

AUTOMATISCHE GENERIERUNG
KOMPLEXER FRAGEN ZUM
INFORMATIONSMANAGEMENT AUF DER
BASIS DER SNIK-ONTOLOGIE

Der Fakultät für Mathematik und Informatik
der Universität Leipzig eingereichte

BACHELORARBEIT

zur Erlangung des akademischen Grades

Bachelor of Science
(B. Sc.)

im Fachgebiet Informatik vorgelegt von

Arne Roszeitis

Leipzig, den 25.01.2022

AUTOR:

Arne Roszeitis

TITEL:

*Automatische Generierung komplexer Fragen zum Informationsmanagement
auf der Basis der SNIK-Ontologie*

INSTITUT:

Institut für Medizinische Informatik, Statistik und Epidemiologie

Medizinische Fakultät

Universität Leipzig

BETREUER:

Prof. Dr. Alfred Winter

Dr. Konrad Höffner

DANKSAGUNG

Ich möchte an dieser Stelle bei einigen für diese Arbeit und für mich unabdingbaren Menschen bedanken.

Der erste und größte Dank geht an meine Eltern und meine Freunde für die generelle Unterstützung und die Unterstützung in meinem Studium im Speziellen, die sie mir zuteil werden ließen. Namentlich hier ein großer Dank an Viktor, Maria, Robin, Yannik, Ferdinand, Paulina, Mona, Deniz und Niel.

Desweiteren möchte ich mich noch bei meinem Betreuer, Dr. Konrad Höffner, für die großartige Betreuung bedanken. sowie Prof. Winter, Dr. Jahn und Dr. Schneider für ihre Unterstützung bei dieser Bachelorarbeit. Ein weiterer Dank geht an Prof. Bogdan, Dr. Thomas Schmid und Max Braungardt für die Chance, als SHK wichtige Erfahrungen sammeln zu können.

Außerdem bedanke ich mich bei Stefan Neumeier, der mir die Mathematik verständlich auseinandernehmen konnte.

Alle anderen, nicht namentlich genannten Menschen, denen ich hier nicht gerecht werden konnte, sei versichert, dass ich trotzdem beim Schreiben dieser Danksagung an sie gedacht habe.

INHALTSVERZEICHNIS

Abbildungsverzeichnis	vii
Tabellenverzeichnis	vii
Algorithmen	viii
1 Einleitung	1
1.1 Gegenstand	1
1.2 Problemstellung	2
1.3 Motivation	3
1.4 Zielsetzung	3
1.5 Aufgabenstellung	4
1.6 Aufbau der Arbeit	4
2 Grundlagen	5
2.1 Graphentheorie	5
2.1.1 Graph	5
2.1.2 Weg	5
2.1.3 Pfad	5
2.2 Multiple Choice Quiz	5
2.3 Semantic Web	6
2.3.1 Ontologie	6
2.3.2 Open- vs. Closed-World-Assumption	7
2.3.3 URI, URL und IRI	7
2.3.4 Auftretende Probleme	7
2.3.5 Resource Description Framework	8
2.3.6 Resource Description Framework Schema	9
2.3.7 Linked Open Data	9
2.4 Implementierung	10
2.4.1 SPARQL	10
2.4.2 Python	12
3 Stand der Forschung	15
3.1 SNIK	15
3.1.1 Allgemein	15
3.1.2 SNIK-Quiz	16
3.1.3 Vorhandene komplexe Fragen	19
3.2 Fragegenerierung aus graphbasierten Daten	24
4 Lösungsansatz	25
4.1 Lösungsansatz zum Problem P1	25
4.2 Lösungsansatz zum Problem P2	25
4.3 Lösungsansatz zum Problem P3	25
5 Ausführung und Ergebnisse	27
5.1 Initiale Einschätzung der Strategien	27
5.2 Technische Herausforderungen	31
5.2.1 Begrenzung des Massenspeichers	31
5.2.2 Möglichkeiten der SPARQL-Abfragen	31

5.2.3	Verfügbarer RAM	33
5.2.4	Verfügbare Zeit	33
5.3	Neue Strategien	33
5.3.1	intertwined-Strategie	34
5.3.2	close-by-Strategie	41
5.3.3	occurence-Strategie	45
5.4	Evaluation	49
5.4.1	Aufbau der Evaluation	49
5.4.2	Evaluationsergebnis	50
6	Diskussion	53
6.1	Zur initialen Einschätzung	53
6.2	Erstellte Strategien und Evaluation	53
6.2.1	Bereits vorhandene Strategien	53
6.2.2	intertwined-Strategie	53
6.2.3	close-by-Strategie	53
6.2.4	occurence-Strategie	54
6.2.5	Evaluation	54
6.3	SNIK	54
6.3.1	Zur SNIK-Ontologie	54
6.3.2	Zum SNIK-Quiz	54
	Zusammenfassung	57
	Literatur	59
I	Appendix	
A	Evaluation	65
A.1	Fragebogen	65
A.2	Interview-Ergebnisse	67
A.2.1	Interviewpartnerin 1	67
A.2.2	Interviewpartnerin 2	68
A.2.3	Interviewpartnerin 3	69
B	SPARQL-Programme	71
C	Pythonprogramme	73

ABBILDUNGSVERZEICHNIS

Abbildung 2.1	Tripel in RDF	8
Abbildung 2.2	Stufen zu LOD	10
Abbildung 3.1	Ausschnitt SNIK-Graph	15
Abbildung 3.2	SNIK Metamodell	17
Abbildung 3.3	SNIK-Quiz	18
Abbildung 3.4	Quiz-Ergebnis	18
Abbildung 5.1	Plot für $f(x)$	28
Abbildung 5.2	Plot für $g(x)$	29
Abbildung 5.3	intertwined-Strukturen	34

TABELLENVERZEICHNIS

Tabelle 2.1	RDF-Syntax	9
Tabelle 3.1	SNIK-Klassen	16
Tabelle 3.2	Strategien	19
Tabelle 3.3	Fragetypen Winter u. a., 2011	23
Tabelle 3.4	Fragetypen Ammenwerth u. a., 2014	23
Tabelle 5.1	Strategiebewertung	30
Tabelle 5.2	Benchmarkergebnisse	32
Tabelle 5.3	Genutzte Präfixe	34
Tabelle 5.4	intertwined-Strategie	35
Tabelle 5.6	close-by-Strategie	41
Tabelle 5.5	intertwined-Ergebnisbeispiele	42
Tabelle 5.7	close-by-Ergebnisbeispiele	45
Tabelle 5.8	occurence-Strategie	46
Tabelle 5.9	occurence-Ergebnisbeispiele	49
Tabelle 5.10	Komplexität Strategien	50
Tabelle 6.1	Vorschläge SNIK-Quiz	55

ALGORITHMEN

Alg. 2.1	Praktisches Beispiel 1	11
Alg. 2.2	Praktisches Beispiel 2	11
Alg. 2.3	Praktisches Beispiel 3	12
Alg. 2.4	Pythonbeispiel	13
Alg. 5.1	Benchmarking-Query	31
Alg. 5.2	Grundlegende Idee der intertwined-Strategie .	36
Alg. 5.3	<i>intertwined</i> -Strategie	36
Alg. 5.4	Python intertwined Programmcode	39
Alg. 5.5	Disktraktorengenerierung der <i>close-by</i> -Strategie	43
Alg. 5.6	<i>occurrence</i> -Strategie: Taktisches IM	46
Alg. 5.7	Python <i>occurrence</i> Programmcode	46
Alg. B.1	<i>occurrence</i> -Strategie: Begriffe, welche nur im Strategischen Informationsmanagement (IM) auftreten	71
Alg. B.2	<i>occurrence</i> -Strategie: Begriffe, welche sowohl im Strategischen IM als auch im Taktischen IM auftreten	71
Alg. B.3	<i>occurrence</i> -Strategie: Begriffe, welche weder im Strategischen IM noch im Taktischen IM auftreten	71
Alg. C.1	<i>close-by</i> -Einfügen in Fragetext	73
Alg. C.2	Python Quizsolver Programmcode	74

AKRONYME

ASCII	American Standard Code for Information Interchange
BELL	Besondere Lernleistung
CIO	Chief Information Officer
DFG	Deutsche Forschungsgemeinschaft
FTP	File Transfer Protocol
GMDS	Deutsche Gesellschaft für Medizinische Informatik, Biometrie und Epidemiologie
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IM	Informationsmanagement
IMISE	Institut für Medizinische Informatik, Statistik und Epidemiologie
IRI	Internationalized Resource Identifiers
JSON	JavaScript Object Notation
LOD	Linked Open Data
OWL	Web Ontology Language
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
SNIK	Semantisches Netz des integrierten Informationsmanagements im Krankenhaus
SPARQL	SPARQL Protocol And RDF Query Language
SQL	Structured Query Language
Turtle	Terse RDF Triple Language
UCS	Universal Character Set
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WWW	World Wide Web

EINLEITUNG

1.1 GEGENSTAND

Mit der zunehmenden Digitalisierung des alltäglichen Lebens wird es auch in der Lehre wichtiger, sich anzupassen und mittels digitaler Angebote Möglichkeiten zur besseren Bildung der Studenten anzubieten.

In der Medizinischen Informatik als „Wissenschaft der systematischen [...] Darstellung, [...] Aufbewahrung [...] und Bereitstellung von [...] Informationen und Wissen in der Medizin und im Gesundheitswesen“ (GMDS, 2021)¹ ist es umso relevanter, da die Medizinische Informatik stark von der verfügbaren digitalen Infrastruktur abhängt.

Dabei können auch simulierte Prüfungssituationen mit realistischen Testfragen dem Verständnis des Medizinischen Informationsmanagements und dem Lernerfolg der Studierenden zuträglich sein.

Die Medizinische Informatik wird in der Universität Leipzig durch das Institut für Medizinische Informatik, Statistik und Epidemiologie (IMISE)² repräsentiert.

Eines der dort mitbearbeiteten Projekte, das SNIK-Projekt³, ist das des Semantischen Netzes des Krankenhausinformationsmanagements (Jahn u. a., 2014). Dort werden Informationen vernetzt und so Wissen modelliert. Diese vernetzten Informationen sind dann sowohl menschen- als auch maschinenlesbar.

Dieses Semantische Netz wird unter Anderem zur Verbesserung der Lehre eingesetzt, indem in Vorlesungen Sachverhalte im Krankenhausinformationsmanagement mithilfe eines aus dem Netz erstellten Quizzes abgefragt werden. Dieses Unterprojekt im SNIK-Projekt wird als SNIK-Quiz bezeichnet. Damit werden Fragen aus der SNIK-Ontologie, welche in einer graphenbasierten Datenbank gespeichert ist, erstellt. Dabei werden sowohl richtige als auch falsche Antworten generiert, um so Multiple-Choice-Fragen zu erzeugen.

Zur Erzeugung dieser Fragen werden Strategien⁴ verwendet, mit denen verschiedene Arten von Fragen erzeugt werden können. Diese sind in Datenbank-Abfragen implementiert⁵.

¹ Deutsche Gesellschaft für Medizinische Informatik, Biometrie und Epidemiologie (GMDS)

² <https://www.imise.uni-leipzig.de/Institut>, 27.06.2021

³ Ein Deutsche Forschungsgemeinschaft (DFG)-Projekt, 2014-17, Projektförderkennzeichen 1605/7-1 und 1387/8-1

⁴ Methode, um aus der Ontologie Fragen anhand eines definierten Schemas abzuleiten

⁵ Eine Implementierte Strategie ist ein nicht zwangsweise notwendiges Programm und eine notwendige SPARQL-Abfrage, welche das Ziel einer Strategie durch eine Ausgabe erfüllen

Außerdem gibt es noch andere Unterprojekte wie den SNIK-Graph, der die Ontologie visualisiert.

1.2 PROBLEMSTELLUNG

Sowohl in der Präsenzlehre als auch im E-Learning profitieren sowohl Lehrende als auch Lernende von Lerntransferleistungen⁶, da so Lehrinhalte durch Vorwissen effizienter vermittelt werden können und auch die Prüfungsleistungen mit besseren Ergebnissen abgeschlossen werden können (vgl. Hirtl, 2010), da das Lernen effizienter vonstattengeht und so mehr Zeit für individuell schwierige Themengebiete bleibt.

Die gegenwärtig existierenden Strategien zur Frageerzeugung sind nur bedingt pädagogisch wertvoll, da sie keine Lerntransferleistung beinhalten, sondern nur auf das Abfragen von Definitionen ausgerichtet sind. Außerdem ist bei drei verschiedenen vorhandenen Strategien (Wächtler, 2020) die existierende Variation von Aufgaben und Problemstellungen sehr gering, obwohl die Variation eine gute Möglichkeit zur Transferförderung ist, vgl. Niegemann u. a., 2008.

Es wären also neue geeignete Strategien hilfreich, die eine höhere Lerntransferleistung erfordern, um die daraus resultierenden Aufgaben erfolgreich lösen zu können. Diese könnten auch das Erkennen von Parallelen zwischen dem taktischen Informationsmanagement (Ammenwerth u. a., 2014) und dem strategischen Informationsmanagement (Winter u. a., 2011) fördern. Das ist zum Verständnis und Lerntransfer hilfreich und ermöglicht so Studierenden, die bereits eines dieser Bücher bearbeitet hätten, einen leichteren Einstieg in das jeweils andere Buch.

Dazu wäre es notwendig zu erarbeiten, welche Beziehungen in der Ontologie zielführend eingesetzt werden könnten, wie diese mittels Algorithmen erschlossen und dann in natürlichsprachliche Konstrukte überführt werden könnten. Zusätzlich können mit der Auswahl der geeigneten Strategien auch verschiedene Lernziele besser in den Fokus genommen werden. Durch die Ermittlung von Kriterien zur Einschätzung einer Strategie können die Prüfenden und Lernenden selbst den Fokus setzen sowie die Schwierigkeit der Aufgaben bestimmen, indem sie eine beliebige Anzahl von Fragen aus den verschiedenen Fragentypen gezielt auswählen. Dabei haben sie die Möglichkeit, sich an den gewünschten Kriterien der Strategien zu orientieren.

Um am Ende die erarbeiteten Ergebnisse unabhängig einschätzen zu können, ist es erforderlich, dass nach der Erarbeitung der neuen Fragestrategien eine Gruppe von geeigneten Personen zu den generierten Fragen Stellung bezieht und diese bewertet, vor Allem bezüglich Verständlichkeit und Sinnhaftigkeit der Fragen. Dazu wird es notwendig

⁶ Lerntransfer: kreative Leistung, in einem Kontext erlerntes Wissen auf einen unbekanntem Kontext anwenden zu können

sein, diesen Experten Fragen zu den generierten Fragen zu stellen und die Antworten auszuwerten.

Abschließend soll noch kurz zusammengefasst werden, welche Probleme im Rahmen dieser Bachelorarbeit gelöst werden sollen:

- P₁ Keine Kriterien zur initialen Einschätzung von Strategien
- P₂ Die Komplexität der gestellten Aufgaben ist zu gering
- P₃ Eine unabhängige Einschätzung zur Verständlichkeit und Sinnhaftigkeit der Aufgaben ist erforderlich, um gute Chancen auf Lernerfolg für Studierende zu gewährleisten

1.3 MOTIVATION

E-Learning erhält seit langer Zeit (vgl. Fischer u. a., 2003) immer mehr Bedeutung, insbesondere seit dem Ausbruch von COVID-19 (vgl. Dittler und Kreidl, 2021). Daher haben Online-Wissensquellen beim Lernen eine höhere Bedeutung. Wenn diese Quellen dann auch noch ein Werkzeug bieten, um nachhaltig lernen zu können, das Wissen also zu behalten und einsetzen zu können, steigert das den Wert dieses Werkzeugs für die digitale Bildung ungemein.

Ein solches Werkzeug stellt das SNIK-Quiz dar, welches aus der SNIK-Ontologie Multiple-Choice-Fragen erstellen und so den Wissensstand prüfen und festigen kann.

Es ist zwar prinzipiell möglich, mit denen in der BELL beschriebenen Algorithmen eine komplette Multiple-Choice-Prüfung zu konstruieren, allerdings sind dies immer noch Aufgaben ohne Lerntransferleistung, die bis jetzt reine Definitionen abfragen. Für einfache Prüfungsaufgaben ist dies ausreichend, allerdings sollten Prüfungen auch Aufgaben beinhalten, die einen Lerntransfer, also die Anwendung des Gelernten beinhalten. Von neuen Strategien für Prüfungsfragen profitieren somit die Lehrenden durch weniger Arbeitsaufwand bei der Prüfungsvorbereitung und die Studierenden durch bessere Lernmöglichkeiten vor der Prüfung .

1.4 ZIELSETZUNG

Zu jedem Problem P_i ist folgendes Ziel Z_i zum Lösen des jeweiligen Problems angedacht:

- Ziel Z₁ Entwicklung von Kriterien zur initialen Einschätzung
- Ziel Z₂ Entwicklung von Strategien zur Generierung komplexer Fragen
- Ziel Z₃ Evaluierung der Ergebnisse durch eine Expertengruppe hinsichtlich Lesbarkeit und praktischer Anwendbarkeit

1.5 AUFGABENSTELLUNG

Zu den jeweiligen Zielen Z_i sind folgende Aufgaben A_i mit den jeweiligen Unteraufgaben zu lösen:

- Aufgabe zu Z_1 :
 - Aufgabe A_1 : Entwickle Kriterien zur initialen Einschätzung
- Aufgaben zu Z_2 :
 - Aufgabe $A_{2.1}$: Erstellen neuer Strategien
 - Aufgabe $A_{2.2}$: Erarbeiten der Algorithmen zur Implementierung
 - Aufgabe $A_{2.3}$: Implementieren der Strategien sowie falls Ergebnisse der Query noch nicht natürlichsprachlich sind, Übersetzen der Ergebnisse in natürlichsprachliche Sätze
- Aufgaben zu Z_3 :
 - Aufgabe $A_{3.1}$: Erstellen der Evaluation
 - Aufgabe $A_{3.2}$: Ausführen der Evaluation
 - Aufgabe $A_{3.3}$: Auswerten der Evaluation

1.6 AUFBAU DER ARBEIT

In Kapitel 1 wird der Leser an das Thema der Arbeit herangeführt, die Probleme werden erörtert und die Ziele und Aufgaben, die hier gelöst werden sollen, vorgestellt. Danach folgen die Grundlagen in Kapitel 2 mitsamt der Begriffserklärung.

Anschließend kommt in Kapitel 3 der Stand der Forschung und in Kapitel 4 der Lösungsansatz zur komplexen Fragegenerierung, gefolgt von Kapitel 5 mit der Durchführung und den Ergebnissen. In Kapitel 6 werden die Ergebnisse dann kritisch betrachtet und ein Ausblick gegeben.

GRUNDLAGEN

2.1 GRAPHENTHEORIE

Der gesamte Unterabschnitt bezieht sich auf Meinel und Mundhenk, 2015.

2.1.1 *Graph*

Ein (gerichteter) *Graph* G ist ein geordnetes Paar $G = (V, E)$. V stellt dabei die Menge der *Knoten* und E die Menge der *Kanten* dar, wobei gilt, dass $E \subseteq V^2$.

2.1.2 *Weg*

Ein *Weg* w in einem Graphen $G = (V, E)$ ist eine Abfolge von Knoten (v_0, \dots, v_n) , sodass $\forall i \in \{0, \dots, n-1\} : (v_i, v_{i+1}) \in E$.

