Rechercheberichte

Universität Leipzig – Softwarepraktikum 2014

Semantic Chess

Jonas Herrig
Hanno Krümpelmann
Nathalie Bargenda
Duc Hieu Nguyen
Stefan Süssmeier
Lisa Höncke

Inhaltsverzeichnis

1. Begriffe

Schach

Schach ist ein strategisches Brettspiel, bei dem zwei Spieler abwechselnd Schachfiguren auf einem sogenanntem Schachbrett bewegen. Jeder Spieler besitzt einen König, eine Dame, zwei Springer, zwei Läufer, zwei Türme und acht Bauern mit diversen Zugregeln. Ziel des Spiels ist, den Gegner schachmatt zu setzen, das heißt seine als König bezeichnete Spielfigur unabwendbar anzugreifen.

Um bestimmte Partien dieses Spiels in einer Datenbank suchen zu können, könnten vom User folgende Anfragen gestellt werden:

> Gebe mir alle Spieler, die eine Elo über 2000 besitzen.
> Wie sind die Namen alle Großmeister?
> Welcher Internationaler Meister hat die Größte Elo-Zahl?

Elo-Zahl

Die Elo-Zahl ist eine Wertungszahl, die die Spielstärke von Schachspielen beschreibt. Das Elo-System teilt die Spieler in verschiedene Klassen ein. Die Titel "Großmeister" und "Internationaler Meister" können einer Personen zugewiesen werden, wenn die festgelegte Normen erfüllt sind und die Mindest-Elo-Zahl errreicht wurde. Tuniere werden nach der durchschnittlichen Elo-Zahl der Teilnehmer in Kategorien eingeteilt.

Client-Server-Architektur

Mit Client-Server-Architektur ist gemeint, dass eine Aufgabe im Netzwerk auf Client und Server verteilt wird. Die verwendete Software unterteilt dabei in Client und Server. Der Client fordert dabei vom Server einen Dienst an, welcher entsprechend die Anforderung beantwortet. Client- und Serverteil eines Programms sind meist auf verschiedenen Rechnern eines Netzwerks.

Eine auch für dieses Softwareprojekt sinnvolle gängige Unterteilung ist es, die GUI und alle mit Eingaben verbundenen Programmteile auf dem Client laufen zu lassen, jedoch die Datenbank und die Suchalgorithmen auf einem Server zu implementieren.

Question Answering

Question Answering ist ein Teilgebiet der Informatik, das sich mit der Informationsbeschaffung aus natürlichen Sprachen beschäftigt. Das Hauptziel besteht darin Systeme zu entwickeln, die in der Lage sind auf von Menschen gestellte Fragen automatisch zu antworten. Um diese zu generieren stellt das implementierte Programm entweder Anfragen an eine Datenbank oder es sammelt Informationen aus einer Kollektion von unstrukturierten Dokumenten in natürlicher Sprache.

Anfragesprache

Eine Anfragesprache ist eine formale Sprache, die dazu benutzt wird um Anfragen (queries) an Datenbank und Informationssysteme zu stellen. Generell gibt es zwei Arten, zum einen die Datenbank-Anfragesprache, die versucht sachliche Antworten auf tatsächliche Fragen zu geben, zum Anderen die Abfragesprache, die versuchen alle relevanten Dokumente zu einer Anfrage zu finden.

Metadaten/Metainformationen

Als Metadaten werden strukturierte Daten bezeichnet, welche Informationen über Merkmale anderer Daten enthalten aber nicht die Daten selbst. Beispielsweise Informationen über ein Buch wie Autor, Titel, Schlagwörter etc. aber nicht den Text selbst.

Framework

Ein Framework ist ein wiederverwendbares Programmgerüst welches man als Grundstruktur für ein eigenes Projekt verwenden kann. Der Entwickler kann es dann in die eigene Anwendung einbauen und nach seinen Bedürfnissen um die erwünschten Funktionen erweitern und verwenden.

Triple Store

Ein Triple-Store ist eine Datenbank, in der Daten in Form von Triples gespeichert sind und entsprechend für die Abfrage durch Abfragesprache bereit stehen. Dabei hat ein Tripel immer die Form Subjekt-Prädikat-Objekt.

