Projektangebot

Verantwortlicher: Daniel Alexander

Daniel Alexander, Jakob Kusnick, Richard Khulusi, Christopher Schott, Tobias Turke, Nikolas Oberhäuser, Hans Dieter Pogrzeba, Felix Conrads

Inhaltsverzeichnis

1. Zielbestimmung

Ziel des Projekts ist die Erstellung einer Anwendung zum Sammeln und zielgerichteten Auswerten von Schachdaten.
Zu diesem Zweck müssen drei Crawler erstellt werden, die das Internet nach Schachdaten im PGN-Format, Informationen zu Schachspielern und Events durchsuchen. Diese gesammelten Daten werden dann in RDF-Triples konvertiert, die sich in einem Triplestore speichern lassen. Mit Mitteln der Data-Fusion werden die gesammelten Daten schrittweise in die Menge der bereits vorhandenen Daten eingefügt.
Weiterführend kann eine Anwendung erstellt werden, mit der die im Triplestore hinterlegten Datensätze ausgewertet und abgefragt werden können.
Am Ende soll diese Anwendung dann Fragen wie beispielsweise die Folgende beantworten können: „Welche Spieler aus Deutschland mit einer ELO von über 2500 haben im Jahr 2007 in China gespielt?“

2. Voraussetzungen

Eine Voraussetzung, um das Projekt erfolgreich durchzuführen, ist u.a. der Konverter von PGN zu RDF, welchen die Gruppe SWP13-SC im Vorjahr erstellt hat.
Weiter wird der durch den Benchmark Test als Vorprojekt gefundene Triplestore benötigt.
Um das Projekt möglichst effizient zu bearbeiten, ist ein grundlegendes Verständnis von LIMES und dessen Benutzung sowie Wissen über die Funktionsweise eines Crawlers unabdingbar.

3. Designübersicht

3.1 Wichtige Rollen

3.1.1 User

Der größte Anteil von Anwendern wird der **User** sein. Dieser besitzt keinerlei Schreibrechte an der Datenbank. Er darf lediglich lesende Anfragen stellen.
Der User hat die Möglichkeit über facettenbasiertes Browsing eine SPARQL-Anfrage zu erzeugen oder über eine Eingabezeile Suchanfragen direkt in SPARQL an den Triplestore zu stellen.
Es werden dem User generelle Informationen über den Datenbestand, wie etwa wie viele Schachpartien enthalten sind, präsentiert.

3.1.2 Administrator

Der **Administrator** ist ein Benutzer mit erhöhten/erweiterten Rechten, die ihm zusätzlich zu den öffentlich verfügbaren Funktionalitäten auch die Möglichkeit der Bearbeitung der Daten, sowie Zugriff auf die Funktionalitäten des Crawlers gestatten. Er kann Zeitintervalle der Crawlersuche direkt beeinflussen und den Speicherinhalt des Triplestore verwalten.

3.2 Nutzerszenarien

3.2.1 User

1. Aufruf der Homepage:
   Der User bekommt seine Eingabemöglichkeiten aufgezeigt. Dies umfasst eine Eingabezeile für SPARQL-Queries (und optional ein Formular zur Auswahl interessanter Informationen).

2. Eingabe einer Suchanfrage über Eingabezeile:
   Der User gibt eine SPARQL-Query in die Eingabezeile ein und bekommt als Ergebnis eine Antwortseite dargestellt.

(zusätzliche optional umzusetzende Szenarien)

3. Eingabe einer Suchanfrage in Satzform:
   Der User gibt einen Fragesatz ein. Dieser wird vom System in eine SPARQL-Query konvertiert und anschließend entsprechend bearbeitet. (z.B. „Welcher deutsche Spieler besitzt die höchste Elo?“)
4. Erstellen von Statistiken:
   Der User kann über das Anfrageformular eine Statistik erfragen. (z.B. „Welches sind die 10 besten Eröffnungen für weiß?“)
5. Erfragen von Personen- oder Partiedetails über Anfrageformular:
   Der User benutzt ein bereitgestelltes Formular, um konkrete Informationen zu einzelnen Personen (Schachspielern) oder Partien abzufragen. Dabei kann mittels Linking unter anderem auch zusätzliches Material wie etwa ein Profilbild o.Ä. von anderen Webseiten bereitgestellt werden.