2.1.3 *Pfad*

Ein *Pfad* p ist ein *Weg* w , bei dem kein Knotenpaar mehr als einmal in w enthalten ist.

2.2 MULTIPLE CHOICE QUIZ

Man kann grundsätzlich zwei verschiedene Arten von Fragen unterscheiden: offene und geschlossene Fragen (Ellinger, 2017). Diese können auch gemischt auftreten. Offene Fragen haben keine vorgegebenen Antwortmöglichkeiten und sind selten vorzufinden. Da in den unter Anderem in Abschnitt 3.2 referenzierten Quellen nur die geschlossenen Fragen behandelt werden, soll es hier auch ausschließlich darum gehen.

Eine geschlossene Frage ist eine Frage, auf die es eine Anzahl von möglichen, vorgegebenen Antworten gibt. Von diesen können je nach Quiz-Art eine oder mehrere richtig sein. Die falschen Antwortmöglichkeiten bezeichnet man als *Distraktoren*.

Ein Multiple-Choice-Quiz ist also eine Menge von geschlossenen Fragen. Von diesen müssen dann eine bestimmte Anzahl richtig beantwortet werden, Jeremias und Rabe, 2021 empfehlen hier, die Wahrscheinlichkeit, nur durch raten zu bestehen, auf unter 3% zu beschränken. Empfehlungen für komplexe Fragen aus Jeremias und Rabe, 2021

sind, dass die Fragen selbst dem normalen Lösungsweg folgen und einen Anwendungsbezug aufweisen sollten.

Alle Fragen sollten außerdem keine reinen Fakten abfragen, sondern mittels einer Verbindung zur praktischen Anwendung einen speziellen Fokus bedienen.

Auch wichtig ist nach der Quelle, dass Fragen auch ohne die Antwortmöglichkeiten beantwortbar sind. Fragen sollten nach Möglichkeit ausführlich und Antwortmöglichkeiten kurz sein.

2.3 SEMANTIC WEB

Das Semantic Web soll Daten des Internets maschinen- und gleichzeitig menschenlesbar machen. Es existiert somit neben dem „Datei-Web“ als „Daten-Web“.

In diesem sind Daten in einer Form gespeichert, wie man sie auch in verschiedenen Datenbanken findet. Es baut auf verschiedenen, bereits vorhandenen Technologien, Standards und Protokollen wie zum Beispiel Uniform Resource Identifier (URI), Resource Description Framework Schema (RDFS), SPARQL und JavaScript Object Notation (JSON). Im Gegensatz zum World Wide Web (WWW), welches in Dokumenten strukturiert ist, ist das Semantic Web in Ressourcen organisiert.

2.3.1 *Ontologie*

Eine Ontologie ist in der Informatik „eine Art [formales] Wissensobjekt“ (Guarino, Oberle und Staab, 2009). Dies bedeutet, dass innerhalb der Ontologie Informationen in einer festgelegten Form miteinander in Beziehung gesetzt werden, um so Wissen zu repräsentieren. Dieses kann sowohl durch Maschinen als auch durch Menschen gelesen werden, muss aber zumindest der Prädikatenlogik erster Stufe genügen, also nicht zwangsläufig menschenlesbar sein. Dies bedeutet, dass Semantische Netze gegenüber Ontologien eine Untermenge bilden, da jedes Semantische Netz anhand dieser Definition eine Ontologie, aber nicht umgekehrt ist (Jahn u. a., 2014).

Die Web Ontology Language (OWL) ist das Werkzeug, um die oben beschriebenen Ziele im Semantic Web umzusetzen und enthält alle dafür notwendigen Funktionen. Eine der wichtigsten Eigenschaften von OWL ist dabei, dass für Maschinen aus unter Zuhilfenahme von OWL aufgebauten Datensätze viele Schlüsse mittels Prädikatenlogik erster Stufe möglich sind, welche ohne OWL nicht oder nur sehr schwer möglich wären (McGuinness und Harmelen, 2009).

2.3.2 *Open- vs. Closed-World-Assumption*

Nach Allemang und Hendler, 2011 ist das Semantic Web auf der *Open World Assumption* aufgebaut, d.h., dass Fakten, die noch nicht in der Ontologie enthalten sind, auch also wahr angenommen werden müssen. Im Gegensatz dazu bezieht sich die *Closed World Assumption* auf den Grundsatz, dass für alles, was wahr ist, auch bekannt ist, dass es wahr ist.

Für diese Arbeit muss allerdings davon ausgegangen werden, dass das SNIK innerhalb der *Closed World Assumption* existiert und dass das Netz korrekt ist und damit alle Knoten und Kanten darin auch korrekt sind. Diese Annahme muss gemacht werden, da ansonsten Distraktoren (Abschnitt 2.2) nicht zwangsweise falsch sein müssten. Um dies kurz an einem Beispiel zu erläutern, sei $V = \{A, B, C, D\}$ und $E = \{(A, B), (B, C), (C, A)\}$. Die Frage sei jetzt, welcher Knoten keine Kante mit anderen besitzt („Welches Subjekt ist thematisch nicht passend?“). In der *Closed World Assumption* ist die Frage eindeutig beantwortbar. In der *Open World Assumption* allerdings nicht, da hier ja noch eine Kante (x, D) , $x \in \{A, B, C\}$ existieren könnte, die nur noch nicht in unserem Graphen existiert.

2.3.3 *URI, URL und IRI*

Ein URI ist eine Buchstaben- und Ziffernsequenz, mit der „abstrakte oder physikalische Ressourcen eindeutig beschrieben werden können“ (Berners-Lee, Fielding und Masinter, 2005). Dieselbe Funktion erfüllen auch IRIs. Allerdings sind die Sequenzen von URIs immer eine Teilmenge des US-ASCII-Zeichensatzes (Duerst und Suignard, 2005).

Dies führt zu Problemen für Personen, die nicht das lateinische Alphabet als Standard benutzen. Deswegen wurde der IRI eingeführt, um so den Menschen einen besseren Umgang mit Identifiern zu ermöglichen. IRIs werden im UCS abgebildet, auch Unicode genannt, welcher wesentlich mehr Zeichen als ASCII umfasst.

URLs sind spezielle URIs, die nicht nur Ressourcen identifizieren, sondern auch den Ort angeben, wo diese Ressource im WWW zu finden ist. Dabei gibt es Schlagworte wie HTTPS oder FTP, die angeben, über welches Protokoll auf diese Ressourcen zugegriffen werden soll (Berners-Lee, 1994).

2.3.4 *Auftretende Probleme*

Die vier Hauptprobleme, welche Conejo, Barros und Bertoa, 2018 in Abschnitt 3 beschreiben, können aufgrund des Arbeitsumfangs dieser Arbeit nicht gelöst werden. Diese sind:

1. Falsche Label innerhalb des Graphen ausschließen

2. wie in Abschnitt 2.3.2 beschrieben, noch nicht beinhaltete Informationen
3. „unerwünschte“ Ergebnisse, welche nur von Menschen erkannt werden können. Die notwendigen Tester existieren im Rahmen dieser Arbeit nicht
4. Spezialfälle, in denen z.B. Abkürzungen verwendet werden, können zur Verwirrung führen, wenn nur die Abkürzung und nicht die ausgeschriebene Variante oder umgekehrt bekannt ist

2.3.5 Resource Description Framework

2.3.5.1 Zugrundeliegender Gedanke

Das Resource Description Framework (RDF) ist „ein Rahmen, um Informationen im Netz darzustellen.“ (Graham und Carroll, 2014). Die Menge der Subjekt-Prädikat-Objekt-Tripel bildet dann in ihrer Gesamtheit einen gerichteten Graphen, um so die Ressourcen zu beschreiben. Der entstehende Graph wird auch Wissensbasis genannt. Die Ressourcen innerhalb des Graphen und auch die Graphen selbst werden mittels URIs beschrieben. RDF-Datasets organisieren dann die RDF-Graphen-Sammlungen, um so eine weitere Organisationsstufe hinzuzufügen. RDF wurde im Gegensatz zu vielen Standards im WWW auch dafür entwickelt, maschinenlesbar zu sein, sodass nicht nur Menschen Informationen aus dem Semantischen Netz beziehen können. Dafür können Daten in der eigenen Datenbank auch mit den Daten Dritter kombiniert werden, um so die Informationen besser in Kontext zu setzen (Abschnitt 2.3).

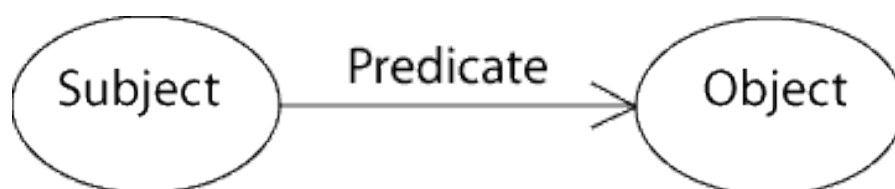


Abbildung 2.1: Subjekt-Prädikat-Objekt-Tripel
(<https://www.w3.org/TR/rdf-concepts/>)

Das zugrundeliegende Modell lässt sich gut mit einem natürlich-sprachlichen englischen Satz vergleichen: Es enthält Subjekt und Objekt, welche über das Prädikat verbunden werden. So existiert in SNIK zum Beispiel das Tripel „CIO - is responsible for - Management of Information Systems“, welches genau diesen Zusammenhang besitzt. Dabei werden häufig die gleichen Subjekte mehrmals benutzt, um so Einschränkungen zu definieren oder mehr Kontext und Informationen zu liefern. Die Syntax ist in Tabelle 2.1 enthalten.

2.3.5.2 *Syntax*

<RDF-Tripel>	::=	<Subjekt><Prädikat><Objekt>
<Subjekt>	::=	„RDF URI Referenz“ „Leerer Knoten“
<Prädikat>	::=	„RDF URI Referenz“
<Objekt>	::=	„RDF URI Referenz“ „Leerer Knoten“ <Literal>
<Literal>	::=	<Einfaches Literal> <Typisiertes Literal>
<Einfaches Literal>	::=	„Zeichenkette“ „Zeichenkette“ „Sprachkennzeichnung“
<Typisiertes Literal>	::=	„Zeichenkette“ „RDF URI Referenz“

Tabelle 2.1: Syntax von RDF, Notation angelehnt an die Backus-Naur-Form

Eine „RDF URI Referenz“ ist eine Zeichenkette ohne Kontrollcharaktere, welche eine valide URI ergeben würde.

Ein „Leerer Knoten“ (*blank node*) ist eine „Ressource, für die kein URI oder Literal gegeben ist“ (Hayes und Patel-Schneider, 2014).

Die „Sprachkennzeichnung“ beschreibt, in welcher Sprache der menschenlesbare Text gespeichert ist, z.B. „@en“ oder „@de“.

2.3.6 *Resource Description Framework Schema*

RDFS „ist eine Erweiterung des [...] RDF-Vokabulars“ (Brickley und Guha, 2014). Es stellt bestimmte Klassen und Eigenschaften bereit, welche es vereinfachen, Ressourcen zu beschreiben. Solche sind zum Beispiel `rdfs:class`, `rdfs:langString` als Klassen und `rdfs:range` oder `rdfs:subClassOf` als Prädikate (siehe Abbildung 2.1).

2.3.7 *Linked Open Data*

Linked Open Data (LOD) sind offene Daten im Web, welche verlinkt sind und dadurch sowohl menschen- als auch maschinenlesbar sind. Die vier geltenden Kriterien für *Linked Data* sind, dass URIs genutzt werden sollen, um Ressourcen zu benennen und diese durch Hypertext Transfer Protocol (HTTP) URIs zu finden sind. Außerdem müssen offene Standards eingesetzt werden, um Informationen über die Ressourcen zu erhalten und, wenn die Daten veröffentlicht werden, sollen sie andere Ressourcen ebenfalls per HTTP URIs benennen.

Nach einem Vorschlag von Tim Berners-Lee müssen diese Daten bestimmte, aufeinander aufbauende Bedingungen erfüllen, um sich LOD nennen zu können. Es solle es fünf Stufen geben, bei welchen als letzte LOD als Ergebnis des Prozesses steht. Die erste Stufe zu erreichen,

heißt, dass die Daten offen im Web sind, mit einer entsprechenden Lizenz. Die zweite Stufe ist dann, dass die Daten maschinenlesbar sind. Danach, für Stufe drei, müssen die Daten in zwei oder mehr nicht-proprietären Formaten vorliegen, für Stufe vier sollen diese Daten dann W₃C-Standards nutzen, z.B. RDF. Um die Stufe der LOD zu erreichen, muss eine Verlinkung mit den Daten anderer Anbieter von Daten der Stufe vier oder fünf erfolgen. Dies bezieht sich auf Berners-Lee, Fielding und Masinter, 2009.

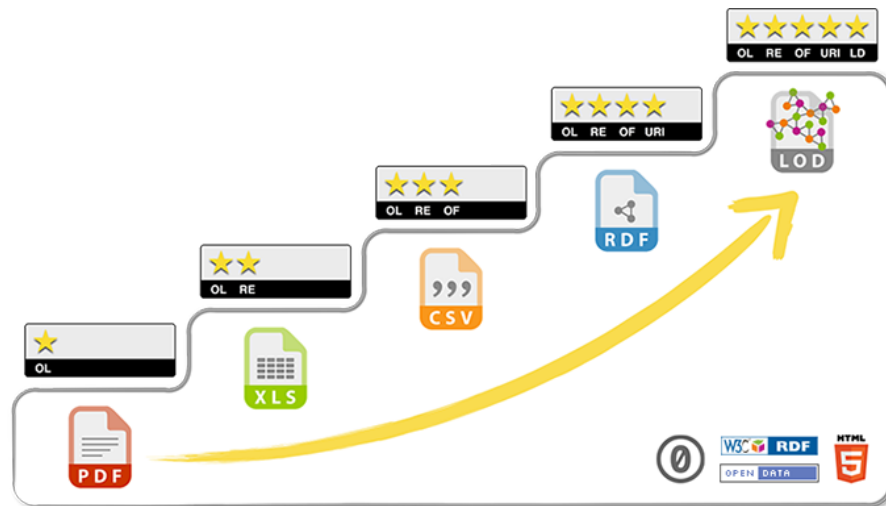


Abbildung 2.2: Stufen zu LOD

(<https://5stardata.info/images/5-star-steps.png>)

2.4 IMPLEMENTIERUNG

2.4.1 SPARQL

Die Hauptressource hierfür ist Harris und Seaborne, 2013, wobei es hier um die Abfragesprache gehen soll, nicht um das Protokoll über dem HTTP.

2.4.1.1 Hintergrund

Um in Semantischen Netzen Informationen zu erhalten, reicht es nicht, die Daten in einer Datenbank bereitzustellen. Dafür bedarf es auch noch der Datenbankabfragesprachen, um Informationen extrahieren zu können. Im Fall der RDF-graphbasierten Datenbanken, auf denen die Semantischen Netze aufbauen, ist diese Abfragesprache SPARQL. Diese wurde am 15. Januar 2008 von der W₃C offiziell als Empfehlung vorgestellt (Herman, 2008).

2.4.1.2 Grundlegende Syntax

Die in dieser Arbeit verwendete Syntax der Abfragen besteht, wie auch in SQL, aus dem Grundbaustein *SELECT* und *WHERE* sowie dem optionalen, aber häufig gebrauchten *FROM*, mit welchem zusätzliche Ressourcen definiert werden. Daneben gibt es weitere Schlagworte, welche bei Gebrauch erläutert werden.

SELECT definiert, welche Werte aus der Query zurückgegeben werden sollen. Diese Rückgabe erfolgt in Form einer Tabelle.

FROM definiert, aus welche(r/n) Datenbank(en) die gesuchten Daten extrahiert werden.

Die *WHERE*-Klausel dient zur Einschränkung der Suche, indem Bedingungen angegeben werden, um die Suche zu verfeinern.

Variablen werden mit einem „?“ begonnen, dahinter steht dann der Variablenname.

Dank RDF ist es möglich, bestimmte *Präfixe* zu definieren, um so Schreibaufwand zu sparen und die Optik und Verständlichkeit der Queries zu verbessern. Diese *Präfixe* bestehen aus dem Bezeichner, einer Abkürzung, welche dann auch die URI referenziert und der URI selbst. Das Ganze endet mit einem Punkt:

PREFIX *Abkürzung*: <URI>.

Die in dieser Arbeit verwendeten Präfixe sind in Tabelle 5.3 zu finden.

2.4.1.3 Praktisches Beispiel

Ein praktisches Beispiel zu einer einfachen Query für SNIK wäre:

```
SELECT (?q AS $term) (?ans AS $definition)
WHERE {
  ?class a owl:Class .
  ?class rdfs:label ?q .
  ?class skos:definition ?ans .
}
GROUP BY ?q
```

Alg. 2.1: Praktisches Beispiel 1

In dieser Query werden alle Klassen (*?class a owl:Class .*) gesucht, welche sowohl einen Bezeichner (*?class rdfs:label ?q .*) und eine Definition besitzen (*?class skos:definition ?ans .*). Dann werden sowohl das Label als auch die Definition in zwei Spalten geordnet (*SELECT SAMPLE(str(?q)) SAMPLE(str(?ans))*) an den Anfrager zurückgegeben.

Zu beachten ist hier, dass sowohl Ergebnisse in englischer wie auch in deutscher Sprache gemischt ausgegeben werden. Um also nur englische Ergebnisse zu erhalten, muss man die Daten filtern:

```
SELECT (?q AS $term) (?ans AS $definition)
WHERE {
  ?class a owl:Class .
  ?class rdfs:label ?q .
```

```

?class skos:definition ?ans .
FILTER (langMatches(LANG(?q), "en"))
FILTER (langMatches(LANG(?ans), "en"))
}
GROUP BY ?q

```

Alg. 2.2: Praktisches Beispiel 2

Teile der oben gezeigten Queries enthalten bereits eine besondere Syntax, genannt Terse RDF Triple Language (Turtle). Diese erlaubt es, für Menschen besser lesbare Queries zu erzeugen, so zum Beispiel durch die Nutzung der *Präfixe* oder durch *a* als Prädikat. Außerdem können bei gleichem Subjekt die Queries einfacher gestaltet werden, indem aus

```

?a ?b ?c .
?a ?d ?e .

```

```

?a ?b ?c;
?d ?e .

```

wird.

Sollten sogar Subjekt und Prädikat gleich sein, so kann aus

```

?a ?b ?c .
?a ?b ?d .

```

```

?a ?b ?c,
?d .

```

werden (Beckett und Berners-Lee, 2011). Entsprechend kann man die ohnehin schon bessere Variante noch einmal vereinfachen:

```

SELECT (?q AS $term) (?ans AS $definition)
WHERE {
?class a owl:Class ;
      rdfs:label ?q ;
      skos:definition ?ans .
FILTER (langMatches(LANG(?q), "en"))
FILTER (langMatches(LANG(?ans), "en"))
}
GROUP BY ?q

```

Alg. 2.3: Praktisches Beispiel 3

Damit hat man nun eine gute lesbare, einfache Query zur Generierung aller englischen Terme mitsamt Definition.

2.4.2 Python

Die Skriptsprache Python entstand Ende der 1980er Jahre als Ergebnis der Arbeit von Guido von Rossum im Zusammenhang seiner Erfahrungen mit der Programmiersprache ABC (Venners, 2013).

