Provenance Tracking von Fakten in DBpedia

Arbeitsplan

Betreuer:

Betreuer:	Johannes Frey
Betreuer:	Kay Müller
Tutor:	Marvin Hofer

Team:

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

Leipzig, am 16. Januar 2017

Inhaltsverzeichnis

1 Projektvision **3**

2 Voraussetzungen **3**

3 Designübersicht und Funktionalität **3**
3.1 Designübersicht 3
3.2 Funktionalitäten 4
3.2.1 Funktionale Anforderungen 4
3.2.2 Nichtfunktionale Anforderungen 4
3.3 Nutzerszenarien 4
3.3.1 Use Case 1 4
3.3.2 Use Case 2 4
3.3.3 Use Case 3 5

4 Arbeitspakete **5**

5 Vorprojekt **6**

6 Glossar **7**

1 Projektvision

Es soll ein leichtgewichtiges und paralleles Konsolenprogramm entwickelt werden, welches die aktuellen Fakten aus DBpedia über eine Entität (z.B. dass eine Person im Vorstand einer Firma ist) auf Basis der Versionshistorie (komplette Revisionen inkl. Metadaten) von Wikipedia zurückverfolgt und somit ermittelt, wann und von wem ein Fakt in Wikipedia eingetragen worden ist.

Darüber hinaus sollen aus diesen Metadaten Informationen über die Vertrauenswürdigkeit der Fakten berechnet werden, indem beispielsweise ermittelt wird, wie oft diese gesichtet worden sind oder wie dynamisch sich diese in der Vergangenheit verändert haben.

2 Voraussetzungen

Um die Applikation den Anforderungen entsprechend ausführen zu können, müssen folgende Voraussetzungen erfüllt werden:

- Java-Laufzeitumgebung. Da die Applikation in Java umgesetzt wird, benötigt die ausführende Komponente eine Java-Laufzeitumgebung.
- leistungsstarkes System. Da die Anwendung mit sehr großen Datenmengen arbeitet, ist es notwendig, dass über eine ausreichende Serverleistung verfügt werden kann, damit nötige Anfragen in akzeptabler Zeit bearbeitet werden können.
- Weiterhin kann davon ausgegangen werden, dass während des Implementierungsprozesses weitere Voraussetzungen entdeckt und hinzugefügt werden, da in der aktuellen Phase noch keine genaueren Angaben gemacht werden können.

Im Team müssen weiterhin folgende Voraussetzungen erfüllt werden:

- Installation und Eintreten in Slack und der entsprechenden Projektgruppe.
- Installation von Git und Eintreten in das enstprechende Projekt

3 Designübersicht und Funktionalität

3.1 Designübersicht

Da das Tool nicht für den regelmäßigen Gebrauch konzipiert ist und hauptsächlich von DBpedia benutzt wird, verzichten wir bewusst auf ein Benutzerinterface. Ein einfaches Webinterface könnte allerdings für das Debugging implementiert werden, um zu Testzwecken einzelne Abfragen stellen zu können.

Im Programm werden zuerst die relevanten Revisionsnummern aus dem Wikipediadumps extrahiert. Anschließend werden diese Revisionsnummern an das Extraction-Tool übermittelt, um die RDF-Graphen zu erzeugen. Mithilfe der RDF-Daten werden nun Differenzen erkannt und gespeichert, sodass diese später interpretiert werden können.

3.2 Funktionalitäten

3.2.1 Funktionale Anforderungen

FA10 Benutzung durch Konsole

FA20 Revisionen für Artikel verfügbar machen

FA30 Aus verfügbar gemachten Revisionen Revisionsnummern extrahieren

FA40 RDF-Graphen anhand der Revisionsnummer bereitstellen

FA50 Triple Differenzen zuverlässig erkennen

FA60 Erkannte Triple Differenzen speichern und somit abrufbar machen.

FA70 Parallelverarbeitung nutzen um auf große Datenmengen effektiv anwendbar zu sein

3.2.2 Nichtfunktionale Anforderungen

NA10 Webinterface um einfacheres Testen zu ermöglichen

NA20 Herausfiltern von Revisionen ohne relevante Änderungen

NA30 Implementierung von Parametern um auf den konkreten Anwendungsfall anpassbar zu sein

NA40 Aus den generierten Daten auf weiteres wie z.B. Zuverlässigkeit von Fakten schließen

NA50 Absturzsicherung, wie z.B. durch ein Log

3.3 Nutzerszenarien

3.3.1 Use Case 1

Beschreibung: Der Nutzer möchte die Provenienz für die gesamte DBPedia erhalten

Erforderliche Funktionalität: Aufgrund der großen Datenmengen die hier verarbeitet werden müssen muss das Tool effektiv arbeiten um in endlicher Zeit zu terminieren. Dafür sollte Parallelverarbeitung genutzt werden. (siehe FA70)

