Bachelorarbeit am Institut für Informatik der Freien Universität Berlin

Human-Centered Computing (HCC)

Implementierung einer Software zum regelmäßigen Crawling der DFG Förderungsdaten unter besonderer Berücksichtigung der Datenqualität

Daniel Spaude

Betreuerin und Erstgutachterin: Prof. Dr. C. Müller-Birn

Zweitgutachter: Prof. Dr. Lutz Prechelt

Berlin, 28.10.2018

b

Eidesstattliche Erklärung

Ich versichere hiermit an Eides Statt, dass diese Arbeit von niemand anderem als meiner Person verfasst worden ist. Alle verwendeten Hilfsmittel wie Berichte, Bücher, Internetseiten oder ähnliches sind im Literaturverzeichnis angegeben, Zitate aus fremden Arbeiten sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

Berlin, den 29. Oktober 2018

Daniel Spaude

Zusammenfassung

Die Deutsche Forschungsgemeinschaft (DFG) als der bedeutendste Drittmittelgeber in Deutschland vergibt regelmässig Förderungen an in der Wissenschaft tätige Personen und Institutionen. Über die Webanwendung GEPRIS[1] stellt die DFG Informationen über die von ihr genehmigten Förderungsprojekte bereit, unter anderem zu dem Förderungszeitraum, der Disziplin, dem Forschungsthema und den beteiligten Personen und Institutionen. Aus diesen Daten lassen sich nicht nur Erkenntnisse zur Förderungspraxis der DFG gewinnen, sondern auch Einsichten in weitere Fragestellungen wie wissenschaftliche Kooperationen und der Grad an Vernetzung innerhalb der wissenschaftlichen Gemeinschaft, historische Entwicklungen wie Bedeutungszuwachs oder -verlust verschiedener Disziplinen und Institutionen oder auch die geografische Verteilung von wissenschaftlichen Aktivitäten.

Die bereitgestellten Daten sind bis heute lediglich über die Benutzerschnittstelle der GEPRIS-Anwendung verfügbar, es existiert keine offizielle öffentliche API oder ein Datenbank-Export, über welche man die Daten strukturiert konsumieren und automatisiert verarbeiten könnte. Das macht eine tiefergehende Arbeit mit den Daten je nach Fragestellung äußerst schwer bis unmöglich.

Frühere Bachelorarbeiten, welche vom gleichen Lehrstuhl betreut wurden, haben sich mit dieser Problematik bereits befasst und eine Webanwendung entwickelt[2] bzw. optimiert[3], welche einerseits die Daten der Gepris über einen Web-Scraper automatisiert erfasst und strukturiert ablegt und andererseits ein erstes Visualisierungsszenario abdeckt.

Der Fokus der Arbeiten lag auf der Beschaffung der Daten und der Umsetzung und Optimierung eines ersten interaktiven Visualisierungsszenarios, eine hinreichende systematische Auseinandersetzung hinsichtlich der Datenqualität fand nicht statt.

Die vorliegende Arbeit hat als Ziel, eine leicht in Betrieb zu nehmende Neuentwicklung der Crawler-Lösung zu implementieren, welche den Fokus nicht auf Visualisierungskonzepte, sondern die Datenqualität legt. Dabei steht vor allem die Identifizierung geeigneter Datenqualitätskriterien und die Beschreibung geeigneter Konzepte zu deren Messung und Einhaltung hinsichtlich der vom Crawler gewonnen Daten im Vordergrund. Soweit im Rahmen der Arbeit machbar, sollen dann die wichtigsten dieser Konzepte implementiert werden.

[1] `http://gepris.dfg.de/gepris/OCTOPUS`

[2] Stefan Rolfs, Entwicklung und Evaluation einer interaktiven Visualisierung der DFG-Förderprojekte unter Einbeziehung von Domänen-Experten, 2013

[3] Thomas Büttner, Performance-Optimierung einer O/R-Mapper basierten Webanwendung, 2015

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Der zu dieser Arbeit zugehörige Quellcode ist unter folgender URL abrufbar: `https://github.com/spaudanjo/ba-gepris-crawler`

1.1 Thema und Zielsetzung der Arbeit

Die Deutsche Forschungsgemeinschaft (DFG) als der bedeutendste Drittmittelgeber in Deutschland vergibt regelmässig Förderungen an in der Wissenschaft tätige Personen und Institutionen. Über die Webanwendung GEPRIS[1] stellt die DFG Informationen über die von ihr genehmigten Förderungsprojekte bereit, unter anderem zu dem Förderungszeitraum, der Disziplin, dem Forschungsthema und den beteiligten Personen und Institutionen. Aus diesen Daten lassen sich nicht nur Erkenntnisse zur Förderungspraxis der DFG gewinnen, sondern auch Einsichten in weitere Fragestellungen wie wissenschaftliche Kooperationen und der Grad an Vernetzung innerhalb der wissenschaftlichen Gemeinschaft, historische Entwicklungen wie Bedeutungszuwachs oder -verlust verschiedener Disziplinen und Institutionen oder auch die geografische Verteilung von wissenschaftlichen Aktivitäten.

Die bereitgestellten Daten sind bis heute lediglich über die Benutzerschnittstelle der GEPRIS-Anwendung verfügbar, es existiert keine offizielle öffentliche API oder ein Datenbank-Export, über welche man die Daten strukturiert konsumieren und automatisiert verarbeiten könnte. Das macht eine tiefergehende Arbeit mit den Daten je nach Fragestellung äußerst schwer bis unmöglich.

Frühere Bachelorarbeiten, welche vom gleichen Lehrstuhl betreut wurden, haben sich mit dieser Problematik bereits befasst und eine Webanwendung entwickelt[2] bzw. optimiert[3], welche einerseits die Daten der Gepris über einen Web-Scraper automatisiert erfasst und strukturiert ablegt und andererseits ein erstes Visualisierungsszenarios abdeckt.

Der Fokus der Arbeiten lag auf der Beschaffung der Daten und der Umsetzung und Optimierung eines ersten interaktiven Visualisierungsszenarios, eine hinreichende systematische Auseinandersetzung hinsichtlich der Datenqualität fand nicht statt.

Die vorliegende Arbeit hat als Ziel, eine leicht in Betrieb zu nehmende Neuentwicklung der Crawler-Lösung zu implementieren, welche den Fokus nicht auf Visualisierungskonzepte, sondern die Datenqualität legt.

[1] `http://gepris.dfg.de/gepris/OCTOPUS`

[2] Stefan Rolfs, Entwicklung und Evaluation einer interaktiven Visualisierung der DFG-Förderprojekte unter Einbeziehung von Domänen-Experten, 2013

[3] Thomas Büttner, Performance-Optimierung einer O/R-Mapper basierten Webanwendung, 2015

1.2 Vorgehen bei der Umsetzung

Die zwei Schwerpunkte bei der Umsetzung sind zum einen die Entwicklung der Crawler-Lösung an sich. Dafür ist einerseits eine Identifikation der zu erfassenden Resourcentypen bzw. HTML-Seiten seitens des Gepris-Systems nötig, andererseits müssen geeignete Ansätze zur Extraktion und Bereitstellung der Daten identifiziert werden. Es muss geeignete Technologie wie die Programmiersprache oder das Persistenzformat ausgewählt werden und es müssen Besonderheiten wie vom Gepris-System genutzte Cookie-basierte Sessionkonzepte beim Crawling berücksichtig werden

Der andere Schwerpunkt, die Sicherstellung der Datenqualität, erfordert vor allem die Identifizierung geeigneter Datenqualitätskriterien und die Beschreibung geeigneter Konzepte zu deren Messung und Einhaltung hinsichtlich der vom Crawler gewonnen Daten. Um möglichst systematisch geeignete Kriterien zu bestimmen, möchte ich bei der Auswahl auf ein Paper zurückgreifen, welches ein ähnliches Anliegen, jedoch für eine andere Domäne verfolgt, nämlich die Bestimmung von Datenqualitätskriterien für Knowledge Graphen im Bereich Linked Open Data. Soweit im Rahmen der Arbeit machbar, sollen dann möglichst viele dieser Kriterien und Konzepte zur Messung und Einhaltung implementiert werden.

1.3 Aufbau der Arbeit

Zunächst soll ein Einführung in die Domäne, das heisst in die Besonderheiten der von der DFG bereitgestellten Informationen bezüglich ihrer Förderungsaktivitäten, und in die Webanwendung Gepris gegeben werden: Welche Konzepte und Resourcen sind hier zentral? Welche Möglichkeiten bietet des System derzeit, wo liegen seine Grenzen und was sind mögliche interessante Forschungsanfragen an die Datenbasis?

Anschliessend möchte ich mich dem Thema der Datenqualität widmen, zuerst mit einer allgemein Begriffseinführung, dann mit einer systematischen Bestimmung geeigneter Qualitätskriterien für unsere Domäne, inklusive der Skizzierung von Konzepten zur Messung und Sicherstellung. Zwar ist für die Entwicklung des Crawlers die meiste Zeit vorgesehen, im Rahmen der Ausarbeitung wird aber insbesondere auf der gerade erwähnten Bestimmung der Qualitätskriterien der Fokus liegen.

Im Kapitel über den Crawler möchte ich Einblicke in die Umsetzung Crawling-Lösung geben und beschreiben, welche Konzepte und Technologien sie verwendet, im daran anschliessenden Kapitel will ich die Ansätze zur Messung der Datenqualität mittels R beschreiben, woraufhin ein Kapitel zur Auswertung, vor allem hinsichtlich Umfang und Datenqualität eines Crawling-Durchlaufes, folgt.

Abschliessen soll die Arbeit mit einer Zusammenfassung der Ergebnisse, einer Beschreibung gewonnener Einsichten und erkannter Probleme, sowohl hin-

sichtlich der gewonnen Datenbasis, als auch bezüglich der Implementierung und meinem Vorgehen allgemein, sowie ein Ausblick auf mögliche weitere Schritte, wie weitere Verbesserungen an der Umsetzung.

Für die im Folgenden verwendeten personenbezogenen Ausdrücke wurde, um die Lesbarkeit der Arbeit zu erhöhen, die männliche Schreibweise gewählt. Des Weiteren werden eine Reihe von englischen Bezeichnungen verwendet, um einerseits dem interessierten Leser das Studium der häufig vorliegenden englischen Originalliteratur zu erleichtern oder andererseits bestehende Fachbegriffe nicht durch die Übersetzung zu verfälschen. Bei Verweisen auf Internetquellen ist zu beachten, dass die Informationen auf die ich Bezug genommen habe, zum Zeitpunkt der Fertigstellung dieser Arbeit verfügbar waren. Über die Webseite archive.org können in den meisten Fällen ältere Versionen der entsprechenden Seiten abgerufen werden, sollte sich deren Inhalt zwischenzeitlich geändert haben.

2 Die Domäne der Gepris-Daten

Worum geht es bei der Gepris-Anwendung? Gepris steht für "Geförderte Projekte der DFG". Mit ihr "stellt die DFG eine Datenbank im Internet bereit, die über laufende und abgeschlossene Forschungsvorhaben der DFG informiert."[1]

Bevor die Anwendung etwas näher betrachtet wird, möchte ich kurz auf die Deutsche Forschungsgemeinschaft und ihre Förderungs- und Informationspolitik eingehen. Für das Datenmodell insbesondere relevant sind hier die Konzepte der Fachsystematik und der DFG-Programme bzw. DFG-Verfahren.

2.1 Die Deutsche Forschungsgemeinschaft und ihre Informationspolitik bzgl. ihrer Förderungsaktivitäten

Die Deutsche Forschungsgemeinschaft, kurz DFG, ist der größte Drittmittelgeber innerhalb der Wissenschaftsgemeinschafts in Deutschland. Allein im "Jahr 2017 förderte die DFG knapp 32 500 Projekte mit einer jahresbezogenen Bewilligungssumme von 3,2 Milliarden Euro"[2]

Sie finanziert sich fast vollständig durch öffentliche Gelder durch Bund und Länder [3]. Neben der finanziellen Unterstützung versteht sie sich auch als unabhängige Organisations- und Vernetzunginstanz innerhalb der wissenschaftlichen Gemeinschaft. Mit Letzterem ist im Übrigen auch ein wesentlicher Interessenschwerpunkt bezüglich der DFG-Daten berührt: datengestützte Einsichten bezüglich der Kooperationsdynamik und Vernetztungintensitivität innerhalb der deutschen Forschungslandschaft.

Die DFG stellt recht umfassende Berichte zu ihren Förderungsaktivitäten bereit, zum Beispiel in Form von Jahresberichten, statistischen Übersichten[4] oder im Rahmen der Publikation "Förderatlas 2018"[5]. Dabei handelt es sich allerdings fast ausnahmslos um aggregierte, statistische Daten. Obwohl durch öffentliche Gelder finanziert, stellt die DFG bis heute meines Wissens nach keine detaillierten Datenbestände auf Projektebene über ihre Förderungsaktivitäten im Rahmen einer API oder strukturiert abgelegter Datenbankexporte öffentlich und offiziell zur Verfügung, welche umfangreiche, individuell gesteuerte Analysen ermöglichen würden. Die projektzentrierte Informationsanwendung Gepris ist ausschliesslich über eine HTML-Schnittstelle für Endnutzer durchsuchbar, was tiefergehende Explorations- und Analyseszenarien schwer

[1] http://gepris.dfg.de/gepris/OCTOPUS
[2] http://www.dfg.de/gefoerderte_projekte
[3] http://www.dfg.de/download/pdf/dfg_im_profil/geschaeftsstelle/publikationen/dfg_jb2017.pdf
[4] http://www.dfg.de/dfg_profil/zahlen_fakten/statistik/index.html|
[5] http://www.dfg.de/sites/foerderatlas2018/

bis unmöglich macht und was die bereits erwähnte Motivation für die Entwicklung der Crawler-Anwendung darstellt. Man könnte somit sagen, dass die Gepris-Daten einen gefundenen, aber bisher ungehobenen Schatz für solche tiefergehenden Analysen darstellen.

Zwei wesentliche Konzepte innerhalb der Domäne "Förderungsaktivitäten der DFG" stellen die DFG-Fachsystematik und die DFG-Programme bzw. DFG-Verfahren dar, auf die nun eingegangen werden soll.

2.2 Die DFG-Fachsystematik

Die DFG organisiert sich hinsichtlich der fachlichen Zuordnung von Projekten bei ihren Förderungsaktivitäten in einer vierstufigen Systematik, welche näher unter [6] und [7] beschrieben ist und auf deren Stufen hier kurz eingegangen werden soll. Ein wesentlicher Grund für die recht ausdifferenzierte Systematik ist laut DFG die interne operative Organisation, zum Beispiel hinsichtlich der Frage, welcher DFG-Sachbearbeiter für welche Projektanträge und welche Gremien für Bewilligungen und Evaluationen von Projekten verantwortlich ist[8].

1. Stufe: Wissenschaftsbereiche (Scientific Discipline) Auf der obersten Ebene stehen dabei die Wissenschaftsbereiche (Scientific Discipline), von denen es insgesamt vier Stück gibt:

- Geistes- und Sozialwissenschaften
- Lebenswissenschaften
- Naturwissenschaften
- Ingenieurwissenschaften

2. Stufe: Fachgebiete (Research Area) Die nächste Stufe sind die Fachgebiete, wovon es 14 Stück gibt:

- Agrar-, Forstwissenschaften und Tiermedizin
- Bauwesen und Architektur
- Biologie
- Chemie

[6] http://www.dfg.de/en/dfg_profile/statutory_bodies/review_boards/subject_areas/index.jsp

[7] http://www.dfg.de/download/pdf/dfg_im_profil/gremien/fachkollegien/amtsperiode_2016_2019/fachsystematik_2016-2019_de_grafik.pdf

[8] http://www.dfg.de/download/pdf/dfg_im_profil/zahlen_fakten/programm_evaluation/bericht_begutachtungswesen.pdf

- Geisteswissenschaften
- Geowissenschaften
- Informatik, System- und Elektrotechnik
- Maschinenbau und Produktionstechnik
- Materialwissenschaft und Werkstofftechnik
- Mathematik
- Medizin
- Physik
- Sozial- und Verhaltenswissenschaften
- Wärmetechnik/Verfahrenstechnik

3. Stufe: Fachkollegium (Review Board) Dies stellt die vorletzte Ebene dar, welche die einzelnen Fächer (Subject Areas) gruppiert. Es gibt insgesamt 48 Fachkollegien. Beispiele hierfür sind 'Geschichtswissenschaften', 'Zoologie', 'Wasserforschung' oder 'Informatik'.