Python legt seinen Fokus auf eine gute Lesbarkeit für Menschen, Erweiterbarkeit nicht nur vonseiten der Entwickler, sondern auch der Gemeinschaft (*python community*) sowie auf seinen Multi-Paradigmen-Aufbau, welcher im starken Zusammenhang mit der Erweiterbarkeit steht.

Python selbst folgt einer dynamischen Typisierung und eignet sich so im Zusammenhang mit viel syntaktischem Zucker (*syntactic sugar*¹) auch sehr gut für die Arbeit mit Zeichenketten (*Strings*).

Innerhalb dieser Arbeit soll vor Allem auf das manipulieren von Zeichen und Zeichenketten sowie die umfangreichen Funktionen zur Listenmanipulation Wert gelegt werden, da hier Python hauptsächlich dafür eingesetzt werden soll, mögliche nicht natürlichsprachliche Ergebnisse der SPARQL-Queries in natürliche Sprache zu überführen. In diesem Zusammenhang soll hier kurz anhand eines Pythonskripts auf günstige Syntax eingegangen werden:

```
# dies ist nur ein Teil des Skripts
def get_question_content(questions):
    for i in range(len(questions)):
        questions[i] = questions[i][8:]
    return questions
convert_strings(["What is Information Management", "What is IMISE"
    ↪ ])
```

Alg. 2.4: Pythonbeispiel

In der letzten Zeile erfolgt der Funktionsaufruf mit einer beispielhaften Liste als Übergabeparameter.

Diese Funktion (`get_question_content(...)`) erhält eine Liste von Strings (`questions`). Aus dieser Liste werden nun nacheinander (`for i in range(len(questions)):`) die einzelnen Fragen in String-Form ausgelesen. Die einzelnen Strings in der Liste werden dann so gekürzt, dass die ersten 8 Buchstaben abgeschnitten werden (`questions[i] = questions[i][8:]`), das Ende des Strings wird belassen. So können mit Python leicht Strings und Listen manipuliert werden.

¹ Vereinfachungen der Syntax zur besseren Lesbarkeit und Arbeit

STAND DER FORSCHUNG

3.1 SNIK

3.1.1 Allgemein

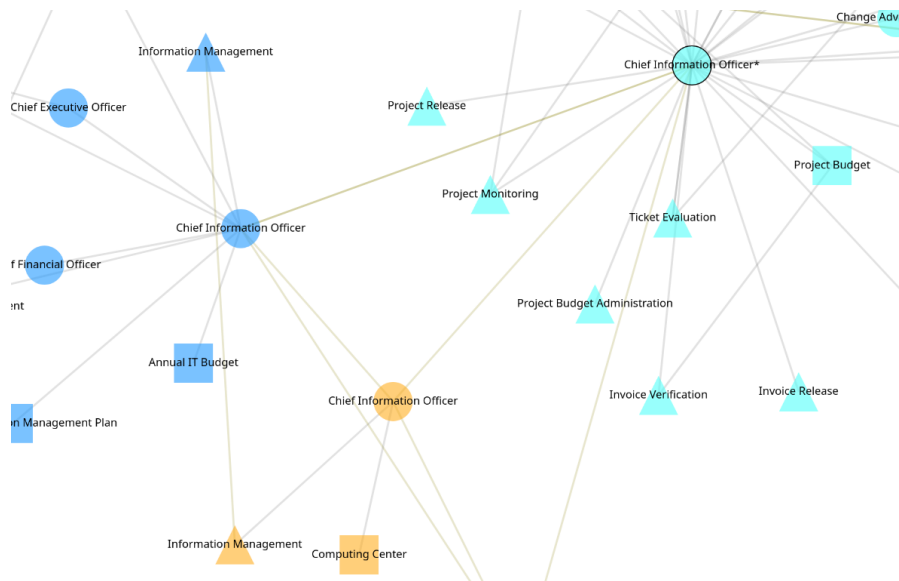


Abbildung 3.1: Ausschnitt aus dem SNIK-Graph
siehe dazu auch Jahn u. a., 2019

Aufgrund der komplexen Systeme und vielfältigen Vorgehensweisen bei Planung und Umsetzung von Krankenhausinformationssystemen stehen Chief Information Officer (CIO) vor vielen wichtigen Entscheidungen.

Das SNIK soll dabei helfen, diese Entscheidungsprozesse zu unterstützen und soll außerdem noch eine Unterstützung in Lehre und Forschung sein.

Es sollen die „komplexe[n] Zusammenhänge der verschiedenen Ansätze des Informationsmanagements [beschrieben werden]“ (Jahn u. a., 2014).

So kann das SNIK nicht nur in seiner Form als Semantisches Netz exploriert werden, es existieren auch Werkzeuge, um Forschung und Lehre nochmals zu verbessern, so in Form des SNIK-Graphen¹ und des SNIK-Quizes².

¹ snik.eu/graph

² snik.eu/quiz

In SNIK werden Daten in einer von drei disjunkten Datenklassen dargestellt:

Klasse	Beschreibung
Rolle	Diese Klasse stellt das Subjekt dar, erklärt also, wer in einem Tripel gemeint ist.
Aufgabe	Die Aufgabe gibt Auskunft, wie etwas von einem Subjekt getan wird, ist selbst also das Prädikat.
Objekttyp	Dies stellt das Objekt eines Satzes oder Tripels dar, gewährt also Auskunft darüber, welche Information noch von Subjekt und Prädikat gebraucht wird.

Tabelle 3.1: Klassen im SNIK-Metatmodell

Zur Unterstützung dieser Erklärung dient hier das offizielle Metamodel von SNIK in Abbildung 3.2.

3.1.2 SNIK-Quiz

Das SNIK-Quiz ist seit dem 9. April 2019 öffentlich als Teil des SNIK-Projekts auf Github verfolgbar ³ und basiert stark auf einem bereits existierenden Quiz, welches ebenfalls auf Github einsehbar ist⁴. Die Hauptaufgabe des SNIK-Quizes ist das bereits im SNIK-Projekt formulierte Ziel, die Lehre zu unterstützen und außerdem Studierenden eine zusätzliche Möglichkeit zu geben, sich auf Prüfungen im Taktischen sowie Strategischen Informationsmanagement vorzubereiten. Es sind bereits einfache Fragen, welche ebenfalls aus dem Semantischen Netz generiert wurden, in der zum Zeitpunkt des Schreibens aktuellen Version ⁵ verfügbar, welche aus der BELL von Wächtler, 2020 sowie von meinem Betreuer stammen.

In Abbildung 3.3 ist zu sehen, dass links oben die Nummer der gerade bearbeiteten Frage zu sehen ist, während rechts oben die bisher erreichte Punktzahl zu sehen ist. Diese errechnet sich aus den gebrauchten Antwortversuchen für die Frage. Außerdem wird oben in zentraler Position die verbleibende Antwortzeit angezeigt. Darunter steht die zu beantwortende Frage sowie die bis zu vier möglichen Antworten, von denen immer genau eine richtig ist. Wird eine Frage falsch beantwortet, wird sie rot markiert und es muss wieder aus den restlichen Möglichkeiten die richtige Antwort gefunden werden. Beantwortet man eine Frage korrekt, erfolgt eine Rückmeldung

³ <https://github.com/snikproject/quiz/commit/094096e86f95bcc71576c6a5452702408423d2c3>

⁴ <https://github.com/davidrayoussef/react-quiz>

⁵ <https://github.com/snikproject/quiz/commit/203602cab486ff31a3436a5a468ac3810caf5236>

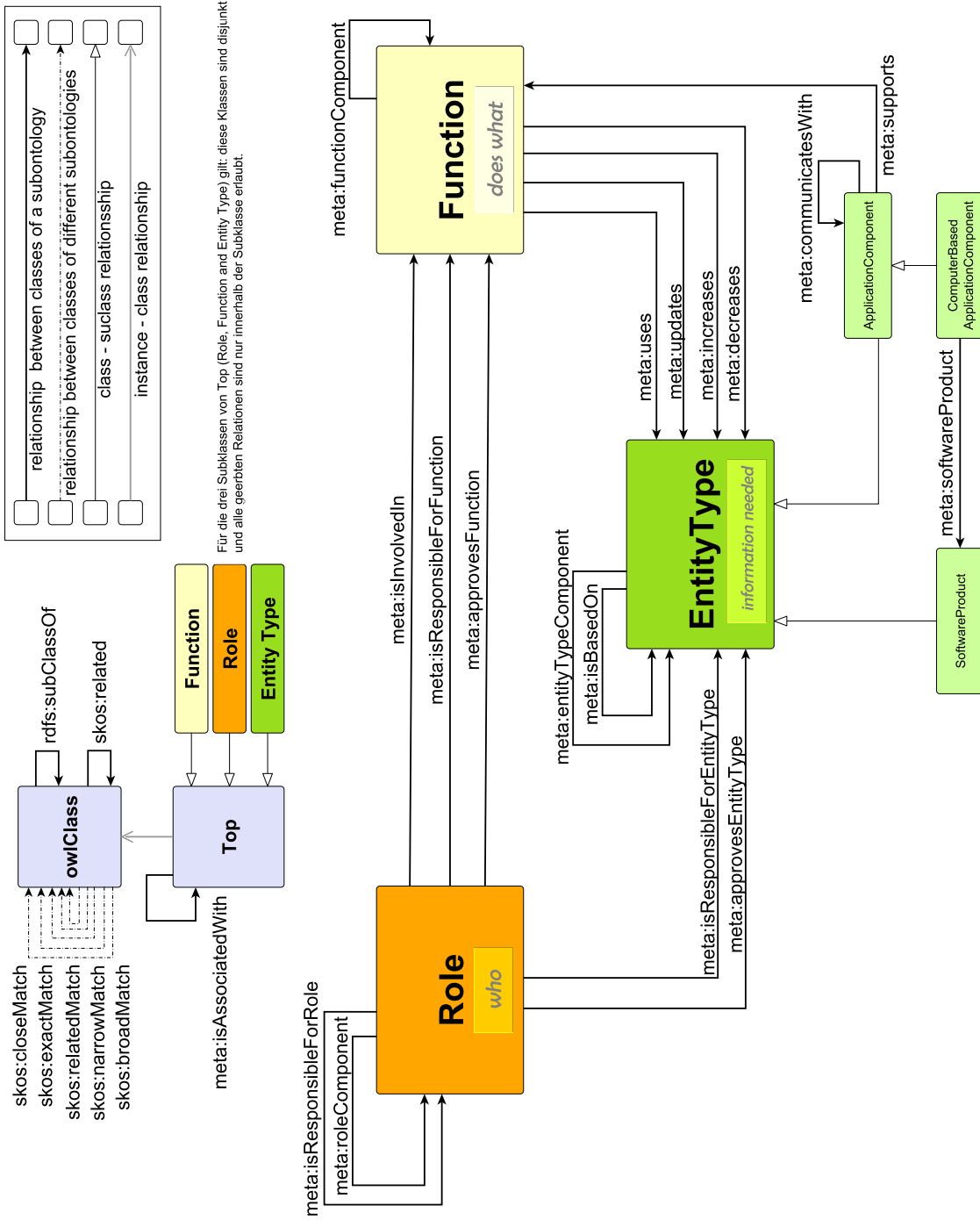


Abbildung 3.2: SNIK Metamodel Version 8

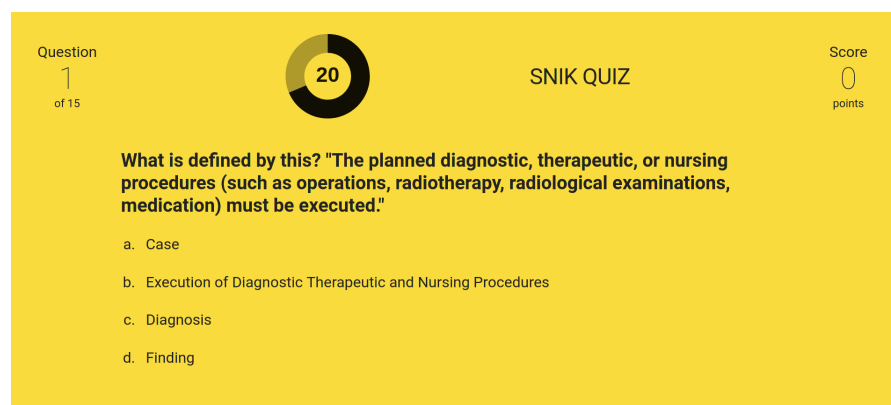


Abbildung 3.3: Benutzeroberfläche des SNIK-Quiz
(<https://snik.eu/quiz>)

in Form der hinzugewonnenen Punktzahl sowie der Information, wie viele Versuche man gebraucht hat. Es gibt grundsätzlich keine negativen Punktzahlen, man kann minimal ein Plus von 0 Punkten erhalten. Am Ende, nachdem alle fünfzehn Fragen beantwortet wurden, wird die erreichte Punktzahl zusammen mit einer kurzen Auswertung ausgegeben, wie in Abbildung 3.4 zu sehen ist.

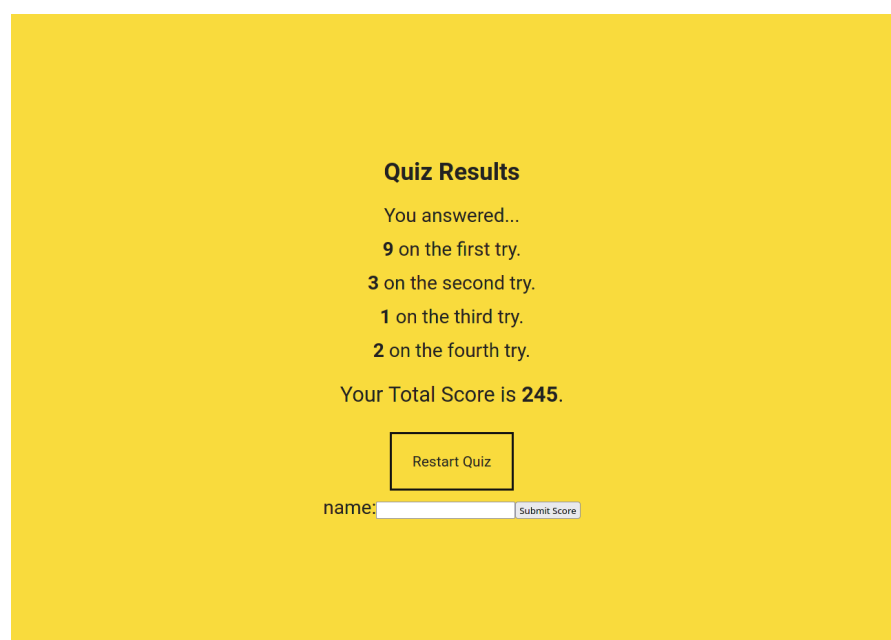


Abbildung 3.4: Ergebnis des SNIK-Quizes mit einer kurzen Auswertung
(<https://snik.eu/quiz>)

Die bereits existierenden Strategien sind in Tabelle 3.2 aufgelistet. Die bereits existierenden Strategien sind vom jetzigen Stand der Forschung innerhalb des Quizes abgeleitet, also der BELL von Wächtler, 2020. Diese sind im Prototypen allerdings zum Zeitpunkt des Schreibens noch nicht implementiert.

Strategie	Beschreibung
definition	Diese Strategie erfragt die korrekte Namen (<code>rdfs:label</code>) bei einer gegebenen Definition (<code>skos:definition</code>). Dabei gibt es vier Antwortmöglichkeiten.
contains	Hierbei werden wieder 4 Subjekte zur Auswahl gestellt, von denen eines dann mit Prädikat und Objekt einen sinnvollen Zusammenhang ergeben muss.
label- definition	Bei dieser Query wird ein Subjekt präsentiert, zu welchem dann aus den vier Optionen die korrekte Definition gefunden werden muss.

Tabelle 3.2: Bereits existierende Strategien im SNIK-Quiz

3.1.3 Vorhandene komplexe Fragen

In den Büchern Winter u. a., 2011 und Ammenwerth u. a., 2014 sind bereits am Ende jedes Kapitels Fragen definiert, welche als komplexe Fragen aufgefasst werden sollen, da diese ja der Lernstandsüberprüfung nach der Bearbeitung des jeweiligen Kapitels dienen. Diese Fragen, vorausgesetzt, sie sind für ein Multiple-Choice-Quiz geeignet, können auch als Ansatzpunkt für neue Strategien dienen.

Begonnen werden soll mit der Analyse der Fragen aus Winter u. a., 2011, indem die Kapitel aufgelistet werden, die Fragetypen eingeordnet und die Eignung für Multiple-Choice-Quizes eingeschätzt werden. Dies ist in Tabelle 3.3 abgebildet.

Kapitel	Frage	Eignung
1	Why is systematic information processing in health care institutions important?	✗
1	What are appropriate models for health information systems?	✓
1	How do health information systems look like and what architectures are appropriate?	✗
1	How can we assess the quality of health information systems?	✗
1	How can we strategically manage health information systems?	✗
1	How can good information systems be designed and maintained?	✗

2	What is the significance of information systems for health care?	X
2	How does technical progress affect health care?	X
2	Why is systematic information management important?	X
3	What is the difference between data, information, and knowledge?	X
3	What are information systems, and what are their components?	X
3	What is information management?	✓
4	What are hospital information systems?	✓
4	What are transinstitutional health information systems?	✓
4	What are challenges for health information systems?	✓
4	What are electronic health records?	✓
5	What are models, metamodels and reference models?	X
5	What are typical metamodels for modeling various aspects of HIS?	✓
5	What is 3LGM ² ?	✓
5	What are typical reference models for HIS?	✓
6	What kind of data has to be processed in hospitals?	✓
6	What are the main hospital functions?	✓
6	What are the typical information processing tools in hospitals?	✓
6	What are the different architectures of HIS?	✓
6	How can integrity and integration be achieved within HIS?	X
6.4	What application components are used in hospitals, and what are their characteristics?	X
6.5	How can architectures of HIS be categorized?	X
6.5	What differs integrity from integration?	X
6.5	What standards and technologies are available to support integration of HIS?	✓
6.5	How can integration efforts be reduced by decreasing the variety of application components in a HIS?	X

6.6	What computer-based and non-computer-based physical data processing systems can be found in hospitals?	✗
6.6	What is meant by the term „infrastructure“?	✓
6.7	How can physical data processing systems be grouped and arranged in order to support application components in an optimal way? What architectures can result?	✗
6.7	What is meant by physical integration?	✓
6.7	How do modern computing centers look like?	✗
7	How do architectures of transinstitutional health information systems differ from those of hospital information systems?	✗
7	What additional challenges do we have to cope with?	✗
7	Which strategies are appropriate for maintaining electronic health records in a transinstitutional health information system?	✗
8	Which facets of quality have to be considered in HIS?	✓
8	What are the characteristics of the quality of structures of HIS?	✗
8	What are the characteristics of the quality of processes of HIS?	✗
8	What are the characteristics of quality of outcome of HIS?	✗
8	What does information management have to balance in order to increase the quality of a HIS?	✗
8	How can quality of HIS be evaluated?	✗
8.2	What criteria for quality of data exist?	✗
8.2	What criteria for computer-based application components and physical data processing systems exist?	✗
8.2	What criteria for the overall HIS architecture exist?	✗
8.3	What are the characteristics of the quality of processes of HIS?	✗
8.4	What are the characteristics of quality of outcome of HIS especially in hospitals?	✗
8.5	What does information management have to balance in order to increase the quality of a health information system?	✗

8.6	What are major phases of an IT evaluation study?	X
8.6	What are major IT evaluation methods?	X
9	What does information management mean and how can strategic, tactical and operational information management be differentiated?	X
9	What organizational structures are appropriate for information management in hospitals?	X
9	What are the tasks and methods for strategic HIS planning?	X
9	What are the tasks and methods for strategic HIS monitoring?	X
9	What are the tasks and methods for strategic HIS directing?	X
9	How can experts for information management in hospitals be gained?	X
9.2	What does information management in general and in hospitals encompass?	X
9.2	What are the three main scopes of information management?	✓
9.2	What are the tasks of strategic, tactical, and operational information management in hospitals?	X
9.2	What is meant by IT service management and how is it related to information management?	X
9.3.4.2	Which organizational units are involved in information management?	✓
9.3.4.2	Which boards and persons are involved in information management?	✓
9.3.4.2	Who is responsible for strategic information management?	✓
9.3.4.2	Who is responsible for tactical information management?	✓
9.3.4.2	Who is responsible for operational information management?	✓
9.3.4.2	Who is the CIO, and what is his or her responsibility?	X
9.4	What are the typical tasks for strategic HIS planning?	✓
9.4	What are the typical methods for strategic HIS planning?	✓
9.4	What is the goal and typical structure of a strategic information management plan?	X

9.5	What are the typical tasks of strategic HIS monitoring?	✓
9.5	What are the typical methods of strategic HIS monitoring?	✓
9.6	What are the typical tasks of strategic HIS directing?	✓
9.6	What are the typical methods of strategic HIS directing?	✓
10	What are health care networks?	✓
10	How can health care networks be described?	✗
10	What organizational structures are appropriate for information management in health care networks?	✗
10	In the end, we are confident that you will be able to answer the question "How can good information systems be designed and maintained?"	✗

Tabelle 3.3: Vorhandene Fragetypen in Winter u. a., 2011, die Eignung bezieht sich auf Multiple-Choice-Fragen

Die 49 von insgesamt 79 Fragen, welche mit „✗“ markiert sind, fallen häufig in die Kategorie der Fragen, welche auf Erläutern oder Beschreiben setzen, um Komplexität zu erreichen. Die 30 mit „✓“ markierten Fragen sind hauptsächlich Fragen im Sinne der *definition-* bzw. *contains-*Strategie. Deutlich schlechter sieht es allerdings mit verallgemeinerbaren Fragen in Ammenwerth u. a., 2014 aus. In Tabelle 3.4 sind exemplarisch die ersten vier Fragen aus dem Buch aufgestellt.