3.2.2 Administrator

1. Einloggen/Anmelden:
   Durch Anmelden am System ist es dem Administrator möglich, erweiterte Systemfunktionen, die hauptsächlich den Crawler und den Triplestore betreffen, auszuführen.
   Bestimmte Aufgaben erfordern direkt diesen Administrator-Zugriff, da nur berechtigte Nutzer in der Lage sein sollten, entsprechende Aufgaben auszuführen.

2. Ausloggen/Abmelden:
   Ist ein Administrator eingeloggt und hat seine Arbeit beendet, so meldet er sich letztlich wieder vom System ab. Dies ist eine übliche Sicherheitsmaßnahme. Auf diesem Wege gibt er seine zusätzlichen Berechtigungen wieder ab. Um das System anschließend wieder verwalten zu können, muss er sich erneut einloggen.

3. Ansehen von Logfiles:
   Kommt es einmal zu einem Fehler, so ist es günstig, auf Logfiles zurückgreifen zu können. Mit ihrer Hilfe kann man Änderungen im System wie das Einfügen, Löschen oder Bearbeiten von Daten im Triplestore nachvollziehen und Rückschlüsse auf den Ursprung des aufgetretenen Fehlers ziehen.

4. Bearbeiten von Daten:
   Der Administrator kann auf jeden gespeicherten Datensatz zugreifen und ihn nachträglich manipulieren. Dies darf dem User unter keinen Umständen möglich sein, da hierdurch die Konsistenz der Daten gefährdet werden würde. Der Administrator kann hierdurch fehlerhafte Daten von Hand korrigieren.

5. Löschen von Daten:
   Fehlerhafte oder korrupte Daten können dem System eventuell Schaden oder Suchanfragen behindern. Werden solche Datensätze erkannt, so sollte die Möglichkeit bestehen, diese Daten aus dem System zu entfernen.

6. Hinzufügen von Daten:
   Der Administrator hat die Möglichkeit, kurzfristig gewonnene Datensätze auch manuell einzutragen. Dies schließt den Import von PGN-Dateien mit ein.

4. Arbeitspakete und Funktionalität

Nr.	Arbeitspaket	Aufwandsanteil
1.1	**Schnittstelle zu Triplestores**	10%
1.2	**Benchmark (Vorprojekt)**	20%
1.3	**Crawler**	40%
1.4	**Sammeln/Data Fusion**	30%
2.1	**Oberfläche (User und Administrator)**	10%
2.2	**Parsing und Verarbeitung von Anfragen**	10%
	Gesamtes Projekt	***120%***

Tabellarische Darstellung der Arbeitspakete und deren Anteil

4.1 Mussziele

4.1.1 Schnittstelle zu Triplestores

Ein wichtiger Grundstein für die Auswertung und Verarbeitung der mit dem Crawler gefundenen Daten ist deren Speicherung in einem passenden Triplestore. Hierfür müssen zunächst die Daten in ein kompatibles Format gebracht werden, dies wurde durch die Vorgängergruppe bereits zu unserem Nutzen ausgearbeitet. Anschließend ist es Aufgabe des Teams, die Daten effizient in den Triplestore zu laden. Um dies zu ermöglichen, soll eine Schnittstelle programmiert werden, so dass Daten komfortabel aus dem Triplestore herausgelesen und hineingeschrieben werden können. Alternativ können vorgefertigte Frameworks dazu dienen, die Schnittstelle zu vereinfachen, indem die Frameworks für das Projekt angepasst werden.

4.1.2 Benchmark (Vorprojekt)

Um eine passende Datenbank für die erstellten Triple zu finden, ist es nötig einen Benchmark Test durchzuführen. Hierfür werden in verschiedene geeignete Triplestores Daten geschrieben und anschließend getestet, wie schnell die Anfragen auf diese Daten verarbeitet werden. So soll ein möglichst effizienter und schneller Triplestore gefunden werden, um einen reibungslosen und angenehmen Arbeitsfluss zu gewährleisten. Nach Möglichkeit geschieht dies für jeden Triplestore nahezu automatisiert, indem entweder ein vorgefertigtes Framework benutzt wird oder die Schnittstelle zu den Triplestores im Vorfeld programmiert wird.