4. Stufe: Fach (Subject Area) Auf der untersten Stufe sind schlussendlich die einzelnen Fächer aufgelistet. Es gibt insgesamt 213 Fächer und Beispiele sind 'Softwaretechnik und Programmiersprachen', 'Physische Geographie', 'Systemische Neurowissenschaft, Computational Neuroscience, Verhalten' oder 'Theater- und Medienwissenschaften'.

2.3 Die DFG-Programme/DFG-Verfahren

Die DFG bietet verschiedene Förderprogramme an, welche sich hinsichtlich personellem Umfang, Laufzeit, Anforderungen an den Antragsteller und anderen Aspekten unterscheiden [9].

Auch im Falle der DFG-Verfahren wurde wieder eine mehrstufige Organisation verwendet, aufgeteilt in die folgenden drei Ebenen:

- Programmgruppe (Programme Group) - Beispielsweise 'Einzelförderungen'
- Programm (Programme) - Beispielsweise 'Heisenberg-Programm'
- Programmlinie (Programme line) - Beispielsweise 'Heisenberg-Professuren'

Insgesamt wurden im aktuellen Crawl 35 Programmlinien gefunden.

[9] Eine Übersicht über die verschiedenen Verfahrenstypen findet sich unter `http://www.dfg.de/en/research_funding/programmes/index.html`

2.4 Bereitgestellte Ressourcen und Felder

Die primären Resourcentypen, welche innerhalb der Gepris-Anwendung und auch in meiner zu implementierenden Crawling-Lösung im Fokus stehen, sind Projekte, Personen und Institutionen. Der Typ 'Projekt' nimmt dabei meiner Ansicht nach wiederum die zentralste Rolle in dieser Triade ein, denn sie sind, wenn man so möchte, auch die operationelle Einheit innerhalb der Förderungspraxis der DFG. Für Projekte werden Anträge gestellt, sie haben ein definiertes Forschungsvorhaben und einen Förderungszeitraum, einen Budgetrahmen, sie müssen bewilligt und ausgewertet werden. Projekte stehen zum einen in Verbindung zu Personen, welche verschiedene Rollen einnehmen können, zum Beispiel als teilnehmende Personen, Wissenschaftler, Kooperationspartner oder, insbesondere bei Projektgruppen, leitende Funktionen wie Projektleiter oder Sprecher. In den meisten Fällen gibt es einen dezidierten Antragsteller oder einen ehemaligen Antragsteller. Zum anderen stehen Projekte in Verbindung zu Institutionen, auch hier wieder in Rollen wie der antragstellenden oder teilnehmenden Institution.

Im Vergleich zu Personen und Institutionen, welche bezüglich der verwendeten Datenfeldtypen wie Name, Dienstanschrift, Telefon etc. sowie einer Auflistung der mit der jeweiligen Person/Institution in Verbindung stehenden Projekte, weisen Projekte hinsichtlich ihrer schematischen Ausprägung eine höhere Variabilität auf. Angelehnt an die Terminologie der objektorientierten Programmierung könnte man sagen, sie sind "polymorph". Abhängig insbesondere vom jeweiligen DFG-Verfahren unterscheiden sich bei Projekten die bereitgestellten Feldtypen teilweise erheblich. Anhand der folgenden drei Tabellen möchte ich nun für den Resourcentyp 'Projekte' auflisten, welche Feldtypen allgemein Verwendung finden. Ich beziehe mich dabei, wie im Folgenden des Öfteren, auf die englischsprachige Version der Gepris-Anwendung, da ich, auf Wunsch eines nicht-deutschsprachigen Mitgliedes des betreuenden Lehrstuhles und weil ich die Datenbasis generell möglichst vielen Interessierten zugänglich machen möchte, innerhalb des Entwicklungsprozesses von der deutschen auf die englische Version bezüglich der Voranalyse und des Crawlings gewechselt bin.

Im Falle der letzten beiden der drei Tabellen existiert eine Spalte 'Variante'. Diese enthält gefundende Varianten bezüglich der Schreibweise von in semantischer Hinsicht ein und demselben Feld. Einige dieser Varianten sind darauf zurückzuführen, dass manchmal Singular und manchmal Plural verwendet wird, je nach Anzahl der genannten Personen bzw. Institutionen des jeweiligen Feldes, andere vermutlich auf inkonsistente Arbeitsabläufe bezüglich der Dateneingabe seitens der DFG. Dies stellt ein besonders zu behandelndes Problem bei der Implementierung des Crawlers dar, welches durch die Übernahme aller Schreibweisen in die für die Extractor-Logik genutzten regulären Ausdrücke gelöst wurde. 2.1 beschreibt dabei die Kernfelder, 2.2 und 2.3 hingegen jene Felder, welche auf Personen bzw. Institutionen verweisen.

Tabelle 2.1: Projekte: Kernfelder

Feldname
DFG Programme
DFG programme contact
Instrumentation Group
International Connection
Major Instrumentation
Participating subject areas
Participating university
Project Description
Project identifier
Subject Area
Subproject of
Term
Website

Erwähnenswert erscheint mir ein Hinweis bezüglich des Feldes 'Subproject of': einige DFG-Verfahren haben den Charakter von Cluster-Projekten (prominentes Beispiel sind die Sonderforschungsbereiche), welche auf eine Liste von Unterprojekten verweisen. Handelt es sich um ein Unterprojekt, wird über dieses Feld der Verweis zum Eltern-/Cluster-Projekt gemacht.

2.5 Möglichkeiten und Grenzen der Gepris-Plattform

Um einen Eindruck von den Möglichkeiten und Grenzen der Gepris-Plattform aus Nutzersicht zu erlangen, möchte ich eine Beispielanfrage skizzieren, in deren Zuge sich schnell weitergehende Fragestellungen auftun, die mit den derzeitigen Möglichkeiten der Gepris-Anwendung schwer oder gar nicht zu beantworten sind.

Nehmen wir an, mein Ausgangsinteresse besteht darin, einen Überblick über alle Einzelförderungen aus dem Fachgebiet 'Informatik-, System- und Elektrotechnik' im Bundesland Berlin zu bekommen, welche ab dem Jahr 2010 gefördert wurden.

Ich starte von der Homepage der Gepris aus (`http://gepris.dfg.de`) und navigiere von dort aus zum Katalog, welcher mir umfangreiche Filteroptionen bereitstellt. In der nachfolgenden Katalogseite werden mir die gefundenen Projekte entsprechend angezeigt und ich kann bei Bedarf die jeweiligen Detailseiten aufrufen, von dort zu in Beziehung stehenden Institutionen und Personen navigieren, von deren Detailseiten ich wiederum zu anderen in Beziehungen stehenden Projekten navigieren kann.

Es wird schnell deutlich, dass man die Domäne als Graphen oder spezifischer auch als soziales Netzwerk auffassen kann, wobei je nach Modellierungsansatz

Tabelle 2.2: Projekte: Felder mit Personenbezug und Namensvariationen

Feldname	**Variante**
Applicants	Applicants
	Applicant
Co-Applicants	Co-Applicants
	Co-Applicant
	Co-applicant
	Co-applicants
Cooperation partners	Cooperation partners
	Cooperation partner
Deputy spokespeople	Deputy spokespeople
	Deputy spokesperson
Foreign spokespeople	Foreign spokespeople
	Foreign spokesperson
Former applicants	Former applicants
	Former applicant
Heads	Heads
	Head
International Co-Applicants	International Co-Applicants
	International Co-Applicant
Participating Persons	Participating Persons
	Participating Person
Participating scientists	Participating scientists
	Participating scientist
Project leaders	Project leaders
	Project leader
Spokespersons	Spokesperson
	Spokespersons

Tabelle 2.3: Projekte: Felder mit Institutionsbezug und Namensvariationen

Feldname	**Variante**
Applying institution	Applying institution
Co-applicant institution	Co-applicant institution
Foreign institution	Foreign institution
Participating Institution	Participating Institution
	Participating institution
Participating university	Participating university
Partner organisation	Partner organisation

zum Beispiel Personen die Knoten die Mitarbeit am gleichen Projekte zwischen zwei Personen (also eine Kooperation) eine Kante darstellen kann. Wür-

de der Gepris-Datenbestand als strukturierter Datenbankexport zur Verfügung stehen, könnte ich nun, entsprechende Expertise vorausgesetzt, zum Beispiel mittels des R-Paketes 'igraph' entsprechende Netzwerkmodellierungen mit gewichteten Kanten und Analysen auf diesem Modell vornehmen oder die Daten in eine Graphdatenbank wie Neo4J importieren. So ließen sich zum Beispiel Fragen beantworten wie:

"Was sind die kürzesten Pfade zwischen den Personen Elfriede Fehr und Lutz Prechelt im Kooperationsgraphen im Zeitraum zwischen 2010 und 2015, wobei nur externe, nicht bei der Freien Universität Berlin arbeitende Personen als Zwischenknoten erlaubt sind?"

Eine andere, eher statistische Fragestellung, welche im Rahmen der ursprünglichen Gepris-Anfrage auftauchen könnte, wäre:

"Erweitere die Projektauswahl auch auf die Regionen Bayern, Hessen und Hamburg, gewichte alle Projekte anhand der Anzahl der beteiligten Wissenschaftler und zeige mir eine nach Regionen gruppierte Visualisierung der Verteilung von besonders schwergewichtigen Projekten". Auch dieses Szenario wäre mit entsprechenden Kenntnissen in einer Programmierumgebung wie 'R' mit realistischem Zeitaufwand umsetzbar, würde der DFG-Datenbestand als strukturierter Gesamtdatensatz zur Verfügung stehen, jedoch nicht mit der aktuellen Gepris-Anwendung. Natürlich müssen bei der Beurteilung der Ergebnisse solcher Berechnungen diverse Aspekte mit beachtet werden, welche die statistischen Berechnungen hinsichtlich der Korrektheit bezüglich den realen Begebenheiten der Projekte beeinflussen und potentiell verfälschen könnten, vor allem was unbekannte Besonderheiten bezüglich der Dateneingabe seitens der DFG betrifft (beispielsweise ungenaue Angaben und von Projekt zu Projekt verschiedene Entscheidungskriterien, ab wann eine Person als "beteiligter Wissenschaftler" für ein Projekt mit aufgeführt werden soll). Nichtsdestotrotz tut sich eine große Anzahl von interessanten Explorationsszenarien mit Hypothesenentwicklungen auf, welche dann in weiteren Schritten verfeinert und durch das Heranziehen weiterer Validierungsschritte fortgeführt werden könnten.

3 Fokus der Arbeit: Messung und Einhaltung der Datenqualität

Die Erfahrungen mit der früheren Implementierung des Gepris-Crawlers haben Mängel hinsichtlich der Datenqualität, insbesondere hinsichtlich der logischen Konsistenz und Vollständigkeit der vom Crawler bereitgestellten Daten erkennen lassen. Als einfaches Beispiel: in der Beziehungstabelle *project_person* des Crawling-Ergebnisses der früheren Version, welche Projekte und Personen miteinander in Beziehung setzt, wurden 62 Einträge gefunden, für die kein zugehöriger Wert in der Tabelle *person* existiert.

```
SELECT COUNT(DISTINCT pp.project)
FROM project_person pp
LEFT OUTER JOIN person per
ON pp.person = per.id
WHERE per.id IS NULL;

RESULT: 62
```

Die Sicherstellung insbesondere dieser fundamentalen Datenqualitäts-Kriterien ist jedoch einer der entscheidendsten Faktoren hinsichtlich der Benutzbarkeit des Systems, denn damit steht und fällt die Qualität eines Anfrageszenarios an die Domäne. Daher soll auf der Datenqualität und hier insbesondere auf der Frage, wie diese möglichst effizient und automatisch garantiert werden kann ein besonderer Fokus liegen. Wie im Verlauf des Kapitels dargelegt wird, hat die Informatik hier in den letzten Jahren den Begriff der Datenqualität systematisiert, beispielsweise durch die Ausdifferenzierung in eine Reihe von Datenqualitäts-Dimensionen und -Kriterien. Bevor diese vorgestellt und durchgesehen und auf Anwendbarkeit auf die Domäne Gepris hin geprüft werden, soll jedoch kurz auf den Begriff der Datenqualität an sich eingegangen werden.

3.1 Was ist Datenqualität?

Es soll zuerst kurz auf die Basisbegriffe “Daten” und “Qualität” des Kompositums “Datenqualität” eingegangen werden, wobei die allgemeinen Erläuterungen der Begriffe jeweils direkt auf die Domäne Gepris übertragen werden sollen. Nach der dann folgenden zuerst intuitiven Einführung des Begriffs “Datenqualität”, ebenfalls veranschaulicht anhand der Gepris-Domäne, soll dann eine weitere Konkretisierung des Begriffs erfolgen, so wie ihn die Informatik systematisch in der Literatur in den letzten Jahren ausdifferenziert hat.

3.1.1 Daten

Die intuitiv richtige Auffassung und Verwendung dieses für die Informatik essentiellen Begriffs wird beim Leser zwar angenommen, dennoch möchte ich der Vollständigkeit halber den Duden zitieren. Er definiert Daten als "durch Beobachtungen, Messungen, statistische Erhebungen u. a. gewonnene [Zahlen]werte und formulierbare Befunde"[1]. In unserem Fall sind dies die von der DFG bereitgestellten Angaben zu ihren Förderungs-Aktivitäten. Daten lassen sich grob in die Kategorien strukturiert, semi-strukturiert und unstrukturiert einordnen[2]. Im Falle der Gepris liegen die Daten grundsätzlich semistrukturiert vor, da diese zwar nicht über ein streng strukturiertes und mit einem Schemata versehenes Format wie beispielsweise CSV, JSON oder in Form eines gängigen Datenbankformats von der DFG bereitgestellt werden, sondern über HTML-Seiten, sich andererseits aber die einzelnen Entitäten sehr ähneln, sowohl was die Benennung der einzelnen Felder als auch was die grundlegende HTML-Seitenstruktur angeht. Insbesondere lassen sich eine überschaubare Anzahl von Entitätstypen bestimmen, z.B. auf oberster Ebene Projekte, Personen und Institutionen, auf niedrigerer Ebene eine Ausdifferenzierung von z.B. Projekten in unterschiedliche DFG-Verfahrenstypen mit jeweils ähnlicher Struktur. Anders verhielte es sich, wenn es sich beispielsweise um ausschließlich unstrukturierte Freitexte handeln würde. In diesem Fall müssten vermutlich Techniken aus den Bereichen Text Mining und Natural Language Processing angewandt werden, um systematisch Daten in ähnlicher Größenordnung zu extrahieren und strukturiert abzulegen wie dies das Ziel des zu implementierenden Crawlers ist. Zwar beinhalten beispielsweise die DFG-Projekte auch Felder mit dem Namen "Projektbeschreibung", welche in Freitext, also unstrukturiert, Informationen enthalten. Dies betrifft aber eben nur einen Teil der Datenmasse.

3.1.2 Qualität

Der Begriff "Qualität" beschreibt nach Definition der ISO (International Organization for Standardization) allgemein bei Produkten den "degree to which a set of inherent characteristics (3.10.1) of an object (3.6.1) fulfils requirements"[3]. Diese allgemeine Definition erfährt beispielsweise bei klassischen Produkten Konkretisierung in Angaben zur physischen Beschaffenheit wie Länge, Größe, Gewicht, aber vor allem auch hinsichtlich seiner funktionalen Eigenschaften, inwieweit es den angedachten Nutzen erfüllt, also wie "zweckmäßig" es ist (fitness for purpose) und die Bedürfnisse des Nutzers erfüllt, also ob es "gebrauchstauglich" ist (fitness for use).

