Softwaretechnik-Praktikum 2016
Datum: 18. Januar 2016

Gruppe: wrd16
Betreuer: René Speck
Tutor: Marius Brunnert

Team:
Tim Niehoff
Felix Lange
Philip Fritzsche
Matti Wilhelmi
Simon Kaleschke
Johannes Leupold

Arbeitsplan

Inhaltsverzeichnis

1 Projektvision

Es soll eine Anwendung implementiert werden, um mit einem Wissensextraktionsframework unstrukturierten Text aus einem vorgegebenen Korpus[1] durch implementierte Interfaces von Named Entity Recognition (NER)- und Relation Extraction (RE)-Tools zu annotieren. Als Ergebnis werden Resource Description Framework (RDF)-Tupel mit Linked Open Data Attributen ausgegeben. Zum Ablegen von Metadaten wird die PROV Ontologie verwendet. Das Programm soll in der Lage sein, sowohl deutsche als auch englische Texte zu analysieren. Zudem soll die Austauschbarkeit der genutzten NER- und RE-Tools angeboten werden.

2 Voraussetzungen

Um unser Projekt später verwenden zu können, wird eine funktionierende Java-Laufzeitumgebung benötigt, da wir es in Java implementieren werden. Weiterhin ein richtig konfigurierter Webserver auf dem Apache Tomcat läuft, um die lokale Nutzung von FOX zu ermöglichen. Die lokale Nutzung wurde angestrebt, um dadurch eventuellen Geschwindigkeitseinbußen vorzubeugen, sowie um ständige Erreichbarkeit zu gewährleisten. Die Apache Lucene Bibliothek sowie der Index der Relationsmuster (BOA) wird außerdem in der Version 4.5 benötigt. Auf die Verwendung von Apache SOLR kann durch die lokale Nutzung verzichtet werden.

3 Designübersicht und Funktionalität

3.1 Designübersicht

Die Aufgabe der Gruppe ist es, ein Tool zu entwickeln, welches aus einem Text oder einer Menge von Texten (wie zum Beispiel eine Kopie der Wikipedia) Informationen im RDF-Format extrahiert. Dabei sollen existierende Tools verwendet werden, die auch austauschbar sein sollen. Zuerst wird ein NER-Tool verwendet, um Entitäten im vorliegenden Text zu markieren und anschließend wird der annotierte Text mit Mustern des BOA-Indexes abgeglichen, um RDF Daten zu erhalten. Unser Tool soll dabei Interfaces als Abstraktionsschichten implementieren, damit diese Tools (FOX und BOA) ausgetauscht werden können. Unser Tool greift nicht direkt auf FOX und BOA zu, sondern immer indirekt über diese Interfaces, welche von Entwicklern solcher Tools implementiert werden sollen. Wir implementieren diese nur für FOX und den BOA-Index. Unser Tool benötigt keine aufwendige Benutzeroberfläche, da nur die zu verarbeitenden Daten übergeben werden sollen und gegebenenfalls die zu verwendenden Tools (für NER und RE). Zu den entstandenen RDF-Daten wird mit der PROV-Ontologie Provenance gespeichert. Die Ausgabe der Daten wird ebenfalls durch ein Interface abstrahiert, wobei wir die Daten in einem einfachen RDF-Format ausgeben werden. Das Programm kann prinzipiell von mehreren Nutzern gleichzeitig genutzt werden, diese sind allerdings unabhängig voneinander, da für jeden Nutzer nur eine Datenmenge verarbeitet werden soll.

[1] siehe: https://datahub.io/de/dataset/dbpedia-abstract-corpus

3.2 Rollen

3.2.1 Benutzer

Wie bereits angemerkt ist unser Tool zwar für den Mehrbenutzerbetrieb geeignet, wenn die verwendeten Tools (für NER und RE) dies ebenfalls sind, allerdings sind die Benutzer generell voneinander unabhängig. Benutzer starten das Programm, übergeben die zu verarbeitenden Daten und erhalten die Ausgabe.

3.2.2 Entwickler

Entwickler müssen Interfaces für zusätzliche NER und RE implementieren, wenn nicht nur FOX und BOA verwendet werden soll oder die Daten auf andere Art ausgegeben oder abgelegt werden sollen. Entwickler können ebenso Benutzer des Tools sein.

