

Wissensmanagement in der Bioinformatik

Exposé zur Diplomarbeit „Semantische Anfragen an PubMed mit SPARQL“

Betreuung: Prof. Ulf Leser

Johannes Vogt

vogt@informatik.hu-berlin.de

Zeitraum: 15.10.2006 – 15.04.2007

1. Motivation

Die Datenbank *PubMed*^[1] enthält Abstracts von medizinischen und biomedizinischen Veröffentlichungen. PubMed umfasst unter anderem die Inhalte der Datenbank *MEDLINE*^[2], mit allein schon über 14 Millionen Einträgen aus 4.800 Journalen. Die Suchmaschine *Entrez*^[3] führt textbasierte Suche auf den PubMed-Inhalte durch. Dabei ist es für den Anwender oft zeitaufwändig, aus der Liste der durch Stichwortsuche gefundenen Suchergebnisse, die wirklich relevanten Einträge zu selektieren. Ein Ansatz, um gezielter nach bestimmten Inhalten in PubMed zu suchen, ist *Ali Baba*^[4]. Durch Textmining erkennt Ali Baba in PubMed-Einträgen biomedizinische Begriffe, zum Beispiel Proteine, und Zusammenhänge zwischen diesen Begriffen, beispielsweise Protein-Protein-Interaktionen. Die Begriffe werden als Knoten, und die Zusammenhänge als Kanten interpretiert. Der visualisierte Graph ermöglicht es dann, sich inhaltlich durch eine Menge von PubMed-Abstracts zu bewegen. Mit der Graph-Anfragesprache *SPARQL*^[5] könnten semantische Anfragen an diesen Graph gestellt werden. Voraussetzung dafür wäre, dass der Ali Baba Graph in *RDF*^[6] vorliegt. RDF ist eine formale Sprache zur Beschreibung von Ressourcen. Der zusätzliche Vorteil dieser Art der semantischen Suche wäre, dass Informationen aus weiteren Quellen miteinbezogen werden könnten. Beispielsweise könnten über die *UniProt*^[7]-ID eines Proteins dessen Kategorien und Überkategorien aus der *Gene Ontology*^[8] abgefragt werden, sofern diese Informationen ebenfalls in Form von RDF-

Graphen vorliegen.

2. Zielsetzung

Das Ziel dieser Diplomarbeit ist es SPARQL-Anfragen an PubMed-Inhalte zu realisieren. Der zugrundeliegende Graph ist dabei der Ali Baba Graph über in PubMed gefundene biomedizinische Objekte und deren Beziehungen zueinander. Hierzu müssen die durch die Ali Baba Textmining-Pipeline gefundenen Begriffe und Relationen in ein zuvor zu entwerfendes RDF-Schema (*RDFS*^[8]) gebracht werden. Mit *Jena ARQ*^[9] existiert ein SPARQL-Prozessor, der die Anfragen auf den RDF-Graph umsetzt. Ein wesentlicher Vorteil von RDF-Modellen besteht in der unkomplizierten Einbeziehung weiterer Daten. Beispielhaft soll dies anhand der Gene Ontology realisiert werden. Die Begriffe des Ali Baba-Graphen sind unter anderem Proteine aus UniProt. Diesen Proteinen sind in UniProt Kategorien zugewiesen, die wiederum eindeutig in der Gene Ontology beschrieben sind. Da SPARQL-Anfragen über mehrere Graphen möglich sind, können diese Zusammenhänge miteinbezogen werden, sobald die entsprechenden Informationen in Form von RDF-Graphen vorliegen (siehe Abbildung 1). Zudem soll das in der Gene Ontology auftretende Problem der transitiven Relationen erörtert werden (siehe 3. *Hintergrund*). Dabei sollen die verschiedenen Möglichkeiten in Betracht gezogen werden, wie auch indirekte Zusammenhänge abgefragt werden können, und mindestens eine dieser Möglichkeiten realisiert werden. Abschliessend soll eine Anfrageschnittstelle implementiert werden, mit der die Ergebnisse, bei denen es sich um Graphen handeln kann, visualisiert werden.