URI (Uniform Resource Identifier)

Eine URI ist allgemein ein global eindeutiger Bezeichner für eine Ressource, welche entweder abstrakt vorliegt oder auch physisch existieren kann. Der Aufbau einer URI variiert je nachdem welche Art von Objekt damit identifiziert werden soll. Meist sind URIs in Form einer URL (Uniform Resource Locator) oder eines URN (Uniform Resource Name) gegeben, es gibt jedoch auch andere Formen. Eine besondere Rolle haben URIs beim RDF Format. Hier werden die einzelnen Bestandteile eines Daten-Tripels größtenteils mit URIs bezeichnet.

IRI (Internationalized Ressource Identifier)

Eine IRI ist im wesentlichen dasselbe wie eine URI, folgt jedoch einem erweiterten Standard bezüglich verwendbarer Zeichen. Dies betrifft insbesondere die Erweiterung um Sonderzeichen diverser Sprachen (z.B. Chinesisch, Japanisch).

URL (Uniform Resource Locator)

Eine URL ist ein Spezialfall einer URI. Eine Ressource wird eindeutig innerhalb des Netzwerks anhand ihres Ortes bezeichnet. Die URL beinhaltet dabei meist Informationen über die Art des Zugriffs auf die Ressource (z.B. beginnen Verweise auf Websites meist mit „http://“ und Verweise auf Email-Adressen mit „mailto“) Somit erledigt eine URL zwei Dinge: sie bezeichnet eindeutig und lokalisiert Ressourcen. Die URL ist im Allgemeinen die häufigste Form einer URI.

Literal

Literale sind atomare Werte in diversen Programmiersprachen, welche zur direkten Darstellung von Werten dienen, beispielsweise float-Zahlen oder Strings. Literale beinhalten keine weiteren Verweise oder Weiterleitungen. Auch in RDF werden Literale eingesetzt. Dabei ist zu beachten, dass von Knoten mit Literalen keine neuen Tripel gebildet werden können. Literale sind im Kontext von RDF also stets nur Objekte (keine Subjekte oder Prädikate, diese müssen URIs sein).

PGN (Portable Game Notation)

PGN ist ein für Menschen als Text lesbares Format zur Speicherung von Schachpartien. Neben den Schachzügen beinhaltet PGN auch Meta-Informationen wie Austragungsort und Spieldatum. Für unser Projekt wurde dieses Format jedoch bereits in RDF überführt.

OWL (Web Ontology Language)

Ist eine durch das W3C spezifizierte formale Beschreibungssprache für Ontologien. Eine Ontologie dient dazu, eine Menge von Begriffen zu ordnen, sowie die Beziehungen zwischen ihnen geordnet darzustellen. Dabei muss eine Ontologie für Maschinen lesbar und anwendbar sein.

OWL bietet eine formale Semantik um eine derartige Ontologie aufzubauen. Dazu erweitert OWL den RDF/RDFS-Syntax um einige mächtigere Elemente.

Pipeline-Verarbeitung

Wenn mehrere Prozesse wie Transformationen oder Validierungen hintereinander auftreten, so spricht man von einer Pipeline oder im Falle von XML-Prozesses von einer XML-Pipeline. Ein Beispiel: Seien T1 und T2 Transformationen. Nun kann man diese Transformationen miteinander verbinden indem man das Eingangsdokument von T1 transformiert und das Ausgangsdokument von T1 dann zum Eingangsdokument von T2 wird.

2. Konzepte

Semantic Web

Semantic Web ist ein Konzept, um den für Menschen gedachten, in natürlicher Sprache formulierten Informationen im World Wide Web eine semantische Bedeutung zuzuordnen, sodass diese auch von Maschinen bzw. Anwendungen, die auf diesen ausgeführt werden, genutzt und verarbeitet werden können. Dies ist für die Verarbeitung von Suchanfragen in natürlicher Sprache von entscheidender Bedeutung.