In Kapitel 1.1 haben wir festgehalten, dass wir ein Informationssystem als das gesamte informationsverarbeitende und informationsspeichernde Teilsystem in einer Einrichtung ansehen wollen. Wieso ist uns hier die Betonung von „gesamte“ wohl wichtig?

Nennen Sie Informationssysteme, die Ihnen in Ihrem Alltag begegnen. Weshalb kann man das Informationssystem einer Einrichtung mit dem Nervensystem und Gedächtnis des Menschen vergleichen?

Erläutern Sie die Bedeutung von Informationssystemen als Qualitätsfaktor für Einrichtungen des Gesundheitswesens (z. B. ein Krankenhaus, eine Arztpraxis) anhand konkreter Beispiele.

Tabelle 3.4: Vorhandene Fragetypen in Ammenwerth u. a., 2014

Diese Fragen sind, wie fast alle in Ammenwerth u. a., 2014, auf Erläutern, Begründen und Beschreiben ausgelegt und können so nicht

sinnvoll verallgemeinert und mittels vorgenerierter Multiple-Choice-Antworten beantwortet werden.

Abschließend lässt sich zu den Fragen aus Ammenwerth u. a., 2014 und Winter u. a., 2011 sagen, dass Fragen, welche mit dem Fragewort „Was“ beginnen, gut geeignet sind, um generalisiert in Multiple-Choice-Quizen verwendet zu werden, im Gegensatz zu Fragen, welche mit „Warum“ beginnen.

3.2 FRAGEGENERIERUNG AUS GRAPHBASIERTEN DATEN

Im Sinne der Fragegenerierung in anderen Arbeiten ergeben sich starke Parallelen zwischen den von mir bearbeiteten Quellen. So sind in diesen Quellen starke Einschränkungen bei den generierten Fragen zu bemerken, meist sind diese in Form der bereits in Abschnitt 3.1.2 besprochenen Fragetypen. Auch in Vega-Gorgojo, 2019 und Heilman, 2011 findet man nur Fragen, welche auf einfache Subjekt-Prädikat-Objekt-Tripeln basieren und keine komplexen Fragen generieren können.

So findet man in Heilman, 2011 ab Seite 13 die Erklärung, dass verschiedene Hindernisse im Weg zu komplexeren Fragen stehen, so grammatikalische und logistische Probleme. Diese logistischen Probleme bestehen laut Heilman darin, dass „erhebliches menschliches Wissen“ (Heilman, 2011) zur Generierung dieser Fragen über ein bestimmtes Fachgebiet notwendig wäre und so die RDF-Datenbank nur per Hand zusammengetragen werden könnte.

Auch in Vega-Gorgojo, 2019 werden diese Schwierigkeiten beschrieben.

Die komplexen, vernetzten Fragen wurden somit nur spärlich in dieser Literatur behandelt.

Nies u. a., 2016 beschreiben eine Möglichkeit, den semantischen Zusammenhang zwischen zwei Knoten in einem semantischen Netz anhand der gemeinsamen eingehenden und/oder ausgehenden Kanten. Dies kann hier dazu benutzt werden, semantisch zusammenhängende Knoten dafür zu nutzen, Distraktoren besser auswählen zu können und so die geschlossenen Fragen schwieriger zu machen.

LÖSUNGSANSATZ

4.1 LÖSUNGSANSATZ ZUM PROBLEM P1

Hierfür muss aufgrund von Zeitlimitationen dieser Arbeit eine Methode zur initialen Einschätzung von Strategien erarbeitet werden. So wird am Anfang der Erarbeitung der Strategien eine quantifizierbare Einschätzung dieser erarbeitet, um die Strategie im Zweifelsfall schnell verwerfen zu können, um sich auf andere Strategien konzentrieren zu können.

Hierfür wird ein arbiträres Maßsystem entworfen, welches sich an 4 Kriterien misst und außerdem einen technisch limitierenden Faktor enthält, um die Erstellung der Queries in vernünftiger Zeit (unter 30 Minuten) und mit beschränkter Speichernutzung (unter 8GB Arbeitsspeicher) zu gewährleisten.

4.2 LÖSUNGSANSATZ ZUM PROBLEM P2

Das zweite Problem ist das der fehlenden Komplexität der Fragen. Dafür müssen neue Strategien exploriert und eingeführt werden, um so das Ziel zu erreichen, die Komplexität der Fragen in einem Quiz zu erhöhen.

Dafür werden *Merkmale* von Strategien genutzt. So können, um komplexere Strategien zu erstellen und die Vernetzung innerhalb der Fragen zu erhöhen, verschiedene Ansätze verfolgt werden. Folgende Ansätze sollen hier evaluiert werden:

1. In- und Ausgrade
2. Besondere Kanten, Wege und Pfade im Graphen
3. Besondere Kantenstrukturen
4. Fragekomplexe mittels aufeinander aufbauender Fragen

4.3 LÖSUNGSANSATZ ZUM PROBLEM P3

Zur abschließenden Evaluation der entwickelten Strategien wird ein Fragebogen erstellt, welcher einer Expertengruppe vorgelegt wird, welche diesen Test jeweils mehrmals durchführt. Dafür wird ein Fragebogen entwickelt, welcher auch die individuellen Kriterien der Befragten mit einschließt.

Die Evaluation ist notwendig, da die Kriterien zur Lösung des Pro-

blems P_1 nicht zur alleinigen Entscheidbarkeit der Komplexität und Güte der Fragen ausreichen.

AUSFÜHRUNG UND ERGEBNISSE

5.1 INITIALE EINSCHÄTZUNG DER STRATEGIEN

Eine initiale Einschätzung ist notwendig, da sowohl die Expertengruppe als auch ich für diese Arbeit nur einen begrenzten Zeitrahmen zur Verfügung haben, um die Strategien auf die geforderten Qualitäten zu prüfen. So leitet sich hier die Notwendigkeit ab, im Voraus für mich die Strategien, alte wie neue, auf Tauglichkeit für die Multiple-Choice-Aufgaben zu untersuchen.

Dafür braucht es Kriterien, hier als *Dimensionen* bezeichnet, an denen die Strategien zu beurteilen sind. Auf jede der Strategien passt beispielsweise die Frage nach der Länge der Antwortmöglichkeiten, eine unpassende Dimension wäre hingegen der Betrag der Menge der Antwortmöglichkeiten pro Frage.

So wird es hier notwendig, eine (übersichtliche) Menge an Dimensionen zu definieren, um anschließend eine numerische Möglichkeit für diese zu definieren, die der Einordnung dient.

In dieser Arbeit sollen hierfür vier Dimensionen definiert werden:

1. Ausführlichkeit der Frage
2. Länge der Antwortmöglichkeiten
3. Vernetzung der Inhalte
4. Semantische Ähnlichkeit der Antwortmöglichkeiten

Die *Ausführlichkeit der Fragen* spiegelt wieder, wie viele (zweckdienliche) Informationen mittels der Fragestellung vermittelt werden. So sind Fragen zu bevorzugen, welche möglichst viele Aussagen sowohl über die Eigenschaften der Frage, z.B. wie viele Antwortmöglichkeiten es gibt, und der Zielstellung der Frage, also was erfragt wird, geben. Auch zusätzliche Informationen, die zur Beantwortung der Frage hilfreich sind, sind positiv zu bewerten. Dazu könnte beispielsweise eine kurze Einordnung der Frage in die Gesamthematik passen oder eine allgemeine Einschränkung, wie z.B. die Beschränkung einer Frage auf einen bestimmten Aspekt wie das Taktische Informationsmanagement.

Die *Länge der Antwortmöglichkeiten* sollte so kurz wie möglich sein, da eine Testperson diese gegeneinander abwägen muss, wobei die Frage gleichzeitig im Kopf behalten werden muss.

Die *Vernetzung der Inhalte* ist die wichtigste Dimension, da hier erfragt wird, wie stark eine Testperson mit der Materie vertraut sein muss, um

diese Frage gut beantworten zu können. So wird verlangt, dass Verbindungen im Graphen erkannt und genutzt werden, um auch themen- und gebietsübergreifend Fragen beantworten zu können. Strategien mit hohen Werten in der Vernetzung haben außerdem den Vorteil, dass gelernte Inhalte eher behalten werden als bei reinen Fragen, die auf reiner Reproduktion der Inhalte basieren (Heilman, 2011). Deshalb ist diese Dimension mit besonderer Stärke zu werten.

Als letzte Dimension wird hier explizit auf die Antwortmöglichkeiten eingegangen, da diese bei hoher semantischer Ähnlichkeit und damit auch thematischer Ähnlichkeit schwerer zu beantworten sind.

Allerdings braucht es zu einer einheitlichen Einschätzung auch eine Metrik, um die Merkmale zu gewichten und am Ende ein numerisches Maß zuordnen zu können. Dieses hat dann außerdem die Eigenschaft, im Gegensatz zu reinen Gefühlsentscheidungen, dass man es evaluieren und falls notwendig auch anpassen kann. Zur initialen Einschätzung sind für die Dimensionen eins, drei und vier Formeln zu entwerfen, welche folgende Eigenschaften vereinen:

1. es werden ganzzahlige Werte im Intervall $[1, 10]$ angenommen
2. niedrige Werte sollen „bestraft“, hohe Werte „gefördert“ werden
3. der Wertebereich ist rational in $[0, 1]$

Eine Ausnahme soll hier die Länge der Antwortmöglichkeiten darstellen, welche eine gesonderte Stellung bekommt: niedrige Werte sollen „belohnt“ werden, da ja die Länge der Antworten auch möglichst niedrig sein soll. Die anderen Kriterien gelten wie oben beschrieben.

Eine Formel, welche diese Anforderungen für die Dimensionen eins, drei und vier erfüllt, ist

$$f(x) = \frac{1}{2} \sin \left(\frac{11}{10\pi} \cdot (x - 1) - \frac{\pi}{2} \right) + 0.5$$

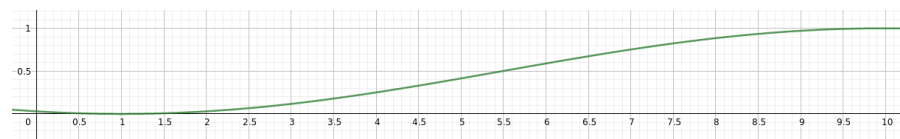


Abbildung 5.1: Funktionsgraph für $f(x)$, rationale Werte sind mit eingezeichnet, um den Verlauf zu verdeutlichen

Für die zweite Dimension hat sich eine verschobene Form der ersten Formel als sinnvoll erwiesen:

$$g(x) = \frac{1}{2} \sin \left(\frac{11}{10\pi} \cdot \left(x + \frac{7.64}{3} \pi \right) - \frac{\pi}{2} \right) + 0.5$$

Wie in den Abbildungen Abbildung 5.1 und Abbildung 5.2¹ zu sehen,

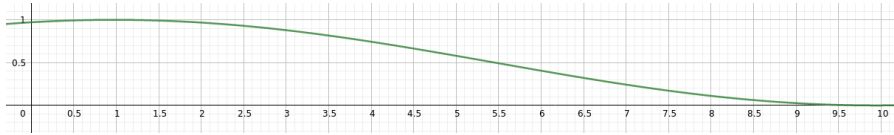


Abbildung 5.2: Funktionsgraph für $g(x)$, rationale Werte sind mit eingezeichnet, um den Verlauf zu verdeutlichen

werden niedrige respektive hohe Werte bestraft, um einen starken Fokus auf die bevorzugte Form der Multiple-Choice-Fragen zu legen. Die bisher existierenden Strategien können dann mithilfe dieser Formeln eingeordnet werden, die Werte a, l, v und s stehen dann jeweils für die vergebenen Punkte pro Dimension, die gestrichelten Werte a', l', v' und s' stehen dann für die Werte, die aus der Formel errechnet werden. Zusammengefasst werden diese Werte dann im Wert f , welcher den Fragewert ausdrückt und im Intervall $[0, 1]$ besteht:

$$f = \frac{a' + l' + 3v' + 2s'}{7}$$

Dabei werden Vernetzung und semantische Ähnlichkeit mit den Faktoren drei bzw. zwei bedacht, um sie stärker zu gewichten als die ersten beiden Dimensionen, welchen eine gleiche Gewichtung zugeordnet wurde.

¹ erstellt mit GeoGebra: <https://www.geogebra.org/calculator>

Strategie	a	a'	l	l'	v	v'	s	s'	f
Aus Vega-Gorgojo, 2019 entnommene Strategien									
boolean	6	0.59	1	1.0	3	0.12	3	0.12	0.31
image	7	0.75	1	1.0	4	0.25	5	0.42	0.48
numeric	7	0.75	1	1.0	4	0.25	7	0.75	0.57
relation	8	0.89	1	1.0	4	0.25	7	0.75	0.59
date	8	0.89	2	0.97	5	0.42	7	0.75	0.66
greatest	7	0.75	1	1.0	5	0.42	8	0.89	0.68
group	8	0.89	1	1.0	6	0.59	7	0.75	0.74
Aus SNIK-Protoyp entnommene Strategien									
definition- a	8	0.89	1	1.0	2	0.03	2	0.03	0.29
contains- a	9	0.97	2	0.97	4	0.25	5	0.42	0.5
Aus Wächtler, 2020 entnommene Strategien									
label- definition	3	0.12	8	0.11	2	0.03	5	0.42	0.17
definition- b	10	1.0	1	1.0	2	0.03	3	0.12	0.33
contains- b	9	0.97	2	0.97	6	0.59	7	0.75	0.74

Tabelle 5.1: Bewertung bereits existierender Strategien; -a und -b nur zur Unterscheidung der sonst doppelt vorhandenen Strategien

In der Tabelle 5.1 sind im obersten Abschnitt die in Vega-Gorgojo, 2019 aufgeführten Strategien zur Fragegenerierung zu sehen. Diese haben oft ähnliche Eigenschaften wie die verbesserte *definition-b*-Strategie, da sie eine einfache Frage formulieren, um dann je vier kurze Antwortmöglichkeiten zu geben. Oft werden dabei aber zusätzliche Informationen wie ein optionales Bild bereitgestellt, was zugleich der Ausführlichkeit der Frage als auch der Vernetzung dient. Hier ist vor allem die *group*-Strategie hervorzuheben. Auch die Distraktoren wurden, wie im Falle der *date*-Strategie, speziell angepasst, um sich dynamisch in Relation zum abgefragten Datum zu verhalten, ältere Datumsangaben also weiter auseinanderliegen, was wiederum die Schwierigkeit erhöhen kann.

Darunter sind die bisher existierenden Strategien des SNIK-Quizes sichtbar, jeweils mit einer entsprechenden Bewertung. Im oberen Bereich sind zwei Strategien zu sehen, welche in dem bisherigen Prototypen eingesetzt werden, die drei unteren Strategien entstammen Wächtler, 2020, welcher die zu der Zeit existierenden Strategien verbes-

sert und die *label-definition*-Strategie hinzugefügt hat. Bei den verbesserten Strategien sieht man eine deutliche Verbesserung hinsichtlich der Dimensionen der generierten Fragen, die *label-definition*-Strategie muss allerdings als Negativbeispiel erhalten: Eine sehr kurze Frage mit wenig zusätzlichen Informationen und sehr langen Antwortmöglichkeiten. Die verbesserte *contains-b*-Strategie hingegen soll in dieser Arbeit als Standard eingesetzt werden, sodass alle neuen Strategien zumindest diesen Fragewert f erreichen müssen, um als „gute“ Strategien dienen zu können.

5.2 TECHNISCHE HERAUSFORDERUNGEN

Die Technischen Herausforderungen dieses Projekts lassen sich auf vier Sachverhalte reduzieren:

1. Begrenzung des Massenspeichers
2. Möglichkeiten der SPARQL-Abfragen
3. Verfügbarer RAM
4. Verfügbare Zeit

5.2.1 *Begrenzung des Massenspeichers*

Um den Computer auch während des Schreibens dieser Bachelorarbeit sinnvoll für andere Aufgaben verwenden zu können, gibt es hier eine Begrenzung des verwendbaren Massenspeichers auf rund 20 GB. Dies ist für die meisten Strategien kein Problem, allerdings wird dieser Punkt in Abschnitt 5.3.1 noch einmal aufgegriffen werden, um eine Schwierigkeit während des Erarbeitens aufzugreifen.

