

Churer Schriften zur Informationswissenschaft

Herausgegeben von
Wolfgang Semar

Arbeitsbereich
Informationswissenschaft

Schrift 133

Chatbots und Semantic-Web – Ein «Dream-Team»?

Einsatz semantischer Technologien in der Chatbot-Entwicklung
und Anwendung im Bibliotheksbereich

Vanessa Seyffert

Chur 2021

Churer Schriften zur Informationswissenschaft

Herausgegeben von Wolfgang Semar

Schrift 133

Chatbots und Semantic-Web

– Ein «Dream-Team»?

Einsatz semantischer Technologien in der Chatbot-Entwicklung und Anwendung im Bibliotheksbereich

Vanessa Seyffert

Diese Publikation entstand im Rahmen einer Thesis zum Bachelor of Science FHGR in Information Science.

Referent: Dr. Heiko Rölke

Korreferentin: Dr. Ana Petrus

Verlag: Fachhochschule Graubünden

ISSN: 1660-945X

Ort, Datum: Chur, Oktober 2021

Vorwort

Ich bedanke mich herzlich bei Marcel Hanselmann, der mich für eine Präsentation von Viola in der FHGR Bibliothek empfangen hat und mir Unterlagen zur Verfügung gestellt hat.

Abstract

Chatbots finden in vielen Bereichen Einsatz. Trotz des Hypes erbringen viele von ihnen noch keine zufriedenstellende Leistung. In der Wissenschaft wird nach Ansätzen gesucht, um Chatbots zu verbessern. Der Einsatz semantischer Technologien scheint vielversprechend: eine grosse Menge an strukturierten Daten steht im WWW für Abfragen zur Verfügung. Die Bachelorarbeit geht auf Chatbot-Projekte ein, die semantische Technologien nutzen. Ontologien können die Wissensbasis eines Chatbots erweitern und seine Sprachanalyse verbessern. Voraussetzung für einen funktionstüchtigen Chatbot ist aber die Kombination mit weitergehenden Technologien. Betrachtet wird auch die Übertragung semantischer Technologien auf den Bibliotheks-Chatbot Viola der FHGR Bibliothek. Eine Hürde beim Einsatz semantischer Technologien im Bibliotheksbereich stellt das Fehlen geeigneter Ontologien dar. Darüber hinaus sind für die selbständige Implementierung eines ausgereiften Chatbots Programmierfähigkeiten vorausgesetzt.

Hinweis zur gendergerechten Sprache:

Um non-binäre Transidentitäten einzuschliessen, wird in dieser Arbeit der Gender-Doppelpunkt (:) verwendet, sofern keine Umschreibung möglich ist.

Inhaltsverzeichnis

1	Einleitung	1
2	Einführung in die Thematik der Chatbots	3
2.1	Der Hype und die Realität	3
2.2	Klassische Umsetzung von Chatbots	4
2.2.1	Artificial Intelligence Markup Language (AIML)	7
2.2.2	Pattern-Matching	8
2.2.3	Parsing	8
2.3	Weiterführende Design-Techniken für Chatbots	9
2.3.1	ChatScript	9
2.3.2	SQL und relationale Datenbanken	10
2.3.3	«Sprachtricks»	10
2.3.4	Markov-Kettenmodell	10
2.3.5	Rekurrente neuronale Netze (RNN)	11
2.3.6	Long short-term memory (LSTM) networks	11
2.3.7	Sequence-to-Sequence-Modell für neuronale Netze	12
2.4	Kategorisierung von Chatbots	12
2.4.1	Verwendungszweck	13
2.4.2	Wissensdomäne	13
2.4.3	Art der angebotenen Dienstleistung	14
2.4.4	Methoden der Inputverarbeitung und Antwortgenerierung (Funktionsweise)	14
2.4.5	Gesprächsdauer	15
2.5	Geschichte der Chatbot-Entwicklung	15
3	Semantic-Web und seine grundlegenden Technologien	17
3.1	eXtensible Markup Language (XML)	18
3.2	Ontologien	18
3.3	Resource Description Framework (RDF)	19
3.4	Web Ontology Language (OWL)	20
3.5	SPARQL Protocol and RDF Query Language	20
4	Herleitung der Forschungsfragen	23
4.1	Einsatz semantischer Technologien in der Chatbot-Entwicklung	23
4.2	Chatbots in Bibliotheken	24
5	Methodische Vorgehensweise	25
6	Darstellung der Ergebnisse	29
6.1	Chatbots mit semantischen Technologien	29
6.1.1	KBot (2020)	32

6.1.2	R2D2 (2020)	36
6.1.3	Ontologiebasierter Chatbot für Modemarken in Pakistan (2019)	37
6.1.4	DBpedia Chatbot (2018)	39
6.1.5	Ontologiebasierter Chatbot für E-Commerce (2018).....	41
6.1.6	OntBot (2011)	42
6.1.7	Project CyN (2004)	43
6.1.8	CyD-Technologie in MAGA (2014) und FRASI (2012)	43
6.1.9	Program W in EHeBby (2008).....	46
6.2	Vorteile von semantischen Technologien in Chatbots	47
6.3	Semantische Technologien für den Bibliotheks-Chatbot Viola.....	52
6.3.1	Der Chatbot «Viola» der FHGR Bibliothek.....	52
6.3.2	Mögliche Übertragung von Semantic-Web-Anwendungen auf Viola.....	53
7	Diskussion der Ergebnisse	57
7.1	Chatbots mit semantischen Technologien im Überblick.....	57
7.2	Erzielte Verbesserungen durch semantische Technologien	60
7.3	Abwägungen für die zukünftige Entwicklung von Viola	61
8	Fazit.....	63
9	Quellenverzeichnis	65

Abbildungsverzeichnis

Abbildung 1: Allgemeine Struktur eines Chatbots, eigene Darstellung nach Ramesh et al. (2017).	5
Abbildung 2: Vereinfachte Darstellung des Chatbot-Aufbaus nach Stucki et al. (2020, S. 8).....	6
Abbildung 3: Der RDF-Graph beschreibt die Beziehung zwischen der Bachelorarbeit und der Autorin (eigene Darstellung).....	19
Abbildung 4: Wissensgraph für die extrahierte Entität, gefragt wurde nach Alan Turing (Ait-Mlouk & Jiang, 2020, S. 149224).....	33
Abbildung 5: Beispiel einer generierten Antwort in KBot (Ait-Mlouk & Jiang, 2020, S. 149226).....	35
Abbildung 6: Ansicht der von Nazir et al. gebildeten Klassen (Nazir et al., 2019, S. 548).	38

Tabellenverzeichnis

Tabelle 1: Beispiel eines Wortfeldes auf Deutsch für die Forschungsfrage 1.....	26
Tabelle 2: Zusammenfassung der Chatbot-Projekte, welche semantische Technologien einsetzen.	31
Tabelle 3: Eigene Zusammenfassung der Ergebnisse aus der Studie von Casas et al. (Casas et al., 2020, S. 281–282).....	49

Abkürzungsverzeichnis

AIML	Artificial Intelligence Markup Language
ALICE	Artificial Linguistic Internet Computer Entity
CUI	Conversational User Interface
FHGR	Fachhochschule Graubünden
GND	Gemeinsame Normdatei
GUI	Graphical User Interface
HTW	Hochschule für Technik und Wirtschaft Chur
JSON	JavaScript Object Notation
KI	Künstliche Intelligenz
LINDAS	Linked Data Service
LOV	Linked Open Data Vocabulary
LSA	Latent Semantic Analysis
LSTM	Long Short-Term Memory
NER	Named Entity Recognition
NLP	Natural Language Processing
NLU	Natural Language Understanding
RDQL	RDF Data Query Language
REGEX	Regular Expression
RNN	Rekurrente neuronale Netze
SPARQL	SPARQL Protocol and RDF Query Language
SQL	Structured Query Language
SVM	Support Vector Machine
URI	Unique Resource Identifier
VIAF	Virtual International Authority File
VIOLA	Virtuelle Online-Assistentin
W3C	World Wide Web Consortium
WWW	World Wide Web
XML	eXtensible Markup Language

1 Einleitung

Ein Chatbot ist ein Computersystem, welches das Gesprächsverhalten des Menschen für einen bestimmten Zweck imitiert (Bagchi, 2020). Als Schnittstelle zwischen einem Mensch und einer Softwareanwendung nutzen Chatbots natürliche Sprache in gesprochener oder geschriebener Form (Galitsky, 2019, S. 13–14). Gleichwohl Chatbots in den letzten Jahren deutlich spürbar in verschiedenen Bereichen implementiert wurden, hält sich deren Qualität oft in Grenzen. Nicht hilfreiche Antworten von Chatbots im Bereich des Kundendienstes führen bei Nutzenden zu Frust, was der ursprünglichen Intention von Chatbots entgegen strebt (Galitsky, 2019, S. 22; Kondylakis et al., 2020, S. 2; Märki, 2020).

Verbesserungsmöglichkeiten für Chatbots werden rege gesucht (Ait-Mlouk & Jiang, 2020; Nazir et al., 2019). Gleichzeitig steht durch den Fortschritt des Semantic-Webs eine grosse Menge an annotierten Daten im Internet zur Verfügung. Können diese Daten das Wissen von Chatbots füttern? Die Literatur weist darauf hin, dass semantische Technologien für die Erstellung von Chatbots eingesetzt werden (Ait-Mlouk & Jiang, 2020; Augello et al., 2014; Nazir et al., 2019). Die vorliegende Bachelorarbeit geht der Forschungsfrage nach, wie Semantic-Web Anwendungen für Chatbots eingesetzt werden und ob diese tatsächlich einen Mehrwert bringen. Um den Praxisbezug herzustellen, wird darüber hinaus untersucht, inwiefern semantische Technologien die Umsetzung eines Bibliotheks-Chatbots unterstützen. Diese Frage wird anhand des Chatbots *Viola* der Bibliothek der Fachhochschule Graubünden (FHGR) beantwortet.

Nach dieser Einleitung folgt im Kapitel 2 eine Einführung in die Thematik der Chatbots. Die Einführung umfasst die Definition von Chatbots, betrachtet ihren fortschreitenden Einsatz in den letzten Jahren (2.1) und gibt Auskunft über deren klassischen und weiterführenden Design-Techniken (2.2 und 2.3). Darüber hinaus wird auf die Kategorisierung (2.4) und die Geschichte der Chatbots (2.5) eingegangen. Kapitel 3 geht auf das Semantic-Web und seine grundlegenden Technologien ein. Die oben erwähnten Forschungsfragen werden in Kapitel 4 hergeleitet. Für die methodische Vorgehensweise ist Kapitel 5 zu konsultieren. Schliesslich werden in Kapitel 6 die Ergebnisse dargestellt. Eine Auflistung von Chatbots mit semantischen Technologien wird aus der wissenschaftlichen Literatur abgeleitet und findet sich in 6.1. Um den Mehrwert von semantischen Technologien in Chatbots zu eruieren, werden aus einer Literatur-Analyse die häufigsten Mängel von Chatbots übernommen und den dokumentierten Verbesserungen von semantischen Technologien gegenübergestellt (6.2). Im Anschluss werden Zweck und Technologie des Chatbots *Viola* kurz eingeordnet (6.3.1). Aufgrund

der vorhergehenden Erkenntnisse wird geprüft, ob semantische Technologien für Viola in Frage kommen (6.3.2). Bevor das Fazit (8) die Arbeit abschliesst, werden die Ergebnisse aus der Untersuchung in Kapitel 7 diskutiert.

2 Einführung in die Thematik der Chatbots

Wie bereits in der Einleitung erwähnt, ahmen Chatbots die menschliche Kommunikation nach. Darüber hinaus haben sie laut Abdul-Kader & Woods (2015, S. 73, zitiert nach Galvao et al., 2004) die Fähigkeit, das Verhalten der Nutzenden zu untersuchen und zu beeinflussen. Hierfür nutzen Chatbots Fragen und Antworten an die Nutzenden. Um einem Chatbot eine Frage stellen zu können, verwenden Nutzende natürliche Sprache in geschriebener oder gesprochener Form. Das Computersystem sucht daraufhin die Antwort, welche am besten für das angesprochene Themengebiet geeignet ist und gibt sie aus (Abdul-Kader & Woods, 2015, S. 73). Da Nutzende aufgrund des Einsatzes natürlicher Sprache nicht in der Anwendung des Chatbots geschult werden müssen, findet eine Verlagerung von grafischen Benutzungsschnittstellen («Graphical User Interfaces» GUI) zu dialogorientierten Benutzungsschnittstellen («Conversational User Interfaces» CUI) statt. Die Mensch-Computer-Interaktion wird dadurch vereinfacht und auch einige der zehn Usability-Heuristiken von Nielsen (2020) werden dadurch umgesetzt (Porreca et al., 2018, S. 158).

Synonyme für Chatbots sind unter anderem Konversationsagenten, Dialogsysteme, Smart Bots, interaktive Agenten, virtuelle oder digitale Assistenten, Frage-/Antwort-Systeme sowie künstliche Gesprächseinheiten (Adamopoulou & Moussiades, 2020; Augello et al., 2014; Galitsky, 2019). In der vorliegenden Arbeit wird vornehmlich der Begriff «Chatbot» genutzt. Personen, welche den Chatbot nutzen, um Information abzufragen, werden «Nutzende» genannt.

2.1 Der Hype und die Realität

Chatbots finden weit verbreiteten Einsatz (Adamopoulou & Moussiades, 2020, S. 373). Gemäss Dale waren sie im Jahr 2016 die meist gehypte Sprachtechnologie des Jahres. Sehr sichtbar sind die digitalen Assistenten der «Big Four»¹, die mithilfe stimmenbasierter Technologien nutzbar sind. Daneben gibt es eine Vielzahl textbasierter Chatbots (Dale, 2016, S. 811). Textbasierte Chatbots finden unter anderem in weit verbreiteten Messaging-Plattformen Anwendung, wie zum Beispiel in Facebook Messenger oder WhatsApp. Auch im Kundendienst, in der Sprachlehre oder sogar als erste Anlaufstelle bei häuslicher Gewalt werden sie eingesetzt (vgl. dazu die Beiträge von Kona Connect, 2020; Lee et al., 2020; Stucki et al., 2020). Das Coaching ist ein weiterer Bereich, der

¹ Gemeint sind Apples Siri, Microsofts Cortana, Amazons Alexa und Googles Assistant (vgl. Dale, 2016, S. 811).

stark von den Fortschritten in der Chatbot-Technologie profitiert. Aufkommende Anwendungen werden beispielsweise bei E-Coaches für das Wohlbefinden älterer Menschen oder für Ernährungstagebücher beobachtet (Casas et al., 2020, S. 280). Das Wall Street Journal warf kürzlich die Frage auf, ob Chatbots in Zukunft allenfalls menschliche Therapeuten ersetzen könnten (Ward, 2021).

Galitsky (2019, S. 1) geht davon aus, dass Suchmaschinen zwar noch lange Zeit im Einsatz sein werden, dass aber in Zukunft mehr Informationen über textbasierte Messenger-Dienste oder über sprachgesteuerte Systeme weitergegeben werden. Gartner Group (2019) schätzt, dass bis 2022 70% der Angestellten täglich mit Konversationsplattformen interagieren werden. Gleichzeitig ist der Markt mit über 2000 Anbietern von Software-Lösungen für Chatbots überfüllt. Viele von ihnen stellen ungenügende Dienstleistungen zur Verfügung in den Bereichen Bereitstellung und Wartung von Chatbots (Gartner Group, 2019).

Fachpersonen stellen fest, dass trotz des breitflächigen Einsatzes von Chatbots in Firmen die Funktionalitäten oft noch unausgereift sind (Kondylakis et al., 2020, S. 2). Der Chatbot der Informatikdienste der ETH kann zum Beispiel in 50% der Fragen weiterhelfen, während er bei 50% der Fragen den Kern des Anliegens nicht zu treffen vermag (Märki, 2020). Galitsky (2019) beschreibt, dass über Chatbots mehr diskutiert werde, als dass sie tatsächlich genutzt würden und merkt an: «Most articles about task-oriented bots capable of conversing sound more like a fake news once one starts a conversation with them» (Galitsky, 2019, S. 22).

Tatsache aber ist, dass immer mehr Firmen und Institutionen die Anschaffung eines Chatbots für Kundendienstleistungen prüfen oder schon seit längerem vorantreiben (Casas et al., 2020, S. 280). Die Schweizerische Post ist als eines von vielen Beispielen zu nennen. Sie entwickelt und betreibt seit einigen Jahren mehrere Chatbots (für eine detaillierte Auflistung der Chatbots siehe Stucki et al., 2020). Sehr verlockend ist das Argument, dass Chatbots in der Lage sind eine grosse Anzahl an Nutzenden gleichzeitig zu betreuen, während menschliche Mitarbeitende nicht immer zur Verfügung stehen. Manche Firmen versprechen sich dadurch Einsparungen im Kundendienst (Valtolina et al., 2018, S. 64).

2.2 Klassische Umsetzung von Chatbots

Die klassische Umsetzung eines Chatbots basiert hauptsächlich auf der Modellierung von Gesprächen. Als Basistechnologie wird in der Regel die Artificial Intelligence Markup Language (AIML) genutzt. Es handelt sich dabei um eine Auszeichnungssprache, die auf

XML basiert und in den späten 1990er Jahren entwickelt wurde (AIML Foundation, 2018). Weitere Details zu AIML folgen im Abschnitt *Artificial Intelligence Markup Language (AIML)*. Die von den Nutzenden an den Chatbot geäußerten Absichten («User-Intents») werden identifiziert mittels Methoden des Natural-Language-Processing (NLP), um sie dann mit den Inhalten aus der Datenbasis des Chatbots abzugleichen (Stucki et al., 2020). Dazu nutzen Chatbots sogenanntes Pattern-Matching, bei dem es sich um Vergleichsregeln handelt (Augello et al., 2017). Für den eingegebenen Text sucht der Chatbot demnach ein übereinstimmendes Muster in der eigenen Wissensbasis. Wurde die geäußerte Absicht entsprechend klassifiziert und eine passende Entität bestimmt, kann eine Antwort-Ausgabe an den Nutzenden erfolgen (Adamopoulou & Moussiades, 2020; Bruno Marietto et al., 2013; Stucki et al., 2020).

Damit ein modularer Entwicklungsansatz verfolgt werden kann, sind beim Entwurf eines Chatbot-Systems seine Bestandteile zu berücksichtigen (Ramesh et al., 2017, S. 339). Stoner et al. (2003) beschreiben die Komponenten eines Chatbots in drei Teilen: *Responder*, *Classifier* und *Graphmaster*. Abbildung 1 zeigt das Zusammenspiel dieser drei Bestandteile.

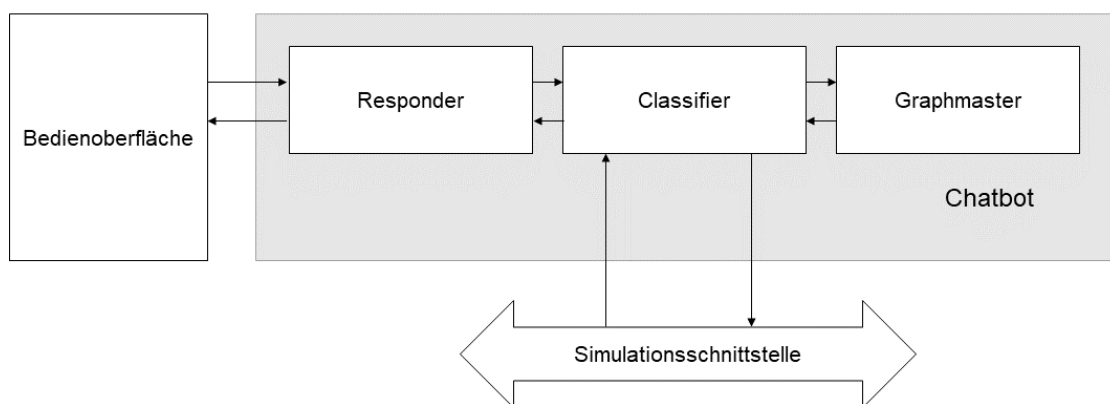


Abbildung 1: Allgemeine Struktur eines Chatbots, eigene Darstellung nach Ramesh et al. (2017).