RDF

Das Ressource Description Framework ist ein Modell zum beschreiben von Aussagen bzw. Wissen. Genauer handelt es sich um Tripel der Form Subjekt – Prädikat – Objekt, welche eine Relation zwischen Subjekt und Objekt repräsentieren. Eine Menge solcher Tripel beschreibt somit einen gerichteten Graphen, wobei Subjekte und Objekte die Knoten und Prädikate die Kanten (von Subjekt zu Objekt) bezeichnen. Nachfolgend einige Beispiele:

Subjekt	Prädikat	Objekt
Kaffee	Enthält	Coffein
Leipzig	Liegt in	Deutschland
Leipzig	Hat Vorwahl	0341

Für RDF gibt es verschiedene Notationsformen, siehe Turtle und RDF/XML.

RDF-Format Turtle

Die Darstellung von RDF mit Hilfe von Turtle ist für den Menschen relativ gut zu lesen und zu schreiben und kann von Maschinen eindeutig interprtiert werden.

URIs werden in spitzen klammern dargestellt.

<http://example.org/MusterMann>

Literale werden in Anführungszeichen angegeben.

Tripel werden durch einen Punkt beendet. Damit nicht immer komplette URIs ausgeschrieben werden müssen, können Präfixe angelegt werden. (@prefix ex: <http://example.org/>.)

ex:MusterMann ist somit äquivalent zum oberen Beispiel.

Möchte man ein Subjekt für mehrere Tripel verwenden, so beendet man den ersten Tripel mit einem Semikolon und führt für den nächsten Tripel,der das gleiche Subjekt besitzen soll, nur mit Prädikat

und Objekt fort.

@prefix ex: <http://example.org/>.

ex:MusterMann ex:WohntIn ex:MusterHausen;

ex:IsstGern ex:Blumenkohl.

Möchte man sowohl Subjekt als auch Prädikate übernehmen, so beendet man mit Komme und führt mit dem Objekt fort

@prefix ex: <http://example.org/>.

ex:MusterMann ex:IsstGern ex:Blumenkohl,

ex:Tomate.

Beide Zeichen können miteinander kombiniert werden.

RDF-Format XML

Auch wenn wenn Turtle vergleichsweise simpel, wird es nicht häufig verwendet. Dies liegt daran, dass es für diese Schreibweise oft keine weit verbreiteten Programmbibliotheken gibt.

Die XML-Darstellung ist zwar nicht so menschenfreundlich wie Turtle. Dagegen bieten aber viele Programmiersprachen Unterstützungstools und Bibliotheken für die Verarbeitung von XML an.

Wie in XML werden ebenfalls Namensräume deklariert.

z.B.: rdf: Subjekt und Objekt werden durch Elemente des Typs rdf:Description beschrieben, wobei rdf:about den Bezeichner des Elements (die URI) angibt.

Das Prädikat wird als verschachteltes Element (Tag) im Element rdf:Description dargestellt.

<rdf:Description rdf:about="http://example.org/MusterMann">

    <ex:WohntIn>

        <rdf:Description rdf:about="http://example.org/MusterHausen">

        </rdf:Description>

    </ex:WohntIn>

</rdf:Description>

Ein Objekt kann durch das Attribut rdf:resource vom Prädikaten angegeben werden.

<rdf:Description rdf:about="http://example.org/MusterMann">

    <ex:WohntIn rdf:resource="http://example.org/MusterHausen">

    </ex:WohntIn>

</rdf:Description>

Literale werden einfach als Inhalte eines Prädikaten-Elements dargestellt.

<rdf:Description rdf:about="http://example.org/Buch">

<ex:Titel>SemanticWeb</ex:Titel>

</rdf:Description>

SPARQL

Die **S**PARQL **P**rotocol **A**nd **R**DF **Q**uery **L**anguage ist eine Abfragesprache für RDF. Hierfür wird eine Engine benötigt, welche die queries verarbeitet. Engines sind z.B. in Apache Jena oder Virtuoso enthalten.