5.2.2 *Möglichkeiten der SPARQL-Abfragen*

Hier gibt es vier Möglichkeiten, per SPARQL Ergebnisse aus Ontologien zu generieren. Diese sollen, anhand der Eignung absteigend aufgelistet, kurz dargestellt werden. Zum Benchmarking dient hier die Query in Algorithmus 5.1, welche jeweils nur ein mal ausgeführt wurde, um Caching als Problem zu umgehen. Die Ergebnisse sind in Tabelle 5.2 zu sehen.

```
SELECT ?a ?b ?c ?d ?e ?f
WHERE {
# Ein Dreieck
?a ?p11 ?c .
?a ?p12 ?b .
?c ?p13 ?b .
# Ein Viereck, dass an einem Punkt mit dem Dreieck verbunden ist
?a ?p21 ?d .
```

```

?d ?p22 ?e .
?e ?p23 ?f .
?f ?p24 ?a .
FILTER(?a < ?b && ?b < ?c && ?c < ?d && ?d < ?e && ?e < ?f)
}

```

Alg. 5.1: Benchmarking-Query

Das Benchmark ergab folgende Laufzeiten, bis ein vollständiges Ergebnis zustande kam:

Service	Zeit	Bemerkung
virtuoso (lokal)	3.1 s	per Stoppuhr gemessen
virtuoso (snik.eu)	3.1 s	per Stoppuhr gemessen
Fuseki SPARQL (lokal)	23.24 s	aus Serverkonsole entnommen
rdflib (lokal)	> 1 h	Berechnung nach einer Stunde mit nicht einem Ergebnis in Konsole abgebrochen

Tabelle 5.2: Benchmarkergebnisse der verschiedenen Abfragemöglichkeiten

5.2.2.1 *virtuoso-open-source*

In der open-source-Implementierung von virtuoso² treten nur wenige Probleme auf, welche zumeist auch umgangen werden können, indem die config-Datei des lokalen Servers angepasst wird. So können eine Limitierung der maximalen erwarteten Zeit pro Anfrage (standardmäßig 400 Sekunden) sowie die maximale RAM-Belegung (standardmäßig 1 GB) verändert werden. Auf meinem lokalen Server habe ich deswegen die Zeitlimitierung ganz gelöscht und die RAM-Limitierung auf 8 GB gesetzt. Somit eignet sich der open-source-Server sehr gut auch für komplexere Queries und ist nur noch durch eine relativ hohe Schwelle an RAM-Verbrauch limitiert und wird deswegen von mir verwendet.

5.2.2.2 *Onlinedienst: snik.eu*

Über snik.eu/sparql wird vom IMISE eine Instanz des virtuoso-Servers bereitgestellt, welche allerdings noch die Limitierungen besitzt, welche ich in der lokalen Variante angepasst habe. Damit ist diese Implementierung weniger geeignet.

² <http://vos.openlinksw.com/owiki/wiki/VOS/>

5.2.2.3 *Fuseki SPARQL*

Auch dieser Dienst³ stellt mir einen lokalen Server zur Verfügung. Da dieser aber leider in meinen Tests bedeutend langsamer war (ca. um den Faktor 7, siehe Tabelle 5.2) als der lokale virtuoso-Server, habe ich mich hier dagegenentschieden.

5.2.2.4 *Eigene Pythonimplementierung*

Die eigene Pythonimplementierung unter der Nutzung der Bibliothek rdflib⁴ schnitt hier am schlechtesten ab, da diese für meine Queries mindestens um den Faktor 1200 mal mehr Zeit brauchte als der lokale virtuoso-Server. Dies ist keine Option.

5.2.3 *Verfügbare RAM*

Bei einigen Versuchen hinsichtlich komplexeren Queries kam es mit dem vom IMISE zur Verfügung gestellten Server zu Situationen, in denen der RAM nicht ausreichte, um die Ergebnisse bis zur Ausgabe zwischenzuspeichern. Nach einer Verlagerung auf die lokalen Ebene waren immerhin statt einem Gigabyte 8 für den lokalen Server zur Verfügung, welche auch meist ausgereicht haben.

5.2.4 *Verfügbare Zeit*

Die Zeitlimitierung wurde lokal umgangen und spielte so für die in diesem Kapitel vorgestellten Queries keine Rolle mehr, allerdings ist auch die Arbeitszeit der Bachelorarbeit limitiert, Fehler in Queries oder Ungenauigkeiten schlagen also mehr zu Buche und so sollten lange Laufzeiten vermieden werden.

5.3 NEUE STRATEGIEN

Die für die Erstellung der Strategien genutzten Präfixe sind in Tabelle 5.3 zu sehen.

³ <https://jena.apache.org/index.html>

⁴ <https://rdflib.readthedocs.io/en/stable/index.html>

Präfix	URI
rdfs	http://www.w3.org/2000/01/rdf-schema#
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
owl	http://www.w3.org/2002/07/owl#
skos	http://www.w3.org/2004/02/skos/core#
meta	http://www.snik.eu/ontology/meta/
ob	http://www.snik.eu/ontology/ob/
bb	http://www.snik.eu/ontology/bb/

Tabelle 5.3: In der Arbeit benutzte Präfixe

5.3.1 *intertwined-Strategie*

In dieser Strategie soll ein Fokus auf die unmittelbare Vernetzung und die explizite Kenntnis des Graphen gelegt werden. Dafür sollen, ausgehend von der Frage, welche der drei Antwortmöglichkeiten besonders stark vernetzt sei, je drei Begriffe gegeben werden. Die richtige Antwortmöglichkeit soll dann drei Begriffe enthalten, welche jeweils paarweise direkt miteinander in Beziehung stehen, bildlich gesprochen also ein Dreieck im Graphen und damit einen Kreis bilden würden, siehe dazu Abbildung 5.3.

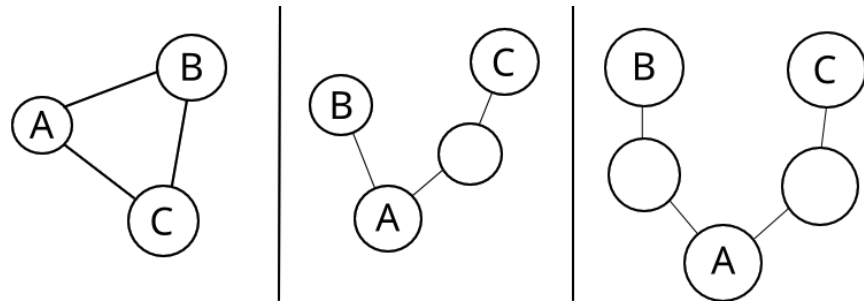


Abbildung 5.3: Von links nach rechts: Struktur der korrekten Strategie, sowie der schweren und leichten Distraktoren-Strategie

Die Distraktoren dürften dieses „Dreieck“ dann nicht bilden. Eine schwere Variation der Distraktoren wäre dann, dass genau eine direkte Kante innerhalb des Begriffstripels existiert, sowie ein Pfad innerhalb des Graphen, der die Länge 2 besitzt. Die leichte Version ergibt sich, wenn es keinen einzigen der Länge 1 innerhalb eines Tripels gibt, aber 2 distinkte Pfade, je mit Länge 2.

5.3.1.1 Initiale Einschätzung

Die Ausführlichkeit der Frage ist, da sie bei allen generierten Tripeln gleich ausfällt, relativ niedrig. Dies kann aber verbessert werden, indem eine thematische Einordnung erfolgt, welche zumindest eine grobe Richtung innerhalb des Graphen vorgibt. Damit wird die Frage für mit dem Thema vertraute Testpersonen nicht einfacher, und die, die keine Ahnung haben, werden auch nicht davon profitieren. Es wäre also ein sinnvolles Zugeständnis in Anbetracht der noch erläuterten anderen Merkmale. Trotz dieser positiven Merkmale ist dennoch die Ausführlichkeit der Fragen durch die Homogenität der Grundfrage nur mit 7 Punkten zu bewerten.

In Bezug auf die Länge der Antwortmöglichkeiten lässt sich ein sehr positives Fazit ziehen, da hier nur je drei Begriffe pro Antwortmöglichkeit gegeben werden. Die Länge ist also mit 1 Punkt zu bewerten.

Die Strategie setzt auf eine starke Kenntnis der Vernetzung und der Zusammenhänge innerhalb des SNIK-Graphen und fordert eine gute Kenntnis der Ontologie, aus welcher der Graph erzeugt wurde. Damit können hier als Fazit 7 Punkte gegeben werden.

Als letztes folgt die Dimension der Semantischen Ähnlichkeit der Antwortmöglichkeiten. Hier soll eine hohe Punktzahl dadurch erreicht werden, dass es einen Ausgangsbegriff geben soll, den alle Antwortmöglichkeiten verwenden, um so zum einen, wie bereits beschrieben, das Themengebiet einzuschränken und zum anderen die Frage schwerer zu machen. Dadurch wird die Frage mit 8 Punkten bewertet.

Dadurch ergibt sich folgende Tabelle als Ergebnis dieser initialen Einschätzung:

Strategie	a	a'	l	l'	v	v'	s	s'	f
contains-b	9	0.97	2	0.97	6	0.59	7	0.75	0.74
intertwined	7	0.75	1	1.0	7	0.75	8	0.89	0.83

Tabelle 5.4: Initiale Bewertung der *intertwined*-Strategie, dazu als Vergleichswert die *contains-b*-Strategie

Es ist also als Ergebnis dieser initialen Einschätzung zu sehen, dass die *intertwined*-Strategie einen Fragewert von 0.83 aufweist und damit nochmals eine Verbesserung zur *contains-b*-Strategie (0.74) darstellen würde.

5.3.1.2 Implementierung

Die Implementierung gestaltete sich komplizierter als gedacht, nachdem Prototyp Algorithmus 5.2 nicht zu einem befriedigendem Ergebnis führte, da aufgrund der Platz- und Zeitkomplexität der Anfrage

der Server nicht die benötigten Ressourcen zur Verfügung stellen konnte.

```

SELECT ?a1l ?a2l ?a3l ?b1l ?b2l ?c1l ?c2l ?d1l ?d2l
WHERE {
  ?a1 ?pa1 ?a2 .
  ?a1 ?pa2 ?a3 .
  ?a2 ?pa3 ?a3 .
  ?a1 rdfs:label ?a1l.
  ?a2 rdfs:label ?a2l.
  ?a3 rdfs:label ?a3l.
  ?a1 ?pb1 ?b1 .
  ?a1 ?pb2 ?b2 .
  FILTER NOT EXISTS { ?b1 ?pb3 ?b2 }
  ?b1 rdfs:label ?b1l.
  ?b2 rdfs:label ?b2l.
  ?a1 ?pc1 ?c1 .
  ?a1 ?pc2 ?c2 .
  FILTER NOT EXISTS { ?c1 ?pc3 ?c2 }
  ?c1 rdfs:label ?c1l.
  ?c2 rdfs:label ?c2l.
  ?a1 ?pd1 ?d1 .
  FILTER NOT EXISTS { ?d1 ?pd3 ?d2 }
  FILTER NOT EXISTS { ?a1 ?pd3 ?d2 }
  ?d1 rdfs:label ?d1l.
  ?d2 rdfs:label ?d2l.
  FILTER(?a1 != ?a2 && ?a1 != ?a3 && ?a1 != ?b1 && ?a1 != ?b2 &&
    ↪ ?a1 != ?c1 && ?a1 != ?c2 && ?a1 != ?d1 && ?a1 != ?d2 && ?
    ↪ b1 != ?c1 && ?c1 != ?d1)
}

```

Alg. 5.2: Grundlegende Idee der intertwined-Strategie

So wurden hier mehrere Subqueries genutzt, welche weniger umfassende Aussagen tätigen, um so die Funktion zu gewährleisten, wie in Algorithmus 5.3.

```

SELECT DISTINCT ?resa (STR(?a1) AS ?als) (STR(?b1) AS ?bls) (STR
  ↪ (?c1) AS ?cls) (STR(?d1) AS ?dcls) (STR(?del) AS ?dels) (
  ↪ STR(?d2bl) AS ?d2bls) (STR(?d2dl) AS ?d2dls)
WHERE {

# Sub-Query mit schwerer Distraktorenstrategie aussen und
  ↪ korrekter Strategie innen
{
SELECT DISTINCT ?resa ?a1 ?b1 ?c1 ?d2a1 ?d2bl ?d2dl
WHERE {

# Zwecks Optimierung hier die Bedingung, dass STR(?a1) = STR(?
  ↪ d2a1), um ähnliche Ergebnisse zu erzeugen
?d2a1 a owl:Class.
?d2a1 rdfs:label ?d2a1 .
FILTER(langMatches(LANG(?d2a1), "en") )
FILTER(STR(?a1) = STR(?d2a1))

```

```

?d2b a owl:Class.
?d2c a owl:Class.
?d2d a owl:Class.

?d2b rdfs:label ?d2bl .
?d2d rdfs:label ?d2dl .
FILTER (langMatches(LANG(?d2bl), "en") && langMatches(LANG(?d2dl)
  ↪ , "en" )

?d2a ?p1 ?d2b .
?d2a ?p21 ?d2c .
?d2c ?p22 ?d2d .

# Filterung: keine gleichen Ergebnisse fuer Distraktoren
FILTER ( ?d2a < ?d2b && ?d2a < ?d2c && ?d2c < ?d2d && ?d2b != ?
  ↪ d2c )

# Filtere unerwunschte Begriffe
FILTER ( ?d2a != <http://www.snik.eu/ontology/meta/EntityType> &&
  ↪ ?d2a != <http://www.snik.eu/ontology/meta/Role> && ?d2a
  ↪ != <http://www.snik.eu/ontology/meta/Function> && ?d2a !=
  ↪ <http://www.snik.eu/ontology/meta/subTopClass> && ?d2a !=
  ↪ <http://www.snik.eu/ontology/meta/Top>)
FILTER ( ?d2b != <http://www.snik.eu/ontology/meta/EntityType> &&
  ↪ ?d2b != <http://www.snik.eu/ontology/meta/Role> && ?d2b
  ↪ != <http://www.snik.eu/ontology/meta/Function> && ?d2b !=
  ↪ <http://www.snik.eu/ontology/meta/subTopClass> && ?d2b !=
  ↪ <http://www.snik.eu/ontology/meta/Top>)
FILTER ( ?d2d != <http://www.snik.eu/ontology/meta/EntityType> &&
  ↪ ?d2d != <http://www.snik.eu/ontology/meta/Role> && ?d2d
  ↪ != <http://www.snik.eu/ontology/meta/Function> && ?d2d !=
  ↪ <http://www.snik.eu/ontology/meta/subTopClass> && ?d2d !=
  ↪ <http://www.snik.eu/ontology/meta/Top>)

# Sub-Sub-Query, korrekte Ergebnisse
{
SELECT DISTINCT ?resa ?al ?bl ?cl
WHERE {

# Dreiecksbeziehungen finden
?a a owl:Class.
?b a owl:Class.
?c a owl:Class.
?a ?p1 ?b .
?a ?p2 ?c .
?b ?p3 ?c .

# Filtere unerwunschte Ergebnisse
FILTER(?p1 != <http://www.w3.org/2004/02/skos/core#closeMatch> &&
  ↪ ?p2 != <http://www.w3.org/2004/02/skos/core#closeMatch>

```

```

    ↪ && ?p3 != <http://www.w3.org/2004/02/skos/core#closeMatch
    ↪ >)

# Ontologieherkunft in menschenlesbare Sprache uebersetzen
?a ^<http://open.vocab.org/terms/defines> ?domain .
BIND ( IF(?domain = <http://www.snik.eu/ontology/he>, "General
    ↪ Information Management", IF(?domain = <http://www.snik.eu/
    ↪ ontology/ob>, "Tactical Information Management", IF(?
    ↪ domain = <http://www.snik.eu/ontology/ciox>, "CIOX", "
    ↪ Strategic Information Management")) AS ?resa)

# keine Dopplungen
FILTER ( ?a < ?b && ?a < ?c && ?b < ?c)

# Filtere unerwünschte Ergebnisse
FILTER ( ?a != <http://www.snik.eu/ontology/meta/EntityType> && ?
    ↪ a != <http://www.snik.eu/ontology/meta/Role> && ?a != <
    ↪ http://www.snik.eu/ontology/meta/Function> && ?a != <http
    ↪ ://www.snik.eu/ontology/meta/subTopClass> && ?a != <http
    ↪ ://www.snik.eu/ontology/meta/Top>)
FILTER ( ?b != <http://www.snik.eu/ontology/meta/EntityType> && ?
    ↪ b != <http://www.snik.eu/ontology/meta/Role> && ?b != <
    ↪ http://www.snik.eu/ontology/meta/Function> && ?b != <http
    ↪ ://www.snik.eu/ontology/meta/subTopClass> && ?b != <http
    ↪ ://www.snik.eu/ontology/meta/Top>)
FILTER ( ?c != <http://www.snik.eu/ontology/meta/EntityType> && ?
    ↪ c != <http://www.snik.eu/ontology/meta/Role> && ?c != <
    ↪ http://www.snik.eu/ontology/meta/Function> && ?c != <http
    ↪ ://www.snik.eu/ontology/meta/subTopClass> && ?c != <http
    ↪ ://www.snik.eu/ontology/meta/Top>)

?a rdfs:label ?al .
?b rdfs:label ?bl .
?c rdfs:label ?cl .

# Nur englischsprachige Ergebnisse
FILTER ( langMatches(LANG(?al), "en") && langMatches(LANG(?bl), "
    ↪ en") && langMatches(LANG(?cl), "en") )
}
}
}
}

# leichtere Distraktorenstrategie
# auch hier aus Performance-Grunden die Abfragen zur Filterung
    ↪ von ?da am Anfang
?da a owl:Class.
FILTER ( ?da != <http://www.snik.eu/ontology/meta/EntityType> &&
    ↪ ?da != <http://www.snik.eu/ontology/meta/Role> && ?da != <
    ↪ http://www.snik.eu/ontology/meta/Function> && ?da != <http

```

```

    ↪ <http://www.snik.eu/ontology/meta/subTopClass> && ?da != <http
    ↪ <http://www.snik.eu/ontology/meta/Top>)
?da rdfs:label ?dal .
FILTER(langMatches(LANG(?dal), "en") )
FILTER(STR(?al) = STR(?dal) )

?db a owl:Class.
?dc a owl:Class.
?dd a owl:Class.
?de a owl:Class.
?da ?p11 ?db .
?db ?p12 ?dc .
?da ?p21 ?dd .
?dd ?p22 ?de .
FILTER ( ?da < ?db && ?da < ?dd && ?db < ?dc && ?dd < ?de && ?db
    ↪ != ?dd && ?dc != ?de )

# wieder unerwünschte Ergebnisse filtern
FILTER ( ?dc != <http://www.snik.eu/ontology/meta/EntityType> &&
    ↪ ?dc != <http://www.snik.eu/ontology/meta/Role> && ?dc != <
    ↪ http://www.snik.eu/ontology/meta/Function> && ?dc != <http
    ↪ <http://www.snik.eu/ontology/meta/subTopClass> && ?dc != <http
    ↪ <http://www.snik.eu/ontology/meta/Top>)
FILTER ( ?de != <http://www.snik.eu/ontology/meta/EntityType> &&
    ↪ ?de != <http://www.snik.eu/ontology/meta/Role> && ?de != <
    ↪ http://www.snik.eu/ontology/meta/Function> && ?de != <http
    ↪ <http://www.snik.eu/ontology/meta/subTopClass> && ?de != <http
    ↪ <http://www.snik.eu/ontology/meta/Top>)

?dc rdfs:label ?dcl .
?de rdfs:label ?del .
FILTER (langMatches(LANG(?dcl), "en") && langMatches(LANG(?del),
    ↪ "en") )

# keine Dopplungen
FILTER(STR(?d2dl) < STR(?dcl) && STR(?d2dl) != STR(?del))
}

```

Alg. 5.3: *intertwined*-Strategie

Allerdings müssen die aus den Queries entstandenen Antwortmöglichkeiten noch in die Fragevorlage eingesetzt werden werden. Dafür dient ein Pythonskript, welches die Ergebnisse aus den csv-Dateien ausliest und einsetzt. Diese generierten Fragen werden dann zusammengefasst und in eine .js-Datei als Ausgabe geschrieben.

```

import csv

def read_file(path):
    line_list = []
    with open(path, mode = 'r') as file:
        csvFile = csv.reader(file)

```

```

        bool_append = True
        for lines in csvFile:
            for i in range(1,4):
                for k in range(4,8):
                    if k == 4 or k== 6:
                        if lines[k].lower == lines[k+1].lower():
                            bool_append = False
                    if lines[i].lower() == lines[k].lower():
                        bool_append = False
                if bool_append:
                    line_list.append(lines)
            else:
                bool_append = True
        return line_list

def insert_into_question(liste):
    jslist = []
    for qu in liste:
        js_question_template = "{ question: <span>In the context
↪ of #4, which one of the following triples belongs together
↪ the most?</span>, answers: [<span>#1</span>,<span>#2</
↪ span>,<span>#3</span>,<span>#4</span>], correct: o }\n"
        ans_list = qu
        correct = ans_list[1] + " - " + ans_list[2] + " - " +
↪ ans_list[3]
        distractor1 = ans_list[1] + " - " + ans_list[4] + " - " +
↪ ans_list[5]
        distractor2 = ans_list[1] + " - " + ans_list[6] + " - " +
↪ ans_list[7]
        js_question_template = js_question_template.replace("#1",
↪ correct)
        js_question_template = js_question_template.replace("#2",
↪ distractor1)
        js_question_template = js_question_template.replace("#3",
↪ distractor2)
        js_question_template = js_question_template.replace("#4",
↪ ans_list[0])
        jslist.append(js_question_template)
    return jslist

def export_to_file(liste, files, append):
    point = "w"
    if append:
        point = "a"
    with open(files, "a") as f:
        for t in liste:
            f.write(t)

export_to_file( insert_into_question(read_file("../SPARQL/
↪ intertwined/intertwined_optimized.csv")) , "../SPARQL/
↪ results/intertwined.js", False)

```

Alg. 5.4: Python intertwined Programmcode

5.3.1.3 Ergebnis

Die *intertwined*-Strategie ergab 712261 verschiedene Fragen. Vier zufällig ausgesuchte Beispiele sind in Tabelle 5.5 aufgeführt.

