Rechercheberichт

dbp17
Provenance Tracking von Fakten in DBpedia

Teilnehmer

Paul Reinhardt
Julius Kluge
Daniel Pohl
Michelle Kampfrath
Nicole Timme
Tobias Krawetzke
Stefan Walter

Betreuer

Hans-Gert Gräbe
Kay Müller
Johannes Frey
Marvin Hofer

Inhaltsverzeichnis

1 Begriffe

1.1 DBpedia

Die DBpedia-Community beschreibt sich selbst als Semantic Web Spiegelserver von Wikipedia. Ihr Hauptanliegen ist es strukturierte Daten von Wikipedia zu extrahieren, diese in ein RDF-Modell zu überführen und im Internet frei zugänglich zu machen. Die gesammelten Daten können über die folgende Website eingesehen werden.

http://dbpedia.org/page/Siemens

Wobei für Siemens jeder Wikipediaseitenname verwendet werden kann.

1.2 Ontologie und Ontologiesprache

Ontologien sind formale Beschreibungen von Konzepten innerhalb einer Wissensdomäne. Hauptziel ist es Wissensdomänen in maschinenlesbarer Form zu modellieren, um so eine Schnittstelle für agentenbasierte Softwaresysteme zu haben, die auf dieses Wissen zurückgreifen, um Aufgaben mit Hilfe von automatisiertem Schließen zu lösen.

Eine Ontologiesprache ist ein formales System zur Beschreibung von Ontologien. Es gibt eine Vielzahl solcher Wissensrepräsentationssprachen, im Bereich des Semantik Web werden häufig die Web Ontology Language kurz OWL oder das Resource Description Framework Schema kurz RDFS verwendet.

Der grundlegende Aufbau einer Ontologie ist unabhängig von der verwendeten Sprache und ergibt sich aus den folgenden drei Hauptelementen.

- **Instanzen** stellen physische oder abstrakte Objekte der realen Welt dar. Beispiele dafür sind Personen oder Abbildungen.
- **Klassen** repräsentieren eine Gruppe von verschiedenen Instanzen, welche gemeinsame Eigenschaften besitzen.
- **Eigenschaften** sind die Attribute einer Klasse. Dies können sowohl andere Klassen, wie zum Beispiel die Person des Vaters oder der Mutter, als auch beschreibende Attribute, etwa der Name der Person sein.
- **Axiome** postulieren Zusammenhänge von Klassen, Eigenschaften und Instanzen

1.3 DBpedia-Ontologie

Die DBpedia-Ontologie wird nach der OWL-Spezifikation erstellt und enthält Klassen, DatentypEigenschaften und ObjectEigenschaften als dafür typische Sprachkonstrukte. Die einzelnen Sprachkonstrukte werden von dem DBpedia-Mappings-Wiki mittels Templates erzeugt. Der Aufbau dieser Templates ist stark an der Template-Struktur der Wikipedia-Infoboxen angeglichen. Dadurch ist ein einfaches überführen der Infobox-Daten in die DBpedia-Ontologie ermöglicht. DBpedia stellt auf der folgenden Seite alle bisher definierten Ontologieklassen zur Verfügung.

http://mappings.dbpedia.org/server/ontology/classes/

1.4 Framework

Ein Framework ist ein unvollständiges Programm, welches für Anwendungen eine wiederverwendbare, gemeinsame Struktur zur Verfügung stellt. Ein Entwickler baut das Framework in die eigene Anwendung ein und erweitert es derart, dass es spezifischen Anforderungen entspricht. Ein Beispiel ist das OpenGL Framework GLFW, welches neben etlichen Hilfsfunktionen, auch die grafische Oberfläche in Form eines Fenster stellt, auf welcher die einzelnen Pixel grendert werden. Frameworks werden vor allem im Bereich der objektorientierten Softwareentwicklung sowie bei komponentenbasierten Entwicklungsansätzen verwendet.