- Der Responder ist die Schnittstelle zwischen den Nutzenden und den Hauptroutinen des Chatbots. Seine Aufgaben umfassen das Transferieren von Daten zwischen den Nutzenden und dem Classifier sowie die Kontrolle der Ein- und Ausgabe.
- Der Classifier ist zwischen dem Responder und dem Graphmaster. Er filtert und normalisiert die Eingabe, teilt diese in logische Komponenten auf und übergibt den normalisierten Satz an den Graphmaster. Darüber hinaus bearbeitet er die

Ausgabe des Graphmasters und behandelt die Anweisungen der Datenbank-Syntax (zum Beispiel AIML).

- Der Graphmaster ist Teil des Musterabgleichs (Pattern-Matching) und organisiert den Inhalt des «Gehirns» des Chatbots. Er speichert und hält die Pattern-Matching-Algorithmen bereit (Stoner et al., 2003).

Eine neuere Beschreibung von Stucki et al. (2020) weist dem Chatbot die Komponenten *Eingabeverarbeitung*, *Dialogsteuerung*, *Quellsysteme* und *Ausgabeerstellung* zu. Illustriert wird dies in Abbildung 2.

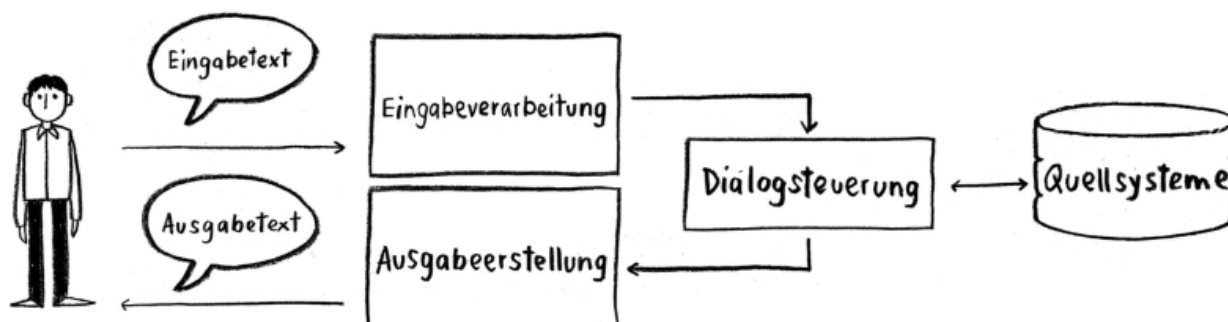


Abbildung 2: Vereinfachte Darstellung des Chatbot-Aufbaus nach Stucki et al. (2020, S. 8)

- In der Eingabeverarbeitung wird der Input der Nutzenden in eine maschinenlesbare Form überführt. Dazu werden Techniken des Natural-Language-Understanding (NLU), einer Methode des NLP, verwendet. Im Anschluss wird die identifizierte Absicht hier klassifiziert und an die Dialogsteuerung übergeben.
- In der Dialogsteuerung wird bestimmt, wie mit der Absicht der Nutzenden weitergearbeitet wird. Mögliche Szenarien sind der Zugriff auf Information aus einer Wissensdatenbank im Backend (Quellsystem), das Auslösen einer Aktion in einem anderen System oder die Verwaltung des Dialogzustands (Merken der Eingabe der Nutzenden und Analyse, ob weitere Information zu erfragen ist).
- Sobald in der Dialogsteuerung die Information für eine passende Antwort gesammelt wurde, werden die Daten an die Answererzeugung weitergeleitet, wo die Ausgabe an die Nutzenden erstellt wird (Stucki et al., 2020, S. 21–23).

Die beiden Modelle von Stucki et al. (2020) und Stoner et al. (2003) beschreiben im Grunde die gleichen Funktionen, sie werden nur anders benannten Komponenten zugewiesen. Die Grenzen zwischen den einzelnen Komponenten verschwimmen beim Vergleich zwischen den beiden Ansätzen. Als Beispiel zu erwähnen ist der Classifier, der bei Stoner et al. (2003) eine eigene Instanz darstellt, während der Classifier bei Stucki et al. (2020) in der Eingabeverarbeitung zu verorten ist. Beide Modelle werden in der

heutigen Chatbot-Entwicklung berücksichtigt, wobei die Darstellung von Stucki et al. (2020) den Mehrwert bietet, dass ein Quellsystem explizit erwähnt wird.

Nachfolgend wird auf einzelne Techniken der klassischen Chatbot-Entwicklung vertiefter eingegangen.

2.2.1 Artificial Intelligence Markup Language (AIML)

AIML wurde zwischen den Jahren 1995 und 2000 entwickelt. Ziel war es eine Auszeichnungssprache für die Erstellung von Chatbots zu entwickeln. Es handelt sich bei AIML um einen XML-Dialekt, der von Maschine wie Mensch leicht verstanden werden kann. Der erste Chatbot, der mit AIML entwickelt wurde und den AIML-Interpreter nutzte, war ALICE. Viele weitere Chatbots wurden auf Grundlage von ALICE weiterentwickelt.

AIML ist Tag-basiert. Die allgemeine Struktur eines AIML-Objekts sieht wie folgt aus:

```
<tagName><ListOfParameters></tagName>
```

<tagName> gibt dabei an, was die <ListOfParameters> entspricht (Ramesh et al., 2017, S. 341). Neben einem öffnenden und einem schliessenden AIML-Tag (<aiml>) am Anfang und Ende des Skripts, gibt es in AIML drei grundlegende Tags (Ramesh et al., 2017, S. 338). Sie werden anhand des folgenden Beispiels veranschaulicht:

```
<category>
  <pattern>MEIN NAME IST *</pattern>
  <template>Hallo <star index="1"/>, wie geht's?</template>
</category>
```

Der Tag <category> umschliesst das Frage-Antwort-Modul. Jede <category> enthält ein <pattern>, das wiederum den Satz enthält, der mit dem Input der Nutzenden verglichen wird. Das <template> enthält die Regeln, die für die Generierung des Antwortsatzes genutzt werden. Darüber hinaus bietet AIML weitere Tags und Zeichen. Um nur einige zu nennen: Platzhaltersymbole («*», «_», «^», «#») werden eingesetzt, damit der Chatbot auch dann antworten kann, wenn kein exaktes Muster in der Wissensdatenbank vorhanden ist. Für Rekursion und das Aufrufen anderer Kategorien kann mit <srai> gearbeitet werden. Mit <topic> kann Kontext für spätere Anwendungsfälle gespeichert werden. Für mehr Kontext gibt es darüber hinaus <that>, das es dem Chatbot ermöglicht, sich an die vergangene Antwort zu erinnern und die Unterhaltung weiterzuführen (Abdul-Kader & Woods, 2015, S. 74; Augello et al., 2014, S. 287).

Um den Input zu verarbeiten, führt die AIML-Engine eine Deperiodisierung und Normalisierung durch. Zuerst wird die Eingabe in mehrere Query-Strings zerteilt mittels der Punktzeichen. Die Normalisierung beinhaltet die Umwandlung in Grossbuchstaben (um den Overhead für die Musterübereinstimmung zu verringern), das Entfernen von Satzzeichen sowie das Ersetzen von Kurzformen durch ihre Erweiterungen. Der AIML-Graphmaster entspricht einer Trie-Datenstruktur.² Die Sammlung der Knoten wird als Nodemapper bezeichnet. Bei der Suche enthält der letzte Knoten oder das erreichte Knotenblatt das Template-Tag. Die Anzahl Knotenblätter entspricht der Anzahl Kategorien `<categories>` (Ramesh et al., 2017, S. 341).

2.2.2 Pattern-Matching

Pattern-Matching (Mustervergleichsregeln) ist auf irgendeine Art in jedem Chatbot zu finden, allerdings unterscheidet sich die Komplexität der Algorithmen. Relativ einfache Mustervergleichsregeln wurden im Chatbot ELIZA eingesetzt. Die Struktur eines eingegebenen Satzes wird ermittelt, um darauf eine vorher definierte Antwort auszugeben. Ramesh et al. (2017, S. 339) führen mehrere Nachteile dieses Ansatzes auf: die Antworten sind wiederkehrend und vorhersehbar. Auch mangelt es den Chatbots oft an einer menschlichen Note. Da auch keine vergangenen Antworten der Konversation gespeichert werden, kann es zu Gesprächsschleifen kommen. Darüber hinaus können die immer komplexeren Pattern-Matching-Algorithmen zu begrenzten Antworten führen, was die Unterhaltung uninteressant machen kann (Ramesh et al., 2017, S. 339). Der Chatbot ist nur so gut wie seine Wissensbasis. Diejenige Person, die den Chatbot entwickelt, baut in der Praxis oft auch diese Wissensbasis (Abdul-Kader & Woods, 2015, S. 73).

2.2.3 Parsing

Beim Parsing wird die Texteingabe analysiert und mittels einer Reihe von NLP-Funktionen behandelt (Abdul-Kader & Woods, 2015, S. 73–74). Der Text wird in ein Set von Strings konvertiert. Diese können auf einfachere Weise gespeichert und verarbeitet werden. Lexikalisches Parsing wird in der Regel verwendet, um die grammatikalische Struktur eines Satzes zu definieren. Anschliessend wird ein Syntaxbaum erstellt und eine lexikalische Struktur bestätigt. Voraussetzung ist, dass die Eingabe ein Satz im Sinne der

² Ein Trie oder auch Präfixbaum ist eine Datenstruktur, die für das Information Retrieval eingesetzt wird. Jeder Knoten des Trie besteht aus mehreren Zweigen. Jeder Zweig repräsentiert ein mögliches Zeichen der Schlüssel (A. B. Butterfield et al., 2016; GeeksforGeeks, 2021).

sprachlichen Regeln bildet (syntaktisches Parsing). Einfache Parser überprüfen, ob die erkennbaren Wörter in der erlaubten Reihenfolge erscheinen. Sie kamen in frühen Chatbots zum Einsatz. Komplexere Parser übernehmen ein komplettes grammatikalisches Parsing der Eingabe, was mithilfe NLP-Techniken ermöglicht wird. Die Schritte des NLP umfassen Tokenisierung, syntaktische, semantische und schliesslich pragmatische Analyse der Eingabe. Durch die syntaktische Analyse erhält jeder Eingabesatz eine Ordnung und Struktur. Mittels semantischer Analyse wird die wörtliche Bedeutung ermittelt und durch die pragmatische Analyse die kontextuelle Bedeutung gefunden (Ramesh et al., 2017, S. 340).

2.3 Weiterführende Design-Techniken für Chatbots

Neben den klassischen Umsetzungstechniken existiert eine Vielzahl an neueren Technologien, die breiten Einsatz in der Chatbot-Entwicklung finden. Die folgenden Abschnitte zeigen auf, welche dieser Techniken in der wissenschaftlichen Literatur diskutiert werden. Die Auflistung ist nicht abschliessend zu betrachten. Auf den Einsatz semantischer Technologien wird im Anschluss im Abschnitt *Einsatz semantischer Technologien in der Chatbot-Entwicklung* eingegangen.

2.3.1 ChatScript

ChatScript strebt danach, die Nachfolge von AIML anzutreten. Im Gegensatz zu AIML, das gemäss Ramesh et al. (2017, S. 342) schwache Pattern-Matching-Algorithmen aufweist, besteht ChatScript aus Regeln, die mit Themen verknüpft sind. In ChatScript wird zuerst das am besten geeignete Thema für die entsprechende Eingabe identifiziert. Anschliessend wird eine Regel innerhalb dieses Themas ausgeführt (Ramesh et al., 2017, S. 342). Unter den Gewinner:innen des Loebner-Preises (mehr zum Loebner-Preis in *Geschichte der Chatbot-Entwicklung*) befinden sich vier Chatbots, die auf ChatScript basieren. Der Chatbot Suzette von Brian Wilcox war der erste Chatbot, der mit ChatScript erstellt wurde (Wilcox, 2019).

Sogenannte «concepts» in ChatScript bilden ein Set von ähnlichen Wörtern. Für die Zuweisung werden die Bedeutungen der Wörter oder andere Eigenschaften wie Wortarten miteinbezogen. Auf diese Weise ist es möglich, ein Konzept aller Nomen oder Adverbien zu erstellen. Bereits existieren Datenbanken von «concepts», die modular sind und wiederverwendet werden können. Darüber hinaus gibt es in ChatScript auch Variablen für die Langzeit-Erinnerung. Sie werden genutzt, um bestimmte Information der Nutzenden (zum Beispiel Namen oder Alter) zu speichern, um später im Verlauf des

Gesprächs wieder darauf zurückzugreifen. Dies ist auch hilfreich, wenn überprüft werden soll, ob eine bestimmte Variable schon besetzt ist (zum Beispiel: ob die Reisedestination im Reise-Buchungssystem schon genannt wurde). Anders als AIML berücksichtigt ChatScript die Gross-/Kleinschreibung, was dazu führt, dass auch durch Grossbuchstaben betonte Wörter erkannt werden (Ramesh et al., 2017, S. 342).

2.3.2 SQL und relationale Datenbanken

Laut Abdul-Kader & Woods (2015, S. 73–74) wurden SQL (Structured Query Language) und relationale Datenbanken im Jahr 2015 «recently» (seit kurzem) in der Chatbot-Entwicklung eingesetzt. Konkret genutzt werden relationale Datenbanken, um die Erinnerung an geführte Konversationen zu unterstützen, indem der Chatbot auf die gespeicherte Historie in der mit SQL erstellten Datenbank zurückgreift. Dies führt dazu, dass die Unterhaltungen weiter führen und gehaltvoller werden. SQL ist die am häufigsten genutzte Datenbanksprache. Ebenfalls sehr bekannt ist MySQL für relationale Datenbanken. Abdul-Kader & Woods (2015) zeigen auf, dass der Gebrauch von SQL in der Chatbot-Entwicklung neue Wege aufgezeigt hat, weisen aber gleichzeitig darauf hin, dass es keinen Nachweis gibt, dass SQL bei Loebner-prämierten Chatbots eingesetzt wurde (Abdul-Kader & Woods, 2015, S. 73–75, 79).

2.3.3 «Sprachtricks»

Mit Sprachtricks sind in Chatbots verfügbare Sätze, Phrasen oder Abschnitte gemeint, welche Vielfalt in die Wissensbasis bringen. Ziel dieser Tricks ist es, den Chatbot überzeugender zu machen. Als Beispiele genannt werden unter anderem das Vorspielen einer persönlichen Geschichte, Tippfehler oder das Vortäuschen von Tastatur-Tippen bei der Answerzeugung (Abdul-Kader & Woods, 2015, S. 73).

2.3.4 Markov-Kettenmodell

Das Markov-Kettenmodell ist ein probabilistisches Modell, das auf der sogenannten Markov Property basiert. Es besagt, dass bei einem gegebenen Index i im Zustandsraum (zum Beispiel zum Zeitpunkt t) die Entscheidung nur vom letzten Zustand vor i (Zeitpunkt $t-1$) abhängt. Die Idee dahinter ist, dass es für jeden Buchstaben oder jedes Wort in einem Textdatensatz eine feste Eintrittswahrscheinlichkeit gibt. Durch die Markov-Kette wird versucht, die Wahrscheinlichkeit von Zustandsübergängen im Laufe der Zeit zu modellieren. Ziel ist es, probabilistisch geeignetere und folglich korrektere Antworten des Chatbots zu finden. Ein Vorteil von Markov-Modellen ist ihre einfache Anwendbarkeit. Sie

ermöglichen es What-if-Queries einfach zu modellieren und Änderungen im System im Laufe der Zeit zu beobachten. Allerdings sind Markov-Modelle vereinfachte Versionen von komplexen Entscheidungsprozessen. Sie lassen nur bedingt die Erstellung reichhaltiger oder komplexer Konversationen zu (Abdul-Kader & Woods, 2015, S. 73; Ramesh et al., 2017, S. 340).

2.3.5 Rekurrente neuronale Netze (RNN)

Unser Verstand nutzt bei der Entscheidungsfindung den Kontext sowie das Verstandene aus dem bisherigen Gespräch, um Schlüsse zu ziehen. Herkömmliche neuronale Netze haben diese Fähigkeit nicht. So ist es nicht möglich, den bisherigen Gesprächsverlauf zu analysieren. Rekurrente neuronale Netze (RNN) hingegen haben die Fähigkeit Information persistent zu halten (Ramesh et al., 2017, S. 342). Sie zielen darauf ab, vorherige Informationen mit der vorliegenden Aufgabe zu verknüpfen. So wird eine Ausgabe basierend auf allen vorherigen Schritten/Informationen und der aktuellen Eingabe abgeleitet. In den letzten Jahren wurden RNN erfolgreich in verschiedenen Bereichen wie Bildunterschrift, Sprachmodellierung, Übersetzung oder Spracherkennung eingesetzt (Ramesh et al., 2017, S. 345).

2.3.6 Long short-term memory (LSTM) networks

Long short-term memory (LSTM) networks (auf Deutsch: Netzwerke mit langem Kurzzeitgedächtnis) sind eine spezielle Art von RNN. Wenn aktuelle Information für die Bildung einer Antwort erforderlich ist, liegt die benötigte Information in der Nähe des Ortes, an dem sie gebraucht wird und RNN können gut damit umgehen. Liegt die Information weiter in der Vergangenheit, wird die Ableitung schwieriger, da die Information weiter entfernt ist. RNN sind zwar in der Lage, entfernte Information zu beziehen, allerdings ist dies in der Praxis schwierig umzusetzen. LSTM können langfristige Abhängigkeiten erlernen, indem sie das so genannte verschwindende Gradientenproblem verhindern. Dieses stellt häufig ein Hindernis beim Training neuronaler Netze dar. Das verschwindende Gradientenproblem wird an dieser Stelle nur oberflächlich erläutert: Das Gradienten-basierte Lernverfahren zielt darauf ab, den Wert eines Parameters zu erlernen, indem erfasst wird, welche Auswirkungen kleine Änderungen des Parameterwertes auf die Ausgabe des Netzes haben. Sind die Auswirkungen sehr klein, lernt das Netz die Änderung des Parameterwertes nicht richtig. Je grösser ein Netz wird, umso kleiner werden die Gradienten der Netzwerkausgabe in Bezug auf die Parameter in den frühen Schichten (für detailliertere Information zu dieser

Thematik siehe Bengio et al., 1994). LSTM lösen dieses Problem. Dadurch sind sie im Gegensatz zu herkömmlichen RNN ideal geeignet, um aus Erfahrungen zu lernen. Ramesh et al. bewerten die Fähigkeit zur Klassifizierung, Verarbeitung und Vorhersage von Zeitreihen bei LSTM höher als bei anderen RNN sowie bei Hidden-Markov-Modellen oder anderen Sequenzlernverfahren. Deshalb sind LSTM geeignet für Chatbots, die häufig auf weit entfernte Information zurückgreifen müssen (Ramesh et al., 2017, S. 345–346).

2.3.7 Sequence-to-Sequence-Modell für neuronale Netze

Dieses Modell wird auch mit «seq2seq» abgekürzt. Es wird hauptsächlich in der maschinellen Sprachübersetzung eingesetzt. Die Ausgangssequenz bildet dabei den Satz in einer bestimmten Sprache und die Zielsequenz das übersetzte Gegenstück in einer anderen Sprache. Auf Chatbots angewendet, stellt die Quellsequenz die Chatnachricht der Nutzenden dar und die Zielsequenz die Antwort des Chatbots. Bestandteile des Sequence-to-Sequence-Modells sind zwei RNN sowie ein RNN-Encoder-Decoder (Ramesh et al., 2017, S. 346).

