

Wilhelm-Ostwald-Schule, Gymnasium der Stadt Leipzig

Universität Leipzig

Medizinische Fakultät

Institut für Medizinische Informatik, Statistik und Epidemiologie

QUESTION ANSWERING AUF SNIK

BESONDERE LERNLEISTUNG

vorgelegt von

Hannes Raphael Brunsch

Besondere Lernleistung Referenzfach Informatik

Leipzig, den 20. Januar 2023

AUTOR:

Hannes Raphael Brunsch

Geboren am 20.10.2005 in Leipzig, Deutschland

TITEL:

Question Answering auf SNIK

INSTITUT:

Institut für Medizinische Informatik, Statistik und Epidemiologie

Medizinische Fakultät Universität Leipzig

REFERENT:

Herr Michael Haase

BETREUER:

Dr. Konrad Höffner

ABSTRAKT

Mit der beständig fortschreitenden Digitalisierung im Gesundheitswesen wird es immer wichtiger, auch das Wissen über das Informationsmanagement, also die Verarbeitung von Informationen und die dazu nötigen Schritte u.v.a.m. dort digital und strukturiert erreichbar zu machen. Diese Arbeit beschäftigt sich mit der vom Institut für Medizinische Informatik, Statistik und Epidemiologie entwickelten Wissensbasis SNIK. Diese enthält Wissen aus dem Bereich des Informationsmanagements im Krankenhaus und soll künftig auch bei dem Studium der Medizininformatik helfen. Momentan ist sie aber nicht mit geschriebener Sprache abrufbar, sondern nur über andere Werkzeuge, welche es schwierig machen, das Wissen zu durchsuchen. Eine Lösung hierfür ist Question Answering (Fragenbeantwortung). Es soll möglich sein, dass ein Nutzer eine englische Frage in Satzform stellt und darauf eine Antwort bekommt. Hierfür gibt es verschiedene Systeme, viele sind allerdings auf andere Wissensbasen spezialisiert. Die Schwierigkeit bei solchen Systemen ist einerseits, die menschlichen Fragen in eine für Computer lesbare Form, d.h. eine Abfrage an die Wissensbasis, umzuformen und andererseits, aus verschiedenen Antwortmöglichkeiten die auszuwählen, welche die fragende Person am wahrscheinlichsten meinte. Das Ziel dieser Arbeit war, nach Systemen zum Question Answering zu recherchieren und letztendlich eines auf SNIK anzuwenden. Die Antworten wurden außerdem anhand eines vorher definierten Fragenkataloges auf ihre Genauigkeit hin überprüft und gewertet.

DANKSAGUNG

Zuallererst möchte ich meinem externen Betreuer, Dr. Konrad Höffner, für die sehr gute und wichtige Betreuung bedanken. Du warst Ansprechpartner bei allen möglichen Fragen und hast sie immer sehr ausführlich erklärt, das Whiteboard hätte ich mir manchmal abfotografieren sollen. Es war auch immer schön, sich in den Stunden, zu denen ich im Institut war, gegenseitig abzulenken. Du hast mir nicht nur Rat für meine BeLL, sondern auch für wissenschaftliche Arbeiten im Allgemeinen und für das Leben gegeben, dafür möchte ich ebenfalls danke sagen, selbst wenn ich gerade letzteres vermutlich nicht immer beachten werde. Besonders hilfreich waren immer deine Verbesserungsvorschläge und dein quasi konstantes Korrekturlesen, welche die Arbeit um einiges verbessert haben.

Danke an Dr. Dennis Diefenbach, Geschäftsführer und technischer Direktor der QA Company, der sich viel Zeit genommen hat, um uns die Möglichkeiten von QAnswer genauer zu erklären und uns bei Fragen bezüglich des Systems geholfen hat. Die Korrektur der SPARQL-Fragen war sehr wichtig für die Arbeit, und die Erklärung der Konfiguration hat die uns offen stehenden Möglichkeiten sehr erweitert. Die Hilfe kam außerdem erstaunlich schnell und flexibel dank der direkten Kommunikation.

Unersetzlich war Dr. Franziska Jahn für die fachliche Korrektur der Fragen, und das sehr schnell. Danke! Noch habe ich die Medizininformatik nicht studiert, da war die Hilfe durch eine Expertin sehr wichtig.

Für die Hilfe bei der Auswahl eines Systems und den Verweis auf TeBaQA und das Question Answering-Leaderboard möchte ich Prof. Dr. Rico Usbeck danken, welcher uns die Welt der aktuellen Question Answering-Systeme eröffnet hat. Bedanken möchte ich mich auch bei meinem internen Betreuer, Herrn Haase, welcher zu Fragen bezüglich der BeLL bereit stand und da auch immer sehr hilfreich war. Danke an meine Eltern, welche diese Arbeit Korrektur gelesen haben. Meinen Dank schulde ich auch Prof. Dr. Alfred Winter, da er mich an Dr. Konrad Höffner weitergeleitet hat und meine BeLL am IMISE erst ermöglicht hat.

Zuletzt möchte ich noch meinen Freunden danken, die mich während der Arbeit immer wieder bei allem Möglichen unterstützt und auch Ablenkung von der Schule geboten haben. Insbesondere danken muss ich hier noch der Person, die mich auch beim dezent zu späten Werkeln an dieser Arbeit unterstützt und mir damit, aber auch sonst immer, sehr geholfen hat. Noch dazu hat sie die Arbeit auch zur Korrektur gelesen!

Danke!

INHALTSVERZEICHNIS

Abstrakt	iii
1 Einleitung	9
1.1 Gegenstand	9
1.2 Problemstellung	10
1.3 Motivation	11
1.4 Zielsetzung	11
1.5 Aufgabenstellung	12
1.6 Aufbau der Arbeit	13
2 Grundlagen	14
2.1 Medizinische Informatik	14
2.1.1 Informationssysteme im Gesundheitswesen	14
2.1.2 Daten, Informationen, Wissen	14
2.2 Geschriebene Sprache	15
2.2.1 Wort und Begriff	15
2.2.2 Synonyme, Homonyme und Ambiguität	16
2.2.3 Lexik, Syntax und Semantik	16
2.3 Semantic Web	18
2.3.1 URIs, IRIs und URLs	18
2.3.2 World Wide Web	18
2.3.3 Semantic Web	18
2.3.4 RDF	19
2.3.5 Ontologien	21
2.3.6 Klassen	22

2.3.7	Individuen und Klassen	23
2.3.8	Wissensbasen	25
2.3.9	Linked (Open) Data	26
2.3.10	SPARQL	26
2.4	Semantisches Question Answering	28
2.5	Automatische Sprachverarbeitung	31
2.5.1	NLP-Pipeline	32
2.5.2	NLP-Modelle und -Repräsentationen	33
2.6	Maschinelles Lernen und neuronale Netze	35
2.6.1	Deep Learning	37
2.6.2	Convolutional Neural Network	37
2.6.3	Recurrent Neural Network	38
2.6.4	LSTMs und GRUs	39
2.6.5	Training von neuronalen Netzen	40
2.6.6	Regelbasiertes System	41
3	Stand der Forschung	42
3.1	Die Ontologie: SNIK	42
3.1.1	Die Architektur SNIKs	42
3.1.2	Der SNIK-Graph	44
3.2	Stichwortsuche	46
3.2.1	Elasticsearch	46
3.3	Question Answering-Systeme	48
3.3.1	Leaderboard	49
3.3.2	gAnswer	50
3.3.3	DeepPavlov	50
3.3.4	TeBaQA	51
3.3.5	QAnswer KG	54

3.3.6 AskNow QA	58
4 Lösungsansatz	61
4.1 Lösungsansatz zum Problem P1	61
4.1.1 Evaluierungsmaße	62
4.2 Lösungsansatz zum Problem P2	64
5 Ausführung der Lösung	65
5.1 Erstellung des Benchmarks	65
5.1.1 Klassifizierung der Textbuchfragen	67
5.1.2 Automatische Erstellung einfacher Fragen	69
5.2 Auswahl eines Kandidaten	72
5.2.1 TeBaQA	73
5.2.2 QAnswer KG	74
6 Ergebnisse	91
7 Diskussion und Ausblick	92
7.1 Benchmark	92
7.2 Systeme	94
 Literatur	 97
 Appendix	
A Syntaktisches und POS-Tagging	106
B Klassifizierung der Textbuchfragen	109
B.1 Erste Einordnung	109
B.2 Formulierung der SPARQL-Abfragen	118
B.3 Antworten auf die SPARQL-Abfragen	125

ABBILDUNGSVERZEICHNIS

Abbildung 2.1	Beispiel für Hierarchie	24
Abbildung 2.2	Beispielhafter Syntaxbaum	32
Abbildung 2.3	Struktur von DNNs	36
Abbildung 2.4	Struktur von CNNs	38
Abbildung 2.5	Struktur von RNNs	39
Abbildung 3.1	SNIK Metamodell Version 8	43
Abbildung 3.2	Ausschnitt SNIK-Graph	45
Abbildung 3.3	DeepPavlov Architektur	51
Abbildung 3.4	QAnswer KG Arbeitsablauf	54
Abbildung 3.5	AskNow Architektur	60
Abbildung 5.1	Durchschnittliches F-Maß bei den automatisch generierten Testfragen in Abhängigkeit zur Anzahl der generierten Trainingsfragen.	82
Abbildung 5.2	Durchschnittliche Präzision bei den automatisch generierten Testfragen in Abhängigkeit zur Anzahl der generierten Trainingsfragen.	83
Abbildung 5.3	Durchschnittlicher Recall bei den automatisch generierten Testfragen in Abhängigkeit zur Anzahl der generierten Trainingsfragen.	84
Abbildung 5.4	Durchschnittlicher Confidence-Wert QAnswers bei den automatisch generierten Testfragen in Abhängigkeit zur Anzahl der generierten Trainingsfragen.	85

Abbildung 5.5	Durchschnittliches F-Maß bei den Lehrbuchfragen im Testset in Abhängigkeit zur Anzahl der generierten Trainingsfragen.	87
Abbildung 5.6	Durchschnittliche Präzision bei den Lehrbuchfragen im Testset in Abhängigkeit zur Anzahl der generierten Trainingsfragen.	88
Abbildung 5.7	Durchschnittlicher Recall bei den Lehrbuchfragen im Testset in Abhängigkeit zur Anzahl der generierten Trainingsfragen.	89
Abbildung 5.8	Durchschnittlicher Confidence-Wert QAnswers bei den Lehrbuchfragen im Testset in Abhängigkeit zur Anzahl der generierten Trainingsfragen.	90

TABELLENVERZEICHNIS

Tabelle 2.1	Beispiele für RDF-Tripel	19
Tabelle 2.2	In dieser Arbeit verwendete Präfixe	20
Tabelle 3.1	Beispiel für invertierte Indizes	47
Tabelle A.1	Penn Treebank-Notation syntaktischer Tags	107
Tabelle A.2	Penn Treebank-Notation von POS-Tags	108
Tabelle B.1	Fragenklassifikation	109

AKRONYME

AFB	Anforderungsbereich
API	Application Programming Interface
BERT	Bidirectional Encoder Representations from Transformers
BOA	Bootstrapping the Web of Data
CDQA	Closed-Domain Question Answering
CNN	Convolutional Neural Network
DAG	Directed Acyclical Graph
DNN	Deep Neural Network
ELMo	Embeddings from Language Models
FSL	Few-Shot Learning
GRU	Gated Research Unit
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IMISE	Institut für Medizinische Informatik, Statistik und Epidemiologie
IRI	Internationalized Resource Identifier
JSON	JavaScript Object Notation

KB	Wissensbasis
KBQA	Knowledgebase Question Answering
LSTM	Long short-term memory
MLM	Masked Language Model
NLP	Natural Language Processing
NLU	Natural Language Understanding
NN	Neural Network
NQS	Normalized Query Structure
ODQA	Open-Domain Question Answering
POS	Part-Of-Speech
QA	Question Answering
QALD	Question Answering over Linked Data
RDF	Resource Description Framework
REST	Representational State Transfer
RNN	Recurrent Neural Network
SQA	Semantisches Question Answering
SNIK	Semantisches Netz des Informationsmanagements im Krankenhaus
SPARQL	SPARQL Protocol and RDF Query Language
Turtle	Terse RDF Triple Language

URI	Uniform Resource Identifier
URL	Uniform Resource Locator
W ₃ C	World Wide Web Consortium
WWW	World Wide Web
XML	Extensible Markup Language
ZSL	Zero-Shot Learning

EINLEITUNG

1.1 GEGENSTAND

„Die Medizinische Informatik ist die Wissenschaft der systematischen Erschließung, Verwaltung, Aufbewahrung, Verarbeitung und Bereitstellung von Daten, Informationen und Wissen in der Medizin und im Gesundheitswesen. Sie ist von dem Streben geleitet, damit zur Gestaltung der bestmöglichen Gesundheitsversorgung beizutragen.“¹

SNIK ist ein Projekt zum Informationsmanagement im Gesundheitswesen. Es fasst Wissen aus drei Lehrbüchern, welches es in semantischer Weise modelliert und publiziert. In der Ontologie (siehe Abschnitt 2.3.5) sind Zusammenhänge zwischen verschiedenen Informationen vorhanden, welche auf der öffentlich erreichbaren Visualisierung in Form eines Graphen dargestellt sind.

Diese Daten liegen strukturiert, aber für Menschen nicht leicht lesbar vor. Daher braucht es andere Methoden, durch sie zu navigieren. Dazu gibt es verschiedene Lösungen:

1. das Verwenden eines [RDF-Browsers](#) (Po, Bikakis et al., 2020, S. 30), der [RDF-Informationen](#) einzeln darstellt,

¹ <https://www.gmds.de/de/aktivitaeten/medizinische-informatik/>, abgerufen am 11. Januar 2022

2. die Graphvisualisierung (Po, Bikakis et al., 2020, S. 32) oder
3. SPARQL, eine Abfragesprache für RDF.

1.2 PROBLEMSTELLUNG

Momentan müssen Studierende der Medizininformatik, die nach Wissen suchen, auf eine der drei oben genannten Optionen zurückgreifen. Jede dieser Möglichkeiten hat jedoch große Nachteile. Der RDF-Browser gibt nur ein sehr beschränktes Ergebnis aus, und serialisiertes RDF selbst zu lesen ist schwer und unpraktisch. Die Graphvisualisierung kann im Zweifel unübersichtlich oder überwältigend sein, da es schwer sein kann, überhaupt die Ressource zu finden, zu der man eine Frage hat, und dann zur Antwort zu navigieren. Im Fall von SPARQL gibt es einen erheblichen Zeitaufwand für die Studierenden, da sie sich dort erst in die Syntax der Abfragesprache und das Vokabular des Fachbereichs einarbeiten müssen.

Daraus ergibt sich das Problem, dass keine der momentan existierenden Lösungen intuitiv genug funktioniert, als dass es nahezu keine Einarbeitungszeit gibt. Die existierenden Lösungen liefern zudem nicht übersichtlich ausreichend Informationen, ihrer Expressivität sind demnach deutliche Grenzen gesetzt.

Obwohl ein Ansatz für die Lösung dieses Problems besteht, Question Answering, wirft dieser direkt ein neues Problem für die entwickelnden Personen auf. Die Implementierung eines Question Answering-Systems mit adäquater Qualität der Antworten ist wesentlich aufwändiger (Diefenbach, Giménez-García et al., 2020, S. 3), als es in einem angemessenem Zeitraum bei stark beschränkten Mitteln möglich ist.

- Problem P₁: Für die Implementierung von Question Answering benötigter Aufwand
- Problem P₂: Fehlende Intuitivität und Expressivität existierender Lösungen

1.3 MOTIVATION

Das Wissen zum Informationsmanagement im Krankenhaus ist komplex und oft nur schwer greifbar. Es liegt in Form von Lehrbüchern, aber auch in [SNIK](#) vor.

Studierende haben selten Zeit, sich ganze Kapitel oder gar Bücher durchzulesen, verfügen jedoch auch nicht über die Kenntnisse [SNIK](#) effektiv zu verwenden. Als Folge müssen sie bei Fragen oft ihren Professor oder andere Studierende hinzuziehen. Es wäre ungemein einfacher, wenn sie das strukturierte Wissen in natürlicher Sprache abfragen könnten. Question Answering-Systeme sind im Idealfall 24 Stunden am Tag und 7 Tage die Woche leicht erreichbar und können sofort antworten. Besonders in Zeiten der Covid-19-Pandemie, in denen direkte Kontakte mit Studierenden und Professoren oft eingeschränkt werden müssen und es digitale Veranstaltungen ohne örtliche Präsenz gibt, ist solch ein Werkzeug sehr hilfreich.

1.4 ZIELSETZUNG

Im Folgenden werden beiden Problemen je ein Ziel zugeordnet.

Das Problem P₁ des Aufwandes der Implementierung eines eigenen Systems kann durch das Verwenden eines bereits existierenden Systems

umgangen werden. Dieses muss das Wissen [SNIKs](#) nutzen, um Fragen zu beantworten.

Das Ziel unterstützt die Lösung des Problems P2 der mangelhaften anderen Möglichkeiten zur Wissensabfrage oder -aneignung in der Medizininformatik. Das zweite Ziel stellt demnach das Erreichen eines semantischen Question Answering-Systems dar, welches Fragen mit hoher Genauigkeit beantworten kann. Bei Question Answering wird eine Frage als Satz in natürlicher Sprache eingegeben, dessen Bedeutung das System versteht und auf Grundlage dessen eine Antwort ausgibt (Hirschman und Gaizauskas, 2001).

- Ziel Z1: Benchmark für Semantisches Question Answering für [SNIK](#)
- Ziel Z2: Semantisches Question Answering-System, welches Fragen zu [SNIK](#) mit hoher Qualität beantwortet.

1.5 AUFGABENSTELLUNG

Den Zielen sind hier jeweils Aufgaben zugeordnet.

- Aufgaben zu Ziel Z1:
 - Aufgabe A1.1: Sammlung von typischen Benutzerfragen an [SNIK](#)
 - Aufgabe A1.2: Beantwortung dieser Fragen
 - Aufgabe A1.3: Methodik entwickeln, um ein System anhand dieser Fragen und Antworten zu bewerten (Benchmark)
- Aufgaben zu Ziel Z2:
 - Aufgabe A2.1: Recherche existierender Semantisches Question Answering Ansätze und Systeme

- Aufgabe A2.2: Auswahl eines geeigneten Systems
- Aufgabe A2.3: Evaluation dieses Systems am Benchmark
- Aufgabe A2.4: Diskussion der Ergebnisse

1.6 AUFBAU DER ARBEIT

Einleitend werden in Kapitel 1 die Problemstellung, Ziele und daraus folgende Aufgaben beschrieben. Danach werden in Kapitel 2 grundlegende Begriffe und für die Arbeit relevante Themengebiete geklärt. In Kapitel 3 wird der aktuelle Stand der Forschung beschrieben, besonders in Hinsicht auf verschiedene Question Answering-Systeme. Kapitel 4 beinhaltet die Lösungsansätze für die oben benannten Aufgaben, Kapitel 5 die Ausführung dieser und Kapitel 6 Ergebnisse. Kapitel 7 ist die kritische Auseinandersetzung mit den Ergebnissen in der Diskussion und ein Ausblick auf mögliche zukünftige Arbeit.

GRUNDLAGEN

2.1 MEDIZINISCHE INFORMATIK

2.1.1 Informationssysteme im Gesundheitswesen

Definition 1 (System). „Ein System ist eine Menge an Elementen und deren Beziehungen.“ Aus dem Englischen von Winter, Haux et al. (2011, S. 30)

Definition 2 (Informationssystem). „Ein Informationssystem ist ein System das Daten, Informationen und Wissen speichert und verarbeitet.“ Aus dem Englischen von Winter, Haux et al. (2011, S. 30)

Definition 3 (Informationssystem im Gesundheitswesen). „Ein Krankenhaus-informationssystem ist das soziotechnologische Subsystem eines Krankenhauses, welches die gesamte Informationsverarbeitung sowie die zugehörigen menschlichen Akteure in deren respektiven Informationsverarbeitungsrollen umfasst.“ Aus dem Englischen von Winter, Haux et al. (2011, S. 37)

2.1.2 Daten, Informationen, Wissen

Daten sind die physische Repräsentation von Informationen oder Wissen, also zum Beispiel die Zeichenkette „Universitätsklinikum“. Daten können neu interpretiert werden. Informationen hingegen sind Fakten, die gegebenenfalls aus Daten hervorgehen, also zum Beispiel die Information, dass

Frau Mustermann verantwortlich für das jährliche IT-Budget des Uniklinikum Leipzigs ist. Wissen ist eine generelle Information über einen Begriff (Winter, Haux et al., 2011, S. 29), also zum Beispiel, dass die Leiterin des Informationsmanagements für das jährliche IT-Budget ihrer Einrichtung verantwortlich ist. Es gibt auch andere Definitionen dieser drei Begriffe, hier werden allerdings diese verwendet.

2.2 GESCHRIEBENE SPRACHE

In der Schriftsprache werden Daten, Information und Wissen durch Sätze überbracht. Da dies als ganzes ein komplexes Thema ist, werden im Folgenden nur die für die Arbeit relevanten Begriffe betrachtet. Für Question Answering ist die gängigste Sprache Englisch, auch diese Arbeit verwendet sie. Deshalb wird in diesem Kapitel insbesondere auf das Englische eingegangen. Die in diesem Kapitel behandelten Themen gelten sowohl für das Deutsche, als auch das Englische.

2.2.1 *Wort und Begriff*

Ein *Wort* ist eine Anreihung von Lauten, die zusammengesetzt eine Bedeutung haben (Dudenredaktion, 2013, S. 1185). Ein *Begriff* ist ein Wort, welches ein Objekt, eine Person o.ä. beschreibt (Dudenredaktion, 2013, S. 235). Es kann mit dem englischen Wort „concept“ übersetzt werden, nicht zu verwechseln mit dem deutschen *Konzept*.

2.2.2 *Synonyme, Homonyme und Ambiguität*

Synonymität bezeichnet den Umstand, dass zwei verschiedene Wörter auf den gleichen Begriff verweisen, so ist etwa die Wortgruppe „bedeutungsgleiches Wort“ synonym zu „Synonym“. Homonyme meinen, dass ein Wort zwei verschiedene Begriffe beschreibt, z.B. die Bank, welche entweder ein Geldinstitut oder eine Parkbank sein kann.

Ambiguität oder Mehrdeutigkeit bedeutet, dass ein Satz oder eine Wortgruppe verschieden interpretiert werden kann. Zur Auflösung dieser Unklarheit wird meistens mehr Kontext benötigt. Homonymität kann also auch als Ambiguität eines einzelnen Wortes beschrieben werden.

In SNIK tritt Homonymität bei einigen in mehreren Büchern verwendeten Begriffen mit je nach Buch unterschiedlicher Bedeutung auf. Beispielsweise ist laut Heinrich, Riedl et al. (2014) ein „Anwendungssystem“ eine Kombination eines Anwendungsprogramms und Daten. In Ammenwerth, Haux et al. (2014) schließt der Oberbegriff „Anwendungssystem“ jedoch auch analoge Methoden zur Speicherung von Daten mit ein.

2.2.3 *Lexik, Syntax und Semantik*

Die *Lexik* beschreibt die Wortarten und beschäftigt sich im Allgemeinen mit dem aus *Lexemen* (hier: Wörtern) bestehenden Wortschatz einer Sprache. Es wird zwischen flektierbaren und unflektierbaren Wortarten unterschieden. Im Deutschen kann man Verben, Substantive, Adjektive, Artikel und Pronomen konjugieren bzw. deklinieren. Sie werden auch flektierbar genannt, ihre Form ist veränderlich. Die Form unflektierbarer Wortarten ist unveränderlich, im Deutschen sind das Partikel wie Adverbien und Interjektionen

(Aleker, Krebsbach et al., 2010, S. 505). Im Englischen gibt es ebenfalls viele flektierbare Wörter Wortarten, diese haben jedoch häufig nur geringe Änderungen in der Form (Blevins, 2006).

Die *Syntax* behandelt den Satzbau und somit die Satzglieder wie Subjekt, Prädikat und Objekt oder Satzarten wie der Aussage-, Aufforderungs- und Fragesatz und deren Unterarten. Unterarten von Fragesätzen lassen sich nach der Form ihrer verlangten Antwort (*Intention*) unterscheiden. So wird bei einer Entscheidungsfrage ein Wahrheitswert erwartet, bei einer Ergänzungsfrage nach einer bestimmten Informationen gesucht und bei einer rhetorische Frage keine Antwort erwartet. Besonders Ergänzungsfragen, welche typischerweise mit einem Fragewort beginnen, sind in dieser Arbeit von besonderer Relevanz.

Die *Semantik* beschäftigt sich mit der Bedeutung der einzelnen Wörter, Sätze oder Zeichen und fokussiert sich dabei im Sinne der Arbeit besonders, aber nicht nur, auf Präpositionen, Adjektive und Adverbien. Sie beschäftigt sich folglich auch mit Synonymen und Homonymen, so wird z.B. versucht, alle möglichen Bedeutungen eines Wortes zu erfahren. Die Semantik ist essentiell, um Fragen, die in natürlicher Sprache gestellt werden, interpretieren zu lassen.

Lexikalisierungen im semantischen Sinne beschreiben die einzelnen Teilbedeutungen eines Wortes (Talmy, 1985). Im allgemeinen Sinne beschreiben Lexikalisierungen den Bedeutungswandel von insbesondere zusammengesetzten Wörtern weg von den Teilbedeutungen der ursprünglichen Bestandteile, hin zu einer neuen, eigenen Bedeutung.

2.3 SEMANTIC WEB

2.3.1 URIs, IRIs und URLs

Uniform Resource Identifier ([URI](#)) werden zur eindeutigen Zuordnung von Zeichenketten zu beliebigen Ressourcen verwendet (Berners-Lee, Fielding et al., 2005). Uniform Resource Locator ([URL](#)) ist eine [URI](#) mit konkreten Zugriffsmöglichkeiten auf die Ressource (Berners-Lee, Masinter et al., 1994). Ein Internationalized Resource Identifier ([IRI](#)) ist ein URI mit einer größeren Anzahl an möglichen Zeichen, da es auch nichtlateinische Schriftzeichen erlaubt.

2.3.2 World Wide Web

Das World Wide Web ([WWW](#)) wurde 1991 von Tim Berners-Lee entwickelt. Es ermöglicht freien Zugriff auf und Austausch von Daten über das Hypertext Transfer Protocol ([HTTP](#)). Die Standards zu dieser Kommunikation setzt das World Wide Web Consortium ([W3C](#)), dessen Direktor Berners-Lee ist. Das Format der Dokumente wird durch die Hypertext Markup Language ([HTML](#)) beschrieben, welche ebenfalls vom [W3C](#) kontrolliert wird (Berners-Lee, Cailliau et al., 1994).

2.3.3 Semantic Web

Im Gegensatz zu [WWW](#), einem Web der Dokumente, ist das Semantic Web ein Web der Daten. Es ist außerdem nicht nur ein verwendbares Artefakt

wie das [WWW](#), sondern auch ein Forschungsfeld (Hitzler, 2021), das sehr stark von anderen Forschungsfeldern, wie etwa Deep Learning und künstlicher Intelligenz, abhängig ist und in seinen Möglichkeiten unterstützt wird (Kotis, Zachila et al., 2021). Teile des Forschungsfeldes sind Ontologien, Linked Data und sogenannte Knowledge Graphs, auf die hier aber nicht weiter eingegangen werden wird. Das Semantic Web als nutzbares Objekt besteht aus strukturierten Daten. Das bedeutet, dass man, im Gegensatz zum [WWW](#), Informationen direkt abfragen kann und nicht erst die [HTML](#)-Dokumente auswerten muss.

2.3.4 Resource Description Framework (RDF)

Funktion	Beispiel 1	Beispiel 2
Subjekt	<code>bb:ChiefInformationOfficer</code>	<code>ex:ErikaMustermann</code>
Prädikat	<code>meta:isResponsibleForEntityType</code>	<code>rdf:type</code>
Objekt	<code>bb:AnnualITBudget</code>	<code>ciox:ChiefInformationOfficer</code>

Tabelle 2.1: Beispiele für RDF-Tripel

Das Resource Description Framework ([RDF](#)) ist das grundlegende zur Beschreibung des Semantic Webs genutzte Format (Hitzler, Kröttsch et al., 2007, S. 35). Mithilfe dieser formalen Sprache, die erstmals 1999 durch das [W3C](#) spezifiziert wurde, sollten zunächst Metadaten modelliert werden. Aufgrund des Fokus auf die Weiterverarbeitbarkeit der Daten ist es für seine Nutzer gut zur Modellierung des Semantic Webs geeignet.

RDF wird in Tripeln aus Subjekt, Objekt und Prädikat modelliert. Es verhält sich hierbei wie in einem einfachen deutschen Satz, das Prädikat

beschreibt also die Beziehung zwischen dem Subjekt und dem Objekt. Alle drei werden normalerweise als URIs angegeben (Berners-Lee, 2006).

Beispiel für solche Tripel aus URIs sind in Tabelle 2.1 zu sehen: Die Leiterin des Informationsmanagements (Subjekt) verantwortet (Prädikat) das jährliche IT-Budget (Objekt). Bei Beispiel zwei ist wichtig, dass `bb:ChiefInformationOfficer` und `ciox:ChiefInformationOfficer` nicht das gleiche Wissen zeigen.

Durch den *Präfix* wird die genaue und einzigartige Adresse festgelegt. Die Wahl des Präfixes ist hierbei den Verantwortlichen überlassen, wichtig ist nur, sie auf eine HTTP-Adresse verweisen zu lassen. Der Präfix „ex:“ wird z.B. häufig für Beispiele benutzt. Alle in dieser Arbeit verwendeten Präfixe sind in Tabelle 2.2 mit ihren URIs aufgelistet.