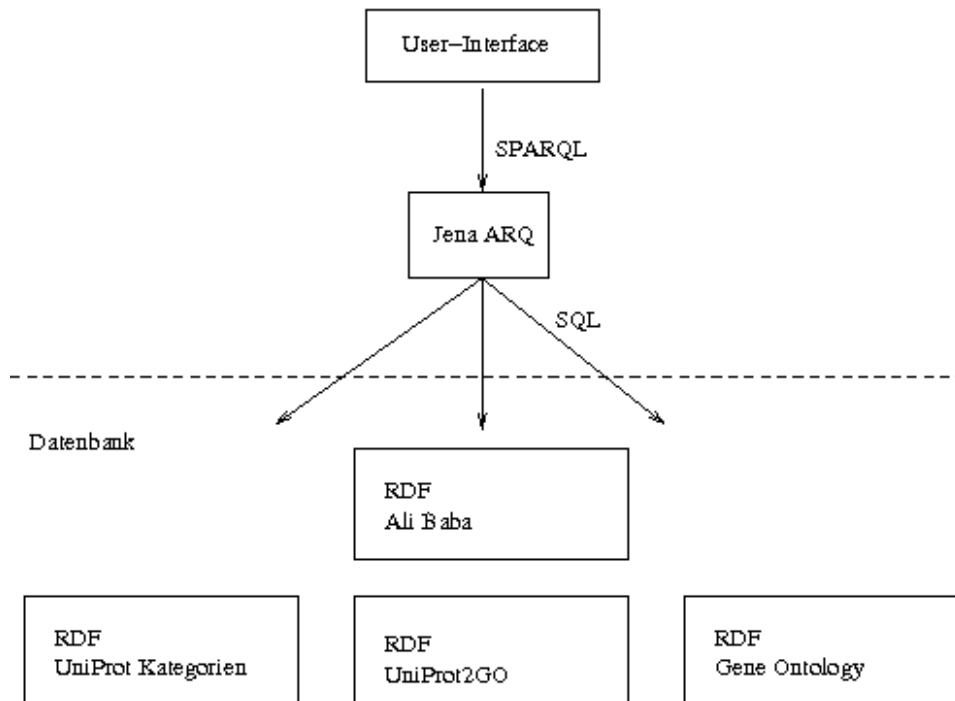


Abbildung 1: SPARQL Anfragen können von Jena-ARQ auf in relationalen Datenbanken liegende RDF-Graphen ausgeführt werden.

3. Hintergrund

RDF und SPARQL

Der W3C-Standard RDF (Resource Description Framework) ist eine formale Sprache zur Graph-Modellierung, entwickelt als grundlegendes Datenmodell des Semantic Web. RDF basiert auf Tripeln, bestehend aus Subjekt, Prädikat und Objekt. Dabei ist das Subjekt eine beliebige *Ressource*, das heisst ein definiertes Element, über das in dem jeweiligen Modell eine Aussage gemacht werden soll. Prädikate definieren den Typ der Aussage des Tripels, also die Art der Verbindung zwischen Subjekt und Objekt. Objekte definieren den Wert des Prädikats. Objekte können Literale oder ebenfalls (wie Subjekte) Ressourcen sein. Dieselbe Ressource kann also in einem Tripel Subjekt und im nächsten Objekt sein. Dadurch entsteht der Graph- und Netzwerk-Charakter von RDF. Jedes RDF-Modell kann als gerichteter Graph verstanden und dargestellt werden. SPARQL (SPARQL Protocol and RDF Query Language), ein Entwurf des W3C, umfasst ein Protokoll und eine Anfragesprache, um einfach und standardisiert auf in RDF vorliegende Informationen zuzugreifen. Die SPARQL-Anfragesprache ist in ihrer Syntax an SQL angelehnt. Entsprechend vom Nutzer definierter Pattern werden Ressourcen oder Tripel ermittelt und zurückgegeben. Anfragen können über mehrere RDF-Graphen gehen.

Ali Baba als RDF

In der Umsetzung des Ali Baba Graphen in RDF werden die gefundenen Begriffe von Ressourcen repräsentiert, und die Relationen zwischen den Begriffen von Prädikaten. Beispielsweise wird eine Relation *co-occurrence* zwischen Protein P1 und Protein P2 als Tripel T1:[P1 *co-occurrence* P2] dargestellt. Hierbei treten im Wesentlichen zwei Schwierigkeiten auf: Zum einen müssen die Tripel reifiziert werden, um die Information darüber zu übernehmen, in welchem Abstract aus PubMed diese Relation gefunden wurde. Das Tripel T1 muss selbst wieder als Ressource verstanden werden können, über das weitere Aussagen getroffen werden können, wie zum Beispiel T2:[T1 *gefunden_in* PMID:12345]. Reifikation auf einem Tripel ist relativ teuer, es sind dazu drei zusätzliche Tripel nötig.^[10] Die zweite Schwierigkeit besteht in ungerichteten Relationen. RDF-Kanten sind grundsätzlich gerichtet, in Ali Baba kommen dagegen auch ungerichtete Relationen vor, wie im obigen Beispiel *co-occurrence*. Um in einer SPARQL-Anfrage nicht beide Richtungen berücksichtigen zu müssen, ist es notwendig zusätzlich die Gegenrichtung in den RDF-Graph einzufügen, und somit das zusätzliche Tripel T1':[P2 *co-occurrence* P1] plus die entsprechenden Tripel für die Reifikation.