1.5 DBpedia Extraction Framework und DBpedia Extractors

Das DBpedia Extraction Framework ist ein flexibles und erweiterbares Framework um von einzelnen Wikipediaseiten strukturierte Informationen zu extrahieren und in die DBpedia-Ontologie zu überführen. Für die eigentliche Funktionsweise und Verwendung des Framework sei auf die Folgende Dokumentation verwiesen.

http://mappings.dbpedia.org/server/extraction/en

Ein Teil des Framework wird von dem InfoboxExtractor ausgemacht, dessen Spezielle Aufgabe darin besteht alle Daten aus den Infoboxen zu extrahieren. DBpedia bietet auf der folgenden Seite ein Onlinetool für die Extraction dieser Daten. Die Ausgabe erfolgt in Form von verschieden RDF-Darstellungen.

http://mappings.dbpedia.org/server/extraction/en

Ein anderer Teil des Framework besteht aus dem MappingExtractor. Dieses Extractor ermöglicht es, mittels manuell angelegter DBpedia-Mappings, vorher festgelegte Daten aus den Infoboxen zu extrahieren. Diese DBpedia-Mappings werden durch das DBpedia Mappings Wiki verwaltet. Dieses Wiki ist auf der folgenden Website zu finden.

http://mappings.dbpedia.org/index.php/Main_Page

1.6 API bzw. Programmierschnittstelle

Eine Programmierschnittstelle ist ein Programmteil, der von einem Softwaresystem anderen Programmen zur Anbindung an das System zur Verfügung gestellt wird.

1.7 MediaWiki WebAPI

Die MediaWiki WebAPI bietet einen einfachen Zugriff auf Wiki-Funktionen, Daten und Metadaten von Wikipediaseiten. Clients fragen bestimmte "actions" , mithilfe des *action* Parameters ab um Informationen zu erhalten. Zum Beispiel kann die folgende URL dazu verwendet werden die ersten 500 Revisionsids, der englischen Wikipediaseite von Siemens zu erhalten.

https://en.wikipedia.org/w/api.php?action=query&prop=revisions&titles=Siemens&rvlimit=max&rvprop=timestamp|user|idshttps://en.wikipedia.org/w/api.php?action=query\&prop=revisions\&titles=Siemens\&rvlimit\newline=max\&rvprop=timestamp|user|ids

Eine genauere Beschreibung der angebotenen Funktionalitäten, ist auf der folgenden Website zu finden.

https://en.wikipedia.org/w/api.php?action=help&modules=main

1.8 Provenienz Tracking

Informationen über Entitäten, Aktivitäten und Personen, die an der Erzeugung oder Veränderung von Daten beteiligt sind werden in der Informatik als Provenienz bezeichnet.

Beim Provenienz Tracking werden diese Informationen gesammelt und ausgewertete um zum Beispiel die Qualität und die Vertrauenswürdigkeit der Daten zu bewerten.

1.9 URI und Fragment-URI

URI steht für Uniform Resource Identifier. Dieser Identifikator besteht aus einer Zeichenfolge, die zur Identifizierung einer abstrakten oder physischen Ressource dient. URIs werden unter anderem zur Bezeichnung von Websiten im Internet und dort vor allem im Wolrd Wide Web eingesetzt.

Ein Fragmentbezeichner wird einer URI hinzugefügt um lokal Teile eines Dokuments zu addressieren. Die Interpretation ist abhängig von der Art der Ressource und dem Parser. Der Fragmentbezeichner wird mit einer Raute # in der URI gekennzeichnet. Wie im folgendem Beispiel zu sehen.

http://example.com/document.html#anker1"

Es wird in "document.html" auf das HTML-Element verwiesen welches das Anker-Attribut name="anker1" enthält.

2 Konzepte

2.1 Semantisches Web

Das semantische Web ist eine Erweiterung des World Wide Web, bei der Informationen und Daten zueinander in Beziehung gesetzt werden. Maschinen oder Computer sollen diese Beziehungen auswerten und Informationen im Zuge einer Abfrage sinnvoll filtern.

Dies geschieht indem Inhalte mit weiterführenden Informationen verknüpft werden, welche ihrerseits weiter verlinkt sind. Durch dieses Vorgehen entsteht ein Graph welcher die gesamten Daten des Word Wide Web verknüpft. Um die Machinenlesbarkeit zu realisieren werden die Daten nach bestimmten Standards Veröffentlicht. Die in diesem Kontext wichtigstes Spezifikation ist RDF.

2.2 XML

XML steht für Extensible Markup Language und ist ein Standard [6] für eine erweiterbare Auszeichnungssprache. Er definiert ein text-basiertes Format zur Darstellung und zum Austausch von meist hierarchisch strukturierten Daten.
XML enthält Elemente, die durch Start- und End-Tags der Form <element–name> und </element–name> voneinander abgegrenzt werden, welche paarweise verschachtelt sind. Dabei gibt es ein ausgezeichnetes Wurzel-/Root Element. Die Elemente können Attribute besitzen, die in folgender Form gekennzeichnet werden: <frequenz einheit="Hz">...</frequenz>. Darüber hinaus sind in der Spezifikation weit mehr Einzelheiten festgelegt. Ein Beispiel für ein XML-Dokument ist

```
<?xml version="1.0"?>
<catalog>
        <book id="bk101">
                <author>Gambardella, Matthew</author>
                <title>XML Developer's Guide</title>
                <genre>Computer</genre>
                <price>44.95</price>
                <publish_date>2000–10–01</publish_date>
                <description>An in–depth look at creating applications with XML.</description>
        </book>
</catalog>
```