Präfix	URI
bb	<code>http://snik.eu/ontology/bb/</code>
ciox	<code>http://snik.eu/ontology/ciox/</code>
he	<code>http://snik.eu/ontology/he/</code>
ob	<code>http://snik.eu/ontology/ob/</code>
meta	<code>http://snik.eu/ontology/meta/</code>
sniko	<code>http://www.snik.eu/ontology/</code>
rdf	<code>http://www.w3.org/1999/02/22-rdf-syntax-ns#</code>
rdfs	<code>http://www.w3.org/2000/01/rdf-schema#</code>
ex	<code>http://example.org/</code>

Tabelle 2.2: In dieser Arbeit verwendete Präfixe

2.3.5 *Ontologien*

„Im Kontext der Informatik ist eine Ontologie eine Menge von repräsentativen Primitiven, welche einen Bereich des Wissens oder eines Diskurses modellieren. Die repräsentativen Primitiven sind typischerweise Klassen (oder Mengen), Attribute (oder Eigenschaften), und Zusammenhänge (oder Beziehungen zwischen Klassenelementen). Die Definitionen der repräsentativen Primitiven enthalten Informationen über ihre Bedeutung und Beschränkungen ihrer logisch konsistenten Anwendung. Im Kontext von Datenbanksystemen kann eine Ontologie als ein Level der Abstraktion von Datenmodellen, analog zu hierarchischen und relationalen Modellen, gesehen werden, aber dazu gedacht, Wissen über Individuen, deren Attribute, und deren Beziehungen zu anderen Individuen zu modellieren. Ontologien sind typischerweise in Sprachen, die Abstraktionen weg von Datenstrukturen und Implementationsstrategien ermöglichen spezifiziert; In Praxis sind die Sprachen von Ontologien in der expressiven Kraft näher an Logik erster Ordnung als alle anderen zur Modellierung von Datenbanken genutzten Sprachen. Deshalb werden Ontologien als auf dem ‚semantischen‘ Level stehend angesehen, wohingegen andere Datenbankschemen auf dem ‚logischen‘ oder ‚physischen‘ Level sind. Durch ihre Unabhängigkeit von Datenmodellen niedrigeren Levels werden Ontologien zum Integrieren von heterogenen Datenbanken genutzt, was die Benutzbarkeit über verschiedene Systeme erlaubt, und die Spezifizierung von unabhängigen, wissensbasierten Dienstleistungen ermöglicht. Im Technologie-Stack der Semantic Web Standards sind Ontologien ihre eigene Schicht. Es gibt nun standardisierte Sprachen und eine Vielzahl an kommerziellen und Open-Source

Werkzeugen, um Ontologien zu erstellen und mit ihnen zu arbeiten. “

Aus dem Englischen von Gruber (2016)

2.3.6 Klassen

In der Informatik beschreiben Klassen die Abstraktion und Kategorisierung von Wissen. Solche Klassen können auch Subklassen von anderen sein. Subklassen können wiederum weitere Subklassen haben. Dahingegen ist die Beziehung zwischen Sub- bzw. Unterklassen U und Eltern- bzw. Oberklassen O eine Teilmengenbeziehung ($U \subseteq O$), das bedeutet, dass jedes Element der Unterklasse auch ein Element der Oberklasse ist. Echte Teilmengenbeziehungen ($U \subset O$) existieren, wenn jedes Element von U ein Element von O ist, O aber mindestens ein Element enthält, das nicht in U ist (Hitzler, Krötzsch et al., 2007, S. 261).

Das Beispiel in Abb. 2.1 zeigt eine Baumstruktur, das heißt jede Subklasse hat nur eine Elternklasse. Es stellt die Hierarchie der Klassen dar. Alle Chief Information Officers sind Teil des Krankenhauspersonals, aber nicht des technischen Personals. Die Taxonomie befasst sich als Wissenschaft mit solchen Hierarchien. Andere Graphformen beinhalten unter anderem die des Diamanten, bei der eine Subklasse mehrere Eltern haben kann. Die Darstellungen solcher Hierarchien müssen allerdings immer ein Directed Acyclical Graph (DAG) sein. Die Beziehungen müssen gerichtet sein und die Form darf keinen Kreis bilden. Ohne einen gerichtete Pfeile würde man aufgrund fehlender Richtung unklare Beziehungen zwischen voneinander abhängigen Klassen haben, da nicht klar ist, welche eine Subklasse von welcher ist. Wäre eine Kreisbildung möglich, führte das dazu, dass alle Klassen, die Teil des Kreises sind, gleich wären.

Seien M_1, M_2, \dots, M_n Mengen,

sowie $M_1 \subseteq M_2 \subseteq \dots \subseteq M_n \subseteq M_1; n \in \mathbb{N}, n > 1$

Daraus folgt aufgrund der Transitivität von \subseteq : $M_1 \subseteq M_n$

Da nun $M_1 \subseteq M_n \wedge M_n \subseteq M_1$ gilt $M_1 = M_n$

Sei $i \in \mathbb{N}, 1 < i < n$ und beliebig wählbar;

Daraus folgt aufgrund der Transitivität von \subseteq :

$M_i \subseteq M_n \wedge M_i \supseteq M_1$

Da $M_1 = M_n$ gilt nun $M_i \subseteq M_1 \wedge M_1 \subseteq M_i$

Sodass $\forall i M_i = M_1 = M_n \quad \square$

2.3.7 Individuen und Klassen

Die Beziehung zwischen Individuum und Klasse entspricht einer Elementbeziehung im Sinne der Mengentheorie der Mathematik, für ein Individuum e einer Klasse K gilt also $e \in K$ (Hitzler, Krötzsch et al., 2007, S. 261). Im Beispiel aus Tabelle 2.1 ist also Frau Mustermann ein Element der Klasse *Leiterin des Informationsmanagements*.

Beziehungen zwischen Individuen untereinander werden *Relationen* genannt. Relationen können als Teilmenge zwischen Individuen $a \in A$ und $b \in B$ vom kartesischen Produkt der Klassen A und B bezeichnet werden, also gilt $(a, b) \in R$, wenn $R \subseteq A \times B$ ist. Als Kurzform kann auch aRb geschrieben werden (Hitzler, Krötzsch et al., 2007, S. 263). Relationen zweier Elemente werden als *binäre Relationen* bezeichnet. Relationen, die nicht binär sind, werden im Folgenden nicht benötigt.

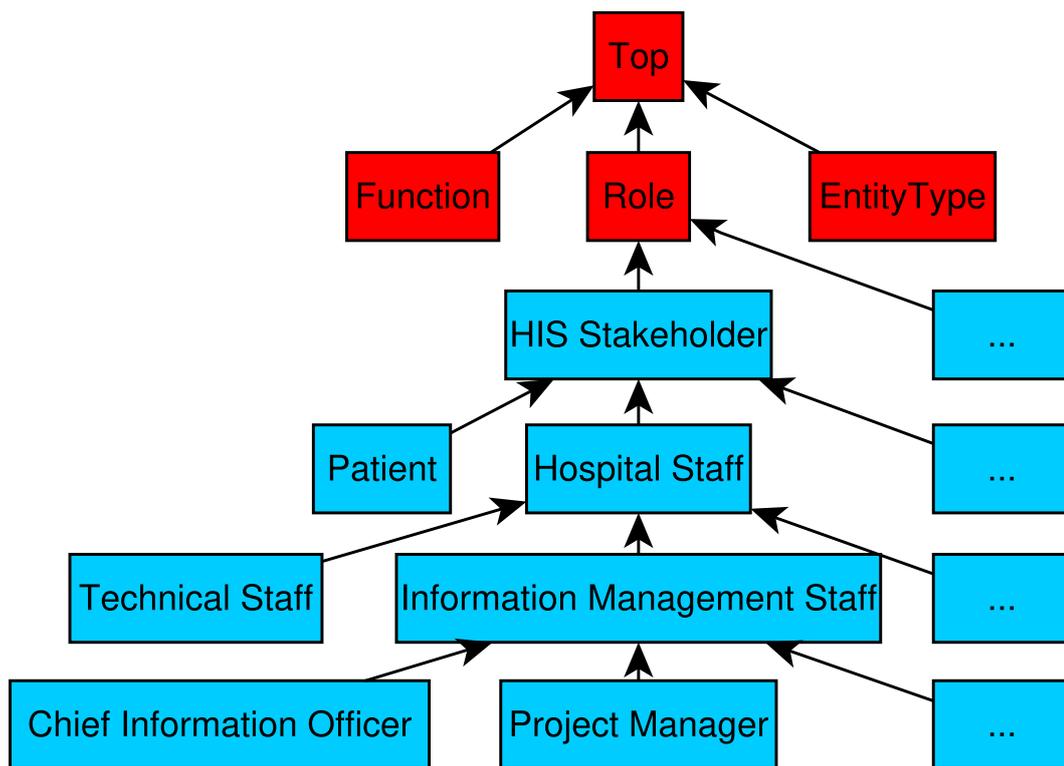


Abbildung 2.1: Eine beispielhafte Hierarchie aus der [SNIK](#)-Ontologie. Quelle: Jahn, Höffner et al. (2019)

In [RDF](#) sind nämlich alle Relationen zweistellig, also Paare. Die Beschreibung dieser Relation wird *Property*, also Attribut oder Eigenschaft, genannt. Properties nehmen die Rolle von Prädikaten an, so ist zum Beispiel `meta:isResponsibleForE` ein Property von der Ressource `bb:ChiefInformationOfficer`, welche im Tripel die Rolle des Subjekts annimmt, mit dem das Objekt darstellenden Wert, `bb:AnnualITBudget`. Sie können auch eingeschränkt werden, indem man mit [RDF](#) ihren *Definitionsbereich* und *Wertebereich* verändert. Damit lässt sich einstellen, welche Datentypen als Subjekt und Objekt akzeptiert werden (Hitzler, Krötzsch et al., 2007, S. 76). Der Definitionsbereich wird über `rdfs:domain` und der Wertebereich über `rdfs:range` bearbeitet.

Literale, die der Wertebereich der *Data*-Properties sind, erfüllen die Funktion der Datentypen. Literale sind Zeichenketten und können über reservierte URIs die Art des vorhandenen Werts anzeigen. Da Properties Klassen als Definitions- und Wertebereich haben, werden sie auch *Object*-Properties genannt. So ist in Tabelle 2.1 `bb:isResponsibleForEntityType` ein Property mit `meta:Role` als Definitionsbereich und `meta:EntityType` als Wertebereich, es beschreibt demnach eine Eigenschaft von Ressourcen.

2.3.8 Wissensbasen

Es wird teilweise auch zwischen A- und TBoxen unterschieden. Eine ABox enthält Wissen über Individuen bzw. Instanzen, wohingegen eine TBox Wissen über Klassen bzw. generelle Schemen enthält (Hitzler, Krötzsch et al., 2007, S. 167). Das heißt, dass ABoxen speziellere Informationen und TBoxen allgemeinere Informationen, also Wissen, enthalten. Man kann ABoxen deshalb auch *Wissensbasen* und TBoxen Taxonomien nennen.

2.3.8.1 Unterschied zwischen relationalen Datenbanken und Wissensbasen

Relationale Datenbanken sind Datenbanken, in welchen die Daten in einer eindeutigen Struktur von Entitäten (Tabellen) vorliegen. In den Entitäten sind spezielle Informationen über die Individuen enthalten. Die Relationen existieren allerdings nur zwischen den Entitäten, nicht zwischen den Individuen. Es sind also immer nur die Klassen miteinander verbunden, die Individuen haben höchstens über Fremdschlüssel etwas miteinander zu tun, dann jedoch nur in einem vorher gesetzten Rahmen. Bei Wissensbasen können Individuen in Relationen mit anderen Individuen sein. Des Weiteren sind Wissensbasen auf zweistellige Relationen beschränkt, wohingegen

relationale Datenbanken eine deutlich höhere Anzahl an Spalten besitzen und somit eine deutlich höhere Stelligkeit der Relationen existiert.

2.3.9 *Linked (Open) Data*

Linked Data beschreibt öffentlich verfügbare Informationen, die mittels [URIs](#) erreichbar und in für Maschinen lesbarer Form vorhanden sind (Bizer, Heath et al., 2011). Es sollen verschiedene open-source lizenzierte Datenquellen in [RDF](#) umgeformt und verbunden werden. Dies kann zu Problemen im Sinne von fehlender Konsistenz der Daten führen (Hitzler, 2021). Berners-Lee (2006) verfasste vier Regeln für das Veröffentlichen von Daten im Internet, um ein umfassendes Linked Data-System zu ermöglichen. Diese sagen im Wesentlichen, dass klar mit [HTTP-URIs](#) benannte Ressourcen standardisiert durch [RDF](#) und SPARQL Protocol and RDF Query Language ([SPARQL](#)) weitere nützliche Informationen und Links zu anderen [URIs](#) enthalten sollen.

2.3.10 *SPARQL*

[SPARQL](#) ist ein [W3C](#)-Standard als Abfragesprache für [RDF](#). Mit ihr kann man durch eine standardisierte Syntax in [RDF](#) geschriebenes Wissen abfragen. Es müssen zuerst die verwendeten Präfixe definiert werden, dann die auszugebenden Variablen genannt und dann die eigentliche Anfrage formuliert werden (Hitzler, Krötzsch et al., 2007, S. 203). Möglich ist auch, dass in dieser Abfrage eine weitere Variable genannt wird. Dies kann zu mehreren Ergebnissen führen. Ein Beispiel für eine [SPARQL](#)-Anfrage ist hier:

```
# Ich bin ein Kommentar.
```

```

# Kommentare werden ignoriert und nicht ausgeführt. Dieser Kommentar
  ↪ hier ist so lang, dass er auf dem Papier über mehrere Zeilen
  ↪ geht, in Wirklichkeit ist er jedoch nur auf einer.

# Definition der Präfixe:
PREFIX bb: <http://snik.eu/ontology/bb/>
PREFIX meta: <http://snik.eu/ontology/meta/>

# Auszugebende Variablen nennen:
SELECT ?aufgabe

# Anfrage:
WHERE
  { ?rolle meta:isResponsibleForEntityType bb:AnnualITBudget .
    ?rolle meta:isResponsibleForFunction ?aufgabe . }

```

Es zeigt alle Aufgaben an, welche die Rolle, die das jährliche IT-Budget verändern kann, hat. Die einzige Person, die dies verändern kann, ist `bb:ChiefInformationOffi`. Dessen Aufgabe ist, wie man sehen kann, wenn man diese Anfrage z.B. online auf der SNIK-Website¹ ausführt, `bb:InformationManagement`.

Wenn man mehrere Beziehungen abfragen will, kann man *property paths*² verwenden. Bei diesen wird zwischen die einzelnen Prädikate ein Zeichen, in diesem Fall ein `|` geschrieben. Diese geben den Pfad an, den die Abfrage verwenden soll. So gibt `/` beispielsweise eine Sequenz an, was im Kontext von Question Answering besonders für komplexe Fragen hilfreich ist. Dabei werden Prädikate hintereinander als „Weg“ der Abfrage geschrieben. Der Charakter `|` steht für einen alternativen Pfad, ähnlich dem logischen *oder*. Um die Suche weiter einzuschränken oder auszuweiten kann man das Schlüsselwort `FILTER` benutzen, nach diesem kann man zum Beispiel auch mehrere Objekte verwenden. Bei beiden dieser Methoden kann man auch

¹ <https://www.snik.eu/sparql>

² <https://www.w3.org/TR/sparql11-query/#propertypaths>, abgerufen am 07. Juni 2022

eine regulären Ausdrücken ähnliche Syntax nutzen. Reguläre Ausdrücke sind Zeichenketten, mit deren Hilfe in Texten nach bestimmten Folgen, Mustern o.ä. gesucht werden kann. Die Auswahl mehrerer Objekte gleichzeitig ist nützlich, um bei [SNIK](#) gleich über alle drei Teilontologien Abfragen zu stellen. Hierfür kann man die gegebene Ressource in der jeweiligen Zeile beispielsweise `?x` nennen, danach fügt man folgende Zeile ein:

```
?x (skos:closeMatch | ^skos:closeMatch)* bb:ChiefInformationOfficer .
```

Da die Teilontologien untereinander mit `meta:closeMatch` verbunden sind, wird diese Beziehung hier abgefragt. Mittels property path und dem Zeichen `^` wird die Suche auch invertiert, sodass auch Ressourcen erhalten werden, wenn die `closeMatch`-Beziehung von der anderen ausgeht. Mit dem Zeichen `*` wird auch `bb:ChiefInformationOfficer` selbst ausgewählt, da dieses Zeichen eine beliebig häufige Anzahl an Wiederholungen des Pfades zulässt. Im Gegensatz zu `+` setzt `*` jedoch keine Mindestanzahl von einer Wiederholung voraus, sondern erlaubt auch Zahlen von null, weshalb auch die Ressource selbst ausgewählt wird.

2.4 SEMANTISCHES QUESTION ANSWERING

Definition 4 (Question Answering). *Question Answering (Fragebeantwortung) behandelt die Beantwortung von Benutzerfragen (Hirschman und Gaizauskas, 2001). Ein Question Answering (QA)-System muss eine Frage analysieren, eine oder mehrere Antworten bereitstellen und dem Nutzer diese präsentieren. Die Fragen sind in natürlicher Sprache gestellt, es wird NLP benötigt, um sie zu verarbeiten.*

Es existieren sowohl Closed-Domain Question Answering (CDQA) als auch Open-Domain Question Answering (ODQA). QA-Programme mit offener Domäne sind schwieriger zu entwickeln, da sie Fragen von allen möglichen Domänen verstehen müssen. Ein Beispiel dafür wäre die Frage, wo eine Bank zu finden sei. Die fragenstellende Person könnte entweder nach der Filiale einer Bankgesellschaft suchen oder eine Sitzgelegenheit meinen. Das Programm, an das die Frage gestellt wird, muss nun mit wenig Kontext herausfinden, welche Deutung impliziert wird. Das kann beispielsweise mit mehr Daten, wie etwa dem vorherigen Suchverlauf (Hat sich die Person vorher nach Parks/Geldfilialen erkundigt?) oder persönlichen Daten, wie dem Alter, möglich sein.

Diese Arbeit beschränkt sich jedoch auf QA mit geschlossener Domäne, SNIK (siehe Kapitel 3). Solche QA-Programme beschränken sich auf einen Fachbereich, was etwa den Kontext vom vorherigen Beispiel bereitstellen würde. Wenn es sich zum Beispiel um eine Wissensbasis zu Parks handelt, ist es wahrscheinlicher, dass die Person nach Sitzgelegenheiten im Park sucht.

Definition 5 (Semantisches Question Answering). *Semantisches Question Answering (SQA), oder auch Knowledgebase Question Answering (KBQA), ist die Beantwortung von Fragen, die RDF-Daten in natürlicher Sprache gestellt werden. Ein Programm für SQA erkennt verschiedene semantische Strukturen in der Frage, zum Beispiel was für ein Typ die Antwort auf die Frage sein soll, wie etwa eine Zeit oder ein Ort (Narayanan und Harabagiu, 2004). Dafür wird NLP verwendet.*

Fragen werden zudem anhand ihrer Komplexität in simple und komplexe Fragen unterschieden. Die Komplexität einer Frage leitet sich daraus ab, wie viele Schritte zum Erreichen der Antwort nötig sind. Bei der Formulierung einer SPARQL-Abfrage zum Erhalten der Antwort lässt sich das gut erkennen.

Es ist zum Beispiel wichtig, wie viele Tripel zur Verknüpfung der Frage mit der Antwort gebraucht, oder ob Filter oder ähnliches verwendet werden.

So ist beispielsweise die Frage „Wer ist für das jährliche IT-Budget verantwortlich?“ nicht komplex, da hier nur die Beziehung zwischen `bb:AnnualITBudget` und den jeweiligen Antworten benötigt werden. Die Frage „Welche Aufgaben hat die Person, welche auch für das jährliche IT-Budget verantwortlich ist?“ könnte schon als komplex gelten, da hier, wie schon in Abschnitt 2.3.10 gezeigt, zwei oder drei Tripel für die eigentliche SPARQL-Abfrage benötigt werden: Eines, um herauszufinden, wer alles für das jährliche IT-Budget verantwortlich ist; eines, um ggf. zu fordern, dass es sich bei der Ressource aus dem ersten Tripel um eine Person handelt; und eines, um letztendlich die eigentliche Intention der Frage, die Aufgaben dieser Person, auszugeben.

Definition 6 (Einfache Frage). *Eine einfache Frage ist eine Frage mit einer eindeutigen Intention, aus der eine oder mehrere Antworten erreicht werden können. Es werden genau zwei Elemente eines Tripels, also zum Beispiel Subjekt und Prädikat, gegeben. In ihr gibt es keine Ein- oder Beschränkungen oder Verzweigungen des Graphen.*

Definition 7 (Komplexe Frage). *Eine komplexe Frage hat auch, wie eine einfache Frage, eine eindeutige Intention, es gibt aber mehr Eingaben als nur zwei Elemente des Tripels. Es können komplexere Graphen entstehen, Verzweigungen und Einschränkungen können existieren.*

Definition 8 (Zusammengesetzte Frage). *Eine zusammengesetzte Frage besteht aus einer oder mehreren Teilfragen, welche miteinander verbunden sind. So besteht beispielsweise die Frage „Wofür sind die Leiterin des Informationsmanagements und der Projektleiter jeweils verantwortlich?“ aus den Teilfragen „Wofür ist die Leiterin des Informationsmanagements verantwortlich?“ und „Wofür ist der Pro-*

jektleiter verantwortlich?“. Die Frage kann also mehrere Intentionen gleichzeitig verfolgen. Teilfragen können sowohl komplexe als auch simple Fragen sein.

2.5 AUTOMATISCHE SPRACHVERARBEITUNG

Natural Language Processing (**NLP**), auf Deutsch *automatische Sprachverarbeitung*, ist die automatische Verarbeitung von natürlicher Sprache, also Sprache, wie sie Menschen sprechen und schreiben. Das große Ziel von **NLP** ist, menschliche Sprache maschinell perfekt verstehen zu können, weshalb das Feld auch Natural Language Understanding (**NLU**) genannt wird. Es behandelt dabei viele Aspekte, von maschineller Übersetzung, womit das Feld überhaupt begann (Jones, 1994), über **QA**, wie es in dieser Arbeit behandelt wird, bis zu Verarbeitung gesprochener Sprache. Bei **NLP** wird Sprache in vier Stufen betrachtet (Reshamwala, Mishra et al., 2013). *Phonologie* existiert nur in gesprochener Sprache und beschreibt die Interpretation der Betonung und Art der Aussprache von Wörtern und ist deshalb für diese Arbeit nicht relevant. *Morphologie* beschreibt die Satzstruktur und die Syntax des Satzes. Während der Betrachtung der Syntax wird der Satz in einen Baum umgeformt, sodass erkennbar ist, welches Wort von anderen abhängt. Ein Beispiel hierfür ist Abb. 2.2. Die *Semantik* des Satzes wird als Drittes betrachtet. Bei der Analyse der *Pragmatik* wird versucht, die wahre Bedeutung des Satzes zu verstehen, also zum Beispiel die Bedeutung von Pronomen oder die Deutung von Sätzen die mehrere Aussagen bedeuten können.

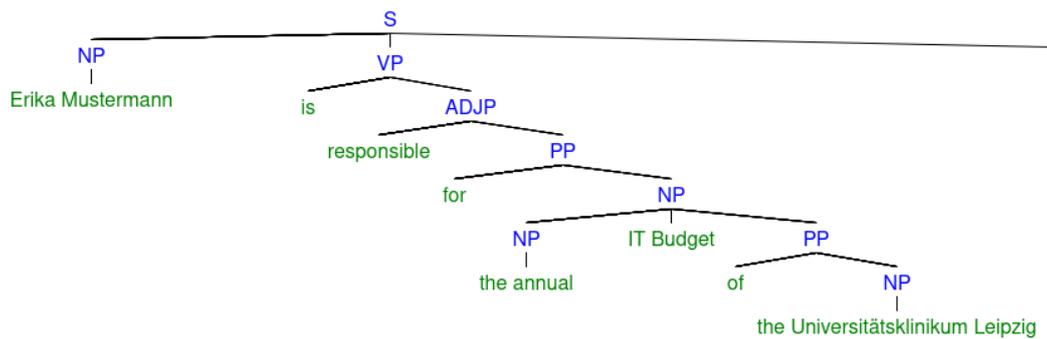


Abbildung 2.2: Beispielhafter Syntaxbaum. Erstellt mit Link Grammar (Sleator und Temperley, 1995) und www.mshang.ca/syntree. *S*: Sentence; *NP*: noun phrase; *VP*: verb phrase; *ADJP*: adjective phrase; *PP*: prepositional phrase. Eingegebener Satz: „Erika Mustermann is responsible for the annual IT Budget of the Universitätsklinikum Leipzig.“

2.5.1 NLP-Pipeline

In einer Pipeline werden verschiedene Komponenten eines Algorithmus oder Verfahren nacheinander ausgeführt. Eine NLP-Pipeline führt verschiedene Teilprogramme aus, da das Feld an sich zu komplex ist, als dass die Arbeit eines Programms es vollkommen lösen könnte. Dabei gibt der Nutzer, hier ein Programmierer, der z.B. ein QA-Programm schreiben will, der Pipeline die Daten, welche je nach Pipeline annotiert vorliegen müssen (Clarke, Srikumar et al., 2012). NLP-Pipelines verwenden Neural Networks (NNs), um funktionieren zu können. Zwei Pipelines werden in Kapitel 3 näher betrachtet.

2.5.2 NLP-Modelle und -Repräsentationen

Modelle für Natural Language Processing verwenden heutzutage meistens Deep Learning. Dazu werden sie vortrainiert, das heißt sie werden schon vor der Anwendung mit einem Verzeichnis an Wörtern bzw. Sätzen, wie zum Beispiel dem englischsprachigen Wikipedia mit etwa 2,5 Millionen Wörtern, oder dem BooksCorpus (Zhu, Kiros et al., 2015), einer Sammlung von Sätzen aus Büchern mit insgesamt etwa 800 Millionen Wörtern, trainiert (Devlin, Chang et al., 2018). Dadurch lernen sie, diese gut zu repräsentieren und können diese dann auch bei anderen Aufgaben anwenden. Es können unbeabsichtigte, aber nützliche, Zusammenhänge zwischen Wörtern abgebildet werden, so kann beispielsweise der Vektor zwischen „Mutter“ und „Vater“ derselbe wie der zwischen „Tante“ und „Onkel“ sein (Mikolov, Sutskever et al., 2013). Ähnlich funktioniert es auch bei komplexeren Zusammenhängen zwischen Wörtern (Mikolov, Chen et al., 2013), was sehr bei der Universalität von Repräsentationen hilft.

Mit *fine-tuning*- oder *feature-basiertem Training* können solche vortrainierten Repräsentationen auch spezifischere Aufgaben erfüllen, also bei einem großen Wortcorpus vortrainiert und für eine weniger häufige Aufgabe verwendet werden. Bei *feature-basiertem Training* werden der Aufgabe spezifische Architekturen verwendet, wie etwa *ELMo*, siehe auch Abschnitt 2.5.2.1. Erst später werden für weitere Funktionen über Repräsentationen hinzugefügt (Peters, Neumann et al., 2018). *Fine-tuning-basiertes Training* basiert auf solchen Repräsentationen, hier werden nur minimal aufgabenspezifische Parameter verwendet. Beispiele für solch ein Vorgehen sind *BERT*, siehe hierfür Abschnitt 2.5.2.2, und OpenAI GPT (Radford, Narasimhan et al., 2018).

2.5.2.1 *ELMo*

Embeddings from Language Models (*ELMo*) (Peters, Neumann et al., 2018) ist ein *NLP*-Modell, welches Wortsequenzen als Vektoren repräsentiert und dadurch sowohl komplexe Wortstrukturen wie Syntax und Semantik und den Kontext der Wörter behält. Im Gegensatz zu anderen Repräsentationsmodellen wird bei *ELMo* also jedes Wort im Kontext des Auftretens repräsentiert. Es basiert auf *LSTMs*, welche später in Abschnitt 2.6.4 betrachtet werden. Diese werden mit einem Wortcorpus vortrainiert und somit gut einsetzbar, und können beispielsweise ohne Veränderung bei Disambiguationsaufgaben verwendet werden. *ELMo* ermöglichte zu der Zeit eine große Verbesserung der Leistung bei *NLP*-Aufgaben.

2.5.2.2 *BERT*

Bidirectional Encoder Representations from Transformers (*BERT*) (Devlin, Chang et al., 2018) ist auch ein *NLP*-Modell, jedoch mit dem hier neuen Ansatz bidirektionaler Transformer (Vaswani, Shazeer et al., 2017). *BERT* nutzt eine von der Cloze-Prozedur inspirierte Vorgehensweise namens Masked Language Model (*MLM*), also „maskiertes Sprachmodell“. Bei dem Cloze-Verfahren werden zufällig Wörter zensiert, wodurch deren Effekt auf die Lesbarkeit von Sätzen getestet wurde. Der Name selbst basiert auf dem Phänomen, dass das menschliche Gehirn unvollkommene Muster vervollständigt. Bei *MLM* wird wie bei der Cloze-Prozedur auch versucht, das zufällig gelöschte Wort anhand des Kontexts wieder einzufügen. Der bi-, also zweidirektionale, Transformer betrachtet den Kontext von beiden anstatt von vorher nur einer Seite, sodass bessere Ergebnisse erzielt werden können. Erst durch diese Vorgehensweise kann von den sehr aufgabenspezifischen

Architekturen weg- und zu fine-tuning-basierten Ansätzen hingegangen werden. Dadurch erreicht BERT in Benchmarks sehr gute Ergebnisse, verglichen mit anderen Modellen. BERT verwendet sogenannte Transformer, welche auf dem in Abschnitt 2.6.3 und in Rush (2018) erklärten Aufmerksamkeitsmechanismus basieren. Es ist außerdem ein DNN, welche im folgenden Abschnitt genauer betrachtet werden.