3.3.2 Use Case 2

Beschreibung: Der Nutzer möchte für einzelne Fakten die Provenienz erhalten

Erforderliche Funktionalität: Damit das Programm hier sowie in Use Case 1 anwendbar ist, müssen Parameter implementiert werden die die Anwendung des Programms einschränken können. (siehe NA30)

3.3.3 Use Case 3

Beschreibung: Der Nutzer möchte die Provenienz aus einem bestimmten Zeitraum erhalten

Erforderliche Funktionalität: Um das Programm auf diesen Fall anpassbar zu machen, müssen Parameter implementiert werden um die extrahierten Revisionsnummern aus einem bestimmten Zeitraum extrahieren zu können. (siehe NA30)

4 Arbeitspakete

4.1 Vorprojekt und Einarbeitung in benötigte Hilfsmittel (30%)

Dieses Arbeitspaket beinhaltet mehrere Teile. Zum einen die weitere Konzipierung und Implementierung des in Punkt 5 beschriebenen Vorprojekts. Des weiteren die Installation benötigter Hilfsmittel, wie zum Beispiel des Extraction Frameworks, sowie die Einarbeitung in diese.

4.2 Revisionsnummern extrahieren (30%)

In diesem Arbeitspaket wird der Teil des fertigen Tools erstellt, welcher die Revisionsnummern aus den Wikipediadumps extrahiert und für die weitere Verarbeitung verfügbar macht. Eine Teilaufgabe davon ist auch die Wahl eines geeigneten XML-Parsers, der auch große Dateien wie die späteren Wikipedia-Dumps verarbeiten kann.
Möglichkeiten wären z.B. FasterXML Jackson[1] oder diverse in der Java Standardbibliothek enthaltene Parser wie SAX oder StAX[2].

4.3 Provenienzerzeugung und Speicherung (15%)

In diesem Arbeitspaket wird der Teil des fertigen Tools erstellt, welcher für die vorher ermittelten Revisionsnummern mit Hilfe des Extraction Frameworks die RDF-Graphen erzeugt, und dann Differenzen zwischen den Tripeln findet und verfügbar macht.

4.4 Implementierung (10%)

In diesem Arbeitspaket werden die in Paket 2 und 3 entstandenen Programmteile zusammengefügt und die grundsätzliche Funktionalität hergestellt.

4.5 Effizienzoptimierung (10%)

In diesem Arbeitspaket soll das in Paket 4 erzeugte funktionierende Programm auf Effizienz untersucht und an den Stellen, an denen dies möglich ist, optimiert werden. So wird in diesem Schritt dafür gesorgt, dass das Programm möglichst parallel arbeiten kann.

[1] `https://github.com/FasterXML/jackson-dataformat-xml`
[2] `https://docs.oracle.com/javase/tutorial/jaxp/index.html`

4.6 Debugging und Feinschliff (5%)

In diesem Arbeitspaket soll das Kernprodukt fertiggestellt und zur Auslieferung bereit gemacht werden. Falls NA10 umgesetzt werden soll, muss das spätestens vor diesem Arbeitspaket erfolgt sein.

4.7 Funktionalität erweitern (Bonus/20%)

In diesem Arbeitspaket werden (falls gegen Ende des Projekts noch Zeit übrig ist) teilweise oder komplett FA70 bis NA50 umgesetzt.

5 Vorprojekt

Als Vorprojekt soll eine Anwendung in Java 1.8 geschrieben werden, welche Provenienz-Daten erzeugt. Als Provenienz-Daten sollen im Vorprojekt der Zeitpunkt der letzten Aktualisierung eines Triples und der Autor dieser Aktualisierung ermittelt werden. Diese Daten werden für mehrere geeignete, also z.B. eine umfangreiche und eine kleine, Wikipedia Seiten erzeugt. Als Eingabe erhält die Anwendung dann den relativen Dateipfad der XML-Datei der jeweiligen Wikipedia Seite, in der die Revisionen vorliegen. Diese XML-Datei wird vorher manuell heruntergeladen und bereitgestellt. Die Ausgabe der ermittelten Daten soll als Tabelle in `.tsv` Form erfolgen.

Im Rahmen dieses Vorprojekts werden also die Funktionalitäten FA30, FA40 und FA50 mindestens teilweise implementiert.

6 Glossar[3]

DBPedia Semantic Web Spiegelserver von Wikipedia in RDF-Format.

RDF Das Resource Description Framework ist ein Datenmodell, welches als Ontologie modelliert ist.

Parallelverarbeitung[4] Parallelverarbeitung beschreibt die simultane Bearbeitung mehrerer Befehlsteile, Befehle oder Programmteile durch eine Zentraleinheit.

Webinterface[5] Als Webinterface bezeichnet man eine Schnittstelle zu einem System, die über das Hypertext Transfer Protocol (HTTP) angesprochen werden kann.

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

(DBPedia) Extraction Framework 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.

[3] für nicht explizit angegebene Quellen: alle Begriffe sind im Recherchebericht enthalten
[4] `http://wirtschaftslexikon.gabler.de/Definition/parallelverarbeitung.html`
[5] `https://de.wikipedia.org/wiki/Webinterface`