5.3.2 *close-by*-Strategie

Diese Strategie legt den Fokus auf den Zusammenhang zwischen den Büchern Ammenwerth u. a., 2014 und Winter u. a., 2011. Dafür werden über die *closeMatch*-Relation Verbindungen zwischen den Büchern gesucht. Dies beinhaltet dann Begriffe, welche in beiden Büchern auftreten. Ausgehend davon sollen dann ebenfalls in beiden Büchern auftretende Begriffe gesucht werden, welche verschiedene Grundklassen aufweisen: im einen Buch als Aufgabe und im anderen als Subjekt. Damit sollen dann Sätze gebildet werden, welche eine Gegenüberstellung in den Büchern erlauben, um so Brücken zu bauen und das Wissen des einen Buches effektiv auf das andere transferieren zu können. Ein Fragesatz könnte dann lauten: „In [*Tactical / Strategic IM*], the [*Subjekt*] [*Prädikat*] the [*Objekt*], while in [*Strategic / Tactical IM*], the [*Subjekt*] [*Prädikat*] the [*Objekt*]“ mit 2 Antwortmöglichkeiten, wahr und falsch.

5.3.2.1 Initiale Einschätzung

Durch den Charakter der Frage und die Einordnung in das jeweilige Buch fällt die Ausführlichkeit der Frage hier hoch aus, da es praktisch keine allgemeinen Informationen mehr gibt, welche zusätzlich und frageunabhängig bereitgestellt werden können. Dies führt zu einer Bewertung mit 10 Punkten. Genauso gut fällt die Bewertung hinsichtlich der Länge der Antwortmöglichkeiten aus.

Durch die Strategie wird selbst eine hohe Vernetzung mit guten Vergleichen gefördert und so das Ausnutzen von bereits vorhandenem Wissen in einem dieser Bücher ausgenutzt. Damit kann es hier eine Bewertung von 10 geben.

Wahr und Falsch selbst werden als Antwortmöglichkeiten von mir mit 6 Punkten bewertet. Tabelle 5.4 wird nun hier in Tabelle 5.6 fortgesetzt:

Strategie	a	a'	l	l'	v	v'	s	s'	f
contains-b	9	0.97	2	0.97	6	0.59	7	0.75	0.74
intertwined	7	0.75	1	1.0	7	0.75	8	0.89	0.83
close-by	10	1.0	1	1.0	10	1.0	6	0.59	0.88

Tabelle 5.6: Initiale Bewertung der *close-by*-Strategie, dazu als Vergleichswert die *contains-b*-Strategie sowie die *intertwined*-Strategie

Frage	korrekte Antwort	Distraktor 1	Distraktor 2
In the context of Strategic Information Management, which one of the following triples belongs together the most?	Chief Information Officer - Department of Information management - Information Management Staff	Chief Information Officer - Strategic Gap - corporate strategy	Chief Information Officer - Ticket Evaluation - Project Monitoring
In the context of Strategic Information Management, which one of the following triples belongs together the most?	Information Management - IT Service Management - Operational Information Management	Information Management - Tactical Management of Information Systems - Information System	Information Management - methodology of Information Management - Operational Management of Information Systems
In the context of CIOX, which one of the following triples belongs together the most?	Chief Information Officer - Journey Execution - Project Monitoring	Chief Information Officer - Project Risk - Strategic HIS Planning	Chief Information Officer - IT General Ledger Account Planning - Project Portfolio
In the context of CIOX, which one of the following triples belongs together the most?	chief information officer - Project Presentation - project waitlist	chief information officer - Strategic Information Management Planning - Project Manager	chief information officer - Project Priorisation - Project Web Site

Tabelle 5.5: Beispiele für generierte Fragen der *intertwined*-Strategie

Auch hier ergibt sich mit einem Fragewert von 0.88 ein Indikator, dass die Strategie implementiert werden soll.

5.3.2.2 Implementierung

```

SELECT DISTINCT ?resa (STR(?al) AS ?als) (STR(?pal) AS ?pals) (
  ↳ STR(?aal) AS ?aals) ?resb (STR(?bl) AS ?bls) (STR(?pbl) AS
  ↳ ?pbls) (STR(?bbl) AS ?bbls)
WHERE {
# finde 2 ueber die closeMatch-Relation verbundene Klassen.
?a skos:closeMatch ?b .
# Diese haben dann jeweils einen anderen zusaetzlichen Knoten,
?a ?pa ?aa .
?b ?pb ?bb .

# welcher aber je von a bzw b verschieden ist
FILTER (?a < ?aa && ?b < ?bb)

# Filtere Ergebnisse heraus, die nur aufgrund der Metaebene der
  ↳ Ontologie in der Ergebnismenge sind
FILTER ( ?pa != <http://www.snik.eu/ontology/meta/roleComponent>
  ↳ && ?pa != <http://www.snik.eu/ontology/meta/
  ↳ entityTypeComponent> && ?pa != <http://www.snik.eu/
  ↳ ontology/meta/functionComponent>)
FILTER ( ?pb != <http://www.snik.eu/ontology/meta/roleComponent>
  ↳ && ?pb != <http://www.snik.eu/ontology/meta/
  ↳ entityTypeComponent> && ?pb != <http://www.snik.eu/
  ↳ ontology/meta/functionComponent>)
FILTER ( ?a != <http://www.snik.eu/ontology/meta/EntityType> && ?
  ↳ a != <http://www.snik.eu/ontology/meta/Role> && ?a != <
  ↳ http://www.snik.eu/ontology/meta/Function> && ?a != <http
  ↳ ://www.snik.eu/ontology/meta/subTopClass> && ?a != <http
  ↳ ://www.snik.eu/ontology/meta/Top>)
FILTER ( ?aa != <http://www.snik.eu/ontology/meta/EntityType> &&
  ↳ ?aa != <http://www.snik.eu/ontology/meta/Role> && ?aa != <
  ↳ http://www.snik.eu/ontology/meta/Function> && ?aa != <http
  ↳ ://www.snik.eu/ontology/meta/subTopClass> && ?aa != <http
  ↳ ://www.snik.eu/ontology/meta/Top>)
FILTER ( ?b != <http://www.snik.eu/ontology/meta/EntityType> && ?
  ↳ b != <http://www.snik.eu/ontology/meta/Role> && ?b != <
  ↳ http://www.snik.eu/ontology/meta/Function> && ?b != <http
  ↳ ://www.snik.eu/ontology/meta/subTopClass> && ?b != <http
  ↳ ://www.snik.eu/ontology/meta/Top>)
FILTER ( ?bb != <http://www.snik.eu/ontology/meta/EntityType> &&
  ↳ ?bb != <http://www.snik.eu/ontology/meta/Role> && ?bb != <
  ↳ http://www.snik.eu/ontology/meta/Function> && ?bb != <http
  ↳ ://www.snik.eu/ontology/meta/subTopClass> && ?bb != <http
  ↳ ://www.snik.eu/ontology/meta/Top>)

# Herkunft (Ontologie) von a und b bestimmen
?a ^<http://open.vocab.org/terms/defines> ?domaina .

```

```

?b ^<http://open.vocab.org/terms/defines> ?domainb .

# Ontologie in natuerliche Sprache umwandeln
BIND ( IF(?domaina = <http://www.snik.eu/ontology/he>, "General
  ↪ Information Management", IF(?domaina = <http://www.snik.eu
  ↪ /ontology/ob>, "Tactical Information Management", IF(?
  ↪ domaina = <http://www.snik.eu/ontology/ciox>, "CIOX", "
  ↪ Strategic Information Management")) AS ?resa)
BIND ( IF(?domainb = <http://www.snik.eu/ontology/he>, "General
  ↪ Information Management", IF(?domainb = <http://www.snik.eu
  ↪ /ontology/ob>, "Tactical Information Management", IF(?
  ↪ domainb = <http://www.snik.eu/ontology/ciox>, "CIOX", "
  ↪ Strategic Information Management")) AS ?resb)

# nur englische Ergebnisse
?a rdfs:label ?al .
FILTER(langMatches(LANG(?al), "en"))
?aa rdfs:label ?aal .
FILTER(langMatches(LANG(?aal), "en"))
?b rdfs:label ?bl .
FILTER(langMatches(LANG(?bl), "en"))
?bb rdfs:label ?bbl .
FILTER(langMatches(LANG(?bbl), "en"))
?pa rdfs:label ?pal .
FILTER(langMatches(LANG(?pal), "en"))
?pb rdfs:label ?pbl .
FILTER(langMatches(LANG(?pbl), "en"))
}

```

Alg. 5.5: Disktraktorengenerierung der *close-by*-Strategie

Da die Erzeugung der Fragen in JavaScript-Form sehr ähnlich abläuft wie in Algorithmus 5.4, ist der Algorithmus dafür in Algorithmus C.1 zu finden.

5.3.2.3 Ergebnis

Die *close-by*-Strategie ergab 7449 verschiedene Fragen, davon 662, welche mit „Wahr“ beantwortet werden müssen. Zwei zufällig ausgesuchte Beispiele sind in Tabelle 5.7 aufgeführt. Die zwei möglichen Antworten sind bei dieser Strategie „Wahr“ und „Falsch“.

Frage	korrekt
In the Tactical Information Management, the Project Execution uses the Resource, while in the Strategic Information Management, the Project Execution uses the Resource.	True
In the Strategic Information Management, the Project Management Board is responsible for the Project Monitoring, while in the Tactical Information Management, the Project Steering Committee uses the Project .	False

Tabelle 5.7: Beispiele für generierte Fragen der *close-by*-Strategie

5.3.3 *occurrence-Strategie*

Hier handelt es sich um eine Spezialisierung von *close-by*, bei der entschieden werden soll, ob ein Begriff in keinem Buch, in Ammenwerth u. a., 2014, in Winter u. a., 2011 oder in beiden Büchern auftritt. Eine mögliche Frage ist hier: „In which books does the term ‚Chief Information Officer‘ occur?“. Auf diese Frage kann dann noch die Definition des Begriffes folgen, um noch mehr Kontext zu liefern.

5.3.3.1 *Initiale Einschätzung*

Die Frage ist zwar knapp, kann aber durch zusätzliche Informationen aufgewertet werden, wie oben beschrieben. Damit kann man hier eine Punktzahl von 8 Punkten geben. Die Antworten können nicht kürzer gefasst werden, sind also mit einem Punkt zu bewerten. Hinsichtlich der Vernetzung ist diese Strategie nicht so stark ausgeprägt wie die *close-by*-Strategie, allerdings wird immer noch ein Fokus auf die Vernetzung gelegt. Damit wird es hier mit 8 Punkten bewertet. Die Antwortmöglichkeiten sind alle ähnlich, wenn auch immer gleich. Im Gegensatz zu bisherigen Strategien werden die Antworten aber selbst nicht Knoten im Graphen sein, daher kann hier nur eine angepasste Bewertung mit 6 Punkten erfolgen. Dies ist knapp besser als die Hälfte der Punkte, was mir angemessen erscheint. Damit ergibt sich folgendes Bild:

Strategie	a	a'	l	l'	v	v'	s	s'	f
contains-b	9	0.97	2	0.97	6	0.59	7	0.75	0.74
intertwined	7	0.75	1	1.0	7	0.75	8	0.89	0.83
close-by	10	1.0	1	1.0	10	1.0	6	0.59	0.88
occurrence	8	0.89	1	1.0	8	0.89	6	0.59	0.82

Tabelle 5.8: Initiale Bewertung der *occurrence*-Strategie, dazu als Vergleichswert die *contains-b*-Strategie sowie die *intertwined*- und *close-by*-Strategie

Aus dieser Tabelle ergibt sich das klar, dass der kritische Fragewert von 0.74 auch hier überschritten wird und sich so die Implementierung lohnt.

5.3.3.2 Implementierung

Aufgrund der Lehren aus Abschnitt 5.3.1 wurde hier die Implementierung wieder auf mehrere Queries aufgeteilt, welche dann wieder mittels Python vereinigt wurden, wie am Beispiel des Taktischen IMs in Algorithmus 5.6 zu sehen. Die restlichen Queries sind im Anhang unter Algorithmus B.1, Algorithmus B.2 und Algorithmus B.3 zu finden.

```
SELECT DISTINCT (STR(?al) AS ?label)
WHERE {
# Term A ist in in der Ontologie OB (Taktisches IM)
?a a owl:Class .
?a ^<http://open.vocab.org/terms/defines> <http://www.snik.eu/ontology/ob> .
?a rdfs:label ?al .
FILTER( langMatches(LANG(?al), "en") )

# A hat keine closeMatches zu anderen Ontologien
FILTER NOT EXISTS {?a skos:closeMatch ?b}
?b ^<http://open.vocab.org/terms/defines> ?domain2 .
FILTER (?domain2 != <http://www.snik.eu/ontology/ob>)
}
```

Alg. 5.6: *occurrence*-Strategie: Taktisches IM

Diese generierten Begriffe, aufgeteilt in 4 Dateien, werden dann mittels eines Pythonskripts zusammengeführt, welches ähnlich wie Algorithmus 5.4 funktioniert.

```
import csv

class csv_reader:
    def __init__(self, file_path):
        self.file_path = file_path
```

```

        self.line_list = []

    def readfile(self):
        with open(self.file_path, 'r') as file:
            csvFile = csv.reader(file)
            for lines in csvFile:
                #print(lines)
                self.line_list.append(lines[0])

    def get_line_list(self):
        return self.line_list

class question_generator:
    def __init__(self, path_list):
        self.extractors = []
        for i in range(0,4):
            self.extractors.append(csv_reader(path_list[i]))
        self.answers = []
        self.answer_list = []

    def controller(self):
        answer_string = ""
        for i in range(0,4):
            self.extractors[i].readfile()
            temp_list = self.extractors[i].get_line_list()
            print(temp_list)
            self.generate_js_answers(temp_list, i)
        for k in self.answer_list:
            answer_string += k
        return answer_string

    def generate_js_answers(self, term_list, index):
        print("Writing index: {}".format(index))
        string_template = "{ question: <span> In which contexts
↪ does the term \"#{1}\" occur? </span>, answers: [<span>
↪ Tactical Information Management</span>,<span>Strategic
↪ Information Management</span>,<span>Both Contexts</span>,<
↪ span>Neither</span>,<span>], correct: #2 }\n"
        # #1 -- term
        # #2 -- index of correct answer: 0 = ob, 1 = bb, 2 = both
↪ , 3 = neither
        answer = ""
        answer_list = []
        for term in term_list:
            answer = string_template.replace("#1", term)
            answer = answer.replace("#2", str(index))
            self.answer_list.append(answer)

class file_handler:
    def __init__(self, file_path):

```

```

        self.file_path = file_path

    def write_txt(self, stringel, appending):
        option = "w"
        if appending:
            option = "a"
        f = open(self.file_path, option)
        f.write(stringel)
        f.close()

    def write_list(self, listi, appending):
        option = "w"
        if appending:
            option = "a"
        f = open(self.file_path, option)
        for item in listi:
            f.writelines(item + '\n')
        f.close()

    def write_tuple_list(self, tuple_list, appending):
        option = "w"
        if appending:
            option = "a"
        f = open(self.file_path, option)
        for tuples in tuple_list:
            sol = ""
            for item in tuples:
                sol += str(item) + " "
            f.writelines(sol + '\n')
        f.close()

qg = question_generator(['../SPARQL/occurence/ob.csv', '../SPARQL/occurence/bb.csv', '../SPARQL/occurence/both.csv',
    ↪ , '../SPARQL/occurence/neither.csv'])
filer = file_handler('../SPARQL/results/occurence.js')
filer.write_txt(qg.controller(), False)

```

Alg. 5.7: Python occurence Programmcode

5.3.3.3 Ergebnis

Die *occurence*-Strategie ergab 2735 verschiedene Fragen. Vier zufällig ausgesuchte Beispiele sind in Tabelle 5.9 aufgeführt. Die vier Antwortmöglichkeiten sind bei dieser Strategie „Taktisches Informationsmanagement“, „Strategisches Informationsmanagement“, „Beides“ (Both Contexts) und „Keines von Beidem“ (Neither).

Frage	korrekt
In which contexts does the term „Automated Observation“ occur?	Tactical Information Management
In which contexts does the term „Health Insurance Company“ occur?	Strategic Information Management
In which contexts does the term „System Analysis“ occur?	Both Contexts
In which contexts does the term „System Administrator“ occur?	Neither

Tabelle 5.9: Beispiele für generierte Fragen der *occurence*-Strategie

5.4 EVALUATION

5.4.1 *Aufbau der Evaluation*

Der Fragebogen ist in drei Teile unterteilt. Der erste Part erfragt die allgemeine Erwartungshaltung gegenüber Multiple-Choice-Tests hinsichtlich Gütekriterien und Komplexität und legt so die Grundlagen, um kritisch die nachfolgenden Fragen zu beurteilen.