[1] `https://www.duden.de/rechtschreibung/Daten`

[2] `https://gi.de/informatiklexikon/semistrukturierte-daten`

[3] ISO 9000:2015(en), Quality management systems — Fundamentals and vocabulary: `https://www.iso.org/obp/ui/#iso:std:iso:9000:ed-4:v1:en`

3.1.3 Daten + Qualität: Datenqualität

So wie auch die Qualität von Softwaresystem beschrieben werden kann - eine ausführliche Behandlung des Themas findet sich zum Beispiel bei [Kan02] - können prinzipiell auch diverse den Datensystemen und Datenartefakten sowie seiner Aufbereitung und Darstellung inhärenten Eigenschaften spezifiziert und gemessen werden. Physische Beschaffenheiten fallen bei der immateriellen Natur hier natürlich weg, aber die Aspekte der Zweckmässigkeit und der Gebrauchstauglichkeit sind auch hier von hoher Bedeutung. Offensichtlich sind zum Beispiel die Fragen, wie gut zugänglich die Daten sind, ob sie verständlich präsentiert sind oder auch ob grundsätzlich die richtigen Objekte bzw. Entitäten und wiederum deren entscheidenden Eigenschaften korrekt gemessen und beschrieben werden. Bei der Domäne DFG/Gepris sind zum Beispiel die Entitätstypen "Projekt" oder "Institution" von zentraler Bedeutung. Die Gepris beschreibt nur eine begrenzte Menge von Eigenschaften, bei den Projekten sind dies zum Beispiel Titel, Förderungszeitraum, Förderungstyp, teilnehmende Wissenschaftler oder auch die antragstellende Institution.[4]

3.1.4 Datenqualität in der Literatur der Informatik

Als ein äußerst richtungsweisendes und eines der am meisten zitiertesten Paper in dem Bereich ist jenes von Wang mit dem Titel "Beyond Accuracy: What Data Quality Means to Data Consumers"[WS96] anzusehen. Es ging von der Annahme aus, dass der bis dahin weit verbreitete Fokus, wenn es um Datenqualität geht, auf Exaktheit (accuracy) vor allem von den Konsumenten von Anwendungen zur Datenbereitstellung meist als zu eng empfunden wird und weitere Dimensionen und Kriterien entscheidend sind für eine umfassende Evaluation und Sicherstellung der Datenqualität von Informationssystemen. Überhaupt rückt das Paper die Rolle und Perspektive des Datennutzers in den Vordergrund und überträgt den Begriff "Fitness of use" aus dem Bereich der Produktqualität aus der Ökonomie, welcher die Wichtigkeit des Konsumenten in den Vordergrund rückt hinsichtlich der Qualitätsbeurteilung, auf den Bereich der Datenqualität. Auf Grundlage einer Umfrage unter den Benutzern von Informationssystemen wurde eine Systematik zur Organisation von Datenqualitätskriterien entwickelt, welche die für Datenkonsumenten entschei-

[4] Da ich bisher keinen Einblick in die interne Arbeitsweise der DFG und deren Projekt Gepris habe, kann ich nur Vermutungen anstellen, doch liegt es nahe, dass diese Daten vermutlich nur zu sehr wenigen einzelnen Zeitpunkten erfasst (beispielsweise sobald ein Förderungsantrag genehmigt wurde) und danach nur teilweise oder gar nicht aktualisiert werden (z.B. wenn eine Förderung beendet wurde oder weitere teilnehmende Wissenschaftler von der DFG zu diesem Projekt erfasst worden sind). Für sehr viele interessante Forschungsfragen bezüglich der DFG-Daten wären aber weitere Eigenschaften interessant und teilweise sogar zwingend nötig. Das monetär ausgedrückte Förderungsvolumen wäre beispielsweise sehr interessant um Schwerpunkte in der DFG-Förderungspolitik zu erkennen. Auch Evaluationsdaten bei abgeschlossenen Projekten, idealerweise nicht nur in Freitext, sondern auch in Form von Metriken, wären sicher interessant.

dend sind.

Wangs Paper, welches 1996 veröffentlicht wurde, war im Bereich klassischer Datenbanksysteme angesiedelt. Aber auch neuere Paper, welche sich beispielsweise mit Datenqualität im Bereich Linked Data und Knowledge Graphen und den spezifischen Eigenheiten und Anforderungen dieser einerseits hinsichtlich der Datenquellen heterogenen, hinsichtlich der Nutzererfahrungen aber möglichst homogenen Domäne beschäftigen, basieren auf Wangs Systematik. Hier wurde das Paper von Färber et al. mit dem Titel "Linked data quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO" [FBMR17] als besonders nützlich empfunden. Zwar ist meine Arbeit und die Domäne Gepris nicht im Bereich Linked Data angesiedelt, die systematische Darstellung der Datenqualitäts-Dimensionen und -Kriterien in diesem Paper bietet sich jedoch auch für die Gepris-Domäne als Grundlage zur Bestimmung geeigneter Kriterien an.

Insgesamt ist eines der Anliegen des recht ausführlichen Papers, eine Systematik zur Bewertung und zum Vergleichen von Open Linked Data Knolwedge Graphen zu entwickeln, nicht nur, aber insbesondere auch hinsichtlich der Datenqualität. Dabei greift es in diesem Rahmen auf die auf Wang zurückgehende Systematik zurück. Genauer gesagt wurde eine erweiterte Version von Wangs Framework genutzt, welche die zusätzlichen Dimensionen consistency, verifiability, offensiveness, licensing und interlinking beinhaltet.

Auffällig ist, dass das Paper zwar viele Zusammenhänge und vor allem die Datenqualitäts-Metriken formal-mathematisch ausdrückt, welches auf den ersten Blick leicht den Anschein einer strikten, objektiven Auswertungsmethode ergibt. Zu beachten ist jedoch, dass sich bei diversen Metriken immer noch sehr viel Komplexität und mathematisch unscharfe Konzepte wie "common sense" oder "trusted source" verbergen, beispielsweise bei der Definition der Funktion semValid, welche bei der Metrik-Formel zur semantischen Validität Verwendung findet[5]. Auf eine streng formale, mathematische Definition von Metriken möchte ich für die Gepris-Domäne verzichten - eine Beschreibung in Worten erscheint mir zweckmäßiger und ausreichend.

Ein Datenqualitätskriterium ist nach Färber "a particular characteristic of data w.r.t. its quality"[6]. Datenqualitätskritierien können nach Färber subjektiver oder objektiver Natur sein. So sind beispielsweise die Glaubwürdigkeit, welche in Verbindung mit der Reputation einer Datenquelle steht, schlecht vollständig objektivierbar[7]. Die Richtigkeit ("accuracy") ist hingegen in den

[5] Siehe [FBMR17], 2.2.1 Accuray, Seite 5

[6] Abschnitt 3 in [FBMR17]

[7] Hier sei angemerkt, dass, auch wenn wir bei unserer Beschäftigung mit der Datenqualität hinsichtlich der DFG-Domäne primär auf Aspekte bezüglich des Crawlingvorgangs und sekundär bezüglich der Gepris-Daten an sich fokussiert sind, es in einem zukünftigen Schritt sinnvoll sein könnte, die Gepris-Daten auch als Knowledge Graphen im Kontext von Open Linked Data aufzufassen und dann dessen allgemeine Kriterien wie Trustworthiness zu beschreiben - insbesondere dann, wenn die Gepris-Daten mit anderen Datenquellen, z.B. mit anderen Knowledge Graphen wie WikiData, integriert werden sollen, denn dann macht ein Vergleich der Kriterien der verschiedenen Datenquellen Sinn.

meisten Fällen, zumindest konzeptionell, leicht objektiv überprüfbar: So kann ich beispielsweise den Förderungsbeginn eines Projektes im Zweifel bei den beteiligten Wissenschaftlern nachfragen und mit den DFG-Angaben abgleichen.

3.2 Der Crawler und die DFG: die zwei grundlegenden Instanzen hinsichtlich der Datenqualität

Ohne schon eine ausdifferenzierte Systematik zur Beschreibung und Unterscheidung einzelner Datenqualitätseigenschaften anzuführen, wird bereits deutlich, dass bei der Betrachtung und Sicherstellung der Datenqualität in zwei Ebenen unterschieden werden kann: die Crawling-Komponente und die Ursprungsdaten wie sie von der DFG über das Gepris-System bereitgestellt werden. Da es für den Benutzer im Endeffekt um die Qualität der Enddaten geht, welche der Crawler ausgibt, macht es Sinn, diese Beurteilung und Sicherstellung der Datenqualität nicht nur einerseits auf meinen Crawler zu beziehen, sondern sie auch auf die Instanz der DFG bzw. das Gepris-Systems zu erweitern.

Hinsichtlicher des Crawlingvorgangs können potentielle Implementierungsfehler die Datenqualität beeinträchtigen, beispielsweise das Nichtbeachten von einzelnen Entitäten wie einer Person, einer Institution oder eines Projektes, sowie andere Verletzungen der Datenkonsistenz, wie beispielsweise das doppelte Aufführen der gleichen Entität.

Der Crawler kann in einigen zentralen Aspekten die Datenqualität aus Nutzersicht zwar auch wesentlich verbessern, eben dadurch, dass er die Daten strukturiert und gesammelt erfasst und abspeichert anstatt nur eine Website-GUI mit rudimentären Abfragemöglichkeiten und dann weitestgehend schemalosen HTML-Präsentationen der Ergebnisse bereitstellt (erst dadurch werden erwähnt komplexere Datenanalyseszenarien wie statische Auswertungen und Vergleiche machbar bzw. für den Nutzer zumutbar). Dies betrifft also vor allem die Organisation und Bereitstellung der Daten. Es ist auch möglich, dass der Crawler in einigen Fällen die Qualität der Daten an sich erhöht. Beispielsweise gibt es auf den Detailseiten zu Personen im Gepris-System keine direkte Verlinkung zu der Institution, bei welcher diese Person aktuell arbeitet, es existiert lediglich die Postanschrift der Arbeitsstelle der Person, welche in fast allen Fällen den Institutionsnamen enthält. Dieser könnte von dem Crawler verwendet werden, um eine explizite Verbindung zu der entsprechenden Institution vorzunehmen. Auch werden für die Adressangaben von Personen auf deren Detailseiten nur die jeweils aktuellen Arbeitsadresse verwendet werden. Es kann jedoch natürlich vorkommen, dass eine Person den Arbeitsplatz wechselt. Und noch offensichtlicher ist das Problem, wenn eine Person verstirbt: in diesem Fall werden keinerlei Adressdaten mehr von der DFG zur Verfügung gestellt und es gibt auch keine historische Suche auf der Gepris-Seite. Damit ergibt sich ein weiterer Aspekt, welcher Bezug zur Datenqualität, genauer, zur Eigenschaft der Vollständigkeit hat. Dadurch dass der Crawler beliebig oft

ausgeführt werden kann, ist es möglich über die Zeit eine Historie solche Änderungen zu speichern und auch nach einem möglichen Arbeitsplatzwechsel oder dem Tod einer Person zu ermitteln, für welche Institution diese zum Zeitpunkt einer Projektbeteiligung gearbeitet hat.

In vielen anderen Aspekten, vor allem jene, welche die Qualität der Daten an sich betrifft, kann der Crawler aber die Qualitätsgüte höchstens halten, im ungünstigen Fall sogar verringern, beispielsweise was die korrekte Zuordnung von Eigenschaftswerten zu Entitäten angeht oder auch die vollständige Erfassung aller von der DFG veröffentlichten Eigenschaften.

Hinsichtlich der Qualität der eigentlichen DFG-Daten, so wie sie auf der Gepris-Website tatsächlich veröffentlicht werden, kann zwar zurecht argumentiert werden, dass die Verantwortung für die Datenqualität hier auf Seiten der DFG liegt und unsere Untersuchung diese ignorieren kann. Für den Endnutzer ist jedoch das Endergebnis interessant, somit auch die Qualität der Daten insgesamt wie er sie konsumieren kann und damit auch die Qualität der Ursprungsdaten wie sie über das Gepris-System zur Verfügung gestellt werden. Für mich als Entwickler des Crawlers besteht das primäre Interesse, da ich hier Handlungsspielraum hinsichtlicher der zu garantieren Qualität habe und diese verbessern kann, sollten Mängel aufgedeckt werden, in den Datenqualitätsaspekten meiner Crawler-Komponente. Jedoch besteht auch der sekundäre Wunsch, sofern zeitlich machbar, rudimentäre Qualitätschecks auch hinsichtlich der Ursprungsdaten der Gepris bereitzustellen und die DFG dann bei Bedarf auf potentiell identifizierte Fehler aufmerksam zu machen.

3.3 'Leichte Inbetriebnahme' als weitere Anforderung an den Crawler

Neben der Datenqualität ist auch die leichte Inbetriebnahme des Crawlers eine entscheidende Anforderung. Idealerweise reicht ein minimales, leicht verständliches, wenig fehleranfälliges und einmaliges Setup aus um die Anwendung einzurichten. Auch der eigentliche Betrieb sollte schnell und mittels eines einzelnen Befehls möglich sein. Sollten während des Crawlings Fehler auftreten und der Prozess vorzeitig terminieren, sollte eine Wiederaufnahme des abgebrochenen Crawls möglich sein, wobei bereits gemachte Fortschritte möglichst beibehalten werden sollten.

Diese Anforderung tangiert im Übrigen auch die Datenqualität: Eine aktuelle Datenbasis und langfristig auch eine Historie, um Änderungen innerhalb des Gepris-Datenbestandes nachzuvollziehen zu können, wird durch eine möglichst nutzerfreundliche und einfache Inbetriebnahme der Software begünstigt.

4 Die ausgewählten Datenqualitäts-Kriterien

Der für mich intuitiv naheliegendste Schwerpunkt hinsichtlich der Sicherstellung der notwendigen Datenqualität liegt auf den bereits erwähnten Eigenschaften der Vollständigkeit und logischen Konsistenz, auch da sich die Verletzung dieser Anforderungen bei der früheren Implementierung des Crawlers als problematisch herausgestellt haben. Dennoch soll in diesem Zuge eine Durchsicht der Datenqualitäts-Kriterien nach [FBMR17] vorgenommen werden und für jedes Kriterium dabei entschieden und argumentiert werden, ob und warum ich sie für die Implementierung grundsätzlich in Betracht ziehen will. Entscheidungskriterien sind dabei vor allem ob das jeweilige Kriterium

- relevant für den Nutzer ist
- sich messen lässt
- sich automatisch oder zumindest semi-automatisch testen lässt
- es im Rahmen der Arbeit realistisch umgesetzt werden kann.

Dabei wurden sämtliche Kriterien, welche sich spezifisch auf die Domäne 'Linked Open Data' und 'Knowledge Graphen' richtet, nicht aufgenommen. Ebenso wurden Kriterien, welche sich mit zeitlichen Aspekten wie der Aktualisierungsfrequenz oder der zeitlichen Einschränkung von Informationen beschäftigen, außen vor gelassen, da dies den Rahmen der Arbeit sprengen würde.

Es ist wichtig festzustellen, dass aufgrund der strukturellen Unterschiede zwischen der Organisationsform von Knowledge Graphen, wie sie im Kontext Linked Data verwendet werden und welche das Färber-Paper adressiert, und der Domäne Gepris in den meisten Fällen nicht zu einer exakten Übernahme des jeweiligen Kriteriums kommen kann, sondern eher zu einer daraus abgeleiteten, in manchen Fällen sogar nur dadurch inspirierten Variante.

Des Weiteren ist es wichtig zu erwähnen, dass die Übernahme eines Kriteriums nicht zwangsläufig mit einer vollständigen oder auch nur teilweisen Implementierung nötiger Massnahmen zur Sicherstellung eben jenes Kriteriums einhergeht. Im Rahmen der vorhandenen Zeit zur Anfertigung der Arbeit müssen eventuell Prioritäten gesetzt werden und im Fokus steht für mich ganz klar die Auseinandersetzung und Dokumentation der Datenqualitätskriterien für die Domäne Gepris per se, im zweiten Schritt dann, sofern genug zeitliche Ressourcen bleiben, eine möglichst umfangreiche Sicherstellung jener Kriterien.

Auch möchte ich darauf hinweisen, dass die Gewichtung der Relevanz für den Nutzer auf meiner subjektiven Einschätzung beruht.

Es folgt die nach Dimensionen, so wie sie auf Wang und Färber zurückgehen (auf eine weitere, noch gröbere Einordnung in Kategorien, so wie von

Färber und Wang vorgenommen, wurde verzichtet), geordnete Auflistung der ausgewählten Kriterien.

4.1 Kriterien aus der Dimension 'Genauigkeit'

4.1.1 Syntaktische Validität der CSV-Ausgabedateien

Dieses Kriterium basiert auf dem Kriterium *Syntactic Validity of RDF documents* nach Färber[FBMR17]. Die Anforderungen an ein valides CSV-Dokument sind im Vergleich zu den Anforderungen an RDF-Dokumente recht überschaubar. Im Wesentlichen muss sichergestellt werden, dass jede Datei eine Kopfzeile mit den Spaltennamen hat und die nachfolgenden Zeilen jeweils durch Zeilenumbruch getrennt sind, die gleiche Anzahl an durch Komma getrennte Spalten hat wie die Kopfzeile und komplexe Spaltenwerte (vor allem jene, welche selbst Kommata enthalten) entsprechend durch Anführungszeichen geschützt sind *(String Escaping)*.