2.4 Kategorisierung von Chatbots

Die für die Erstellung eines Chatbots gewählten Designtechniken hängen von verschiedenen Kriterien ab. Das Verständnis für verschiedene Kategorien von Chatbots hilft dabei, die passende Technologie für einen Chatbot auszuwählen (Nimavat & Champaneria, 2017, S. 1019; Ramesh et al., 2017, S. 347). In der Literatur existieren unterschiedliche Kategorisierungen von Chatbots. Stucki et al. (2020) gliedern die Chatbots in *Verwendungszweck* und *Funktionsweise*. Etwas ausführlicher ist die Klassifikation, die Adamopoulou & Moussiades (2020) zusammengetragen haben; sie umfasst die *Wissensdomäne*, die *Art der angebotenen Dienstleistung*, die *Ziele des Chatbots* sowie die *Methoden der Inputverarbeitung und Antwortgenerierung*. Ramesh et al. (2017) bringen darüber hinaus die *Dauer des Gesprächs* als Kriterium mit ein. Die nachfolgende Übersicht über die verschiedenen Kategorien stellt eine Synthese der Kategoriensysteme von Adamopoulou & Moussiades (2020), Ramesh et al. (2017) und Stucki et al. (2020) dar. Wichtig bleibt zu erwähnen, dass sich Chatbots nicht ausschliesslich einer Kategorie zuweisen lassen. Die meisten von ihnen enthalten Elemente mehrerer Kategorien in unterschiedlicher Ausprägung (Nimavat & Champaneria, 2017, S. 1020).

2.4.1 Verwendungszweck

Unter dem Verwendungszweck lassen sich zielorientierte und nicht-zielorientierte Chatbots zusammenfassen. Zielorientierte Chatbots dienen Nutzenden dazu, ein bestimmtes Anliegen zu lösen. Unterkategorien von zielorientierten Chatbots sind:

- Informationsbots: Sie geben beispielsweise Auskünfte zu Dienstleistungen oder Produkten sowie Anleitungen.
- Transaktionsbots: Sie übernehmen für die Nutzenden eine Aufgabe, wie die Buchung eines Hotelzimmers oder den Kauf von Fahrkarten.

Nicht-zielorientierte Chatbots hingegen dienen in erster Linie dazu, eine soziale Interaktion zu simulieren. Zwar erfüllen diese auch einen Zweck, es handelt sich dabei aber nicht um die Lösung eines konkreten Problems. Stucki et al. (2020, S. 6) weisen darauf hin, dass nicht-zielorientierte Chatbots im Unternehmensumfeld weniger relevant sind, soziale Aspekte bei der Entwicklung von Chatbots in Unternehmen aber auch berücksichtigt werden sollten. Beispiele für nicht-zielorientierte Chatbots sind ELIZA, PARRY und Cleverbot (Stucki et al., 2020, S. 6).

2.4.2 Wissensdomäne

Bei der Einteilung nach Wissensdomäne stellt sich die Frage, auf welches Wissen der Chatbot zugreifen können soll. In diese Typisierung fallen Open-Domain-Chatbots, also solche, die über generelle Themen sprechen können, und Closed-Domain-Chatbots, die auf ein bestimmtes Themengebiet fokussiert sind. Letztere scheitern darin, zu anderen als den vorgesehenen Themen Auskunft geben zu können (Nimavat & Champaneria, 2017, S. 1019).

Interagieren Menschen miteinander, können sie neue Themen anstossen. Das bedeutet, im Laufe eines Gesprächs kann sich die Wissensdomäne verändern. Als Beispiel für solche Konversationen können Interaktionen auf sozialen Medien wie Facebook oder Twitter herangezogen werden. Ramesh et al. (2017, S. 348) beschreiben, dass Open-Domain-Chatbots nicht für einen bestimmten Zweck konzipiert sind. Solche Konversationen zu simulieren ist schwierig, da umfangreiche Wissensbasen für die Modellierung der Antworten notwendig sind. Einfacher ist die Konfiguration von Closed-Domain-Chatbots, da begrenztes und spezifisches Wissen über ein Themengebiet erforderlich ist. Als Beispiel genannt wird ein Kundenbetreuungssystem. Der Zweck und die Wissensdomäne sind klar definiert. Die Antwort soll so schnell wie möglich generiert werden, um das Anliegen zu lösen (Ramesh et al., 2017, S. 348).

2.4.3 Art der angebotenen Dienstleistung

Werden Chatbots nach ihrer angebotenen Dienstleistung typisiert, wird die sentimentale Nähe zwischen Chatbot und Nutzenden gewichtet. Interpersonelle Chatbots können zwar gewisse Persönlichkeit aufweisen und freundlich sein, es wird aber nicht von ihnen erwartet. Intrapersonelle Chatbots sollen den Menschen hingegen wie eine Begleitperson verstehen können. Inter-Agenten-Chatbots kommunizieren mit anderen Chatbot-Systemen, um gewisse Aufgaben zu erledigen. Gemäss Nimavat & Champaneria (2017) wird diese Art von Chatbot in Internet-of-Things-Bereichen vorherrschend sein (Nimavat & Champaneria, 2017, S. 1020).

2.4.4 Methoden der Inputverarbeitung und Antwortgenerierung (Funktionsweise)

Gemäss Ramesh et al. (2017) lassen sich Chatbots in retrieval-basierte und generativ-basierte Modelle unterteilen. Retrieval-basierte Modelle greifen auf vordefinierte Antworten zurück. Um die geeignete Antwort auszuwählen, wenden sie Heuristiken an. Unterkategorien von retrieval-basierten Chatbots sind:

- Regelbasierte Chatbots: Diese können auf einfachen Konzepten beruhen. Regelbasierte Chatbots sind die am weitesten verbreiteten Chatbots.
- Chatbots mit Machine-Learning-Klassifikatoren: Diese Chatbots sind komplexer als regelbasierte und wenden eine Kombination von Machine-Learning-Techniken an.

Retrieval-basierte Modelle haben den Vorteil, dass sie weniger grammatikalische Fehler machen, da die Antworten vordefiniert sind. Nachteil allerdings ist, dass sie keine geeigneten Antworten erzeugen können, sobald sie mit unbekanntem Bedingungen konfrontiert sind. Ebenso haben sie keine Möglichkeit vorherige Teile des Gesprächs oder Kontext (zum Beispiel Orts- oder Personeninformation) zuzugreifen.

Im Gegensatz zu retrieval-basierten Modellen, generieren generative Modelle neue Antworten. Dadurch überwinden sie die Abhängigkeit von vordefinierten Antworten. Sie nutzen zu diesem Zweck eine Reihe verschiedener Techniken, um die Eingabe in eine Antwort zu übersetzen. Sie haben die Möglichkeit, auf vergangene Gesprächsinformation zuzugreifen, womit sie als «intelligenter» gelten. Nachteil des generativen Ansatzes ist die Schwierigkeit, den Chatbot zu trainieren. Für dieses Training sind enorme Mengen an Daten erforderlich. Auch sind generative Modelle anfälliger auf grammatikalische Fehler. Allerdings soll die Mensch-Computer-Interaktion durch die generative Methode menschenähnlicher werden.

Sowohl retrieval-basierte als auch generativ-basierte Modelle lassen sich durch Deep-Learning-Techniken (zum Beispiel Sequence-to-Sequence-Modelle) weiter verbessern. Gemäss Ramesh et al. (2017) geht der Trend in Richtung generative Methoden (Ramesh et al., 2017, S. 346–347).

2.4.5 Gesprächsdauer

Das Kriterium Gesprächsdauer umfasst die Kategorien Kurztext-Konversationen und Langtext-Konversationen. Bei ersteren wird eine Antwort für eine einzelne Anfrage erzeugt. Dauert ein Gespräch länger, kann es notwendig sein, auf vergangene Teile der Interaktion zurückzugreifen, um eine passende Antwort abzuleiten. Langtext-Konversationen sind deshalb schwieriger zu automatisieren (Ramesh et al., 2017, S. 347).

2.5 Geschichte der Chatbot-Entwicklung

Auf die Geschichte der Chatbots wird kurz eingegangen. Für eine ausführlichere Darstellung wird auf den Beitrag von Stucki et al. (2020) verwiesen.

Der erste Einsatz eines Chatbots geht auf die 1960er Jahre zurück. *Eliza* wurde am Massachusetts Institute of Technology (MIT) von Joseph Weizenbaum entwickelt. Sie simulierte eine Psychotherapeutin und konnte mit Menschen aufgrund handgefertigter Skripts kommunizieren. In ihr fand sich ein einfacher Pattern-Matching-Ansatz. Im Jahr 1972 folgte *Parry*, der vom Psychiater Kenneth Colby entwickelt wurde und eine Person mit paranoider Schizophrenie nachahmte. *ALICE* (Artificial Linguistic Internet Computer Entity) von Richard Wallace wurde 1995 veröffentlicht und war mit AIML erstellt worden. Sie gewann den Loebner-Preis (siehe zum Loebner-Preis den nächsten Abschnitt) in den Jahren 2000, 2001 und 2004 (Adamopoulou & Moussiades, 2020, S. 374; Ait-Mlouk & Jiang, 2020, S. 149220–149221; Athreya et al., 2018, S. 143). *Rose* gewann 2014 den Loebner-Preis. Bruce Wilcox hat sie mit ChatScript erstellt. In erster Linie nutzt sie eine Natural-Language-Engine, um die Bedeutung der Eingaben zu erkennen (Ait-Mlouk & Jiang, 2020, S. 149221). Innerhalb weniger Jahrzehnte wurden grosse Fortschritte in der Forschung erzielt. Seit dem Jahr 2010 stehen Chatbots mit künstlicher Intelligenz wie Siri, Cortana oder Google Assistant zur Verfügung (Ait-Mlouk & Jiang, 2020, S. 149220).

Turing-Test und Loebner-Preis

«Können Maschinen denken?». Alan Turing schlug mit seiner aufgeworfenen Frage ein Imitation-Game vor, um die künstliche Intelligenz einer Maschine zu spezifizieren. Die

Messmethode sieht vor, dass ein:e Interrogator:in mit einem Menschen und einer Maschine kommuniziert, ohne zu wissen, welche der beiden Gesprächspartner Mensch oder Maschine ist. Am Ende soll erraten werden, welche der beiden Parteien menschlich ist (Abdul-Kader & Woods, 2015, S. 74). In den 1990er Jahren vereinbarte Hugh Loebner mit dem Cambridge Centre for Behavioural Studies einen Wettbewerb auf Basis des Turing-Tests einzuführen. Hugh Loebner bot eine Goldmedaille von 100'000 Dollar an, für den ersten Chatbot, dessen Antworten nicht von denen eines Menschen unterschieden werden kann. Jährlich wird eine Bronze-Medaille verliehen für den menschlichsten Computer (im Vergleich zu den anderen Teilnehmenden). Zwischen 1991 und 2019 wurde die Preisvergabe jährlich durchgeführt. Den Loebner-Preis hat aber noch kein Chatbot gewonnen (Abdul-Kader & Woods, 2015, S. 74).

3 Semantic-Web und seine grundlegenden Technologien

Das Semantic-Web fusst auf der Problematik, dass Menschen die Bedeutung von Information auf einer Webseite erfassen und in Beziehung setzen können, während eine Maschine das nicht ohne Weiteres zu leisten vermag. In der Informationsrecherche wurden statistische Methoden eingesetzt, doch es wurde angestrebt, auch eine semantische Suche zu ermöglichen. Unterschiedliche Formate, Codierungen und Umsetzungsweisen stellen Hürden dar, wenn es darum geht, Information einheitlich aufzubereiten und wiederzuverwenden (Hitzler et al., 2008, S. 10). Der Lösungsansatz des Semantic-Web besteht darin, Information bereits so zur Verfügung zu stellen, damit Maschinen die Möglichkeit haben, sie zu verarbeiten. Hitzler et al. (2008, S. 11–12) listen die dazugehörigen Voraussetzungen auf:

- Offene Standards, um Information zwischen Anwendungen und Plattformen zu teilen. Grundlegende Standards für das Semantic-Web sind XML, RDF/RDF(S) und OWL. Erklärungen zu diesen Standards erfolgen in den nächsten Abschnitten.
- Die Fähigkeit, implizite Information herausziehen zu können. Dies wird durch Methoden zur Schlussfolgerung ermöglicht.

Die Technologien, welche dem Semantic-Web zugrunde liegen, werden semantische Technologien genannt. Sie sind sogenannte Wissensrepräsentationssprachen und Tools für Ontologien, welche die Erstellung, Wartung und Nutzung von Ontologien ermöglichen (Hitzler et al., 2008, S. 12). Hitzler et al. (2008) distanzieren sich von der Idee, dass durch das Semantic-Web Maschinen in die Lage versetzt würden, die Bedeutung von Information zu verstehen. Sie drücken es bescheidener aus: Information wird durch das Semantic-Web so repräsentiert, dass Maschinen damit in nützlicher Weise umgehen können (Hitzler et al., 2008, S. 11–12).

Die Begriffe Semantic-Web und Linked-Data werden häufig synonym verwendet. Tatsächlich sind die beiden Konzepte eng miteinander verbunden – beide werden vom World Wide Web Consortium (W3C) gefördert (Danowski & Pohl, 2013, S. 4). Laut dem W3C bezeichnet das Semantic-Web die Vision vom Web verlinkter Daten («Web of linked data»). Semantische Technologien werden von Menschen dazu genutzt, Datenspeicher im World Wide Web (WWW) aufzubauen, Vokabulare zu erstellen und Regeln für den Umgang mit Daten zu verfassen. Linked-Data wird durch Technologien wie RDF, SPARQL, OWL und SKOS unterstützt (W3C, 2015).

In den folgenden Abschnitten werden Standards und Technologien für das Semantic-Web kurz erläutert.

3.1 eXtensible Markup Language (XML)

Die Auszeichnungssprache³ eXtensible Markup Language (XML) ist grundlegend dem Web zuzuordnen, gehört aber auch zu den Standards für das Semantic-Web (Hitzler et al., 2008, S. 11). Die Beschreibungssprachen RDF(S) und OWL, welche weiter unten erläutert werden, bauen auf XML auf. XML wird vom W3C für maschinenlesbare Dokumente empfohlen. Die Auszeichnungssprache enthält Tags, welche die logische Struktur von Dokumenten auszeichnen. Da darüber hinaus durch XML unterschiedliche Auszeichnungssprachen allgemein definiert werden können, handelt es sich bei XML um eine sogenannte Meta-Sprache für die Erstellung von Auszeichnungssprachen (Hitzler et al., 2008, S. 18). XML bietet die Möglichkeit Daten soft- und hardwareunabhängig zu speichern, transportieren und mit anderen zu teilen (W3Schools, o. J.).

3.2 Ontologien

Bei Ontologien handelt es sich um semantische Netzwerke. Sie stellen eine Schlüsseltechnologie des Semantic-Webs dar und enthalten Konzepte, die relational und hierarchisch miteinander verbunden sind (Abdul-Kader & Woods, 2015, S. 73–74). Al-Zubaide & Issa (2011, S. 8) beschreiben Ontologien auch als Vokabulare, die für die Beschreibung eines Weltmodells genutzt werden. Sie sind in einer klar definierten Semantik geschrieben und ermöglichen es Computersystemen untereinander zu kommunizieren, unabhängig von deren einzelnen Technologien, Informationsarchitekturen und Anwendungsbereichen (Al-Zubaide & Issa, 2011, S. 8). Die Verbindung zwischen den beschriebenen Konzepten/Objekten kann in einem Graphen dargestellt werden. Maschinen können durch Graphen anhand bestimmter Regeln Schlussfolgerungen anstellen (Abdul-Kader & Woods, 2015, S. 73–74).

Die Nutzung von Ontologien in Chatbots hat zum Ziel, die Beziehungen zwischen den Konzepten zu berechnen. Dazu gehören beispielsweise Synonyme, Hyponyme aber auch andere Beziehungen, die natürlichsprachige Begriffsbezeichnungen darstellen (Abdul-Kader & Woods, 2015, S. 73–74). Mittels semantischer Technologien kann der Chatbot durch die Knoten eines Wissensgraphen navigieren, um eine Verbindung

³ Auszeichnungssprache deshalb, weil solche Sprachen dazu dienen Teile von Text-Dokumenten mit zusätzlicher Information auszuzeichnen beziehungsweise zu annotieren (Hitzler et al., 2008, S. 17).

zwischen den dargestellten Konzepten herzustellen, die in der Interaktion zwischen den Nutzenden und dem Chatbot thematisiert werden (Ramesh et al., 2017, S. 341).

In der vorliegenden Bachelorarbeit wird oft der Ausdruck *Wissensbasis* verwendet, womit Ontologie gemeint ist.

3.3 Resource Description Framework (RDF)

Durch das Resource Description Framework (RDF) werden Informationen (insbesondere Metadaten) von Ressourcen im WWW dargestellt. RDF bezweckt den Datenaustausch im Web, ohne dass dabei die ursprüngliche Bedeutung der Daten verloren geht (Freese, 2007; Hitzler et al., 2008, S. 35). Einem RDF-Ausdruck liegt eine Sammlung von Tripeln zugrunde. Ein Tripel verbindet Dinge (in RDF «Ressourcen» genannt) miteinander und stellt eine Aussage über die Beziehung dieser Dinge dar. Es besteht aus einem Subjekt, einem Prädikat (auch Eigenschaft/Property) und einem Objekt. Das Prädikat zeigt dabei die Beziehung zwischen dem Subjekt und dem Objekt an (Freese, 2007).

Ein RDF-Dokument repräsentiert einen gerichteten Graphen, der aus einer Menge von Knoten besteht. Knoten werden durch gerichtete Kanten miteinander verbunden. Knoten und Kanten sind beide mit eindeutigen Bezeichnern beschriftet (Hitzler et al., 2008, S. 36). Die Ressourcen werden mithilfe Uniform Resource Identifiers (URIs) identifiziert. Sie gelten global und verweisen immer auf die gleiche Ressource in jedem RDF-Dokument, in dem sie vorkommen. Literale stellen in RDF Werte von Daten dar (Freese, 2007). Ein einfaches Beispiel eines RDF-Graphen zeigt die Abbildung 3: Die Ressourcen werden mit URIs bezeichnet; der Name «Vanessa Seyffert» befindet sich in einem Literal, da er die Eigenschaft beziehungsweise den Wert einer Entität angibt.



Abbildung 3: Der RDF-Graph beschreibt die Beziehung zwischen der Bachelorarbeit und der Autorin (eigene Darstellung).

RDF-Schema

Um auch allgemeine schematische Information über einen Datensatz bezeichnen zu können, wird RDF um die Sprache RDF-Schema (RDF(S)) erweitert (Hitzler et al., 2008, S. 36). RDF(S) ist eine Ontologiesprache und ermöglicht es, Hintergrundinformation über die Begriffe in einem Vokabular zu definieren. Das terminologische Wissen wird dazu in RDF(S) in Form von Klassen- und Propertyhierarchien sowie deren Zusammenhängen spezifiziert (Hitzler et al., 2008, S. 66, 86). RDF(S) wird für leichtgewichtige Ontologien eingesetzt, da die Modellierungsmöglichkeiten mit RDF(S) eingeschränkt sind. Zum Beispiel ist es nicht möglich, Klassen- oder Propertybezeichnungen zu negieren, beziehungsweise würde die beabsichtigte Semantik nicht entsprechend interpretiert (Hitzler et al., 2008, S. 119).

3.4 Web Ontology Language (OWL)

Bei der Web Ontology Language (OWL) handelt es sich wie bei RDF(S) um eine Ontologiesprache. Sie ist ausdrucksstärker als RDF(S), das komplexe Zusammenhänge nicht genügend darzustellen vermag. Die aus RDF(S) bereits bekannten Klassen und Propertys können in OWL in komplexe Beziehung zueinander gesetzt werden. Die Extraktion impliziten Wissens wird durch logisches Schlussfolgern ermöglicht (Hitzler et al., 2008, S. 125–126).

3.5 SPARQL Protocol and RDF Query Language

SPARQL steht für das rekursive Akronym SPARQL Protocol and RDF Query Language. Es handelt sich um eine Abfragesprache für RDF, die vom W3C empfohlen wird. SPARQL-Abfragen können an unterschiedliche Datenquellen gerichtet werden (Zou, 2018). Als grundlegendes Anfragemuster werden RDF-Graphen genutzt. Um die Abfragesprache kurz zu veranschaulichen, wird das folgende Beispiel von Hitzler et al. (2008, S. 203) kurz erläutert:

```
PREFIX ex: <http://example.org/>
SELECT ?titel ?autor
WHERE
{
  ?buch ex:VerlegtBei <http://springer.com/Verlag> .
  ?buch ex:Titel ?titel .
  ?buch ex:Autor ?autor . }
}
```

`PREFIX` definiert den Namensraum. Durch `SELECT` wird das Ausgabeformat bestimmt, das heisst, die Werte der genannten Variablen (`?titel` und `?autor`) werden am Ende ausgegeben. `WHERE` leitet die Abfrage ein. Die geschweiften Klammern enthalten ein einfaches Graphmuster. Das Ergebnis der Abfrage wird die bei Springer erschienenen Bücher ausgegeben, die einen Titel und eine:n Autor:in haben (Hitzler et al., 2008, S. 203).