Im zweiten Teil sollen die Strategien hinsichtlich der Güte und Komplexität beurteilt werden. Dafür sollen Bewertungen vergeben werden, sowie ein optionales Kommentar bezogen auf die einzelnen Strategien. Um die Objektivität der Testpersonen sicherzustellen, werden die Fragestrategien gemischt und so als ein einheitliches Quiz präsentiert. Dafür wird das bereits existierende Werkzeug *SNIK-Quiz*⁵ angepasst, um keine Zeitlimitierung zu enthalten. Die Auswahl der Fragen der je fünf Fragen pro Strategie wurde einmalig mittels eines einfachen Pythonskriptes gelöst, welches randomisiert Fragen jeder Strategie wählt. Diese werden dann in das Quiz mit den oben beschriebenen Parametern eingefügt, um sodann von den Experten zur Evaluation benutzt zu werden. Der Fragebogen ist im Anhang unter Appendix A.1 einsehbar.

EVALUIERTE PERSONEN Die in diesem Interviewprozess befragten Experten sind Frau Dr. Birgit Schneider, Frau Dr. Jahn und Frau Perbix, welche zum Zeitpunkt der Abgabe alle am IMISE angestellt sind.

⁵ <https://github.com/snikproject/quiz>

5.4.2 *Evaluationsergebnis*5.4.2.1 *Erwartungen an Fragen*

Die Rohdaten, auf denen die Auswertung basiert, sind in Appendix A.2.1, Appendix A.2.2 und Appendix A.2.3 zu finden.

Die Befragten gaben an, dass gute Fragen sich dadurch auszeichnen, dass die Antworten möglichst homogen, sowohl in der Länge selbst wie auch in der semantischen Äquivalenz sind. Auch sollten verschiedene Kategorien von Fragen auftreten und es sollten möglichst keine großen thematischen Sprünge zwischen den Fragen stattfinden.

Komplexe Fragen sollten sich ihrer Meinung nach durch Vernetzung, und nicht durch sprachliche Komplexität auszeichnen. Auch eine hohe semantische Äquivalenz der Antwortmöglichkeiten wurde hier genannt.

5.4.2.2 *Einschätzung der Strategien*

Strategie	Punkte Interview 1	Punkte Interview 2	Punkte Interview 3
label-definition	3	4	3
definition-b	4	6	3
contains-b	5-6	7	6
intertwined	7	7-8	8
close-by	8	7-8	8
occurence	7	5	7

Tabelle 5.10: Bewertung der Komplexität der Strategien, Skala von 1 bis 10, wobei 10 die höchste Komplexität darstellt

KOMPLEXITÄT

Bereits vorhandene Strategien Wie in Tabelle 5.10 zu sehen ist, schneidet, ähnlich wie in Tabelle 5.1, die *label-definition*-Strategie hinsichtlich der Komplexität mit durchschnittlich 3,33 Punkten am schlechtesten ab. Allerdings schneidet sie relativ zu den anderen beiden Strategien besser ab als in der initialen Einschätzung.

Danach kommt, genau wie in Tabelle 5.1, die *definition-b*-Strategie, welche mit durchschnittlichen 4,33 Punkten fast die Hälfte der möglichen Komplexitätspunkte erhält. Ähnlich wie in der initialen Einschätzung wird auch die *contains-b*-Strategie bewertet, welche mit im Schnitt 6 bis 6,33 Punkten die beste der existierenden Strategien hinsichtlich der Komplexität darstellt.

Neue Strategien Hier schneidet die *occurence*-Strategie mit 6,33 Punkten am schlechtesten ab, allerdings schneidet sie in Hinsicht auf die Komplexität etwas besser ab als die *contains-b*-Strategie, vergleiche dazu Abschnitt 5.3.3.1. Als richtig erweist sich die Einordnung, welche aussagt, dass dies die *schlechteste* der neuen Strategien sei. Da, wie in A.2.1 zu sehen ist, kein umfassendes Wissen über die behandelten Bücher vorausgesetzt werden kann, wurde diese Kategorie als zu schwer empfunden. Allerdings wurde in Appendix A.2.2 sowie in Appendix A.2.3 angemerkt, dass diese Art zu fragen prinzipiell gut sei.

Die *intertwined*-Strategie belegt insgesamt mit 7,33 bis 7,66 Punkten von zehn Möglichen den zweiten Platz hinsichtlich der Komplexität, wobei hier besonders positiv die sprachlich einfache Frage gekoppelt mit den kurzen Antwortmöglichkeiten hervorgehoben wurde. Allerdings wurde auch in Appendix A.2.1 angemerkt, dass der Einstieg schwer sei.

Die *close-by*-Strategie belegt mit 7,66 bis 8 Punkten den ersten Platz hinsichtlich der Komplexität. Allerdings wird dies hier, siehe dazu Appendix A.2.2 und Appendix A.2.3, auch durch sprachliche Komplexität erreicht. Auch Raten war hier gut möglich.

GENERELLE EINSCHÄTZUNG

Bereits vorhandene Strategien Hinsichtlich der *label-definition*-Strategie war die Einschätzung zwar positiv, allerdings wurde die hohe Varianz der Antwortlänge (Appendix A.2.1) sowie der Umstand kritisiert, dass die Antwort teils wörtlich in der Frage enthalten war. Sie wurde als gut eingeschätzt um Grundlagenwissen abzufragen. Dies gilt ebenso für die *definition-b*-Strategie, welche auch als etwas komplizierter als die *label-definition*-Strategie eingeschätzt wurde. Allerdings konnte es hier vorkommen, dass bei Distraktoren Dopplungen auftraten, weil bestimmte Beziehungen innerhalb der Ontologie nicht gut genug beachtet wurden.

Als am besten bewertete der schon vorhandenen Strategien wurde die *contains-b*-Strategie besonders gelobt, da hier Fragen mittels Zusammenhängen und deren Verständnis beantwortet werden müssen. Allerdings kann hier teilweise das Ausschlussverfahren angewandt werden.

Neue Strategien Die *occurence*-Strategie weckte besonders in Appendix A.2.1 wenig Begeisterung. Allerdings wurde sie in Appendix A.2.2 als „gut“ bezeichnet. In Appendix A.2.3 wurde noch gesagt, dass dieser Ansatz ausgenutzt werden sollte, um die Strategie umzugestalten und so nützlicher für die Lehre zu machen.

Die *intertwined*-Strategie wurde sehr positiv aufgenommen, da sie aus Sicht der Experten über keine großen Schwachstellen verfügt. In

Appendix A.2.3 wurde hier allerdings festgestellt, dass Teile des Ergebnisraumes, besonders die Ontologie CIOX⁶, nicht für Studierende geeignet sind.

Die objektiv schlechteste Bewertung durch die Experten hinsichtlich der generellen Gefühle bezüglich der Strategien hat die *close-by*-Strategie erhalten, bei welcher kritisiert wurde, dass die gesamte Schwierigkeit der Frage auf sprachlicher Ebene stattfindet, nicht auf fachlicher Ebene.

5.4.2.3 *Einschätzung der initialen Einschätzung*

Die Kriterien der initialen Einschätzung wurden in diesen Interviews als letztes abgefragt und erhielten übereinstimmend folgende Bewertung:

Die Annahme, dass Fragen, welche mehr Kontext bieten, besser wären, wurde zurückgewiesen und erklärt, dass kürzere Fragen meist besser verständlich seien. Das Merkmal der Länge der Antwortmöglichkeiten wurde teilweise bestätigt, allerdings wurde in Appendix A.2.1 die Anmerkung gemacht, dass Antworten lieber gleich lang sein sollten, oder, in Appendix A.2.3, bei langen Antworten weniger Auswahlmöglichkeiten gegeben werden sollten. Die beiden letzten Merkmale, Vernetzung und semantische Ähnlichkeit der Antwortmöglichkeiten, wurden beide als „wichtig“ (Appendix A.2.2) bezeichnet.

⁶ Ein spezielles CIO-Interview

DISKUSSION

6.1 ZUR INITIALEN EINSCHÄTZUNG

Die initiale Einschätzung (Abschnitt 5.1) erwies sich als ungenau. Die Komplexität konnte nur unzureichend mittels der vier verwendeten Kriterien abgebildet werden, allerdings hat die initiale Einschätzung ihren Zweck erfüllt, welcher war, ein Indikator zur Komplexität zu sein.

Dabei sind Vernetzung und Semantische Ähnlichkeit die wichtigsten Merkmale, wobei auch die syntaktische Ähnlichkeit mit beachtet werden kann, um bessere Einschätzungen zu liefern.

6.2 ERSTELLTE STRATEGIEN UND EVALUATION

6.2.1 *Bereits vorhandene Strategien*

In den Interviews wurde erwähnt, dass es in allen bereits existierenden Strategien zu doppelten Distraktoren kam, welche wohl durch `closeMatch`-Relationen hervorgerufen wurden. Dies könnte man in der SPARQL-Query noch ausschließen und so die Strategie noch verbessern. Auch die Antworten auf eine bestimmte Subklasse von `Top` zu beschränken, also entweder `Role`, `Function` oder `EntityType` würde diesen Strategien sehr helfen, schwierigere Distraktoren zu generieren.

6.2.2 *intertwined-Strategie*

Interessanterweise taucht die von der *intertwined*-Strategie geforderte Struktur nur im Kontext von CIOX und dem Strategischen Informationsmanagement auf, nicht im Taktischen Informationsmanagement. In der Evaluation wurde diese Strategie positiv bewertet. Dementsprechend kann diese Strategie durchaus als erfolgreich gewertet werden.

Ein weiterer Ansatzpunkt könnte hier auch sein, um etwas Schwierigkeit herauszunehmen, drei Tripel anzugeben, von welchen eines nicht semantisch zusammengehört.

6.2.3 *close-by-Strategie*

Die *close-by*-Strategie wurde zurecht als sprachlich komplex bezeichnet. Die Ansichten dazu sind sinnvoll und es sollte durchaus eine

Bearbeitung der Strategie erfolgen, sollte diese aktiv genutzt werden sollen.

6.2.4 *occurence-Strategie*

Hier kann man sagen, dass die Strategie prinzipiell gut funktioniert, auch die Experten sahen dies so. Allerdings könnte man noch Definitionen angeben, wenn die Länge der erfragten Worte eine bestimmte Länge unterschreitet, um so Abkürzungen aufzuklären.

Man kann diese Art, Fragen zu entwickeln, auch weiterhin verwenden und sollte dies auch tun. Damit können auch andere Fragetypen zum Beispiel durch Eingrenzung des Fachgebietes unterstützt werden.

Diese Strategie kann, wie in Interviews gezeigt, auch dazu eingesetzt werden, die Ontologien selbst zu beurteilen, indem eventuelle fehlende Verbindungen hinzugefügt werden oder Begriffe auch in anderen Ontologien closeMatch-Relationen bekommen, da diese dort auch verortet werden können.

6.2.5 *Evaluation*

Von Vorteil wäre hier gewesen, eine quantitative Analyse durchführen zu können, welche statistische Relevanz besitzt. Allerdings war dies hier nicht möglich, da nicht genügend Studierende in diesem Fachbereich gibt. Die Interviews mit den Experten ergaben aber sehr hilfreiche Hinweise zur Verbesserung der jetzt existierenden sechs Strategien.

Auch gut wäre gewesen, wenn die Evaluation bzw. das Gespräch mit den Lehrverantwortlichen des IMISE eher stattgefunden hätte, da so gute Ansätze besser hätten ausgenutzt werden können.

6.3 SNIK

6.3.1 *Zur SNIK-Ontologie*

Im Rahmen der Interviews kamen auch verschiedentlich Verbesserungsvorschläge hinsichtlich der Ontologie. So wurde beispielsweise gesagt, dass man Grundlagenwissen auch in der Ontologie zum Beispiel mittels Wahrheitswerten implementieren könnte und so noch besser Strategien auf Lehrveranstaltungen ausgerichtet werden können.

6.3.2 *Zum SNIK-Quiz*

Während der Arbeit mit dem SNIK-Quiz ergaben sich verschiedentlich Verbesserungsvorschläge. Diese sollen hier kurz aufgeführt werden,

jeweils mit einer kurzen Begründung, warum diese Maßnahmen zu einer Mehrung des didaktischen Wertes des SNIK-Quizes beitragen würde. Die Vorschläge sind in Tabelle 6.1 zu finden.

Nr.	Vorschlag
1	Um bestimmte Mengen von Fragen pro Strategie nutzen zu können, könnte am Anfang des Quizes pro Strategie die Anzahl der aus dem Fragenarray entnommenen Fragen definiert werden. So könnten leichte und schwere Strategien angemessen verteilt getestet werden.
2	Hinzufügen einer Navigationsleiste zur Möglichkeit des Rücksprungs zu schwierigen Fragen
3	Möglichkeit einrichten, die falsch beantworteten Fragen nach dem Abschluss, nicht während des Quizes, noch einmal durchzugehen und nochmals zu beantworten. Dafür wäre es notwendig, sich die falsch beantworteten Fragen abzuspeichern und so könnte (nach größeren Modifikationen) auch eine persönliche Analyse der Schwächen des Getesteten möglich werden.
4	Umbau des Quizes, um nicht nur in Echtzeit SPARQL-Fragen zu erzeugen sondern aus allen Fragen live auszuwählen, um die neuen Strategien besser umsetzen zu können, besonders die <i>intertwined</i> -Strategie, welche sehr rechenaufwendig ist

Tabelle 6.1: Vorschläge zur Verbesserung des SNIK-Quizes

Eine weitere problematische Erkenntnis ist, dass das SNIK-Quiz sich mittels eines einfachen Programmes (siehe Algorithmus C.2) automatisiert lösen lässt. Damit scheidet es für den zukünftigen Einsatz in Prüfungen aus, da es als Open Source verfügbar ist. Kritisch ist hierbei allerdings nicht das Fundament des Programmes, sondern, dass auch die gestellten Fragen und die Antworten öffentlich einsehbar auf github verfügbar sind. So können in Echtzeit die richtigen Fragen aus der Datei ausgelesen werden und die korrekte Lösung übermittelt werden. Eine Möglichkeit, dies zu unterbinden, ist, dass die möglichen Fragen selbst nur als SPARQL-Queries vorliegen, welche dann auf entsprechender Hardware ausgeführt werden müssten, um die Wissensbasis zum Quiz zu erhalten. Auch dies birgt natürlich Risiken, allerdings ist der Aufwand unverhältnismäßig viel höher als zum jetzigen Zeitpunkt, an dem die Fragen mit der korrekten Antwort offen vorliegen.

ZUSAMMENFASSUNG

In dieser Arbeit sollte die fehlende Komplexität der bisher bekannten Fragestrategien des SNIK-Quizes dadurch verbessert werden, dass komplexere Fragestrategien neu entwickelt werden. Dafür musste zunächst eine Voreinschätzung mittels Merkmalen aus der Literaturrecherche vorgenommen werden, um ausreichend komplexe Strategien zu gewährleisten. Danach wurden drei neue Strategien auf Grundlage von einer graphbasierten Datenbank und SPARQL entwickelt, welche danach noch von drei Experten mit den bereits bestehenden Strategien verglichen und als ausreichend komplex eingeschätzt wurden, auch wenn sie noch verbessert werden könnten. Die hier entwickelten Strategien sind die *intertwined*-, die *close-by*- und die *occurence*-Strategie.

Bei der *intertwined*-Strategie werden stark verbundene Tripel gesucht und mit weniger stark verbundenen Tripeln gemischt, um so eine Frage nach dem am meisten verbundenen Tripel zu fragen.

Die *close-by*-Strategie nutzt die *closeMatch*-Relation aus, um einen Vergleich zwischen zwei Gebieten des Informationsmanagements zu ziehen. Dieser Abgleich soll dann mit wahr oder falsch auf Richtigkeit überprüft werden.

Als letztes wurde die *occurence*-Strategie entwickelt, welche überprüft, in welchem Gebiet des Informationsmanagements im Krankenhaus ein Begriff auftritt.

Im Ergebnis lässt sich sagen, dass drei komplexe Strategien entwickelt und getestet wurden. Auch die Kriterien der initialen Einschätzung wurden durch die Experten überprüft und für ausreichend befunden, bis auf ein Kriterium.

LITERATUR

- Allemang, Dean und Jim Hendler (2011). „Chapter 12 - Counting and sets in OWL“. In: *Semantic Web for the Working Ontologist (Second Edition)*. Hrsg. von Dean Allemang und Jim Hendler. Second Edition. Boston: Morgan Kaufmann, S. 249–278. ISBN: 978-0-12-385965-5. DOI: <https://doi.org/10.1016/B978-0-12-385965-5.10012-3>. URL: <https://www.sciencedirect.com/science/article/pii/B9780123859655100123>.
- Ammenwerth, Elske, Reinhold Haux, Petra Knaup-Gregori und Alfred Winter (2014). *IT-Projektmanagement im Gesundheitswesen*. 2. Aufl. Stuttgart, Germany: Schattauer. ISBN: 9783794530717.
- Beckett, David und Tim Berners-Lee (2011). Besucht am 07.09.2021. URL: <https://www.w3.org/TeamSubmission/turtle/>.
- Berners-Lee, Tim (1994). URL: <https://www.w3.org/Addressing/URL/url-spec.txt>.
- Berners-Lee, Tim, Roy Fielding und Larry Masinter (2005). Besucht am 31.08.2021. URL: <https://datatracker.ietf.org/doc/html/rfc3986>.
- Berners-Lee, Tim, Roy Fielding und Larry Masinter (2009). Besucht am 01.09.2021. URL: <https://www.w3.org/DesignIssues/LinkedData.html>.
- Brickley, Dan und R.V. Guha (2014). Besucht am 02.09.2021. URL: <https://www.w3.org/TR/rdf-schema/>.
- Conejo, Ricardo, Beatriz Barros und Manuel F Bertoa (2018). „Measuring the quality of assessment using questions generated from the semantic web“. In: *International Conference on Artificial Intelligence in Education*. Springer, S. 57–69.
- Dittler und Kreidl (2021). *Wie Corona die Hochschullehre verändert Erfahrungen und Gedanken aus der Krise zum zukünftigen Einsatz von eLearning*. Springer. ISBN: 9783658326098. DOI: <https://doi.org/10.25656/01:9169>.
- Duerst, M. und M. Suignard (2005). Besucht am 06.09.2021. URL: <https://www.ietf.org/rfc/rfc3987.txt>.
- Ellinger, Dorothea (2017). „Multiple-Choice-Prüfungen“. In: *Kompetent Prüfungen gestalten*. Hrsg. von Julia Gerick, Angela Sommer und Gerno Zimmermann. utb, S. 137–139. ISBN: 9783825248406.
- Fischer, Martin, Philipp Grollmann, Bibhuti Roy und Nikolaus Steffen (2003). *E-Learning in der Berufsbildungspraxis: Stand, Probleme, Perspektiven*. 6. Bremen: Institut Technik und Bildung.
- GMDS (2021). *Definition Medizinische Informatik*. URL: <https://www.gmds.de/de/aktivitaeten/medizinische-informatik/>.