Messung des Kriteriums

- **Aufnahme in die Liste der zu messenden Kriterien:** Ja
- **Fokussierte Instanz (Crawler, DFG, beides):** Crawler
- **Metrik:** Anteil valider und invalider CSV-Ausgabedateien
- **automatische, semi-automatische oder manuelle Messung:** automatisch
- **Konzept zur Messung:** Wenn sich die CSV-Dateien mit einem gängigen Programm, wie Excel oder R-Studio, laden und korrekt anzeigen lassen und dabei keine Fehler auftreten ist dies eine erste notwendige Überprüfung der syntaktischen Validität der Dateien. Zusätzlich kann ein sogenannter linter zur Überprüfung eingesetzt werden, wie er zum Beispiel unter `https://csvlint.io` bereit steht.
- **Umsetzung der Messung realistisch im Rahmen der Arbeit:** Ja

Sicherstellung des Kriteriums

- **Aufnahme in die Liste der sicherzustellenden Kriterien:** Ja
- **Konzept zur Sicherstellung:** Verwendung einer dezidierten CSV-Bibliothek, welche das korrekte Schreiben und Lesen von CSV-Dateien ermöglicht.
- **Umsetzung der Sicherstellung realistisch im Rahmen der Arbeit:** Ja

Begründung für die Aufnahme

Das Kriterium ist messbar, die Relevanz für den Nutzer ist sehr hoch und die Umsetzung erscheint mir sowohl hinsichtlich der Messung als auch der Sicherstellung im Rahmen der Arbeit als realistisch.

4.1.2 Syntaktische Validität von Literalen für einzelne Spalten

Dieses Kriterium basiert auf dem Kriterium *Syntactic Validity of Literals* nach Färber[FBMR17]. Für gewisse Spalten/Felder kann eine einfache syntaktische Prüfung vorgenommen werden. Beispielsweise kann mittels regulärer Ausdrücke geprüft werden, ob Projekt-Ids (Zahlenwerte) oder Jahresangaben syntaktisch korrekt sind (vierstellige Zahlenwerte).

Messung des Kriteriums

- **Aufnahme in die Liste der zu messenden Kriterien:** Ja
- **Fokussierte Instanz (Crawler, DFG, beides):** beides, primär DFG
- **Metrik:** Anteil syntaktisch valider und invalider Literale
- **automatische, semi-automatische oder manuelle Messung:** automatisch
- **Konzept zur Messung:** Mittels definierter Regeln, zum Beispiel in Form von regulären Ausdrücken können mithilfe eines R-Skriptes sämtliche Literale bestimmter Spalten auf Validität hin überprüft werden.
- **Umsetzung der Messung realistisch im Rahmen der Arbeit:** Ja

Sicherstellung des Kriteriums

- **Aufnahme in die Liste der sicherzustellenden Kriterien:** Ja
- **Konzept zur Sicherstellung:** Neben einer möglichst sorgfältigen Implementierung der Extractor-Logik, so dass der von der DFG veröffentlichte Wert eines Feldes jeweils korrekt übernommen wird, sind keine weiteren Massnahmen ersichtlich.
- **Umsetzung der Sicherstellung realistisch im Rahmen der Arbeit:** Ja

Begründung für die Aufnahme

Das Kriterium ist quantifizierbar, die Relevanz für den Nutzer erscheint mir hoch und die Umsetzung erscheint mir sowohl hinsichtlich der Messung als auch der Sicherstellung im Rahmen der Arbeit als realistisch.

4.1.3 Semantische Validität der Entitäten

Dieses Kriterium basiert auf dem Kriterium *Semantic Validity of Triples* nach Färber[FBMR17]: "The semantic validity of triples is introduced to evaluate whether the meanings of triples with literal values in object position in the KG are semantically correct. A triple is either semantically correct if it is also available from a trusted source (e.g. Name Authority File) or if it is common sense or if the stated property can be measured or perceived by us directly". Semantische Validität meint hierbei also: sind die bereitgestellten oder direkt abzuleitenden Informationen konsistent mit den Beobachtungen bzw. Überprüfungen der beschriebenen Sachverhalte der realen Welt.

Allgemein lässt sich feststellen, dass, um eine absolute Gewissheit hinsichtlich der semantischen Validität zu erzeugen, für jede Entität vermutlich mittels manueller Empirie ein Abgleich mit der Realität vorgenommen werden müsste, was je nach Domäne unterschiedlich aufwendig, in vielen Fällen jedoch sicher enorm komplex sein würde. Auch das Färber-Paper hebt hervor: "Evaluating the semantic validity is hard, even if a random sample set is evaluated manually, e.g., via crowd-sourcing."

Im Falle unserer Domäne gilt zwar: weder drückt die Gepris-Seite direkt formelle Aussagen aus welche Ähnlichkeiten mit RDF-Tripeln hätten, noch ist es im Umfang meiner Arbeit vorgesehen, solche automatisch abzuleiten. Jedoch lassen sich auch semantische Bedingungen beliebig an ganze Entitäten oder einzelne Literale bzw. Spaltenwerte stellen.

Der Umfang und der Anspruch an die Bewertung und Überprüfung semantischer Validität ist sehr offen und kann je nach Fall recht komplex werden. Es würde sich im Übrigen vermutlich als schwer erweisen, eine vertrauenswürdige Quelle neben der DFG zu bestimmen, welche als Referenz zur Überprüfung herangezogen werden kann. Ein Beispiel für eine umfangreiche Überprüfung hinsichtlich der semantischen Validität wäre, die über ein Projekt zur Verfügung gestellten Aussagen in manueller Recherche zu überprüfen, indem beispielsweise die beteiligten Wissenschaftler kontaktiert und befragt werden. Ein gewisses Maß an semantischer Validität lässt sich jedoch auch automatisch überprüfen: beispielsweise lassen sich unter anderem die Anforderungen, dass das Geburtsdatum einer real existierenden Person in der Vergangenheit liegen muss und dass ihr Todesdatum, sofern existent, nach dem Geburtsdatum liegt, ebenso gut logisch und programmatisch ausdrücken und überprüfen wie bestimmte Anforderungen an Identifikationsnummern, welche eingebaute Prüfmechanismen bzw. Prüfsummen bereitstellen, also z.B. Kreditkartennummern oder ISBN-Nummern. Lediglich auf diesen Ansatz soll hier Bezug genommen werden.

Messung des Kriteriums

- **Aufnahme in die Liste der zu messenden Kriterien:** Ja
- **Fokussierte Instanz (Crawler, DFG, beides):** beides, primär DFG

- **Metrik:** Anteil regelverletzender und regelkonformer Überprüfungen. Beispiel: es werden für die Ressource "Projekte" zwei Regeln definiert, z.B. hinsichtlich sinnvoller Angaben zum Förderungsbeginn und Förderungsende sowie hinsichtlich der Angabe eines Eltern-Projektes (ein Projekt sollte nicht sein eigenes Elternprojekt sein). Pro Entität können nun also zwei Regeln geprüft werden, insgesamt ergibt sich also eine Überprüfung von n * 2 Fällen, wobei n die Anzahl der Projekte ist.
- **automatische, semi-automatische oder manuelle Messung:** automatisch
- **Konzept zur Messung:** Mittels eines R-Skriptes und dort definierter Regeln/logischer Einschränkungen können sämtliche Entitäten eines Crawling-Ergebnisses auf die semantische Validität hin überprüft werden.
- **Umsetzung der Messung realistisch im Rahmen der Arbeit:** Ja, bis zu einem gewissen Umfang mittels einiger Regeln.

Sicherstellung des Kriteriums

- **Aufnahme in die Liste der sicherzustellenden Kriterien:** Ja
- **Konzept zur Sicherstellung:** Neben einer möglichst sorgfältigen Implementierung der Extractor-Logik, so dass der von der DFG veröffentlichte Wert korrekt übernommen wird, sind keine weiteren Massnahmen ersichtlich.
- **Umsetzung der Sicherstellung realistisch im Rahmen der Arbeit:** Ja, bis zu einem gewissen Umfang mittels einiger Regeln, welche vom Crawler überprüft werden.

Begründung für die Aufnahme

Das Kriterium ist quantifizierbar (zumindest bezüglich der Menge an definierten Regeln), die Relevanz für den Nutzer erscheint mir im mittleren Bereich angesiedelt zu sein und die Umsetzung erscheint mir sowohl hinsichtlich der Messung als auch der Sicherstellung im Rahmen der Arbeit anhand einer Beispielregel als grundsätzlich realistisch.

4.2 Kriterien aus der Dimension 'Vertrauenswürdigkeit'

4.2.1 Vertrauenswürdigkeit auf Entitätsebene mittels Quellennachweis

Dieses Kriterium basiert auf dem Kriterium *Trustworthiness on statement level* nach Färber[FBMR17]: "The fulfillment of trustworthiness on statement level is determined by an evaluation whether a provenance vocabulary is used. By means of a provenance vocabulary, the source of statements can be stored."

Die zum Zeitpunkt des Crawlings genutzten HTML-Rohdaten werden mit abgespeichert und darüber hinaus können die (jedoch potentiell zwischenzeitlich geänderten) Originalseiten zu den Entitäten über die ebenfalls gespeicherte Id über das Gepris-System erneut aufgerufen werden.

Messung des Kriteriums

- **Aufnahme in die Liste der zu messenden Kriterien:** Ja
- **Fokussierte Instanz (Crawler, DFG, beides):** Crawler
- **Metrik:** Anteil der Entitäten mit und ohne Quellenverweis
- **automatische, semi-automatische oder manuelle Messung:** automatisch
- **Konzept zur Messung:** Mithilfe eines R-Skriptes kann einerseits geprüft werden, ob es Entitäten ohne Id gibt, also ob es Entitäten gibt, bei denen nicht ohne weiteres die Originaldatei auf dem Gepris-System aufgerufen werden kann, und andererseits (für eine bestimmte Menge an Stichproben oder auch für alle gespeicherten Entitäten), ob für die jeweiligen Ids die entsprechende HTML-Datei mit abgespeichert wurden.
- **Umsetzung der Messung realistisch im Rahmen der Arbeit:** Ja

Sicherstellung des Kriteriums

- **Aufnahme in die Liste der sicherzustellenden Kriterien:** Ja
- **Konzept zur Sicherstellung:** Der Crawler speichert für jede Entität die jeweilige ID und die gecrawlte HTML-Datei mit ab.
- **Umsetzung der Sicherstellung realistisch im Rahmen der Arbeit:** Ja

Begründung für die Aufnahme

Das Kriterium ist quantifizierbar, die Relevanz für den Nutzer erscheint mir hoch, denn es erhöht die Vertrauenswürdigkeit zu den bereitgestellten Daten zu einer Entität, wenn die Originaldatenquelle (in diesem Fall die für das Crawling verwendete HTML-Seite des Gepris-Systems) mit referenziert wird bzw. sogar direkt archiviert bereitgestellt wird. Auch die Umsetzung erscheint mir sowohl hinsichtlich der Messung als auch der Sicherstellung im Rahmen der Arbeit als realistisch.

4.2.2 Differenzierung zwischen leeren und unbekannten Werten

Dieses Kriterium basiert auf dem Kriterium *Using unknown and empty values* nach Färber[FBMR17]: "If the data model of the considered KG supports the representation of unknown and empty values, more complex statements can be

represented. For instance, empty values enable to represent that a person has no children and unknown values enable to represent that the birth date of a person in not known. This kind of higher explanatory power of a KG increases the trustworthiness of the KG."

Messung des Kriteriums

- **Aufnahme in die Liste der zu messenden Kriterien:** Ja
- **Fokussierte Instanz (Crawler, DFG, beides):** beides, primär Crawler
- **Metrik:** Binär - entweder der Crawler (bzw. die DFG, sofern diese als Instanz im Fokus der Messung steht) differenziert zwischen leeren und unbekannten Werten (Metrikwert 1) oder er tut es nicht (Metrikwert 0)
- **automatische, semi-automatische oder manuelle Messung:** manuell
- **Konzept zur Messung:** Der Nutzer der Daten kann sich z.B. mittels eines R-Skriptes überzeugen, ob zwischen leeren (beispielsweise ein leerer String) und unbekannten Werten (im Falle von R z.B. der Wert NA) differenziert wird. [1]
- **Umsetzung der Messung realistisch im Rahmen der Arbeit:** Ja (Metrik ist vom Nutzer manuell zu messen, daher kein Implementierungsaufwand)

Sicherstellung des Kriteriums

- **Aufnahme in die Liste der sicherzustellenden Kriterien:** Ja
- **Konzept zur Sicherstellung:** Die Extractor-Logik unterscheidet zwischen nicht gefundenen Feldern (unbekannt) und gefundenen Feldern ohne Wert (leerer Wert).
- **Umsetzung der Sicherstellung realistisch im Rahmen der Arbeit:** Eventuell

Begründung für die Aufnahme

Das Kriterium ist quantifizierbar, wenn auch nur binär, die Relevanz für den Nutzer erscheint mir im mittleren Bereich zu liegen und die Umsetzung erscheint mir als realistisch.

[1] Sollten keine Beispiele für die Differenzierungen in den Ausgabedateien des Crawlers gefunden werden, ist dies streng genommen noch kein ausreichendes Indiz dafür, dass der Crawler dieses Konzept tatsächlich hinreichend und akkurat unterstützt. In diesem Fall müsste entweder auf die Angaben in der Dokumentation des Crawlers vertraut werden oder eine eigenständige Überprüfung (z.B. mittels eines automatischen Test-Setups) der Crawler-Logik erfolgen.

4.3 Kriterien aus der Dimension 'Konsistenz'

4.3.1 Konsistenz bezüglich definierter Beziehungseinschränkungen

Dieses Kriterium basiert auf dem Kriterium *Consistency of statements w.r.t. relation constraints* nach Färber[FBMR17]: "This metric is intended to measure the degree to which the instance data are consistent with the relation restrictions specified on the schema level (e.g., rdfs:range, and owl:FunctionalProperty).".

Zwar ist das ursprüngliche Kriterium nach Färber sehr RDF-spezifisch und eine direkte Übernahme, würde ein Äquivalent zum Konzept der *functionalProperties* in unserer Domäne voraussetzen. Allerdings inspiriert dieses Kriterium zu einer Überprüfung vorher zu definierender Konsistenzregeln hinsichtlich der Beziehungen zwischen Entitäten. Beispielsweise ließe sich eine Regel ausdrücken, welche erwartet, dass die Dienstanschriften aller im System gespeicherter Personen sich auf eine ebenfalls im System hinterlegte Institution beziehen lassen. Ein noch relevanteres Beispiel wäre die Anforderung, dass für jede Personen-Id, die in einer Beziehungs-Tabelle vorliegt (zum Beispiel jene Tabelle, welche Beziehungen zwischen Projekten und Personen abbildet) auch ein entsprechender Eintrag in der Personen-Tabelle vorliegt.

Messung des Kriteriums

- **Aufnahme in die Liste der zu messenden Kriterien:** Ja
- **Fokussierte Instanz (Crawler, DFG, beides):** beides
- **Metrik:** Verhältnis der Anzahl an Beziehungen welche Beziehungseinschränkungen verletzen und der Anzahl an Beziehungen welche diese einhalten
- **automatische, semi-automatische oder manuelle Messung:** automatisch oder semi-automatisch
- **Konzept zur Messung:** Mittels eines R-Skripte können die Regeln in beliebigem Umfang definiert und diese auf ein Crawling-Ergebnis angewandt und die Anzahl der Regelverstöße dokumentiert werden.
- **Umsetzung der Messung realistisch im Rahmen der Arbeit:** Ja

Sicherstellung des Kriteriums

- **Aufnahme in die Liste der sicherzustellenden Kriterien:** Ja
- **Konzept zur Sicherstellung:** Neben einer möglichst sauberen Umsetzung vor allem der Extraktor-Logik sind keine weiteren Ansätze zur Sicherstellung ersichtlich. Unter der Annahme, dass die definierten Regeln innerhalb der Domäne Gepris Sinn machen, liegt darüber hinaus dann die Verantwortung zur Sicherstellung der Einhaltung bei der DFG.
- **Umsetzung der Sicherstellung realistisch im Rahmen der Arbeit:** Ja

Begründung für die Aufnahme

Das Kriterium ist quantifizierbar, die Relevanz für den Nutzer erscheint mir hoch und die Umsetzung erscheint mir sowohl hinsichtlich der Messung als auch der Sicherstellung im Rahmen der Arbeit als prinzipiell realistisch.