2.6 MASCHINELLES LERNEN UND NEURONALE NETZE

Neuronale Netze sollen, indem sie die Struktur des menschlichen Gehirns nachstellen, komplexe Aufgaben, wie etwa das autonome Fahren oder Natural Language Processing, ausführen. Dazu werden *Knoten* oder *Neuronen* in verschiedenen Schichten miteinander verbunden. Es gibt drei Arten von diesen: Eingabe- (input), versteckte (hidden) und Ausgabe-Schichten (output). Zwischen den Schichten sind die Knoten verbunden, jede Verbindung hat ein bestimmtes Gewicht, wie in Abb. 2.3 dargestellt wird. Mit diesem Gewicht wird der weitergereichte Wert multipliziert, welcher in der Eingabeschicht erstmals festgelegt wird. Bei jedem Knoten werden die aus den niederen Schichten ankommenden Werte addiert, sodass ein neuer Wert entsteht. Diese Summe wird dann mithilfe einer Funktion, meistens einer hyperbolischen oder Sigmoidfunktion ($\text{sig}(t) = \frac{1}{1+e^{t-1}}$), verändert und dann wieder weitergereicht. Es werden genau diese Funktionen genommen, da deren Ableitungen die Fehlerberechnung leichter machen (LeCun, Bengio et al., 2015). So kann das, je nach der Anzahl der versteckten Schichten, lange weitergehen. Anfangs gab es meist nur eine versteckte Schicht, später wurden mehrere von diesen eingefügt und sogenannte Deep Neural Networks entstanden.

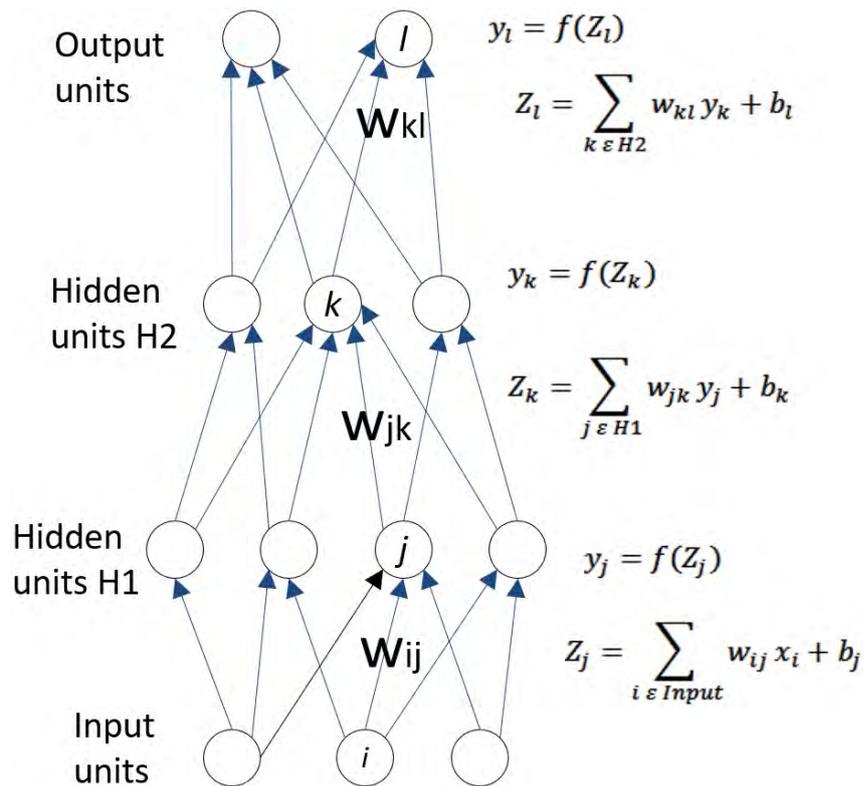


Abbildung 2.3: Darstellung der Struktur eines mehrschichtigen neuronalen Netzes und der Fehlerberechnung. Quelle: Shrestha und Mahmood (2019)

2.6.1 *Deep Learning*

Deep Learning ist ein Teilgebiet des maschinellen Lernens, das [NNs](#), zu Deutsch *neuronalen Netze*, mit mehreren versteckten Schichten enthält (Vargas, Mosavi et al., 2017). Es befasst sich mit Deep Neural Networks ([DNNs](#)), welche häufig eine deutlich höhere Effizienz als [NNs](#) mit nur einer versteckten Schicht aufweisen. Das bedeutet, dass die Ergebnisse für weniger oder gleich viel Training bessere Ergebnisse erzielen. Die Qualität der Ergebnisse lässt sich z.B. bei der Klassifizierung von Bildern gut ermitteln. Hier wird dem [NN](#) ein Bild als Eingabe gegeben, welches es dann einordnen und beispielsweise die dargestellten Objekte, wie zum Beispiel Ampeln, erkennen soll. Für Deep Learning gibt es verschiedene Architekturen.

2.6.2 *Convolutional Neural Network*

Convolutional Neural Networks ([CNNs](#)) werden vor allem zur Bilderkennung und [NLP](#) verwendet. Sie basieren auf dem Aufbau des visuellen Kortex des Menschen, was sich in der dreidimensionalen Anordnung der Knoten verdeutlicht, welche in [Abb. 2.4](#) dargestellt ist. Die Neuronen sind auf jeder Schicht, welche hier auch *Filter* oder *Kern* genannt werden, zweidimensional aufgebaut, aber nicht wie sonst zu jedem Neuron der nächsten Schicht verbunden, sondern nur zu einem *lokalen rezeptiven Feld* (LeCun, Bengio et al., 2015), also einer kleineren Gruppe an Knoten, in der nächsten Schicht zusammengeschlossen. Dies verkürzt die Trainingszeit, da jeder Knoten nur einen Teil der Eingabe sieht, aber alle nach dem gleichen Muster suchen. Die kürzere Trainingszeit ist auch dadurch bedingt, dass manche Schichten

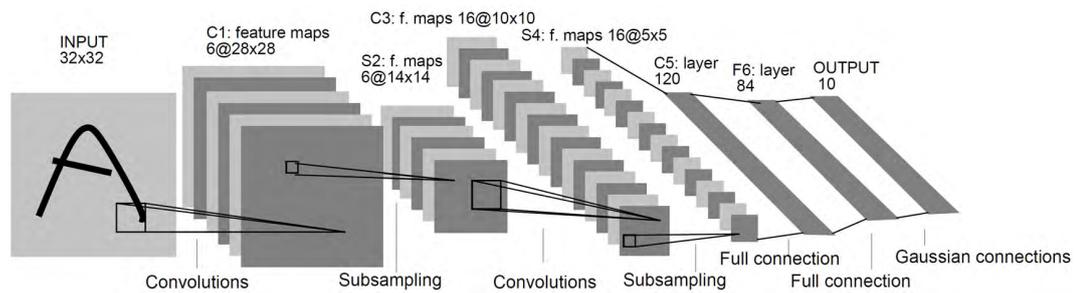


Abbildung 2.4: Die Struktur eines CNN. Quelle: Shrestha und Mahmood (2019).

ihre Gewichte teilen. Für die eigentliche Klassifikation der Eingabe sind dann allerdings die hinteren Schichten vollständig verbunden.

2.6.3 Recurrent Neural Network

In einem Recurrent Neural Network (RNN) formen die Knoten einen Kreis, sodass die Ausgabe der einen Schicht die Eingabe einer anderen wird, wodurch dem NN Informationen über die Ausgaben aus den vorherigen Schichten bekannt sind (siehe Abb. 2.5). Typischerweise gibt es in solchen NNs nur eine Schicht. Ein Vorteil dieser Art von NNs ist, dass sowohl eine Reihe an Eingaben als auch eine Reihe an Ausgaben möglich wird. Dies ist besonders für Video- oder Audioverarbeitung äußerst praktisch (Shrestha und Mahmood, 2019).

RNNs werden außerdem für Encoder-Decoder-Modelle (Sutskever, Vinyals et al., 2014) verwendet. Diese bestehen aus zwei RNNs, eines von ihnen ist ein Encoder (Verschlüsseler), das andere ein Decoder (Entschlüsseler). Besonders ist, dass die einzelnen NNs nicht einzeln, sondern das gesamte System zusammen trainiert wird (Cho, Van Merriënboer et al., 2014). Sie wurden zuerst für maschinelle Übersetzung eingesetzt, sind jetzt aber als Methode etabliert.

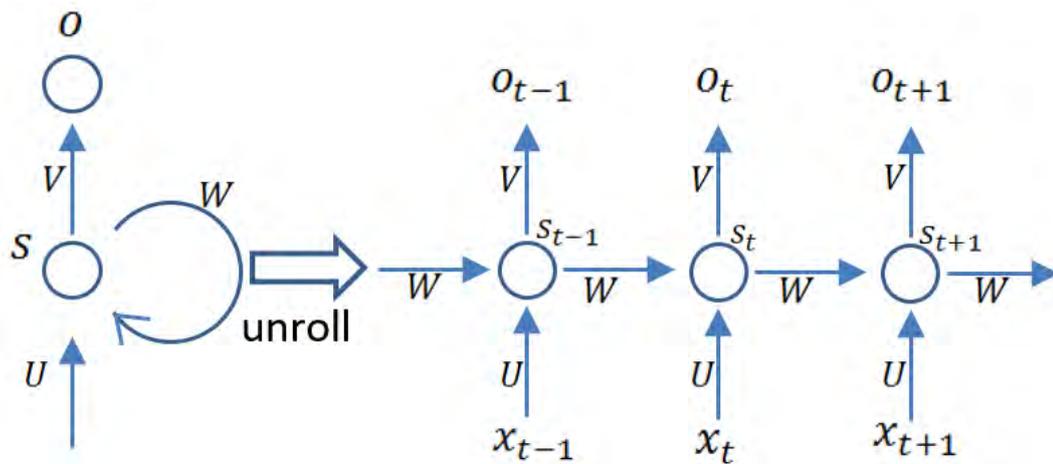


Abbildung 2.5: Die Struktur eines RNN. Quelle: Shrestha und Mahmood (2019).

Der sogenannte Attention-Mechanismus (Aufmerksamkeitsmechanismus) basiert auf dem Encoder-Decoder-Modell. Es löst das Problem des Modells, dass dort alle Informationen der Eingabe in einen Vektor bestimmter Länge getan werden müssen und somit Probleme bei langen Sätzen wie diesem hier entstehen können. Der Attention-Mechanismus löst dieses Problem, indem bei jeder Iteration ein variabler Kontextvektor erzeugt wird (Bahdanau, Cho et al., 2014). Dieser ähnelt sehr der menschlichen Aufmerksamkeit, was den Namen des Mechanismus erklärt. Für die RNN werden statt LSTMs GRUs verwendet.

2.6.4 Long short-term memory und Gated Research Units

Long short-term memory (LSTM), zu Deutsch „langes Kurzzeitgedächtnis“, ist eine Form von RNNs (Hochreiter und Schmidhuber, 1997). Es wird zum Beispiel von Google und Amazon für deren Stimmenerkennungssoftware genutzt. Im Gegensatz zu RNNs hat LSTM nicht das Problem, dass der Gradi-

ent langsam verringert wird. Dies ist so, weil es aus mehreren Zellen, durch die eingegeben, ausgegeben und vergessen werden kann, besteht. Gated Research Unit (GRU) sind ein anderer Typ simpler RNNs und wurden von Chung, Gulcehre et al. (2014) entwickelt.

2.6.5 *Training von neuronalen Netzen*

Neuronale Netze basieren darauf, trainiert zu werden. Bei diesem Training werden die Gewichte der einzelnen Verbindungen zufällig verändert, so dass am Ende selbst die programmierende Person nicht weiß, wie genau es funktioniert. Es gibt viele Trainingstypen, aber es wird vor allem zwischen zwei unterschieden: *überwachtes* (supervised) und *unüberwachtes* (unsupervised) Lernen (Mahesh, 2020). Diese haben jeweils noch Untertypen, diese werden hier allerdings nicht weiter betrachtet. Beim überwachten Lernen hat das Programm eine kurtierte und annotierte Eingabe, das Programm erhält ein Paar aus Eingabe und erwarteter Ausgabe. Um zum Beispiel Bücher digitalisieren zu können sind viele Trainingsdaten nötig. Dafür hat die Carnegie Mellon University in Pittsburgh reCAPTCHA entwickelt, welches in seiner ersten Version ein Wort zeigte, welches der Benutzer abschreiben sollte (Von Ahn, Maurer et al., 2008). Google kaufte es, um damit Bücher und Hausnummern automatisch zu digitalisieren. Die Möglichkeiten dieser Technologie erstrecken sich bis hin zum autonomen Fahren, wo zur Erkennung von Straßenschildern, Ampeln und anderen Verkehrsteilnehmern enorme Datenmengen benötigt werden. Google hat die Eingaben, muss sich für die Ausgaben allerdings an seine Nutzer wenden (Walch, Colley et al., 2019). Ohne die Ausgaben hätte man nämlich unüberwachtes Lernen. Dieses gibt dem zu trainierenden NN keine Lösung vor und ist eher

zur allgemeinen Mustererkennung gut. Ein weiterer Trainingsansatz ist das Zero-Shot Learning (ZSL). Hier soll das Programm anhand einer Beschreibung von Objekten lernen, sie zu erkennen. Der große Vorteil hieran ist, dass keine annotierten Daten benötigt werden, um auch seltene oder neue Vorkommnisse sicher erkennen zu können. Im Gegensatz zum überwachten Lernen sind die Daten hier nicht gekennzeichnet, d.h. dem Programm wird nicht übermittelt, was in der Eingabe ist (Rahman, Khan et al., 2018). Ähnlich zum ZSL ist das Few-Shot Learning (FSL). Hierbei wird ein Modell vortrainiert und später im *fine-tuning* von der generellen auf die spezifische Ebene geführt. Es wird beispielsweise von BERT genutzt.

2.6.6 Regelbasiertes System

Vor neuronalen Netzen wurden vor allem die äußerst strikten regelbasierten Systeme zur Lösung von komplexen Problemen genutzt. Sie bestehen aus einer Wissensbasis und einem Regelinterpreter. Die Wissensbasis enthält Regeln, welche aus einer *Atendenz* (Bedingung) und einer *Konsequenz* (Resultat) bestehen, und Fakten. Wenn die Bedingung einer Regel wahr wird, folgt das Resultat. Regelbasierte Systeme ermöglichen ausführliche Begründungen, welche stark dem subjektiven menschlichen Denken ähneln. Wissen lässt sich außerdem häufig leicht in einer solchen Regelform ausdrücken. Es gelten nicht immer alle Regeln und Fakten, sie können dynamisch ein- und ausgeschaltet werden (Hayes-Roth, 1985). Regelbasierte Systeme werden heute weniger benutzt, in Verbindung mit neuronalen Netzen können sie jedoch immer noch sehr nützlich sein.

STAND DER FORSCHUNG

3.1 DIE ONTOLOGIE: SNIK

Das semantische Netz des Informationsmanagements im Krankenhaus ([SNIK](#)) ist eine am Institut für Medizinische Informatik, Statistik und Epidemiologie ([IMISE](#)) der Universität Leipzig entwickelte, die Domäne des Informationsmanagements im Krankenhaus betreffende Ontologie (Schaaf, Jahn et al., 2015). Sie behandelt Wissen über Krankenhausinformationssysteme und deren Management. Dieses wurde aus drei Lehrbüchern (Winter, Haux et al. (2011), Ammenwerth, Haux et al. (2014) und Heinrich, Riedl et al. (2014)) manuell extrahiert und in RDF modelliert. Des Weiteren wurde auch noch ein Interview mit dem Leiter des Informationsmanagements des Universitätsklinikum Leipzigs, Stefan Smers, geführt.

3.1.1 „Eine Ontologie von Ontologien“ – Die Architektur SNIKs

Die einzelnen Bücher liegen jeweils in einer eigenen Subontologie vor. Es gibt also drei große Teilontologien der [SNIK](#)-Ontologie, die die Präfixe *bb*, *ob* und *he* haben. Das Interview mit dem Leiter des Informationsmanagements befindet sich in der Subontologie mit dem Präfix *ciox*. Zur Beschreibung der Relationen gibt es eine Ontologie mit dem Präfix *meta*, in der größtenteils Properties stehen. Diese enthalten dann jeweils Klassen von Individu-

SNIK Meta-Modell

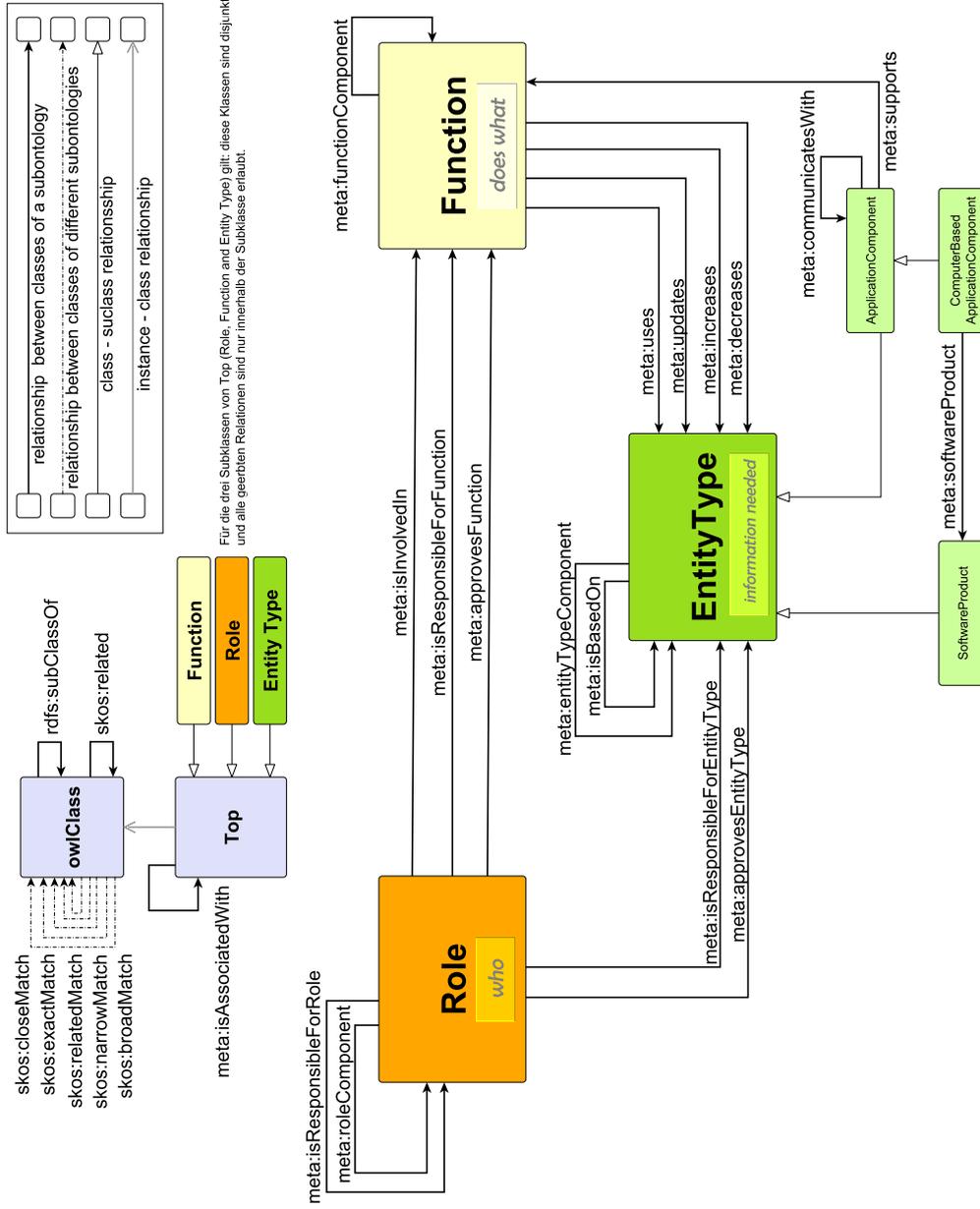


Abbildung 3-1: Das SNIK Metamodell Version 8. Quelle: https://www.snik.eu/public/SNIK_Metamodel1_V8.svg

en, z.B. `bb:ChiefInformationOfficer` anstelle von `ex:ErikaMustermann`. Es geht hier nämlich, wie gesagt, um Textbuchwissen und nicht um spezielle Vorgänge (Höffner, Jahn et al., 2017).

Die Entitäten `SNIKs` in einer Subontologie sind, wie in Abb. 3.1 zu sehen, primär nach ihrem Typ gegliedert. Es gibt Rollen, die eine Person einnehmen kann, beispielsweise `bb:ChiefInformationOfficer`, Aufgaben, die eine Rolle hat, wie z.B. `bb:HISBudgetPlanning`, und Objekttypen, die Informationen, die für diese Aufgabe benötigt werden, repräsentieren, wie z.B. `bb:AnnualITBudget`. Zwischen diesen gibt es Relationen, die deren Beziehung beschreiben.

Die Subontologien sind theoretisch unabhängig voneinander. Sie sind größtenteils dann verbunden, wenn sie ähnliche Prinzipien wie ein anderer Punkt einer anderen Subontologie darstellen. Für das Tripel mit dem Subjekt `bb:ChiefInformationOfficer` gibt es das Prädikat `skos:closeMatch` mit dem Objekt `ciox:ChiefInformationOfficer`. Es gibt dieses Tripel auch mit diesem als Subjekt und dem anderen als Objekt.

3.1.2 *Der SNIK-Graph*

Der `SNIK-Graph`¹ ist die Visualisierung von `SNIK`. Hier werden die einzelnen Entitäten nach ihrer entsprechenden Subontologie gefärbt visuell dargestellt. Die beschrifteten Verbindungen bzw. Kanten zwischen den Formen stellen Relationen dar, welche, wenn sie zwischen zwei Entitäten unterschiedlicher Subontologien verlaufen, keine Richtung haben. Entitäten werden durch Dreiecke, Quadrate oder Kreise dargestellt und vertreten die verschiedenen Typen.

¹ <https://www.snik.eu/graph>

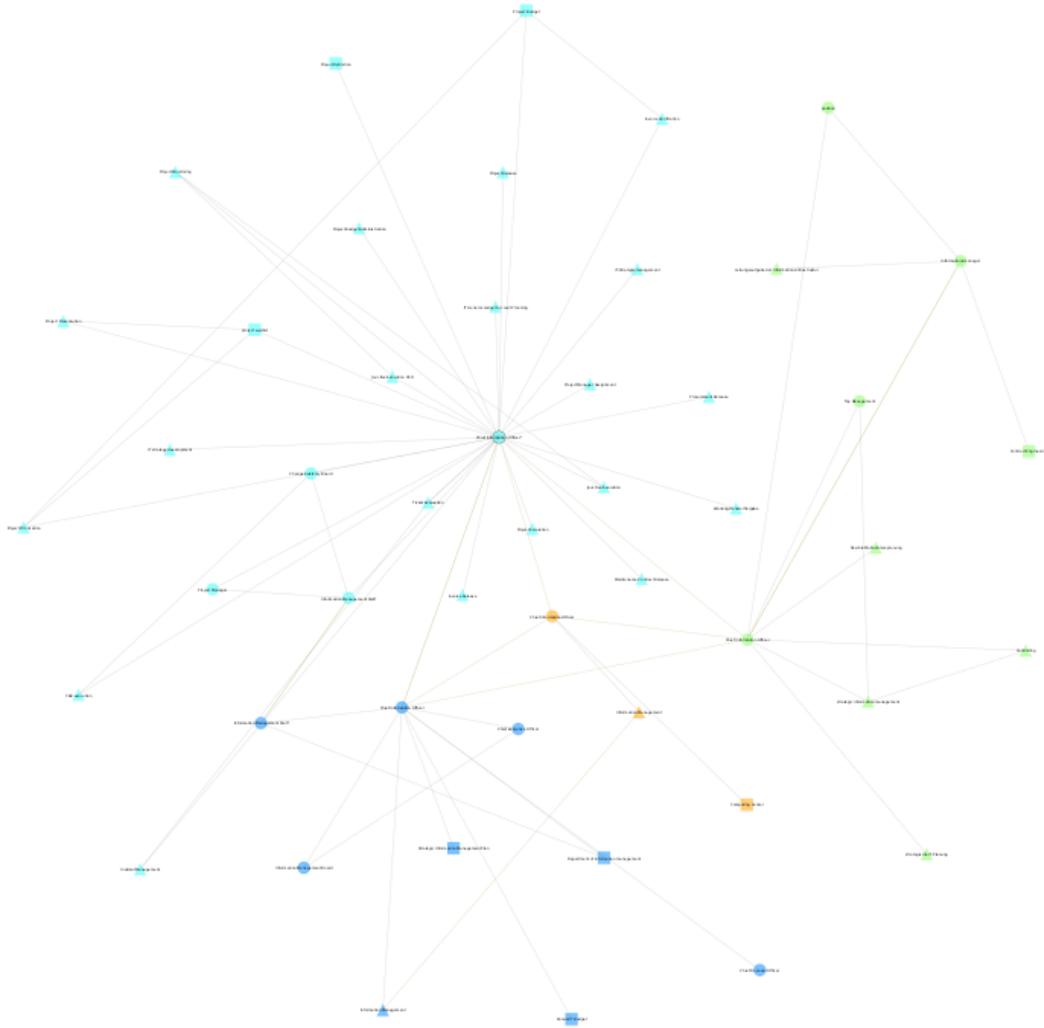


Abbildung 3.2: Ein Ausschnitt des SNIK-Graphen.

Screenshot von <https://www.snik.eu/graph>, abgerufen am 18. Januar 2023

In Abb. 3.2 wird ein Ausschnitt **SNIKs** dargestellt, und zwar die Klasse des Chief Information Officers in vier Teilontologien, sowie die Verbindungen zwischen ihnen und angrenzenden Klassen. Diese Verbindungen sind meist gerichtet, es sei denn, es handelt sich um Beziehungen zwischen den Ontologien. Solche Beziehungen gehen in beide Richtungen, weshalb einfach eine Linie und kein Pfeil angezeigt wird.

3.2 STICHWORTSUCHE

Die Stichwortsuche ist wohl mit die am weitesten verbreitete Form, um sich Informationen anzueignen. Im Gegensatz zu Question Answering werden hier statt ganzen Sätzen nur einzelne Stichwörter eingegeben, womit letztendlich Ergebnisse angezeigt werden. Das Problem (bzw. der Ansatz) dabei ist nicht unbedingt, den Nutzer zu verstehen, sondern in abertausenden Möglichkeiten die möglichst relevantesten herauszusuchen. Dazu werden oft Listen, wo die Antwortmöglichkeiten indiziert sind, verwendet. Stichwortsuche wird nicht nur im **WWW**, sondern auch in vielen anderen Bereichen eingesetzt. Manche **QA**-Systeme verwenden es z.B. auch für bestimmte Module.

3.2.1 *Elasticsearch*

Elasticsearch² ist ein Open-Source Programm, das einfaches Indizieren, Suchen und Analysieren von Daten ermöglichen soll (Andhavarapu, 2017, S. 8). Es kann verschiedenste Datentypen durchsuchen, auch Text und Zahlen. Dazu serialisiert es Dokumente in JavaScript Object Notation (**JSON**),

² <https://www.elastic.co>

einer auch für die Programmiersprache JavaScript, welche häufig auf Webseiten verwendet wird, lesbare Notation. In dieser Serialisierung nutzt es einen *invertierten Index*, d.h. es indiziert das Dokument für alle verschiedenen vorkommenden Wörter (Yan, Ding et al., 2009). Das Wort bzw. die Zeichenkette ist also ein Schlüssel mit den Dokumenten als Wert. Nehme man z.B. die beiden Sätze „War is peace.“ und „Freedom is slavery.“ (Orwell, 1948, S. 6) als eigene Dokumente, würde es wie in Tabelle 3.1 aussehen. Das Wort „is“ kommt sowohl in Dokument 1 und 2, das Wort „war“ nur in Satz 1 und das Wort „freedom“ in nur in Satz 2. Die Zahl nach dem Doppelpunkt zeigt die Position im Satz an, „peace“ ist z.B. an dritter Stelle im ersten Satz. Elasticsearch verwendet Representational State Transfer (REST)-Application

Schlüssel	Wert(e)
war	1:1
is	1:2,2:2
peace	1:3
freedom	2:1
slavery	2:3

Tabelle 3.1: Beispiel für invertierte Indizes

Programming Interfaces (APIs), also Programmierschnittstellen, die mittels HTTP-Anfragen mit Seiten im Extensible Markup Language (XML)-Format, wie z.B. HTML, kommunizieren. Mit diesen kann es in der Tabelle nach Informationen suchen. Das tut es, indem es die Seiten und Positionen, auf denen die eingegebenen Wörter stehen, mithilfe der invertierten Indizes herausucht. Je näher die eingegebenen Wörter aneinanderstehen, und je eher die

gefundenen Wörter mit den eingegebenen übereinstimmen, desto besser ist das Ergebnis und desto eher wird es angezeigt (Andhavarapu, 2017, S. 23). Es können auch verschiedene Synonyme und phonetisch ähnlich klingende Wörter ausprobiert werden.

3.3 QUESTION ANSWERING-SYSTEME

Zur Recherche wurden verschiedene Surveys bezüglich [KBQA](#) und [CDQA](#) verwendet. Außerdem wurden über die Herausforderung Question Answering over Linked Data ([QALD](#)) weitere Systeme evaluiert.

Bei der [QALD-Challenge](#) werden [QA](#)-Systeme anhand eines Benchmark objektiv bewertet. Diese fand schon mehrmals statt, die zur Zeit dieser Arbeit aktuelle ist [QALD-9](#) (Ngomo, 2018)³, für [QALD-10](#) können jedoch bereits Systeme eingereicht werden⁴.

In jeder dieser Versionen gibt es unterschiedliche Trainingsdatensätze und Teilnehmer. Die Teilnehmer müssen zu dem jeweiligen Trainingsdatensatz natürlichsprachige Fragen in [SPARQL](#)-Abfragen umformen, folglich das grundlegende Problem des Question Answerings lösen.

