Gottfried Wilhelm
Leibniz Universität Hannover
Fakultät für Elektrotechnik und Informatik
Fachgebiet Data Science and Digital Libraries

Facettensuche Für das Suchsystem von Open Research Knowledge Graph (ORKG)

Bachelorarbeit

im Studiengang Informatik

von

Ahmad Ramadan
Martrikelnummer: 10015970

Prüfer: Prof. Dr. Sören Auer
Zweitprüfer: Dr. Markus Stocker
Betreuerin: Golsa Heidari

Hannover, 31. März 2021

Erklärung der Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit selbständig und ohne fremde Hilfe verfasst und keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel verwendet habe. Die Arbeit hat in gleicher oder ähnlicher Form noch keinem anderen Prüfungsamt vorgelegen.

Hannover, den 31. März 2021

Ahmad Ramadan

Danksagung

Zuerst möchte ich Herren Prof. Sören Auer danken, für seine Bereitschaft, meine Arbeit zu prüfen und mich an seinem Institut zu betreuen.
Außerdem danke ich Herren Dr. Markus Stocker, der mir stets bei Fragen und zur Orientierung zur Seite stand.
Besonders danke ich meine Betreuerin Frau Golsa Heidari, für die vielen hilfreichen Ratschläge und Zielsetzungen.
Abschließend möchte ich mich bei allen Mitglieder von dem ORKG Team für ihre Unterstützung bedanken.

Zusammenfassung

In dieser Arbeit wird versucht, das Prozess der Informationssuche zu optimieren. und den damit verbundenen Zeitbedarf zu reduzieren. Dafür wird auf dem Open Research Knowledge Graph (ORKG) zurückgegriffen. Hier ist es möglich, wissenschaftliche Papers als wissen Graphen zu darstellen, und diese Anhand der gemeinsamen Properties zu vergleichen.

Dabei werden eine große Anzahl an Kontributionen gleichzeitig verglichen. Somit entsteht das sogenannten Information Overload Problem. Hier ist es aufgrund der großen Menge an Informationen sehr schwer die gesucht Information zu finden und extrahieren. Um diese zu lösen, wird in dem Comparison Ansicht bei der ORKG ein Faceted Search System integriert.

Dieses System soll dem Nutzer Helfen, durch die Eingabe von mehreren Kriterien die Anzahl der vorhanden Kontributionen zu minimieren. Da Wissen Graphen sehr heterogen sind, soll das Faceted Search System dynamisch bleiben. Hierfür werden die Daten in verschiedene Klassen unterteilt, um für jede diese Klassen eine bestimmte Eingabeoberfläche anzubieten. Das gewährleistet, dass dieses System einfach zu bedienen bleibt, und somit wird der Nutzer davon nicht abgelenkt.

Es ist dabei wichtig, dass der Nutzer die gesuchte Informationen vorher kennt. Damit wird dem Nutzer gelingen, den Informationsbedarf in ein oder mehreren Kriterien umzuwandeln. Die wiederum mit Hilfe des Faceted search Systems in Informationen umgewandelt werden.

Abstract

In this work, an attempt is made to optimize the process of information searching, and to reduce the associated time required for that. For this purpose the ORKG is used. Here, it is possible to represent scientific papers as knowledge graphs, which could be compared on the basis of common properties.

Thereby a large number of contributions are shown at the same time. And because of that, the so-called problem of information overload arises. Due to the large amount of information, it is very difficult to find and extract some desired informations. To solve this, a faceted search system is integrated in the comparison view at the ORKG.

This system should help the user to minimize the number of available contributions by entering specified criteria. Since knowledge graphs are very heterogeneous, the faceted search system should remain dynamic. For this purpose, the ORKGs data is divided into different classes in order to provide each class of them a specific input interface. This ensures that the system remains easy to use and does not distract the user from using it.

It is important that the user knows the information he is looking for beforehand. This will help the user to convert his information needs into one or more criteria. Which will be converted back into information using the Faceted search system.

Inhaltsverzeichnis

Akronyme

GUI Graphical User Interface

IR Information Retrieval

ML Machine Learning

ORKG Open Research Knowledge Graph

REGEX Regular expressions

UX User Experience

Kapitel 1

Einleitung

Heutzutage ist die Informationsforschung mit viel Zeitverlust verbunden. Neben der riesigen Menge an Wissenschaftlichen Wissen spielen nicht digitale Informationsquellen dabei eine große Rolle. Dieses Prozess an Informationsrecherche ist sehr Mühsam und beansprucht viel Zeit. Dieses Problem lässt sich mit der Digitalisierung bekämpfen. Dabei können zur Verbesserung der Situation automatisierte Algorithmen eingesetzt werden.

Das ORKG[1] bittet eine Schnittstelle, die den Wissenschaftlern ermöglicht Publikationen strukturiert und semantisch zu beschreiben. Unter anderen können diese dann von Maschinen interpretiert und automatisiert verarbeitet werden. Dabei können Kontributionen miteinander verknüpft werden [4].

Verschiedene Kontributionen können mittels Machine Learning (ML) Algorithmen auf Ähnlichkeit überprüft werden und miteinander tabellarisch vergleichen. somit lässt sich ein Vergleich von vielen Beiträgen, die sich mit einem bestimmten Forschungsproblem beschäftigen, automatisch erstellen. Während diese einem Forscher wertvolle visuelle Informationen vermittelt, entsteht dabei eine großes Problem nämlich Informationsüberflutung. da die Anzahl an vergleichbaren Kontributionen meistens sehr groß ist, lassen sich spezifische Informationen nur schwer extrahieren. Dieses Problem kann auf verschiedene Arten gelöst werden. Unter anderem können solche Tabellen mittels Eingabe von einer spezifischen Abfrageon the fly erstellt werden. Zum Beispiel durch mehrere Wörter, die den Informationsbedarf beschreiben. Dies ist jedoch ineffizient, da für die Erstellung eines solchen Vergleichs ML-Algorithmen verwendet werden, die viel Rechenressourcen verbrauchen. Eine deutlich effektivere

[1] `https://www.orkg.org`

Lösung für das Problem ist ein Faceted Search System. Während sich bisher der Nutzer alle Kontributionen gleichzeitig anschaut, könnte mithilfe bestimmter Kriterien die Anzahl an Kontributionen begrenzt weerden und für den Nutzer uninteressante Kontributionen eliminiert werden. Wodurch im Endeffekt das Problem mit Informationsüberflutung nicht mehr entsteht [6] und die Zeitaufwand für die Informationsuche verringert wird.