4.4 Kriterien aus der Dimension 'Vollständigkeit'

4.4.1 Vollständige Schemaabdeckung

Dieses Kriterium basiert auf dem Kriterium *Schema Completeness* nach [FBMR17]: "By means of the criterion Schema completeness, one can determine the completeness of the schema w.r.t. classes and relations. The schema is assessed by means of a gold stan- dard. This gold standard consists of classes and relations which are relevant for the use case. For evaluating cross-domain KGs, we use as gold standard a typical set of cross-domain classes and relations." Wie das Färber-Paper überzeugend feststellt, ist die Bewertung der Schema-Vollständigkeit abhängig von dem jeweiligen Nutzungsszenario und seinen Anforderungen. Hier macht besonders die Betonung der Existenz der zwei involvierten Instanzen Sinn: der DFG/Gepris als Ursprungsdatenquelle und dem Crawler als Erfassungs- und Transformationsprozess. Ob der von der DFG bereitgestellte Umfang des Datenschemas für ein Nutzungsszenario ausreicht, hängt jeweils von eben diesem ab und kann daher nicht allgemein bestimmt werden. Für viele Forschungsfragen werden die von der DFG bereitgestellten Felder ausreichen, für viele andere wären aber ein weitreichenderes Datenschema wünschenswert, beispielsweise Felder welche das monetäre Förderungsvolumen angeben. Für die Crawling-Komponente lässt sich jedoch allgemein leicht folgendes zu erreichende Ideal hinsichtlich der Schema-Vollständigkeit bestimmen: alle Felder und Informationen welche das Gepris-System grundsätzlich bereitstellt werden auch von dem Crawler erfasst und strukturiert abgespeichert.

Messung des Kriteriums

- **Aufnahme in die Liste der zu messenden Kriterien:** Ja
- **Fokussierte Instanz (Crawler, DFG, beides):** Crawler
- **Metrik:** Anzahl der vom Crawler erfassten Schema-Bestandteile (Einheit: Felder/Spalten) im Verhältnis zu allen von der DFG bereitgestellten Feldern.
- **automatische, semi-automatische oder manuelle Messung:** automatisch oder semi-automatisch
- **Konzept zur Messung:** Beim Untersuchen der DOM-Struktur der Gepris-Webseiten hat sich herausgestellt, dass die CSS-Klassen und CSS-IDs der Feldbezeichner stets einem ähnlichen Muster folgen. Anhand die-

ses Musters kann die Selektion aller überhaupt vorhandener Felder auf allen HTML-Seiten erfolgen und eine duplikatfreie Liste der Feldnamen erzeugt werden. Diese Bestimmung kann innerhalb des Crawler-Prozesses selbst erfolgen oder anschließend mittels eines R-Skriptes. Ebenfalls in R kann dann überprüft werden, wie viele der ermittelten Felder eine Entsprechung in den gespeicherten CSV-Ausgabedateien in Form von Spaltennamen aufweisen. Sollten weitere relevante Informationen in HTML-Sektionen enthalten sein, welche sich nicht über dieses CSS-Selektionsschema automatisch identifizieren lassen, müssten diese manuell identifiziert und entsprechend individuelle CSS-Selektoren zur Extractor-Logik des Crawlers hinzugefügt werden.

- **Umsetzung der Messung realistisch im Rahmen der Arbeit:** Eventuell

Sicherstellung des Kriteriums

- **Aufnahme in die Liste der sicherzustellenden Kriterien:** Ja
- **Konzept zur Sicherstellung:** Bei der Entwicklung der Extractor-Logik sollte zur Bestimmung der Referenzliste aller zu beachtender Felder der in "Konzept zur Messung" erwähnte CSS-Selektor-Ansatz gewählt werden. Dann sollten all jene Felder auch von der Extractor-Logik beachtet werden oder zumindest transparent gemacht werden, welche Felder nicht extrahiert werden.
- **Umsetzung der Sicherstellung realistisch im Rahmen der Arbeit:** Eventuell

Begründung für die Aufnahme

Das Kriterium ist quantifizierbar, die Relevanz für den Nutzer erscheint mir hoch und die Umsetzung erscheint mir sowohl hinsichtlich der Messung als auch der Sicherstellung im Rahmen der Arbeit als prinzipiell realistisch.

4.4.2 Vollständige Spaltenbelegung

Dieses Kriterium basiert auf dem Kriterium *Column Completeness* nach [FBMR17]: "By means of the column completeness criterion, one can determine the degree by which the attributes of a class, which are defined on the schema level, exist on the instance level of the KG." Wir haben keinen Einfluss darauf ob die DFG die für einen Entitätstyp vorgesehenen Felder für jede Entitätsinstanz belegt. Jedoch ergibt sich hinsichtlich des Crawlers eine ähnliche Argumentation wie bei dem Kriterium "Schema Completeness" / "Vollständige Schemaabdeckung": Das zu erreichende Ideal ist es, dass alle Spaltenbelegungen (sprich: alle vorhandenen Feldwerte), welche von der DFG bereitgestellt werden, auch von dem Crawler erfasst, verarbeitet und bereitgestellt werden.

Messung des Kriteriums

- **Aufnahme in die Liste der zu messenden Kriterien:** Ja
- **Fokussierte Instanz (Crawler, DFG, beides):** beides, primär Crawler
- **Metrik:** Verhältnis der vom Gepris-System übernommenen Spaltenbelegungen und der fehlenden Spaltenbelegungen
- **automatische, semi-automatische oder manuelle Messung:** manuell oder semi-automatisch
- **Konzept zur Messung:** Manuelle oder semi-manuelle (z.B. durch R-Skripte) Stichproben: Eine hinsichtlich der Mächtigkeit zu bestimmende Testmenge an Entitäten, für welche dann jeweils mittels Abgleich zwischen der zugehörigen Gepris-Webseite (durch Aufruf entweder der aktuellen Version oder der vom Crawler gespeicherten Version) und der Repräsentation der Entität im Crawling-Ergebnis (zugehörige Zeilen in den entsprechenden CSV-Dokumenten) der Metrikwert bestimmt und entsprechend aggregiert wird.
- **Umsetzung der Messung realistisch im Rahmen der Arbeit:** Ja

Sicherstellung des Kriteriums

- **Aufnahme in die Liste der sicherzustellenden Kriterien:** Ja
- **Konzept zur Sicherstellung:** Neben einer möglichst sorgfältigen Implementierung der Extractor-Logik, so dass der von der DFG veröffentlichte Wert korrekt übernommen wird, sind keine weiteren Massnahmen ersichtlich.
- **Umsetzung der Sicherstellung realistisch im Rahmen der Arbeit:** Ja

Begründung für die Aufnahme

Das Kriterium ist quantifizierbar, die Relevanz für den Nutzer erscheint mir hoch und die Umsetzung erscheint mir sowohl hinsichtlich der Messung als auch der Sicherstellung im Rahmen der Arbeit als realistisch.

4.4.3 Vollständige Populationsabdeckung

Dieses Kriterium basiert auf dem Kriterium *Population Completeness* nach [FBMR17]: "The population completeness metric determines the extent to which the considered KG covers the basic population. The assessment of the completeness of the basic population is performed by a gold standard, which covers both well-known resources (called "short head", e.g., the n largest cities in the world according to the number of inhabitants) and little-known resources

(called "long tail"; e.g., municipalities in Germany)." Wir haben keinen Einfluss darauf ob die DFG alle Entitätsinstanzen, die sie grundsätzlich im Rahmen des Gepris-Systems publizieren könnte, auch tatsächlich bereitstellt. Jedoch ergibt sich hinsichtlich des Crawlers eine ähnliche Argumentation wie bei den Kriterien ""Vollständige Schemaabdeckung" und "Vollständige Spaltenbelegung": das zu erreichende Ideal (in der Terminologie des Färber-Papers: der Goldstandard) ist es, dass alle Entitäten, welche von der DFG/Gepris bereitgestellt werden, auch von dem Crawler erfasst, verarbeitet und bereitgestellt werden.

Messung des Kriteriums

- **Aufnahme in die Liste der zu messenden Kriterien:** Ja
- **Fokussierte Instanz (Crawler, DFG, beides):** Crawler
- **Metrik:** Anteil der Entitäten mit und ohne Quellenverweis
- **automatische, semi-automatische oder manuelle Messung:** automatisch
- **Konzept zur Messung:** Die DFG gibt an mehreren Stellen die Anzahl an im Gepris-System bereitgestellten Instanzen pro Entitätstyp an, so z.B. im Rahmen des sogenannten Datenmonitors[2] oder innerhalb der Navigationszeile in den Indexseiten im Rahmen der Katalogseite bzw. der Suchfunktion. Wir vertrauen hier primär auf die Angabe der Katalogseite bzw. der Suchfunktion und verwenden diese Zahl als Refernzwert, da sie, sofern keine Fehler im Gepris-System vorliegen, stets den aktuell korrekten Wert bezüglich der Anzahl der momentan im System verfügbaren Ressourcen angibt. Diesen Wert kann der Crawler mit speichern und damit zur anschliessenden Validierung mittels eines einfachen Vergleichs der Anzahl an ermittelten Zeilen in der Tabelle der jeweiligen Ressource zur Verfügung stellen, beispielsweise mithilfe eines R-Skriptes.
- **Umsetzung der Messung realistisch im Rahmen der Arbeit:** Ja

Sicherstellung des Kriteriums

- **Aufnahme in die Liste der sicherzustellenden Kriterien:** Ja
- **Konzept zur Sicherstellung:** Neben einer möglichst sorgfältigen Implementierung des Crawlers, insbesondere innerhalb der ersten Schritte zur Bestimmung aller zu crawlenden Entitäten, so dass dem Crawler keine von der DFG veröffentlichten Instanzen entgehen, sind keine weiteren Massnahmen ersichtlich.
- **Umsetzung der Sicherstellung realistisch im Rahmen der Arbeit:** Ja

[2] `http://gepris.dfg.de/gepris/OCTOPUS?task=showMonitor`

Begründung für die Aufnahme

Das Kriterium ist quantifizierbar, die Relevanz für den Nutzer erscheint mir hoch und die Umsetzung erscheint mir sowohl hinsichtlich der Messung als auch der Sicherstellung im Rahmen der Arbeit als realistisch.

4.5 Kriterien aus der Dimension 'Verständlichkeit'

4.5.1 Unterstützung von Mehrsprachigkeit

Dieses Kriterium basiert auf dem Kriterium *Labels in multiple languages* nach [FBMR17]: "Labels are usually provided in English as the "basic language". The now introduced metric for the criterion labels in multiple languages determines whether labels in other languages than English are provided in the KG." Die DFG stellt die Förderungsdaten grundsätzlich in Englisch und in Deutsch bereit. Der Crawler könnte somit auch beide Sprachversionen verarbeiten und bereitstellen.

Messung des Kriteriums

- **Aufnahme in die Liste der zu messenden Kriterien:** Ja
- **Fokussierte Instanz (Crawler, DFG, beides):** beide, primär der Crawler
- **Metrik:** Anzahl der unterstützten und nicht unterstützten Sprachen, in Relation zur Gesamtmenge der von der DFG bereitgestellten Sprachen.
- **automatische, semi-automatische oder manuelle Messung:** manuell
- **Konzept zur Messung:** Der Nutzer kann sich leicht selbst über die Anzahl der bereitgestellten Sprachen eines Crawlingvorgangs durch einen schnellen Blick in den Ausgabeordner informieren.
- **Umsetzung der Messung realistisch im Rahmen der Arbeit:** Ja

Sicherstellung des Kriteriums

- **Aufnahme in die Liste der sicherzustellenden Kriterien:** Ja
- **Konzept zur Sicherstellung:** Die Crawling- und insbesondere die Extractor-Logik muss die jeweiligen Eigenarten jeder zu unterstützenden Sprache beachten.
- **Umsetzung der Sicherstellung realistisch im Rahmen der Arbeit:** Ja

Begründung für die Aufnahme

Das Kriterium ist quantifizierbar, die Relevanz für den Nutzer erscheint mir im mittleren Bereich zu liegen - eine englische Version der bereitgestellten Daten sollte in den meisten Fällen ausreichen, in manchen Szenarien, z.B. für Vergleiche, ist die Bereitstellung beispielsweise der deutschen Version jedoch hilfreich. Die Umsetzung erscheint mir sowohl hinsichtlich der Messung als auch der Sicherstellung im Rahmen der Arbeit als prinzipiell realistisch.

4.5.2 Verständlichkeit und Dokumentation der Ausgabedateien

Dieses Kriterium basiert auf dem Kriterium *Understandable RDF serialization* nach [FBMR17]: "RDF/XML is the recommended RDF serialization format of the W3C. However, due to its syntax RDF/XML documents are hard to read for humans. The understandability of RDF data by humans can be increased by providing RDF in other, more human-understandable serialization formats such as N3, N-Triple, and Turtle." Idealerweise sind die CSV-Dateien im Ausgabeordner eines Crawling-Vorganges sowie deren Spalten ausreichend gut benannt. Zusätzlich macht eine Einführung in die Domäne und der zentralen Entitätstypen und der Beziehungen der Entitäten zueinander im Rahmen einer README-Datei Sinn.

Messung des Kriteriums

- **Aufnahme in die Liste der zu messenden Kriterien:** Ja
- **Fokussierte Instanz (Crawler, DFG, beides):** Crawler
- **Metrik:** Anteil der Felder/Spalten mit und ohne ausreichende Beschreibung.
- **automatische, semi-automatische oder manuelle Messung:** manuell
- **Konzept zur Messung:** Der Nutzer müsste selbstständig feststellen, ob alle Entitäten und Felder/Spalten verständlich benannt und ausreichend dokumentiert sind. Der Aufwand für eine automatische Messung erscheint hier nicht gerechtfertigt und auch in einigen Aspekten schwierig bis unmöglich, da menschliche Intelligenz für die Bewertung nötig ist.
- **Umsetzung der Messung realistisch im Rahmen der Arbeit:** Ja

Sicherstellung des Kriteriums

- **Aufnahme in die Liste der sicherzustellenden Kriterien:** Ja
- **Konzept zur Sicherstellung:** Die README-Datei muss um entsprechende Erklärungen der Tabellen, Spalten und Beziehungen erweitert werden.

- **Umsetzung der Sicherstellung realistisch im Rahmen der Arbeit:** Ja

Begründung für die Aufnahme

Das Kriterium ist quantifizierbar, die Relevanz für den Nutzer erscheint mir hoch zu sein - die Sicherstellung dieses Kriteriums erhöht die Verständlichkeit und damit die Nutzbarkeit der Datenbasis für den Nutzer. Die Umsetzung erscheint mir sowohl hinsichtlich der Messung als auch der Sicherstellung im Rahmen der Arbeit als realistisch.

4.6 Kriterien aus der Dimension 'Interoperabilität'

4.6.1 Bereitstellung in mehreren Datenformaten

Dieses Kriterium basiert auf dem Kriterium *Provisioning of several serialization formats* nach [FBMR17]: "The interpretability of RDF data of a KG is increased if besides the serialization standard RDF/XML further serialization formats are supported for URI dereferencing." Primär ist eine Bereitstellung der gecrawlten und extrahierten Daten im CSV-Format vorgesehen. Jedoch könnten weitere Ausgabeformate wie beispielsweise in Form von sqlite-Datenbankdateien, RDF- oder JSON-Dateien in Betracht gezogen werden.

Messung des Kriteriums

- **Aufnahme in die Liste der zu messenden Kriterien:** Ja
- **Fokussierte Instanz (Crawler, DFG, beides):** Crawler
- **Metrik:** 0 falls nur CSV als Ausgabeformat unterstützt wird, 1 falls weitere Ausgabeformate unterstützt werden
- **automatische, semi-automatische oder manuelle Messung:** manuell
- **Konzept zur Messung:** Der Nutzer der Daten kann sich sehr schnell und einfach durch einen Blick in den Ausgabeordner des Crawlers überzeugen, ob neben den CSV-Dateien noch weitere Ausgabedateien erzeugt worden sind.
- **Umsetzung der Messung realistisch im Rahmen der Arbeit:** Ja

Sicherstellung des Kriteriums

- **Aufnahme in die Liste der sicherzustellenden Kriterien:** Ja
- **Konzept zur Sicherstellung:** Der Crawler transformiert die CSV-Dateien zusätzlich in ein anderes Ausgabeformat.
- **Umsetzung der Sicherstellung realistisch im Rahmen der Arbeit:** Ja

Begründung für die Aufnahme

Das Kriterium ist zumindest binär quantifizierbar (entweder es gibt ein zusätzliches Ausgabeformat oder nicht), die Relevanz für den Nutzer erscheint mir im mittleren Bereich zu liegen - einige Nutzer bevorzugen z.B. eine sqlite-Datenbankdatei anstelle einer Sammlung von CSV-Dateien. Tatsächlich gab es eine entsprechende Anfrage eines der Mitglieder vom betreuenden Lehrstuhl. Die Umsetzung erscheint mir sowohl hinsichtlich der Messung als auch der Sicherstellung im Rahmen der Arbeit als prinzipiell realistisch.