3.3 Ziele

3.3.1 Muss-Ziele

/10/ Das Tool muss in einem vorliegenden Korpus Entitäten kennen. Dieser Korpus muss beliebig austauschbar sein können.

/20/ Das Tool muss Relationen zwischen den Entitäten erkennen und diese im RDF Format darstellen.

/30/ Diese Erkennung von Entitäten und Relationen soll mit Hilfe von bereits vorhandenen Tools erfolgen.

/40/ Die verwendeten Tools sollen austauschbar sein. Um dies zu garantieren, wird jedes Tool nie direkt verwendet, sondern immer über eine zusätzliche Abstraktionsschicht in Form eines Interfaces.

/50/ Für je ein Tool muss ein Interface implementiert sein. Wir verwenden FOX und BOA.

/60/ Zu den erzeugten Daten muss Provenance mittels PROV-O Ontologie gespeichert werden.

3.3.2 Kann-Ziele

/70/ Um große Datenmengen effektiver verarbeiten zu können, kann Parallelverarbeitung zum Einsatz kommen.

/80/ Es kann eine graphische Benutzeroberfläche oder eine Weboberfläche implementiert werden, um das Tool benutzerfreundlicher zu machen.

/90/ Zusätzlich zu FOX und BOA können Interfaces für andere Tools mitgeliefert werden.

/100/ Es können eigene Ansätze für RE implementiert werden.

4 Arbeitspakete

4.1 Einarbeitung in die Ressourcen (10%)

Jedes Teammitglied erarbeitet sich selbstständig ein Verständnis aller Ressourcen, die später im Projekt eingesetzt werden. Vor allem sind hierbei die zu verwendenden Tools FOX und BOA wichtig. Sobald dies geschehen ist, kann mit dem eigentlichen Projekt begonnen werden.

4.2 Einbindung der Tools (30%)

Die bereits zuvor genannten Tools werden erst einmal unabhängig voneinander lauffähig gemacht. (Das Named Entity Regconition Tool FOX hierbei auf einem eigenen Tomcat-server)

4.3 Ablegen der Provenance mittels PROV-O (15%)

Die Provenance der generierten Daten soll mit Hilfe der PROV-Ontologie abgelegt werden. Das PROV-Datenmodell enthält folgende Informationen:

- Zu welcher Zeit wurden die Daten erstellt?
- Aus welchen Daten wurden die Daten generiert?
- Welche Tools haben die Daten generiert?

Die PROV-Ontologie sieht vor, die betreffenden Daten im RDF-Format abzulegen. Diese Daten sollen von unserem Projekt an die generierten Entitäts- und Relationsdaten angehängt werden.

4.4 Erstellung eines Interface für die Kommunikation FOX/BOA (20%)

Da die Tools FOX und BOA von uns zwar beispielhaft verwendet werden sollen, aber auch durch andere NER und RE-Tools austauschbar sein sollen, befasst sich dieser Arbeitsschritt damit, ein Interface zu erstellen, dass dem Programm eine Kommunikation mit und zwischen den Tools erlaubt.

4.5 Erarbeitung mindestens einer Datenausgabe (15%)

Wenn die Tools laufen und auf den Datensätzen arbeiten, soll ein Interface erstellt werden, dass die im Programmablauf generierten Daten in einem computer- und menschenlesbaren Format ausgegeben und evtl. abgespeichert werden können.

4.6 Fertigstellung und Feinschliff (5%)

Sobald die bisherigen Punkte abgeschlossen sind, ist das Projekt in einem nahezu abgabefertigen Zustand. Dann kann das Projekt aber noch kleine Änderungen erfahren, die die Grundfunktionalität nicht einschränken, aber bspw. den Vorstellungen des Project Owners mehr entsprechen.

4.7 Kann-Ziele

Falls am Ende Zeit übrig ist, können noch die oben genannten Kann-Ziele (wie zum Beispiel eine graphische Oberfläche) umgesetzt werden.

5 Vorprojekte

Im Vorprojekt sollen die in den Arbeitspaketen 1 und 2 genannten Arbeitsschritte durchgeführt werden. So wird das Team bereits das Grundverständnis für das zu erstellende Programm haben. Außerdem arbeitet das so erstellte Vorprojekt bereits dem ganzen Projekt zu, so dass sich, ohne großen Aufwand, mit ihm weiterarbeiten lässt.