1.1 Problemstellung

Bei dem Integrieren von einem Faceted Search System in ORKG kommt man auf verschiedene Probleme: Die Kontributionen sind zwar strukturiert, jedoch sind die einzelne Daten nicht Typ-sicher. Beispielsweise hat das Forschungsbeginnsdatum der identische Typ wie das Forschungsstandort. d.h.

$$type(\text{"}01.12.2020\text{"}) == type(\text{"}Germany\text{"})$$

Da verschiedene Datentypen verschiedene Eingabeoberfläche beanspruchen müssen zunächst Methoden definiert werden, die jeweils den Datentyp erkennt und dann entsprechend die sinnvollste Eingabemethode anzubieten.

Außerdem wird an dem ORKG immer noch weiterentwickelt, was eine große Herausforderung stellt, nämlich soll bei der Entwicklung der Faceted search system viel Wert auf die Wartbarkeit und Erweiterbarkeit gelegt werden.

Da die Visualisierung des Vergleichs zwischen mehreren Kontributionen viel Platz beansprucht, soll das Graphical User Interface (GUI) von dem Faceted Search System möglichst wenig platz aufnehmen jedoch muss es sowohl einfach zu bedienen als auch intuitiv bleiben.

Zusammengefasst ist das Ziel dieser Arbeit die Entwicklung vom einem Faceted Search System, das einem Wissenschaftler die Möglichkeit bietet, ein eher Allgemeineren Vergleich von Mehreren Kontributionen durch Eingabe von bestimmte Kriterien auf die Kontributionen, die aus Sicht dieser Wissenschaftler für einen spezifischen Forschungsproblem am interessantesten sind unter Berücksichtigung der Wartbarkeit und Erweiterbarkeit dieses System.

1.2 Lösungsansatz

Bei der Entwicklung Vom Faceted search system muss zunächst in der vorhanden Vergleichsseite (engl. Comparison Page[2]) ein benutzerfreundliche Eingabeoberfläche integriert werden, mit dessen Hilfe ein Nutzer bestimmte Kriterien einfach und Präzise auswählen kann, ohne vom Ziel, nämlich die gesuchte Information zu finden, abgelenkt zu werden. Hierfür soll jeder Property zu dem richtigen Datentyp einsortiert werden. um diese jeweils geeignete Eingabemethoden zuzuordnen, beispielsweise können numerische Attributen mit Vergleich-Operatoren wie „größer als“ begrenzt werden, während diese bei textuellen Attributen nicht möglich wären. Dabei sind Zum klassifizieren Von Datentypen Approximationsalgorithmen gut geeignet. Diese sind Methoden die keine 100 prozentige Lösung liefern. Außerdem lassen sich standardisierte Dateneinträge mittels Regular expressions (REGEX) gut erkennen.

Nach der Validierung der eingegebenen Kriterien, werden diese mit passenden Methoden bearbeitet und die Kontributionen, die Vom Nutzer ausgewählte Kriterien erfüllen, werden ermittelt.

Anschließen muss ein Algorithmus Zum Filtern von Kontributionen einwickelt werden. Dabei sollen die unerwünschte Kontributionen eliminiert werden. Diese lassen sich durch Unerfüllbarkeit der vom Nutzer definierten Kriterien identifizieren.

1.3 Struktur der Arbeit

Der weitere Verlauf dieser Arbeit umfasst fünf Kapitel. In dem Kapitel 2 Werden die Verwandte Arbeiten zu Faceted Search Systeme besprochen. Kapitel 3 stellt das Open Research Knowledge Graph dar. Kapitel 4 verdeutlicht die Durchführung des Faceted Search Systems. bei dem darauf folgenden Kapitel 5 werden die Ergebnisse diskutiert. Das Kapitel 6 bildet eine Zusammenfassung der Arbeit und gibt Möglichkeiten für Zukünftige Arbeiten.

[2]Ein Beispiel ist Hier zu finden: `https://www.orkg.org/orkg/comparison/R44930`

Kapitel 2

Verwandte Arbeiten

Der Hauptpunkt dieser Arbeit ist die Entwicklung von einem Faceted Search System, Da Diese Thema bereits intensiv geforscht ist, ist es wichtig verschiedene Ansätze anzuschauen. dadurch werden wertvolle Einblicke gewonnen.

Eine umfangreiche Quelle wurde von Tunkelang schon erfasst [5]. Die grundlegende Konzepte von einem Faceted Search System sind hier ausführlich beschrieben. Diese bietet Forscher, die sich mit dem Fachgebiet Information Retrieval (IR) nicht vertraut sind, eine solide Basis.

Zudem beschreibt er mehrere Beispiele Zur Entwicklung von einem Faceted Search System, diese sind sowohl aus Akademische als auch kommerzielle Forschung entstanden.

Außerdem werden verschiedene Orientierungspunkte, die bei dem Erstellen von dem GUI zu beachten sind, präsentiert.

Abschließend stellt er verschiedene Herausforderungen vor, die beim Realisieren einen solchen System zustande kommen könnten. Dabei werden diese um wertvolle Best-Practices und Tipps ergänzt.

Wei et al. [6] stellen ein allgemeines Faceted Search Framework vor. Darunter beschreiben sie wichtige Methoden und Techniken. Unter anderem wird dargestellt, wie Facetten aus unstrukturierte Daten extrahiert werden können.