4 Herleitung der Forschungsfragen

In diesem Kapitel werden die Forschungsfragen und die ihnen zugrunde liegenden Überlegungen aufgezeigt.

4.1 Einsatz semantischer Technologien in der Chatbot-Entwicklung

Der Nachteil im klassischen Chatbot-Design besteht darin, dass alle möglicherweise eintreffenden Konversationselemente vorgängig gebaut werden müssen, da der Chatbot ansonsten die Absichten der Nutzenden nicht korrekt zuweisen kann (Di Blas et al., 2019, S. 6). Dies führt dazu, dass Tausende von möglichen Antworten im Chatbot-System manuell einzutragen sind (Porreca et al., 2018, S. 159). Darüber hinaus werden die Antworten des Chatbots auf diese Weise voraussehbar und repetitiv (Ramesh et al., 2017, S. 339).

Generell sollten Chatbots das Wissen in einer angemessenen Weise repräsentieren, doch genau diese Darstellung des Wissens stellt eine der grossen Grenzen in der Chatbot-Technologie dar. Weitere Hürden sind die Informationsbeschaffung und die Dialogfähigkeit. Zwar ist es möglich auf einfache Weise Chatbots umzusetzen, doch die einfachen Mustervergleichsregeln beschränken die Fähigkeiten des Systems. Darüber hinaus gilt die händische Erstellung einer Wissensbasis für den Chatbot als teuer und langweilig (Augello et al., 2014, S. 286). Der Aufbau einer starken Wissensbasis kann Jahre in Anspruch nehmen. Eine Übertragung auf andere Wissensdomänen ist im Anschluss kaum möglich (Al-Zubaide & Issa, 2011, S. 8).

In der Literatur werden verschiedene Ansätze präsentiert, um den Prozess des Wissensdesigns und der Antwortgenerierung zu verbessern. Eine Möglichkeit besteht darin, das formalisierte Wissen aus Ontologien oder linguistischen Ressourcen zu nutzen. Dies kann es dem Chatbot ermöglichen, emotionale Inhalte zu erkennen oder aufgrund semantischer Analysefähigkeiten Inferenzschlüsse zu ziehen. Dies begünstigt auch den Umgang mit verschiedenen Ausdrücken der Nutzenden sowie das Erkennen von Ähnlichkeitsbeziehungen zwischen Wörtern (Augello et al., 2014, S. 286, 2017, S. 237).

Durch den Fortschritt des Semantic-Web ist eine grosse Menge an Daten im WWW verfügbar geworden. Die in Wissensbasen strukturierten Daten können mit Chatbots verknüpft und Nutzenden zugänglich gemacht werden (Ait-Mlouk & Jiang, 2020, S. 149220). In der Literatur konzentrieren sich viele neue Lösungsansätze auf Künstliche

Intelligenz (KI). Dabei werden jedoch die Stärken von Linked-Data und semantischen Technologien ausser Acht gelassen (Ait-Mlouk & Jiang, 2020, S. 149228–149229).

Die Bachelorarbeit hat zum Ziel zu untersuchen, welche semantischen Technologien für die Erstellung von Chatbots eingesetzt werden. Die ersten zwei zu behandelnden Forschungsfragen lauten:

1. *Welche semantischen Technologien werden für die Erstellung von Chatbots eingesetzt?*
2. *Wie werden die semantischen Technologien für die Erstellung von Chatbots eingesetzt?*

Darüber hinaus werden Erkenntnisse darüber gewonnen, inwiefern semantische Technologien die Erstellung von Chatbots unterstützen, sprich welche Vorteile diese Technologien für Erstellung und Funktion von Chatbots bieten. Die dritte Forschungsfrage lautet demnach:

3. *Inwiefern unterstützen die semantischen Technologien die Erstellung von Chatbots?*

4.2 Chatbots in Bibliotheken

In einem zweiten Schritt wird die Forschungsfrage weiter eingegrenzt auf ein bestimmtes Themengebiet. Die Autorin bezieht sich aufgrund ihrer Tätigkeit in einer Hochschulbibliothek auf Chatbots in Bibliotheken. Das Thema Chatbots in Bibliotheken wird seit mehreren Jahren untersucht und gemäss Literatur wird aktuell nach Wegen gesucht, Chatbots in Bibliotheken zu verbessern (unter anderem in den Studien von Bagchi, 2020; Hanselmann, 2020; Mckie & Narayan, 2019; Panda & Chakravarty, 2021; Vincze, 2017). Die Bibliothek der FHGR verfügt über einen Chatbot, der Auskunftsdienstleistungen anbietet.⁴ Freundlicherweise wird der Chatbot zu Anschauungszwecken für diese Bachelorarbeit zur Verfügung gestellt. Anhand des Beispiels dieses Chatbots wird die Forschungsfrage geklärt, wie sich die Anwendung semantischer Technologien auf die Chatbot-Entwicklung im Bibliotheksbereich übertragen lässt. Die vierte Forschungsfrage lautet:

4. *Wie lässt sich die Anwendung semantischer Technologien auf die Chatbot-Entwicklung im Bibliotheksbereich übertragen?*

⁴ Der Chatbot Viola der FHGR-Bibliothek läuft über Pandorabots und ist über folgenden Link erreichbar: <https://www.pandorabots.com/pandora/talk?botid=8f147f626e3770b1>.

5 Methodische Vorgehensweise

Für die Einarbeitung in das Themengebiet wurden zu Beginn der Arbeit grundlegende Kenntnisse über konventionelle Entwicklungsweisen, Eigenschaften, Kategorien und Anwendungen von Chatbots angeeignet. Die Bachelorarbeit startete mit einer systematischen Literaturrecherche. Neben der Bibliotheksplattform Swiscovery, die vor allem für die Einstiegsrecherche genutzt wurde, wurden die folgenden Datenbanken abgesucht:

- IEEE Xplore Digital Library
- Library, Information Science & Technology Abstracts (LISTA)
- Library and Information Science Abstracts (LISA)
- Infodata via WTI TecFinder

Die Wahl fiel auf diese Datenbanken, da sie einen Bezug zur Informationswissenschaft aufweisen und der Autorin zur Verfügung standen. Die Recherche wurde mit Google Scholar und einer Suche nach dem Schneeballprinzip ergänzt. Für die systematische Suche wurde für jede Forschungsfrage ein sogenanntes Wortfeld ausgefüllt. Bei einem solchen werden die Kernbegriffe einer Forschungsfrage herausgefiltert, um deren Synonyme, Ober- und Unterbegriffe, verwandten Begriffe usw. in einem Raster zu sammeln. Dies dient der anschliessenden Erarbeitung von Suchstrings für die unterschiedlichen Quellen. Jedes Wortfeld wurde immer auch auf Englisch übersetzt, da die meiste Literatur zu Chatbots auf Englisch zur Verfügung steht. Tabelle 1 zeigt beispielhaft das ausgefüllte Wortfeld für die Forschungsfrage 1: «Welche semantischen Technologien werden für die Erstellung von Chatbots eingesetzt?».

Tauchten im Lauf der Recherche weitere Begriffe auf, wurden diese im Wortfeld ergänzt. Als Ergebnis für die Beantwortung der ersten und zweiten Forschungsfrage wurde aufgrund der gefundenen Literatur eine Liste der Chatbot-Projekte erstellt, die semantische Technologien nutzen (zu finden in 6.1 Chatbots mit semantischen Technologien **Fehler! Verweisquelle konnte nicht gefunden werden.**). Interessant war dabei auch zuzuordnen, für welche Chatbot-Kategorie semantische Technologien eingesetzt werden. Die Autorin erhoffte sich dadurch Erkenntnis darüber zu erhalten, ob es bestimmte Chatbot-Kategorien gibt, für die semantische Technologien besonders gut in Frage kommen. Dieser Erkenntnisgewinn stand aber nicht im Fokus der Arbeit.

Kernbegriffe	Semantische Technologie	Erstellung	Chatbot
Synonyme	Semantic-Web	Herstellung, Entwicklung, Produktion	virtueller Assistent, Talkbot, Konversationsagenten, Dialogsysteme, Smart Bots, interaktive Agenten, digitale Assistenten, künstliche Gesprächseinheiten
Oberbegriffe			Computersystem, Computerprogramm
Unterbegriffe	SPARQL, RDF, RDF(S), Ontologie, Wissensgraph, RDF- Graph, Web Ontology Language, OWL	Design	
Verwandte Begriffe	Linked-Data, annotierte Datensätze/Ressourcen, semantisches Netzwerk, Wissensrepräsentation, Protégé		Künstliche Intelligenz, Roboter, AIML
Antonyme			

Tabelle 1: Beispiel eines Wortfeldes auf Deutsch für die Forschungsfrage 1.

Um die dritte Forschungsfrage zu beantworten, wurden aus der wissenschaftlichen Literatur Messkriterien für die Evaluierung der Qualität von Chatbots erhoben (zusammengefasst in 6.2 Vorteile von semantischen Technologien in Chatbots). Damit die Vorteile semantischer Technologien dokumentiert werden können, war eine Untersuchung der gefundenen Chatbot-Projekte geplant. Knackpunkt dabei war, dass nur eine der neun aufgelisteten Chatbot-Ideen tatsächlich als öffentlich zugänglicher Chatbot konzipiert wurde. Zwar wurde innerhalb einiger der Chatbot-Projekte durch die Forschenden selbst eine Evaluierung durchgeführt, jedoch nicht bei allen Projekten und die Ergebnisse waren nicht immer aussagekräftig. Die Autorin dieser Bachelorarbeit fokussierte sich deshalb auf häufig auftretende Mängel bei Chatbots, die aus der Literatur abgeleitet wurden. Die Mängel wurden dahingehend untersucht, ob sie durch die vorgängig erfassten semantischen Technologien behoben werden konnten. So war es schliesslich möglich die vorteilhaften Auswirkungen semantischer Technologien in der Chatbot-Entwicklung zu benennen.

Aufbauend auf den bis dahin erzielten Erkenntnissen wurde die vierte Forschungsfrage in Angriff genommen. Um die Anwendung semantischer Technologien bei einem Bibliotheks-Chatbot zu betrachten, wurde der Chatbot *Viola* der FHGR Bibliothek als

Anschauungsobjekt herangezogen. Aufgrund des Zwecks und der Zielgruppe von Viola wurde abgewogen, inwieweit eine Erweiterung mit semantischen Technologien Sinn macht (6.3 Semantische Technologien für den Bibliotheks-Chatbot Viola). Im Anschluss wurden die semantischen Technologien aus den identifizierten Chatbot-Projekten auf ihre Eignung für Viola hin überprüft.

6 Darstellung der Ergebnisse

In diesem Kapitel werden die Ergebnisse der Untersuchung wiedergegeben. Im Unterkapitel 6.1 wird den Forschungsfragen nachgegangen, welche semantischen Technologien für die Erstellung von Chatbots eingesetzt werden und wie diese eingesetzt wurden. Unterkapitel 6.2 zeigt auf, inwiefern semantische Technologien die Erstellung von Chatbots unterstützen. Schliesslich wird in 6.3 darauf eingegangen, wie sich die Anwendung semantischer Technologien auf den Bibliotheks-Chatbot Viola übertragen lässt.

6.1 Chatbots mit semantischen Technologien

Stock (2001), Koutamanis (2002) und Mori et al. (2003) schlugen bereits früh Mechanismen zur Folgerung von Inferenzschlüssen vor. Mori et al. (2003, S. 271) hielten ebenfalls fest, dass die Umsetzung digitaler Assistenten mithilfe semantischer Technologien für die Entwickler:innen von Vorteil sei, da sie nicht jede einzelne Eingabe der Nutzenden vorhersagen müssen und sich stattdessen auf die Erstellung der Unterhaltung innerhalb einer bestimmten Domäne konzentrieren können (Mori et al., 2003, S. 271). Gemäss Zdravkova (2000) war das grösste Manko der meisten Chatbots im Jahr 2000 die Generierung von Antworten, nachdem die Eingabe mit den in der Datenbank gespeicherten Mustern abgeglichen wurde. Zwar fand ein Parsing, doch nie eine semantische Analyse statt, obwohl die Fähigkeit die Anfrage der Nutzenden zu verstehen das anspruchsvollste Ziel intelligenter Chatbots sei. Um dieses Problem zu lösen, schlug Zdravkova die Integration eines Lexikons vor, das mit einem Thesaurus erweitert wurde. Dadurch wurde eine semantische Analyse simuliert. Der von ihr erstellte Prototyp eines Chatbots war in der Lage mithilfe vergangener Konversationen neue Lexeme und Syntagma-Strukturen zu erlernen (Zdravkova, 2000, S. 189–191).

Freese (2007) präsentierte einen Ansatz, um RDF-Tripel in AIML-Themen/-Kategorien umzuwandeln. Das Domänenwissen aus unterschiedlichen Datenbasen kann so in einer Chatbot-Unterhaltung eingebunden werden. Mit dem System «Semetag» machte es Freese möglich, Metadaten aus gängigen Anwendungen mit RDF zu sammeln und in einem Paket zu verwalten. Für Semetag wird Jena verwendet, ein Java Framework für die Entwicklung von Semantic-Web- und Linked-Data-Anwendungen.⁵ Gemäss Freese verfügt Jena über eine RDF-API, eine OWL-API, eine Abfragefunktionen mit RDQL (RDF

⁵ Download und Dokumentation von Apache Jena sind zu finden unter <https://jena.apache.org/>.

Data Query Language) und eine regelbasierte Inferenzmaschine. Nachdem Semetag zur Verfügung stand, erstellte Freese eine Schnittstelle zwischen einem AIML-Chatbot und den RDF-Metadaten. Das Ergebnis nannte er AIMEE (Artificially Intelligent Metadata Enabled Entity). Integriert wurde in AIMEE die ALICE-Wissensbasis sowie Komponenten von Semetag. Für die Umsetzung von AIMEE wurden Java und PHP eingesetzt. Zwar konnte durch diesen Ansatz die Wissensbasis auf einfache Weise erweitert werden, allerdings stellte die Konversation mit den Nutzenden immer noch eine Herausforderung dar. Freese musste eine Reihe von Abfragen weiterhin von Hand entwickeln, um die Fragemöglichkeiten der Nutzenden abzudecken (Freese, 2007).

Unger et al. (2012) stellten einen Ansatz vor, bei welchem Linked-Data mittels natürlicher Sprache für die Beantwortung von Fragen genutzt werden kann. Die Anfrage der Nutzenden wurden dabei einer linguistischen Analyse unterzogen, um anschliessend eine SPARQL-Vorlage zu erzeugen, die mit URIs gefüllt wurde. In diesem Projekt lag der Fokus auf der Umwandlung der Eingaben. Die semantische Struktur der Eingabe wurde genau erfasst, um die notwendigen Entitäten mit NLP- und statischen Methoden zu identifizieren (Unger et al., 2012, S. 639–640).

Laut Nazir et al. (2019, S. 547) gilt die Entwicklung von Chatbots, die auf Ontologien basieren, generell als eine vielversprechende Methode. Neben den oben bereits erwähnten Ansätzen finden sich in Tabelle 2 Chatbot-Projekte, welche das Ziel verfolgen, semantische Technologien für die Entwicklung von Chatbots zu nutzen. Enthalten sind neuere Projekte der letzten vier Jahre, allerdings wurden auch ältere Projekte aufgelistet, die dabei helfen eine Gesamtübersicht über die eingesetzten Technologien zu erhalten. Eine kurze Erläuterung zu den Tabellen-Kolonnen:

- Chatbot aktiv? zeigt an, welche der Chatbots online zur Verfügung stehen und genutzt werden können.
- In der Chatbot-Kategorie wird darauf eingegangen, ob es sich um Open- oder Closed-Domain Chatbots handelt. Grund dafür ist die Frage der Autorin, ob sich semantische Technologien auch für Open-Domain-Chatbots eignen. Auch wird die Art der Inputverarbeitung/Antwortgenerierung sowie die Zielorientierung aufgelistet, um allenfalls Schlüsse daraus ziehen zu können. Konnte die Kategorie aufgrund der Literatur nicht identifiziert werden, wurde sie weggelassen.
- Unter Technologien/Tools/Wissensbasen sind nicht alle in den Chatbots eingesetzten Tools aufgelistet, sondern diejenigen, welche für den Bereich Semantic-Web relevant sind. Dazu gehört beispielsweise, welche Ontologie für den Chatbot jeweils verwendet wurde, auf welchem Wege diese abgefragt wurde usw.

Name / Projekt	Chatbot aktiv?	Chatbot-Kategorie	Technologien/Tools/Wissensbasen	Entwicklungsteam
KBot	-	Closed-Domain, retrieval-basiert, zielorientiert	Abfrage von DBpedia, Wikidata und myPersonality mittels SPARQL. Unklar, ob AIML eingesetzt wurde (eher nicht).	Ait-Mlouk & Jiang (2020)
R2D2	-	Closed-Domain, retrieval-basiert, zielorientiert	Abfrage von DBpedia mittels SPARQL. Nutzende müssen Anfragen bereits in Form von Tripeln stellen.	Kondylakis et al. (2020)
[Chatbot für Modemarken in Pakistan]	-	Closed-Domain, retrieval-basiert, zielorientiert	Selbständig Ontologie für Modemarken mit Protégé erstellt. Bereitstellung der Ontologie mit Jena, Abfrage mit SPARQL.	Nazir et al. (2019)
DBpedia Chatbot	✓	Closed-Domain, retrieval-basiert, zielorientiert	Abfrage von DBpedia (mit SPARQL), OpenStreetMap und weiterer Datenbasis zu DBpedia Community-Fragen. Einsatz von Qanary. RiveScript statt AIML für reine Texteingaben.	Athreya et al. (2018)
[Ontologiebasierter Chatbot für E-Commerce]	-	Closed-Domain, retrieval-basiert	Selbständige Erstellung der Ontologie durch Abfrage der Ebay-API. Ontologie in Protégé, Anwendung von JAPE und Python.	Vegešna et al. (2018)
OntBot	-	retrieval-basiert	Übertragung der Ontologie in eine relationale Datenbank, um diese dann abfragen zu können. Keine konkrete Umsetzung des Ansatzes bekannt, deshalb keine konkreten Tools/Ontologien erwähnt.	Al-Zubaide & Issa (2011)
Project CyN	-	Closed-Domain, retrieval-basiert	Abfrage von OpenCyc durch AIML-Chatbots	Daxtron Laboratories (2004)
Project CyD	-	Closed-Domain, retrieval-basiert, zielorientiert	Anpassung von CyN an den in Java ProgramD geschriebenen AIML-Interpreter. Abfrage von OpenCyc und ResearchCyc durch AIML-Chatbots.	Augello et al. (2012/2014)
EHeBby	-	Nicht-zielorientiert	ProgramW (AIML-Interpreter auf Basis von Java), Abfrage von WordNet, um Sprachanalysefähigkeit des Chatbots zu verbessern.	Pilato et al. (2008)

Tabelle 2: Zusammenfassung der Chatbot-Projekte, welche semantische Technologien einsetzen.

6.1.1 KBot (2020)

Ait-Mlouk & Jiang (2020) erstellten eine Architektur für die Abfrage verschiedener Wissensbasen, darunter DBpedia, WikiData und myPersonality. myPersonality ist eine der grössten sozialwissenschaftlichen Forschungsdatenbanken. Sie wurde von über sechs Millionen Freiwilligen auf Facebook genährt. Initiiert wurde das Projekt von Daniel Stillwell im Jahr 2007, indem er einen Persönlichkeitsfragebogen auf Facebook teilte. Die anonymisierten Daten sind über eine virtuelle Datenbank für Forschende zugänglich (Ait-Mlouk & Jiang, 2020, S. 149225). Bei DBpedia handelt es sich um ein Projekt, bei welchem strukturierte Inhalte aus verschiedenen Wikimedia-Projekten extrahiert werden. Die im Web verfügbaren Daten sind in einer Art Wissensgraph strukturiert und können beispielsweise mit SPARQL abgefragt werden (DBpedia Association, 2021). Die Wissensbasis Wikidata ist ein Wissensgraph, der von der Wikimedia Foundation betreut wird. Auch Wikidata bietet eine SPARQL-Query. Wikidata steht nicht nur Wikimedia-Projekten zur Verfügung, sondern kann frei weitergenutzt werden (Wikidata, 2019).