- Graham, Klyne und Jeremy J Carroll (2014). Besucht am 02.09.2021. URL: <https://www.w3.org/TR/rdf-concepts/>.
- Guarino, Nicola, Daniel Oberle und Steffen Staab (2009). „What is an ontology?“ In: *Handbook on ontologies*. Springer, S. 1–17.
- Harris, Steve und Andy Seaborne (2013). Besucht am 06.09.2021. URL: <https://www.w3.org/TR/sparql11-query/>.
- Hayes, Patrick J. und Peter F. Patel-Schneider (2014). Besucht am 03.09.2021. URL: <https://www.w3.org/TR/2014/REC-rdf11-mt-20140225/#blank-nodes>.
- Heilman, Michael (2011). „Automatic Factual Question Generation from Text“. Diss. DOI: 10.1.1.435.7619.
- Herman, Ivan (2008). Besucht am 17.10.2021. URL: https://www.w3.org/blog/SW/2008/01/15/sparql_is_a_recommendation/.
- Hirtl, Isabella (2010). „Lernen und Lerntransfer - Ein Experiment im Rechnungswesenunterricht“. Diplomarbeit. Wirtschaftsuniversität Wien, S. 19–21.
- Jahn, Franziska, Konrad Höffner, Birgit Schneider, Anna Lörke, Thomas Pause, Elske Ammenwerth und Alfred Winter (2019). „The SNIK Graph: Visualization of a Medical Informatics Ontology“. In: *MEDINFO 2019: Health and Wellbeing e-Networks for All*. IOS Press, S. 1941–1942.
- Jahn, Franziska, Michael Schaaf, Barbara Paech und Alfred Winter (2014). „Ein Semantisches Netz des Informationsmanagements im Krankenhaus“. In: *Informatik 2014*. Hrsg. von E. Plödereder, L. Grunzke, E. Schneider und D. Ull. Lecture Notes in Informatics. Bonn: Gesellschaft für Informatik e.V., S. 1491–1498.
- Jeremias, Xenia V. und Christian Rabe (2021). URL: <https://www.th-wildau.de/hochschule/zentrale-einrichtungen/zentrum-fuer-qualitaetsentwicklung/e-learning/e-assessment/leitfaden-e-assessment/>.
- McGuinness, Deborah L. und Frank van Harmelen (2009). Besuch am 08.09.2021. URL: <https://www.w3.org/TR/owl-features/>.
- Meinel, Cristoph und Martin Mundhenk (2015). *Mathematische Grundlagen der Informatik*. Springer, S. 223–234. ISBN: 978-3-658-09885-8.
- Niegemann, Domagk, Hessel, Hein, Hupfer und Zobel (2008). *Kompendium Multimediales Lernen*. Springer, S. 290–297. ISBN: 978-3-540-37225-7.
- Nies, Tom de u. a. (2016). „ESWC“. In: S. 69–84. DOI: https://doi.org/10.1007/978-3-319-34129-3_5.
- Vega-Gorgojo, Guillermo (2019). „Clover Quiz: a trivia game powered by DBpedia“. In: *Semantic Web* 10(4). Hrsg. von Jens Lehmann, S. 779–793.
- Venners, Bill (2013). Besucht am 08.09.2021. URL: <https://www.artima.com/articles/the-making-of-python>.

- Wächtler, Max Niclas (2020). „Automatische Erstellung von Quizfragen aus einer Ontologie von Krankenhausinformationssystemen“. BeLL. Wilhelm-Oswald-Gymnasium Leipzig.
- Winter, Alfred, Reinhold Haux, Elske Ammenwerth, Birgit Brigel, Nils Hellrung und Franziska Jahn (2011). *Health Information Systems: Architectures and Strategies*. Health Informatics. Springer London. ISBN: 9781849964418. URL: <https://books.google.de/books?id=RzvmrgwCWncC>.

Teil I

APPENDIX



EVALUATION

A.1 FRAGEBOGEN

1. Nennen Sie bitte kurz Ihre Erwartungen an gute Fragen in einem Multiple-Choice-Quiz.
2. Nennen Sie bitte kurz Ihre Erwartungen an komplexe Fragen in einem Multiple-Choice-Quiz.
3. Im Folgenden werden Sie gebeten werden, verschiedene Fragen zu beantworten, je 5 pro existierender Strategie. Bitte ordnen Sie diesen Strategien hinsichtlich der empfundenen Komplexität eine Zahl zwischen eins und zehn zu und äußern Sie dazu, wenn möglich, einen Kommentar.

Strategie	Bewertung	Kommentar
label-definition		
definition		
contains		
intertwined		
close-by		
occurence		

4. Bitte beurteilen Sie außerdem die Strategien hinsichtlich Ihrer in 1. und 2. aufgeschriebenen Kriterien.

Strategie	Kommentar
label-definition	
definition	
contains	
intertwined	
close-by	
occurence	

5. Bitte beurteilen Sie rückblickend die folgenden, in der Arbeit verwendeten Merkmale zur Beurteilung von Fragestrategien:

Strategie	Kommentar
Ausführlichkeit der Frage	
Länge der Antwortmöglichkeiten	
Vernetzung der Inhalte	
Semantische Ähnlichkeit der Antwortmöglichkeiten	

A.2 INTERVIEW-ERGEBNISSE

Die Interviews wurden als formloses Gespräch geführt. Die hier gemachten Notizen, sind, wenn nicht durch Anführungsstriche gekennzeichnet, nur meine Betrachtungsweise der Kommentare, sollten aber den Geist der Antwort widerspiegeln.

A.2.1 Interviewpartnerin 1

1) keine Antwortmöglichkeiten, die semantische Äquivalenz zur Frage herstellen

keine zu großen Längenschwankungen in den Antworten

2)

Wenn man nicht sofort sehen kann, was die richtige Antwort ist.

Ausschlussverfahren nicht leicht anwendbar.

3)

ld: 3 , hohe Varianz der Länge der Antworten, Auswahlkriterium Länge der Antworten, Antwort teils in Frage enthalten

d: 4, Dopplungen der Distraktoren, bei besseren Antwortmöglichkeiten gute Fragen, wenn auch wenig komplex (Auswendiglernen)

c: 5-6, wenig komplexe Antwortmöglichkeiten aufgrund von Ausschlussverfahren, schön kurz, auf Wissensbasis -> gute Frage, selten Dopplungen

i: 7 , schwerer Einstieg, positive Bewertung aufgrund von simultaner Länge, keine offensichtlichen Ausschlusskriterien, kein komplexer Satzbau, positive Rückmeldung, „mehr Nachdenken als Lesen“

cb: 8 , komplizierter Satzbau, Komplexität aufgrund von Möglichkeiten des Fehlersuchens, Distraktorfragen erkennbar an wenig sinnvollen Verben im zweiten Teilsatz, Raten möglich

o: 7 , ich mag es überhaupt nicht. Kein Wissen über Hintergrund -> Erhöhung der Schwierigkeit der Fragen, kein umfassendes Wissen über beide Bücher

4) siehe 3.

gut : i, c

mittel: cb, d

schlecht: o, ld

5)

1. nö, weil desto mehr lesen, kompliziertere Grammatik, zu lang schlechter. Kürzer besser -> kein gutes Qualitätskriterium

2. nicht zwangsweise, Länge der Antwortmöglichkeiten sollte lieber jeweils ca. gleich sein

3. super, kein Auswendiglernen, gelerntes kann angewandt werden

4. gutes Merkmal, Ausschlussverfahren erschwert

A.2.2 Interviewpartnerin 2

1)

nicht trivial: keine offensichtlichen Lösungen

Grundlagenwissen und zum anderen auch Detailwissen -> verschiedenen Kategorien

bei mehreren Fragen: keine großen thematischen Sprünge

2)

Komplexität nicht im Sprachverständnis (keine doppelten Verneinungen), sondern im Inhalt

nicht nur Auswendiglernen von Definitionen, sondern selbst Zusammenhänge herstellen

Anmerkung: nicht einfach in MCQ

3)

ld: 4; teilweise „seltsame“ Formulierungen

d: 6; nett, teils offensichtliche Lösungen, teils zu lange Lösungen („mühsam“)

c: 7; teilweise Antworten doppelt, „nicht fair“

Anmerkung: Open World -> Problem

i: 7-8; schön kurze Antworten, „sehr schön“

Anmerkung: CIOX wird nicht gelehrt

cb: 7-8 ; zuviel Sprachkompetenz, schwierig aufgrund des langen Satzes, schwierig durch Nachdenken

o: 5; „an sich [...] gute Art zu fragen“

4)

ld: gut um Grundlagenwissen abzufragen

d: gut um Grundlagenwissen abzufragen, etwas schwieriger als labeldefinition

Anmerkung: Grundlagenwissen evtl in Ontologie kennzeichnen

c: „gefällt mir auch sehr gut, weil das Zusammenhänge abfragt“, Zweifel an Ontologie

i: „hat mir auch gut gefallen“, „Zusammenhänge werden abgefragt“

cb: „mag ich nicht so gern“, sinnvoll wenn gründliches Lesen wichtig ist

o: auch gut, schwierig wenn „both“ oben steht

5)

1: kontextabhängig, kurze Fragen besser erfassbar, kurze Fragen lieber, eher kein Kriterium

2: positiv, kurze Antworten besser

3: wichtiges Kriterium, gut wenn es verschiedene Vernetzungsgrade gibt, Vernetzung interessanter

4: wichtig, sonst zu trivial

Diskussion: Kapitelspezifische Fragen

A.2.3 Interviewpartnerin 3

1)

- Lernziele widerspiegeln
- wesentliche Dinge werden gefragt
- knifflige Fragen -> keine offensichtlichen Antworten

2)

- Zusammenhänge abgefragt- „teil von“-Beziehung semantisch nicht wertvoll -> sem. wertvolle Info
- wesentliche Punkte abgefragt

3)ld: 3, doppelte Distraktoren, nur Antwortmöglichkeiten von einer Klasse

d: 3 offensichtliche Antworten, zwei gleiche Distraktoren

Anmerkung: evtl. Filter für gemeintem Term, Application Component
-> keine Aufgaben in Distraktoren

c: 6 schwierig, doppelter Distraktor, nur Antwortmöglichkeiten von einer Klasse, lieber „Which function is the CIO responsible for?“, nicht eindeutig beantwortbar

i: 8, irrelevant als Prüfungsfrage (wegen CIOX keine Allgemeinheit)

cb: 8, inhaltlich teils nichtssagend, Formulierungen, Zusammenhänge müssen bekannt sein

o: 7, guter Fragetyp,

4)

ld + d: Abfrage von Faktenwissen, gute Strategie

c: gut, da Zusammenhänge abgefragt werden -> approves nebensächlich, use und update wichtiger

Anmerkung: A updates B, Oberklasse von A updates dann auch B

i: „am meisten“ -> schlecht einschätzbar, in der Form nicht geeignet für die Wissensabfrage

cb: Ein Teilsatz schlechter, Beziehung zwischen beiden Sätzen

„Während man im SIM das und das überwacht, wird im TIM das und das überwacht“ wieder mit richtigen Metamodelltypen

o: Ontologie nicht unbedingt korrekt -> Anderer Fragetyp sollte das ausnutzen

5)

1. auch kurze treffsichere Fragen -> teils teils

2. ja, aber bei längeren Antworten nicht viele Auswahlmöglichkeiten

3. schnell ziemlich unkonkret, aber Vernetzung wichtig, keine „großen Wolken“ an Beziehungen einsetzen

4. wichtig, aber dann auch nur semantisch ähnliche Aufgaben oder Objekttypen, auch syntaktische Ähnlichkeit wichtig

SPARQL-PROGRAMME

```

SELECT DISTINCT STR(?al) AS ?label
WHERE {
  ?a a owl:Class .
  ?a ^<http://open.vocab.org/terms/defines> <http://www.snik.eu/
    ↪ ontology/bb> .
  ?a rdfs:label ?al .
  FILTER( langMatches(LANG(?al), "en") )
  ?a ?p ?b .
  ?b ^<http://open.vocab.org/terms/defines> ?domain2 .
  FILTER (?domain2 != <http://www.snik.eu/ontology/ob>)
}

```

Alg. B.1: *occurrence*-Strategie: Begriffe, welche nur im Strategischen IM auftreten

```

SELECT STR(?al) AS ?label
WHERE {
  ?a a owl:Class .
  ?b a owl:Class .
  ?a skos:closeMatch ?b .
  ?b ^<http://open.vocab.org/terms/defines> <http://www.snik.eu/
    ↪ ontology/bb> .
  ?a ^<http://open.vocab.org/terms/defines> <http://www.snik.eu/
    ↪ ontology/ob> .
  ?a rdfs:label ?al .
  FILTER( langMatches(LANG(?al), "en") )
}

```

Alg. B.2: *occurrence*-Strategie: Begriffe, welche sowohl im Strategischen IM als auch im Taktischen IM auftreten

```

SELECT DISTINCT STR(?al) AS ?label
WHERE {
  ?a a owl:Class .
  ?a ^<http://open.vocab.org/terms/defines> ?domain .
  FILTER (?domain != <http://www.snik.eu/ontology/ob> && ?domain !=
    ↪ <http://www.snik.eu/ontology/bb>)
  FILTER NOT EXISTS {
    ?a skos:closeMatch ?b.
    ?b ^<http://open.vocab.org/terms/defines> ?domain2.
    FILTER (?domain2 == <http://www.snik.eu/ontology/ob> || ?domain2
      ↪ == <http://www.snik.eu/ontology/bb>)
  }
  ?a rdfs:label ?al .
  FILTER( langMatches(LANG(?al), "en") )
}

```

}

Alg. B.3: *occurence*-Strategie: Begriffe, welche weder im Strategischen IM noch im Taktischen IM auftreten

```

import csv

def read_file(path):
    line_list = []
    with open(path, mode='r') as file:
        csvFile = csv.reader(file)
        for lines in csvFile:
            #print(lines)
            line_list.append(lines)
    return line_list

def insert_into_question(liste, answer):
    jslist = []
    for qu in liste:
        js_question_template = "{ question: <span>#1</span>,
↪ answers: [<span>#2</span>,<span>#3</span>,<span>#4</span>], correct: #4
↪ }\n"
        question_template = "In the #1, the #2 #3 the #4, while
↪ in the #5, the #6 #7 the #8 ."
        k = 0
        for i in range(0,8):
            k = i + 1
            question_template = question_template.replace("#"+str
↪ (k), qu[i])
            #print(question_template)
            k = 0
            i = 0
            ans=[question_template, "True", "False", str(answer)]
            for i in range(0,4):
                k = i + 1
                js_question_template = js_question_template.replace("
↪ #"+str(k), ans[i])
                jslist.append(js_question_template)
                #print(js_question_template)
    return jslist
## While in ?resa, the ?al ?pal the ?aal, in the ?resb, the ?
↪ bl ?pbl the ?bbl
# { question: <span>#1</span>, answers: [<span>#2</span>,<
↪ span>#3</span>,<span>#4</span>], correct: #4 }

def export_to_file(liste, files, append):
    point = "w"
    if append:
        point = "a"
    with open(files, "a") as f:

```

```

        for t in liste:
            f.write(t)

l1 = insert_into_question( read_file("../SPARQL/close-by/close-
    ↪ by-true.csv"), 0)
l2 = insert_into_question( read_file("../SPARQL/close-by/close-
    ↪ by-false.csv"), 1)
export_to_file(l1, "../SPARQL/results/close-by.js", True)
export_to_file(l2, "../SPARQL/results/close-by.js", True)

```

Alg. C.1: *close-by*-Einfügen in Fragetext

```

from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys

from enum import Enum
import time
import random
import itertools
import copy

QUESTIONS_PER_STRAT = 20
INPUT_EVAL_QUESTIONS = []

# to get a list: [e.value for e in QUESTION_TYPE]
class QUESTION_TYPE(Enum):
    DEFINITION = 0
    CONTAINS = 1*QUESTIONS_PER_STRAT
    INTERTWINED = 3*QUESTIONS_PER_STRAT
    CLOSE_BY = 2*QUESTIONS_PER_STRAT
    OCCURENCE = 4*QUESTIONS_PER_STRAT

def init():
    temp = []
    with open('./evaluation.js', 'r') as f:
        temp = f.readlines()
    for sentence in temp:
        INPUT_EVAL_QUESTIONS.append(sentence.replace(" ", ""))

def get_question_type(question: str):
    if question.split("?")[0] == "What is defined by this":
        #print("Def")
        return QUESTION_TYPE.DEFINITION
    if question.split(" ")[0] == "What":
        #print("Contains")
        return QUESTION_TYPE.CONTAINS
    if (question.split(" ")[0] + " " + question.split(" ")[1] + "
    ↪ " + question.split(" ")[2]) == "In the context":
        #print("Inter")
        return QUESTION_TYPE.INTERTWINED

```

```

if (question.split(" ")[0] + " " + question.split(" ")[1]) ==
↳ "In which":
    #print("Occur")
    return QUESTION_TYPE.OCCURENCE
#print("Close")
return QUESTION_TYPE.CLOSE_BY

def equals_question(ind: int, qu_text: str) -> bool:
# {question: <span> q </span>, answers: [<span>0</span>,<span>
↳ >1</span>,<span>2</span>,<span>3</span>], correct: k},
if INPUT_EVAL_QUESTIONS[ind].split("<span>")[1].split("</span>
↳ >")[0].replace("\", "") == qu_text.replace(" ", "").
↳ replace("\", ""):
    return True
return False

def answer_match_finder(i: int, answer_list: list) -> (bool, int)
↳ :
temp_ans_list = copy.copy(answer_list)
temp_ans_list = [x.replace(" ", "") for x in temp_ans_list]
temp_ans_list.sort()
temp_input_answers = INPUT_EVAL_QUESTIONS[i].split("[")[1].
↳ split("]")[0].replace("<span>", "").replace("</span>", "")
↳ .replace(" ", "")[:-1].split(",")
temp_input_answers.sort()
correct_index_of_input = int(INPUT_EVAL_QUESTIONS[i].split("
↳ correct:")[1].replace(" ", "")[0:1])
correct_ordered_input = INPUT_EVAL_QUESTIONS[i].split("[")
↳ [1].split("]")[0].replace("<span>", "").replace("</span>",
↳ "").replace(" ", "")[:-1].split(",")
if temp_ans_list == temp_input_answers:
    # correct question answers found
    for n in range(len(answer_list)):
        if answer_list[n].replace(" ", "") ==
↳ correct_ordered_input[correct_index_of_input]:
            return (True, n)
return (False, -1)

def solver_fragmented():
init()
browser = webdriver.Firefox()
browser.get('localhost:3000')
question_text = ""
for i in range(len([e.value for e in QUESTION_TYPE]) *
↳ QUESTIONS_PER_STRAT):
    question = browser.find_element(By.CLASS_NAME, "question-
↳ title")
    question_text = question.text
    a = browser.find_element(By.CLASS_NAME, "question-answers
↳ ")
    answer_list=a.text.split("\n")
    breaker = False

```

```

        # iterate over various questions: i := question index;
        ↪ questions are located in one file, divided by type only by
        ↪ line number; there are QUESTIONS_PER_STRAT number of
        ↪ questions per strategy
        for p in range(get_question_type(question_text).value,
        ↪ get_question_type(question_text).value +
        ↪ QUESTIONS_PER_STRAT+1):
            if equals_question(p, question_text):
                bool_yes, index = answer_match_finder(p,
        ↪ answer_list)
                if bool_yes:
                    question.send_keys(Keys.TAB*(index+1) + Keys.
        ↪ RETURN)
                    time.sleep(2)
                    breaker = True
            if breaker:
                break
solver_fragmented()

```

Alg. C.2: Python Quizsolver Programmcode

ERKLÄRUNG

Ich versichere, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe, insbesondere sind wörtliche oder sinngemäße Zitate als solche gekennzeichnet.

Mir ist bekannt, dass Zuwiderhandlung auch nachträglich zur Aberkennung des Abschlusses führen kann.

Ich versichere, dass das elektronische Exemplar mit den gedruckten Exemplaren übereinstimmt.

Leipzig, 25.01.2022

Arne Roszeitis