Um Nutzern mit Informationen nicht zu Überfluten, muss nur eine begrenzte Anzahl an Facetten angeboten, Dabei sollen nur die Facetten, die am wichtigsten sind, angezeigt werden. Hierfür beschreibt Wei et al. wie ein solches Prozess realisiert werden kann.

Während diese Arbeit viele wertvolle Ansätze anbietet, lassen sich die

hier verwendeten Methoden sehr begrenzt auf dem ORKG anzuwenden. Nämlich sind diese Techniken für unstrukturierte Daten geeignet. Es handelt sich bei dem ORKG um strukturierte Daten.

Jedoch sind einige dieser Methoden für das Filtern von Rich-text Attributen passend. Beispielsweise könnte eine Auswahl an Facetten von solche Attributen generiert werden, diese müssen zunächst auf die Nützlichkeit eingestuft. Dem Nutzer wird am ende nur die am wichtigsten Facetten zur Auswahl angezeigt.

Ein weiteres bemerkenswertes Paper im Bezug auf der Dynamischen Erstellung von Facetten über Wissensgraphen wurde von Feddoul et al. produziert [1]. Diese beschäftigt sich im Gegenteil zum vorherigen Arbeit mit strukturierte Daten. Nämlich Linked Data. Ähnlich zum [6] werden hier Methoden sowohl zum Generieren der Facetten als auch zum zu Rangieren der Facetten.

Hier wird ein Workflow für die Entwicklung eines Faceted Search Systems definiert. In der ersten Phase müssen Facettenkandidaten generiert werden. Dabei werden sowohl direkte als auch indirekte Eigenschaften berücksichtigt. Wobei indirekte Eigenschaften ab bestimmten Pfadlänge nicht mehr angenommen werden. Zudem werden numerische literalen in mehrere Bereiche umgewandelt.

Schließlich werden die Facettenkandidaten auf die Nützlichkeit intra eingestuft. Dabei sind die Facettenkandidaten ab einer bestimmten Wertung überflüssig, und somit wird die Anzahl Facettenkandidaten reduziert. Die restliche Facettenkandidaten werden kategorisiert.

Anschließend werden die Facettenkandidaten auf Ähnlichkeit inter eingestuft. Hier werden vor allem Facettenkandidaten, die semantisch sehr nah zu einander sind, eliminiert. In [2] wird dieses Verfahren umfangreicher beschrieben.
Es stellte heraus. Während die meisten dieser Arbeiten wertvolle Ansätze Zur Entwicklung von einem Faceted Search System anbieten, müssen zur Integration in dem ORKG verschieden Anpassung vorgenommen werden. In dieser Arbeit soll für bestimmte Datentypen spezielle Eingabemethoden entwickelt werden, dieses soll ermöglichen dem Nutzer präzise aber gleichzeitig schnell die Anzahl an Ergebnisse zu begrenzen. Zum Beispiel könnte eine Anfrage so aussehen:

$$(1 < x \leq 4) \wedge (x \neq 2)$$

x steht hier für eine ordinäre Property.

Solche Kriterien lassen sich durch automatisch generierte Facetten nicht beschreiben. Hierfür sollen die Ergebnisse einer Anfrage online berechnet.

Kapitel 3

Einführung in das Open Research Knowledge Graph

Das ORKG bittet eine Plattform, die ermöglicht wissenschaftliche Paper in eine strukturierte und semantische Darstellung zu beschreiben. Auf dieser Weise sind die wissenschaftliche Kontributionen einfacher zu finden und vergleichen. Dank der gewonnen strukturierten Schema lassen sich diese Kontributionen mittels automatisierte Algorithmen bearbeiten.

Da automatisierte Methoden sich für die Erstellung solcher Daten nicht einigt. Wird es bei dem ORKG viel Wert auf das Crowdsourcing gelegt. Diese hat den großen Vorteil, dass die Daten eine gute Qualität haben, im Vergleich zu Daten, die automatisiert erstellt wurden. Jedoch lässt sich hiermit das Multi-Source Problem nicht vermeiden. Dazu gehören Probleme wie nicht standardisierte Daten und Namenskonflikte.

Ein der wichtigsten Features, die von dem ORKG angeboten wird, ist die Erstellung vom sogenannten State-of-the-Art Comparison [3]. Hierfür werden mehrere Kontributionen in einem Forschungsgebiet auf die semantische Ähnlichkeit geprüft. Dadurch kann dann eine Comparison generiert werden. Eine Comparison besteht aus mehreren Kontributionen, die anhand gemeinsamen Eigenschaften (engl. Properties) tabellarisch dargestellt werden. Jede Kontribution hat eine eindeutigen Identifier. Diese heißt ContributionsID. Verschiedene Kontributionen haben verschiedene Strukturen und damit entsteht das Bedürfnis an einem dynamischen Facted Search System. Diese Soll dem Nutzer helfen die Anzahl der Kontributionen zu reduzieren. das geschieht mittels Konfiguration von mehreren Kreieren im Zusammenhang mit den verfügbaren Properties.

Kapitel 4

Methodik

Wie bereits in der Einleitung erläutert, ist das Ziel dieser Arbeit das Integrieren von einem Faceted Search System innerhalb der State-of-the-Art Comparison, das vom ORKG angeboten wird.

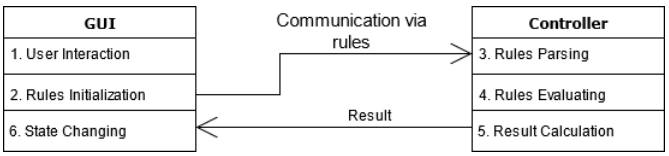

Abbildung 4.1: Die Architektur des Faceted Search System

Abbildung 4.1 beschreibt die Faceted Search System Architektur, das innerhalb dieser Arbeit implementiert wurde. Dieses System besteht aus zwei separaten Klassen, die mittels sogenannten Regeln kommunizieren: Das Graphical User Interface und der Controller.