4.1.3 Crawler

Die Crawler bilden das zentrale Element des Projektes. Sie sollen Schachdaten, wie Informationen zu Schachspielern, Events und Partien, im Internet suchen.
Dabei werden sowohl komprimierte wie auch nicht komprimierte Daten betrachtet.
Diese Daten werden zusammentragen und anschließend zur Weiterverarbeitung zugänglich gemacht. Hierzu wird ein Zwischenspeicher aufgebaut, um effizienteres Abarbeiten der gefunden Datenbestände zu ermöglichen.
Besonderes Augenmerk gilt es, auf das richtige „Durchwandern" des Internets zu werfen, sodass nötige Webseiten besucht und nur relevante Inhalte heruntergeladen werden. Ebenso wichtig ist es, Webseiten in kurzen Zeitintervallen nicht mehrfach zu besuchen. Die Crawler werden auf Basis eines vorhandenen Crawler-Frameworks, wie zum Beispiel Crawler4j oder WebSPHINX, entwickelt.

4.1.4 Sammeln / Data Fusion

Da eine möglichst lückenlose Datenlandschaft erstrebenswert wäre, ist es wichtig, die bei verschiedenen Quellen gefundenen Daten zu sammeln und anschließend sinnvoll zu fusionieren. Gegebenenfalls kann das Fehlen von einzelnen Informationen durch die Rekonstruktion aus vorhandenen, korrespondierenden Inhalten behoben werden.
Hierzu wird das Framework LIMES benutzt um zwischen den vorliegenden Dateien im Zwischenspeicher und dem Triplestore schnell zu vergleichen. Einem Spiel, Event oder Spieler wird eine URI mit einer fortlaufenden ID zugeordnet.
Um einen konsistenten Datenbestand zu gewährleisten, werden Regeln definiert, so dass zum Beispiel ein Turnier nicht über mehrere Jahre laufen kann.

4.2 Kannziele

4.2.1 Oberfläche (User und Administrator)

Ein optionales Ziel wäre es, eine Benutzeroberfläche zu schaffen, über die ein User bequem auf die Daten zugreifen kann. Dies umfasst eine komfortable Eingabemöglichkeit der Suchanfragen, möglicherweise über ein Textfeld in Satzform oder über facettenbasiertes Browsing. Eventuell wäre auch ein gesonderter Verwaltungsbereich für Administratoren von Vorteil. Zusätzlich könnten noch einige zusätzliche visuelle Features eingebunden werden, wie zum Beispiel die übersichtliche Ansicht der Daten im Allgemeinen und das Anzeigen von eventuellen Spielerbildern oder eine kompaktere Ansicht für mobile Endgeräte im Speziellen.
Möglich wäre es zudem eine Art Statistik der gestellten Anfragen oder der wichtigsten Schachdaten anzuzeigen.
Um ein Anfrageergebnis für spätere Verwendung zu sichern, könnte eine Option zum Speichern eingeführt werden, sodass die angefragten Daten als PDF oder in anderen Formaten heruntergeladen werden können.
Eine Idee der Erweiterung wäre überdies eine Einbettung des Suchformulars in eine andere Webseite. So kann Schachinteressierten im Internet eine Möglichkeit geboten werden, direkt und unkompliziert Antworten auf ihre Fragen zu finden.

4.2.2 Parsing und Verarbeitung von Anfragesätzen

Ein kleineres Gebiet ist die Verarbeitung der gestellten Suchanfragen. Diese sollte einfach ablaufen, wenn die Anfragen in „roher" Form, also SPARQL-Queries, gestellt werden. Schwieriger wird es, wenn eine entsprechende Oberfläche zulässt, dass Anfragen als Satz gestellt werden. Hierzu dient facettenbasiertes Browsing, wie etwa ein Graphical SPARQL-Builder.

5. Meilensteine

In diesem Abschnitt sollen die Meilensteine der Implementierung festgehalten werden. Auf diese Weise wird ein grober Ablaufplan für das Projekt vorgestellt. Es wird dennoch parallel an allen Meilensteinen gearbeitet.

