befasst.

Fußnoten

1 http://wiki.dbpedia.org/

2 https://tools.ietf.org/html/rfc3987

3 https://github.com/AKSW/RDFUnit

4 https://www.ruby-lang.org/en/about/

5 http://guides.rubyonrails.org/getting_started.html

6 https://rubygems.org/gems/sparql/versions/1.99.0

7 https://docs.travis-ci.com/user/languages/ruby/

8 https://en.wikibooks.org/wiki/Ruby_Programming/Unit_testing

9 https://de.wikipedia.org/wiki/ERuby

10 https://de.wikipedia.org/wiki/Ruby_on_Rails

11 http://www.w3.org/TR/sparql11-query/

12 https://tools.ietf.org/html/rfc3986

13 https://tools.ietf.org/html/rfc3986#section-1.1.3