Eine Regel ist ein Objekt mit vordefiniertem Struktur. Diese beschreibt ein Kriterium im Zusammenhang mit einem Property. Somit können Kontributionen anhand deren Properties und dazugehörigen Wert eine Regel erfüllen oder auch nicht. Ein Regel besteht im Allgemeinen aus drei Attributen:

- propertyId: Diese ist für jede Property Eindeutig
- type: Verschiedene Datentypen haben verschiedene Regeltypen
- value: Ist der Wert mit der, die Attributen von einer Kontribution verglichen werden.

Beispielsweise kann ein Regel, die nur Kontributionen mit einem Studienbeginn Datum nach dem 2020-04-03 akzeptiert, so aussehen:

- propertyId: *P15699*

- type: *gteDate*
- value: *2020-04-03*

Das GUI realisiert die Interaktion mit dem Nutzer. Dort wird anhand des Datentyps von einem Property eine möglich passende Eingabeoberfläche angeboten. Hierdurch gelingt dem Nutzer die Erstellung von Regeln. Anschließend können diese Regeln an den Controller weitergeleitet.

Dort werden die ausgewählten Regeln Interpretiert. Zuletzt werden die Kontributionen berechnet, die die eingegebene Regeln erfüllen. und somit können schließlich die unerwünschten Kontributionen im GUI eliminiert werden.

Im Folgenden Abschnitte werden die einzelnen Komponenten von diesem System detaillierter besprochen.

4.1 Entwurf der graphischen Oberfläche

Eine Schnittstelle, die sich schwer bedienen lässt, macht dem Nutzer keinen guten Eindruck. Solche Schnittstelle sind oft mit schlechtem User Experience (UX) verbunden. und sollen im allgemeinen vermieden werden. Um das UX zu optimieren werden zunächst drei verschieden layoutsmöglichkeiten untersucht und miteinander verglichen. Laut [5] sind zwei Layouts bei einem Faceted Search System besonders üblich.

Bei dem Ersten ist das Hauptteil von dem System in zwei horizontale Abschnitte unterteilt. Dabei befindet sich links die Eingabeoberfläche, und Rechts mit etwas größeren Platz die gesuchte Information. Im ORKG ist es die State-of-the-Art Comparison.

Abbildung 4.2 zeigt, wie das beim ORKG aussehen könnte. Bei solchen Art von Layout ist es sehr Wahrscheinlich, dass der Nutzer dieses Faceted Search Tool findet und es in Betracht nimmt. Gleichzeitig wird der Nutzer vom hauptsächlich Ziel nicht abgelenkt. Nämlich die gesuchte Information zu finden. Da die Eingabeoberfläche und das Ergebnis auf der gleichen Höhe sind, lassen sich ausgewählte Regeln leicht modifizieren, ohne die Ergebnisse aus der Sicht zu verlieren.

Dieses Layout einigt sich am besten für Applikationen, bei denen sich die Ergebnisse vertikal platz nehmen, zum Beispiel e-Shops wie Amazon. Im ORKG nimmt sich die Comparison den platz nur horizontal. Da die Eingabeoberfläche ungefähr 1/3 von dem gesamten Breite nimmt, lassen sich, wie in der Abbildung 4.2 zu sehen ist, nur eine geringere Anzahl

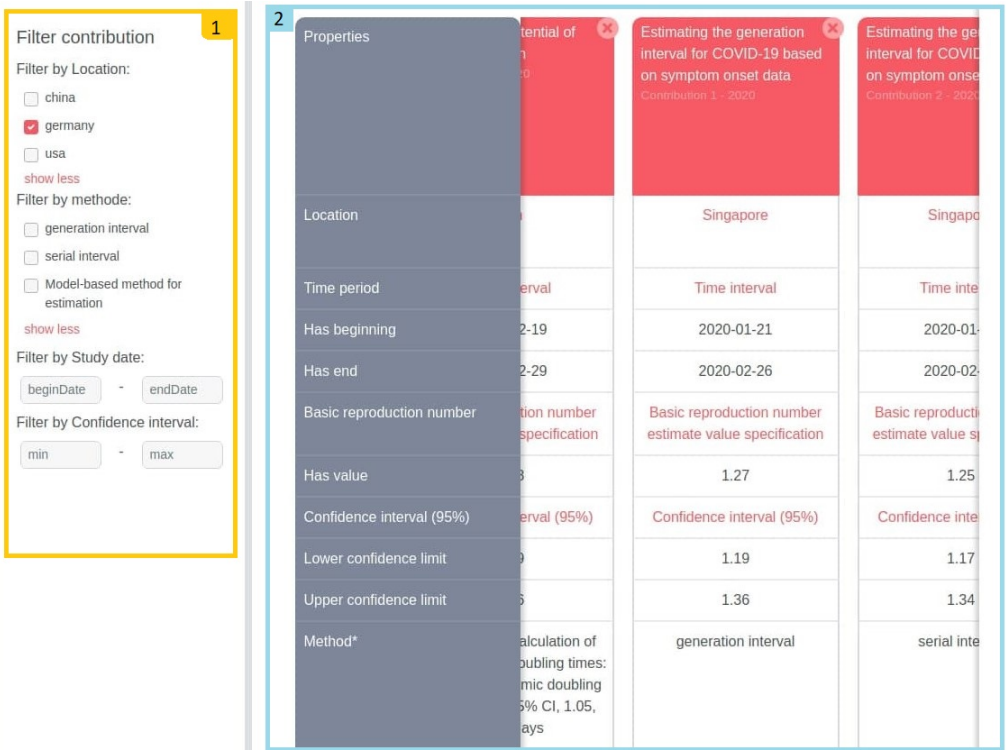

Abbildung 4.2: Leftsided UI des Faceted Search Systems in ORKG
1) Eingabeoberfläche des Systems. 2) Ergebnisse (State-of-the-Art Comparison)

an Kontributionen darstellen. Somit nutzt dieses Layout in dem Fall den Platz deutlich unpraktisch.