5.1 Meilenstein - Triplestore und Crawler

Abgabedatum: 07.04.2014

Der Erste Meilenstein wird die Fertigstellung der Crawler und deren Verbindung mit dem gewählten Triplestore sein. Das Vorprojekt liefert dann bereits den zu verwendeten Triplestore als Ergebnis und stellt eine Schnittstelle zu diesem bereit, die im Folgenden weiterverwendet werden kann.
Eine ausführliche Testreihe, die den Funktionsumfang der Crawler und deren Korrektheit prüft, schließt den Ersten Meilenstein ab.
An diesem Punkt kann der Triplestore bereits mit Daten gefüllt werden.

5.2 Meilenstein - Data-Fusion

Abgabedatum: 05.05.2014

Der Zweite Meilenstein wird mit der optimalen Integration der gesammelten Daten in die bereits im Triplestore vorhandenen Daten gleichzusetzen sein.
Zu diesem Zweck muss ein Algorithmus entworfen werden, der korrekte Data-Fusion ermöglicht. Dadurch werden dann nicht nur neue Daten gesammelt, sondern auch vorhandene Daten ergänzt und korrigiert. So kann es beispielsweise passieren, dass es Einträge mit chinesischen Zeichen gibt, die unter Verwendung des ASCII- bzw. Unicode-Zeichensatzes nicht korrekt lesbar sind oder innerhalb der deutschen Sprache keinen Sinn ergeben. Diese Einträge werden dann sofern vorhanden durch Einträge ersetzt, die dem verwendeten Standard entsprechen. Weiterhin ist in diesem Zusammenhang ein Algorithmus zur Erkennung von Duplikaten zu implementieren, der die Aufnahme doppelter Datensätze verhindert. Die genannten Algorithmen werden bereits während der Implementierung streng auf ihre Richtigkeit getestet.

5.3 Meilenstein - Parsing, Anfrageverarbeitung und Web-Oberfläche

Abgabedatum: 12.05.2014

Zum Abschluss des Dritten Meilensteins ist eine Web-Oberfläche zu erstellen, mit deren Hilfe der Nutzer Anfragen an das System stellen kann. In diesem Kontext gilt es, einen Parser zu implementieren, der die Eingaben der Nutzer in SPARQL-Queries konvertiert, die anschließend an den Triplestore weitergeleitet werden. Das Ergebnis der Anfrage muss ebenso in einer für den Nutzer les- und verwendbaren Form ausgegeben werden.

6. Qualitätssicherung

Diese Qualitätsanforderungen sind während des Projektes von den Teammitgliedern zu beachten und zu erfüllen:

Produktqualität	Sehr gut	Gut	Normal	Nicht relevant
Funktionalität			x	
Zuverlässigkeit		x		
Benutzbarkeit		x		
Effizienz	x			
Änderbarkeit		x		
Übertragbarkeit		x		

Begründung:

Bei dem Projekt steht vor allem die Effizienz im Vordergrund. Sowohl die Crawler als auch die Triplestores müssen schnell die jeweiligen Daten verarbeiten, da es sonst bei den gewaltigen Datenmengen zu großen Verzögerungen kommen kann.
Weniger wichtig sind dagegen Zuverlässigkeit, Benutzbarkeit, Änderbarkeit und Übertragbarkeit, dennoch dürfen sie aus verschiedenen Gründen nicht vernachlässigt werden. Die Crawler sollten möglichst ohne Ausfälle arbeiten, weil eine bestimmte Zeit benötigt wird um eine neue Suche zu starten. Daher muss eine gewisse Zuverlässigkeit gewährleistet sein. Die Benutzbarkeit darf nicht vernachlässigt werden, denn die Crawler sollten schnell und unkompliziert gestartet werden können. Die Änderbarkeit muss gut sein, um kurzfristig Änderungen an den Suchbedingungen ausführen zu können. Um auf verschiedenen Servern zu laufen, muss Übertragbarkeit in einem gewissen Maße gegeben sein.
Der Funktionalität schließlich ist keine besondere Aufmerksamkeit zu schenken, da sowohl die Crawler als auch die Schnittstelle zum Triplestore nur ihre spezielle Aufgabe erfüllen und keinen erweiterten Funktionsumfang bieten müssen.

7. Glossar (s. externes Dokument)