Der Trainingsdatensatz für [QALD-9](#) ist DBpedia, [QALD-10](#) nutzt hingegen WikiData. [QALD-10](#) setzt außerdem einen größeren Fokus auf Multilingualismus, indem es einerseits auch eine besondere Version von [QALD-9-Plus](#) als Trainingsfragen anbietet, andererseits jedoch auch in der ersten Aufgabe

³ [QALD-9-Plus](#) nahm die Fragen aus [QALD-9](#) und übersetzte diese in acht verschiedene Sprachen (Deutsch, Französisch, Russisch, Ukrainisch, Belarussisch, Armenisch, Baschkirisch, Litauisch), um Entwicklung im Bereich der Multilingualismus von Systemen voranzutreiben, gilt aber nicht als offizielle Version der [QALD-Challenge](#) (Perevalov, Diefenbach et al., 2022).

⁴ <https://www.nliwod.org/challenge>, abgerufen am 17. Januar 2023

die Beantwortung von in vier Sprachen (Russisch, Chinesisch, Deutsch und Englisch) vorliegenden Fragen fordert.

DBpedia ist eine Wissensbasis, welche aus den Daten Wikipedias besteht. Es ist das Paradebeispiel für Linked Data und wird häufig zum Training von QA-Programmen genutzt. Die Daten sind frei verfügbar und in verschiedenen Serialisierungen, wie z.B. im Terse RDF Triple Language (*Turtle*)-Format, verfügbar. Über *SPARQL* können auch direkt online Abfragen getätigt werden. DBpedia ist außerdem in mehreren Sprachen verfügbar (Lehmann, Isele et al., 2015).

3.3.1 *Leaderboard*

Das Question Answering-Leaderboard⁵ (Perevalov, Yan et al., 2022) hat es sich zur Aufgabe gemacht, dem häufig als sehr uneinheitlich und unübersichtlich (Diefenbach, 2018b) beschriebenen Feld des *KBQA* eine vereinheitlichte Liste mit verfügbaren QA-Systemen zu geben. Hier werden Fragenkataloge wie *QALD-9* als Benchmarks auf die verschiedenen *KBQA*-Systeme angewendet und die Ergebnisse aufgezeichnet. Des Weiteren sind die 98 Systeme in einer Liste mit dem Namen des Systems, einer Beschreibung und Links zu dem originalen Paper und einer Demonstration, *REST-API* oder dem GitHub-Projekt, falls diese öffentlich verfügbar existieren, ausgestattet. Dies vereinfacht eine systematische Suche enorm, da viele Systeme schwer auffindbar sind, sowohl in der Dokumentation als auch in Form des implementierten Systems selbst.

⁵ <https://github.com/KGQA/leaderboard>

3.3.2 *gAnswer*

*gAnswer*⁶ (Zheng und Zhang, 2019) wurde von dem Wangxuan Institute of Computer Technology entwickelt und arbeitet mit Wissensbasen, um Question Answering-Aufgaben zu lösen. Hierfür werden die Fragen in Unterfragen aufgespalten und daraus je ein Syntaxbaum erstellt. Es erzielte durch eine durchdachte Vorbereitung der Trainingsdaten sehr schnelle Trainingszeiten bei niedrigem Speicherverbrauch und hoher Genauigkeit bei der Beantwortung von Fragen. Das System hat außerdem erkannt, dass man, um effizientes Question Answering zu erreichen, Subgraphen vergleichen kann und somit auch einen neuen Ansatz für die Homonymität hat. Das bedeutet, dass Wörter, die Verschiedenes bedeuten können, erst nach anfänglicher Lokalisierung des Kontexts betrachtet werden. Es werden also zuerst Wörter, deren Bedeutung eindeutig ist, betrachtet, und von da aus die kürzeste Verbindung zu einer Bedeutung des fragwürdigen Wortes (Zou, Huang et al., 2014). Darauf bauen viele andere QA-Systeme auf.

3.3.3 *DeepPavlov*

*DeepPavlov*⁷ (Burtsev, Seliverstov et al., 2018) ist eine open-source Bibliothek zur Entwicklung von Dialogsystemen. Es ist in *Models* und *Skills* organisiert. Das System ist hochflexibel und auf verschiedenste Aufgaben ausgelegt, vor allem aber Dialogsysteme bzw. Chat Bots. Für QA gibt es bisher nur Ansätze bezüglich ODQA. Es können keine eigenen Daten eingegeben werden, weshalb es hier nicht verwendet wurde.

⁶ <https://github.com/pkumod/gAnswer>

⁷ <https://www.deeppavlov.ai>

3.3.3.1 Architektur

Ein *Modell* ist eine in TensorFlow (Abadi, Agarwal et al., 2015), einer Schnittstelle für maschinelles Lernen, implementierte Funktion einer NLP-Pipeline, die sowohl ein neuronales Netz als auch ein nichtneuronales oder regelbasiertes System sein kann. Modelle können auch andere Modelle enthalten. Eine *Fähigkeit* besteht aus Modellen, jedoch kann sie nur Zeichenketten als Ein-/Ausgabe haben. Sie werden deshalb häufig im Dialog verwendet. Die Modelle in Fähigkeiten werden über einen *Chainer* verbunden, der die Konfigurationsdatei einliest und so die Parameter der Modelle festsetzt. Fähigkeiten und Modelle werden gleich implementiert und unterscheiden sich nur in den unterschiedlichen Ein- und Ausgabemöglichkeiten. Mehrere Fähigkeiten formen einen *Agenten*, wie in Abb. 3.3 sichtbar ist. Ein Agent kann die verschiedenen Fähigkeiten, aus denen er besteht, in einer Unterhaltung mit dem Benutzer verwenden und zwischen ihnen wechseln.

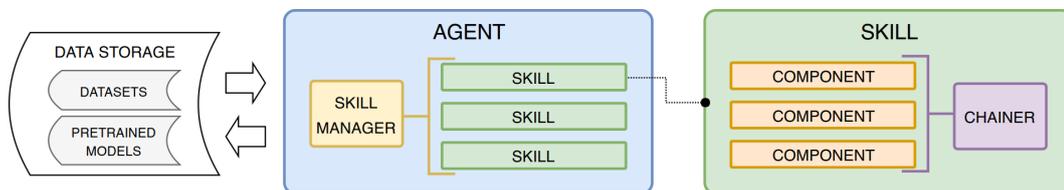


Abbildung 3.3: Die Architektur von DeepPavlov. Quelle: Burtsev, Seliverstov et al. (2018)

3.3.4 TeBaQA

TeBaQA⁸ (Vollmers, Jalota et al., 2021) wird von der Forschungsgruppe DICE an der Universität Paderborn entwickelt. Es basiert darauf, Fragen

⁸ <https://github.com/dice-group/TeBaQA>

anhand ihrer Struktur zu gruppieren. Dabei werden isomorphe [SPARQL](#)-Anfragen, d.h. Anfragen gleicher Struktur, einer Vorlage zugeordnet. Somit können schnellere Antwortzeiten ermöglicht werden. Mit Struktur ist die Form der RDF-Relationen gemeint, wobei diese als Kanten zwischen den verschiedenen Informationen gesehen werden.

3.3.4.1 *Architektur*

TeBaQA wird in fünf Phasen aufgebaut. In der ersten Phase, dem *Preprocessing* (Vorverarbeitung), werden Wörter, die keinen semantischen Wert und somit keine Information tragen, entfernt. Das können etwa Artikel sein, wodurch die Assoziation der Wörter mit vielen unzusammenhängenden Einträgen vermieden wird. Damit solche Wörter, die auch Teil der Entitäten, wie z.B. „the“ in `bb:AdaptabilityOfTheHIS`, nicht auch dort herausgefiltert werden, müssen die Wörter gruppiert und die resultierenden Gruppen überprüft werden. Die zweite Phase befasst sich mit der Isomorphie der Abfragegraphen und der Vorlagenklassifizierung. Hier wird die zu nutzende Vorlage der Abfrage identifiziert und darüber Eigenschaften der Frage, also z.B. Fragewörter, Anzahl der Nomen, mindestens benötigte Tripel oder ob Personen referenziert werden. Die Einordnung in eine Vorlage geschieht, wie vorher schon erwähnt, über die Struktur der [SPARQL](#)-Abfrage. Die Reihenfolge der Elemente wird hierbei nicht betrachtet. Beispielsweise resultiert die Frage „Wofür ist die Leiterin des Informationsmanagements zuständig?“ in einer Abfrage mit der gleichen Struktur wie die Abfrage der Frage „Wovon wird 3LGM² erzeugt?“. Die [SPARQL](#)-Abfrage könnte bei beiden so aussehen, bei der ersten Abfrage wäre das Individuum im Subjekt `bb:ChiefInformationOfficer` und das Prädikat

`meta:isResponsibleForEntityType`, bei der zweiten Frage das Subjekt `bb:3LGM2` und das Prädikat `meta:isBasedOn`.

```
PREFIX bb: <http://snik.eu/ontology/bb/>
PREFIX meta: <http://snik.eu/ontology/meta/>
SELECT ?uri
WHERE
  { <Subjekt> <Prädikat> ?uri }
```

Das Ergebnis, ?uri, ist `bb:AnnualITBudget` bzw. `bb:UmlClassDiagram`, also beide Male eine Entität. Wenn die Vorlage zugeordnet wurde, können Eigenschaften der Frage, wie oben beschrieben, in einen Vektor getan und weiterverwendet werden. Im dritten Schritt werden speziell referenzierte Individuen und Klassen identifiziert, also nicht nur wie im 2. Schritt generelle Informationen über den Inhalt der Frage. Hier werden auch verschiedene Synonyme der eingegebenen Wörter untersucht, um ein möglichst genaues Ergebnis zu erreichen. Im vierten Schritt wird die Anfrage geschrieben, also die Vorlage gefüllt und eventuelle Parameter gesetzt. Die letzte Phase behandelt das Auswählen der Lösung, die dem Nutzer übergeben werden soll. Dazu wird wieder auf das Fragewort geblickt, um herauszufinden, in was für einer Form, z.B. einem Datum, der Nutzer die Antwort erwartet. Das erste Substantiv nach dem Fragewort wird überprüft, um den Numerus der Frage zu erkennen. Des Weiteren werden die Lösungskandidaten auf ihre Ähnlichkeit mit der Frage überprüft, woraus eine Bewertung und letztendlich die zu präsentierende Antwort berechnet wird.

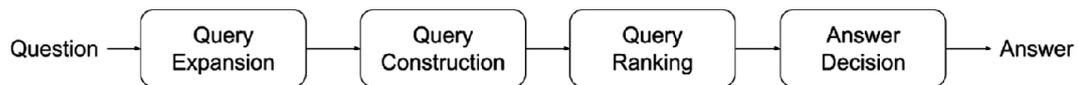


Abbildung 3.4: Arbeitsablauf von QAnswer KG. Quelle: Diefenbach, Giménez-García et al. (2020).

3.3.5 QAnswer KG

QAnswer KG (Diefenbach, Giménez-García et al., 2020) ist ein an der Universität Saint-Étienne entwickeltes QA-System, das vor allem das von Diefenbach (2018b) genannte Problem der fehlenden Portabilität in diesem Feld lösen soll: Viele Systeme wurden als Forschungsprojekt für Benchmarks wie QALD-9 geschrieben und sind deshalb nur mit großen Datensätzen wie DBpedia oder Wikidata kompatibel. Hier ist es jedoch möglich, auf einer Website einfach eigene Daten hochzuladen und dort QA zu betreiben. Die Modelle müssen aber noch auf die Fragen trainiert werden, das vortrainierte Modell soll nur Sprache an sich verstehen können und nicht den speziellen Datensatz. Alles in allem kann man dennoch sagen, dass das System vergleichsweise leicht zu verwenden ist. Selbst ein fast nicht trainiertes System bietet oft noch akzeptable Antworten. Es ist leider nicht open-source, steht online⁹ aber frei zur Verfügung und wird es nach Nachfrage auch noch bis mindestens 2024 bleiben.

3.3.5.1 Architektur

QAnswer KG beantwortet eine Frage in vier Schritten, die in Abb. 3.4 dargestellt sind.

⁹ <https://app.qanswer.ai/>

Zuerst werden alle sogenannten *N-Gramme* gebildet, d.h. alle möglichen Kombinationen von aufeinanderfolgenden Wörtern mit der maximalen Länge n . Die maximale Länge hier ist die Anzahl der Wörter im Satz, d.h. der Satz selbst ist auch ein N-Gramm. Wenn man zum Beispiel die Frage „Is war peace?“ stellt, sucht das Programm für jedes Wort sowie für die Kombinationen „is war“, „war peace“ und „is war peace“ mithilfe des auf natürliche Sprache vortrainierten Modells die am ehesten passenden Werte im Datensatz heraus. Dabei werden normalerweise auch viele falsche Ressourcen ausgewählt, da besonders bei längeren Fragen viele N-Gramme existieren.

Danach werden aus all diesen Werten alle möglichen [SPARQL](#)-Anfragen erstellt. Dabei werden auch die falschen Interpretationen weitergeführt, jedoch sollte auch die richtige dabei sein.

Deshalb müssen die Abfragen im dritten Schritt bewertet werden. Dazu werden mithilfe von verschiedenen Parametern und einer künstlichen Intelligenz die Kandidaten evaluiert. Parameter sind etwa die Anzahl der in der Abfrage verwendeten mit einer Ressource assoziierten Wörter verglichen mit der Anzahl der Wörter in der Frage, oder die Ähnlichkeit der Label mit den entsprechend ausgewählten Wörtern. Label werden meist über das [RDF-Property](#) `rdfs:label` festgelegt.

Dadurch soll die Interpretation, die der Intention der Frage am nächsten kommt, am höchsten gewertet sein. Diese wird im letzten Schritt analysiert und es wird entschieden, ob es die richtige Antwort ist. Damit wird ein „confidence score“, zu Deutsch etwa „Vertrauenswert“, erstellt. Sollte dieser unter 50% liegen, gilt die Frage als falsch beantwortet. Sie wird jedoch trotzdem ausgegeben.

3.3.5.2 *Training und maschinelles Lernen*

Trainiert werden können bei QAnswer die in Abb. 3.4 dargestellten Schritte drei und vier, also *Query Ranking* und *Answer Decision*. Es kann also nur beeinflusst werden, welche Antwort QAnswer warum auswählt, nicht aber, welche Antwortmöglichkeiten es erstellt.

Mittels fine-tuning werden die vortrainierten Modelle für diese beiden Schritte mittels gesammelten Feedbacks adjustiert. Die Modelle nehmen die Anzahl der Variablen, Tripel in den SPARQL-Abfragen und Wörter der natürlichsprachigen Fragen, die durch die Ressourcen abgebildet sind, als Eingabewerte auf. Außerdem gibt zwei weitere Kriterien, die verwendet werden, sofern es erforderlich ist. Diese sind die Unterschiedlichkeit der eingegebenen Wörter verglichen mit den Labels der Ressourcen und die Summe der Relevanz von Ressourcen in der Abfrage. Die Relevanz wird durch die Anzahl ein- und ausgehender Verbindungen oder manuelle Zuweisungen festgelegt. Idealerweise werden alle fünf Kriterien verwendet.

Da alle diese Kriterien letztendlich in Zahlenwerten zum Ausdruck kommen, können diese zu einer neuen kombiniert werden. Hier kommt das Training ins Spiel. Dieses bestimmt nämlich die Wichtung der einzelnen Zahlen beim Zusammenrechnen. Die Wichtung kommt als Faktor vor dem Wert des Kriteriums zum Ausdruck, die Kriterien werden linear verrechnet, also addiert.

Sofern Trainingsdaten verfügbar sind, werden die Abfragen mittels des F-Maßes (siehe Abschnitt 4.1.1) zusätzlich bewertet (Diefenbach, 2018a).

3.3.5.3 Vorteile, Grenzen und Probleme

Im Gegensatz zu anderen Ansätzen verwendet QAnswer KG kein herkömmliches NLP, wo auf sprachliche Eigenschaften acht gegeben wird, wie man bei TeBaQA gut sehen kann. Dies führt dazu, dass auch Stichwortsuche und Fragen mit fragwürdiger Grammatik und teils auch Rechtschreibung beantwortet werden können, was in der Praxis sehr hilfreich ist. Nicht jede die Applikation benutzende Person ist in diesen Bereichen sicher, besonders, wenn das Programm auch im öffentlichen Bereich einsetzbar sein soll. Ein weiterer ausschlaggebender Punkt ist, dass dadurch Fragen in verschiedenen Sprachen gestellt werden können, solange die Ontologie in diesen verfügbar ist. Außerdem können verschiedene Ontologien gleichzeitig durchsucht werden, was insbesondere in Zeiten von Linked Open Data hilfreich ist. Systeme wie gAnswer benötigen die Daten auch in einer speziellen Form, um sie verarbeiten zu können, hier reichen die RDF-Daten aus. Das hilft den Entwicklern, die dieses System verwenden wollen, weiter, indem es den Entwicklungsaufwand senkt.

Speziell benötigt QAnswer keine Vorgänge wie *entity linking* benötigt, eine spezielle Form der Daten, die etwa in gAnswer 2 verwendet wird. Allgemein werden keine annotierten Daten benötigt. Des Weiteren benötigt das System aufgrund des ausgiebigen Vortrainings mittels Wörterbücher verschiedener Sprachen keine großen Datenmengen und viele Parameter, um die Genauigkeit brauchbar zu machen, sondern fällt allein auf Rückmeldungen der das System nutzenden Personen zurück, um fine-tuning zu betreiben. Besonders dieser Punkt sichert die Portabilität nicht nur zwischen den Sprachen, sondern auch zwischen verschiedenen Datensätzen und macht QAnswer sehr flexibel.

Diese die Grammatik des Satzes nicht beachtende Methode bringt allerdings auch Nachteile mit sich. Der größte ist wohl, dass das Wissen sich in einer bestimmten Form befinden muss. Es müssen `rdfs:labels` o.ä. verwendet werden, anderes wird nicht erkannt. Des Weiteren müssen die Label eine Annotation, die deren Sprache bestimmt, besitzen, also zum Beispiel „Chief Information Officer“@en, „Leiter des Informationsmanagements“@de für das Label vom `bb:ChiefInformationOfficer`. Bei `SNIK` und den meisten anderen Wissensbasen ist das allerdings kein Problem, da sie es bereits besitzen. Das zieht allerdings auch mit sich, dass andere Informationen, die in der Wissensbasis existieren, aber nicht im Label gespeichert sind, nicht mit betrachtet werden. Eine weitere bedeutende Grenze ist die Limitierung der Komplexität der `SPARQL`-Abfrage auf maximal drei Tripel, sodass besonders komplexe Fragen mit kompliziertem Muster nicht beantwortet werden können. In der Praxis reichen diese allerdings für die meisten Fragen aus. Gäbe es diese Limitation nicht, würde die Beantwortung längerer Fragen aufgrund des Versuches, aus den N-Grammen zu komplexe Muster zu erstellen, vermutlich mit einer geringeren Genauigkeit der Ergebnisse und einer langsameren Geschwindigkeit ausfallen.

QAnswer KG kann, wie man an Benchmarks sieht, mit den bisher existierenden Systemen mithalten, ist jedoch in der Intuitivität und Portabilität weit über ihnen. Mit Training kann auch eine sehr hohe Genauigkeit erzielt werden.

3.3.6 *AskNow QA*

AskNow QA (Dubey, Dasgupta et al., 2016) ist ein an der Universität Bonn entwickeltes System, welches darauf basiert, die eingegebene Frage erst

in eine Normalized Query Structure (NQS), auf Deutsch etwa *normalisierte Fragenstruktur*, umzuformen, also in eine einheitliche Syntaxstruktur. Dazu wird auch *Part-Of-Speech (POS)-Tagging* verwendet. Hier bekommen Wörter *Tags*, welche die (kontextuelle) Funktion und Art des Wortes beschreiben. So bedeutet hier beispielsweise der Tag DT *Determiner*, also ein Wort wie „many“, welches die Anzahl von etwas bestimmen kann. Ein Syntaxbaum mit syntaktischen Tags wurde bereits in Abb. 2.2 gezeigt, diese unterscheiden sich allerdings von POS-Tags. Mehr dazu in Anhang A.

3.3.6.1 Architektur

Das System versucht zuerst, die eingegebene Frage in eine normale Form zu bringen, von welcher aus dann weitergearbeitet werden kann. Zuerst kommt immer ein Wort mit dem POS-Tag WH, also ein Fragewort. Darauf folgt ein Hilfstoken, der die Wörter „is“, „is kind of“, „much“, „might be“ und „does“ enthält. Dieser hilft dabei, sprachliche Ambiguitäten aufzulösen, meistens wird das zu verwendende Wort auch durch das Fragewort vorgegeben. Auf dieses folgt dann mindestens eine, bei zusammengesetzten Fragen mehrere, Intention, eine Relation als Prädikat und eine Eingabe, welche auf eine Ressource zurückführt. Beziehungen und Eingaben kann es bei komplexen Fragen auch mehrere geben, dann aber auch immer in dieser Reihenfolge. Bei diesen folgt vor der Beziehung außerdem immer ein Kausalwort, welches die Suche einschränken kann.

Zuerst wird, wie schon gesagt, die NQS gebildet. Dazu annotiert der *Query Processor* die Wörter mit POS-Tags, der *Auxiliary Relation Handler* extrahiert den Hilfstoken und der *Token Merger* fasst die annotierten Wörter auf Grundlage ihrer POS-Tags zu sinnvollen Wortgruppen zusammen. Diese Wortgruppen werden dann in ihre jeweiligen Funktionen der NQS zeigen.

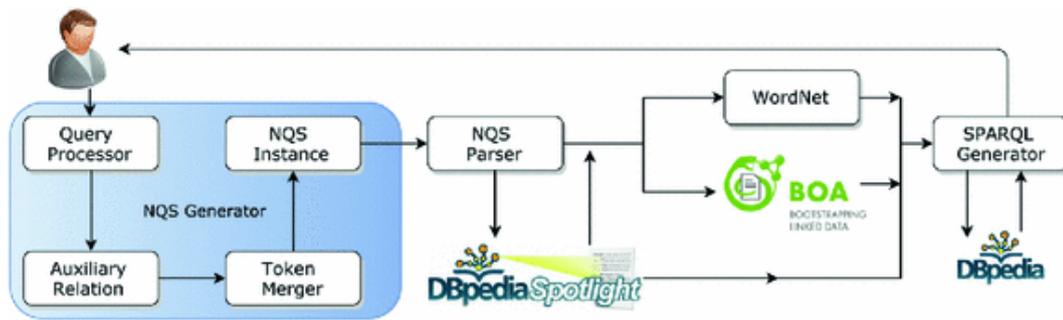


Abbildung 3.5: Die Architektur von AskNow. Quelle: Dubey, Dasgupta et al. (2016)

Sätze werden also so umgeformt, dass sie leicht von *NQS-Parser* gelesen werden können. Dieser analysiert zuerst, welcher Antworttyp erwartet wird, also zum Beispiel ein boolescher Wert, eine Anzahl oder eine Liste. Dies wird mithilfe der Frageintention und des Frageworts vollbracht. Dann werden im *Entity Mapping* den Eingaben und Beziehungen Ressourcen der Wissensbasis zugeordnet. Dazu wird hier DBpedia Spotlight (Mendes, Jakob et al., 2011) verwendet, ein Programm zur Annotation von Textdokumenten mit DBpedia-URIs. Um die Intention mit einem Prädikat zu verbinden, wird versucht, sie über einen einfachen Vergleich, Synonyme (Miller, 1995) oder Sprachmuster (Gerber und Ngomo, 2011) mit einem Label zu assoziieren. Wenn dies fehlschlägt, wird die Frage als unbeantwortbar eingestuft. Die Vorlage zur letztendlichen Erstellung einer SPARQL-Abfrage wird durch die *NQS* vorgegeben, sodass nur noch die Ressourcen eingesetzt werden müssen.

LÖSUNGSANSATZ

4.1 LÖSUNGSANSATZ ZUM PROBLEM P1

Das erste Problem befasst sich mit der Situation, dass ein QA-System im Rahmen dieser Arbeit aufgrund des Aufwands schwer selbst entwickelt werden kann. Stattdessen muss ein bereits existierendes System ausgewählt werden. Hierzu müssen andere Systeme anhand eines Benchmarks evaluiert werden. Für diesen müssen typische Nutzerfragen und Antworten auf diese gesammelt werden.

Dazu bieten sich die Fragen aus den Büchern an, welche schon von Rozeitis (2022) gesammelt und hinsichtlich der Beantwortbarkeit mit der SNIK-Ontologie klassifiziert worden sind. Für diese müssen Antworten herausgesucht und in SPARQL-Abfragen umgeformt werden, um das Training zu ermöglichen. Da dies jedoch nicht unbedingt genug Fragen sind, müssen weitere durch SNIK beantwortbare Fragen formuliert werden. Dies geschieht jedoch aufgrund der Zeitbeschränkung und etwas besserem Training von Modellen nur in der englischen Sprache. Etwa die Hälfte der Fragen wird zum Training, die andere Hälfte zur Kontrolle jeweils vor und nach dem Training genutzt, um Daten hinsichtlich der Effektivität des Trainings und letztendlich des Systems in der Fragenbeantwortung zu erhalten. Da dies aber vermutlich nicht genügend Fragen sind, gibt es noch die Möglichkeit der automatischen Generierung von Fragen, in dem für jedes Tripel auto-

matisch zwei Fragen generiert werden, eine nach dem Objekt und eine nach dem Subjekt. Die Fragen können etwa die Struktur *Fragewort – Prädikat – Subjekt* haben, wobei das Fragewort durch das Prädikat bestimmt werden kann. Dies ist, aufgrund der vergleichsweise geringen Zahl an Prädikaten, leicht realisierbar. Die Antwort ist das Subjekt oder Objekt, das andere von beiden würde jeweils die Rolle des Subjekts in der Frage einnehmen. Je nachdem, ob das Objekt oder Subjekt gegeben ist, ändert sich natürlich auch das Fragewort. Hierbei wird eine riesige Menge an Fragen generiert, die aber alle die gleiche, oder zumindest sehr ähnliche, Struktur haben. Es müssen auch manche Fragen gefiltert oder gar nicht erst generiert werden, etwa die Fragen nach dem Label oder alternativen Labels. Es sollen also keine Beziehungen zwischen Literalen abgefragt werden, nur zwischen Ressourcen. Dadurch kann die große Datenmenge [SNIKS](#) genutzt werden. Es werden 100 in die Test- und der Rest in die Trainingsgruppe eingeteilt.

Um den Effekt des Typs der Fragen in der Trainingsgruppe auszuwerten, werden die beiden Frage-Antwort-Paaren bestehenden Datensätze sowohl je getrennt und einmal zusammen auf den gleichen, untrainierten Datensatz angewandt. Zur Untersuchung des Effekts der Anzahl der Fragen werden in Schritten von zehn Fragen automatisch neue, untrainierte Modelle erstellt und mit der entsprechenden Anzahl von Fragen trainiert.

4.1.1 *Evaluierungsmaße*

Zur Evaluierung der Systeme werden bei den Fragen je die Indikatoren der Genauigkeit (*precision*) p , Trefferquote (*recall*) r und F-Maß f verwendet. Die Genauigkeit lässt sich berechnen indem man die Anzahl richtigen Systemantworten durch die Anzahl aller Systemantworten teilt. Wenn ein

System beispielsweise insgesamt neun Antworten zurück gibt, davon jedoch nur drei richtig sind, hat es eine Genauigkeit von $\frac{1}{3}$. Wenn das System gar keine Antworten zurück gibt, wird eine Genauigkeit von 0 angenommen. Die Trefferquote lässt sich berechnen, indem man die Anzahl der richtigen Systemantworten durch die Anzahl aller richtigen Antworten teilt. Wenn es beispielsweise bei der vorherigen Frage insgesamt sechs richtige Antworten gibt, hat das System eine Trefferquote von $\frac{1}{2}$. Das F-Maß wird durch das harmonische Mittel der Genauigkeit und Präzision berechnet, wäre in diesem Beispiel also $\frac{2}{5}$. Die Werte lassen sich pro Frage also wie folgt berechnen, wenn C die Menge aller möglichen korrekten Antworten ist und O die Menge aller Systemantworten:

$$p = \frac{|O \cap C|}{|O|}$$

$$r = \frac{|O \cap C|}{|C|}$$

$$f = \frac{2pr}{p+r}$$

Am Ende sollen die Werte für alle Fragen mithilfe des arithmetischen Mittels zusammengefasst werden, sodass am Ende die drei Werte jeweils gemittelt vorliegen. Dies ist eine Standardmethode bei der Evaluation von QA-Systemen und wird auch als *macro-average* bezeichnet (Ngomo, 2018).

$$\emptyset p = \frac{1}{n} * \sum_{i=1}^n p_i$$

$$\emptyset r = \frac{1}{n} * \sum_{i=1}^n r_i$$

$$\emptyset f = \frac{1}{n} * \sum_{i=1}^n f_i$$

4.2 LÖSUNGSANSATZ ZUM PROBLEM P2

Das zweite Problem betrifft die mangelhafte Gebrauchstauglichkeit der anderen Möglichkeiten und somit den Zwang, für solch ein System Question Answering zu verwenden. Deshalb muss nach funktionierenden und somit portablen Systemen gesucht werden. Diese sollten nach Möglichkeit trainiert werden können und vor allem nicht nur Wissensbasen wie DBpedia als Quelle verwenden, sondern müssen es auch ermöglichen, eigene Daten zu nutzen, da die Verwendung mit [SNIK](#) sonst unmöglich ist. Zur Recherche sollen, wie bereits in Kapitel [3](#) dargelegt, Surveys und das [QA-Leaderboard](#) verwendet werden. Diese geben über die Güte der Antworten und die grundlegende Funktionsweise Aufschluss, können aber, besonders bei Forschungsprojekten, die nur über eine bestimmte Zeit finanzielle Mittel erhalten und somit die Pflege nicht gewährleistet werden kann, schnell veralten. Besonders bei Benchmarks wie [QALD-10](#) ist es außerdem oft so, dass Systeme nur für große Wissensbasen wie Wikidata und die Fragen im Benchmark erstellt wurden und somit nicht portabel sind. Gerade das Leaderboard ist jedoch eine nützliche Ressource, da hier viele [QA-Systeme](#) mit dem entsprechenden Paper, was sonst auch nicht überall vorliegt, und, falls diese existiert, eine Demo des Systems, was der anfänglichen Evaluation hilft. Ziel ist es, mindestens ein System zu finden, und, falls mehr, aus den gefundenen das beste System auszuwählen. Es wird allerdings nur die Teilontologie aus Winter, Haux et al. ([2011](#)) betrachtet, da es kein gefundenes System es schafft, alle Teilontologien zu verbinden. Aufgrund der Komplexität wurde also nur die Abfrage einer Teilontologie als Ziel gesetzt.