Bei dem Zweiten Layout wird die Eingabeoberfläche oberhalb der State-of-the-Art Comparison platziert. Abbildung 4.3 stellt das Layout innerhalb des ORKGs dar. Hier besteht im Vergleich zum ersten Layout den Vorteil, dass der platz geschickt ausgenutzt wird. Dieses ermöglicht eine größere Anzahl an Kontributionen gleichzeitig sichtbar zu halten.

Wie in der Abbildung 4.3 zu sehen ist, steht zwischen der Eingabeoberfläche und der State-of-the-Art Comparison einen sehr großen Abstand. Somit lassen sich Änderungen an den ausgewählten Kriterien sehr mühsam machen. Diese hat dennoch einen guten Vorteil, nämlich ist der Nutzer gezwungen die gewünschte Kriterien zu formulieren, bevor er die Ergebnisse überhaupt gesehen zu haben. Diese führt dazu, dass man produktiver arbeitet [5]. Jedoch sind solche komplexe Kriterien von nicht erfahrenen Nutzern nur schwer Ausdrückbar. Somit einigt sich dieses Layout hierfür nicht gut.

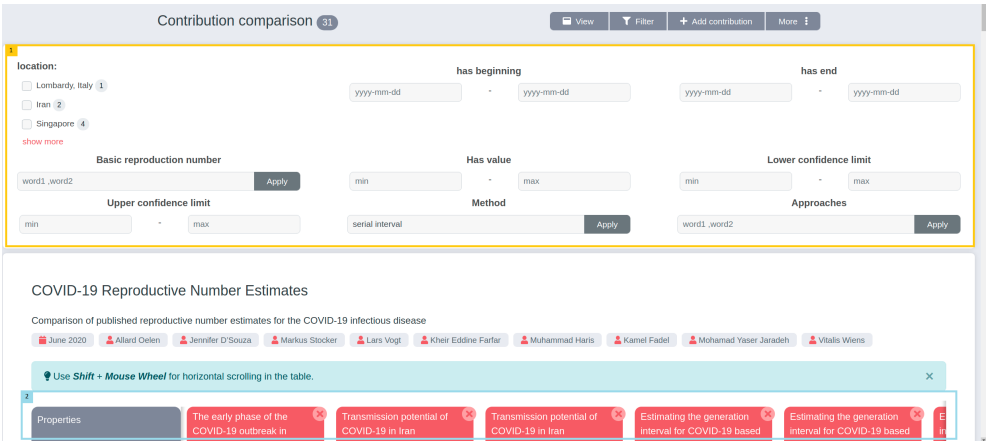

Abbildung 4.3: Vertical UI des Faceted Search Systems in ORKG
1) Eingabeoberfläche des Systems. 2) Ergebnisse (State-of-the-Art Comparison)

Schließlich wird hier das dritte Layout vorgestellt, Dieses wurde von dem bekannten Tabellenkalkulationsprogramm Excel[1] inspiriert. Hier wird im normalen Modus keine Eingabeoberfläche dargestellt. Stattdessen wird für jede Property eine dazugehörige Taste implementiert. Betätigt der Nutzer eins dieser Tasten, so öffnet sich ein Dialogbox mit der zum ausgewählten Property zugehörigen Eingabeoberfläche. In der Abbildung 4.4 lässt sich dieser Prozess deutlich nachvollziehen.

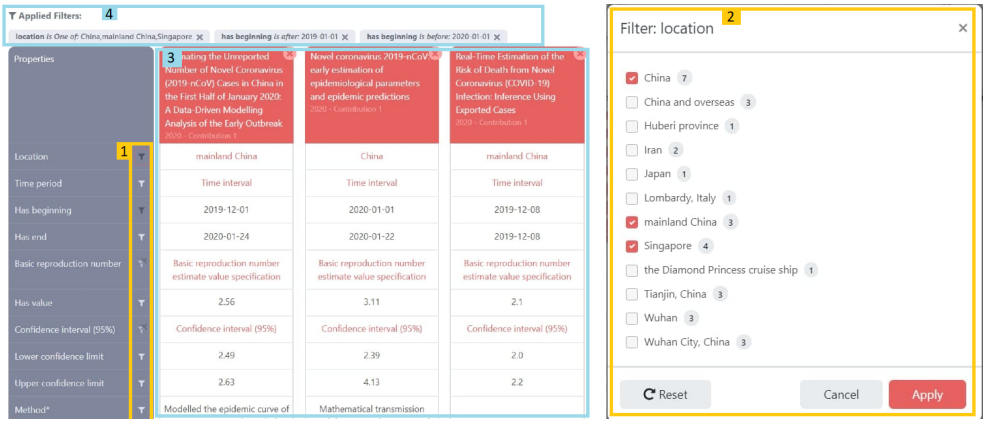

Abbildung 4.4: Internal UI des Faceted Search Systems in ORKG
1) Taste des jeweiligen Property. 2) Eingabeoberfläche des Systems.
3) Ergebnisse (State-of-the-Art Comparison) 4) Ausgewählte Kriterien

Im Vergleich zu den Abbildungen 4.2 und 4.3 wird hier der Platz deutlich effektiver ausgenutzt. So können eine größere Anzahl an Kontributionen gleichzeitig angezeigt werden, und somit wird der Nutzer von dem Ziel, die gesuchte Information zu finden, nicht abgelenkt. Jedoch

[1] https://de.wikipedia.org/wiki/Microsoft_Excel

sind zwei neue Probleme hiermit aufgetreten.

Zum Einen ist es möglich, dass Nutzern die Existenz von dem Faceted Search System überhaupt nicht wahrnehmen. Hierfür soll es auf das System hingewiesen werden. Dabei wird auf die Affordanz von den Tasten eingesetzt. Haben diese die Form eines Trichters, so wird es den Nutzern klarer, dass es sich hierbei um eine Filterungsmöglichkeit handelt.

Zum Anderen entsteht das Problem, nämlich nachdem der Nutzer das Dialogbox schließt und die Ergebnisse anschaut, ist es nicht mehr Erkennbar, in welchem Zustand sich der Nutzer befindet. d.h. der Nutzer kann nicht nachlesen welche Kriterien er eingegeben hat. Um dieses Problem zu beheben, wird ein Bereich mit allen ausgewählten Kriterien oberhalb der State-of-the-Art Comparison integriert.