4.7 Kriterien aus der Dimension 'Interlinking'

4.7.1 Validität der ursprünglichen Gepris-Seiten-URLs

Dieses Kriterium basiert auf dem Kriterium *Validity of external URIs* nach [FBMR17]: "The considered KG may contain outgoing links referring to RDF resources or Web documents (non-RDF data). [...] Linking to external resources always entails the problem that those links get invalid over time. This can have different causes, such as that the URI is not available anymore. We measure the validity of external URIs by evaluating the URIs from an URI sample set w.r.t. whether there is a timeout, a client error (HTTP response 4xx) or a server error (HTTP response 5xx)." Es können die (jedoch potentiell zwischenzeitlich geänderten) Originalseiten zu den Entitäten über die vom Crawler ebenfalls gespeicherte Id über das Gepris-System erneut aufgerufen werden. Diese Seiten können mit der Zeit nicht mehr erreichbar sein.

Messung des Kriteriums

- **Aufnahme in die Liste der zu messenden Kriterien:** Ja
- **Fokussierte Instanz (Crawler, DFG, beides):** Beides
- **Metrik:** Anteil der validen (erreichbaren) und invaliden Gepris-Seiten-URls
- **automatische, semi-automatische oder manuelle Messung:** manuell
- **Konzept zur Messung:** Mittels manueller oder semi-manueller Stichproben beliebiger Grösse können die Gepris-Seiten-URLs erneut aufgerufen werden und deren HTTP-Status-Codes überprüft werden.
- **Umsetzung der Messung realistisch im Rahmen der Arbeit:** Ja

Sicherstellung des Kriteriums

- **Aufnahme in die Liste der sicherzustellenden Kriterien:** Ja
- **Konzept zur Sicherstellung:** Der Crawler speichert für jede Entität die jeweilige ID und die gecrawlte HTML-Datei mit ab.

- **Umsetzung der Sicherstellung realistisch im Rahmen der Arbeit:** Ja

Begründung für die Aufnahme

Das Kriterium ist quantifizierbar, die Relevanz für den Nutzer erscheint mir im mittleren Bereich zu liegen - es erhöht die Vertrauenswürdigkeit zu den bereitgestellten Daten zu einer Entität, wenn die Originaldatenquelle (in diesem Fall die fürs Crawling verwendete HTML-Seite des Gepris-Systems) mit referenziert wird. Die Umsetzung erscheint mir sowohl hinsichtlich der Messung als auch der Sicherstellung im Rahmen der Arbeit als prinzipiell realistisch.

5 Der Crawler

Ich möchte nun einen Einblick in die Entwicklung des eigentlichen Kernstücks meiner Arbeit geben: die Crawling- und Extracting-Komponente. Spätestens nachdem in Absprache mit dem Lehrstuhl entschieden wurde, dass der Fokus der Entwicklung auf der Datenqualität und der leichten Inbetriebnahme der Anwendung liegen solle, habe ich von der Idee einer Webanwendung Abstand genommen und mich stattdessen für eine einfache Konsolenanwendung entschieden. Diese Anwendung soll, wenn einmal gestartet, im Hintergrund den zeitlich recht aufwendigen (je nach Qualität der Internetverbindung habe ich Laufzeiten zwischen 2 und 8 Stunden erlebt) Crawling- und Extracting-Vorgang ohne weitere Benutzerinteraktion autonom durchlaufen. Sie soll im wesentlichen zwei Befehle entgegennehmen: einen für das Starten eines kompletten neuen Crawling-Durchlaufs und einen für die Wiederaufnahme eines bereits begonnenen, aber nicht beendeten Durchlaufs. Daneben soll noch das obligatorische Hilfe-Kommando ('–help') bereitstehen, welches über die Benutzung der Anwendung Auskunft gibt.

5.1 Technologieauswahl und Implementierungskonzepte

Bezüglich der Technologieauswahl habe ich mich für die Programmiersprache Scala entschieden. Als Alternative hatte ich kurz die dynamisch typisierte Sprache Python in Erwägung gezogen, auch, weil es für Python eine recht ausgewachsen wirkende, unter einer Open Source-Lizenz bereitstehende Web-Scraping-Lösung namens 'Scrapy' gibt[1]. Es war jedoch zum einen zum Zeitpunkt der Technologiewahl noch nicht gänzlich klar, wie umfangreich die zu entwickelnde Lösung tatsächlich wird, weswegen ich mich nicht an eine derartige spezialisierte Lösung binden wollte. Zum Anderen habe ich mit Scala in der Vergangenheit sehr gute Erfahrungen gemacht und vor allem die Tatsache, dass es erstens über ein statisches und dennoch aus Programmierersicht sehr flexibles und leichtgewichtiges Typsystem verfügt, es zweitens in den allermeisten Fällen die leichte Einbindung bereits in großer Zahl existierender Java-Bibliotheken ermöglicht und drittens die funktionalen und asynchronen Programmierparadigmen sehr gut unterstützt, ohne dabei dogmatisch zu sein (es lässt sich mit Scala ebenso 'klassisch imperativ' wie rein objektorientiertes 'Java ohne Semikolons' schreiben). Insbesondere das für das funktionale Paradigma typische zustandslose Programmieren mit Verzicht auf Variablen (Scala kennt neben 'var' zur Variablendeklaration auch das Schlüsselwort 'val' zur Deklaration von unveränderlichen Werten) und das typische Aneinanderketten

[1] `https://scrapy.org/`

von elementaren, bereits aus dem Lambda-Kalkül bekannten Funktionen wie map, flatMap, filter oder reduce, welche auf elementaren, oft Listen-basierten Datenstrukturen arbeiten und auf unterschiedliche Weise vom Programmierer definierte Funktionen als weiteres Argument entgegennehmen, deren jeweiliger Scope/Funktionskörper oft sehr überschaubar ist, macht sowohl das Lesen auf Code-Ebene, als auch das Verständnis des Daten- und Kontrollflusses meiner Erfahrung nach leichter, als dies bei klassischer imperativer Programmierung der Fall ist. Funktionale Aspekte finden sich zwar immer mehr auch in ursprünglich imperativ angelegten Sprachen, jedoch ist Scala meiner Erfahrung nach von Grund auf mit diesem Ansatz konzipiert worden.

5.1.1 Reaktive, streamorientierte Programmierung

Die Kernidee des funktionalen Programmieransatz wurde in letzter Zeit durch ein weiteres Paradigma mit dem Namen 'reaktive, streamorientierte Programmierung' [2] aufgegriffen und weitergedacht. Nach wie vor stehen bei diesem Ansatz die Komposition von elementaren Funktionen auf der Grundlage von map, filter, flatMap, reduce etc. und Variationen davon im Vordergrund, jedoch werden diese nun nicht mehr auf während des Funktionsaufrufes hinsichtlich ihres Inhalt feststehende Listen-basierte Datenstrukturen angewandt, sondern auf zeitlich und bezüglich ihres Inhaltes offene Datenströme erweitert. Ich kann somit beispielsweise Funktionsketten schreiben, welche sich ganz ähnlich wie in klassischer funktionaler Programmierung mit Listen-basierten Datenstrukturen konzipieren lassen, aber die zur Laufzeit dynamisch auf neue 'Listenelemente' (genauer: neue Datenstromelemente oder Ereignisse) reagieren und die Funktionskette ebenfalls durchlaufen. Eine solche Funktionskette heisst im Kontext der von mir gewählten Lösung (akka-streams) Graph. Datenströme lassen sich in der reaktiven streamorientierten Programmierung beliebig aufteilen, an weitere, potentiell nebenläufige oder sogar verteilte Verarbeitungseinheiten weitergeben und dann wieder vereinigen und schliesslich von einer oder mehreren finalen Senken auf einem beliebigen Zielmedium (Datenbank, Dateisystem etc) persistieren oder an andere Dienste oder den Nutzer ausgeben. Ein wesentliches Konzept dabei ist das sogenannte 'backpressure'. Dieses ermöglicht es, einer in der Verarbeitungskette später angesiedelten Verarbeitungseinheit ein entsprechendes Signal an die vorherige Einheit zu geben, falls diese mit der Verarbeitung nicht nachkommen sollte, was in der Realität leicht zu Problemen wie z.B. Speicherüberläufen führen kann. Dieses Signal kann bis an die ursprüngliche Quelle durchgegeben werden, so dass die gesamte Verarbeitungskette stets Rücksicht auf das jeweils aktuell schwächste Glied nimmt. Insbesondere auch da am Anfang der Entwicklung zum Einen noch angedacht war, keine diskreten Zwischenstufen innerhalb des Crawlers einzubauen, wel-

[2] `https://www.informatik-aktuell.de/entwicklung/programmiersprachen/reactive-programming-mehr-als-nur-streams-und-lambdas.html`; für eine eher akademische Einführung in die Konzepte, siehe Kapitel 3 in `http://lup.lub.lu.se/luur/download?func=downloadFile&recordOId=8932146&fileOId=8932147`

che Zwischenergebnisse persistieren, und zum anderen beispielsweise mehrere, parallel schreibende finale Persistenzstufen geplant waren (zum Beispiel eine für die CSV-Ausgabe und eine für das direkte Schreiben in eine Datenbank, sobald ein Ergebnis die gesamte Crawling- und Extracting-Sequenz durchlaufen hat), wirkten die erwähnten Eigenschaften der reaktiven streamorientierten Programmierung auf mich überzeugend, weswegen ich mich für die Benutzung einer entsprechenden Bibliothek namens 'akka-streams'[3] entschied. Diese wird von der gleichen Firma (Lightbend) als Open Source-Software entwickelt wird welche auch maßgeblich an der Weiterenticklung von Scala beteiligt ist.

5.1.2 CSV als flexibles Persistenzmedium

Als Persistenzmedium wurde zuerst auf die dateibasierte, relationale Datenbanklösung sqlite gesetzt, nach anfänglichen Performance-Problemen mit der verwendeten Scala-sqlite-Bibliothek wurde jedoch auf einen einfachen CSV-Dateibasierten (comma separated values) Ansatz gewechselt, welcher auch ein Höchstmaß an Portabilität und einfachen Zugriff gewährleistet. Dabei wurden teilweise, aber nicht immer, Normalisierungsansätze aus dem Bereich der relationalen Datenbankmodellierung angewandt, insbesondere wenn es um die Persistierung von Beziehungen zwischen Entitäten geht. Eine strikte Normalisierung erschien mir jedoch als nicht notwendig, denn zum Einen handelt es sich um keine transaktionale Datenbank mit Schreibzugriffen bzw. Änderungen nach der erstmaligen Erstellung durch den Crawler, zum Anderen macht es dieser pragmatische Ansatz es dem Nutzer (im Kopf hatte ich dabei konkret oft einen Nutzer, welcher die Daten über die Programmiersprache R und der Software RStudio lädt) leicht, die Daten zu laden und dann selbst nach eigenem Bedarf entsprechend zu organisieren. CSV als Format ist nicht sonderlich komplex, dennoch sind einige Besonderheiten zu beachten, beispielsweise dass Spaltenwerte, die selbst ein Komma beinhalten, entsprechendes string escaping ausgezeichnet werden oder dass jede Zeile die korrekte Anzahl an Spalten (sprich: Kommas) aufweist. Diese Aspekte tangieren das Datenqualitätskriterium 'Syntaktische Validität der CSV-Ausgabedateien' und ich habe mich entschieden, eine etablierte und spezialisierte Scala-Bibliothek zum Lesen und Schreiben der CSV-Dateien zu verwenden[4].

5.1.3 Selektion der Felder mittels CSS und regulärer Ausdrücke

Um die Informationen aus den von der Gepris-Anwendung beschafften HTML-Seiten zu extrahieren, ist ein entsprechender HTML-Parser nötig, welcher die DOM-Konzepte (Document Object Model) in Scala zur Verfügung stellt. Somit können zum Beispiel anhand von CSS-Selektoren Felder ausgewählt und deren Inhalt extrahiert werden. Dafür wurde die Java-basierte und damit in Scala

[3] `https://doc.akka.io/docs/akka/2.5/stream/stream-introduction.html`
[4] `https://github.com/tototoshi/scala-csv`

ebenso nutzbare Bibliothek 'JSoup'[5] verwendet.

Bei der Untersuchung der DOM-Struktur der Gepris-Seiten hat sich herausgestellt, dass leider nicht für jeden Feldtyp ein eigener, eindeutiger CSS-Selektor auf Basis von eindeutigen CSS-Klassen und CSS-Ids existiert. Deswegen wurde zuerst versucht, die Feldnamen (also zum Beispiel 'Antragsteller' oder 'Mitverantwortliche Kooperationspartner' beziehungsweise deren englische Gegenstücke) mithilfe der CSS-Pseudoklasse ':matches' in die CSS-Selektoren mit einzubeziehen. Leider hatte sich jedoch herausgestellt, dass der Unterstützungsumfang dieses Konzeptes bezüglich regulärer Ausdrücke für einige Fälle nicht mächtig genug war und es teilweise zu mehrdeutigen Selektionen kam, was zu doppelten Wertselektionen führte. Deswegen wurde für bestimmte Problemfälle ein gemischter Ansatz gewählt: mittels JSoup und CSS-Selektion wurden zuerst anhand einer generischen Selektorregel alle Felder inklusive Feldnamen und Feldwert auf einer Geprisseite bestimmt, woraufhin mittels der vollständigen Unterstützung regulärer Ausdrücke durch Scala der jeweils gewünschte spezifische Eintrag extrahiert wurde (siehe nachfolgende Implementierung der Methode 'extractResourceIdsFromLinkByResourceTypeAndRegex', welche eine Liste von mittels CSS/JSoup bereits vorselektierten DOM-Elementen sowie eine Liste von regulären Ausdrücken erwartet).

```
1  def extractResourceIdsFromLinkByResourceTypeAndRegex(elements: Seq[
       Element])(resourceType: String)(regexes: Seq[String]) = {
2
3    // Scala uses the $ symbol for String interpolation.
4    // For escaping a dollar symbol as wanted part of the string, we can
         use $$
5    val enrichedRegexes = regexes.map(r => s"^\\s*$r\\s*$$")
6
7    enrichedRegexes.flatMap { regex =>
8
9      val resourceIdRegex = raw"""\/gepris\/$resourceType/(\d*)""".r
10     val matchedElements: Seq[Element] = elements.filter(s => s.text().
         matches(regex))
11
12     val matchedHrefs = matchedElements
13       .flatMap(
14         _.nextElementSibling()
15           .select("a")
16           .eachAttr("href")
17       )
18
19     val resourceIds = matchedHrefs.map { matchedHRefElement =>
20       matchedHRefElement match {
21         case resourceIdRegex(id) => id
22         case _ => ""
23       }
24     }.filterNot(_ == "")
25
```

[5] https://jsoup.org/

```
26        resourceIds
27      }
28    }
```

5.1.4 Docker als Container-Technologie

Neben der Datenqualität war eines der Kernziele, die Anwendung möglichst leicht in Betrieb nehmen zu können. Docker[6] als Container-Technologie bietet hier den Vorteil, dass alle Abhängigkeiten wie die Java Runtime Environment, die Scala-spezifischen Bibliotheken sowie alle anderen von der Anwendung genutzten Bibliotheken in Form eines einzelnen, plattformunabhängigen Docker-Images mit ausgeliefert werden können. Als einzige vom Nutzer zu installierende Abhängigkeit bleibt damit nur Docker selber. Auch die Beschaffung der Anwendung ist dann sehr leicht, sofern das entsprechende Image über ein zentrales Docker-Repository[7] zur Verfügung gestellt wird. In dem Fall reicht ein einfacher Befehl der folgenden Form:

```
1 docker pull spaudanjo/gepriscrawler:0.3
```

Hierbei ist 'spaudanjo' der Nutzer- beziehungsweise Organisationsname, 'gepriscrawler' der Repository-Name und '0.3' die Versionsnummer.