Mithilfe von z.B. der Document Type Definition (DTD) oder einem XML Schema (XSD) kann teilweise innerhalb der allgemeinen Regeln eine eigene Struktur bzw. Syntax definiert werden, was XML erweiter- und anpassbar macht.

2.3 RDF

Das Resource Description Framework ist ein Datenmodell, welches als Ontologie modelliert ist und Ursprünglich entwickelt wurde um Metadaten zu verarbeiten. Mittlerweile gilt RDF als ein grundlegender Baustein des Semantischen Webs und wird Beispielsweise zum verbessern von Such-Engines oder zur Beschreibung von Beziehungen zwischen Webseiten verwendet.

Die semantischen Elemente des RDF-Modell entsprechen denen einer Ontologie, wobei Instanzen als Ressourcen bezeichnet werden. Auf syntaktischer Ebene werden diese Elemente zu einfachen Aussagen in Tripel-Darstellung kombiniert.

$$< \text{Subject} >< \text{Predicate} >< \text{Object} >$$

Dabei entspricht jedes Tripel einer Relation zwischen dem Subjekt und dem Objekt. Dadurch sind auch die ontologischen Axiome in das RDF-Modell integriert. Außerdem beschreibt das RDF-Modell einen Graph mit Subjekten und Objekten als Knoten bzw. Blätter und Prädikaten als Kanten.

2.4 RDF/XML

RDF/XML ist eine mögliche Spezifikation für eine konkrete RDF Syntax, bei welcher die Kanten und Knoten mittels URIs zu identifizieren werden. Diese Variante wird auch von DBpedia verwendet. Für eine genauere Spezifikation sei auf die Website

https://www.w3.org/TR/rdf-syntax-grammar

verwiesen. Im folgenden sei die Anwendung an einem Beispiel aufgezeigt.

```
1  <?xml version="1.0" encoding="utf-8" ?>
2  <rdf:RDF
3          xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4          xmlns:dbp="http://dbpedia.org/property/"
5          xmlns:dbo="http://dbpedia.org/ontology/">
6
7          <rdf:Description rdf:about="http://dbpedia.org/resource/Arena_Leipzig">
8              <dbo:location rdf:resource="http://dbpedia.org/resource/Leipzig" />
9              <dbp:location rdf:resource="http://dbpedia.org/resource/Leipzig" />
10         </rdf:Description>
11 </rdf>
```

Zeile 1 spezifiziert das Dokument als XML-Dokument. Der RDF-Graph beginnt in Zeile 2 und endet in Zeile 11. In den Zeilen 3-5 werden Abkürzungen für verschiedene Namensbereiche festgelegt, welche durch die unterschiedlichen URIs realisiert sind. Sinngemäß werden dadurch verschiedene Vokabulare repräsentiert. Das Schlüsselwort "about" in Zeile 7 setzt die URI des Subjektes das Schlüsselwort "resource" in den Zeilen 8 und 9 setzt die URI des Objektes. Die Sequenz "dbo:location" zeigt das dass Prädikat "location" dem DBpedia-Ontologie-Vokabular entnommen wird. Übersetzt bedeutet Zeile 8 die URIs "http://dbpedia.org/resource/Arena_Leipzig" und "http://dbpedia.org/page/Leipzig" befinden sich im dem dbo-Vokabular in der Relation "location". Das heißt ein RDF-Parser für das dbo-Vokabular würde das folgende Tripel erzeugen.

```
<http://dbpedia.org/resource/Arena_Leipzig><http://dbpedia.org/ontology/location>
<http://dbpedia.org/resource/Leipzig>
```

Analog würde ein RDF-Parser für das dbp-Vokabular das folgende Tripel liefern.

```
<http://dbpedia.org/resource/Arena_Leipzig><http://dbpedia.org/property/location>
<http://dbpedia.org/resource/Leipzig>
```

Durch die "Description" Anweisung in Zeile 7 wird dem RDF-Graph ein neuer innerer Knoten zugefügt.