Apache Jena – ARQ

Apache Jena ist ein Framework für Java, mit welchem RDF-Graphen geladen und gespeichert werden können. Jena repräsentiert diese Graphen als abstrakte Modelle, welche die Daten aus Dateien, Datenbanken oder URLs beziehen. Jena enthält desweiteren eine Anfrage-Engine (genannt ARQ), um SPARQL-queries auf diesen Modellen auszuführen.

Virtuoso

Virtuoso ist ein „Universaler Datenserver", der verschiedene Modelle unterstützt. Unter anderem kann Virtuoso als Triplestore für RDF-Daten genutzt werden. An diesen können SPARQL-queries gestellt werden. Im Unterschied zu Jena ist Virtuoso eine eigenständige Serveranwendung.

Benchmark

Benchmarks werden genutzt um Programme oder Konzepte im Hinblick auf Effizienz und Effektivität zu testen. Grade bei Suchalgorithmen und der dem Projekt vorliegenden Datenmenge sind diese offensichtlich entscheident um eine aktzeptable Performance zu erreichen. Sollten die Benchmarks mangelhafte Ergebnisse hervorbringen so muss der Code bzw. das Konzept überarbeitet werden. Für dieses Projekt müssen die Benchmarks/Testprogramme selbst geschrieben werden und optimalerweise bereits in einem frühen Stadium des Projekts zum Testen eines Prototypen verwendet werden.

3. Aspekte

Ausgangspunkt unseres Projekts

Der vorige Jahrgang der Gruppe semantic chess hat bereits ein Programm entworfen, welches die Schachdaten wahlweise als RDF-XML oder RDF-Turtle bereitstellt. Unsere Aufgabe wird es also sein, eine Client-Server Lösung zu entwickeln, die eine Suche auf diesen Daten mittels natürlicher (englischer) Sprache implementiert.

Die Umsetzung dieser Anforderung bietet zwei Kernprobleme: Einerseits muss eine Lösung gefunden werden, Fragen in natürlicher Sprache semantisch und inhaltlich für die Such-Engine in Form einer SPAQRL-Anfrage lesbar zu machen. Andererseits muss die Suche selbst effektiv implementiert werden, sodass der Dienst Laufzeiten bietet, die für einen durchschnittlichen User akzeptabel sind.

Existierende Arbeiten

Glücklicherweise gibt es bereits Arbeiten, die sich isoliert mit jedem unserer Kernprobleme befasst haben. So hat eine Arbeitsgruppe der Universität Leipzig im Paper „TBSL Question Answering System Demo“ sich detailliert mit dem Problem befasst, wie bestmöglich aus natürlicher Sprache eine SPARQL-Anfrage generiert wird.

Des weiteren stellen Denis Lukovnikov und Axel-Cyrille Ngonga Ngomo in ihrem Paper „SESSA - Keyword-Based Entity Search through Coloured Spreading Activation“ einen effektiven Suchalgorithmus für RDF-Daten vor.

Diese existierenden Arbeiten bieten uns einen inhaltlich starken Ansatzpunkt für unser Projekt.

Zusätzliche Anforderungen

Neben den funktionalen Anforderungen ist unser Ziel auch, ein ansprechendes User-Interface zu gestalten, sowie die Unterteilung der Client-Server-Architektur sinnvoll zu gestalten.

4. Quellen

Hitzler, Pascal. Semantic Web: Grundlagen. Berlin: Springer Berlin, 2008. Print.

http://www.itwissen.info/definition/lexikon/Client-Server-Architektur-C-S-client-server-architecture.html

http://www.w3.org/TR/uri-clarification/

http://de.wikipedia.org/wiki/Framework

http://en.wikipedia.org/wiki/Question_answering

http://jena.apache.org/index.html

http://virtuoso.openlinksw.com/

http://whatis.techtarget.com/definition/framework

http://www.enzyklo.de/Begriff/Benchmark

http://www.itwissen.info/

http://www.itwissen.info/definition/lexikon/uniform-resource-identifier-URI.html

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

https://de.wikipedia.org/wiki/Resource_Description_Framework

https://en.wikipedia.org/wiki/Jena_%28framework%29

https://en.wikipedia.org/wiki/Semantic_Web

https://en.wikipedia.org/wiki/SPARQL