Außerdem hat dieses Layout den Vorteil, dass Nutzern dazu gezwungen ist, die Properties einzeln zu begrenzen. Und somit befolgt man das divide and conquer Prinzip. Hiermit ist es deutlich einfacher der gewünschte Informationsbedarf in komplexe Kriterien zu umwandeln, da das Problem in mehrere kleinere Teilprobleme zerlegt werden, die einfach zu lösen sind.

Da dieses Layout 4.4 im Vergleich zum Layout 4.2 und 4.3 sich für das ORKG am besten einigt, wurde die Entscheidung getroffen, dieses anzuwenden.

4.2 Datentypen Erkennung

Ein Faceted Search System soll auf die verschiedene Datentypen adoptiert werden, so lassen sich nur bestimmte Operatoren auf einem Datentyp anzuwenden, beispielsweise können zwei Forschungsstandorte nur auf die Gleichheit verglichen werden.

Während die Papers in dem ORKG strukturiert sind, und somit von Maschinen Interpretierbar sind, was das Ziel von dem System ist, haben sowohl die Properties als auch die Atrributen eines Papers in dem ORKG keinen eindeutigen Datentyp. Nämlich sind alle diese Verschiedene Daten als Strings gespeichert.

Im Vergleich zu einem statischen Faceted Search System, wo die Struktur von Verschiedene Objekte klar sind. Zum Beispiel bei einem e-Shop haben alle Produkte die gleichen Attributen, diese sind Preis, Bewertung etc..
Sind Wissensgraphen von Natur aus heterogen. So haben unterschiedliche Comparisons unteschiedliche Struktur. Dafür muss ein dynamischen

Faceted Search System zum Einsatz kommen.

Bei jeder Comparison werden die Werte einer Property von allen Kontributionen gesammelt. Diese Daten werden durch Testsalgorithmen laufen. Dabei wird die Property den Datentyp mit den meisten Treffers zugewiesen. In dieser Arbeit werden zwei unterschiedliche Datentypen betrachtet, die jeweils in zwei Unterklassen unterteilt sind. Diese sind :

- Categorical Data unterteilt in Purely Categorical Data und Richtext Data
- Ordinal Data unterteilt in Numerical Data und Dates.

Datentyp	**Erlaubte Operatoren**
Purely Categorical Data	== , !=
Richtext Data	== , != , includes
Numerical Data	==, != , >, <
Dates	==, != , >, <

Tabelle 4.1: Datentypen mit jeweils den erlaubten Operatoren.

Tabelle 4.1 zeigt die in dieser Arbeit betrachtenden Datentypen mit den erlaubten Operatoren. In den nächsten Unterabschnitten wird auf die Verschiedene Datentypen detaillierter eingegangen.

4.2.1 Categorical Data

Unter Categorical Data versteht man textuelle Daten, die sich nur auf die Gleichheit überprüft werden können. D.h. zwei Instanzen dieser Daten sind entweder identisch oder nicht. Diese werden auf Purely Categorical Data und Richtext Data unterteilt. Mit Purely werden diese Daten auf solche Property begrenzt, Die sich wie Kategorien verhalten. Solche Daten beschreiben eine Eigenschaft von einer Kontribution. Beispielsweise porperties wie Land, Namen oder auch Ja/ Nein Property gehören hierzu. Um diese zu erkennen wird es angenommen, dass Dateninstanzen aus diesem Typ meistens weniger als fünf Wörter sind. Dieses Verfahren gilt als heuristisches Algorithmus, jedoch funktioniert diese für das ORKG gut.

Für diesen Art an Properties werden dem Nutzer die verschiedene Werte aller Kontributionen als Facetten Zur Auswahl dargestellt.

Unter Richtext Data befinden sich hier auch textuelle Daten. Jedoch verhalten sich diese nicht wie Kategorien. Sondern eher als Beschreibungen oder Erklärungen. Hierfür wird es geraten, dass diese sich aus langen

Sätze gebildet sind. D.h. Instanzen dieser Daten sind meistens mehr als fünf Wörter.

Da ein Ziel von dem Faceted Search System ist, die Zeitaufwand zur Findung der Information zu minimieren. Ist es Wichtig, dass die Interaktion mit dem System einfach und schnell beleibt. Somit ist es Unpraktisch, die verschiedene Werte hier als Facetten Zur Auswahl darzustellen. Im Vergleich zum Purely Categorical Data sind hier die Instanzen deutlich länger und somit mühsamer zum Lesen bzw. zum Scannen. Daher einigt sich hier ein textuelles Eingabefeld. Somit können durch Eingabe von kurzen Wörtern ein bestimmtes Kriterium ausgewählt werden.

4.2.2 Ordinal Data

Darunter versteht man Daten, die einsortiert werden können. D.h. neben die Möglichkeit mehrere Dateninstanzen auf die Gleichheit zu überprüfen. lassen sich unter dieser Klasse auch Operationen wie „größer als“ und „kleiner als“ ausführen. Hierfür werden die Daten auf zwei Klassen unterteilt: Numerical data und Dates.

Für der Vergleich von Daten dieser Klasse, ist es Vorausgesetzt, dass die Operanden den identischen Datentyp haben. Z.b kann ein Datum mit einer Zahl nicht verglichen werden. Daher soll jeder Eingabe vom Nutzer validiert werden. Um fehlerhafte Abfragen zu eliminieren.

Um den Nutzer nicht zu überfordern, werden hier nur drei Operatoren zur Verfügung gestellt, Mit denen komplexe Kriterien formulierbar sind. Diese Sind: „größer als oder gleich“ , „kleiner als oder gleich“ und „nicht gleich“ .