Transitive Relationen

In der Gene Ontology (GO), die unter anderem in RDF vorliegt, gibt es die transitiven Relationen *is_a* und *part_of*. Das heisst, wenn *A is_a B* und *B is_a C*, dann gilt implizit auch *A is_a C*, ohne dass dafür ein explizites Tripel vorliegen muss. Beispielsweise gibt es in GO den Term *all*. Indirekt gilt für alle Terme *t* in GO: *t is_a all*. GO ist als gerichteter azyklischer Graph

strukturiert. Da SPARQL diese indirekten Relationen nicht abfragen kann, gibt es an dieser Stelle entweder die Möglichkeit, alle implizit geltenden Relationen dem Graphen explizit hinzuzufügen, oder Konzepte zu entwickeln, um eine rekursiven Anfrage auf mehrere SPARQL-Anfragen abzubilden.

Jena: Semantic Web Framework

Jena^[11] ist ein von HP Labs entwickeltes, java-basiertes Semantic Web Framework, das unter anderem einen RDF-Parser, eine Programmierschnittstelle zur Manipulation von RDF-Graphen sowie verschiedene Möglichkeiten der Datenspeicherung, darunter auch die Anbindung einer relationalen Datenbank, zur Verfügung stellt. Somit bietet Jena den Rahmen für die Erstellung des Ali Baba-Graphen, und dessen Speicherung in einer Datenbank. Jena enthält mit dem Modul *ARQ* darüber hinaus einen SPARQL-Prozessor, mit dem Anfragen an RDF-Graphen (vorliegend in den von Jena unterstützten Formaten) ausgeführt werden können.

4. Vorgehen

Der erste Schritt dieser Arbeit besteht in der Darstellung und Speicherung des Ali Baba-Graphen in RDF. Aus den von der Ali Baba Textmining Pipeline erzeugten XML-Dokumente müssen dazu die Graph-Informationen mit Hilfe von Jena in RDF umgesetzt werden. Als Speichermedium ist eine Oracle-Datenbank vorgesehen. Zuvor muss das verwendete RDF-Schema entworfen werden. Anschliessend muss ein zweiter RDF-Graph erstellt werden, der Objekten (Proteinen) aus UniProt Kategorien zuweist. Diese Kategorisierungen sind in der UniProt-Datenbank bereits definiert, und können aus UniProt extrahiert werden, zum Beispiel aus in *UniProt XML format*^[12] vorliegenden Daten. Zudem wird ein weiterer RDF-Graph benötigt, der die UniProt Kategorien mit GO-Begriffen mappt. Eine solche Mapping-Liste ist bereits vorhanden^[13], und kann entsprechend in RDF übersetzt werden. Die vierte Graph ist dann die Gene Ontology selber. Diese liegt bereits in RDF vor^[14]. Aufsetzend auf ARQ kann dann eine Anfrageschnittstelle implementiert werden, die SPARQL-Anfragen auf die erzeugten Graphen entgegennimmt, und die Ergebnisse visualisiert. Anschliessend können Möglichkeiten erörtert werden, wie die indirekten Relationen in GO mit SPARQL abgefragt werden können. Dazu muss der GO-RDF-Graph eventuell erweitert werden. Zuletzt wird die Performance des Systems betrachtet und ein Ausblick auf mögliche Erweiterungen auf weitere Datenquellen gegeben.

Literatur

[1] PubMed, 2006. <http://www.pubmed.gov>

[2] MEDLINE, 2006. <http://medline.cos.com/>.

- [3] Entrez, 2006. <http://www.ncbi.nlm.nih.gov/Database/index.html>.
- [4] Aplake, C., Schiemann, T., Pankalla, M., Hakenberg, J and Leser, U. (2006). „AliBaba: PubMed as a Graph.“
- [5] SPARQL, 2006. <http://www.w3.org/TR/rdf-sparql-query/>.
- [6] RDF, 2006. <http://www.w3.org/RDF/>.
- [7] UniProt, 2006. <http://www.expasy.uniprot.org/>.
- [8] RDFS, 2006. <http://www.w3.org/TR/rdf-schema/>.
- [9] ARQ – A SPARQL Processor for Jena, 2006. <http://jena.sourceforge.net/ARQ/>.
- [10] Antoniou, G., van Harmelen, F. (2004). „A Semantic Web Primer“, S. 67.
- [11] Jena – A Semantic Web Framework for Java, 2006. <http://jena.sourceforge.net/>.
- [12] UniProt XML format, 2006.
<http://www.pir.uniprot.org/support/docs/uniprot.xsd>.
- [13] Mapping of SWISS-PROT KEYWORDS to GO terms, 2006.
<http://www.geneontology.org/external2go/spkw2go>.
- [14] GO – XML Version, 2006.
<http://www.geneontology.org/GO.format.shtml#XML>.