

## **Excerpt zur Bachelorarbeit „Semantic Chess“**

Die Arbeit zum Thema „Semantic Chess“ besteht aus zwei Aufgabenstellungen. Im ersten Schritt ist das Ziel das Schachdateiformat PGN in RDF zu konvertieren. Auf dieser Datenbasis wird dann eine semantische Suche entwickelt. Als Ergebnis soll das Question Answering System Nutzern Antworten auf natürlichsprachliche Anfragen geben. Ein Beispiel wäre: „Show me all won games by Carlsen against Nakamura“.

### **Struktur der Arbeit**

Die Arbeit besteht aus sieben Abschnitten. Die Einleitung beschreibt die Problemstellung und die Motivation. Danach wird ein Überblick über den aktuellen Stand der Technik gezeigt, sowie relevante Arbeiten vorgestellt. Der folgende Abschnitt enthält den logischen Aufbau, gefolgt von der Implementierung und der Lizenz des fertigen Codes. Das fertige Programm erhält eine Evaluierung in Form eines Benchmarks. Den Schluss der Arbeit bilden eine Zusammenfassung und das Literatur- und Referenzverzeichnis.

### **Benchmark**

Das Benchmark enthält 50 Schachfragen, die ein Nutzer dem Programm stellen könnte. Die Qualität der erhaltenen Antworten wird am Ende der Arbeit durch die Berechnung von Precision, Recall und F-Measure bewertet.

### **Ansatz zur Implementierung**

Die PGN-RDF-Konvertierung erzeugt RDF-Tripel, die ebenfalls wieder in PGN-Dateien umgewandelt werden können. Die Tripel werden in einem Triplestore gespeichert. Bei der Übersetzung helfen Wörterbücher, die z.B. den ECO-Code in den Eröffnungsnamen umwandelt, einem Spieler seinen Titel zuordnet oder aus der PGN allgemeine Schachbegriffe herausliest.

Das Question Answering System wird Template-basiert sein. Das bedeutet, dass aus den Fragen des Benchmarks SPARQL-Templates manuell erzeugt werden. Sie helfen später dabei Antworten auf die Anfragen zu finden.

Ein POS-Tagger kategorisiert zunächst die Wörter in der Nutzeranfrage. Zum Teil selbst erstellte Lexika stellen sicher, dass diese Ausdrücke erkannt werden. Danach ermittelt eine Ähnlichkeitsmessung für die Anfrage das passende Template.

Im Anschluss sollen die Slots der Templates ersetzt werden. Dazu müssen die

passenden Relationen und Entitäten ermittelt werden. Hierbei helfen Pattern-Bibliotheken und Frameworks, wie `fox-demo.aksw.org`. Nach der Identifizierung gibt es verschiedene mögliche SPARQL-Anfragen. Um die gewünschte Antwort auf die Nutzerfrage zu finden, müssen diese Anfragen in einem Ranking geordnet werden. Das System ermittelt aus dem Ranking die passende Antwort für den Nutzer.

Damit der Nutzer mit dem System interagieren kann, wird ein Interface entwickelt.