5.2 Architektur

Anfangs wurde angedacht, einen einzelnen Graphen komplett mit akka-streams umzusetzen. Das hiesse, die gesamte Crawling-Komponente wäre eine aus Subgraphen zusammengesetzte Funktionskette, welche den Datenstrom, angefangen vom Startbefehl der Anwendung, über die Beschaffung der zu crawlenden URLs, der Extraktion der Inhalte, bis hin zur Persistierung abbildet. Nach einigen Problemen mit der für mich noch neuen Technologie akka-streams und der Einsicht, dass es Sinn macht, nicht ausschliesslich erst am Ende das Gesamtergebnis der Verarbeitung (sprich: die fertigen CSV-Dateien mit den extrahierten Werten) zu persistieren, sondern auch bereits Zwischenschritte wie zum Beispiel die HTML-Dateien, sobald diese zur Verfügung stehen (damit eventuell abgebrochene Crawling-Durchläufe später wieder aufgenommen werden können), habe ich mich jedoch zu einer klaren Aufteilung in mehrere, komplett unabhängige Graphen entschieden. Dieser Ansatz reduzierte die Komplexität und die Probleme während der Entwicklung. Die Graphen repräsentieren dabei jeweils eine von mehreren Ausführungsstufen und sind in eigenen Ordnern (benannt nach dem Schema stage0 bis stage4) organisiert. Sie werden streng sequentiell nacheinander durch die Klasse GeprisCrawler.scala aufgerufen und jeder Graph speichert sein jeweiliges Verarbeitungsergebnis in einem eigenen

[6] `https://www.docker.com/resources/what-container`
[7] Beispielswiese docker-hub: `https://hub.docker.com/`

physischen Ordner, welcher dann wiederum als Eingabe für die jeweils nächste Verarbeitungsstufe und ihren Graphen dient.

Um möglichst schnell einen ersten Prototypen zu entwickeln und weil anfangs noch nicht komplett klar war, wie viel Differenzierungsbedarf bei der Behandlung der unterschiedlichen Ressourcentypen besteht, wurde zuerst für jeden der drei zentralen Ressourcentypen auf der Gepris-Website (Projekt, Institution, Person) in jeder Stufe jeweils ein eigener Graph entwickelt. Dieser Ansatz wurde jedoch aufgrund der starken Ähnlichkeiten in einen generischeren Ansatz geändert. Durch derartiges Refactoring konnte die Anzahl an Codezeilen von über 3000 auf unter 1800 gesenkt werden. Das nachfolgende Codebeispiel zeigt exemplarisch einen Ausschnitt aus dem Graphen der Stufe 1 (zu finden im Ordner stage0)[8].

```
1  val resourceIdsAndNameCsvWriterSink: Sink[CSVRow, Unit] =
       CrawlerHelpers.createCsvFileWriterSink(exportPath, s"${
       resourceName}_ids_and_names", Seq(s"${resourceName}_id", "name"
       ), append = false)
2
3  val cookiePinger: SourceShape[Boolean] = b.add(Source.repeat(false
       ))
4  val cookieFlow: FlowShape[Boolean, Cookie] = b.add(CookieFlowGraph
       .graph)
5
6  val numberOfResources = b.add(NumberOfResourcesGraph.graph(
       resourceName))
7  val numberOfResourcesBC = b.add(Broadcast[Int](2))
8  val paginatedResourceCatalogUrls = b.add(
       PaginatedResourceCatalogUrlsGraph.graph(resourceName))
9  val resourceCatalogPagesToResourceIdsToCrawl: FlowShape[(Cookie,
       String), (String, String)] = b.add(
       ResourceCatalogPagesToResourceIdsToCrawlGraph.graph(
       resourceName, resourceLinkCssSelector))
10 val zipCookiesWithPaginatedResourceCatalogUrls = b.add(Zip[Cookie,
        String])
11 val cookieBalancer = b.add(Balance[Cookie](2))
12 val resourceIdsAndNamesBC = b.add(Broadcast[(String, String)](2))
13
14 val numberOfResourceIdsToStatusFileSink = CrawlerHelpers.
       createTextFileWriterSink(s"$exportPath/number_of_${resourceName
       }s_to_crawl.txt")
15 cookiePinger ~> cookieFlow ~> cookieBalancer
16 cookieBalancer.out(0).take(1) ~> numberOfResources.in
17
18 numberOfResources.out ~> numberOfResourcesBC
19
20 numberOfResourcesBC.out(0) ~> paginatedResourceCatalogUrls.in
21 numberOfResourcesBC.out(1).map(_.toString) ~>
       numberOfResourceIdsToStatusFileSink
```

[8]Eine Übersicht der verwendeten akka-stream-Operatoren (wie Balance, Broadcast oder Zip) findet sich unter `https://doc.akka.io/docs/akka/2.5/stream/operators`

```
22      paginatedResourceCatalogUrls.out ~>
            zipCookiesWithPaginatedResourceCatalogUrls.in1
23      cookieBalancer.out(1) ~>
            zipCookiesWithPaginatedResourceCatalogUrls.in0
24
25      zipCookiesWithPaginatedResourceCatalogUrls.out ~>
            resourceCatalogPagesToResourceIdsToCrawl.in
26
27      resourceCatalogPagesToResourceIdsToCrawl ~> resourceIdsAndNamesBC
28      resourceIdsAndNamesBC.out(0).map(x => Seq(x._1, x._2)) ~>
            resourceIdsAndNameCsvWriterSink
29
30      SourceShape(resourceIdsAndNamesBC.out(1))
```

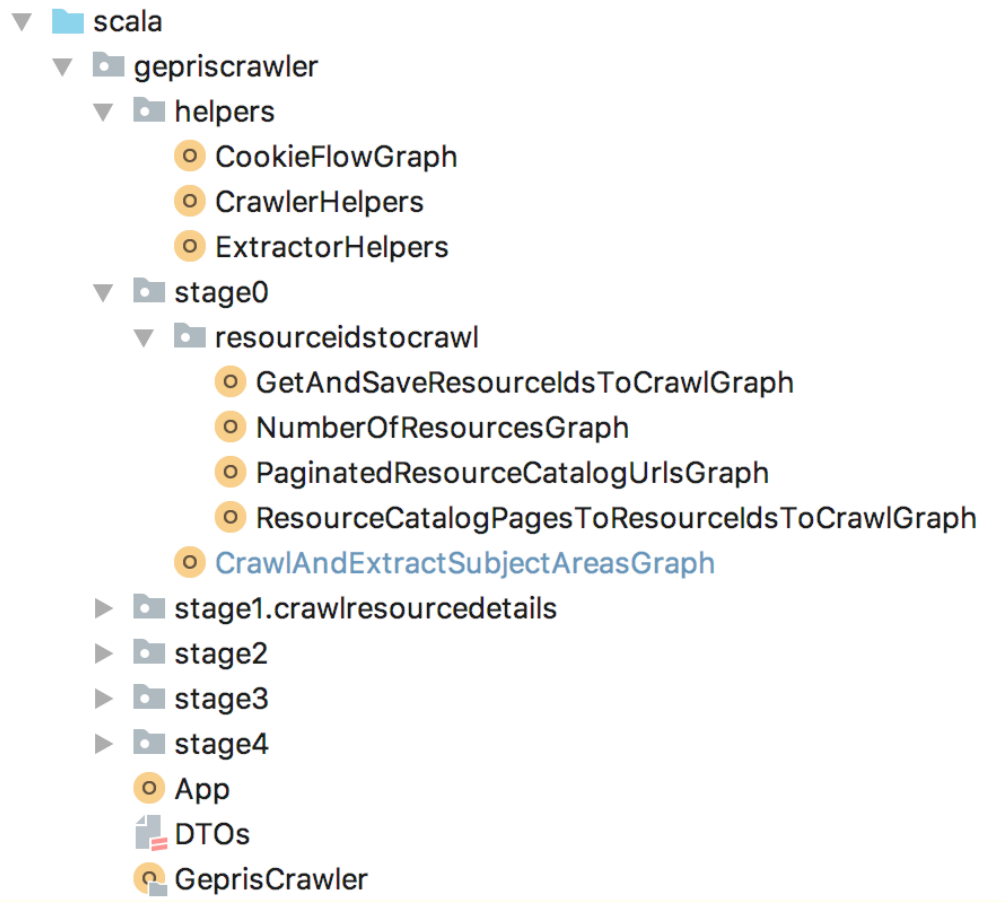

Abbildung 5.1: Übersicht der ersten Ebene der Quellcode-Organisation

6 Messung der Datenqualität mit R

Die ausgewählten Datenqualitätskriterien sollen hinsichtlich des Crawlers auch untersucht werden und wo sinnvoll bezüglich konkreter Crawling-Ergebnisse gemessen werden. Zuerst war angedacht, auch diese Messung im Rahmen der Crawler-Komponenten als eine weitere, finale Stufe als Graphen zu implementieren, welcher die verschiedenen Kriterien und die für sie definierten Regeln überprüft, misst und die Ergebnisse in Form von Textdateien mit abspeichert. Auch wenn ein prototypischer Versuch für das Kriterium 'Konsistenz bezüglich definierter Beziehungseinschränkungen' erfolgreich war (der Quellcode ist im Package stage4 zu finden), habe ich mich schlussendlich dafür entschieden, sämtliche Überprüfungen und Messungen im Rahmen eines R-Skripts umzusetzen, genauer gesagt in Form eines markdownbasierten R-Notebooks[1]. Die Überlegung dahinter ist, dass der Nutzer der Software so besser versteht, welche Kriterien wie geprüft werden und er die Prüfungen unabhängig vom Crawler nach belieben erneut ausführen und sie ebenfalls beliebig anpassen und erweitern kann. Darüber hinaus stellt dieser Ansatz dem User bereits einen ersten Einstiegspunkt für die Beschäftigung mit den Daten bereit, von wo aus er tiefergehende Analysen oder Visualisierungen vornehmen kann. Ein Vorteil der R-Notebooks ist auch, dass ihre Ausführung inklusive des Programmcodes, der Anmerkungen und der ermittelten Ergebnisse zum Beispiels mittels der 'Knit'-Funktion in RStudio als PDF oder HTML speichern lassen. Es entsteht somit eine Mischung aus prosaischer und programmatischer Dokumentation, welche die gemachten Aussagen leicht nachvollziehbar und reproduzierbar macht.

6.1 Beispiele für die Messung einiger Kriterien

Alle automatisch- oder semi-automatisch durchgeführten Überprüfungen sind in dem der Anwendung beiliegenden R-Notebook 'dataquality-checks.Rmd' zu finden und sie verwenden häufig das R-Paket 'dplyr'. Die folgenden abgedruckten R-Skripte sind Beispiele für die Prüfung einiger Qualitätskriterien und direkt dem erwähnten R-Notebook entnommen. Eine Einführung in 'dplyr', welches Bestandteil der Paketsammlung 'tidyverse' ist, findet sich zum Beispiel bei [Sta17][2].

[1] `https://rmarkdown.rstudio.com/r_notebooks`

[2] Abgerufen unter `http://joeystanley.com/downloads/171110-tidyverse_handout.pdf`

6.1.1 Kriterium 'Syntaktische Validität von Literalen einzelner Spalten'

Für ausgewählte Spalten/Felder kann eine einfache syntaktische Prüfung vorgenommen werden. Beispielsweise kann mittels regulärer Ausdrücke geprüft werden, ob Projekt-Ids (Zahlenwerte) oder Jahresangaben syntaktisch korrekt sind (vierstellige Zahlenwerte). Wir beschränken uns auf eine Regel, weitere können bei Bedarf nach dem gleichen Ansatz hinzugefügt werden.

Wir wollen hier exemplarisch die Regel "Für die Felder 'funding_start_year' and 'funding_end_year' sind nur 4-stellige Zahlenwerte erlaubt" auf Einhaltung hin überprüfen.

Zuerst schränken wir die zu untersuchende Mengen ein auf jene Fälle, welche überhaupt einen Wert für das Start- bzw. Endjahr zugewiesen bekommen haben. Dabei schliessen wir zum Beispiel die Werte "ongoing" und leere Zeichenketten aus:

```
dq_check_for_valid_funding_start_and_end_years = function(projects) {
  cases_with_start_year_defined = projects %>%
    filter(
      funding_start_year != "",
      funding_start_year != "ongoing",
      !is.na(funding_start_year)
    )

  cases_with_end_year_defined = projects %>%
    filter(
      funding_end_year != "",
      funding_end_year != "ongoing",
      !is.na(funding_end_year)
    )


  number_of_valid_start_years = grepl('\\d{4,4}',
      cases_with_start_year_defined$funding_start_year, perl = T) %>%
    sum

  number_of_valid_end_years = grepl('\\d{4,4}',
      cases_with_end_year_defined$funding_end_year, perl = T) %>%
    sum

  total_number_of_cases = nrow(cases_with_start_year_defined) + nrow(
      cases_with_end_year_defined)

  total_number_of_valid_cases = number_of_valid_start_years +
      number_of_valid_end_years

  dq_value = total_number_of_valid_cases / total_number_of_cases

  invalid_cases_for_start_year = cases_with_start_year_defined %>%
    filter(!grepl('\\d{4,4}', funding_start_year, perl = T))

  invalid_cases_for_end_year = cases_with_end_year_defined %>%
```

```
    filter(!grepl('\\d{4,4}', funding_end_year, perl = T))

  return(
    list(
      "dq_value" = dq_value,
      "invalid_cases_for_start_year" =
          invalid_cases_for_start_year$project_id,
      "invalid_cases_for_end_year" =
          invalid_cases_for_end_year$project_id
      )
  )
}
```

6.1.2 Kriterium 'Semantische Validität der Entitäten'

Ich möchte hier exemplarisch einen Test für die Regel "Für die Ressource 'Projekt' müssen die werte des Feldes 'funding_start_year' gleich oder kleiner sein als die Werte des Feldes 'funding_end_year' (sofern letzteres gesetzt ist)" in Form einer R-Funktion angeben:

```
dq_check_semantic_validity_of_enteties_start_funding_year_before_end_funding_year
     = function(projects) {

  projects_with_start_and_end_years = projects %>%
    filter(grepl('\\d{4,4}', funding_start_year, perl = T)) %>%
    filter(grepl('\\d{4,4}', funding_end_year, perl = T)) %>%
    mutate(funding_start_year = as.numeric(funding_start_year)) %>%
    mutate(funding_end_year = as.numeric(funding_end_year))


  total_number_of_cases = nrow(projects_with_start_and_end_years)

  invalid_cases = projects_with_start_and_end_years %>%
    filter(funding_start_year > funding_end_year)

  number_of_invalid_cases = nrow(invalid_cases)

  dq_value = (total_number_of_cases-number_of_invalid_cases) /
      total_number_of_cases

  return(
    list(
        "dq_value" = dq_value,
        "invalid_cases" = invalid_cases$project_id
      )
    )
}
```

6.1.3 Kriterium 'Vertrauenswürdigkeit auf Entitätsebene mittels Quellennachweis'

Für dieses Kriterium testen wir, für wie viele Entitäten der drei Resourcentypen wir die zugehörigen und vom Crawler zu speichernden HTML-Seiten der Gepris-Anwendung im Ordner 'stage1/RESOURCE_TYPE/html' auffinden können.

```
check_number_of_existing_html_files_for_ressource_type = function(
    resource_type, resource_ids) {
  filenames_for_resource = lapply(resource_ids, function(resource_id) {
      return(
    paste(root_html_path, "/", resource_type, "/html/", resource_id, ".
        html", sep = "")
    )})

  number_of_existing_files = sum(unlist(lapply(filenames_for_resource,
      file.exists)))
}
```

6.1.4 Kriterium 'Vollständige Populationsabdeckung'

Der Crawler speichert die Anzahl der von der Gepris bereitgestellten Entitäten pro Entitätstyp in der ersten Stufe des Crawlingvorgangs ab. Wir können diese Angabe dabei als Referenzwert nutzen und mit der Anzahl der vom Crawler bereitgestellten Entitäten vergleichen. Diesen Wert holt sich der Crawler dabei jeweils über die Suchfunktion des Gepris-Systems, welche die Gesamtanzahl der Entitäten der aktuell durchsuchten Ressource innerhalb des Navigationsmenüs angibt (vgl. Abbildung 6.1)

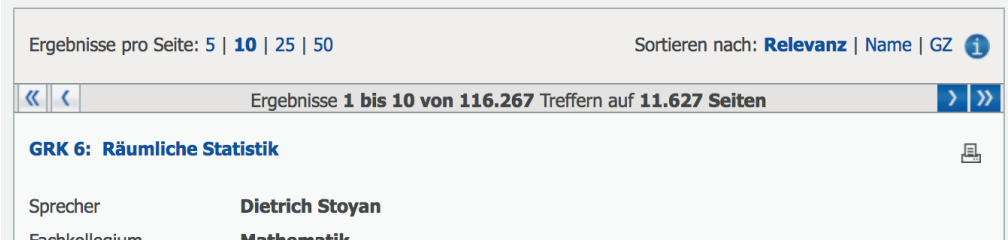

Abbildung 6.1: Die Navigationsleiste der Such- und Katalogseite

```
get_no_of_resources_from_txt_file = function(resource_type) {
  no_of_resources_in_gepris_website = as.numeric(
    str_replace_all(
      read_file(
        paste(root_path_for_number_of_ressources, "/number_of_",
            resource_type, "s_in_gepris_system.txt", sep = "")
      ), "[\r\n]" , ""
    )
```

```
  )
}

no_of_projects_in_gepris_website = get_no_of_resources_from_txt_file("
    project")
no_of_institutions_in_gepris_website = get_no_of_resources_from_txt_file
    ("institution")
no_of_persons_in_gepris_website = get_no_of_resources_from_txt_file("
    person")

no_of_projects_in_crawled_data = nrow(projects)
no_of_institutions_in_crawled_data = nrow(institutions)
no_of_persons_in_crawled_data = nrow(persons)

total_number_of_resources_in_gepris_website =
    no_of_projects_in_gepris_website +
    no_of_institutions_in_gepris_website +
    no_of_persons_in_gepris_website
total_number_of_resources_in_crawled_data =
    no_of_projects_in_crawled_data + no_of_institutions_in_crawled_data
    + no_of_persons_in_crawled_data

dq_value = total_number_of_resources_in_crawled_data /
    total_number_of_resources_in_gepris_website
```

7 Auswertung der Datenqualität

Ich möchte nun eine Zusammenfassung der Ergebnisse hinsichtlich der Datenqualität geben.