Neben einer interaktiven Benutzungsoberfläche schlagen Ait-Mlouk & Jiang einen maschinellen Lernansatz vor, der die Klassifizierung der Absichten miteinbezieht und ebenso NLU-Techniken für das Verständnis der Anfragen der Nutzenden enthält. Um die Wissensbasen abzufragen, werden aufgrund der vorher identifizierten Absichten SPARQL-Abfragen generiert. Durch den modularen Ansatz ist KBot fähig, unterschiedliche Aufgaben auszuführen und Informationen aus mehreren Quellen zu sammeln. Die gesammelte Information präsentiert er den Nutzenden in Form von Wissenskarten. Bei der Modellierung dieses Chatbots stellten sich verschiedene Herausforderungen, darunter die Klassifizierung der Absichten der Nutzenden, das Verständnis für die Anfragen der Nutzenden, die Unterstützung verschiedener Wissensdatenbanken sowie die Mehrsprachigkeit (Ait-Mlouk & Jiang, 2020, S. 149222). Im Folgenden wird konkret auf die Arbeitsmechanismen des Chatbots eingegangen.

NLU-Modul

Das NLU-Modul erkennt als erstes die Sprache der Nutzenden. Damit stellt KBot bereits eine besondere Funktion zur Verfügung, denn viele andere Chatbots können nur in einer Sprache genutzt werden. Da eine grosse Anzahl von Ressourcen auf Englisch verfügbar ist, wurden die meisten Chatbots für englische Nutzende entwickelt. Damit KBot die Sprache der eingegebenen Anfrage automatisch erkennt, haben Ait-Mlouk & Jiang eine langdetect-Bibliothek verwendet, wodurch das Modul Anfragen analysieren und die Sprache automatisch erkennen kann. Die identifizierte Sprache wird anschliessend vom

Modul «Response Retrieval» verwendet. Ist keine Identifikation der Sprache möglich, wird die englisch Sprache genutzt (Ait-Mlouk & Jiang, 2020, S. 149223).

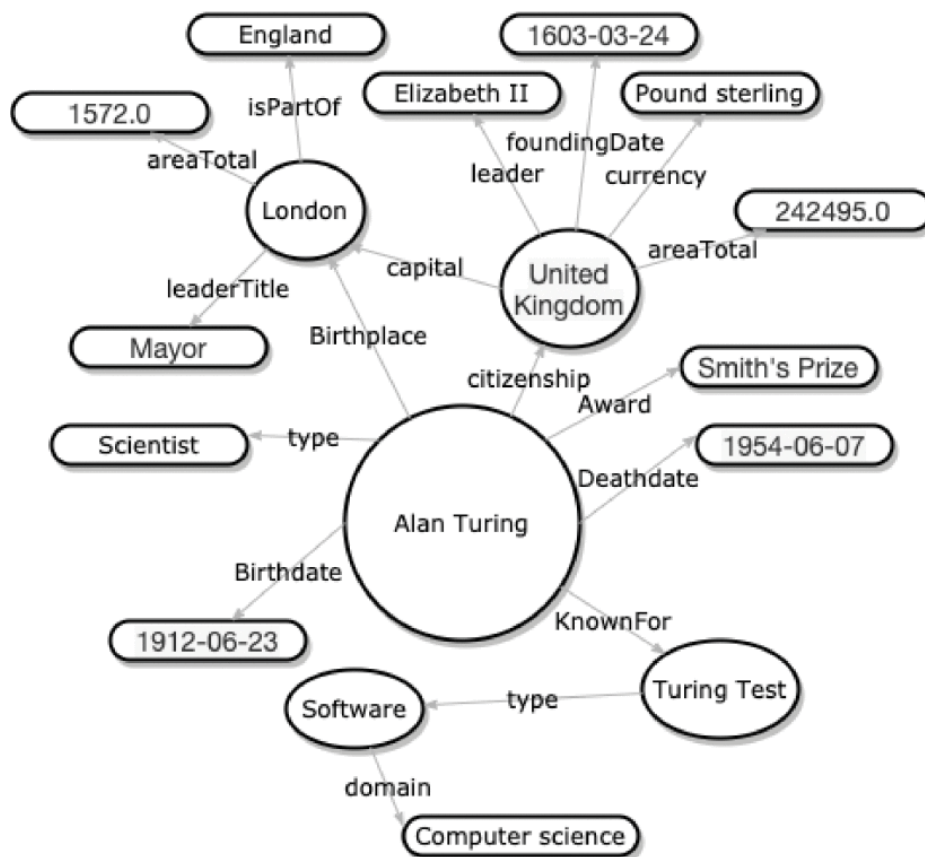


Abbildung 4: Wissensgraph für die extrahierte Entität, gefragt wurde nach Alan Turing (Ait-Mlouk & Jiang, 2020, S. 149224).

Nach der Spracherkennung wird die Absicht mittels «Support Vector Machine» (SVM) klassifiziert, um die verschiedenen Arten von Anfragetypen voneinander zu unterscheiden. Ebenfalls wird ein maschinelles Lernmodell namens «Support Vector Classifier» (SVC) eingesetzt, um die Algorithmen für die Klassifizierung zu unterstützen. Die Absichtsklassifizierung ist normalerweise der erste Schritt in Konversationssystemen. Das Parsing bei KBot geschieht zusammen mit «Regular Expressions» (RegEx). Um die in der Anfrage genannten Entitäten zu extrahieren, kommt die «Named Entity Recognition» (NER) zum Einsatz. Das Submodul NER übernimmt die Kernaufgabe für das Verständnis der Anfrage in natürlicher Sprache (Ait-Mlouk & Jiang, 2020, S. 149222–149223). NER extrahiert Information über verschiedene Typen von Entitäten, Beziehungen oder Ereignisse und nutzt dafür Tokenisierung und Wortart-Tagging. In Abbildung 4 ist ein Wissensgraph dargestellt, der aufgrund der extrahierten Entität aus der Absicht erstellt wurde (Ait-Mlouk & Jiang, 2020, S. 149224):

Je nachdem, ob die Anfrage nach Faktenwissen (gehört zu Klasse Faktenwissen) oder nach Wissen aus dem sozialen Netzwerk (myPersonality – gehört zur Klasse soziales Netzwerk) verlangt, wird die Anfrage durch das Submodul «QuestionToSparql» in eine SPARQL-Anfrage für eine der Wissensdatenbanken DBpedia oder Wikidata generiert, oder für das Datenset von myPersonality. Nach Auslesen der passenden Antwort wählt das Modul «Response Selection» die relevante Antwort und präsentiert sie den Nutzenden (Ait-Mlouk & Jiang, 2020, S. 149222–149223).

Information Retrieval

Für das Information Retrieval entnimmt der Chatbot durch «Keyword Extraction» (Entnahme von Schlüsselwörtern) die wichtigsten Wörter aus der Anfrage. Es handelt sich um eine Textanalysetechnik, bei der automatisch die Begriffe identifiziert werden, die den Inhalt des Textes zusammenfassen und die Hauptthemen benennen. Die SPARQL-Abfragen werden anschliessend je nach Wissensbasis (DBpedia, Wikidata, myPersonality) interpretiert. Die Herausforderung in diesem Bereich betraf einerseits die effiziente Generierung der SPARQL-Abfrage aufgrund der Nutzendenanfrage, andererseits stellt aber auch die Abfrage verschiedener Wissensbasen ein Hürde dar (Ait-Mlouk & Jiang, 2020, S. 149224). In anderen Chatbots werden oft Trainingsdaten verwendet oder es wird nur auf eine Wissensbasis fokussiert. Nachteil dabei ist, dass andere Wissensquellen so nicht berücksichtigt werden und sich die möglichen Antworten reduzieren (Ait-Mlouk & Jiang, 2020, S. 149225).

Aufgrund der grossen Datenmenge in den Wissensbasen war es notwendig, eine Technik zu wählen, welche Text für die Antwortausgabe automatisch zusammenfasst. Ait-Mlouk & Jiang haben hierzu die NLTK-Bibliothek⁶ verwendet, um die Ergebnisse mit TF-IDF zusammenzufassen und zu verbessern. TF-IDF steht für «Term Frequency» (Vorkommenshäufigkeit) und «Inverse Document Frequency» (inverse Dokumenthäufigkeit). Die TF gibt an, wie häufig ein Wort im Dokument vorkommt. Geteilt wird diese Zahl durch die Gesamtzahl an Wörtern im Dokument. Die IDF für ein bestimmtes Wort hängt nicht vom einzelnen Dokument ab, sondern von der Gesamtheit der Dokumente in der Dokumentensammlung. Bei der IDF wird die Anzahl der Dokumente in der Sammlung geteilt durch die Anzahl derjenigen Dokumente aus derselben Sammlung, die das bestimmte Wort enthalten (Ait-Mlouk & Jiang, 2020, S. 149225).

⁶ Das Natural Language Toolkit (NLTK) bietet Hilfestellung in der Programmierung von Python-Programmen, um Textkorpora mittels NLP-Methoden zu verarbeiten. Mehr dazu auf <https://www.nltk.org/>.

Das Modul für die Antwortauswahl sammelt schliesslich alle abgerufenen Informationen und präsentiert sie den Nutzenden. Ein Wissensgraph ermöglicht eine effiziente Verarbeitung der Entitäten, Objekte und Ereignisse. Er verknüpft und kombiniert die in den zuvor eingesetzten Modulen gesammelte Information. Das Zusammenspiel der vorgestellten Module wird in Abbildung 5 veranschaulicht: Mit NLU wird die eingegebene Anfrage verarbeitet; «capital» und «Spain» werden als essentielle Begriffe herausgefiltert. Im Wissensgraph (KG für «Knowledge Graph») werden die Begriffe miteinander kombiniert, um anschliessend das zugehörige Subjekt zu präsentieren. Die Antwort wird als Wissenskarte dargestellt, die mittels TF-IDF zusammengefasst wurde (Ait-Mlouk & Jiang, 2020, S. 149225–149226).

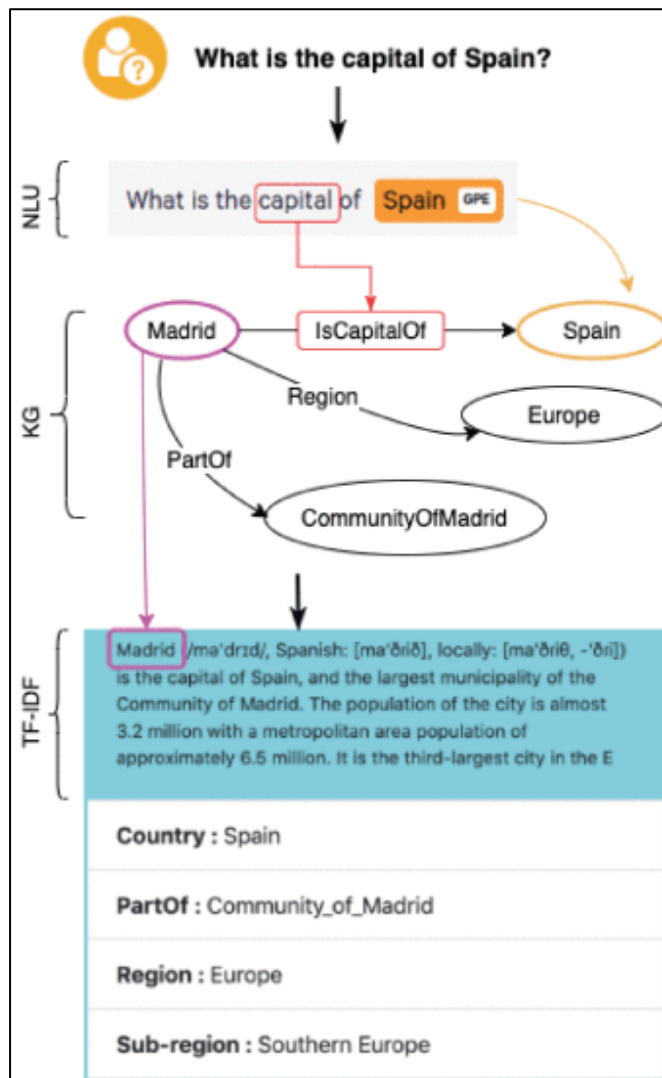


Abbildung 5: Beispiel einer generierten Antwort in KBot (Ait-Mlouk & Jiang, 2020, S. 149226).

Ebenfalls in KBot implementiert ist eine Datenbankansicht, in der die Anfragen der Nutzenden sowie Feedbacks gespeichert werden. Diese werden analysiert, um die Anwendung zu verbessern. Darüber hinaus wurde das Chatterbot-Framework⁷ eingesetzt, um den Chatbot für gewisse Aufgaben zu trainieren (Ait-Mlouk & Jiang, 2020, S. 149226).

6.1.2 R2D2 (2020)

Ein weiteres Projekt, bei dem SPARQL für die Abfrage einer Ontologie genutzt wird, ist R2D2. Es handelt sich um einen Chatbot, mit welchem auch DBpedia abgefragt wird. In diesem Chatbot werden die Nutzenden vom Chatbot instruiert, wie sie die Eingabe zu formulieren haben, damit der Input in einem Triple erfolgt. Unterstützt werden die Nutzenden während der Eingabe ihrer Fragen durch eine Autovervollständigungs-Funktion (Kondylakis et al., 2020, S. 1–2). Die Architektur von R2D2 setzt sich aus der DBpedia Wissensbasis mit SPARQL-Endpoint und den beiden folgenden Schichten zusammen:

Graphical User Interface (GUI)

Die Schnittstelle zu den Nutzenden wurde mit HTML, CSS und JavaScript umgesetzt. Ebenso kam Bootstrap zum Einsatz, um eine responsive Website zu gewährleisten. Auf der Website zu finden ist der Chatbot R2D2, der Anfragen in Form von Triples akzeptiert (*Entity, Property, Value*). Sobald Nutzende anfangen etwas einzutippen, hilft die Autovervollständigung mit Vorschlägen. Wurde eine Entität eingegeben, ergeben sich bereits die verfügbaren Verbindungen, was die Eingabe erleichtern soll. Nach Eingabe einer Anfrage wird die Antwort in Form von Wissenskarten präsentiert. Triviale Antwortmöglichkeiten ohne Zugriff auf DBpedia wurden von Kondylakis et al. ergänzt (Kondylakis et al., 2020, S. 4).

Intelligent Services

Die erwähnte Autovervollständigung wurde mit einem JQuery-Autocomplete-Widget erstellt. Für die Suche innerhalb DBpedia wurde dessen Lookup-Service genutzt und erweitert. Beim Vorgang der Autovervollständigung werden DBpedia-URIs zum

⁷ Bei ChatterBot handelt es sich um eine Python-Bibliothek. Sie bietet eine Auswahl von Algorithmen für maschinelles Lernen, um Antworten zu erzeugen. Weitere Details dazu finden sich unter <https://chatterbot.readthedocs.io/en/stable/>.

Nachschriften verwendet. Zurückgegeben werden die Daten entweder in XML oder JSON.

Das folgende Beispiel zeigt eine SPARQL-Abfrage in DBpedia für eine eingetippte Entität (Kondylakis et al., 2020, S. 5):

```
SELECT DISTINCT ?label
WHERE {
    <ENTITY> ?p ?o.
    ?p <http://www.w3.org/2000/01/rdf-schema#label> ?label.
    FILTER(LANG(?label) = "" || LANGMATCHES(LANG(?label), "en"))
}
```

Der Service «Text2SPARQL» formuliert nach Erhalt der Nutzendenanfrage eine SPARQL-Abfrage. Sollte eine nicht wohl geformte Anfrage gestellt worden sein, würde der Chatbot dem:der Nutzenden einen entsprechenden Hinweis liefern. «Text2SPARQL» fügt für die eingegebene Anfrage die Präfixe für die SPARQL-Abfrage hinzu und identifiziert die Triple-Muster. Die Variablen aus dem Triple werden nach «SELECT» eingefügt (Kondylakis et al., 2020, S. 6). Der Service «Results2Text» gibt die abgerufene Antwort an die GUI weiter, wo sie den Nutzenden angezeigt wird (Kondylakis et al., 2020, S. 6–7).

6.1.3 Ontologiebasierter Chatbot für Modemarken in Pakistan (2019)

Nazir et al. (2019) haben für ihren Chatbot einen Datensatz von 5000 Fragen und Antworten zu den Top-10 Modemarken in Pakistan ausgewertet. Sie haben eine Dialogschnittstelle für die Abfrage dieses Datensatzes entwickelt. Umgesetzt haben sie die Arbeit mit Protégé (inklusive Onto-Graf- und OWL-Plugin für die Visualisierung der Ontologie) und für die Datenabfrage haben sie SPARQL genutzt (Nazir et al., 2019, S. 547). Protégé ist ein frei verfügbares open-source Tool, mit welchem Ontologien erstellt und angepasst werden und abgefragt werden können. Enthalten sind unter anderem auch sogenannte Reasoner, welche Schlussfolgerungen durchführen und dabei helfen, die Ontologie auf ihre Konsistenz hin zu überprüfen.⁸

Nazir et al. haben nicht eine bereits bestehende Ontologie für ihren Chatbot verwendet, sondern selbständig eine in Protégé aufgebaut. Thematisch fokussierten sie sich auf die

⁸ Zur Verfügung gestellt wird Protégé von der Stanford University. Das Programm kann über die Webseite heruntergeladen werden: <https://protege.stanford.edu/>.

Top10 Modemarken in Pakistan und bauten auf allgemeinen Informationen auf, wie das Angebot an Marken, Dienstleistungen, Kleidungsstücke und Accessoires. Als Quellen für die Ontologie dienten unter anderem Facebook-Seiten sowie offizielle Webseiten (Nazir et al., 2019, S. 548, 552). Die Bildung der Ontologie umfasste folgende Schritte:

1. Vorbereitung des Datensatzes: Die Daten wurden einerseits manuell mittels verschiedener Techniken (Umfragen, Interviews) gesammelt. Um die Information von Facebook-Posts und -Kommentaren zu sammeln, wurde das Add-on «Scrapper» in Google Chrome verwendet. Das Resultat der Sammlung umfasste einen Korpus unstrukturierter und inkonsistenter Daten, die zunächst verarbeitet werden mussten, damit sie für Semantic-Web-Anwendungen zur Verfügung standen. Der Korpus umfasste am Ende 5000 Sätze.
2. Kompetenzfragen: Nach Erfassung der Daten wurde geklärt, welche Fragen die Ontologie beantworten können muss. Allgemein hilft dieser Schritt dabei die Klassen und Hierarchiestufen der Ontologie festzulegen. Beispiele von Fragen der vorliegenden Ontologie sind: «Welche Marke bieten Accessoires an?» oder «Welche Marke bietet 50% Rabatt?».
3. Konzepte und Klassen: Aufgrund der Kompetenzfragen wurden die Ober- und Unterklassen bestimmt. Das Ergebnis der Klassenstruktur ist in Abbildung 6 zu sehen.
4. Festlegung der Eigenschaften («Properties»), Attribute und Prädikate: Klassen sind nicht selbstbeschreibend, weshalb definiert wird, wie sie zueinander stehen (Nazir et al., 2019, S. 548).

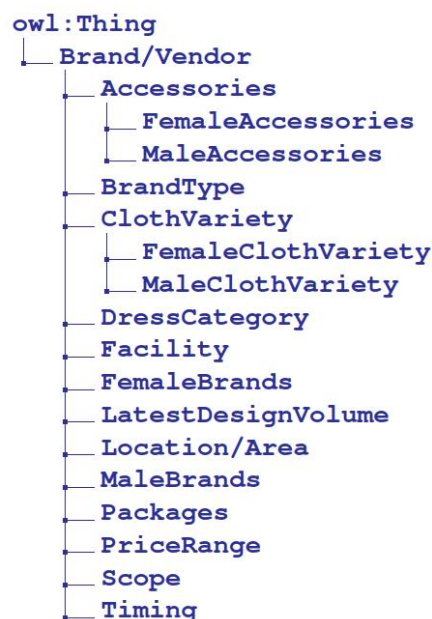


Abbildung 6: Ansicht der von Nazir et al. gebildeten Klassen (Nazir et al., 2019, S. 548).