AUSFÜHRUNG DER LÖSUNG

5.1 ERSTELLUNG DES BENCHMARKS

Zur Erstellung eines Benchmarks braucht es zwei Dinge: Natürlichsprachige Fragen und die Antworten als SPARQL-Abfrage. Der Ansatz, um die Fragen zu erhalten, ist, die bereits durch Roszeitis (2022) klassifizierten Fragen aus Winter, Haux et al. (2011) zu verwenden¹. Diese wurden jedoch für ein Quiz auf Basis von SNIK klassifiziert, weshalb wir nun neue Kriterien benötigen.

Die Fragen müssen durch eine SPARQL-Query abfragbar sein, das heißt die möglichen Antworttypen sind Mengen an Ressourcen mit mindestens null Elementen, Literale und einen Wahrheitswert, der sich bei affirmativen Fragen auf die Existenz der Menge oder ihrer Elemente bezieht. Nicht abfragbar sind jedoch Größe die Menge, dies ist pädagogisch nicht sinnvoll, da die Ontologie unvollständig sein könnte. Nach der *open world assumption* ist nicht sämtliches existierendes Wissen in SNIK vorhanden, es ist also nur ein Teil des Existenten in der Ontologie. Wenn beispielsweise nach der Anzahl unterschiedlicher Arten von Architekturen von Informationssystemen im Gesundheitswesen gefragt würde, sollte das System „zwei“ antworten. Es könnte jedoch noch mehr geben, weshalb solche Fragen nicht unterstützt werden sollen. Des Weiteren sollen keine Aggregate, also Operationen wie

¹ Die Fragen sind auch in Tabelle B.1 zu finden.

Summe, Durchschnitt, o.ä. abfragbar sein, da **SNIK** keine statistischen Daten enthält. So ergeben solche Fragen aber keinen Sinn. Zuletzt soll **SNIK** auch nicht Sachverhalte erklären können, auch, weil nach dem aktuellen Forschungsstand solches nicht möglich ist. Dafür bräuchte man eine künstliche Intelligenz, welche Sachverhalte verstehen und in eigenen Worten wiedergeben kann. Außerdem ist das nicht das Ziel dieser Arbeit. Darauf basierend ist feststellbar, dass nur faktische Fragen beantwortbar sein sollen, also Fragen, die mit Fakten beantwortet werden. Andere Fragetypen sind solche wie temporale Fragen, welche sich mit Zeitwerten befassen, solche gibt es in **SNIK** jedoch nicht. Prozedurale Fragen, meist, aber nicht immer, erkennbar am Schlüsselwort „wie“. Sie erhalten Prozesse oder Erklärungen von Schrittabfolgen als Antworten. Sie sind, genau wie kausale Fragen, welche mit einem Grund o.ä. beantwortet werden, aus den Tripeln schlecht ableitbar. Kausale Fragen haben oft das Fragewort „Warum“. Geographische Fragen sind, wie temporale Fragen, sinnlos, da solche Daten hier nicht gespeichert werden. Häufige Fragen sind auch Vergleiche, oder auch evaluierende Fragen. Hier sollen Gemeinsamkeiten und Unterschiede festzustellen. Man kann sie also als zusammengesetzte Frage aus drei Teilfragen verstehen. Angenommen man soll zwei fiktive Ressourcen $?x$ und $?y$ vergleichen.

SELECT *

WHERE

```
{ <x> ?p ?ox .
  <y> ?p ?oy . }
```

Sei die Menge E wie folgt definiert: $E(s) = \{(p, o) \in KB \mid (s, p, o) \in KB\}$. Sie enthält also, abhängig vom Subjekt s , die geordnete Menge (p, o) aus Objekt und Prädikat. Diese sind Element der Wissensbasis, sowohl als Paar als auch

als Tripel. Folgende Mengenoperationen müssten ausgeführt werden, um die Gemeinsamkeiten und Unterschiede einzeln zu betrachten:

$$E(x) \cap E(y)$$

$$E(x) \setminus E(y)$$

$$E(y) \setminus E(x)$$

Dies ist, weil es eine zusammengesetzte Frage ist und der Komplexität einer solchen Operation im Allgemeinen, noch nicht möglich. Das Metamodell [SNIKS](#) stellt die Aufgabe der Ontologie gut dar: *Wer macht was und womit, nicht warum, wann oder wie.*

Das System soll nach [Definition 6](#) und [Definition 7](#) simple und komplexe Fragen beantworten können, oder zumindest mit ihnen trainiert werden. Es soll jedoch keine zusammengesetzten Fragen nach [Definition 8](#) beantworten können, da die einerseits oft in mehrere Unterfragen aufgespalten werden können und andererseits nur schwer beantwortbar sind. Letztlich ist auch anzumerken, dass nur Fragen aus Anforderungsbereich ([AFB](#)) 1 ([Anderson und Krathwohl, 2001](#)), und auch bei diesen nur ein Teil, beantwortet werden können. Unter diesen fallen Operatoren wie „angeben“ oder „aufzählen“.

5.1.1 Klassifizierung der Textbuchfragen

Nun sollen die Fragen aus [Winter, Haux et al. \(2011\)](#) einzeln nach Fragetyp und -art, welche oben erklärt wurden, klassifiziert werden. Sie wurden von [Roszeitis \(2022\)](#) bereits für das Quiz eingeordnet, in der Tabelle als „Original“ erkennbar. In [Tabelle B.1](#) in [Anhang B](#) werden sie erneut eingeordnet, Unterschiede sind **fett** gedruckt. Hier wird eine Frage als geeignet eingestuft, wenn sie faktisch und simpel oder komplex, nicht aber zusam-

mengesetzt sind. Außerdem wird den Fragen eine ID gegeben, welche aus Kapitelnummer und einer fortlaufenden Zahl gebildet wird.

Hier wurden insgesamt 47 der 79 Fragen als geeignet befunden, verglichen mit 30 vorher. Dies beruht größtenteils auf zwei Gründen:

1. Den Unterschieden zwischen dem Stellen von Fragen in Form eines Quizzes und der Fragenbeantwortung und
2. fälschlicher Einordnung einzelner Fragen mit dem Fragewort „how“ als prozedural.

Ersteres ist beispielsweise bei den Fragen 6.6/1, 8.6/1 oder 9/2 der Fall. Häufig wurden nach Definition 7 komplexe Fragen aussortiert, besonders bei Einschränkungen, wie etwa Frage 7/3. Die vermutlich fälschliche Einordnung liegt bei 6.5/1, 8/6 oder 10/2 vor. Jedoch können von den initial als geeignet befundenen Fragen nicht alle verwendet werden, da einige nicht in der Ontologie modelliert sind. Diese sind 4/3, 6/1, 6/3, 8/4, 8.4/1 und 9/2 sowie 9.3.4.2/3 bis 9.3.4.2/5, 9.4/1 und 10/3. Somit verbleiben letztendlich 36 Fragen, welche im Folgenden in SPARQL-Abfragen umgewandelt werden.

Aus den in Abschnitt 5.1.1 ausgewählten Fragen werden nun SPARQL-Abfragen gebildet und die Antworten durch Dr. Franziska Jahn korrigiert. Diese sind auch in Anhang B in Anhang B.2 vorhanden.

Nutzer, die die jeweiligen Fragen fragen, sollen, basierend auf den obigen Abfragen, die Antworten aufgeführt in Anhang B.3 erhalten. ²

² Momentan stehen hier noch einige falsche Antworten, diese sind **fett** gedruckt. In der finalen Version dieser Arbeit werden diese allerdings so gut wie möglich korrigiert sein.

5.1.2 Automatische Erstellung einfacher Fragen

Für die Erstellung der einfachen Fragen muss die großen Datenmenge aus Tripeln [SNIKS](#) genutzt werden. Fragen können sowohl nach dem Subjekt als auch nach dem Objekt oder Prädikat des Tripels gestellt werden. Für die ersteren beiden Intentionen ist das Fragewort über das Prädikat herausfindbar und ist immer „Who“ oder „What“. Für die Frage nach dem Prädikat ist die Zeichenkette immer „How are ?sl and ?ol related?“, wobei ?sl für das Label des Subjekts und ?ol für das Label des Objekts steht. Danach folgt das Label des Prädikats des Tripels und das Label der anderen gegebenen Ressource.

Die Erstellung der Abfragen ist aufgrund der vielen Funktionen, die [SPARQL](#) insbesondere auch für die Verarbeitung von Zeichenketten bereitstellt, pro Typ über eine einzige [SPARQL](#)-Abfrage möglich, welche, vom Betreuer Dr. Konrad Höffner bereitgestellt, auf der des [SNIK-Quizzes](#) basiert.

SPARQL-Abfrage für Fragen nach dem Subjekt

```
SELECT DISTINCT REPLACE(REPLACE(REPLACE(REPLACE(
    CONCAT("What ",?pl, " ", ?ol, "?"),
    "What is responsible", "Who is responsible"),
    "What approves", "Who approves"),
    "What is involved", "Who is involved"),
    "What .* component", "What has the component") as ?question,
CONCAT("SELECT DISTINCT ?s WHERE { ?s <", STR(?p), "> <", STR(?o), ">."
    ↪ }) as ?sparql
FROM sniko:meta
FROM sniko:bb
{
    ?s ?p ?o.
```

```

?p rdfs:domain [rdfs:subClassOf meta:Top].
?p rdfs:range [rdfs:subClassOf meta:Top].
?s a [rdfs:subClassOf meta:Top].
?o a [rdfs:subClassOf meta:Top].
?p rdfs:label ?pl. FILTER(langmatches(lang(?pl), "en")).
?o rdfs:label ?ol. FILTER(langmatches(lang(?ol), "en")).
}
ORDER BY RAND()

```

Zuerst wird die Frage mithilfe der String-Funktionen generiert. Anfangs wird eine Zeichenkette aus dem Fragewort „What“, welches das von Prädikaten am häufigsten verwendete ist, dem Label des Prädikats und dem Label des Objekts gebildet, um eine bearbeitbare Grundlage zu haben, welche oft schon so verwendet werden kann. Es werden die einzelnen Prädikate, welche das Fragewort „Who“ haben oder für die Nutzung als Frage leicht verändert werden müssen, durchgegangen und gegebenenfalls mithilfe der Funktion REPLACE ersetzt.

Über die Auswahl von den Teilontologien meta und bb über das Schlüsselwort FROM wird gewährleistet, dass nur für diese Fragen generiert werden. Es wird außerdem garantiert, dass nur Prädikate, die Beziehungen zwischen Ressourcen darstellen, genutzt werden, wie in Abb. 3.1 zu sehen ist.

In der eigentlichen Abfrage wird festgelegt, dass das Prädikat den Definitionsbereich und Wertebereich `rdfs:subClassOf meta:top` haben muss, also nur Beziehungen zwischen Ressourcen der `SNIK`-Ontologie darstellen soll. Außerdem wird als erstes das Tripel `?s ?p ?o` hingeschrieben, um die Beziehung dieser drei Variablen zu sichern. Dann werden das Objekt sowie die Subjekte und deren englischsprachige Labels ausgewählt. Da eine Ressource mit dem gleichen Prädikat als Objekt zu mehreren anderen Ressourcen in Ver-

bindung stehen kann, wird eine Zeichenkette erstellt, die die Labels der Subjekte mit Semikolons getrennt ausgibt. Die URIs werden auch ausgegeben und es wird alles zufällig sortiert.

Die Frage zur Generierung der Fragen nach dem Objekt sieht der nach dem Subjekt ziemlich ähnlich, hier wird immer das Fragewort „What“ verwendet, etwa bei „What is the Chief Information Officer responsible for?“. Allerdings sieht man an dieser Abfrage auch die Schwierigkeit dieser Art von Fragen: Es gibt Prädikate, welche aus zwei oder mehr Wörtern bestehen, bei denen das Subjekt zwischen diesen stehen muss, wie hier zum Beispiel der Chief Information Officer zwischen „is“ und „responsible for“. Dies muss jedoch nicht pro Prädikat einzeln gemacht werden, hierzu kann ein regulärer Ausdruck genutzt werden. Außerdem müssen manche Prädikate, wie zum Beispiel „approves“, konjugiert werden. Dies lässt sich auch mithilfe eines regulären Ausdrucks für alle Prädikatlabel mit nur einem REPLACE-Statement erreichen. Die Abfrage lautet wie folgt:

```
# SPARQL-Abfrage für Fragen nach dem Objekt
SELECT DISTINCT CONCAT(
  "What ", REPLACE(REPLACE(REPLACE(
    STR(?pl), ".* component", CONCAT("are components of ", STR(?sl))),
    "^is ([a-z]* [a-z]*)", CONCAT("is ", STR(?sl), " $1")),
    "^[a-z]*e)s", CONCAT("is $1d by ", STR(?sl))),
  "?") as ?question,
CONCAT ("SELECT DISTINCT ?o WHERE { <", STR(?s), "> <", STR(?p), ">
  ⇨ ?o. }") as ?sparql
FROM sniko:meta
FROM sniko:bb
{
  ?s ?p ?o.
```

```

?p rdfs:domain [rdfs:subClassOf meta:Top].
?p rdfs:range [rdfs:subClassOf meta:Top].
?s a [rdfs:subClassOf meta:Top].
?o a [rdfs:subClassOf meta:Top].
?s rdfs:label ?sl. FILTER(langmatches(lang(?sl), "en")).
?p rdfs:label ?pl. FILTER(langmatches(lang(?pl), "en")).
}
ORDER BY RAND()

```

Dadurch konnten 621 Fragen nach dem Subjekt und 374 Fragen nach dem Objekt generiert werden. Dies entspricht nicht ganz der Anzahl von Tripeln der Teilontologie *bb*, was zuerst erwartet war. Die Anzahl wird allerdings einerseits dadurch verringert, dass manche Ressourcen kein Label haben und dies in der Abfrage gefordert wird. und dass bei der Frage nach den Subjekten viele Fragen durch das Schlüsselwort `DISTINCT` gekürzt wurden. Die Fragen nach einem Objekt mit dem gleichen Prädikat von verschiedenen Subjekten waren natürlich immer gleich, äquivalent zu den Fragen nach dem Subjekt mit verschiedenen Objekten und gleichen Prädikaten.

5.2 AUSWAHL EINES KANDIDATEN

Die Auswahl eines oder mehrerer Kandidaten gestaltete sich aufgrund von nicht instand gehaltener Programme und der Spezialisierung von vielen Systemen auf DBpedia oder ähnliche Wissensbasen sehr schwierig. Bei `gAnswer2` war es beispielsweise nicht möglich, überhaupt ein Programm, was man hätte ausprobieren können, zu finden, bei `gAnswer` war die Vorbereitung der Daten sehr aufwändig und die gegebenen Werkzeuge nicht mehr funktional. `DeepPavlov` war am Anfang sehr vielversprechend, beson-

ders aufgrund des modularen Aufbaus. Jedoch gibt es keine Möglichkeit, andere Daten als Wikidata zu verwenden, weshalb es für den Einsatz mit [SNIK](#) nicht nutzbar ist. Ähnlich sieht es mit AskNow aus, wofür aufgrund des Entity Linkings DBpedia verwendet werden muss. In der Dokumentation³ wird allerdings geschrieben, dass in der Zukunft auch die Verwendung anderer Quellen möglich sein sollen. All diese und andere Probleme beschrieb bereits Diefenbach (2018b). Er nannte speziell die Probleme der

- Mehrsprachigkeit,
- Portabilität,
- Skalierbarkeit,
- Robustheit,
- Fähigkeit, über mehrere Dateien zu suchen und
- Präsentation der Antwort.

Hier sind vor allem die Portabilität und Robustheit, aber auch teilweise die Präsentation der Antwort problematisch.

5.2.1 TeBaQA

TeBaQA war anfangs sehr vielversprechend, mehr noch als DeepPavlov. Es ermöglicht die Verwendung von eigenen [RDF](#)-Tripeln ermöglichen, indem die `indexing.properties`-Datei verändert wird. Es gibt zwei Ordner, einen namens `ontology` und einen namens `data`. Da [SNIK](#), wie in Abschnitt 3.1

³ <http://docs.deeppavlov.ai/en/master/features/models/kbqa.html>, abgerufen am 9. Mai 2022

schon erwähnt, eine Ontologien aus Ontologien ist, wird in den ontology-Ordner nur die meta.ttl-Datei gelegt, welche die Teilontologie enthält, in welcher die Properties gespeichert sind. Die restlichen Tripel kommen alle in den data-Ordner, da es bei TeBaQA auch möglich ist, mehrere Dateien als Quelle zu verwenden. Die auf .flag endenden Variablen wurden auf true gesetzt. Andere Dateien wurden nicht verändert.

Leider warf das System immer wieder Fehler bezüglich der Elasticsearch-Konfiguration, weshalb wir ein Docker-Image, also eine auf Anwendungsebene virtuelle Umgebung, mit Elasticsearch 6.6.1, TeBaQA und der SNIK-Ontologie erstellt haben. Die Ontologie ist zusätzlich noch ein weiteres Mal in dem Docker-Image, um Probleme mit der Reihenfolge des Ladens der Daten zu beheben. Letztendlich können wir das System aber nicht nutzen, da immer wieder zu Laufzeitfehlern auftreten, sodass weder SPARQL-Abfragen noch Ergebnisse angezeigt werden.

5.2.2 *QAnswer KG*

QAnswer KG präsentierte sich als einfache Methode, mithilfe von eigenen Daten Question Answering zu betreiben. Es funktioniert, indem man auf der Website⁴ einen Account erstellt und in diesem seine RDF-Daten in Form von n-Tripeln hochlädt, einem Serialisierungsformat wie Turtle für RDF.

5.2.2.1 *Anfängliche Konfiguration und Probleme*

Beim Einrichten der Umgebung wurde die Sprache auf Englisch gestellt, da in dieser auch die Fragen beantwortet werden sollen.

⁴ <https://app.qanswer.ai/>

QAnswer KG versucht, sich auf eine Antwort zu konzentrieren und stellt nur die als am besten bewertete SPARQL-Abfrage dar. Hier wird das Problem der Ambiguität deutlich, da bei der Frage „What is the chief information officer responsible for?“ sowohl `meta:isResponsibleForEntityType`, `meta:isResponsibleForFunction` und `meta:isResponsibleForRole` gemeint sein könnten. Der Ansatz für die Lösung dieses Problems ist es, das System alle drei ausführen zu lassen, oder ihm das zumindest zu ermöglichen. Dies kann zum Beispiel über Property Paths realisiert werden. In diesem Fall erlauben Property Paths dem Prädikat, in der Abfrage verschiedene Ressourcen darstellen zu können. Es werden letztendlich drei Abfragen ausgeführt, eine für jede mögliche Kombination der Attribute, also hier einmal pro Prädikat. Statt allein `meta:isResponsibleForEntityType` steht durch die Nutzung von Property Paths nun `meta:isResponsibleForRole | meta:isResponsibleForFunction | meta:isResponsibleForEntityType` dort. Die ganze Abfrage sieht unter Verwendung von Property Paths wie folgt aus:

```
SELECT ?o
WHERE {
  bb:ChiefInformationOfficer (meta:isResponsibleForRole |
    ↔ meta:isResponsibleForFunction |
    ↔ meta:isResponsibleForEntityType) ?o .
}
```

Jedoch kann QAnswer keine Property Paths austellen, weshalb diese Option zur Lösung des Ambiguitätsproblems wegfällt.

Eine andere Möglichkeit, die obige Frage mittels einer SPARQL-Abfrage zu beantworten, ist mittels der `rdfs:subPropertyOf`-Beziehung:

```
SELECT ?o
```

```
WHERE {
  bb:ChiefInformationOfficer ?p ?o .
  ?p rdfs:subPropertyOf* meta:isResponsibleFor .
}
```

Diese regelt die Hierarchie von Properties. So sind die drei, welche hier alle in einer Abfrage zusammengefasst werden sollen, alle ein Subproperty von `meta:subPropertyOf`.

Dabei wichtig ist, dass hier * statt + oder gar nichts nach `rdfs:subPropertyOf` steht, falls `meta:isResponsibleFor` selbst genutzt wird, und nicht nur die untergeordneten Properties. Diese Zeichen geben wieder einen Property Path an, der von QAnswer wieder nicht gelöst werden kann.

Deshalb mussten vor dem Training die `rdfs:subPropertyOf+`-Beziehung materialisiert werden⁵, das heißt es wurden alle transitiven Subpropertybeziehungen zu Trainingszwecken mittels dem SPARQL-Befehl CONSTRUCT zu direkten Subklassenbeziehungen umgeformt.

Das ist mit dieser SPARQL-Abfrage möglich:

```
CONSTRUCT {?s ?p ?o}
FROM sniko:meta
FROM sniko:bb
WHERE {
  ?s ?pp ?o .
  ?pp rdfs:subPropertyOf+ ?p .
}
```

Es werden erst alle Tripel gefunden, die ausschließlich aus den Teilontologien meta und bb bestehen, da dies den Umfang dieser Arbeit beschreibt.

⁵ Es werden nur `rdfs:subPropertyOf+`-Beziehungen materialisiert, da das Tripel mit der übergeordneten Property schon in der Wissensbasis (KB) ist.

Somit werden auch nur Prädikate aus `meta` betrachtet, nicht aber `rdfs` oder anderen. Danach werden alle Subproperties des Prädikats aus dem ersten Tripel ausgewählt. Mittels `CONSTRUCT` werden die ausgewählten Prädikate anstelle dem anderen Property eingesetzt. Diese Tripel werden dann manuell den `RDF`-Daten hinzugefügt⁶

Zusätzlich werden auch die `rdfs:subClassOf+`-Beziehungen materialisiert. Dies muss geschehen, da die Antworten für manche Textbuchfragen alle transitiven Subklassen sind.

Das kann mittels der folgenden Abfrage bewerkstelligt werden:

```
CONSTRUCT {?s ?p ?oo}
FROM sniko:meta
FROM sniko:bb
WHERE {
  ?s ?p ?o .
  FILTER(?oo!=owl:Thing && ?oo!=meta:Top) .
  FILTER(STRSTARTS(STR(?oo), "http://www.snik.eu/ontology/bb")) .
  ?o rdfs:subClassOf+ ?oo .
}
```

Beide dieser `SPARQL`-Abfragen wurden vom Betreuer bereitgestellt. Durch die Materialisierung kann beispielsweise für die Frage 8/6, „How can quality of HIS be evaluated?“, die Klasse `bb:CaseStudy` gefunden werden (siehe Anhang B.3), obwohl diese eine direkte Subklasse von `bb:QualitativeEvaluationMethod` ist, welche wiederum eine direkte Subklasse von `bb:EvaluationMethod` ist, welche in der in Anhang B.2 sichtbaren Abfrage gesucht wird. Für die

⁶ Verwendete n-Tripel-Datei verfügbar unter:

<https://github.com/Yagnap/BeLL-Question-Answering-auf-Linked-Data-SNIK/blob/main/Data/qanswerset.nt>

KB mit den unmaterialisierten transitiven Subklassenbeziehungen sieht die Abfrage so aus:

```
SELECT DISTINCT ?s
WHERE
  { ?s rdfs:subClassOf+ bb:EvaluationMethod . }
```

Nach der Materialisierung kann das + entfernt werden, wodurch folgende Abfrage ausreicht:

```
SELECT DISTINCT ?s
WHERE
  { ?s rdfs:subClassOf bb:EvaluationMethod . }
```

Die erste Abfrage kann nicht durch QAnswer generiert werden, die zweite jedoch schon.

Wenn keine Lösung für die Frage gefunden wurde, gab es als Ausgabe meist die Ressource selbst, also zum Beispiel `bb:ChiefInformationOfficer`. Dies kann zwar nützlich, aber auch verwirrend sein und sollte mithilfe von Training verhindert werden. Die Präzision der Ergebnisse ohne jegliches fine-tuning ist aber trotzdem erstaunlich.

Praktisch für die Lokalisierung der Fehler war die Funktion, sich alle generierten Anfragen anzeigen zu lassen. Somit konnte erahnt werden, warum es das macht, was es macht.

Mit der Konfiguration an sich konnten schon viele Fehler behoben werden. So gibt es eine Liste von „stop words“, welche nicht betrachtet werden. Unter diesen sind häufig verwendete Präpositionen, Konjunktionen, Verben oder Füllwörter wie „and“ oder „many“. Diese sollen verhindern, dass falsche Ressourcen gefunden werden. Hier musste diese Liste allerdings so modifiziert werden, dass das Wort `for` nicht mehr darin vorkommt, denn ohne es können Prädikate wie „responsible for“ schwer erkannt werden,

besonders, da die beiden Wörter oft getrennt im Satz vorkommen. Entfernt wurden auch `define` und `describe`. Somit verblieben dann folgende Wörter in der Liste der Stopp-Wörter: `a`, `about`, `all`, `an`, `and`, `are`, `as`, `at`, `be`, `by`, `can`, `did`, `do`, `does`, `from`, `give`, `goes`, `had`, `has`, `have`, `here`, `how`, `in`, `into`, `is`, `its`, `list`, `many`, `most`, `my`, `no`, `of`, `on`, `or`, `s`, `show`, `some`, `something`, `such`, `tell`, `the`, `their`, `these`, `they`, `this`, `to`, `using`, `was`, `were`, `what`, `which`, `will`, `with`, `yes`.

Aus den *Hidden Properties* wurde `rdfs:subClassOf` hinzugefügt. Hidden Properties sind solche, die nicht explizit in der Frage benutzt werden, aber trotzdem gemeint werden können. Es wurde `skos:altLabel` als weiteres Label für Ressourcen hinzugefügt, sodass QAnswer auch diese beachtet. Die Mappings mussten auch dahingehend verändert werden, als dass `skos:definition` und `rdfs:comment` als Beschreibung hinzugefügt wurden. Die Definition wurde auch bei den Ergebnissen angezeigt, sodass es bei Ergebnissen von Fragen nach der Definition als richtig erachtet wurde, wenn die zu definierende Ressource richtig erkannt wurde.

Es gibt auch die Möglichkeit, direkt Wörter als Aliase für URIs zu nutzen. Bei den *Property Mapping* können URIs und dafür stehende Lexikalisierungen in Abhängigkeit gebracht werden, wie bei einem Wörterbuch. Dies wurde hier für „phases“ und „methods“ bei `meta:updates` sowie „tasks“ und `meta:functionComponent` gemacht. Für `meta:entityTypeComponent` wurde „facets“ hinzugefügt. Notwendig war es aufgrund der Funktionsweise QAnswers; ohne Lexikalisierungen wüsste das Programm nicht, wofür die Ressourcen stehen. Da die Properties in `SNIK` sehr allgemein gehalten sind, fehlen oft Labels, mithilfe deren QAnswer auf die Ressource als Teil der Antwort schließen kann. Für diese stehen die Lexikalisierungen zur Verfügung.

5.2.2.2 *Training*

Das Frage-Antwort-Set aus dem Lehrbuch wurde in zwei Hälften geteilt, eine zum Training und eine zum Testen. Die Fragen wurden randomisiert in die Gruppen eingeteilt, sodass 18 im Trainingsdatensatz und 18 im Testdatensatz sind.

Von den automatisch generierten Paaren wurden 100 für das Testen und 895 für das Training genutzt.

Das Training verläuft so ab, dass dem System automatisiert pro Schritt zehn weitere automatisch generierte Paare aus dem Trainingsdatensatz gegeben werden, wodurch jede Runde mit zehn zusätzlichen Fragen trainiert wird. Der Trainingsatz der Lehrbuchfragen wird in der ersten Runde nicht mit verwendet, in der zweiten Runde werden von Anfang an alle Textbuchfragen verwendet.

Dann wird QAnswer mithilfe der [API](#)-Methode trainiert. Auf das trainierte System wird der Testdatensatz zur Evaluierung angewandt. Über eine [API](#)-Abfrage wird die Antwort mit dem höchsten Confidence-Wert, eine [SPARQL](#)-Abfrage, geholt und auf dem [SPARQL](#)-Endpunkt von [SNIK](#) ausgeführt. Die Antworten der richtigen Lösung wurden schon vorher gespeichert und werden nun mit denen der Abfrage von QAnswer verglichen.

Dann wird das trainierte Modell zurückgesetzt, die gegebenen Frage-Antwort-Paare bleiben jedoch erhalten. In [Abb. 5.1](#) sieht man, wie groß das F-Maß abhängig von der Fragenanzahl ist. Es sind zwei Datensätze dargestellt. Der eine wurde andere ausschließlich über die [SPARQL](#)-generierten Frage-Antwort-Paare trainiert, beim anderen wurden die Lehrbuchfragen beim Training inkludiert. Am Anfang, als noch gar keine oder nur sehr wenige Fragen zum Training verwendet wurden, ist das F-Maß eher gering. Besonders beim Training mit den Lehrbuchfragen liegt es bei dem

Modell ohne Training deutlich unter 30%. Ab 30 Fragen nähert es sich den Werten des Trainings ohne die Lehrbuchfragen an und übersteigt diese bei 40 Fragen erstmals. Bei 70 Fragen erreicht das F-Maß des Trainings mit den Lehrbuchfragen bei 90% ein Maximum und bleibt sehr lange nahezu konstant.