2.5 Turtle

Turtle (Terse RDF Triple Language) ist ein für Menschen gut lesbares Format der RDF-Triple. Bei dieser Formatierung werden zu Beginn sogenannte Prefixes für Namensräume definiert, welche diese dann im Dokument ersetzen. Dazu wird ein URI in der Regel bis einschließlich des letzten / und ein Fragment-URI bis einschließlich der # abgekürzt. Der dahinter stehende Rest des URIs wird nicht abgekürzt und im Dokument an das Präfix angehängt. Zur Festlegung eines Prefixes wird dies mit dem Ausdruck "@prefix" eingeleitet, worauf das definierte Prefix mit einem Doppelpunkt folgt. Das Prefix kann nach Belieben frei definiert werden, allerdings haben sich für gängige Vokabulare unverbindliche Standard-Prefixes durchgesetzt. Die Funktionsweise sei hier nur an einem Beispiel demonstriert.

```
1  @prefix ex: <http://example.org/> .
2  @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
3  @prefix foaf: <http://xmlns.com/foaf/0.1/> .
4
5  ex:max_mustermann rdf:type foaf:Person .
6
7  <http://example.org/max_mustermann><http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
8  <http://xmlns.com/foaf/0.1/Person>
```

In den Zeilen 1-3 bis drei werden Präfixes festgelegt, wobei "rdf" und "foaf" auch Beispiele für die anfangs erwähnten Standard Vokabulare sind. Zeile 5 zeigt die Kurzform von den Zeilen 7 und 8. Eine vollständige Spezifikation ist auf der folgenden Website zu finden.

https://www.w3.org/TR/turtle/

2.6 SPARQL

SPARQL steht für SPARQL Protocol and RDF Query Language und ist eine graphenbasierte Abfragesprache für RDF. DBpedia stellt dafür auf der folgenden Webseite eine Onlineschnittstelle zur Verfügung.

http://dbpedia.org/snorql

Deren Verwendung hier an einer kleine Beispielanfrage gezeigt werden soll. "Alle Werke von Leonardo da Vinci mit Titel und deren länderspezifischen Namen".

```
PREFIX dbpediaO: <http://dbpedia.org/ontology/>
SELECT ?title ?lable WHERE{
?work dbpediaO:author :Leonardo_da_Vinci.
?work dbpedia2:title ?title.
?work rdfs:label ?lable.}
```

Um sich bei der Anfrage Zeit zu sparen und die Leserlichkeit zu verbessern, kann am Anfang der Abfrage ein PRÄFIX erstellt werden um die URI zu verkürzen. Auf der oben angegebenen Website sind schon einige PRÄFIXe vorgegeben, welche gegebenenfalls verwendet werden können. Hier wird noch dbpediaO: <http://dbpedia.org/ontology/ hinzugefügt.

Die explizite Abfrage beginnt dann nach SELECT. Variablen werden mit vorangestelltem '?' (oder '$') gekennzeichnet und als Ergebnis bekommt man alle möglichen Belegungen der Variablen in Tabellenform zurückgegeben. Unsere Variablen sind ?title und ?lable.

In dem WHERE Bereich werden die Variablen belegt. Wir definieren die Variable ?work als alle von Leonardo da Vinci geschriebene Werke. Nun erstellt man die Tripel für die Abfragen von ?title und ?lable. (Subjekt, Prädikat, Objekt). Bei ?title werden die Titel (dbpedia2:title) aller Werke (?work) von Leonardi da Vinci ausgegeben. Bei ?lable werden die länderspezifischen Namen (rdfs:label) aller Werke (?work) ausgegeben.

3 Aspekte

3.1 Aktueller Stand und Lösungsansatz

Bisweilen beachtet DBpedia nicht die Aktualität der Wikipedia-Daten. Dadurch ergeben sich Unzuverlässigkeiten welche zum Teil vermeidbar sind. Ändert sich zum Beispiel die Mitarbeiterzahl in einem Unternehmen normalerweise jeden Monat aber seit längerer Zeit nicht mehr, so lässt sich auf Nichtaktualität schließen.

Eine naheliegende Möglichkeit dieses Problem zu umgehen ist ein provenienzielles Tracking der einzelnen Fakten um die zeitlichen Änderungen zu analysieren.

3.2 Ziel

Ziel dieses Projektes ist die Entwicklung einer zum DBpedia Extraction Framework parallelen Anwendung, welche ausgehend von einer Wikipediaseite mittels der MediaWiki WebAPI die alten Revisionsnummern ermittelt. Anhand dieser Nummern werden mittels des DBpedia Extraction Framework die zugehörigen RDF Tripel erzeugt und miteinander verglichen. Falls sich die Tripel von je zwei aufeinander folgenden Revisionen unterscheiden werden die zu den Fakten gehörenden Änderungsdaten für spätere Auswertungen zwischengespeichert.