Nach Konzipierung der Ontologie wurden die Regeln für das Backend ausgearbeitet, welche verschiedene Abfrageszenarien abdecken. Das Ziel war, dass das System kontextorientierte Antworten liefert (Nazir et al., 2019, S. 550). Bereitgestellt wurde die Ontologie schliesslich mit Jena, einer Java-basierten Bibliothek, welche Anwendungen des Semantic-Web unterstützt. Um auf die aktive Ontologie zugreifen zu können, mussten die Präfixe für OWL, RDF und die aktuellen Ontologie-Zugriffspfade gesetzt werden. Die Properties spielen beim Abruf der Ontologie-Daten eine wichtige Rolle. Der Zugriff auf die Daten erfolgt über die Object-Properties und Data-Properties, die Teil von RDF(S) sind (Nazir et al., 2019, S. 551).

Die abschliessende Evaluation des Systems wurde durch drei Experten durchgeführt. Die Bewertungskriterien umfassten Genauigkeit, Zusammenhang (zwischen Frage und Antwort), Non-Sequitur (Logik der Antwort), Strukturiertheit und Kontextorientierung. Die Untersuchung zeigte, dass in den Bereichen Genauigkeit, Non-Sequitur und Strukturiertheit gute Ergebnisse erzielt wurden. Bei Folgefragen allerdings zeigte der Chatbot eine schlechte Leistung. Die Autor:innen führen dies auf den Mangel an natürlicher Sprachgenerierung zurück. In den Bereichen Zusammenhang und Kontextorientierung seien noch erhebliche Verbesserungen zu erzielen (Nazir et al., 2019, S. 551).

6.1.4 DBpedia Chatbot (2018)

Neue Mitglieder in Communities stellen immer wieder bereits beantwortete Fragen. Durch FAQs, Mailing-Listen und Foren wird versucht, dieses häufig auftretende Problem zu umgehen. Athreya et al. (2018) präsentieren anhand der DBpedia Community ihren Chatbot, der auf bereits beantworteten Fragen basiert. Die Architektur ist dabei auf einen modularen Ansatz ausgelegt. Den Nutzenden steht die Möglichkeit offen, ihre Anfragen via Text oder Buttons einzugeben. Der Chatbot antwortet schliesslich mittels kurzen Textnachrichten oder ausgefeilteren interaktiven Nachrichten. Der DBpedia Chatbot wurde auch im Hinblick auf die Anbindung unterschiedlicher Plattformen entwickelt. Kern des Systems ist das Spring-Framework (für Java) (Athreya et al., 2018, S. 143).

Für den Query-Lifecycle wurden verschiedene Wissensbasen miteinander verwoben. Der Prozess für die eintreffenden Anfragen umfasst

- a)** eine Webhook für jede eintreffende Anfrage, welche die eintreffenden Anfragen behandelt und zum Routing-Modul weiterleitet;
- b)** die Weiterleitung der Anfrage aufgrund des Anfragetyps. Die Texteingabe wird verarbeitet, parametrisiert und von einem Template-Handler bearbeitet. Bei reinen

Texteingaben nutzt der DBpedia Chatbot RiveScript⁹, um die Absichten der Nutzenden zu identifizieren und zu klassifizieren. Die Klassen umfassen einerseits regelbasierte DBpedia-Fragen im Zusammenhang mit der Community usw. Andererseits gibt es die Klasse der sachlichen Fragen zu Inhalten in DBpedia oder auch Ortsanfragen (beantwortet mit OpenStreetMap) oder sogenanntes «Banter», was eher einer Plauderei mit dem Chatbot entspricht und wenig mit seiner Hauptfunktion zu tun hat. Falls für die Plauderei keine passende Antwort gefunden wird, wird auf ELIZA zurückgegriffen.

Handelt es sich um parametrisierte Anfragen, so werden diese ausgelöst, sobald die Nutzenden auf einen Button klicken (zum Beispiel einen «learn more» Button);

- c) die Antwortgenerierung, bei der die Antwort in ein Format gewandelt wird, das der Sprache der ausgebenden Plattform (Slack, Facebook Messenger, Webseite) entspricht;
- d) das Senden der Antwort, welches die Antwort am Ende der jeweiligen Plattform zurückschickt (Athreya et al., 2018, S. 144, 146).

Wie bereits erwähnt, wird für die Klassifikation der Absichten RiveScript verwendet. Da der DBpedia Chatbot verschiedene Arten von Anfragen beantworten muss, ist dies ein wichtiger Schritt. Auf die Klassen von Fragen wird im Folgenden noch einmal näher eingegangen:

Community-Fragen

Der DBpedia-Handler ist auf Fragen rund um die DBpedia-Community spezialisiert. Die zugehörige Wissensbasis basiert auf verschiedenen Hintergrundkonversationen, worunter die Mailinglisten in DBpedia fallen, die für die Beantwortung via regelbasiertem Ansatz untersucht wurden. Interessante Frage-Antwort-Paare wurden für die Konversations-Szenarien aus diesen Mailinglisten entnommen. Auf die Details der Bearbeitung des Datensets wird an dieser Stelle nicht weiter eingegangen, da sie für diese Arbeit nicht relevant sind. Die ausführliche Beschreibung kann dem Artikel von Athreya et al. entnommen werden (Athreya et al., 2018, S. 144).

Sachliche Fragen

Für die Beantwortung von Fragen in natürlicher Sprache oder für Faktenwissen nutzt der DBpedia Chatbot eine Kombination aus Qanary von WDAqua und WolframAlpha.

⁹ RiveScript ist neben AIML und ChatScript eine weitere Skriptsprache für Chatbots: <https://www.rivescript.com/about>.

Qanary ist eine Methode, um Frage-Antwort-Systeme zu erstellen. WDAqua stellt das Framework zur Verfügung, das die Abfrage mehrerer RDF-Datenspeicher ermöglicht (WDAqua, 2020). Vorteil von Qanary ist seine Wiederverwendbarkeit in verschiedenen Domänen. Es ist auf bestehende Ontologien ausgerichtet und bietet einen praktischen Ansatz für die schnelle Implementierung neuer Frage-Antwort-Systeme (Both et al., 2016, S. 625). Im DBpedia Chatbot sind die Antworten von Qanary DBpedia-URIs oder RDF-Literale, während die Antworten von WolframAlpha mit DBpedia noch verknüpft werden müssen. Bei den Resultaten wird den RDF-Literalen mehr Gewicht beigemessen als den DBpedia URIs, da die RDF-Literale in der Regel genauere Antworten liefern. Trotzdem werden die Antworten aus beiden Systemen mit einer UNION-Strategie zusammengeführt, um sie den Nutzenden zu präsentieren (Athreya et al., 2018, S. 145).

Ausgabe der Antworten

Für die Präsentation der Ergebnisse reichert der DBpedia Chatbot die Antworten mit RDF-URIs an. Die Wissenskarten für die Nutzenden sind ähnlich gestaltet wie die Infoboxen von Wikipedia. Sie basieren auf dem zugrundeliegenden Wissensgraphen. Die anzuzeigenden Eigenschaften werden nach ihrer Relevanz sortiert. Um die User-Experience zu steigern, werden nach Beantwortung der Anfrage nützliche weitere Schritte mittels Buttons und Links angezeigt (Athreya et al., 2018, S. 145).

Der DBpedia Chatbot ist über folgende Webseite zugänglich: <https://chat.dbpedia.org/>.

6.1.5 Ontologiebasierter Chatbot für E-Commerce (2018)

Für den ontologiebasierten Chatbot im Bereich E-Commerce, nutzten Vegesna et al. (2018) die API der frei zugänglichen Ebay-Website. Mit den Daten, die mit der API gesammelt werden konnten, wurde in Protégé eine Ontologie erstellt. Die Ontologie-Vorlage ruft die Datenquellen unter Verwendung von JAPE-Regeln¹⁰ ab. Stehen die Daten vollständig in der Vorlage zur Verfügung, ist die Befragung des Chatbots möglich. Die Absichten der Nutzenden werden mit NLP-Methoden in Wit.ai strukturiert. Auf Grundlage der Absicht wird anschliessend eine Funktion in Python aufgerufen, ebenso wird die API der Ontologie-Vorlage abgerufen, um die Antwort an die Nutzenden auszugeben (Vegesna et al., 2018, S. 51–52).

¹⁰ JAPE steht für Java Annotation Patterns Engine. Eine Einführung bietet die folgende Seite: <https://gate.ac.uk/sale/tao/index.html#x1-2040008>.

6.1.6 OntBot (2011)

Al-Zubaide & Issa (2011) präsentieren einen Ansatz für eine Mapping-Technologie, bei der Ontologien in relationale Datenbanken transformiert werden. Der sogenannten OntBot nutzt dann das Wissen aus der relationalen Datenbank, um seine Antworten zu erstellen. Eine manuelle Erstellung der kompletten Wissensbasis ist so nicht nötig. Für die Speicherung der Daten aus der Ontologie in einer relationalen Datenbank haben sich die Autoren entschieden, weil dies die beste Möglichkeit bietet Daten zu speichern, zu aktualisieren oder zu manipulieren. Auch der Umgang mit grossen Datenmengen ist in einer relationalen Datenbank gewährleistet. Hinzu kommt, dass Eigenschaften wie Transaktionsmanagement, Sicherheit und Integritätskontrolle gewährleistet sind, was ein zusätzlicher Vorteil gegenüber traditionellen Dateisystemen darstellt (Al-Zubaide & Issa, 2011, S. 7).

Im Jahr 2011 war eine Reihe von Techniken bekannt, die sich entweder für die Abbildung von Ontologien in relationalen Datenbanken oder umgekehrt anboten. Die Transformation der Ontologie in eine relationale Datenbank basiert auf einer Reihe von Zuordnungsregeln. Die Konstrukte des ontologischen Modells müssen auf die Konstrukte der relationalen Datenbank übertragen werden. Zum Beispiel stellt eine Klasse in OWL eine relationale Tabelle dar. Die Eigenschaften einer Klasse werden als Attribute in der Tabelle erfasst und die Vererbungsbeziehungen zwischen den Klassen werden durch Fremdschlüssel zwischen den Tabellen umgesetzt (Al-Zubaide & Issa, 2011, S. 8–9).

Für die Verarbeitung der eingegangenen Fragen werden, wie bei anderen Chatbots auch, NLP-Techniken angewendet. Herzstück von OntBot bildet die Inferenz-Engine. Die Autoren nennen sie das «Gehirn des Chatbots». Hier wird die normalisierte Anfrage spezifiziert und falls die Eingabe der Nutzenden ausserhalb des Themenbereichs liegt, wird OntBot einen Hinweis über die abgedeckte Domäne liefern. Es ist zu erwähnen, dass hinter OntBot kein AIML steht. Al-Zubaide & Issa schlagen stattdessen vor, das Programm mit VB.Net oder einer anderen Programmiersprache zu schreiben. Ebenfalls in der Inferenz-Engine angesiedelt, ist die Suche nach einer passenden Regel für die Beantwortung der Anfrage. OntBot behandelt verschiedene Arten von Anfragen, die auf einfache oder komplexe Weise (zum Beispiel Abfrage mehrerer Tabellen) zu bearbeiten sind. Wurde die passende Regel identifiziert, wird die Anfrage im Query-Processor tatsächlich an die Wissensbasis von OntBot gestellt. Ausgegeben wird die Antwort in einem benutzungsfreundlichen Format (Al-Zubaide & Issa, 2011, S. 8, 10–11).

Eine der grössten Herausforderungen sei gemäss Al-Zubaide & Issa die Datenabfrage, da es an einfach zu bedienenden Schnittstellen mangle. Um Ontologie-Repositoryen

nutzen zu können, ist Wissen über den Inhalt der Ontologien notwendig. Ebenso müssen OWL (Web Ontology Language), RDF (Resource Description Framework) und SPARQL bekannt sein und eingesetzt werden können. Durch den vorgeschlagenen Ansatz können Nutzende mit OntBot über natürliche Sprache interagieren, sodass das Erlernen einer Abfragesprache nicht mehr notwendig ist. Zusätzlich wird eine einfach zu bedienende Nutzungsoberfläche zur Verfügung gestellt (Al-Zubaide & Issa, 2011, S. 7).

Al-Zubaide & Issa erwähnen, dass eine prototypische Umsetzung ihres Ansatzes stattfindet. Allerdings konnte kein Nachweis gefunden, dass das Forschungsprojekt weiterentwickelt wurde.

6.1.7 Project CyN (2004)

Für das CyN-Projekt wurde der Pattern-Matching-Interpreter von ALICE (genannt Program N) mit OpenCyc verlinkt. OpenCyc war die Open-Source-Version von Cyc, einer grossen Common-Sense-Wissensbasis. Die Integration der Ontologie ermöglicht es dem Chatbot, auf eine grosse Menge an Wissen zuzugreifen, die in Cyc verfügbar ist. Die in AIML enthaltenen «categories» werden durch die Information in OpenCyc verallgemeinert (Augello et al., 2014, S. 287). Entwickelt wurde diese Lösung von Daxtron Laboratories. Ein Ziel des Systems war es, einen Parser zur Verfügung zu stellen, der für Pattern-Matching-Verfahren in AIML-Chatbots funktioniert und OpenCyc integriert. Geeignet ist OpenCyc für Grundwissen. Darüber hinaus wurde ein Set von AIML-Kategorien als Beispiel erstellt, um darzustellen, wie Antworten mithilfe von OpenCyc beantwortet werden können. Das Tool, das mit C++ geschrieben wurde, ist offen und kann noch immer frei heruntergeladen, genutzt und verändert werden. Es handelt sich bei CyN allerdings nicht um einen fertigen Chatbot – ein vollständiges AIML-Set ist notwendig (Daxtron Laboratories, 2010). Hinzu kommt, dass die Ontologie OpenCyc gemäss Wikipedia (siehe [Cyc](#)) seit 2017 nicht mehr verfügbar ist.

6.1.8 CyD-Technologie in MAGA (2014) und FRASI (2012)

Der Name für die CyD-Technologie setzt sich zusammen aus «Cyc» und «ProgramD» und ist angelehnt an das CyN-Projekt. Augello et al. (2014) nutzten das Tool aus dem CyN-Projekt und passten es für ihre Zwecke an. Als Vorteil von CyN erwies sich seine leichte Erweiterbarkeit. Es wurde an den AIML-Interpreter angepasst, der in Java ProgramD geschrieben wurde. Für Augello et al. ergaben sich aus der Verwendung der Java-Programmiersprache Vorteile – unter anderem war es ihnen so möglich die verfügbare Java-API für die Cyc-Wissensbasis zu nutzen. Die AIML-Sprache haben sie

mit neuen Tags angereichert, damit der Chatbot die ResearchCyc¹¹-Ontologie für Common-Sense-Reasoning und Inferenzschlüsse nutzt. Der Chatbot fragt die Ontologie direkt aufgrund seiner Regeln in der Wissensbasis ab (Augello et al., 2014, S. 288).

Die Antworten, welche aus der Cyc-Ontologie generiert werden, werden in einen natürlichsprachigen Satz eingebettet. Als Anschauungsbeispiel wird folgender Auszug aus dem AIML-File herangezogen, den Augello et al. (2014, S. 289) in ihrem Artikel präsentieren:

```
<category>
  <pattern>IS * A *</pattern>
  <template>
    <cyssystem>
      (cyc-query '
        (#$genls
          <cycterm><star index="1"/></cycterm>
          <cycterm><star index="2"/></cycterm>
        )#$EverythingPSC)
    </cyssystem>!!
  </template>
</category>
```

Die extra für dieses Projekt erstellten AIML-Tags (im Beispiel **blau hervorgehoben**) befinden sich innerhalb des Template-Tags. Sie sind damit eingebettet in die Vorgaben für die Antwortgenerierung. Nach dem Tag `<cyssystem>` folgt die Query-Anfrage für die Cyc-Ontologie. Die Prädikate *genls* («generalization», in vorliegendem Beispiel ersichtlich) und *isa* («is a») sind die zwei gebräuchlichsten innerhalb von Cyc. Sie bezeichnen Hyponymie- beziehungsweise Hyperonymie-Beziehungen (Cox, 2005).

Aufgrund der Integration von Cyc wird das Schreiben von Kategorien vereinfacht, da generische Standardkategorien für allgemeine Themen erstellt werden können. Durch die Platzhaltersymbole (`<star index="1"/>`) wird in der Cyc-Wissensbasis nach dem eingegebenen Begriff der Nutzenden gesucht. Ein weiterer Vorteil besteht gemäss den Autor:innen darin, dass Laien die Ontologie auf einfache Weise abfragen und erweitern können (Augello et al., 2014, S. 289).

¹¹ ResearchCyc war ebenfalls eine Version von Cyc. Sie war für Forschungszwecke bestimmt.

Anwendung der CyD-Technologie in Auskunftssystemen

Die CyD-Technologie wurde im Projekt MAGA (Mobile Archeological Guide at Agrigento) eingesetzt, das im Bereich des kulturellen Erbes angesiedelt ist. Nutzende erhielten dabei die Möglichkeit, sich via mobile Geräte über Kulturstätte zu informieren. Die Abfrage des Systems erfolgte über gesprochene Sprache (Augello et al., 2014, S. 290).

Aufgrund der grossen Menge an Daten in der Cyc-Ontologie lässt sich die Technologie auf weitere Domänen übertragen. So wurde sie zum Beispiel auch implementiert für die Abfrage von Dienstleistungen auf einem Universitätscampus oder zur Unterstützung von Nutzenden bei ihren Einkaufsaktivitäten (Augello et al., 2014, S. 290–291).

Auch im Projekt FRASI (Framework for Agent-based Semantic-aware Interoperability) wurde die CyD-Technologie umgesetzt. Bei FRASI handelt es sich um ein Industrieprojekt im Bereich der Milchverarbeitung, bei dem eine Möglichkeit gefunden werden sollte Kundenanfragen zu bearbeiten. Augello et al. (2012) entschieden sich auch bei FRASI für die Methode des Pattern-Matchings und die Implementierung von ALICE, da ALICE einfach umzusetzen ist und auf Open-Source-Technologie basiert. Eine grosse Einschränkung von ALICE ist allerdings die starre Wissensbasis, die in Frage-Antwort-Paare organisiert ist und deren Regeln nicht veränderlich sind (Augello et al., 2012, S. 186–187). Erweitert wurde ALICE im FRASI-Projekt mit den Modulen *Enhanced Symbolic Reduction (E-SRAI)*, dem *Bootstrapping-Modul* und dem *CyD-Modul*.

Die Hauptaufgabe des Moduls *Enhanced Symbolic Reduction (E-SRAI)* ist es, die Anfragen der Nutzenden zu verstehen. Dazu prozessiert es die Eingabe der Nutzenden und reduziert sie auf eine simplere Struktur. Mittels syntaktischer, semantischer und pragmatischer Regeln werden bestimmte Variablen extrahiert. Ebenfalls Teil des E-SRAI-Moduls ist die Analyse der Fragen. Dieser Prozess besteht darin, den Text in Sätze zu zerlegen und anschliessend die Wortarten im Satz zu erkennen (grammatikalische Analyse). Eine logische Analyse entscheidet, welche Rollen die Begriffe eines Satzes spielen (Augello et al., 2012, S. 189).

Das *Bootstrapping-Modul* wird verwendet, um die Wissensbasis automatisch zu erweitern. Der im Hintergrund durchgeführte Prozess umfasst die automatische Erstellung von Kategorien für typische domänenorientierte Fragen. Ziel dabei ist es das Füllen der AIML-Wissensbasis zu vereinfachen (Augello et al., 2012, S. 191). Die AIML-Wissensbasis in FRASI umfasst vier Typen von Kategorien:

- Atomic sind spezifische Kategorien, die definiert wurden, um einen vorgegebenen Satz von Fragen vorwegzunehmen. Die Antworten in dieser Kategorie können

manuell vom Entwicklungsteam erstellt werden oder durch Abfragen einer Ontologie automatisch generiert werden (im AIML Bootstrap Module).