Bei 520 bzw. 570 Fragen sinken die Werte für beide Graphen aber sehr abrupt sehr stark ab, Varianz wird viel größer, mit Werten zwischen fast 10% und mehr als 90%.

Auch die Graphen von Präzision (Abb. 5.2) und Recall (Abb. 5.2) bieten ein ähnliches Bild. Sie haben auch am Anfang kurz geringere Werte, eine starke, konstante Mitte und danach eine extreme Varianz.

Der Confidence-Wert sieht im Allgemeinen deutlich konstanter aus, an ihm spiegeln sich aber auch einige der beim F-Maß betrachteten Phänomene. Anfangs ist der Wert durchschnittlich etwas geringer als später, aber immer noch fast überall über 50%. Ab etwa 100 Fragen erreicht es ein Niveau zwischen 80% und 90%, auf dem es bleibt. Ab 510 bzw. 560 Fragen kommt es jedoch zu größeren Ausreißern, die teilweise bis auf 11% heruntergehen.

Die Graphen der Lehrbuchfragen sehen denen der automatisch generierten auf den ersten Blick nicht sehr ähnlich aus, es lassen sich aber doch einige Gemeinsamkeiten erkennen. Sie sind über die Mitte auch recht konstant und haben mit mehr Trainingsfragen auch eine sehr viel größere Varianz. Anfangs sind die Ergebnisse auch hier schlechter.

Die durchschnittliche Varianz ist hier aber größer als bei den generierten Fragen, und außerdem sind die Werte für F-Maß und co. deutlich niedriger. Bei den automatisch generierten Fragen waren die Werte konstant bei und über 90%, hier nur knapp über 25%. Außerdem gibt die Verwendung

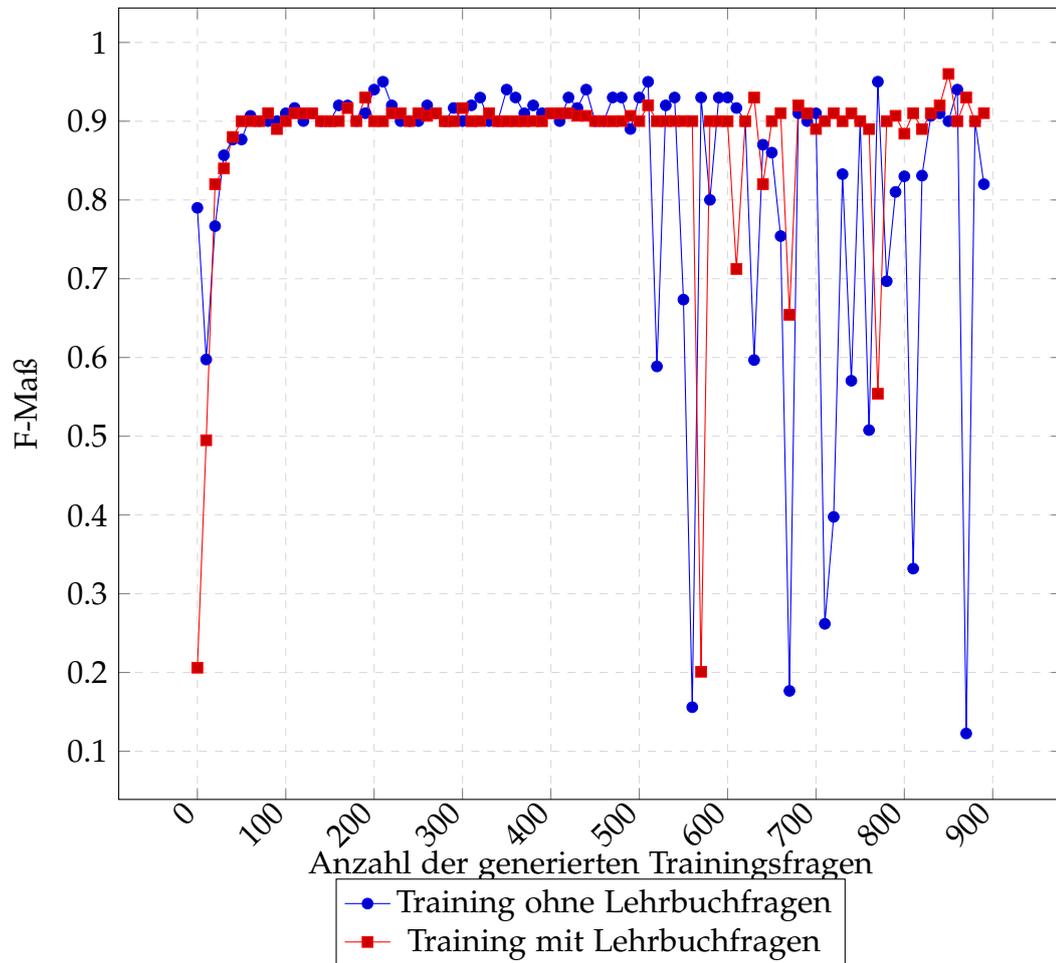


Abbildung 5.1: Durchschnittliches F-Maß bei den automatisch generierten Testfragen in Abhängigkeit zur Anzahl der generierten Trainingsfragen.

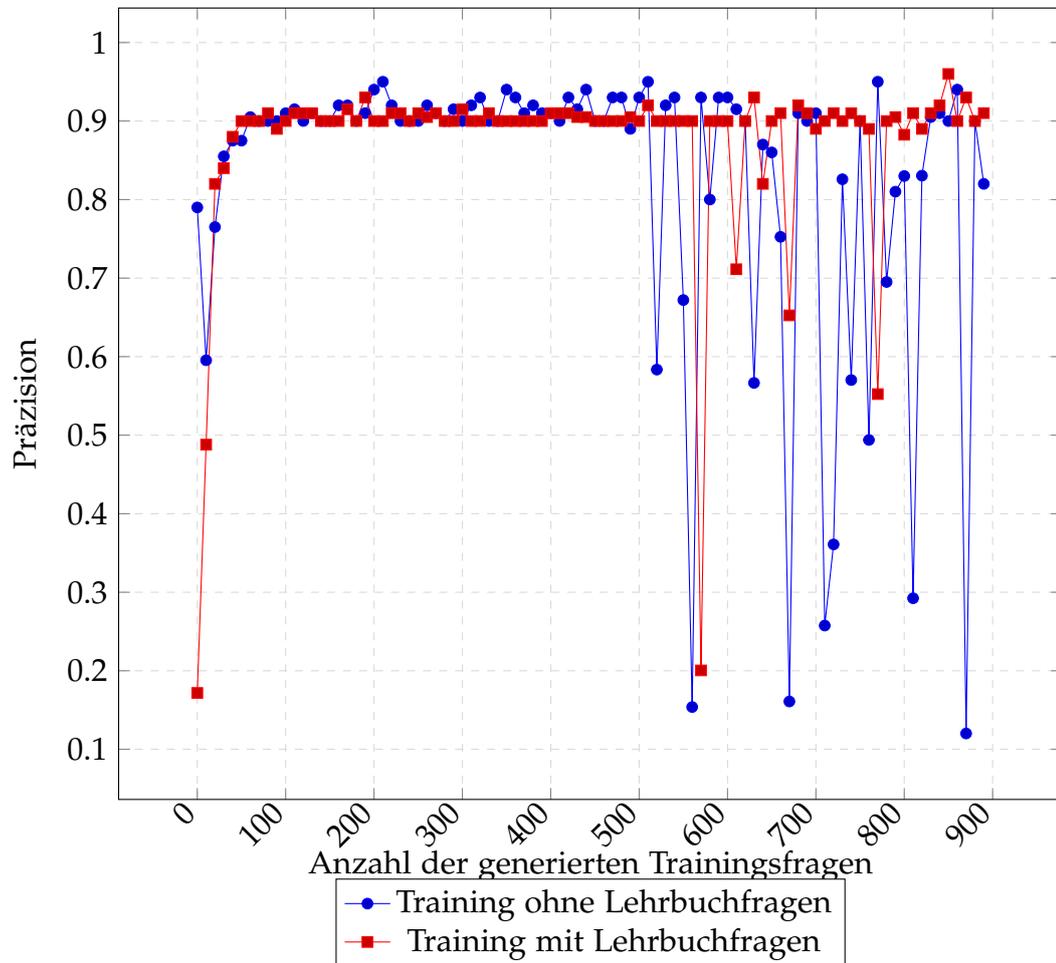


Abbildung 5.2: Durchschnittliche Präzision bei den automatisch generierten Testfragen in Abhängigkeit zur Anzahl der generierten Trainingsfragen.

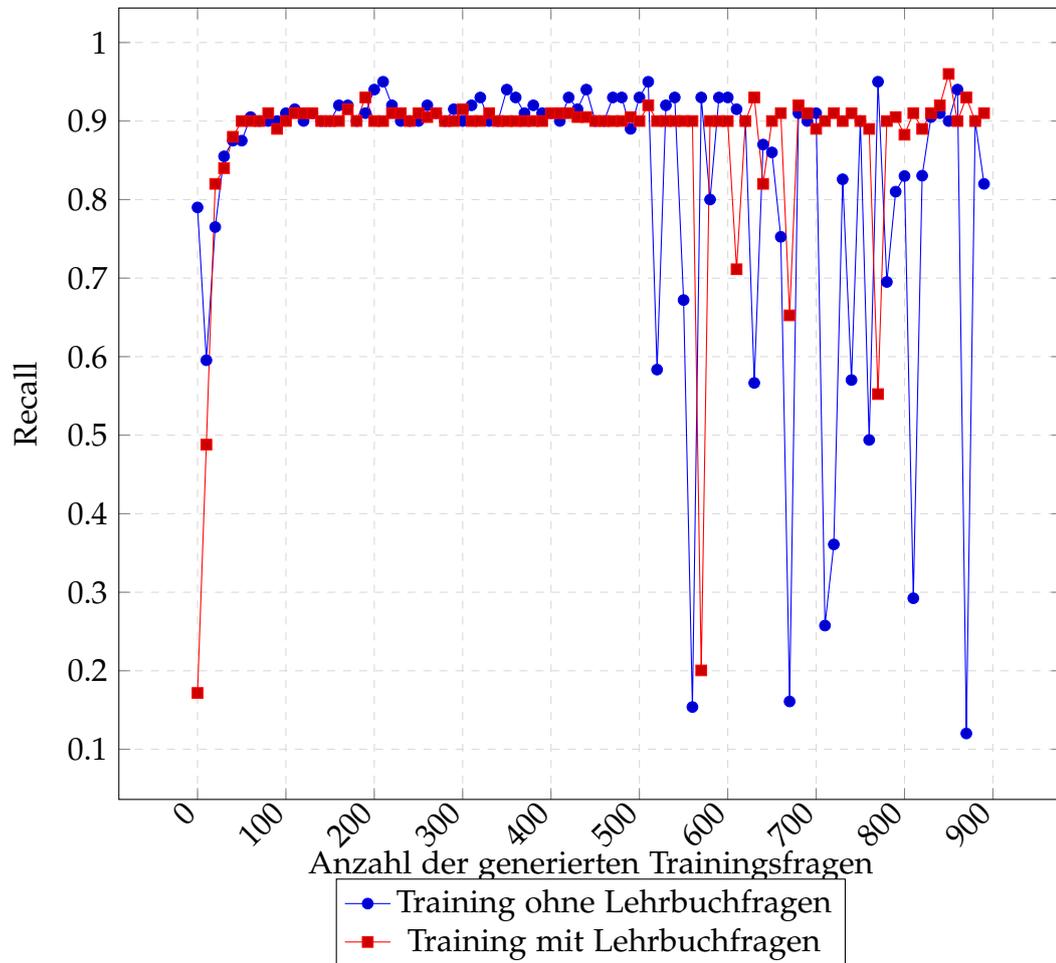


Abbildung 5.3: Durchschnittlicher Recall bei den automatisch generierten Testfragen in Abhängigkeit zur Anzahl der generierten Trainingsfragen.

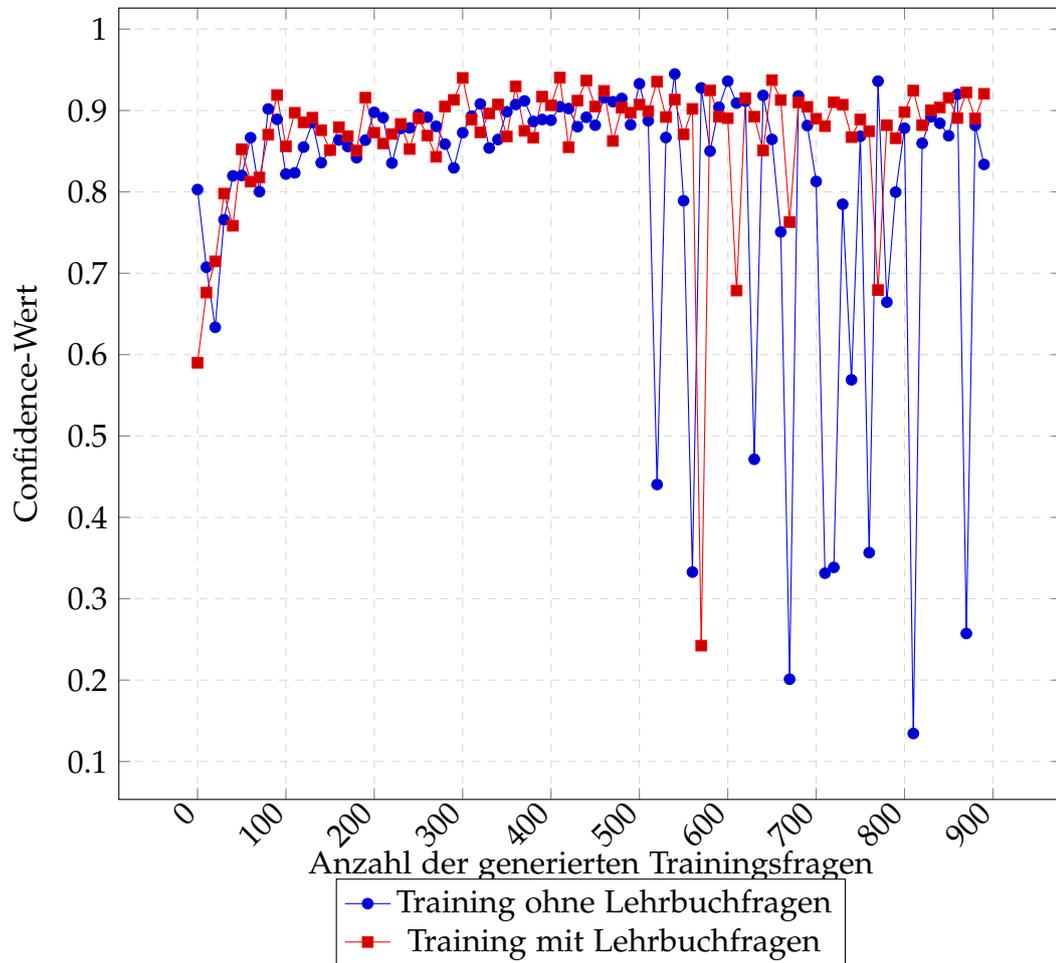


Abbildung 5.4: Durchschnittlicher Confidence-Wert QAnswers bei den automatisch generierten Testfragen in Abhängigkeit zur Anzahl der generierten Trainingsfragen.

des Lehrbuchfragen-Trainingssets deutlich schlechtere Ergebnisse, bei den generierten Fragen war dieser Unterschied auch deutlich geringer.

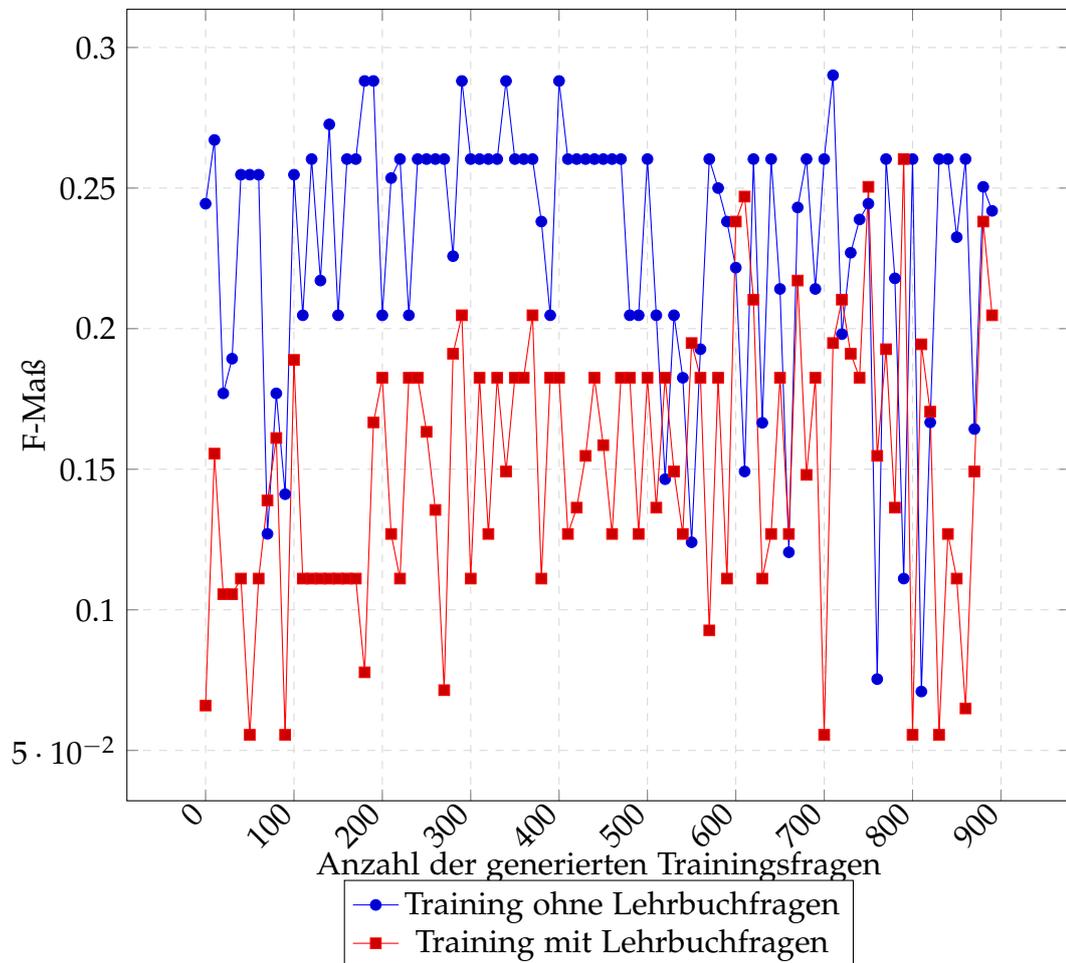


Abbildung 5.5: Durchschnittliches F-Maß bei den Lehrbuchfragen im Testset in Abhängigkeit zur Anzahl der generierten Trainingsfragen.

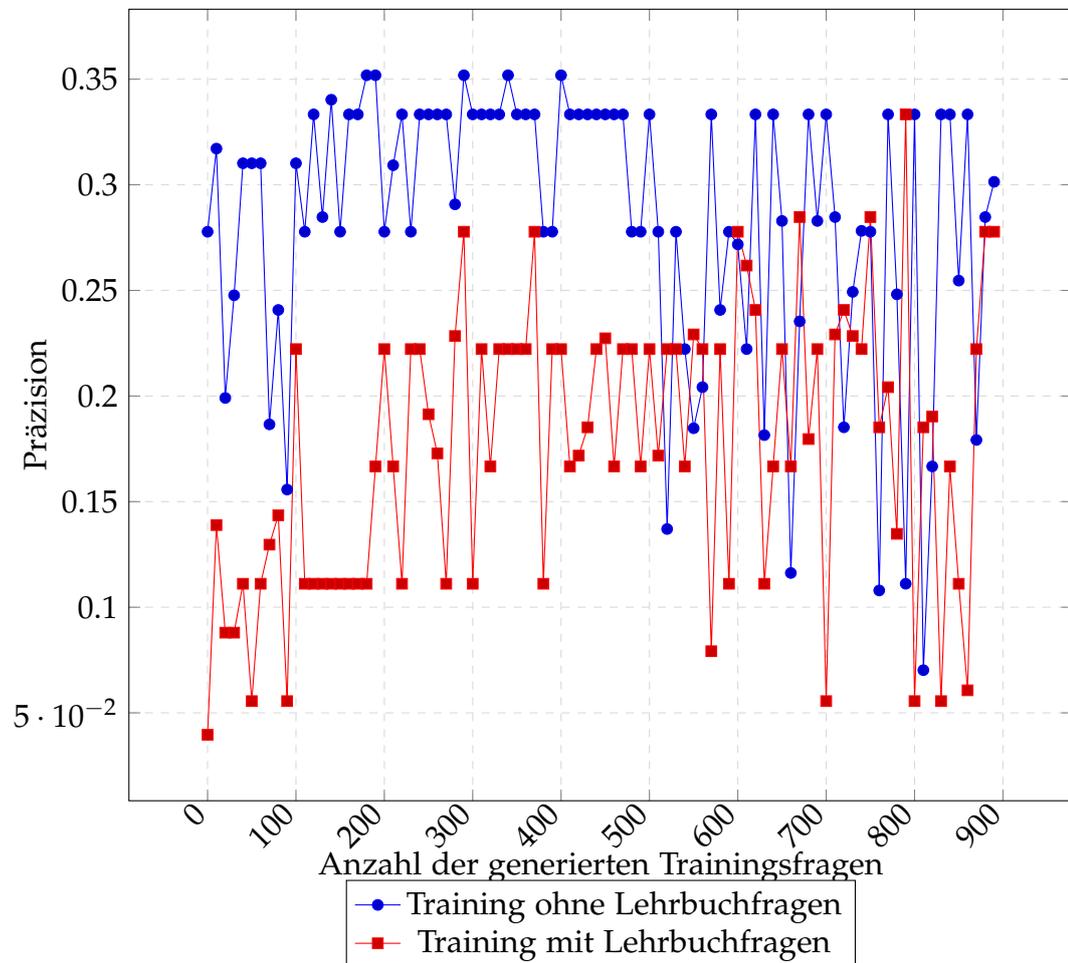


Abbildung 5.6: Durchschnittliche Präzision bei den Lehrbuchfragen im Testset in Abhängigkeit zur Anzahl der generierten Trainingsfragen.

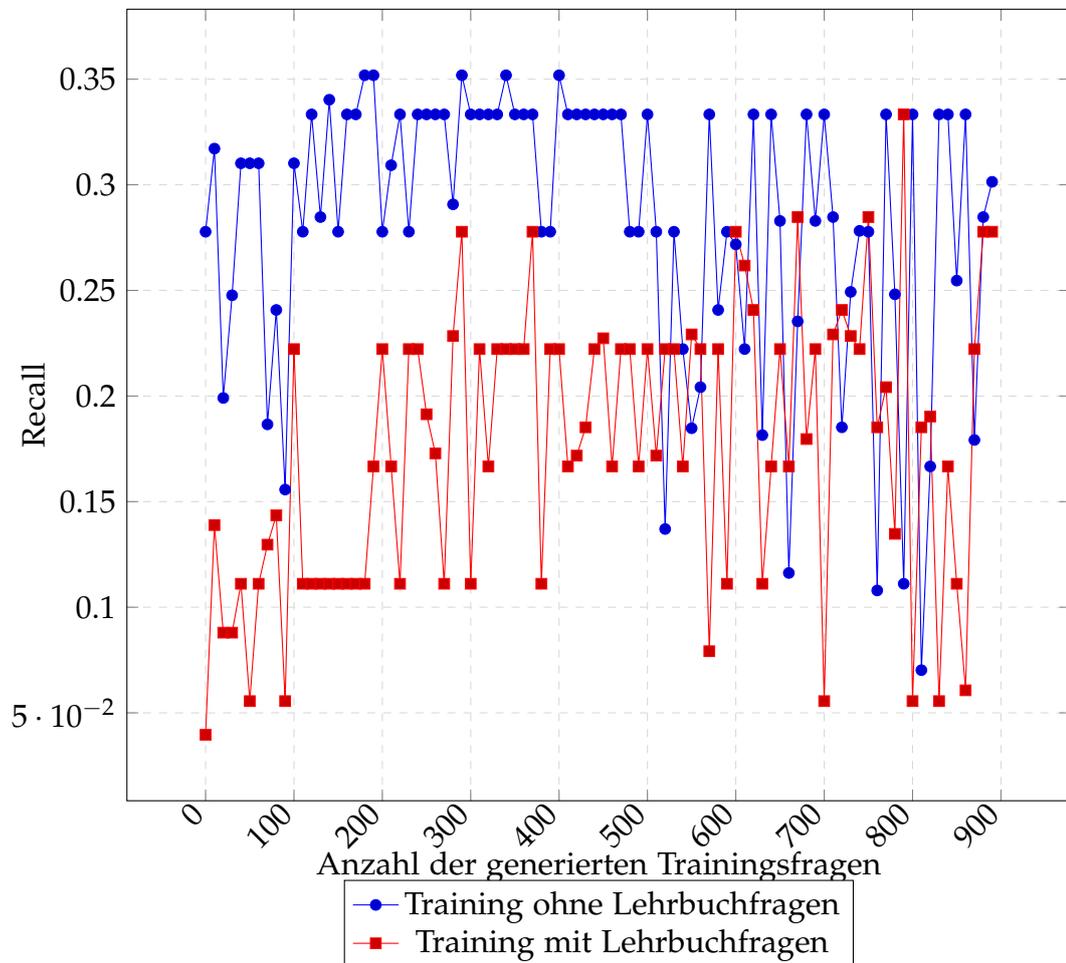


Abbildung 5.7: Durchschnittlicher Recall bei den Lehrbuchfragen im Testset in Abhängigkeit zur Anzahl der generierten Trainingsfragen.

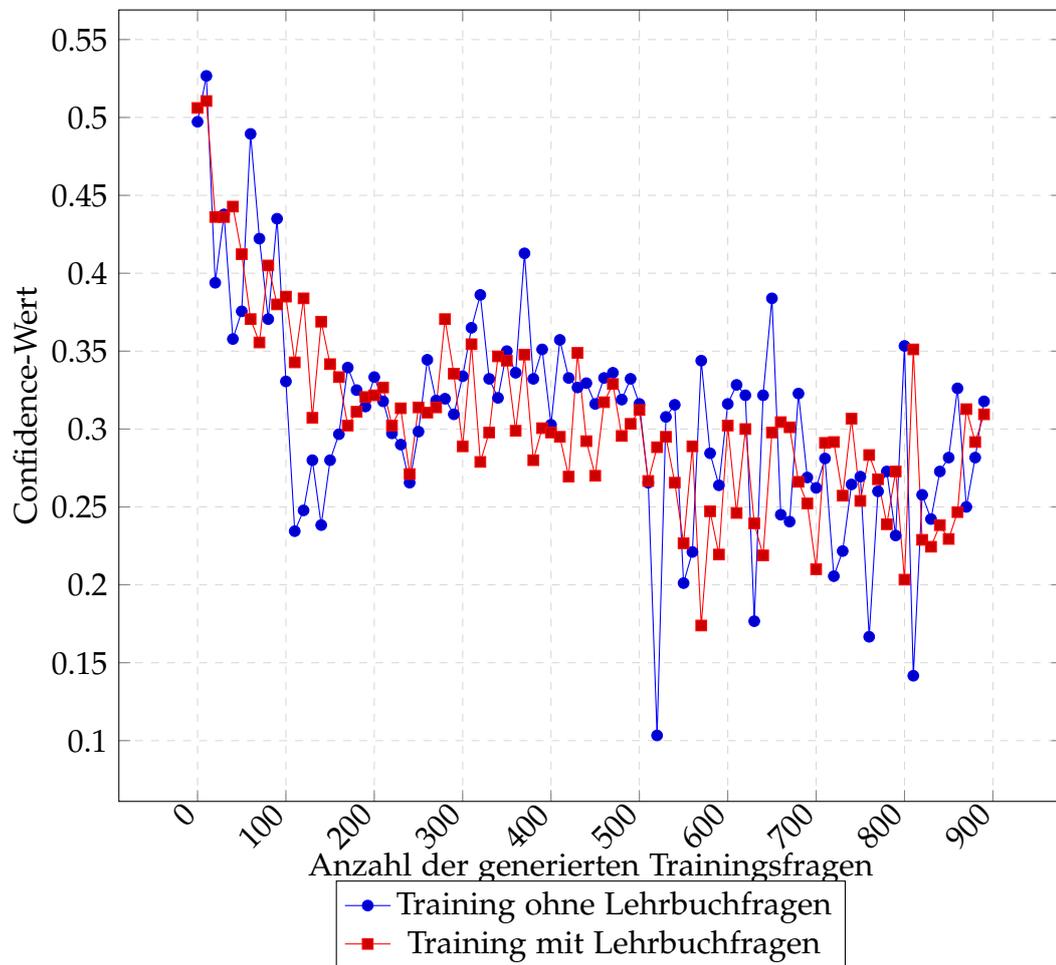


Abbildung 5.8: Durchschnittlicher Confidence-Wert QAnswers bei den Lehrbuchfragen im Testset in Abhängigkeit zur Anzahl der generierten Trainingsfragen.

ERGEBNISSE

Es wurde ein Benchmark aus 36 Lehrbuchfragen, von denen jeweils 18 zufällig der Test- und Trainingsgruppe zugeordnet wurden, erstellt und auf QAnswer verwendet. Ein weiterer Benchmark aus 995 automatisch generierten, sehr einfachen Fragen wurde automatisch erstellt, 100 von den Fragen wurden dem Testset, 895 dem Trainingsset zugeordnet. Die Teilontologie aus bb wurde mit einem QA-System, QAnswer KB, genutzt und mithilfe des Benchmarks aus Lehrbuchfragen trainiert und evaluiert. Die Ergebnisse sind als bei sehr einfachen Fragen sehr gut, Grenzen werden aber schnell erreicht.

