

Exposé zur Studienarbeit

Verwaltung und Überlagerung mehrerer Graphen in AliBaba

Ahmet Emre Kutbay

24. Juni 2006

Betreuer: Jörg Hakenberg, Conrad Plake, Prof. Ulf Leser
HU Berlin, Institut für Informatik

1 Hintergrund

Ali Baba ist eine Client-Server-Applikation zur Extraktion und graphischen Darstellung von Assoziationen zwischen biologischen Objekten in Abstracts.

Wird eine Suchanfrage in den Client eingegeben, so wird sie an den Server weiter geschickt. Dieser leitet sie an PubMed[1] weiter, und sammelt die Ergebnisse wieder ein. Die Ergebnisse sind Abstracts von Dokumenten, die von PubMed bezüglich der Suchanfrage gefunden wurden. Die Abstracts werden mit Hilfe von Tags in einer Textminingpipeline annotiert. Gefunden werden dabei Proteine, Spezies, Medikamente, Zellen, Krankheiten und Gewebearten.

Anschließend erkennt ein Parser, ob zwei annotierte Wörter in einem Abstract in einem Zusammenhang stehen. In [2] sind die Assoziationen aufgelistet, die gefunden werden können. Die annotierten Abstracts mit den Informationen über die Assoziationen der einzelnen Elemente werden an den Client zurückgeschickt, welcher anschließend die graphische Darstellung realisiert.

Die Oberfläche des Clients ist aufgeteilt in einen Bereich für die graphische Darstellung, einen Bereich, der eine Übersicht über die gefundenen Objekte bzw. Abstracts zeigt, und einen Bereich, in dem Abstracts dargestellt werden. Die graphische Darstellung bildet die Suchergebnisse als Netzwerk bzw. Graphen ab. Biologische Elemente werden durch Knoten repräsentiert, die je nach Kategorie des Elements in verschiedenen Farben markiert sind.

Assoziationen werden als Kanten zwischen zwei Elementen dargestellt. Je stärker zwei Elemente miteinander assoziiert sind, desto dunkler sind die Kanten gezeichnet. Wurde durch das Parsen des Abstracts die Rolle zweier Elemente bei ihrer Interaktion erkannt, so sind die Kanten mit dieser beschriftet.

2 Motivation und Ziele

Die aus Texten extrahierten Informationen sollen mit Daten aus anderen Quellen angereichert und verglichen werden. Als Schritt in diese Richtung wird es in dieser Studienarbeit ermöglicht werden, Graphen in einer lokalen Datenbank

persistent abzuspeichern, sie zu bearbeiten und mehrere Graphen zu verwalten. Als weitere Datenquelle werden KEGG[3] Pathways in AliBaba mit einbezogen.

Um dem Anwender die Möglichkeit zu geben, Graphen übersichtlicher zu gestalten, müssen Positionsinformationen der Knoten berücksichtigt und beim Ablegen der Graphen in die Datenbank ebenfalls mit abgespeichert werden.

3 Vorgehensweise

Zunächst müssen die Knotenpositionen in das Datenmodell des Graphen mit einbezogen werden. Daraufhin wird ein weiterer Ansichtsmodus erstellt, der die Knoten anhand ihrer internen Koordinaten positioniert. Anschliessend wird das Hinzufügen, Löschen und das Bearbeiten von Knoten und Kanten umgesetzt.

Die Speicherung des Graphen, der annotierten Abstracts sowie der Filtereinstellungen und Positionsinformationen erfolgt auf einem mySQL-Server. Da bereits das lokale Abspeichern des Graphen im GraphML Format[4] implementiert ist, wird sich das Datenbankschema am GraphML Schema orientieren.

Ist bereits ein Graph im Client geladen, so wird man beim Laden eines weiteren Graphen die Möglichkeit haben zu entscheiden, ob der vorhandene Graphen ersetzt oder ob der vorhandene Graph mit dem neuen Graphen überlagert werden soll. Bei der Überlagerung zweier Graphen werden Knoten, die das gleiche Objekt repräsentieren, vereint. Um zwei überlagerte Graphen trotzdem unterscheiden zu können, werden diese unterschiedlich eingefärbt.

Zur Kommunikation mit der lokalen Datenbank wird die JDBC-API¹ eingesetzt. Zum Bearbeiten und Darstellen des Graphen wird das Java Framework Prefuse[5] eingesetzt, welches ebenfalls die Darstellung der Pathways aus KEGG übernimmt. Die Pathways werden mittels eines Dumps in die MySQL-Datenbank eingelesen.

Literatur

- [1] *Entrez PubMed*. <http://www.pubmed.org>.
- [2] *PubMed as a graph*. <http://wbi.informatik.hu-berlin.de:8080/about.html>.
- [3] *Kyoto Encyclopedia of Genes and Genomes*. <http://www.genome.jp/kegg>.
- [4] *The GraphML File Format*. <http://graphml.graphdrawing.org>.
- [5] *Interactive information visualization toolkit*. <http://www.prefuse.org>.

¹Java Database Connectivity