Dates verhalten sich sehr ähnlich wie Numerical Data. Jedoch werden diese Daten anders als bei Zahlen validiert. Z. b. müssen valide Datumseingaben das Format[2] „JJJJ-MM-TT“ haben. Daher sind diese Daten zu unterscheiden.

4.3 Vorbereitung der Daten

Eine Compariosn besteht aus Properties als Zeilen und Kontributionen als Spalten. Somit die Struktur einer $N \times M$ Matrix, mit N die Anzahl der Properties und M die Anzahl der Kontributionen. Bei Solchen Strukturen sind die Durchläufe sehr zeitaufwendig. Zum Beispiel ist die

[2]Das ORKG nutzt das Datumsformat der ISO Standard

Zeitkomplexität eines Durchlaufs hier $O(RNM)$ für R die Anzahl an ausgewählten Regeln.

Da das Faceted Search System auf dem Front-end arbeitet, muss der Ressourcen Verbrauch möglichst gering gehalten werden. Sonst können Nutzer mit begrenzten Rechenkapazität dieses Tool vermeiden.

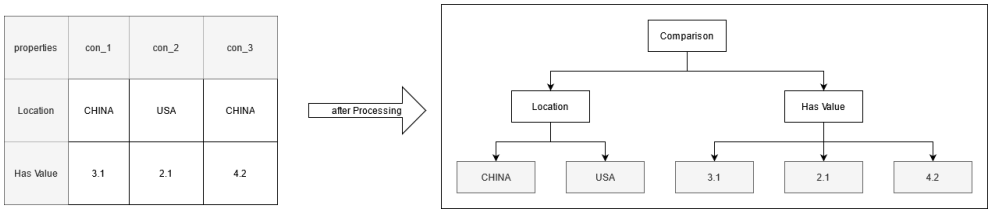

Abbildung 4.5: Unterschied zwischen den Datenstruktur bei einer Comparison. Bei der rechten Struktur werden die ContributionIDs in den Blätter gespeichert (hier grau gefärbt)

Um dieses Vorgang zu optimieren, wird diese Matrix zur Initialisierung der Comparison in eine Dictionary (deutsch Zuordnungstabelle) umgewandelt. Dabei wird die Matrix in kleinere Dictionarys unterteilt, Wo jede Dictionary nur die Informationen von einem Property enthält. Zudem sind Einträge von der Dictionary, eine Zuordnung von den verschiedenen Values zu den ContributionIDs, die diese Property Value hat. Abbildung 4.5 verdeutlicht den Unterschied Zwischen den Beiden Strukturen.

Während hier einen Vorteil gewonnen wird, Nämlich wird die Datenbearbeitung deutlich schneller. werden die Properties von einander getrennt. Diese führt dazu, dass die Regeln, die sich auf mehreren Properties beziehen, sich nur zur Laufzeit berechnen lassen. Da die Regeln Zugriff nur auf den dazugehörigen Property Dictionary hat.

4.4 Berechnung der Ergebnisse

Nachdem der Nutzer ein oder mehrere Kriterien eingegeben hat. Werden diese zur Bearbeitung an den Controller weitergeleitet. Mit Hilfe der initialisierten Dictionary wird jede Regel entsprechend ausgewertet. Dabei wird aus jede Regel die ContributionIDs berechnet, die diese Regel erfüllen. Tabelle 4.2 beschreibt, wie die ContributionIDs für jede Regel berechnet werden.

Rule type	**eine Contribution erfüllt diese Regel**
One of	falls der Wert von der Contribution einer von den ausgewählten Werte ist
Include	falls der Wert von der Contribution den ausgewählten Wert beinhaltet
greater than or equal	falls der Wert von der Contribution größer oder gleich den ausgewählten Wert ist
less than or equal	falls der Wert von der Contribution kleiner oder gleich den ausgewählten Wert ist
Not equal	falls der Wert von der Contribution nicht gleich den ausgewählten Wert ist

Tabelle 4.2: Regeln mit Ihrer Beschreibung

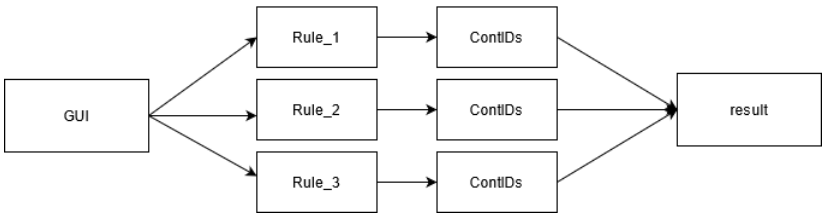

Abbildung 4.6: Berechnung der Ergebnisse via Map-reduce Prinzip

Da die Auswahl von Mehreren Regeln möglich ist. Werden gleich soviel ContributionIDs-listen erstellt. Daraus wird die Schnittmenge gebildet. Abbildung 4.6 zeigt wie dieses Vorgang läuft. Das Ergebnis wird an das GUI weitergeleitet. Schließlich werden die unerwünschten Kontributionen anhand der Ergebnisse eliminiert.

Kapitel 5

Ergebnisse und Diskussion

Durch das hier resultierte Faceted Search System wurde dem Nutzer die Möglichkeit gegeben, die Anzahl der Kontributionen effektiv zu reduzieren. Zum einen wird die Zeitbedarf von die Informationssuche deutlich minimiert. Zum Anderen kann eine Comparison, auf die Informationen, die aus Sicht der Forscher interessant sind, begrenzt werden. Diese führt dazu, dass wertvolle Einblicke und Zusammenhänge zwischen übrig gebliebene Kontributionen einfach zu extrahieren sind.
Mit der Verwendung der im Abschnitt 4.1 vorgestellten GUI, lassen sich komplexe aber auch präzise Kriterien formulieren. Dabei werden verschiedene Datentypen entsprechend behandelt, somit wird es gewährleistet, dass für jeden Datentyp die Formulieren eines Kriteriums intuitiv bleibt. Außerdem wird es gesichert, dass die Eingaben validiert sind, um fehlerhafte Ergebnisse zu vermeiden.
Es ist zu betrachten, dass Das Faceted Search System deutlich effektiv ist, wenn der Nutzer die gesuchte Informationen vorher kennt, d.h. Der Nutzer kennt sich mit ein oder mehreren Eigenschaften der gewünschten Kontributionen gut aus. Somit Wird dem Nutzer gelingen, die gesuchte Kontributionen durch Kriterien zu beschreiben, und im Endeffekt Zeit sparen. Diese ist aber nicht der Fall wenn die gesuchte Informationen für den Nutzer neu sind.