Das System kann unter https://app.qanswer.ai/public-share?kb=SNIK_BB&type=graph&user=kirdie&lang=en aufgerufen werden.

DISKUSSION UND AUSBLICK

7.1 BENCHMARK

Der Benchmark wurde erstellt, indem Fragen aus Winter, Haux et al. (2011) genommen wurden und nach ihrer theoretischen Beantwortbarkeit klassifiziert wurden. Die Eignung dieser Fragen für ein System, das anhand der gegebenen Frage und den in ihr enthaltenen Wörtern herausfinden soll, die auf die Frage zutreffende Antwort finden soll, ist fraglich. Oft gibt es Fragen wie 9.3.4.2/1, „Which organizational units are involved in information management?“, bei der die Ressourcen nicht mit ihrem Namen genannt werden. Hier wird das Prädikat `meta:functionComponent` verwendet, um die richtige Antwort zu erhalten, obwohl `meta:isInvolvedIn` deutlich naheliegender ist. Dies lässt auf eine eventuelle Untauglichkeit der Fragen für die Ontologie schließen, da die Ontologie meist nur sehr generelle Properties hat, um die Beziehungen darzustellen.

Es ist aber trotzdem gelungen, die gegebenen Fragen, welche teilweise beantwortet werden können, einen Benchmark zu erstellen. Dieser besteht aus Fragen, welche auf in Winter, Haux et al. (2011) vorhandenes Wissen zurückgreifen und dieses festigen soll. Dies ist eigentlich perfekt für die Nutzung mit `SNIK`, nur die Formulierung ist manchmal nicht optimal. Es lässt sich allerdings auch sagen, dass die Fragen für Question Answering von simplen Fragen vergleichsweise definitiv höchst komplex waren. Die

einfachen Fragen waren recht schnell generierbar und haben die große Datenmenge [SNIKS](#) gut genutzt, auch wenn durch die Anzahl der Tripel in der Teilontologie [bb](#), 2056, erst mehr Fragen erhofft waren. Dies war allerdings aufgrund von fehlenden Labels oder Subjekten, welche mit vielen Objekten über das gleiche Prädikat verbunden waren und somit die gleiche Frage erzeugten, was über das Schlüsselwort `DISTINCT` gekürzt wurde.

Es könnte darüber nachgedacht werden, die automatisch generierten Fragen komplexer zu machen, dies ist aber nicht unbedingt erforderlich. Es konnten viele Fragen automatisch generiert werden, was viel mehr Trainingsfragen bedeutet hat, was auch dem fine-tuning hilft. Gerade die Fragen mit verschiedenen Intentionen sind hilfreich. Allerdings würde damit die Funktion der einfachen Fragen als Basiswerte negiert die Rolle der Lehrbuchfragen als komplexere, der Realität näher liegende Fragen behindert.

In Zukunft sollten noch mehr schwierigere Fragen erstellt werden. Denkbar wäre vor allem, Studierende der Medizininformatik nach Fragen, die sie fragen würden, zu fragen. Für die Realisierung wäre vermutlich die Nutzung ein digitaler Fragebogen möglich, über den die Studierenden benachrichtigt würden und in den sie Fragen eintragen könnten, eine analoge Lösung wäre auch denkbar. Problematisch hieran wäre, sofern genügend Fragen gefunden würden, der große Arbeitsaufwand für die manuelle Erstellung der [SPARQL](#)-Abfragen, als auch die Möglichkeit der Ontologie, diese zu beantworten. Die Fragen würden von Menschen erstellt, die wirklich eine Antwort auf die Frage erhalten wollen, und würden sehr individuell und vielfältig gestaltet sein. Dies birgt aber wieder das Problem, dass sie von [QA](#)-Systemen aufgrund ihrer Art, Komplexität oder Formulierung nicht beantwortet werden können. Es ist aber trotzdem ein Ansatz, der betrachtet werden sollte.

7.2 SYSTEME

Die Erwartung war, dass das Feld mittlerweile reif genug für Systeme ist, die das einfache und präzise Nutzen von semantischem Question Answering ermöglichen. Es war aber sehr schwer, Systeme mit ausreichender Dokumentation und bereitgestelltem Programm zu finden, die noch dazu mit eigenen Daten und vielleicht sogar über mehrere Ontologien verwendbar waren. Die meisten davon funktionieren jedoch nicht.

Es muss allerdings auch gesagt werden, dass [SNIK](#) nicht besonders für semantisches Question Answering geeignet ist. Dieses geht meist, wie in Abschnitt 3.3 gesehen, nach der Methode vor, dass die Prädikate der [SPARQL](#)-Abfrage den Prädikaten des Satzes entsprechen sollen. Da [SNIK](#) auf verhältnismäßig wenigen Prädikaten beruht, deren Label noch dazu oft wenig in einem natürlichen Satz verwendet würden, ist es meist unpraktikabel, solche zu verwenden. Ein Ansatz für die Lösung dieses Problems ist die Bootstrapping the Web of Data ([BOA](#)) Pattern Library (Gerber und Ngo-mo, 2011), welche aus freiem Text [RDF](#)-Prädikate extrahieren soll, oder die manuelle Erweiterung durch alternative Labels.

An Ausnahmen wie [meta:isResponsibleForFunction](#) sieht man, dass diese deutlich besser funktionieren, da sie etwa in der Frage „What is the CIO responsible for?“ vorhanden sind. Prädikate wie [meta:entityTypeComponent](#) mit dem Label „entity type component“ haben es dort schwerer, da sie auf das Metamodell [SNIKS](#) zurückgreifen und nicht den normalen Sprachgebrauch widerspiegeln. Hilfreich dafür sind definitiv die Property Mappings von [QAnswer](#), aber diese haben auch nur eine auf die eingegebenen Lexikalisierungen begrenzte Wirkungsweite.

Problematisch an QAnswer ist, dass die Abfragen, denen der höchste Confidence-Wert zugeordnet wurde, manchmal nicht als erstes ausgeführt werden. Dies ist selbst der Fall, wenn der Wert unter 50% liegt. Unter 50% Confidence sieht QAnswer eine Antwort als falsch an und gibt sie nicht aus (Diefenbach, Giménez-García et al., 2020). Des Weiteren soll die Antwort mit dem höchsten Confidence-Wert ausgegeben werden, wobei es hierbei noch zu einigen Verrechnungen anderer Faktoren kommen kann (Diefenbach, 2018a). So führt QAnswer beispielsweise bei der Frage „What is used by Medical and Nursing Care planning?“, einer automatisch generierten Frage, die SPARQL-Abfrage mit den drei Prädikaten *function component*, *is involved in*, *is responsible for*, *supports* aus. Der Confidence-Wert für diese Antwort ist 53%, der Confidence-Wert für die richtige Antwort 93%; diese wird aber nicht aufgeführt. Die richtige Antwort enthält nur das Property *uses* und hat *Medical and Nursing Care planning* an der ersten Stelle des Tripels, ist also auch bedeutend einfacher.

Noch stärker sichtlich wird dieses Problem, wenn man einen Rechtschreibfehler einbaut und „Medicl“ statt „Medical“ schreibt. Hier wird eine andere falsche Antwort ausgeführt und gezeigt, diese hat nur einen Confidence-Wert von 35%, die richtige, die hinter einem Menu versteckt ist, wieder einen von 93%.

Es ist nicht, wie anfangs gewollt, gelungen, ein semantisches QA-System über alle drei Teilontologien zuverlässig verwendbar zu machen. Gelingen ist, ein QA-System über eine Teilontologie verwendbar zu machen, die Erfolgsrate ist beim untrainierten System jedoch schon überraschend gut.

Für die Verwendung der Teilontologien zusammen fehlt nicht nur ein System, dass diese Aufgabe bewältigt, sondern vor allem auch schlicht die Trainingsfragen. Bereits die aus dem blauen Buch waren nur teilweise ge-

eignet, für das Training für aller Teilontologien müsste man vermutlich Nutzerfragen von Studierenden sammeln. Noch dazu kommt das Problem der Ambiguität verschiedener Begriffe, welches mit verschiedenen Teilontologien eingeführt wird. Natürlich können die Ressourcen, die mit `skos:closeMatch` oder `skos:exactMatch` verbunden sind, als ähnlich, vielleicht sogar gleich definiert werden. Aber es gibt ja auch Ressourcen, die nur über `skos:broadMatch` oder gar nicht verbunden sind und trotzdem fast gleich heißen. Solche Ambiguitäten gilt es aufzulösen, das ist ein weiterer Grund, warum hier nur eine Teilontologie abgefragt wurde.

In Zukunft wäre das Erstellen eines eigenen, auf [SNIK](#) spezialisierten, [QA](#)-Systems denkbar, was aber aufgrund der hohen Komplexität unpraktisch erscheint.

Für eine Nutzung des Systems durch Studenten muss erprobt werden, wie brauchbar die Ergebnisse für Fragen sind, die sie wirklich stellen. Wenn das ein Mittelwert zwischen den Ergebnissen der automatisch generierten Fragen und den Lehrbuchfragen ist, dann stellt das eine gute Option dar, um zumindest einen Ansatz für eine Beantwortung der Fragen zu erhalten.

Die Systeme mit etwa 300 automatisch generierten Trainingsfragen, die im Trainingsset keine Lehrbuchfragen hatten, waren mit Abstand am erfolgreichsten. In einer möglichen praktischen Nutzung von [QAnswer](#) sollten auf jeden Fall solche verwendet werden.

LITERATUR

- Abadi, Martín, Ashish Agarwal, Paul Barham et al. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. URL: <https://www.tensorflow.org/>.
- Aleker, Wolfgang, Kirsten Krebsbach, Elfriede Kuntz et al. (2010). *Blickfeld Deutsch*. Schöningh. ISBN: 978-3-14-028235-2.
- Ammenwerth, Elske, Reinhold Haux, Petra Knaup-Gregori und Alfred Winter (2014). *IT-Projektmanagement im Gesundheitswesen*. 2. Aufl. Stuttgart, Germany: Schattauer. ISBN: 9783794530717.
- Anderson, Lorin W und David R Krathwohl (2001). *A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives*. Longman.
- Andhavarapu, Abhishek (2017). *Learning Elasticsearch*. Packt Publishing Ltd.
- Bahdanau, Dzmitry, Kyunghyun Cho und Yoshua Bengio (2014). „Neural machine translation by jointly learning to align and translate“. In: *arXiv preprint arXiv:1409.0473*.
- Berners-Lee, Tim (2006). „Linked data-design issues“. In: <http://www.w3.org/DesignIssues/LinkedData.html>.
- Berners-Lee, Tim, Robert Cailliau, Ari Luotonen, Henrik Frystyk Nielsen und Arthur Secret (1994). „The world-wide web“. In: *Communications of the ACM* 37.8, S. 76–82.
- Berners-Lee, Tim, Roy T. Fielding und Larry Masinter (Jan. 2005). *Uniform Resource Identifier (URI): Generic Syntax*. Techn. Ber. 3986. Internet

- Engineering Task Force (IETF), Network Working Group. URL: <https://tools.ietf.org/html/rfc3986> (besucht am 25. 03. 2020).
- Berners-Lee, Tim, Larry Masinter und Mark McCahill (Dez. 1994). *Uniform Resource Locators (URL)*. Techn. Ber. 1738. Internet Engineering Task Force (IETF), Network Working Group. URL: <https://tools.ietf.org/html/rfc1738> (besucht am 25. 03. 2020).
- Bizer, Christian, Tom Heath und Tim Berners-Lee (2011). „Linked data: The story so far“. In: *Semantic services, interoperability and web applications: emerging concepts*. IGI global, S. 205–227.
- Blevins, James P (2006). „22 English Inflection and Derivation“. In: *The handbook of English linguistics*, S. 507.
- Burtsev, Mikhail, Alexander Seliverstov, Rafael Airapetyan, Mikhail Arkhipov, Dilyara Baymurzina, Nickolay Bushkov, Olga Gureenкова, Taras Khakhulin, Yurii Kuratov, Denis Kuznetsov et al. (2018). „Deeppavlov: Open-source library for dialogue systems“. In: *Proceedings of ACL 2018, System Demonstrations*, S. 122–127.
- Cho, Kyunghyun, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk und Yoshua Bengio (2014). „Learning phrase representations using RNN encoder-decoder for statistical machine translation“. In: *arXiv preprint arXiv:1406.1078*.
- Chung, Junyoung, Caglar Gulcehre, KyungHyun Cho und Yoshua Bengio (2014). „Empirical evaluation of gated recurrent neural networks on sequence modeling“. In: *arXiv preprint arXiv:1412.3555*.
- Clarke, James, Vivek Srikumar, Mark Sammons und Dan Roth (2012). „An NLP Curator (or: How I Learned to Stop Worrying and Love NLP Pipelines).“ In: *LREC*, S. 3276–3283.

- Devlin, Jacob, Ming-Wei Chang, Kenton Lee und Kristina Toutanova (2018). „Bert: Pre-training of deep bidirectional transformers for language understanding“. In: *arXiv preprint arXiv:1810.04805*.
- Diefenbach, Dennis (2018a). *EP18305035*. European Patent Application.
- Diefenbach, Dennis (2018b). „Question answering over Knowledge Bases“. Diss. Université de Lyon.
- Diefenbach, Dennis, José Giménez-García, Andreas Both, Kamal Singh und Pierre Maret (2020). „QAnswer KG: Designing a portable question answering system over RDF data“. In: *European Semantic Web Conference*. Springer, S. 429–445.
- Dubey, Mohnish, Sourish Dasgupta, Ankit Sharma, Konrad Höffner und Jens Lehmann (2016). „AskNow: A framework for natural language query formalization in SPARQL“. In: *European Semantic Web Conference*. Springer, S. 300–316.
- Dudenredaktion (2013). *Duden, Die deutsche Rechtschreibung*. Bd. 26. Duden. ISBN: 978-3-411-04016-2.
- Gerber, Daniel und A-C Ngonga Ngomo (2011). „Bootstrapping the linked data web“. In: *1st Workshop on Web Scale Knowledge Extraction@ ISWC*. Bd. 2011, S. 61.
- Gruber, Tom (2016). „Ontology“. In: *Encyclopedia of Database Systems*. Hrsg. von Ling Liu und M. Tamer Özsu. New York, NY: Springer New York, S. 1–3. ISBN: 978-1-4899-7993-3. DOI: [10.1007/978-1-4899-7993-3_1318-2](https://doi.org/10.1007/978-1-4899-7993-3_1318-2).
- Hayes-Roth, Frederick (1985). „Rule-based systems“. In: *Communications of the ACM* 28.9, S. 921–932.
- Heinrich, Lutz J, René Riedl und Dirk Stelzer (2014). *Informationsmanagement: Grundlagen, Aufgaben, Methoden*. De Gruyter.

- Hirschman, Lynette und Robert Gaizauskas (2001). „Natural language question answering: the view from here“. In: *natural language engineering* 7.4, S. 275–300.
- Hitzler, Pascal (2021). „A review of the semantic web field“. In: *Communications of the ACM* 64.2, S. 76–83.
- Hitzler, Pascal, Markus Krötzsch, Sebastian Rudolph und York Sure (2007). *Semantic Web: Grundlagen*. Springer-Verlag.
- Hochreiter, Sepp und Jürgen Schmidhuber (1997). „Long short-term memory“. In: *Neural computation* 9.8, S. 1735–1780.
- Höffner, Konrad, Franziska Jahn, Christian Kücherer, Barbara Paech, Schneider Brigel, Martin Schöbel, Sebastian Stäubert und Alfred Winter (2017). „Technical Environment for Developing the SNIK Ontology of Information Management in Hospitals“. In: *Studies in Health Technology and Informatics* 243, S. 122–126.
- 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, The 17th World Congress of Medical and Health Informatics, Lyon*.
- Jones, Karen Sparck (1994). „Natural language processing: a historical review“. In: *Current issues in computational linguistics: in honour of Don Walker*, S. 3–16.
- Kotis, Konstantinos Ilias, Konstantina Zachila und Evaggelos Papanikolaou (2021). „Machine Learning Meets the Semantic Web“. In: *Artificial Intelligence Advances* 3.1.
- LeCun, Yann, Yoshua Bengio und Geoffrey Hinton (2015). „Deep learning“. In: *nature* 521.7553, S. 436–444.

- Lehmann, Jens, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer et al. (2015). „Dbpedia – A large-scale, multilingual knowledge base extracted from wikipedia“. In: *Semantic web* 6.2, S. 167–195.
- Mahesh, Batta (2020). „Machine learning algorithms-a review“. In: *International Journal of Science and Research (IJSR)*. [Internet] 9, S. 381–386.
- Mendes, Pablo N, Max Jakob, Andrés García-Silva und Christian Bizer (2011). „DBpedia spotlight: shedding light on the web of documents“. In: *Proceedings of the 7th international conference on semantic systems*, S. 1–8.
- Mikolov, Tomas, Kai Chen, Greg Corrado und Jeffrey Dean (2013). „Efficient estimation of word representations in vector space“. In: *arXiv preprint arXiv:1301.3781*.
- Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado und Jeff Dean (2013). „Distributed representations of words and phrases and their compositionality“. In: *Advances in neural information processing systems* 26.
- Miller, George A (1995). „WordNet: a lexical database for English“. In: *Communications of the ACM* 38.11, S. 39–41.
- Narayanan, Srini und Sanda Harabagiu (2004). *Question answering based on semantic structures*. Techn. Ber. INTERNATIONAL COMPUTER SCIENCE INST BERKELEY CA.
- Ngomo, Ngonga (2018). „9th challenge on question answering over linked data (QALD-9)“. In: *language* 7.1, S. 58–64.
- Orwell, George (1948). *Nineteen Eighty-Four*. Penguin Books.
- Perevalov, Aleksandr, Dennis Diefenbach, Ricardo Usbeck und Andreas Both (2022). „QALD-9-plus: A Multilingual Dataset for Question Answering over DBpedia and Wikidata Translated by Native Speakers“. In: 2022

- IEEE 16th International Conference on Semantic Computing (ICSC)*. IEEE, S. 229–234.
- Perevalov, Aleksandr, Xi Yan, Liubov Kovriguina, Longquan Jiang, Andreas Both und Ricardo Usbeck (2022). „Knowledge Graph Question Answering Leaderboard: A Community Resource to Prevent a Replication Crisis“. In: *arXiv preprint arXiv:2201.08174*.
- Peters, Matthew E., Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee und Luke Zettlemoyer (2018). „Deep contextualized word representations“. In: *CoRR abs/1802.05365*.
- Po, Laura, Nikos Bikakis, Federico Desimoni und George Papastefanatos (2020). „Linked data visualization: techniques, tools, and big data“. In: *Synthesis Lectures on Semantic Web: Theory and Technology* 10.1, S. 1–157.
- Radford, Alec, Karthik Narasimhan, Tim Salimans und Ilya Sutskever (2018). *Improving language understanding with unsupervised learning*. Techn. Ber.
- Rahman, Shafin, Salman Khan und Fatih Porikli (2018). „A unified approach for conventional zero-shot, generalized zero-shot, and few-shot learning“. In: *IEEE Transactions on Image Processing* 27.11, S. 5652–5667.
- Reshamwala, Alpa, Dharendra Mishra und Prajakta Pawar (2013). „Review on natural language processing“. In: *IRACST Engineering Science and Technology: An International Journal (ESTIJ)* 3.1, S. 113–116.
- Roszeitis, Arne (2022). „Automatische Generierung komplexer Fragen zum Informationsmanagement auf der Basis der SNIK-Ontologie“. Fakultät für Mathematik und Informatik der Universität Leipzig.
- Rush, Alexander M (2018). „The annotated transformer“. In: *Proceedings of workshop for NLP open source software (NLP-OSS)*, S. 52–60.
- Schaaf, Michael, Franziska Jahn, Kais Tahar, Christian Kucherer, Alfred Winter und Barbara Paech (2015). „Entwicklung und Einsatz einer Domä-

- nenontologie des Informationsmanagements im Krankenhaus“. In: *Informatik 2015*. Lecture Notes in Informatics 246. Hrsg. von Douglas W. Cunningham, Petra Hofstedt, Klaus Meer und Ingo Schmitt.
- Shrestha, Ajay und Ausif Mahmood (2019). „Review of deep learning algorithms and architectures“. In: *IEEE Access* 7, S. 53040–53065.
- Sleator, Daniel DK und Davy Temperley (1995). „Parsing English with a link grammar“. In: *arXiv preprint cmp-lg/9508004*.
- Sutskever, Ilya, Oriol Vinyals und Quoc V Le (2014). „Sequence to sequence learning with neural networks“. In: *Advances in neural information processing systems* 27.
- Talmy, Leonard (1985). „Lexicalization patterns: Semantic structure in lexical forms“. In: *Language typology and syntactic description* 3.99, S. 36–149.
- Taylor, Ann, Mitchell Marcus und Beatrice Santorini (2003). „The Penn treebank: an overview“. In: *Treebanks*, S. 5–22.
- Vargas, Rocio, Amir Mosavi und Ramon Ruiz (2017). „Deep learning: a review“. In.
- Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser und Illia Polosukhin (2017). „Attention is all you need“. In: *Advances in neural information processing systems*, S. 5998–6008.
- Vollmers, D., R. Jalota, D. Moussallem, H. Topiwala, A. C. N. Ngomo, R. Usbeck und IAIS (2021). „Knowledge Graph Question Answering using Graph-Pattern Isomorphism“. In: *Further with Knowledge Graphs: Proceedings of the 17th International Conference on Semantic Systems, 6-9 September 2021, Amsterdam, The Netherlands*. Bd. 53. IOS Press, S. 103.

- Von Ahn, Luis, Benjamin Maurer, Colin McMillen, David Abraham und Manuel Blum (2008). „recaptcha: Human-based character recognition via web security measures“. In: *Science* 321.5895, S. 1465–1468.
- Walch, Marcel, Mark Colley und Michael Weber (2019). „CooperationCaptcha: on-the-fly object labeling for highly automated vehicles“. In: *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, S. 1–6.
- 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>.
- Yan, Hao, Shuai Ding und Torsten Suel (2009). „Inverted index compression and query processing with optimized document ordering“. In: *Proceedings of the 18th international conference on World wide web*, S. 401–410.
- Zheng, Weiguo und Mei Zhang (2019). „Question answering over knowledge graphs via structural query patterns“. In: *arXiv preprint arXiv:1910.09760*.
- Zhu, Yukun, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba und Sanja Fidler (2015). „Aligning books and movies: Towards story-like visual explanations by watching movies and reading books“. In: *Proceedings of the IEEE international conference on computer vision*, S. 19–27.
- Zou, Lei, Ruizhe Huang, Haixun Wang, Jeffrey Xu Yu, Wenqiang He und Dongyan Zhao (2014). „Natural language question answering over RDF: a graph data driven approach“. In: *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, S. 313–324.

APPENDIX

SYNTAKTISCHES UND POS-TAGGING

Sowohl **POS-** als auch syntaktisches Tagging, also das Tagging von entweder einzelnen Wörtern oder Satzbauteilen, geschieht nach verschiedenen Standards. In dieser Arbeit wird, wenn es denn geschieht, die Penn Treebank-Notation verwendet. Diese wird im Folgenden so aufgeschlüsselt, wie es hier benötigt wird. Es wurden jedoch einzelne Zeichen o.ä. ausgelassen, wie etwa einzelne Tags für verschiedene Anführungszeichen beim **POS-**Tagging.

Abkürzung	Bedeutung
ADJP	Adjective phrase
ADVP	Adverb phrase
NP	Noun phrase
PP	Prepositional phrase
S	Simple declarative clause
SBAR	Subordinate clause
SBARQ	Direct question introduced by wh-element
SINV	Declarative sentence with subject-aux inversion
SQ	Yes/no questions and subconstituent of SBARQ excluding wh-element
VP	Verb phrase
WHADVP	Wh-adverb phrase

Abkürzung	Bedeutung
WHNP	Wh-noun phrase
WHPP	Wh-prepositional phrase
X	Constituent of unknown or uncertain category

Tabelle A.1: Penn Treebank-Notation syntaktischer Tags nach Taylor, Marcus et al. (2003)

Abkürzung	Bedeutung	Beispiel
CC	Coordinating conjunction	and, or
CD	Cardinal Number	one, 5
DT	Determiner	both, some
FW	Foreign word	oui, si
IN	Preposition/subordinating conjunction	upon, if
JJ	adjective	complex, forty-second
JJR	Adjective, Comparative	simpler, healthier
JJS	Adjective, Superlative	simplest, healthiest
LS	List item marker	-, 1.
MD	Modal	could, can't
NN	Noun, singular or mass	hospital, knowledge
NNS	Noun, plural	hospitals, systems
NNP	Proper noun, singular	Leipzig, Mustermann
NNPS	Proper noun, plural	Americas, Joneses
PDT	Predeterminer	all, both

Abkürzung	Bedeutung	Beispiel
POS	Possessive ending	's
PRP	Personal pronoun	us, it
PP	Possessive pronoun	our, theirs
RB	Adverb	definitely, very
RBR	Adverb, comparative	greater, more
RBS	Adverb, superlative	greatest, most
RP	Particle	up, away
SYM	Symbol	⊆, §
TO	Infinitival to	to
UH	Interjection	uh, wow
VB	Verb, base form	learn, stay
VBD	Verb, past tense	learned, stayed
VBG	Verb, gerund/present participle	learning, staying
VBN	Verb, past participle	learned, stayed
VBP	Verb, non-3rd person singular present	learn, stay
VBZ	Verb, 3rd person singular present	learns, stays
WDT	Wh-determiner	whichever, what
WP\$	Wh-pronoun	which, whom
WP\$	Possessive wh-pronoun	whose
WRB	Wh-adverb	how, why

Tabelle A.2: Penn Treebank-Notation von POS-Tags nach Taylor, Marcus et al. (2003)

KLASSIFIZIERUNG DER TEXTBUCHFRAGEN

B.1 ERSTE EINORDNUNG

Die erste Einordnung der Fragen aus Winter, Haux et al. (2011) erfolgt hier¹:

Tabelle B.1: Klassifizierung der Fragen aus Winter, Haux et al. (2011) basierend auf Roszeitis (2022). Fett: Unterschiedliche Klassifizierung bezüglich der Eignung. Komma bei Frageart: Teilfragen haben unterschiedliche Frageart. Fragetyp und -art nach Anfangsbuchstaben abgekürzt.