- Default enthalten Platzhaltersymbole im <pattern> und Standard-AIML-Tags im <template>.
- Ontology based Kategorien enthalten bestimmte Tags, um mit der Ontologie zu kommunizieren. Das <pattern> kann auch Platzhaltersymbole enthalten, um einen Abgleich mit der Ontologie herbeizuführen.
- Bei Ultimate default finden sich ausschliesslich Platzhalter im <pattern>. Um diese Anfragen zu bearbeiten, vollzieht das Modul E-SRAI das <template> (Augello et al., 2012, S. 188).

Kombination der CyD-Technologie mit dem LSAbot

LSA steht für «Latent Semantic Analysis». Gemäss Augello et al. (2014, S. 293–294) kann die LSA-basierte Methode dabei helfen, Chatbots mit intuitiven und assoziativen Fähigkeiten auszustatten. Durch LSA lässt sich die kontextuelle Bedeutung von Wörtern durch statistische Berechnungen extrahieren und darstellen, wenn die Methode auf grosse Textkorpora angewendet wird (Landauer et al., 1998). Entwickelt wurde LSAbot im Jahr 2004 von Salvatore La Bua (La Bua, 2021). Augello et al. (2014) nutzten ebenfalls LSA für die Umsetzung eines Chatbots. Als erstes erstellten sie eine AIML-Wissensbasis, die mit einem natürlichsprachigen Korpus erweitert wurde. Mithilfe der LSA-Methode wurde die Wissensbasis codiert, sodass dem Chatbot eine Assoziationsfähigkeit ermöglicht wurde. Die codierte Nutzendenanfrage wurde dazu mit dem codierten Wissensselement in der Wissensbasis des Chatbots verglichen (Augello et al., 2014, S. 294).

Augello et al. (2014, S. 294) integrierten die CyD-Technologie (siehe 6.1.8 CyD-Technologie in MAGA (2014) und FRASI (2012)) schliesslich im LSABot, um die beiden vorgestellten Technologien miteinander zu kombinieren. Der neue Chatbot soll demnach in der Lage sein, sowohl die rationale Ebene (mittels Zugriff auf die Cyc-Ontologie) als auch die intuitive Ebene (mithilfe des semantischen Raums und der LSA-Methode) abzudecken (Augello et al., 2014, S. 294–298).

6.1.9 Program W in EHeBby (2008)

In Program W ist ein AIML-Interpreter, der in Java geschrieben wurde. Durch ihn ist es möglich mittels AIML-Tags WordNet abzufragen (Caronia et al., o. J.). WordNet ist eine frei verfügbare, englischsprachige lexikalische Datenbank. Verschiedene Wortarten sind darin durch semantische und lexikalische Beziehungen miteinander verknüpft (Princeton

University, 2021). Auch in diesem Projekt wurden neue AIML-Tags erstellt, welche die direkte Abfrage von WordNet mittels WordNet-Regeln ermöglicht. Ziel dieser Technologie ist es die Sprachanalysefähigkeit des Chatbots zu verbessern, indem beispielsweise Begriffe durch ihre Lemmata oder bestehenden Beziehungen ausgewertet werden können (Augello et al., 2014, S. 292). Eine Anwendung des Programms fand sich im Chatbot EHeBby. Ziel war es EHeBby Sinn für Humor zu verleihen, indem er selbständig humorvolle Sätze generiert, aber auch lustige Sätze von Nutzenden erkennt, wobei der Fokus auf der Erkennung humoristischer Inhalte lag. Der Chatbot wurde mittels des Frameworks von ALICE entwickelt und in den Yahoo!-Messenger eingebettet. Für die Erkennung humoristischer Inhalte wurde unter anderem WordNet einbezogen (Pilato et al., 2008).

6.2 Vorteile von semantischen Technologien in Chatbots

In diesem Abschnitt wird der Frage nachgegangen, inwiefern semantische Technologien die Erstellung von Chatbots unterstützen, beziehungsweise welche Vorteile semantische Technologien für das Funktionieren eines Chatbots bringen. Dazu werden als erstes mögliche Evaluationsmethoden für Chatbots betrachtet. Zwar haben einzelne der in 6.1 Chatbots mit semantischen Technologien aufgelisteten Projekte selbständig ihren Chatbot evaluiert und die Ergebnisse in den Publikationen aufgelistet (darunter Ait-Mlouk & Jiang, 2020; Kondylakis et al., 2020; Nazir et al., 2019), allerdings haben nicht alle ihre Chatbots eigenständig überprüft. Casas et al. (2020) haben in einer Meta-Analyse zur Chatbot-Entwicklung und -Evaluation ebenfalls festgestellt, dass Chatbots zum Teil gar nicht oder nicht vollständig evaluiert wurden. Sie begründen dies mit dem enormen Arbeitsaufwand, der für die Entwicklung eines Chatbots erforderlich ist. Neue Technologien auszuprobieren und gleichzeitig die Benutzungsfreundlichkeit zu testen, erfordert viel. Auch wird der Fokus in Forschungsprojekten auf bestimmte Fortschritte eingeschränkt, sodass nicht alle Aspekte eines Chatbots berücksichtigt werden. Zudem sind für Entwicklung und Testing eines Produkts oft unterschiedliche Kompetenzen notwendig und auch finanzielle Einschränkungen können eine Rolle spielen (Casas et al., 2020, S. 283).

Anhand wissenschaftlicher Publikationen untersuchten Casas et al. (2020) die eingesetzten Methoden und Techniken in der Chatbot-Evaluierung zwischen den Jahren 2016 und 2020. Sie stellten dabei fest, dass kein standardisiertes Verfahren existiert, um die Qualität von Chatbots zu messen. Viele Gruppen und Individuen haben selbständig Standards entwickelt oder sich von anderen inspirieren lassen, um jeweils den eigenen

Chatbot zu beurteilen. In ihrer Review trugen Casas et al. die Methoden zusammen und betrachteten diese anhand der ISO-Norm für Gebrauchstauglichkeit (ISO 9214). Gemäss dieser ISO-Norm gelten *Effektivität* (in der Aufgabenlösung), *Effizienz* (bei der Aufgabenbearbeitung) und *Zufriedenstellung* (der Nutzenden) als Mass für die Gebrauchstauglichkeit. Identifiziert haben Casas et al. (2020) einen Trend in Richtung Evaluation der Effizienz. Sie führen dies darauf zurück, dass im geschäftlichen Kontext immer mehr aufgabenorientierte Chatbots entwickelt werden (Casas et al., 2020, S. 280). Hinsichtlich der Chatbot-Kategorien differenzierten Casas et al. (2020) ausschliesslich zwischen aufgabenorientierten und dialogorientierten Chatbots¹² – dies aus folgendem Grund: Bei den aufgabenorientierten Chatbots spielt vor allem die erfolgreiche Ausführung der Aufgabe eine Rolle. Mit zunehmend schneller Aufgabenerledigung steigt die Erfolgsquote. Für dialogorientierte Chatbots hingegen ist nicht das Ergebnis ausschlaggebend, sondern die Unterhaltung an sich (Casas et al., 2020, S. 281).

In ihrer Meta-Analyse fassen Casas et al. (2020) die Evaluations-Merkmale in einer Tabelle zusammen. Folgende Tabelle ist aus der Studie von Casas et al. abgeleitet, jedoch wurde sie durch zusätzliche Erklärungen zu den einzelnen Merkmalen erweitert. Die Studiennummer zeigt an, aus welcher Quelle die Merkmale stammen. Es handelt sich um die folgenden Publikationen:

1. Radziwill & Benton (2017)
2. Maroengsit et al. (2019)
3. Jadeja & Varia (2017)
4. Venkatesh et al. (2018)

Abschliessend ist festzuhalten, dass es nicht eine Evaluationsmethode gibt, die bedeutend häufiger in den wissenschaftlichen Publikationen vertreten war. Die Methoden werden in der Praxis je nach Chatbot-Kategorie ausgewählt. So wird bei einem dialogorientierten Chatbot zum Beispiel mehr Wert auf die Verarbeitung natürlicher Sprache gelegt, während ein aufgabenorientierter Chatbot eher anhand seiner Problemlösungsfähigkeit beurteilt wird (Casas et al., 2020, S. 283). Maroengsit et al. (2019) erwähnen, dass für die Wahl einer Evaluationsmethode der Chatbot-Typ, seine Domäne, sowie Ziele und Zweck der Evaluierung ausschlaggebend sind (Maroengsit et al., 2019, S. 117).

¹² Entspricht den in 2.4.1 angesprochenen Kategorien zielorientiert und nicht-zielorientiert.

Kriterien nach ISO 9214	Studie Nr.	Merkmale ausgehend von der Meta-Analyse
Effektivität	1	Funktionalität: betrifft Eigenschaften wie präzise Sprachsynthese, präzise Interpretation der Befehle, sprachliche Genauigkeit, allgemeine Benutzungsfreundlichkeit, Problemlösung. Menschlichkeit: umfasst Durchschaubarkeit, Einbezug von Fehlern zur Erhöhung der Realitätsnähe, Fähigkeit bestimmte Fragen zu beantworten (zielt darauf ab, den Turing-Test zu bestehen).
	2	Inhaltliche Evaluation: automatisch oder durch Experten.
	3	Perspektive des Information-Retrievals: misst zum Beispiel, wie genau und schnell der Chatbot auf die Anfrage von Nutzenden antwortet.
	4	Anwendungsbereich: wie gut wird die Domäne abgedeckt?
Effizienz	1	Performance: bezieht sich auf die Leistung des Chatbots im Hinblick auf sein Ziel. Darunter können die Robustheit gegenüber Manipulationen oder unerwarteten Eingaben fallen, aber auch die effektive Zuweisung von Funktionen oder die Vermeidung unangemessener Äusserungen.
	2	Funktionale Evaluation: umfasst aufgabenorientierte Evaluation (beliebt bei aufgabenorientierten Chatbots) und Nutzungsstatistiken.
	3	Linguistische Perspektive: misst, ob der Chatbot gut in Qualität, Quantität, Beziehung und Umgangsform abschneidet. Perspektive der künstlichen Intelligenz: testet, ob der Chatbot den Turing-Test bestehen kann.
	4	Kohärenz, Gesprächstiefe und thematische Vielfalt.
Zufriedenstellung	1	Wirkung: beinhaltet die Begrüssung und die Vermittlung von Persönlichkeit, fragt aber auch danach, ob der Chatbot das Gespräch anregt, emotionale Information vermittelt (z. Bsp. Tonfall), Wärme und Authentizität zeigt, Aufgaben angenehmer und interessanter macht oder die Stimmung der Nutzenden liest. Ethik und Verhalten: beinhaltet unter anderem Respekt, Inklusion und Wahrung der Würde, Schützen und Respektieren der Privatsphäre, Sensibilität gegenüber Sicherheit, Vertrauenswürdigkeit sowie Bewusstsein für Trends und sozialen Kontext. Accessibility: stellt die Frage, ob der Chatbot die Absicht erkennt, ob er auf soziale Hinweise und unterschiedliche Bedürfnisse reagieren kann.
	2	Zufriedenheit der Nutzenden: fragt Nutzende direkt, was sie von Chatbot halten.
	3	Perspektive der Nutzenden: Zufriedenheit der Nutzenden, Gebrauchstauglichkeit usw.
	4	Dialogorientierte Benutzungserfahrung.

Tabelle 3: Eigene Zusammenfassung der Ergebnisse aus der Studie von Casas et al. (Casas et al., 2020, S. 281–282).

Resultate von Evaluationen verschiedener Chatbots wurden noch nicht miteinander verglichen, weshalb keine generelle Aussage darüber getroffen werden kann, ob beispielsweise eine bestimmte Technologie besonders gut funktionierende Chatbots erwirkt. Auch hier wirkt sich das Fehlen eines generellen Frameworks für die Evaluierung von Chatbots negativ aus. Maroengsit et al. (2019) bringen die Idee auf, die Qualität von Chatbots auch anhand ihrer Datenquelle (Datenbank oder Ontologie) zu untersuchen, allerdings wurde dies noch nicht gemacht (Maroengsit et al., 2019, S. 118).

Da die oben aufgelisteten Evaluationsmethoden nicht für die in 6.1 Chatbots mit semantischen Technologien erwähnten Projekte angewendet werden können (da nur einer der Chatbots zugänglich ist), müssen für die Beantwortung der Forschungsfrage andere Bewertungskriterien herangezogen werden. Vishwakarma & Pandey (2021) haben einen Überblick über verschiedene Design-Techniken für Chatbots erstellt und diese miteinander verglichen. Bei der Auswahl der Chatbots fokussierten sie sich auf die hauptsächlich verwendeten Methoden. Fokussiert haben sie dabei auf Chatbots, welche der Kategorie der generativ-basierten Modelle angehören und die maschinelles Lernen einsetzen. Besonders interessant für die vorliegende Arbeit sind die häufig auftretenden Mängel bei Chatbots, welche Vishwakarma & Pandey in ihrer Arbeit zusammengetragen haben (Vishwakarma & Pandey, 2021, S. 72, 77).

- Streng regelbasiertes System: Bei früheren Chatbots wurden feste Regelsätze sowie ein einfacher vorlagenbasierter Abgleich eingesetzt. Maschinelle Lernmethoden basierten auf einfachen Ansätzen.
- Grammatikalische Fehler: Diese werden nicht gespeichert, weshalb Erinnerung und Verbesserung ausbleiben.
- Vordefinierte oder geschlossene Domäne: Einige Chatbots können nur vorher festgelegte Datenbankabfragen übernehmen.
- Zweideutigkeit: Zweck und Bedeutung von Wörtern in Sätzen sind nicht eindeutig.
- Struktur der Sprache: Jede Sprache weist eine andere Satzstruktur auf. Auch Zeichensetzung sowie die Verwendung von Leerzeichen variiert. Viele Chatbots sind nicht in der Lage unterschiedliche Sprachen voneinander zu unterscheiden.
- Semantik: Einige Chatbots kommen bei der Eingabeverarbeitung und Antwortgenerierung nicht mit der natürlichen Sprache zurecht. Die Bedeutung der menschlichen Sprache wird vom Chatbot nicht verstanden.
- Analyse der Stimmung: Chatbots sind nicht in der Lage die Gefühlslage der Nutzenden zu erkennen.

- Vorschläge aus dem System: Einige Chatbots stellen keine Fragen zum Thema der Nutzenden, klären nicht auf und geben keine Empfehlungen ab. Sie holen Information aus der Wissensdatenbank und lernen nicht auf Grundlage früherer Antworten.
- Genauigkeit: Einige Chatbots haben eine schwache Neigung, das Thema plötzlich zu wechseln und eine Antwort darauf zu geben. Dies ist auch darauf zurückzuführen, dass sie oft ohne Bedeutung interagieren. Die Genauigkeit des Chatbots leidet darunter (Vishwakarma & Pandey, 2021, S. 77).

Vishwakarma & Pandey (2021) schlagen vor, diese Mängel mittels Deep-Learning-Techniken zu überwinden. Der Chatbot wäre dadurch in der Lage, angemessene Antworten zu produzieren und die Absicht der Nutzenden besser auszuwerten (Vishwakarma & Pandey, 2021, S. 77). Deep-Learning ist ein vielversprechender Ansatz. Ausser Acht gelassen wird bei dieser Argumentation aber, dass auch weitere Techniken existieren, um manche der Mängel zu beheben. Aufgrund der Erkenntnisse aus 6.1 Chatbots mit semantischen Technologien wird belegt, dass gewisse Mängel mithilfe von semantischen Technologien teilweise beseitigt werden können. Auf Lösungsvorschläge für einzelne Defizite wird im Folgenden eingegangen.

➤ Vordefinierte/geschlossene Domäne

Chatbots können so konzipiert werden, dass sie auf mehrere Wissensbasen/Ontologien zugreifen können. Anschaulich wurde das bei KBot demonstriert, der auf mehrere Ontologien zugreifen kann, womit die Themenvielfalt in der Konversation vergrössert wird. Zwar ist die Wissensdomäne durch die angehängte Ontologie weiterhin beschränkt und der Chatbot gehört dadurch immer noch zur Kategorie Closed-Domain. Allerdings helfen Semantic-Web-Anwendungen dabei, die Grenzen von Closed-Domain-Chatbots auszuweiten.

➤ Zweideutigkeit, Genauigkeit und Semantik

Diese drei Mängel wurden zusammengefasst, da sie alle mit der Bedeutung von Wörtern/Sätzen zu tun haben. Semantische Technologien sind eigentlich dazu prädestiniert, Beziehungen zwischen Wörtern zu bezeichnen. Durch die relationalen und hierarchischen Verbindungen in Ontologien und die Regeln zur Schlussfolgerung ist es möglich, die Bedeutung von Wörtern zu verdeutlichen. Durch Inferenzschlüsse kann neues Wissen generiert werden. Allerdings ist bei den gefundenen Chatbots mit Semantic-Web-Anwendungen kein Hinweis zu finden, dass durch den Einbezug von Ontologien die Bedeutung von Wörtern besser hätte identifiziert werden können. Nazir et al. (2019) haben durchaus versucht, mit ihrem ontologiebasierten Chatbot

kontextorientierte Antworten zu erzeugen. Die Ergebnisse ihrer Evaluation waren in diesem Bereich aber nur wenig über dem Durchschnitt (Nazir et al., 2019, S. 551). Eine Möglichkeit für den Einbezug der kontextuellen Bedeutung scheint die LSA-Methode, wie sie Augello et al. (2014) in FRASI eingesetzt haben. Allerdings handelt es sich dabei nicht um eine semantische Technologie. Betreffend Genauigkeit und plötzlichem Wechsel des Themas in einer Unterhaltung, kann der Einbezug verschiedener Wissensbasen durchaus den Effekt erzeugen, dass zwischen verschiedenen Themenbereichen gewechselt werden kann. Alle Chatbots, die auf DBpedia zugreifen, sind in der Lage über unterschiedlichste Themen Auskunft geben zu können.

➤ Analyse der Stimmung

Der Chatbot EHeBby zeigt, dass semantische Anwendungen (im vorliegenden Fall WordNet) durchaus genutzt werden können, um Stimmungen zu erkennen. Bei EHeBby ging es um die Identifikation humoristischer Inhalte. Die Evaluation von EHeBby hat ergeben, dass der Chatbot in der Lage war 66% der lustigen Sätze und 81% der nicht-lustigen Sätze zu erkennen. Zwar wurde in diesem Projekt ebenfalls mit LSA gearbeitet, allerdings stellte die Kombination mit WordNet einen Mehrwert dar, um das humoristische Level eines Satzes verstehen zu können (Pilato et al., 2008). Was für humoristische Inhalte funktioniert, kann auch auf andere Gefühlslagen übertragen werden.

Der Einsatz von semantischen Technologien vermag nicht alle Mängel aufzulösen, vereinzelt lassen sich aber durch den Einbezug von Ontologien Verbesserungen erzielen. Techniken des Semantic-Web bringen also Vorteile bei der Chatbot-Entwicklung. Im Idealfall werden semantische Technologien mit anderen Technologien kombiniert, um einen zufriedenstellenden Chatbot zu erstellen.

6.3 Semantische Technologien für den Bibliotheks-Chatbot Viola

In diesem Unterkapitel wird der Chatbot Viola der FHGR Bibliothek als Beispiel genommen, um den Einsatz semantischer Technologien im Bibliotheksbereich zu betrachten.

6.3.1 Der Chatbot «Viola» der FHGR Bibliothek

Die Virtuelle Online-Assistentin (VIOLA) entstand im Jahr 2009 im Rahmen der Bachelorarbeit von Renate Gygax an der Hochschule für Technik und Wirtschaft (HTW) Chur (Gygax, 2009). Der Chatbot beantwortet Fragen zur Nutzung der FHGR Bibliothek

und läuft derzeit über die Legacy AIML 1.0 Plattform von Pandorabots. Dementsprechend ist er auch in AIML 1.0 geschrieben. Violas Aussensicht wurde mit HTML editiert. Die Erstellung von Viola bedeutete einen grossen Arbeitsaufwand. Zumindest die Gesprächsteile für Small-Talk konnten bei der Implementierung von der Kornhausbibliothek mit kleinen Änderungen übernommen werden. Heute wird Viola von Marcel Hanselmann, einem Mitarbeiter der FHGR Bibliothek, weiterhin gepflegt. Neben der Aktualisierung der AIML-Files umfasst die Pflege die Kontrolle und Abarbeitung der Log-Dateien, welche vergangene Konversationen mit Nutzenden und nicht beantwortbare Fragen anzeigen (M. Hanselmann, persönliche Kommunikation, 29. Juni 2021).

