

Integration der Human Phenotype Ontology (HPO) in ein medizinisches Forschungsnetz

Lukas Welte

Inhalt

1. Motivation
2. Datenerfassung in der medizinischen Forschung
3. Anforderungen
4. Implementierung
5. Fazit

1. Motivation

- Standardisierung durch HPO
- Notizen werden gemacht
→ potentielle Daten
- Kein Reviewprozess der Notizen
→ Informationsverlust

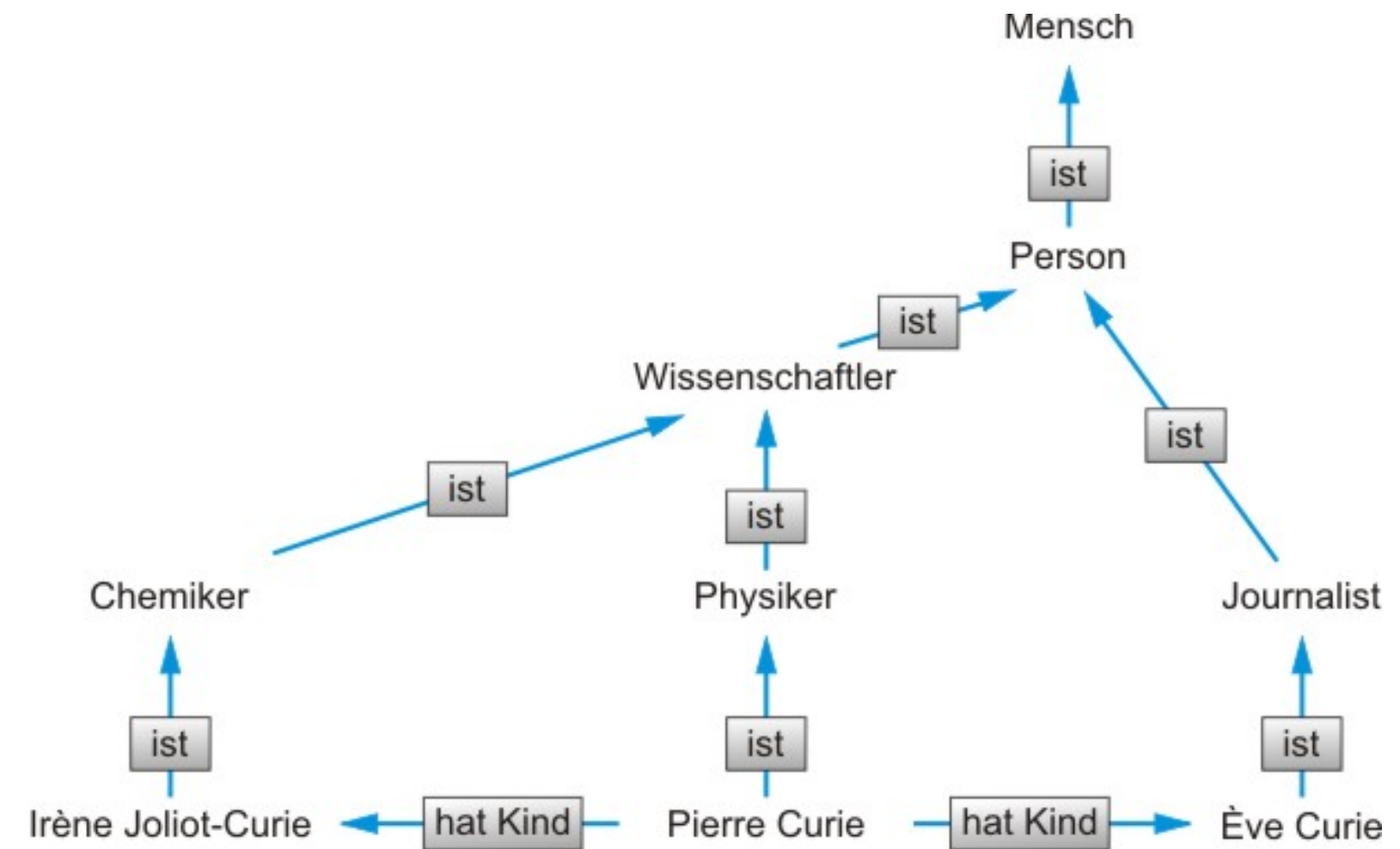
2. Datenerfassung in der medizinischen Forschung