Kapitel/ID	Frage	Fragetyp	Frageart	Eignung	Original
1/1	Why is systematic information processing in health care institutions important?	K	S	✗	✗
1/2	What are appropriate models for health information systems?	F	S	✓	✓

¹ Basierend auf Roszeitis (2022). Als CSV verfügbar unter:

<https://github.com/Yagnap/BeLL-Question-Answering-auf-Linked-Data-SNIK/blob/main/Data/Tabellen/Anhang/fragenklassifikation.csv>

Kapitel/ID	Frage	Frageart	Frageart	Eignung	Original
1/3	How do health information systems look like and what architectures are appropriate?	P, F	Z	✗	✗
1/4	How can we assess the quality of health information systems?	P	S	✗	✗
1/5	How can we strategically manage health information systems?	P	S	✗	✗
1/6	How can good information systems be designed and maintained?	P	S	✗	✗
2/1	What is the significance of information systems for health care?	K	S	✗	✗
2/2	How does technical progress affect health care?	P	S	✗	✗
2/3	Why is systematic information management important?	K	S	✗	✗
3/1	What is the difference between data, information and knowledge?	V	Z	✗	✗
3/2	What are information systems, and what are their components?	F	Z	✗	✗
3/3	What is information management?	F	S	✓	✓
4/1	What are hospital information systems?	F	S	✓	✓

Kapitel/ID	Frage	Frageart	Frageart	Eignung	Original
4/2	What are transinstitutional health information systems?	F	S	✓	✓
4/3	What are challenges for health information systems?	F	S	✓	✓
4/4	What are electronic health records?	F	S	✓	✓
5/1	What are models, metamodels and reference models?	F	Z	✗	✗
5/2	What are typical metamodels for modeling various aspects of HIS?	F	S	✓	✓
5/3	What is 3LGM ² ?	F	S	✓	✓
5/4	What are typical reference models for HIS?	F	S	✓	✓
6/1	What kind of data has to be processed in hospitals?	F	S	✓	✓
6/2	What are the main hospital functions?	F	S	✓	✓
6/3	What are the typical information processing tools in hospitals?	F	S	✓	✓
6/4	What are different architectures of HIS?	F	S	✓	✓
6/5	How can integrity and integration be achieved within HIS?	P	S	✗	✗

Kapitel/ID	Frage	Frage typ	Frage art	Eignung	Original
6.4/1	What application components are used in hospitals, and what are their characteristics?	F	Z	✗	✗
6.5/1	How can architectures of HIS be categorized?	F	S	✓	✗
6.5/2	What differs integrity from integration?	V	Z	✗	✗
6.5/3	What standards and technologies are available to support integration of HIS?	F	Z	✗	✓
6.5/4	How can integration efforts be reduced by decreasing the variety of application components in a HIS?	P	K	✗	✗
6.6/1	What computer-based and non-computer-based physical data processing systems can be found in hospitals?	F	Z	✗	✗
6.6/2	What is meant by the term „infrastructure“?	F	S	✓	✓

Kapitel/ID	Frage	Frage typ	Frage art	Eignung	Original
6.7/1	How can physical data processing systems be grouped and arranged in order to support application components in an optimal way? What architectures can this result?	P, F	Z	✗	✗
6.7/2	What is meant by physical integration?	F	S	✓	✓
6.7/3	How do modern computing centers look like?	P	S	✗	✗
7/1	How do architectures of transinstitutional health information systems differ from those of hospital information systems?	V	Z	✗	✗
7/2	What additional challenges do we have to cope with?	F	S	✗	✗
7/3	Which strategies are appropriate for maintaining electronic health records in a transinstitutional health information system?	F	K	✓	✗
8/1	Which facets of quality have to be considered in HIS?	F	S	✓	✓

Kapitel/ID	Frage	Frageart	Frageart	Eignung	Original
8/2	What are the characteristics of the quality of structures in HIS?	F	S	✓	✗
8/3	What are the characteristics of the quality of processes of HIS?	F	S	✓	✗
8/4	What are the characteristics of quality of outcome of HIS?	F	S	✓	✗
8/5	What does information management have to balance in order to increase the quality of a HIS?	K	S	✗	✗
8/6	How can quality of HIS be evaluated?	F	K	✓	✗
8.2/1	What criteria for quality of data exist?	F	S	✓	✗
8.2/2	What criteria for computer-based application components and physical data processing systems exist?	F	Z	✗	✗
8.2/3	What criteria for the overall HIS architecture exist?	F	S	✓	✗
8.3/1	What are the characteristics of the quality of processes of HIS?	F	S	✓	✗

Kapitel/ID	Frage	Frageart	Frageart	Eignung	Original
8.4/1	What are the characteristics of quality of outcome of HIS especially in hospitals?	F	S	✓	✗
8.5/1	What does information management have to balance in order to increase the quality of health information systems?	K	S	✗	✗
8.6/1	What are major phases of an IT evaluation study?	F	S	✓	✗
8.6/2	What are major IT evaluation methods?	F	S	✓	✗
9/1	What does information management mean and how can strategic, tactical and operational information management be differentiated?	F, V	Z	✗	✗
9/2	What organizational structures are appropriate for information management in hospitals?	F	S	✓	✗
9/3	What are the tasks and methods for strategic HIS planning?	F	K	✓	✗
9/4	What are the tasks and methods for strategic HIS monitoring?	F	K	✓	✗

Kapitel/ID	Frage	Frage typ	Frage art	Eignung	Original
9/5	What are the tasks and methods for strategic HIS directing?	F	K	✓	✗
9/6	How can experts for information management in hospitals be gained?	P	S	✗	✗
9.2/1	What does information management in general and in hospitals encompass?	K	K	✗	✗
9.2/2	What are the three main scopes of information management?	F	K	✓	✓
9.2/3	What are the tasks of strategic, tactical and operational information management in hospitals?	F	Z	✗	✗
9.2/4	What is meant by IT service management and how is it related to information management?	F	Z	✗	✗
9.3.4.2/1	Which organizational units are involved in information management?	F	K	✓	✓
9.3.4.2/2	Which boards and persons are involved in information management?	F	section	✓	✓

Kapitel/ID	Frage	Frage typ	Frage art	Eignung	Original
9.3.4.2/3	Who is responsible for strategic information management?	F	S	✓	✓
9.3.4.2/4	Who is responsible for tactical information management?	F	S	✓	✓
9.3.4.2/5	Who is responsible for operational information management?	F	S	✓	✓
9.3.4.2/6	Who is the CIO, and what is his or her responsibility?	F	Z	✗	✗
9.4/1	What are the typical tasks for strategic HIS planning?	F	K	✓	✓
9.4/2	What are the typical methods for strategic HIS planning?	F	K	✓	✓
9.4/3	What is the goal and typical structure of a strategic information management plan?	F	Z	✗	✗
9.5/1	What are the typical tasks of strategic HIS monitoring?	F	S	✓	✓
9.5/2	What are the typical methods of strategic HIS monitoring?	F	S	✓	✓
9.6/1	What are the typical tasks of strategic HIS directing?	F	S	✓	✓

Kapitel/ID	Frage	Frage typ	Frage art	Eignung	Original
9.6/2	What are the typical methods of strategic HIS directing?	K	S	✓	✓
10/1	What are health care networks?	F	S	✓	✓
10/2	How can health care networks be described?	F	S	✓	✗
10/3	What organizational structures are appropriate for information management in health care networks?	F	S	✓	✗
10/4	How can good information systems be maintained?	P	S	✗	✗

B.2 FORMULIERUNG DER SPARQL-ABFRAGEN

Im Folgenden wurden die natürlichsprachigen Fragen mit SPARQL-Abfragen beantwortet²:

```
# 1/2 What are appropriate models for health information systems?
SELECT DISTINCT ?s1
```

² Als CSV verfügbar unter:

https://github.com/Yagnap/BeLL-Question-Answering-auf-Linked-Data-SNIK/blob/main/Data/Tabellen/Anhang/formulierung_sparql_bb_textbuchfragen.csv

WHERE

```
{ ?s1 rdfs:subClassOf+ bb:Model . }
```

3/3 What is information management?

SELECT DISTINCT ?o1

WHERE

```
{ VALUES ?o1 {bb:InformationManagement } . }
```

4/1 What are hospital information systems?

SELECT DISTINCT ?o1

WHERE

```
{ VALUES ?o1 { bb:HospitalInformationSystem } . }
```

4/2 What are transinstitutional health information systems?

SELECT DISTINCT ?o1

WHERE

```
{ VALUES ?o1 { bb:TransinstitutionalHealthInformationSystem } . }
```

4/4 What are electronic health records?

SELECT DISTINCT ?o1

WHERE

```
{ VALUES ?o1 { bb:ElectronicHealthRecord } . }
```

5/2 What are typical metamodels for modeling various aspects of

↔ HIS?

SELECT DISTINCT ?s1

WHERE

```
{ ?s1 rdfs:subClassOf+ bb:Metamodel . }
```

```
# 5/3 What is 3LGM2?
```

```
SELECT DISTINCT ?o1
WHERE
  { VALUES ?o1 { bb:3LMG2 } . }
```

```
# 5/4 What are typical reference models for HIS?
```

```
SELECT DISTINCT ?s1
WHERE
  { ?s1 rdfs:subClassOf+ bb:ReferenceModel . }
```

```
# 6/2 What are the main hospital functions?
```

```
SELECT DISTINCT ?s1
WHERE
  { ?s1 rdfs:subClassOf bb:HospitalFunction . }
```

```
# 6/4 What are different architectures of HIS?
```

```
SELECT DISTINCT ?s1
WHERE
  { ?s1 rdfs:subClassOf+ bb:HisArchitecture . }
```

```
# 6.5/1 How can architectures of HIS be categorized?
```

```
SELECT DISTINCT ?s1
WHERE
  { ?s1 rdfs:subClassOf+ bb:HisArchitecture . }
```

```
# 6.6/2 What is meant by the term "infrastructure"?
```

```
SELECT DISTINCT ?o1
WHERE
  { VALUES ?o1 { bb:HisInfrastructure } . }
```

6.7/2 What is meant by physical integration?

```
SELECT DISTINCT ?o1
WHERE
  { VALUES ?o1 { bb:PhysicalIntegration } . }
```

7/3 Which strategies are appropriate for maintaining electronic
 ↔ health records in a transinstitutional health information
 ↔ system?

```
SELECT DISTINCT ?s1
WHERE
  { ?s1 rdfs:subClassOf bb:EhrStrategy . }
```

8/1 Which facets of quality have to be considered in HIS?

```
SELECT DISTINCT ?s1
WHERE
  { bb:HisQuality meta:entityTypeComponent ?o1 . }
```

8/2 What are the characteristics of the quality of structures in
 ↔ HIS?

```
SELECT DISTINCT ?s2
WHERE
  { bb:QualityOfHISStructures meta:entityTypeComponent ?o1 .
    ?s2 rdfs:subClassOf ?o1 . }
```

8/3 What are the characteristics of the quality of processes of
 ↔ HIS?

```
SELECT DISTINCT ?s1
WHERE
```

```
{ ?s1 rdfs:subClassOf bb:QualityOfHISProcesses . }
```

```
# 8/6 How can quality of HIS be evaluated?
```

```
SELECT DISTINCT ?s1
```

```
WHERE
```

```
{ ?s1 rdfs:subClassOf+ bb:EvaluationMethod . }
```

```
# 8.2/1 What criteria for quality of data exist?
```

```
SELECT DISTINCT ?s1
```

```
WHERE
```

```
{ ?s1 rdfs:subClassOf bb:QualityOfData . }
```

```
# 8.2/3 What criteria for the overall HIS architecture exist?
```

```
SELECT DISTINCT ?s1
```

```
WHERE
```

```
{ ?s1 rdfs:subClassOf bb:QualityOfHISArchitecture . }
```

```
# 8.3/1 What are the characteristics of the quality of processes  
↔ of HIS?
```

```
SELECT DISTINCT ?s1
```

```
WHERE
```

```
{ ?s1 rdfs:subClassOf bb:QualityOfHISProcesses . }
```

```
# 8.6/1 What are major phases of an IT evaluation study?
```

```
SELECT DISTINCT ?o1
```

```
WHERE
```

```
{ bb:ItEvaluationStudyManagementAndExecution
```

```
↔ meta:functionComponent ?o1 . }
```

8.6/2 What are major IT evaluation methods?

```
SELECT DISTINCT ?s1
WHERE
  { ?s1 rdfs:subClassOf bb:EvaluationMethod . }
```

9/3 What are the tasks and methods for strategic HIS planning?

```
SELECT DISTINCT ?o1
WHERE
  { bb:StrategichISPlanning meta:functionComponent ?o1 . }
```

9/4 What are the tasks and methods for strategic HIS monitoring?

```
SELECT DISTINCT ?o1
WHERE
  { bb:StrategichISMonitoring meta:functionComponent ?o1 . }
```

9/5 What are the tasks and methods for strategic HIS directing?

```
SELECT DISTINCT ?o1
WHERE
  { bb:StrategichISDirecting meta:functionComponent ?o1 . }
```

9.2/2 What are the three main scopes of information management?

```
SELECT DISTINCT ?o1
WHERE
  { bb:InformationManagement meta:functionComponent ?o1 . }
```

9.3.4.2/1 Which organizational units are involved in information
↔ management?

```
SELECT DISTINCT ?s1
WHERE
```

```
{ ?s1 meta:functionComponent bb:InformationManagement . }
```

```
# 9.3.4.2/2 Which boards and persons are involved in information  
↔ management?
```

```
SELECT DISTINCT ?s1
```

```
WHERE
```

```
{ ?s1 meta:isResponsibleForFunction bb:InformationManagement . }
```

```
# 9.4/2 What are the typical tasks for strategic HIS planning?
```

```
SELECT DISTINCT ?s1
```

```
WHERE
```

```
{ ?s1 rdfs:subClassOf bb:StrategichISPlanning . }
```

```
# 9.5/1 What are the typical tasks of strategic HIS monitoring?
```

```
SELECT DISTINCT ?s1
```

```
WHERE
```

```
{ ?s1 rdfs:subClassOf bb:StrategichISMonitoring . }
```

```
# 9.5/2 What are the typical methods of strategic HIS monitoring?
```

```
SELECT DISTINCT ?o1
```

```
WHERE
```

```
{ bb:StrategichISMonitoring meta:uses ?o1 . }
```

```
# 9.6/1 What are the typical tasks of strategic HIS directing?
```

```
SELECT DISTINCT ?o1
```

```
WHERE
```

```
{ bb:StrategichISDirecting meta:functionComponent ?o1 . }
```

```
# 9.6/2 What are the typical methods of strategic HIS directing?
```

```

SELECT DISTINCT ?o1
WHERE
  { bb:StrategichISDirecting meta:uses ?o1 . }

# 10/1 What are health care networks?
SELECT DISTINCT ?o1
WHERE
  { VALUES ?o1 {bb:HealthCareNetwork} . }

# 10/2 How can health care networks be described?
SELECT DISTINCT ?o1
WHERE
  { VALUES ?o1 {bb:HealthCareNetwork} . }

```

B.3 ANTWORTEN AUF DIE SPARQL-ABFRAGEN

Zuletzt eine Auflistung der Antworten auf die SPARQL-Abfragen aus Anhang B.2. Diese können über <https://www.snik.eu/sparql> ausgegeben werden.

Frage 1/2: What are appropriate models for health information systems?

- Technical Model (`bb:TechnicalModel`)
- Strategic Alignment Model (`bb:StrategicAlignmentModel`)
- Reference Model (`bb:ReferenceModel`)
- Organizational Model (`bb:OrganizationalModel`)
- OpenEHR Model of Processes (`bb:OpenEHRModelOfProcesses`)

- OpenEHR Model of Content (`bb:OpenEHRModelOfContent`)
- Model of the Planned HIS (`bb:ModelOfThePlannedHIS`)
- Model of the Current HIS (`bb:ModelOfTheCurrentHIS`)
- Information System Model (`bb:InformationSystemModel`)
- Information Processing Model (`bb:InformationProcessingModel`)
- Functional Model (`bb:FunctionalModel`)
- Data Reference Model (`bb:DataReferenceModel`)
- Data Model (`bb:DataModel`)
- Business Reference Model (`bb:BusinessReferenceModel`)
- Business Process Model (`bb:BusinessProcessModel`)
- UML Activity Diagram (`bb:UmlActivityDiagram`)
- Process Chain (`bb:ProcessChain`)
- Petri Net (`bb:PetriNet`)
- Event-Driven Process Chain (`bb:EventDrivenProcessChain`)
- UML Class Diagram (`bb:UmlClassDiagram`)
- Class Diagram (`bb:ClassDiagram`)
- Reference Model for the Domain Layer of Hospital Information Systems (`bb:ReferenceModelForTheDomainLayerOfHospitalInformationSystems`)
- OAI3 (`bb:OAI3`)

- ISO/OSI Reference Model (`bb:IsoosiReferenceModel`)
- IHE Integration Profile (`bb:IheIntegrationProfile`)
- HL7 Reference Information Model (`bb:HL7ReferenceInformationModel`)
- Tan's Critical Success Factor Approach (`bb:TansCriticalSuccessFactorApproach`)
- Component Alignment Model (`bb:ComponentAlignmentModelOfMartin`)
- IHE Patient Demographics Query (`bb:IhePatientDemographicsQuery`)
- Cross-Enterprise Document Sharing (`bb:CrossEnterpriseDocumentSharing`)

Frage 3/3: What is information management?

- Information Management (`bb:InformationManagement`)

Frage 4/1: What are hospital information systems?

- Hospital Information System (`bb:HospitalInformationSystem`)

Frage 4/2: What are transinstitutional health information systems?

- Transinstitutional Health Information System
(`bb:TransinstitutionalHealthInformationSystem`)

Frage 4/4: What are electronic health records?

- Electronic Health Record (`bb:ElectronicHealthRecord`)

Frage 5/2: What are typical metamodels for modeling various aspects of HIS?

- Technical Metamodel (`bb:TechnicalMetamodel`)

- Organizational Metamodel (`bb:OrganizationalMetamodel`)
- Information System Metamodel (`bb:InformationSystemMetamodel`)
- Functional Metamodel (`bb:FunctionalMetamodel`)
- Data Metamodel (`bb:DataMetamodel`)
- Business Process Metamodel (`bb:BusinessProcessMetamodel`)
- HL7 Reference Information Model (`bb:HL7ReferenceInformationModel`)
- 3LGM² (`bb:3LGM2`)
- 3LGM²-S (`bb:3LGM2S`)
- 3LGM²-M (`bb:3LGM2M`)
- 3LGM²-B (`bb:3LGM2B`)

Frage 5/3: What is 3LGM2?

- 3LGM² (`bb:3LGM2`)

Frage 5/4: What are typical reference models for HIS?

- Reference Model for the Domain Layer of Hospital Information Systems (`bb:ReferenceModelForTheDomainLayerOfHospitalInformationSystems`)
- OAI3 (`bb:OAI3`)
- ISO/OSI Reference Model (`bb:IsoosiReferenceModel`)
- IHE Integration Profile (`bb:IheIntegrationProfile`)
- HL7 Reference Information Model (`bb:HL7ReferenceInformationModel`)

- IHE Patient Demographics Query (`bb:IhePatientDemographicsQuery`)
- Cross-Enterprise Document Sharing (`bb:CrossEnterpriseDocumentSharing`)

Frage 6/2: What are the main hospital functions?

- Administrative Function (`bb:Administration`)
- Research and Education Function (`bb:EducationResearch`)
- Management (`bb:Management`)
- Patient Care(`bb:PatientCare`)

Frage 6/4: What are different Architectures of HIS?

- Homogeneous Architecture (`bb:HomogeneousArchitecture`)
- Heterogeneous Architecture (`bb:HeterogeneousArchitecture`)

Frage 6.5/1: How can architectures of HIS be categorized?

- Homogeneous Architecture (`bb:HomogeneousArchitecture`)
- Heterogeneous Architecture (`bb:HeterogeneousArchitecture`)

Frage 6.6/2: What is meant by the term „infrastructure“?

- HIS Infrastructure (`bb:HisInfrastructure`)

Frage 6.7/2: What is meant by the term physical integration?

- Physical Integration (`bb:PhysicalIntegration`)

Frage 7/3: Which strategies are appropriate for maintaining electronic health records in a transinstitutional health information system?

- The Strategy of Independent Health Banks
([bb:TheStrategyOfIndependentHealthBanks](#))
- The Regional- or National-Centric Strategy
([bb:TheRegionalOrNationalCentricStrategy](#))
- The Provider-Centric Strategy ([bb:TheProviderCentricStrategy](#))
- Patient-Centric Strategy ([bb:PatientCentricStrategy](#))

Frage 8/1: Which facets of quality have to be considered in HIS?

- Quality of HIS Outcome ([bb:QualityOfHISOutcome](#))
- Quality of HIS Processes ([bb:QualityOfHISProcesses](#))
- Quality of HIS Structures ([bb:QualityOfHISStructures](#))

Frage 8/2: What are the characteristics of the quality of structures in HIS?

- Access Integration ([bb:AccessIntegration](#))
- Data Integration ([bb>DataIntegration](#))
- Functional Coverage of the Application Component
([bb:FunctionalCoverageOfTheApplicationComponent](#))
- Functional Integration ([bb:FunctionalIntegration](#))
- ISO 9241-110 User Interface Design Quality
([bb:ISO9241110UserInterfaceDesignQuality](#))
- Performance of Application Components
([bb:PerformanceOfApplicationComponents](#))

- Semantic Integration (`bb:SemanticIntegration`)
- Software Ergonomics (`bb:SoftwareErgonomics`)
- Software Quality (`bb:SoftwareQuality`)
- Stability of Application Components (`bb:StabilityOfApplicationComponents`)
- Accuracy (`bb:Accuracy`)
- Authenticity of Data (`bb:AuthenticityOfData`)
- Availability of Data (`bb:AvailabilityOfData`)
- Completeness (`bb:Completeness`)
- Confidentiality (`bb:Confidentiality`)
- Durability (`bb:Durability`)
- Integrity of Data (`bb:IntegrityOfData`)
- Relevancy (`bb:Relevancy`)
- Reliability of Data (`bb:ReliabilityOfData`)
- Security of Data (`bb:SecurityOfData`)
- Standardization of Data (`bb:StandardizationOfData`)
- Adaptability of the HIS (`bb:AdaptabilityOfTheHIS`)
- Balance of Computer-Based and Non-Computer-Based Tools
(`bb:BalanceOfComputerBasedAndNonComputerBasedTools`)
- Balance of Data Security and Working Processes
(`bb:BalanceOfDataSecurityAndWorkingProcesses`)

- Balance of Documentation Quality and Documentation Efforts
(`bb:BalanceOfDocumentationQualityAndDocumentationEfforts`)
- Balance of Functional Leanness and Functional Redundancy
(`bb:BalanceOfFunctionalLeannessAndFunctionalRedundancy`)
- Balance of Homogeneity and Heterogeneity
(`bb:BalanceOfHomogeneityAndHeterogeneity`)
- Controlled Redundancy of Data (`bb:ControlledRedundancyOfData`)
- Functional Leanness (`bb:FunctionalLeanness`)
- Functional Redundancy (`bb:FunctionalRedundancy`)
- Heterogeneity of the HIS Architecture (`bb:HeterogeneityOfTheHISArchitecture`)
- Homogeneity of the HIS Architecture (`bb:HomogeneityOfTheHISArchitecture`)
- Saturation (`bb:Saturation`)
- Transparency (`bb:Transparency`)

Frage 8/3: What are the characteristics of the quality of processes of HIS?

- Controlled Transcription of Data (`bb:ControlledTranscriptionOfData`)
- Efficiency of Information Logistics (`bb:EfficiencyOfInformationLogistics`)
- Leanness of Information Processing Tools
(`bb:LeannessOfInformationProcessingTools`)
- Multiple Usability of Data (`bb:MultipleUsabilityOfData`)
- Patient-Centered Information Processing
(`bb:PatientCenteredInformationProcessing`)

Frage 8/6: How can quality of HIS be evaluated?

- Usability Study (`bb:UsabilityStudy`)
- SWOT Analysis (`bb:SwotAnalysis`)
- Quantitative Evaluation Method (`bb:QuantitativeEvaluationMethod`)
- Qualitative Evaluation Method (`bb:QualitativeEvaluationMethod`)
- Delphi Survey (`bb:DelphiSurvey`)
- Qualitative Observation (`bb:QualitativeObservation`)
- Qualitative Interview (`bb:QualitativeInterview`)
- Qualitative Content Analysis (`bb:QualitativeContentAnalysis`)
- Case Study (`bb:CaseStudy`)
- Utility Analysis (`bb:UtilityAnalysis`)
- User Survey (`bb:UserSurvey`)
- Time Measurement (`bb:TimeMeasurement`)
- Return-on-Investment Study (`bb:ReturnOnInvestmentStudy`)
- Quantitative Questionnaire (`bb:QuantitativeQuestionnaire`)
- Patient Satisfaction Survey (`bb:PatientSatisfactionSurvey`)
- Event Counting Study (`bb:EventCountingStudy`)
- Effectiveness Study (`bb:EffectivenessStudy`)
- Cost-Effectiveness Analysis (`bb:CostEffectivenessAnalysis`)

- Cost–Benefit Analysis ([bb:CostBenefitAnalysis](#))
- Consensus Method ([bb:ConsensusMethod](#))
- Work Sampling ([bb:WorkSampling](#))
- Time-Motion Analysis ([bb:TimeMotionAnalysis](#))
- Unstructured Interview ([bb:UnstructuredInterview](#))
- Semistructured Interview ([bb:SemistructuredInterview](#))

Frage 8.2/1: What criteria for quality of data exist?

- Accuracy ([bb:Accuracy](#))
- Authenticity of Data ([bb:AuthenticityOfData](#))
- Availability of Data ([bb:AvailabilityOfData](#))
- Completeness ([bb:Completeness](#))
- Confidentiality ([bb:Confidentiality](#))
- Durability ([bb:Durability](#))
- Integrity of Data ([bb:IntegrityOfData](#))
- Relevancy ([bb:Relevancy](#))
- Reliability of Data ([bb:ReliabilityOfData](#))
- Security of Data ([bb:SecurityOfData](#))
- Standardization of Data ([bb:StandardizationOfData](#))

Frage 8.2/3: What criteria for the overall HIS architecture exist?

- Adaptability of the HIS (`bb:AdaptabilityOfTheHIS`)
- Balance of Computer-Based and Non-Computer-Based Tools
(`bb:BalanceOfComputerBasedAndNonComputerBasedTools`)
- Balance of Data Security and Working Processes
(`bb:BalanceOfDataSecurityAndWorkingProcesses`)
- Balance of Documentation Quality and Documentation Efforts
(`bb:BalanceOfDocumentationQualityAndDocumentationEfforts`)
- Balance of Functional Leanness and Functional Redundancy
(`bb:BalanceOfFunctionalLeannessAndFunctionalRedundancy`)
- Balance of Homegeneity and Heterogeneity
(`bb:BalanceOfHomegeneityAndHeterogeneity`)
- Controlled Redundancy of Data (`bb:ControlledRedundancyOfData`)
- Functional Leanness (`bb:FunctionalLeanness`)
- Functional Redundancy (`bb:FunctionalRedundancy`)
- Heterogeneity of the HIS Architecture (`bb:HeterogeneityOfTheHISArchitecture`)
- Homogeneity of the HIS Architecture (`bb:HomogeneityOfTheHISArchitecture`)
- Saturation (`bb:Saturation`)
- Transparency (`bb:Transparency`)

Frage 8.3/1: What are the characteristics of the quality of processes of HIS?

- Controlled Transcription of Data (`bb:ControlledTranscriptionOfData`)

- Efficiency of Information Logistics ([bb:EfficiencyOfInformationLogistics](#))
- Leanness of Information Processing Tools
([bb:LeannessOfInformationProcessingTools](#))
- Multiple Usability of Data ([bb:MultipleUsabilityOfData](#))
- Patient-Centered Information Processing
([bb:PatientCenteredInformationProcessing](#))

Frage 8.6/1: What are major phases of an IT evaluation study?

- Execution of an IT Evaluation Study ([bb:ExecutionOfAnITEvaluationStudy](#))
- First Study Design ([bb:FirstStudyDesign](#))
- Operationalization of Methods and Detailed Study Plan
([bb:OperationalizationOfMethodsAndDetailedStudyPlan](#))
- Report and Publication of Study ([bb:ReportAndPublicationOfStudy](#))
- Study Exploration ([bb:StudyExploration](#))

Frage 8.6/2: What are major IT evaluation methods?

- Delphi Survey ([bb:DelphiSurvey](#))
- Qualitative Evaluation Method ([bb:QualitativeEvaluationMethod](#))
- Quantitative Evaluation Method ([bb:QuantitativeEvaluationMethod](#))
- SWOT Analysis ([bb:SwotAnalysis](#))
- Usability Study ([bb:UsabilityStudy](#))

Frage 9/3: What are the tasks and methods for strategic HIS planning?

- Long-Term HIS Planning (`bb:LongTermHISPlanning`)
- Portfolio Management (`bb:PortfolioManagement`)
- Short-Term HIS Planning (`bb:ShortTermHISPlanning`)

Frage 9/4: What are the tasks and methods for strategic HIS monitoring?

- Ad Hoc Monitoring (`bb:AdHocMonitoring`)
- HIS Certification (`bb:HisCertification`)
- Permanent Monitoring (`bb:PermanentMonitoring`)
- Strategic Information Management (`bb:StrategicInformationManagement`)

Frage 9/5: What are the tasks and methods for strategic HIS directing?

- Adoption of Project Result (`bb:AdoptionOfProjectResult`)
- Monitoring the Project Progress (`bb:MonitoringTheProjectProgress`)
- Project Initiation (`bb:ProjectInitiation`)
- Project Resource Allocation (`bb:ProjectResourceAllocation`)
- Project Time Allocation (`bb:ProjectTimeAllocation`)
- Strategic Information Management (`bb:StrategicInformationManagement`)

Frage 9.2/2: What are the three main scopes of information management?

- IT Service Management (`bb:ItServiceManagement`)
- Modeling Information Systems (`bb:ModelingInformationSystems`)

- Operational Information Management (`bb:OperationalInformationManagement`)
- Strategic Information Management (`bb:StrategicInformationManagement`)
- Tactical Information Management (`bb:TacticalInformationManagement`)

Frage 9.3.4.2/1: Which organizational units are involved in information management?

- Hospital Administration (`bb:HospitalAdministration`)

Frage 9.3.4.2/2: Which boards and persons are involved in information management?

- Chief Information Officer (`bb:ChiefInformationOfficer`)

Frage 9.4/2: What are the typical tasks for strategic HIS planning?

- HIS Budget Planning (`bb:HisBudgetPlanning`)
- IT Investment Planning (`bb:ItInvestmentPlanning`)
- Long-Term HIS Planning (`bb:LongTermHISPlanning`)
- Short-Term HIS Planning (`bb:ShortTermHISPlanning`)
- Strategic Alignment (`bb:StrategicAlignment`)

Frage 9.5/1: What are the typical tasks of strategic HIS monitoring?

- Ad Hoc Monitoring (`bb:AdHocMonitoring`)
- Continuous HIS Auditing (`bb:ContinuousHISAuditing`)
- Continuous Quality Improvement Process
(`bb:ContinuousQualityImprovementProcess`)

- IT Investment Justification (`bb:ItInvestmentJustification`)
- Report to the Hospital's Board of Directors
(`bb:ReportToTheHospitalsBoardOfDirectors`)
- Reporting to the CEO (`bb:ReportingToTheCEO`)
- Structural Quality Assessment (`bb:StructuralQualityAssessment`)

Frage 9.5/2: What are the typical methods of strategic HIS monitoring?

- Functional Redundancy Rate (`bb:FunctionalRedundancyRate`)
- HIS Quality (`bb:HisQuality`)
- Key Performance Indicator (`bb:KeyPerformanceIndicator`)
- Strategic Information Management Plan (`bb:StrategicInformationManagementPlan`)

Frage 9.6/1: What are the typical tasks of strategic HIS directing?

- Adoption of Project Result (`bb:AdoptionOfProjectResult`)
- Monitoring the Project Progress (`bb:MonitoringTheProjectProgress`)
- Project Initiation (`bb:ProjectInitiation`)
- Project Resource Allocation (`bb:ProjectResourceAllocation`)
- Project Time Allocation (`bb:ProjectTimeAllocation`)
- Strategic Information Management (`bb:StrategicInformationManagement`)

Frage 9.6/2: What are the typical methods of strategic HIS directing?

- HIS Architecture (`bb:HisArchitecture`)

- Information Management Organizational Structure
(`bb:InformationManagementOrganizationalStructure`)
- Migration Path (`bb:MigrationPath`)
- Project Portfolio (`bb:ProjectPortfolio`)
- Strategic HIS Monitoring Result (`bb:StrategicHISMonitoringResult`)
- Strategic Information Management Plan
(`bb:StrategicInformationManagementPlan`)

Frage 10/1: What are health care networks?

- Health Care Network (`bb:HealthCareNetwork`)

Frage 10/2: How can health care networks be described?

- Health Care Network (`bb:HealthCareNetwork`)

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.

Mit der schulinternen Verwendung der Arbeit bin ich einverstanden.

Leipzig, 20. Januar 2023

Hannes Raphael Brunsch