7.1 Syntaktische Validität der CSV-Ausgabedateien

Alle vom Crawler erzeugten CSV-Dateien liessen sich jederzeit mittels R öffnen und bestanden die Überprüfung durch den Onlinevalidator csvlint.io.

7.2 Syntaktische Validität von Literalen für einzelne Spalten

7.2.1 Geprüfter Aspekt: Korrektes Jahresformat

Es wurde die Tabelle projects.csv mittels der Regel "Für die Felder 'funding_start_year' and 'funding_end_year' sind nur 4-stellige Zahlenwerte erlaubt" überprüft.

7.2.2 Ergebnis

Dabei wurden keine Regelverletzungen festgestellt.

7.3 Semantische Validität der Entitäten

7.3.1 Geprüfter Aspekt: Korrekte Jahresangaben

Es wurde die Regel "Für die Ressource 'Projekt' müssen die Werte des Feldes 'funding_start_year' gleich oder kleiner sein als die Werte des Feldes 'funding_end_year' (sofern letzteres gesetzt ist)" geprüft.

Ergebnis

Von den 116267 Projekten wurde dabei eins gefunden, welches diese Regel verletzte. Zum Zeitpunkt der letzten Überarbeitung des R-Notebooks hat dieses Projekt [1] tatsächlich fehlerhafte Angaben zum Förderungszeitraum gemacht ("Term: from 2013 to 2012").

[1] `http://gepris.dfg.de/gepris/projekt/233526993?language=en`

7.4 Vertrauenswürdigkeit auf Entitätsebene mittels Quellennachweis

Alle Ressourcen wurden mit der entsprechenden Id abgespeichert und es lag ebenfalls für alle die zugehörige HTML-Datei mit abgespeichert vor.

7.5 Differenzierung zwischen leeren und unbekannten Werten

Dieses Kriterium konnte im Rahmen der Bachelorarbeit nicht mit der nötigen Sorgfalt beachtet werden.

7.6 Konsistenz bezüglich definierter Beziehungseinschränkungen

7.6.1 Geprüfter Aspekt: Korrekte Abbildung auf die DFG-Fachsystematik

Geprüft wurde die Regel "Alle Fachgebiete (subject_areas) aus den Projekten sind auch in der offiziellen DFG-Fachsystematik vertreten (gespeichert in der CSV-Datei subject_areas.csv)".

Ergebnis

Zum Zeitpunkt der letzten Überarbeitung dieses R-Notebooks (2018-10-25) waren 42,6% aller aus Projekten extrahieren Fachgebiete nicht in der offiziellen Fachsystematik zu finden.

Interpretation

Dies liegt daran, dass die Extractor-Logik des Crawlers nicht in allen Fällen korrekt funktioniert. Diese trennt die einzelnen Fachgebiete anhand von Kommas. Es kann aber passieren, dass ein Fachgebiet selbst Kommas enthält. Ein Beispiel dafür ist das folgende Fachgebiet: "Hydrogeology, Hydrology, Limnology, Urban Water Management, Water Chemistry, Integrated Water Resources Management"

In vielen Fällen werden auf den Gepris-Seiten jedoch die einzelnen Fachgebiete durch Kommas getrennt, in anderen Fällen jedoch durch Zeilenumbruch. Für das soeben als Beispiel genannte Fachgebiet wird beispielsweise im Falle des Projektes mit der Id 240126350 dieses Fachgebiet, welches selbst Kommas im Namen enthält, von dem zweiten Fachgebiet "Soil Sciences" durch einen Zeilenumbruch getrennt. Daher ist eine zuverlässige Umsetzung der Extractor-Logik schwierig. Eine Möglichkeit dies in der Zukunft zu lösen, könnte darin liegen, für jedes Projekt den gesamten String (also den kompletten Feldwert

auf der Webseite), welche die Fachgebiete beinhaltet, auf Vorkommnisse von Fachgebieten aus der offiziellen DFG-Fachsystematik (hinterlegt in der Datei "subject_areas.cvv" hin zu prüfen. Dies ist zwar hinsichtlich des Laufzeitverhaltens kostenintensiver als ein einfaches Trennen anhand von Zeilenumbrüchen und Kommas, jedoch akkurater.

Doch auch mit diesem Ansatz würden einige der Problemfälle bestehen bleiben: Es gibt auch Fälle, wo die nicht gelungene Zuordnung nicht auf eine Extractor-Logik zurückzuführen ist, welche nicht alle Besonderheiten abdeckt, sondern auf Inkonsistenzen seitens der Gepris. Beispielsweise wird im Falle des Projektes mit der Id 5122166 direkt auf der Webseite [2], das Fachgebiet "Animal Physiology and Biochemistry" angegeben, für welches es keine Entsprechung innerhalb der Tabelle subjec_areas.csv (die offizielle DFG-Fachsystematik) gibt.

7.6.2 Geprüfter Aspekt: Keine 'toten' Verweise

Geprüft wurde die Regel "Für alle Personen-Ids und Institutions-Ids, welche in einer Beziehungstabelle auftauchen, sind auch in der entsprechenden Primärtabelle (persons bzw. institutions) vertreten".

Ergebnis

Zum Zeitpunkt der letzten Überarbeitung dieses R-Notebooks (2018-10-25) wurden für 'Institutionen' keine Regelverletzungen festgestellt.

Im Falle von 'Personen' wurden zwei Personen-Ids ohne Einträge in der Personen-Tabelle gefunden:

- Person mit der Id 282670177 für das Projekt mit der Id 282669151
- Person mit der Id 285790938 für das Projekt mit der Id 285789434

Interpretation

In beiden Fällen handelt es sich bei der Rolle der Personen um ausländische Kooperationspartner. Scheinbar sind diese beiden Personen bisher nicht über den Personenkatalog der Gepris-Anwendung auffindbar und wurden daher vom Crawler nicht erfasst.

7.7 Vollständige Schemaabdeckung

Der manuelle Abgleich hat hierbei ergeben, dass für folgende Projekt-bezogene Felder, extrahiert auf Basis der generischen CSV-Datei "generic_field_extractions.csv" keine Entsprechung in einer der explizit Projekt-bezogenen CSV-Dateien gefunden wurden:

- DFG programme contact

[2] `http://gepris.dfg.de/gepris/projekt/5122166?language=en`

- Major Instrumentation
- Instrumentation Group

Dabei ist anzumerken, dass das Feld “Subproject of” umbenannt wurde in “parent_project_id”, das Feld “term” aufgeteilt wurde in die Felder “funding_start_year” und “funding_end_year” und Namensvariationen von in semantischer Hinsicht ein und demselben Feld auf einen einzigen, einheitlichen Feldnamen abgebildet werden.

Darüber hinaus sind andere, eher “versteckte” Informationen extrahierbar, welche von dem hier vorgestellten Ansatz mittels des generischen CSS-Selektors nicht erfasst werden. So sind zum Beispiel verstorbene Personen mit einem kleinen Kreuz hinter ihrem Namen markiert und der Text zur Auswertung von abeschlossenen Projekten befindet sich auf einer jeweils gesonderten HTML-Seite. So müssten diese Information streng genommen auch als “Feld” mit in die Auswertung und, sofern gewünscht, mit in den Crawler übernommen werden, derzeit werden beide Informationen vom Crawler nicht erfasst.

7.8 Vollständige Spaltenbelegung

Im Zuge des manuellen Abgleichs mit den Originalseiten der Gepris wurden anhand einer kleinen Stichprobe folgender Mangel entdeckt: Die Antragsteller (Applicants) für das Projekt mit der Id 40157239[3] wurden nicht extrahiert.

Für die Qualitätsmetrik wurde insgesamt folgender Wert anhand der (sehr kleinen) Testprobe gemessen, welche lediglich einen Anhaltspunkt für weitere, größere Proben geben kann: 80% der getesteten Beispielprojekte haben sämtliche Spalten vollständig und korrekt belegt.

7.9 Vollständige Populationsabdeckung

Die Messung hat ergeben, dass sämtliche von der Gepris über die Katalogseiten zur Verfügung gestellten Ressourcen vom Crawler erfasst und bereitgestellt wurden.

7.10 Unterstützung von Mehrsprachigkeit

Ursprünglich wurde die deutsche Version gecrawled, im Laufe der Arbeit habe ich aber die Selektor-Logik auf die englische Version umgestellt. Eine gleichzeitige Unterstützung beider Sprachen ist derzeit nicht implementiert.

[3] `http://gepris.dfg.de/gepris/projekt/40157239?language=en`

7.11 Verständlichkeit und Dokumentation der Ausgabedateien

Die CSV-Dateien haben selbsterklärende Namen und Spaltenbezeichnungen und es steht eine Einführung in das Domänenmodell im Rahmen der Datei 'READE.md' bereit.

7.12 Bereitstellung in mehreren Datenformaten

Es steht im Unterordner eine 'stage3/sqliteexport' eine Beispielimplementierung für den Export in eine sqlite-Datenbank bereit. Diese Funktion wurde jedoch in der aktuellen Version des Crawlers deaktiviert, da ich momentan keinen Bedarf für ein zusätzliches Exportformat sehe und die Laufzeit des Crawlers durch den Export somit unnötig erhöht werden würde.

7.13 Validität der ursprünglichen Gepris-Seiten-URLs

Auf Grundlage von 100 zufällig bestimmten Ressourcen wurden die zugehörigen URLs automatisch aufgerufen und deren Ergebnis auf die von der Gepris-Anwendung genutze Fehlermeldung hin untersucht. In keinem der Fälle wurde eine nicht mehr zugängliche und damit invalide URL identifiziert.

8 Zusammenfassung und Ausblick

Im Rahmen der vorliegenden Arbeit wurden zwei zentrale Artefakte entwickelt: Zum einen die Entwicklung einer möglichst leicht in Betrieb zu nehmenden Crawler-Software, welche die von der DFG bereitgestellten Daten bezieht, verarbeitet und als strukturierten Datensatz bereitstellt. Zum anderen die Beschreibung einer Reihe von Datenqualitätskriterien für die Domäne des Crawlers, sowie deren Auswertung und der Einbezug bei der der Entwicklung des Crawlers. Dazu wurde auf die auf Wang[WS96] und Färber[FBMR17] zurückgehende Systematisierung der Datenqualität zurückgegriffen.

8.1 Einsichten bezüglich der Datenqualität und der Implementierung

Insgesamt ist festzustellen, dass der entwickelte Crawler in seiner aktuellen Form die meisten, aber nicht alle der ausgewählten Qualitätskriterien für die überwiegende Menge der Datensätze einhält. In einigen Fällen ist die mangelnde Datenqualität auf Inkonsistenzen seitens der DFG zurückzuführen. Eines der auffälligsten Beispiele ist die nicht gelungene Abbildung der Fachgebiete der Projekte auf die offizielle Fachsystematik. Fast alle von der DFG bereitgestellten Felder werden von dem Crawler extrahiert, weitere Extractor-Regeln können relativ leicht eingebaut werden.

Im Sinne einer Reflexion meiner Herangehensweise hat sich bei mir vor allem folgende Erkenntnis eingestellt: der ursprüngliche Versuch, eine allumfassende Lösung in Form einer Konsolenanwendung zu schreiben, welche auch die Datenqualitätsbewertung vornimmt, hat sich als ineffizient erwiesen, weswegen ich von diesem Ansatz im Laufe der Entwicklung Abstand genommen habe und die Bewertung ausschliesslich auf Basis von R vorgenommen habe. Vermutlich würde es sogar Sinn machen, die Konsolenanwendung von der Aufgabe des Extractings soweit wie möglich zu befreien und dieses ebenfalls in R vorzunehmen (im Sinne von "Der Crawler sollte crawlen und R ist das richtige Werkzeug für Fragen zur Datenqualität und zur Analyse"). Mit der Wahl der reaktiven, streamorientierten Bibliothek 'akka-streams' bin ich insgesamt zufrieden, auch wenn es eine erhebliche Lernkurve verursachte, ich zumindest in einem Fall ein zu lösendes Problem nicht in angemessener Zeit lösen konnte und mir andere Anwendungsfälle für diese Bibliothek als angemessenere Kandidaten erscheinen. Vor allem in Fällen von verteilten und nebenläufigen Akteuren mit starken Fluktuationen des Verarbeitungsdurchsatzes zwischen diesen bietet sich diese Technologie an.

Anfangs erschien mir die systematische und formelle Auseinandersetzung

mit den Qualitätskriterien zumindest in einigen Fällen als unangemessen, oft wurden Aspekte behandelt, welche mir intuitiv klar erschienen. Viele der vorgestellten Kriterien sind hinsichtlich der Sicherstellung vor allem durch eine nach softwaretechnischen Maßstäben möglichst fehlerfreien und sauberen Entwicklung der Crawling- und Extractor-Funktionen sicherzustellen und auch die Messung bzw. das Überprüfen auf Einhaltung der Kriterien ist in vielen Fällen in der Praxis nicht sonderlich kompliziert. Im Vordergrund stand jedoch im Wesentlichen eine Sensibilisierung für die Thematik der Datenqualität und eine systematische Bestimmung nötiger Kriterien, deren Einhaltung dann geprüft werden kann, wodurch mögliche noch vorhandene Schwächen transparent gemacht werden. Insgesamt habe ich dadurch einen nun geschulteren Blick für diesen vermutlich oft vernachlässigten Aspekt bei der Entwicklung von datenzentrierten Anwendungen und ein Gefühl für geeignete methodische Ansätze zur Einhaltung und Evaluierung der Datenqualität.

8.2 Ausblick

Als mögliche weitere Schritte für die Erweiterung der Lösung kommen zum Beispiel in Frage:

- eine Erfassung der Beziehungen zwischen Projekten und Institutionen auch ausgehend von den Institutionsseiten und nicht nur ausgehend von den Projektseiten (es hat sich herausgestellt, dass in vielen Fällen für Projekte, welche auf einer Institutionsseite im Gepris-System als mit dieser Institution in Beziehung stehend genannt werden, diese Angaben nicht auch auf der jeweiligen Projektseite genannt werden)

- Ein Erfassen und Abspeichern der Information, ob eine Person verstorben ist (passiert in Form eines kleinen Kreuzes neben dem Namen der Person)

- Ein Crawling und Bereitstellen der Fachsystematik wie sie auf der Katalog- und Suchseite der Gepris-Website aufgebaut ist - diese unterscheidet sich von der offiziell veröffentlichten Fachsystematik, welche von meiner Implementierung momentan benutzt wird

- gleichzeitige Unterstützung für sowohl die englische als auch die deutsche Version der Gepris-Daten

- Unterstützung einer Änderungs-Historie, so dass verschiedene Crawling-Ergebnisse miteinander verglichen und Änderungszeiten erkannt werden können

- die Implementierung einer Recovery-Strategie, so dass der Crawler im Fehlerfall nicht manuell neugestartet werden muss

- die Bereitstellung der Anwendung als im Hintergrund oder auf einem Server laufender Service, welche zum Beispiel als cronjob den Crawling-vorgang regelmässig und ohne weiteres Eingreifen selbstständig startet

Literaturverzeichnis

[FBMR17] Michael Färber, Frederic Bartscherer, Carsten Menne, and Achim Rettinger. Linked data quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO. *Semantic Web*, 9:1–53, March 2017.

[Kan02] Stephen H. Kan. *Metrics and Models in Software Quality Engineering*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 2002.

[Sta17] Joey Stanley. An Introduction to Tidyverse. 2017.

[WS96] Richard Y. Wang and Diane M. Strong. Beyond Accuracy: What Data Quality Means to Data Consumers. *J. Manage. Inf. Syst.*, 12(4):5–33, March 1996.