2.1 Strukturierte Datenerfassung

- Schematische Daten
- Je mehr Schema desto mehr Struktur
- Verbessert Wiederverwend- und Maschninenverarbeitbarkeit
- z.B. Relationale Datenbanken, Fragebögen

2.1.1 Ontologie

- Philosophie: Einteilung des Seienden und der Möglichkeit
- Informatik: Spezifizierung einer Konzeptionalisierung
 - Teilt Entitäten in Begriffe und Relationen



2.1.1 Ontologie - Human Phenotype Ontology

- Phänotypische Abnormalitäten
- Integriert vorhandene Ontologien
- mehr als 11000 Terme
- über 115000 Annotationen

2.1.2 Natural Language Processing

- Maschinelle Verarbeitung natürlicher Sprache
- Umwandlung von Freitexten in strukturierte Daten

2.1.2 Natural Language Processing - Funktion

1. Spracherkennung
2. Tokenisierung
3. Morphologische Analyse
4. Syntaktische Analyse
5. Semantische Analyse
6. Dialog und Diskursanalyse

2.2 Freitext Datenerfassung

- Gegenteil der Strukturierten Datenerfassung
- Einfache Erfassung
- Schwere Auswertung



3. Anforderungen

3. Anforderungen

Annahme:

- Visiten der Patienten sind vorhanden
- Visiten enthalten u.a. Symptome und Freitext

3.1 Visitenbrowser

- Einfacher und schneller Zugriff auf Visiten
- Basisinformationen einer Visite

3.2 Visiten Detail

- Überblick über gefundene Terme
- Zusatzinformationen zu Termen
- Löschen eines Terms
- Manuelles Ergänzen von Termen

3.3 Visiten Editor

- Zuordnen von Wörtern zu einem Term
- Erstellen in der HPO nicht vorhandener Terme

3.4 Daten Auswertung

- Anfallende Daten können ausgewertet werden
- Persistierung in statistisch auswertbarem Format

3.5 Deidentifizierung der Daten

- Ersetzen aller Namen durch `[patient]`
- Erleichtert weitergabe der Texte an Dritte

3.6 Daten Integrität

- Nur lesender Zugriff auf HPO und Klinik Datenbank
- Ermöglicht Plug and Play Applikation

4. Implementierung

4.1 Technologiestack

- JSF
- Glassfish
- MySQL
- OpenNLP

4.2 Termsuche

1. Filtern von Elementen
2. Gruppierung von Elementen
3. Suche in der HPO
4. Trefferauswertung

4.2.1 Filtern von Elementen

Heute morgen hatte der Patient starkes Nasenbluten und seine Hand zuckte.

4.2.1 Filtern von Elementen

~~Heute morgen hatte~~ der Patient starkes Nasenbluten
~~und~~ seine Hand zuckte.