6 Glossar

6.1 Framework

Ein Framework (englisch für Rahmenstruktur) ist ein Programmiergerüst, das in der Softwaretechnik, insbesondere im Rahmen der objektorientierten Softwareentwicklung sowie bei komponentenbasierten Entwicklungsansätzen, verwendet wird. Ein Framework ist selbst noch kein fertiges Programm, sondern stellt den Rahmen zur Verfügung, innerhalb dessen der Programmierer eine Anwendung erstellt. Die Entwickler bauen das Framework in ihre eigene Applikation ein, und erweitern es derart, dass es ihren spezifischen Anforderungen entspricht.

6.2 Korpus

Ein Korpus ist eine Sammlung schriftlicher oder gesprochener Äußerungen in einer oder mehreren Sprachen. Die Daten des Korpus sind digitalisiert, d.h. auf Rechnern gespeichert und maschinenlesbar. Die Bestandteile des Korpus, die Texte oder Äußerungsfolgen, bestehen aus den Daten selbst sowie möglicherweise aus Metadaten, die diese Daten beschreiben und aus linguistischen Annotationen, die diesen zugeordnet sind.

6.3 Linked Open Data

Linked Open Data (LOD) bezeichnet im World Wide Web frei verfügbare Daten, die per URI identifiziert sind und darüber direkt per HTTP abgerufen werden können und ebenfalls per URI auf andere Daten verweisen. Idealerweise werden zur Kodierung und Verlinkung der Daten das Resource Description Framework (RDF) und darauf aufbauende Standards wie SPARQL und die Web Ontology Language (OWL) verwendet, so dass Linked Open Data gleichzeitig Teil des Semantic Web ist.

6.4 Named Entity Recognition (NER)

Named Entity Recognition ist ein Teilgebiet der Wissensextraktion. Hierbei werden meist unannotierte Textblöcke genommen und in annotierte Textblöcke umgewandelt. Dabei können im Text Personen, Unternehmen, Zeitangaben etc. aufgefunden werden. Die Ausgabe kann dann weiterverwendet werden.

6.5 Relationship Extraction (RE)

RE ist wie NER ein Teilgebiet der Wissensextraktion, bei dem aus einem annotierten Text und einer Wissensdatenbank aus bekannten Relationen neue Beziehungen zwischen Entitäten abgeleitet werden. BOA ist ein Tool, was dieses Vorgehen implementiert, indem es DBPedia als Wissensdatenbank nutzt.

6.6 PROV-Ontology

Die PROV-Ontologie beschreibt die Codierung des PROV Datenmodells in OWL 2, also die dabei genutzten Klassen, Objekte und Attribute.

6.7 Resource Description Framework

Resource Description Framework ist ein Framework mit dem Informationen über Resourcen dargestellt werden können. Die Grundstruktur bilden dabei Tripel aus Subjekt, Prädikat und Objekt, wobei diese Resourcen darstellen und das Prädikat die Beziehung zwischen Subjekt und Objekt beschreibt. Die Menge dieser Tripel bildet einen (gerichteten) Graph, in dem Subjekte und Objekte die Knoten und Prädikate die Kanten sind. Resourcen werden durch IRIs identifiziert, welche an allen Positionen des Tripels auftreten können. Alternativ können Subjekte und Objekte auch blank nodes sein, Prädikate auch Literale.

6.8 Wissensextraktion

Moderne Netzwerke stellen immense Datensammlungen bereit, sodass oft weniger das Beschaffen von Informationen, als vielmehr das Herausfiltern der relevanten Essenz, den größten Aufwand darstellt. Hier können Methoden der Wissensextraktion helfen, essentielle Inhalte aus vorliegenden Daten als höherwertiges Wissen zu extrahieren und damit konkrete Fragestellungen einfacher beantwortbar zu machen.

6.9 FOX und BOA

FOX ist ein Named Entity Regconition Tool. Es fasst verschiedene andere NER Tools zusammen und sucht Entitäten wie bspw. Orte und Personen in einem Text oder einer Internetseite. BOA ist ein Relation Extraction Tool. Es sucht im Text Relationen wie bspw. „geboren in“und erstellt Tripel wie bspw. (Angela Merkel, studiert in, Leipzig)