3.3 Mögliche Erweiterungen

Eine Möglichkeit die Laufzeit der oben beschriebenen Grundanwendung zu verbessern, ist die Vorfiltrierung der Revisionen. Das heißt es werden mittels der MediaWiki WebAPI Informationen aus den Revisionen abgefragt, welche Rückschlüsse auf etwaige Änderungen der Infoboxdaten zu lassen. Falls es keine Änderungen gibt wird diese Revision von der Anwendung nicht weiter beachtet.

Die Auswertung der gesammelten Metadaten gibt Potential für eine zusätzliche Erweiterung der Grun-

danwendung.

3.4 Anmerkungen und Hinweise

Wikipedia gestattet zunächst nur den Zugriff auf die letzten 500 Revisionen so dass dadurch die Revisionstiefe vorerst beschränkt ist. Registrierte Bots erhalten Zugriff auf die letzten 5000 Revisionen.

3.5 Programmiersprache

Die Zielstellung lässt sich gut objektorientiert modellieren Beispielweise können das einlesen einer URL und die formatierte Zwischenspeicherung des Inhaltes, die Schnittstelle zu dem DBpedia Extraction Framework oder Textformatierung als einzelne Komponenten beschrieben werden. Da es sich bei dem Softwarepaket um eine Parallele Anwendung, welche nur die Framework-Schnittstelle benötigt, handeln wird, ist grundsätzlich jede objektorientierte Programmiersprache verwendbar. Für dieses Projekt fällt die Wahl v.a. aufgrund der bisherigen Programmiererfahrung der Projektteilnehmer auf Java, obwohl das Framework in Scala geschrieben ist. Weiterhin spricht folgendes dafür

- Java unterstützt durch integrierte Stream-Methoden standardmäßig dass effiziente lesen aus URLs und verschiedenen Textdateiformaten sowie das schreiben in Files oder auch die Konsole.
- Java unterstützt das Client-Server Paradigma, wodurch ein einfaches erstellen von Webanwendungen ermöglicht ist.
- Java bietet Möglichkeiten zur Erzeugung und Verarbeitung von regulären Ausdrücken
- Java ermöglicht das einfache erstellen von dialogbasierten Graphik User Interfaces.
- Java dient der Erzeugung plattformunabhängiger Software, wodurch Indirekt das Ziel der DBpedia-Community, die Informationen von Wikipedia in ein RDF-Graph zu überführen und diese semantisch strukturierte Form für möglichst viele Menschen frei zugänglich zu machen, unterstützt wird.

Das Big-Data-Paradigma wird von der Java Standardspezifikation nicht mit unterstützt. Dennoch existieren für diesen eventuell auftretenden Nachteil externe Lösungen in Form von Big Data Frameworks wie zum Beispiel Apache Spark, siehe dazu die folgende Website

https://spark.apache.org

Quellen

[1] URL: http://dbpedia.org/page/Siemens (besucht am 19. 12. 2016).

[2] URL: http://mappings.dbpedia.org/server/ontology/classes/ (besucht am 19. 12. 2016).

[3] URL: https://en.wikipedia.org/w/api.php?action=query&prop=revisions&titles=Siemens&rvlimit=max&rvprop=timestamp|user|ids (besucht am 19. 12. 2016).

[4] URL: https://en.wikipedia.org/w/api.php?action=query\&prop=revisions\&titles=Siemens\&rvlimit\newline=max\&rvprop=timestamp|user|ids (besucht am 19. 12. 2016).

[5] URL: https://en.wikipedia.org/w/api.php?action=help&modules=main (besucht am 19. 12. 2016).

[6] URL: https://www.w3.org/TR/xml/ (besucht am 19. 12. 2016).

[7] URL: https://www.w3.org/TR/rdf-syntax-grammar (besucht am 19. 12. 2016).

[8] URL: http://dbpedia.org/resource/Arena_Leipzig (besucht am 19. 12. 2016).

[9] URL: http://dbpedia.org/page/Leipzig (besucht am 19. 12. 2016).

[10] URL: https://www.w3.org/TR/turtle/ (besucht am 19. 12. 2016).

[11] URL: http://dbpedia.org/snorql (besucht am 19. 12. 2016).

[12] URL: http://dbpedia.org/ontology/ (besucht am 19. 12. 2016).

[13] URL: https://spark.apache.org (besucht am 19. 12. 2016).