4.2.2 Gruppierung von Elementen

- der Patient starkes Nasenbluten
- seine Hand zuckte

4.2.3 Suche in der HPO

— seine Hand zuckte

— seine

— hand

— zuckte

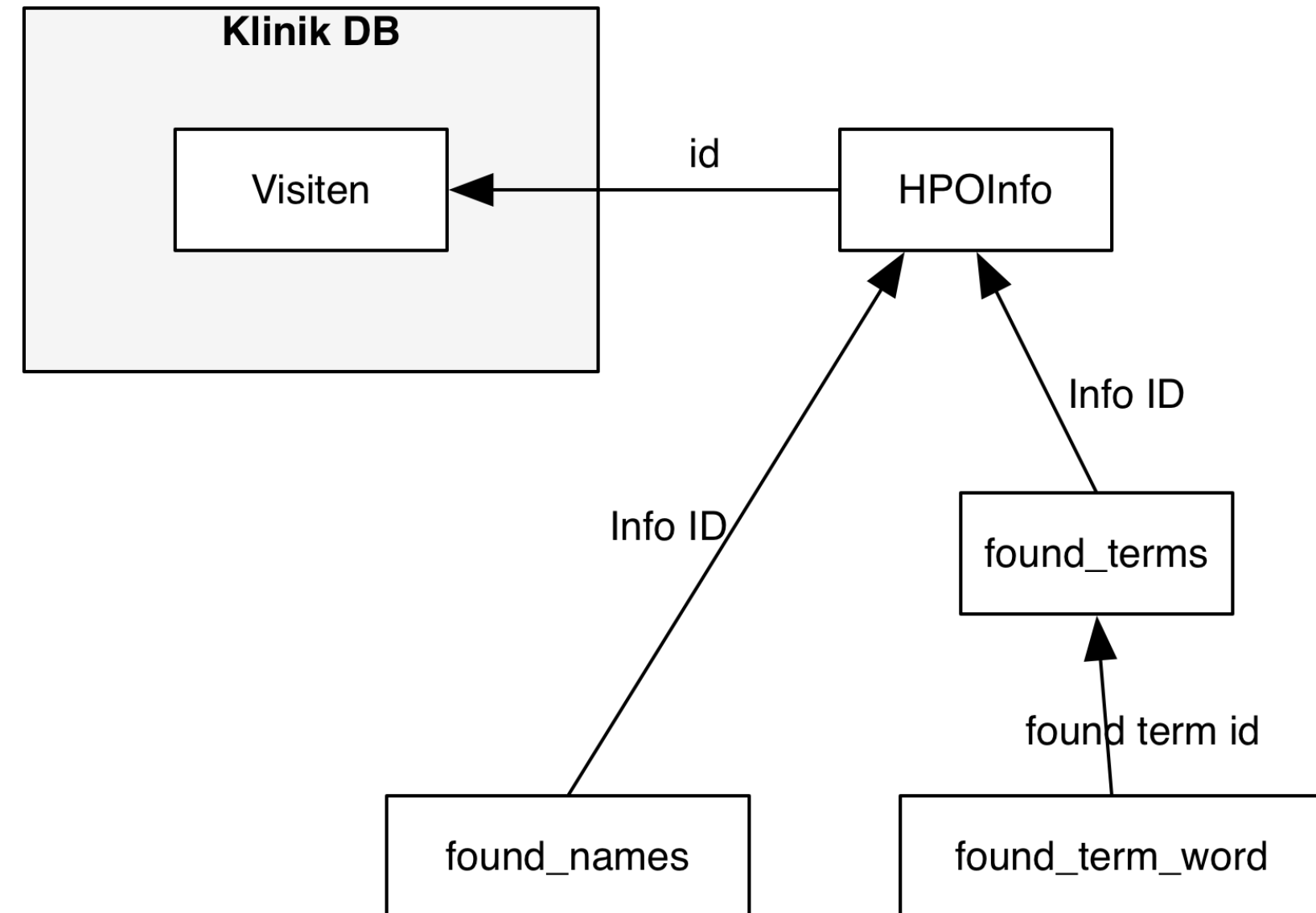
→ Alle Permutationen

4.2.4 Trefferauswertung

- Qualität bestimmt durch die Anzahl der Wörter in der Suche, in Relation mit Anzahl der im Term passenden Wörtern

4.3 Persistierung

- In Programmeigener Datenbank gespeichert
- Ein Datensatz (HPOInfo) je Analysedurchlauf



4.4 Demo

4.5 Probleme

- OpenNLP Erkennungs Modelle
- RAM Verbrauch
- OpenNLP Verarbeitungszeit
- Erkennung von Namen
- Datenbank Abfragen

5. Fazit

5.1 Ergebnis

- Auswertung bestehender Daten
- Standardisierung der Daten
- HPO könnte schnell weiterentwickelt werden

5.2 Verbesserungsmöglichkeiten

- Optimierung der Datenbank Abfragen
- Verwendung einer verteilten Datenbank optimiert für die Suche (ElasticSearch)
- Konfigurierbarer gestalten
- Auch ohne UI verwendbar machen

5.3 Ausblick

- Automatische Diagnoseerstellung aus Freitext
- Präzise Epidemie Auswertung

Vielen Dank für ihre Aufmerksamkeit

Referenzen

haken.jpg

ontologiebsp.jpg

freitext.jpg

zettelhaufen.jpg