Außerdem ist das resultierende System Erweiterbar. Diese Eigenschaft ist bei dem ORKG besonders wichtig, Da dieses noch ein junges Projekt ist. Hierfür ist der Kern dieses Systems schon vorhanden. Somit lassen sich hier weitere Funktionalitäten einfach integrieren. Hierfür muss nur eine neue Regel hinzugefügt werden.

So ist es Möglich, einen neuen Datentyp zu definieren. um für diesen die Eingabemethode für den Nutzer zu erleichtern. Dabei sollen nur drei Methoden realisiert werden. Eine Methode Zur Erkennung von diesem

Datentyp. Eine möglichst intuitive Eingabeoberfläche, mit ihrer Hilfe die Erstellung von Regeln realisiert wird. Zuletzt muss eine Verfahren entwickelt werden, die diese Regeln auswertet und ContributionIDs daraus berechnet.

Beispielsweise können Forschungsstandorte als GeoNames betrachtet werden. Mit der Hilfe von GeoNames ist es dann möglich, semantische Regeln auszuführen. eine Kriterium könnte so Aussehen: Alle Kontributionen mit dem Forschungsgebiet in Europa. Dabei werden nicht nur Kontributionen, die explizit Europa als Forschungsgebiet haben, sondern auch andere, die deren Forschungsgebiete geographisch in Europa liegt. Hierfür soll zuerst der Datentyp erkannt werden. diese kann mit Hilfe eine externe GeoNames API[1] geschehen. Zunächst soll im GUI die neue Regeln erstellbar sind. Am Ende wird mit Hilfe der API ein Algorithmus zur Auswertung der neuen Regeln, entwickelt werden.

[1] `http://www.geonames.org/`

Kapitel 6

Fazit und Ausblick

In dieser Arbeit wird versucht, das Information Overload Problem bei Comparisons im ORKG zu lösen. Hierfür wird ein dynamische Faceted Search System entwickelt, das einem Wissenschaftler die Möglichkeit gibt, bei einer Comparison durch Auswahl von mehreren Kriterien auf die aus Sicht der Wissenschaftler interessanten Kontributionen zu begrenzen.

Somit kann sich der Nutzer auf die gewünschte Kontributionen fokussieren. Dabei lassen sich die Properties in zwei Klassen unterteilen: Categorical Data und Ordinal Data. Diese Klassifizierung wird gebraucht, um für jede Klasse sinnvolle Eingabenmethoden realisieren.

In Zukunft sollte das hier entwickelt Faceted Search System erweitert werden. Zum einen können dabei neue Datentypen in das System speziell behandelt werden. Wie schon im Kapitel 5 besprochen könnten zum Beispiel Geonames erkannt werden. Um für diese Daten eine neue Eingabemethode zur Verfügung zu stellen. Nämlich können neue sinnvolle Regeln wie „Forschungsstandort liegt in Europa“ realisiert werden.

Zum Anderen können auch Regeln betrachtet werden, die sich auf mehr als eine Property beziehen. Hierfür soll dem Nutzer die Möglichkeit gegeben, die Kontributionen anhand der Differenzen von mehreren Properties zu begrenzen. Beispielsweise kann ein Regel so Aussehen:
$study\_end - study\_begin = 40$ days
Außerdem könnte dieses System um neue Funktionalitäten erweitert werden, auch diese, die mit dem Filtern nicht zu tun haben. so können aus einer Property anhand die Kontributionenswerte, Statistische Angaben berechnet werden. Zum Beispiel können bei Numerische Daten solche werte wie der Mittelwert viele bedeutungsvolle Einblicke dem Forscher ermitteln.

Literaturverzeichnis

[1] L. Feddoul, S. Schindler, and F. Löffler. Automatic facet generation and selection over knowledge graphs. In M. Acosta, P. Cudré-Mauroux, M. Maleshkova, T. Pellegrini, H. Sack, and Y. Sure-Vetter, editors, *Semantic Systems. The Power of AI and Knowledge Graphs*, pages 310–325, Cham, 2019. Springer International Publishing.

[2] L. Feddoul, S. Schindler, and F. Löffler. Semantic relatedness as an inter-facet metric for facet selection over knowledge graphs. In P. Hitzler, S. Kirrane, O. Hartig, V. de Boer, M.-E. Vidal, M. Maleshkova, S. Schlobach, K. Hammar, N. Lasierra, S. Stadtmüller, K. Hose, and R. Verborgh, editors, *The Semantic Web: ESWC 2019 Satellite Events*, pages 47–51, Cham, 2019. Springer International Publishing.

[3] M. Y. Jaradeh, A. Oelen, K. E. Farfar, M. Prinz, J. D'Souza, G. Kismihók, M. Stocker, and S. Auer. Open research knowledge graph: Next generation infrastructure for semantic scholarly knowledge. In *Proceedings of the 10th International Conference on Knowledge Capture*, K-CAP '19, page 243–246, New York, NY, USA, 2019. Association for Computing Machinery.

[4] M. Y. Jaradeh, A. Oelen, K. E. Farfar, M. Prinz, J. D'Souza, G. Kismihók, M. Stocker, and S. Auer. Open research knowledge graph: Next generation infrastructure for semantic scholarly knowledge, 2019.

[5] D. Tunkelang. Faceted search. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 1(1):1–80, 2009.

[6] B. Wei, J. Liu, Q. Zheng, W. Zhang, X. Fu, and B. Feng. A survey of faceted search. *Journal of Web Engineering*, 12:41–64, 02 2013.