Exposé zur Studienarbeit

Filtern von Fremdschlüsseln aus Inklusionsbeziehungen

Autor: Oliver Albrecht
E-Mail: oliver.albrecht@informatik.hu-berlin.de

Betreuerin: Jana Bauckmann

1. Motivation

Im Aladin-Projekt[1] (ALmost Automatic Data INtegration) des Lehrstuhls „Wissensmanagement in der Bioinformatik“[2] werden Verfahren zur automatischen Erkennung von Strukturen in unstrukturierten biowissenschaftlichen Datenbanken untersucht. Bei diesen Datenbanken sind in vielen Fällen die Fremdschlüsselbeziehungen nicht bekannt. Um diese zu finden, wurde ebenfalls am Lehrstuhl der SPIDER Algorithmus entwickelt [1]. Dieser findet Inklusionsabhängigkeiten in Datenbanken.

Inklusionsabhängigkeit (IND) bedeutet, dass alle Werte eines Attributs A in den Werten eines anderen Attributs B zu finden sind. Dabei ist A das *abhängige Attribut* und B das *referenzierte Attribut*.
Jeder Fremdschlüssel ist auch eine Inklusionsabhängigkeit, doch nicht jede Inklusionsabhängigkeit ist auch ein Fremdschlüssel. Attributpaare können sich auch in ihren Werten überdecken, ohne dass eine echte Fremdschlüsselbeziehung besteht. Dies ist häufig bei Attributen der Fall, die aus relativ wenigen Werten bestehen.

Um aus den gefundenen INDs die Fremdschlüssel heraus zu filtern, müssen diese anhand von zu definierenden Heuristiken untersucht werden. Mit diesen soll eine Bewertung über die Wahrscheinlichkeit getroffen werden, dass eine Inklusionsabhängigkeit einen Fremdschlüssel repräsentiert.

2. Zielsetzung

Das Ziel dieser Studienarbeit ist es, verschiedene Heuristiken zu finden und zu evaluieren, die Fremdschlüsselbeziehungen zu bewerten und dabei semantisch sinnvolle und sinnlose Inklusionsabhängigkeiten zu unterscheiden. Dabei sollen bereits bekannte Heuristiken angewendet und neue entwickelt werden.
Das Ergebnis soll ein Vergleich verschiedener Heuristiken sein, die Inklusionsabhängigkeiten untersuchen und eine Bewertung geben, welche davon Fremdschlüsselkandidaten sind und welche nicht. Grundlage sind die gefundenen Inklusionsabhängigkeiten des SPIDER Algorithmus [1].

1 http://www.informatik.hu-berlin.de/forschung/gebiete/wbi/research/projects/aladin/
2 http://www.informatik.hu-berlin.de/forschung/gebiete/wbi

3. Herangehensweise

Die Studienarbeit wird sich in die folgenden Punkte gliedern:

1. **Importieren von Beispieldatenbanken in eine Testumgebung**
   Es sollen vorhandene Datenbanken in eine Testumgebung importiert werden, so dass diese zur Evaluation der benutzten Heuristiken heran gezogen werden können. Dabei werden zwei Arten von Datenbanken unterschieden:
   a) Datenbanken bekannter Struktur zum Testen der Heuristiken:
   - Life Sciences: UniProt, SCOP
   - FilmDBs: Filmdienst, IMDB, Movielens
   - TPC-H

   b) Datenbanken mit unbekannter Struktur zum Testen des Filtergrades auf unbekannten Daten und für Performancemessungen:
   - PDB

2. **Auswertung und Umsetzung vorhandener Heuristiken**
   Die benutzten Heuristiken sollen in einem Java Programm umgesetzt werden. Dieses Programm führt die Heuristiken auf den Inklusionsabhängigkeiten aus und gibt eine Bewertung der einzelnen Inklusionsabhängigkeiten zurück. Diese Bewertung erfolgt durch einen Zahlenwert, der die Höhe der Wahrscheinlichkeit wiedergibt, dass eine Inklusionsabhängigkeit ein Fremdschlüssel ist.

   Heuristiken die umgesetzt werden sollen sind:
   - Der potentielle Fremdschlüssel enthält mindestens k Werte. Dabei sollen Fremdschlüsselkandidaten ausgeschlossen werden, die nur aus wenigen Werten bestehen, jedoch auf ein Attribut mit vielen verschiedenen Werten verweisen. Ein sinnvoller Schwellwert für k soll dabei gefunden werden.
   - Werte im Fremdschlüssel überdecken mindestens N% der Werte im Primärschlüssel der verknüpften Tabelle. Sollte es nur wenige Referenzen vom Fremdschlüssel zum referenzierten Primärschlüssel geben, ist die Verknüpfung sehr wahrscheinlich nicht sinnvoll.
     Dabei soll ein sinnvoller Schwellwert für N gefunden werden.
   - Jedes Attribut darf maximal ein Fremdschlüssel sein und nicht zu mehreren Tabellen eine Verknüpfung darstellen. Eine Verknüpfung zu mehreren Tabellen würde einen Fremdschlüssel mit hoher Wahrscheinlichkeit ausschließen.

3. **Finden weiterer Heuristiken**
   Es sollen Möglichkeiten gefunden werden, Fremdschlüsselbeziehungen anhand anderer Heuristiken auszuwerten
   Diese neu zu erarbeiteten Heuristiken sollen in das vorher erstellte Java Programm integriert und dort zur Auswertung zur Verfügung gestellt werden.

4. **Erstellung der Dokumentation**
   Im letzten Schritt erfolgt die Erstellung einer Dokumentation der erarbeiteten Ergebnisse. Dabei sollen die Herangehensweise und spätere Ergebnisse detailliert beschrieben werden. Ziel ist es, einen Überblick über die gefunden Heuristiken zu verschaffen und deren Vor- und Nachteile aufzuzeigen.

5. **Quellen**

6. [1] J. Bauckmann, U.Leser, F.Naumann, V.Tietz. Efficiently Detecting Inclusion Dependencies. International Conference on Data Engineering (ICDE 2007), Istanbul, Turkey
7. [2] J. Bauckmann. Automatically Integrating Life Science Data Sources. VLDB 2007 PhD Workshop, Vienna, Austria.