Viola soll den Informationsbedarf der Nutzenden im Bereich der Standardanfragen decken. Ihre Hauptfunktion besteht darin, Studierenden der FHGR (insbesondere neuen Studierenden) so schnell wie möglich Auskunft zu geben. Neben der Hauptzielgruppe der Studierenden gibt Viola auch Dozierenden, Mitarbeitenden und Externen Information weiter (Gygax, 2009, S. 56, 58).

Der Zweck von Viola könnte besser erfüllt werden, wenn sie direkt Auskunft über die Bestände der Bibliothek geben könnte. Es wird deshalb vorgeschlagen, sie dahingehend zu erweitern, dass sie auf die bibliografischen Daten der Bibliothek zugreifen und diesbezügliche Angaben an die Nutzenden weitergeben kann.

6.3.2 Mögliche Übertragung von Semantic-Web-Anwendungen auf Viola

Es drängt sich sofort die Frage auf, welche weitere Wissensquelle für den Chatbot Viola in Frage kommt. Für die Suche nach Ontologien stehen verschiedene Quellen zur Verfügung: i.) Im Linked Open Data Vocabulary (LOV)¹³ sind derzeit 760 Vokabulare gelistet. Durch die Visualisierung der Vokabulare auf der Website des LOV lässt sich direkt auf deren Grösse schliessen. Die Suchfunktion ermöglicht auch eine Einschränkung auf deutsche Vokabulare, was im Fall von Viola nötig ist. Da der Chatbot in deutscher Sprache interagiert, muss auch seine erweiternde Wissensbasis auf Deutsch verfügbar sein. ii.) Auf opendata.swiss finden sich frei zugängliche Daten von Schweizer Behörden. Darunter zu finden ist beispielsweise auch der Eintrag zu Linked Swissbib (mehr zu Linked Swissbib weiter unten). Betrieben wird die Plattform vom Bundesamt für Statistik. iii.) Über den Dienst Linked Data Service (LINDAS) des

¹³ <https://lov.linkeddata.es/dataset/lov/>

Bundesamts für Umwelt stellen Schweizer Behörden ihre Daten als Linked-Data zur Verfügung. Noch befinden sich im Datenkatalog erst acht Datensets.

Ausgehend von diesen Quellen wurden folgende Wissensbasen gefunden, die für Viola in Frage kommen könnten. Es folgt eine Abwägung zur Eignung dieser Wissensbasen:

- Linked Swissbib wäre prädestiniert gewesen für eine Anbindung an einen Bibliotheks-Chatbot. Bei Swissbib handelte es sich um einen Katalog, an den die meisten schweizerischen Hochschulbibliotheken und weitere Institutionen angeschlossen waren. In Linked Swissbib wurden die Metadaten der Dokumente in den Bibliotheken in ein semantisches Format basierend auf RDF umgewandelt. Darüber hinaus bestand eine Verknüpfung mit VIAF¹⁴, GND¹⁵ und DBpedia. Aufgrund der Lancierung der Swiss Library Services Platform (SLSP) Ende 2020 hat Swissbib kurze Zeit später den Betrieb eingestellt. Die letzte Aktualisierung von Linked Swissbib fand im Juli 2017 statt (opendata.swiss, 2017).
- Der Linked-Data-Service der Deutschen Nationalbibliothek bietet ebenfalls die GND, aber auch alle Daten der deutschsprachigen Nationalbibliografie als RDF an (Deutsche Nationalbibliothek, 2019). Ein SPARQL-Endpoint ist zurzeit nicht verfügbar, aber in Planung (Eichenauer et al., 2018, S. 6). Bei diesem Service ist zu beachten, dass deutschsprachige Dokumente nur ein Teil des Bestandes der FHGR Bibliothek abdecken. Ein tatsächlicher Nachweis des Bibliotheks-Bestandes im Chatbot kann damit nicht erbracht werden.
- DBpedia wird auch auf Deutsch angeboten. Durch die Einbindung von DBpedia könnte Viola direkt mit Faktenwissen weiterhelfen und Fragen beantworten wie zum Beispiel: «Was bedeutet Städtebau?». Bei der Einbindung von DBpedia würde es sich allerdings um eine Erweiterung der bisherigen Funktion von Viola handeln.

Unabhängig davon, welche der Ontologien genutzt wird, ist eine Technik für den Zugang zu wählen. Hierzu werden die Projekte aus 6.1 Chatbots mit semantischen Technologien noch einmal kurz betrachtet. Beim Chatbot *KBot* von Ait-Mlouk & Jiang (2020) wurden verschiedene Techniken miteinander vereint. Wie die SPARQL-Abfrage genau ausgelöst wird, lässt sich aus dem Text nicht erkennen. Eine weiterführende Recherche hat keine neueren Quellen zutage gebracht. So kann auch nicht auf ein Repository mit dem Programm/Quellcode zugegriffen werden.

¹⁴ VIAF steht für Virtual International Authority File und vereint mehrere Normdateien weltweit: <http://viaf.org/>. Gehostet wird der Service von OCLC.

¹⁵ Die GND ist die Gemeinsame Normdatei, die von der Deutschen Nationalbibliothek und deutschsprachigen Bibliotheksverbänden verwaltet wird: <https://www.dnb.de/gnd>.

Auch bei *R2D2* von Kondylakis et al. (2020) ist nicht ganz klar, mit welcher Programmiersprache im Hintergrund gearbeitet wurde. Auch hier brachte die weitere Recherche keine Ergebnisse.

Für den *Chatbot für Modemarken* (Nazir et al., 2019) wurde zuerst selbständig eine Ontologie erstellt. Für den Bau einer Ontologie wird Protégé vorgeschlagen. Zumindest in diesem Paper wird deutlich gemacht, mit welchen Tools gearbeitet wurde. Die Ontologie wurde mit Jena zur Verfügung gestellt, dem open-source Semantic-Web Framework für Java. Eine Umsetzung mithilfe von Jena erfordert also Java-Kenntnisse.

Der *DBpedia Chatbot* (Athreya et al., 2018) ist der einzige Chatbot in der Liste, der abgerufen und getestet werden kann. Auch besteht die Möglichkeit diesen Chatbot lokal zu installieren und auf einer beliebigen Webseite einzubetten. Im Unterschied zu Viola werden die Texteingaben nicht mit AIML, sondern mit RiveScript verarbeitet. In der Projekt-Dokumentation ist vollständig aufgelistet, welche Tools für den DBpedia Chatbot genutzt wurden. Die Entwicklung des Projekts ist auf GitHub einsehbar und kann wiederverwendet werden, allerdings ist zu prüfen, ob sich die Anwendung auch in Kombination mit AIML umsetzen liesse. Eine weitere Möglichkeit besteht darin, einzelne Komponenten des Projekts zu verwenden. Qanary steht ebenfalls auf GitHub zur Verfügung. Diese Methodik basiert auf RDF und unterstützt den Aufbau von Frage/Antwort-Systemen, unabhängig von den verwendeten Ontologien oder Programmiersprachen (Both et al., 2016, S. 627). Im Rahmen dieser Bachelorarbeit wurde der DBpedia Chatbot ausprobiert. Dabei fiel auf, dass er bei Texteingaben auffallend häufig keine Antwort liefern kann. Die Begründung des Chatbots lautete, dass DBpedia «down» sei. Allerdings funktionierte zu diesen Zeiten die direkte SPARQL-Abfrage über den DBpedia SPARQL Query Editor oder den DBpedia SNORQL jeweils problemlos. Es scheint, dass mit dem DBpedia Chatbot etwas nicht in Ordnung ist.

Für den *Chatbot im Bereich E-Commerce* (Vegesna et al., 2018) wurde wie beim Chatbot für Modemarken eine eigene Ontologie erstellt. Die Abfrage der Daten erfolgte schliesslich unter Anwendung der Jape-Regeln. Neben Java-Kenntnissen sind für diese Lösung auch Python-Kenntnisse notwendig, da der Abruf der Funktion in Python erfolgt.

Da es sich bei *OntBot* (Al-Zubaide & Issa, 2011) lediglich um einen Vorschlag und um keine konkrete Umsetzung handelt, können keine Werkzeuge für die Chatbot-Erstellung abgeleitet werden.

Für *Project CyN* (Daxtron Laboratories, 2010) wurde die Abfrage einer Ontologie im Einklang mit AIML konzipiert, was Viola entgegen kommt. Zwar wurde CyN für die Abfrage von OpenCyc entwickelt (das nicht mehr verfügbar ist), mit Kenntnissen in der

Programmiersprache C++ könnte das Programm (auf der Seite von Daxtron verfügbar) aber angepasst werden.

Auch die *CyD-Technologie* (Augello et al., 2014) ist auf AIML abgestimmt, allerdings ist das in Java geschriebene Programm nicht zugänglich.

ProgramW (Caronia et al., o. J.) ist ein in Java geschriebener AIML-Interpreter. Die Erweiterung mit neuen AIML-Tags ermöglicht es dem Chatbot das lexikalische Wörterbuch WordNet abzufragen. Dieses Programm kann über Sourceforge heruntergeladen werden. Für eine SPARQL-Abfrage eines anderen Vokabulars müsste das Programm angepasst werden.

Zusammenfassend sind also die Komponenten des DBpedia Chatbots, des Project CyN und des ProgramW für eine Weiterverwendung geeignet. Beim erstgenannten Projekt müsste die Abstimmung mit AIML überprüft werden. Die anderen beiden Projekte sind bereits auf AIML angepasst.

7 Diskussion der Ergebnisse

Es folgt die Diskussion der in Kapitel 6 Darstellung der Ergebnisse dargestellten Ergebnisse.

7.1 Chatbots mit semantischen Technologien im Überblick

Für die Ergebnisse im Unterkapitel erachtete die Autorin es als notwendig, neben den semantischen Technologien auch die weiteren umgesetzten Technologien zu betrachten, um ein gesamtheitliches Bild der Funktionsweise der Chatbots zu erhalten. Dies ermöglichte es, zu verstehen, was alles für das Funktionieren eines Chatbots notwendig ist.

Bei den aufgelisteten Chatbot-Projekten fällt auf, dass viel Aufwand investiert wurde in das Verständnis der Fragen der Nutzenden und die Klassifizierung ihrer Absichten. Dabei lag der Fokus auf den Methoden des NLP. Statistische Vorgehensweisen kamen zum Einsatz, aber auch weiterreichende Formen des maschinellen Lernens. Neben den diesbezüglichen Ausführungen erscheint die Abfrage einer Wissensbasis für die Generierung der Antwort zum Teil fast nebensächlich. Die Lösung erfolgte oft durch die Abfrage einer Ontologie mit SPARQL.

Es fällt auf, dass nur bei einem der neun aufgelisteten Chatbots eine Implementation zu finden war. Dies liegt teilweise daran, dass es sich in den beschriebenen Projekten lediglich um Ansätze oder Lösungsvorschläge handelt, ohne dass tatsächlich eine Umsetzung erfolgte (zum Beispiel bei OntBot). Es ist möglich, dass aus den vorgestellten Projekten tatsächlich weitere Chatbots entstanden sind. Dies aufzudecken ist aber schwierig, da funktionstüchtige Chatbots im Internet nicht einfach so Auskunft darüber geben, auf welche Weise sie erstellt wurden.

Betreffend den Wissensbasen zeichnet sich DBpedia als beliebte Abfrage-Basis ab. Verwendet wird die Ontologie in KBot, R2D2 oder dem DBpedia Chatbot. Aufgrund der breiten Abdeckung verschiedener Wissensdomänen eignet sich DBpedia schnell, um generelle Information über ein beliebiges Themengebiet zu erhalten.

Um auf die Besonderheiten der einzelnen Chatbot-Projekte einzugehen, werden sie an dieser Stelle rekonstruiert. Die Umsetzung von KBot verspricht viel. Einerseits ist es der einzige unter den aufgelisteten Chatbots, der mehrsprachig funktioniert aufgrund der Arbeit mit einer langdetect-Bibliothek. Hier kommt auch zugute, dass die Wissensbasis DBpedia ebenfalls mehrsprachig zur Verfügung steht – eine Grundvoraussetzung für die mehrsprachige Abfrage. Darüber hinaus wurde der Chatbot von vornherein so konzipiert,

dass gleich mehrere Wissensbasen abgefragt werden können. In anderen Projekten wurde oft nur auf eine Wissensbasis fokussiert, was das volle Potential semantischer Technologien noch nicht ausschöpft. Es scheint, dass bei KBot kein AIML eingesetzt wurde, ganz deutlich angesprochen wird das aber im zugehörigen Paper nicht. Das ist schade, da dadurch eine direkte Übertragung auf den Bibliothekschatbot Viola (der ausschliesslich mit AIML gebaut wurde) nicht möglich scheint.

R2D2 ist ebenfalls ein Chatbot, der für die Abfrage von DBpedia konzipiert wurde. Hier ist der Nachteil, dass Anfragen nicht schon vom Chatbot umgewandelt und in Tripel transformiert werden, sondern dass die Anfrage bereits in Form eines Tripels gestellt werden muss. Gemäss den Autor:innen stelle das zwar für die Nutzenden keine Probleme dar. Nach einer kurzen Einführung seien sie in der Lage, die Anfragen gewinnbringend zu stellen. Keine obligate Einführung für die Nutzenden wäre aber noch besser. Ein grosser Mehrwert von Chatbots ist die Abfragemöglichkeit in natürlicher Sprache, was in diesem Beispiel umgangen wird. Betreffend der gängigen Skriptsprachen für Chatbots scheint es, dass R2D2 komplett ohne diese auskommt.

Für den Chatbot zu den Top10 Modemarken in Pakistan wurde extra eine Ontologie erstellt. Dieser Umstand führt dazu, dass sich ein Vorteil des Semantic-Webs – das allseitige Teilen von Daten – nicht ausschöpfen lässt. Dies ist vermutlich darauf zurückzuführen, dass eine solch spezifische Wissensbasis schlicht noch nicht existierte. Das Beispiel zeigt, dass die Erstellung einer Ontologie möglich ist, allerdings stellt sich die Frage, inwiefern dies sinnvoll ist, beziehungsweise ob sich der Aufwand lohnt in Anbetracht der isolierten Nutzbarkeit einer solchen Ontologie. Auch bei diesem Chatbot wurde kein AIML eingesetzt.

Auch beim DBpedia Chatbot wurde ein Teil der Wissensbasis selbständig erstellt, was auf den eng fokussierten Zweck (die Betreuung der Community-Mitglieder) zurückzuführen ist. Interessant an diesem Chatbot ist seine Konzipierung für unterschiedliche, breit genutzte Plattformen: der Chatbot ist über Slack, Facebook oder die eigene Webseite erreichbar. Die Begründung für diese plattformübergreifende Bereitstellung liegt wahrscheinlich in der Notwendigkeit der öffentlichen Zugänglichkeit. Im zugehörigen Paper waren die Ausführungen zu den eingesetzten Tools aufschlussreich. Unter den vorgestellten Tools scheint Qanary es Wert näher betrachtet zu werden, da es auf bestehende Ontologien ausgerichtet ist.

Project CyN diente als Vorlage für weitere Projekte, in denen mittels Chatbots auf Ontologien zugegriffen wurde. Die Ontologie OpenCyc, die nicht mehr verfügbar ist,

schien als «common-sense» Wissensbasis eine geeignete Erweiterung für Chatbots mit unterschiedlichen Wissensdomänen zu sein.

Mit der CyD-Technologie wurde das Project CyN an den in Java Programm geschriebenen AIML-Interpreter angepasst. Bei CyD und CyN ist sehr interessant zu sehen, wie die AIML-Tags für die Abfrage einer Ontologie erweitert wurden. Dies zeigt, dass eine Erweiterung von AIML möglich ist, allerdings sind dazu Programmierkenntnisse notwendig. Die anschließende Einbindung der spezifischen Tags in die AIML-Datei stellt dann aber ein kleiner Schritt dar, der mit grundlegenden Kenntnissen in AIML und den Ontologie-Abfragesprachen zu bewerkstelligen ist. Der Einsatz von CyD wurde in den Projekten Maga (für archäologische Kulturstätten) und FRASI (für den Bereich Milchverarbeitung) demonstriert. In den Ausführungen zu FRASI findet sich eine ausführliche Beschreibung dazu, wie die wichtigen Begriffe aus der Anfrage extrahiert werden, damit die Ontologie entsprechend korrekt abgefragt werden kann. Dies ist ein wichtiger Prozessschritt für das zufriedenstellende Funktionieren des Chatbots. Das besondere an den beiden Projekten Maga und FRASI ist auch, dass sie nicht auf Englisch, sondern auf Italienisch funktionieren. Generell wurde in den gefundenen wissenschaftlichen Publikationen die Umsetzung von Chatbots in anderen Sprachen als Englisch nicht diskutiert. Da die Autorin dieser Arbeit selbständig anhand eines Tutorials einen Chatbot mit AIML erstellt hat, ist ihr aufgefallen, dass ein Chatbot in deutscher Fassung mehr Denkarbeit und Variation erfordert als ein englischer Chatbot. Zum Beispiel können Verbformen und Pronomen auf Deutsch weniger generisch eingesetzt werden als auf Englisch. Die Autorin geht davon aus, dass sich eine derartige Diskussion eher auf die Literatur zu NLP-Techniken verlagert und deshalb bei den Ausführungen zu den Chatbots nicht erwähnt wird.

Schliesslich gab es aber doch einen Ansatz, bei dem die Sprachanalysefähigkeit des Chatbots verbessert werden sollte. Bei EHeBby wurde das Program W (ein AIML-Interpreter auf Basis von Java) eingesetzt. Für die Verbesserung des Chatbot-Wortschatzes wurde WordNet eingebunden. Dies ist das einzige der vorgestellten Projekte, bei dem es nicht primär darum ging durch Ontologien den Inhalt der Antworten zu erweitern, sondern um eine präzisere Einschätzung der Absichten der Nutzenden (und anschliessend bessere Antwortfindung) herbeizuführen.

Abschliessend wird festgestellt, dass es durch den Einsatz semantischer Technologien möglich ist Inferenzprozesse für Chatbots zu nutzen. Zusätzlich erweitert die Anbindung einer Wissensbasis die Antwortmöglichkeiten. Die Chatbots, welche semantische Technologien nutzen, wurden mit unterschiedlichen Programmier- und Skriptsprachen

umgesetzt. Unter den gefundenen Projekten finden sich Chatbots auf der Basis von AIML (etwas ältere Projekte) und RiveScript, aber auch Java wurde häufig in verschiedenen Formen implementiert.

Grundsätzlich bieten sich unterschiedliche Ontologien für die Anbindung an einen Chatbot an – die Auswahl der Ontologie bezieht sich auf den Zweck des Chatbots und seine abzudeckende Domäne. Ein vermutlich häufig auftretendes Problem ist das Fehlen einer bereits fertigen Ontologie für die spezifischen Bedürfnisse der Nutzenden. Zwar ist es möglich selbständig eine Ontologie zu erstellen, doch Aufwand und Ertrag sind dabei abzuwägen.

Es bleibt noch auf die Frage aus dem Titel der Arbeit einzugehen: Sind Chatbots und Semantic-Web ein Dream-Team? Das Cambridge Dictionary (o. J.) definiert ein Dream-Team wie folgt: «A group of people who have been specially chosen to work together and are considered to be the best at what they do» (Cambridge Dictionary, o. J.).

Die Antwort lautet: Ja, Chatbots und das Semantic-Web sind ein Dream-Team, wenn es darum geht, eine Wissensbasis für den Chatbot zu erschliessen. In diesem Fall arbeiten die beiden sehr gut zusammen. Das Aber bezieht sich auf all die anderen Funktionen, die ein Chatbot erfüllen muss. Die Analyse und Klassifizierung der Absichten der Nutzenden ist sorgfältig umzusetzen, da dies über die Leistung des Chatbots entscheidet. Hierfür sind weitere Technologien neben den semantischen einzusetzen. Dazu zählen zum Beispiel Methoden des NLP oder des Deep-Learnings. Ebenso ist die Antwortgenerierung nicht ausschliesslich durch semantische Technologien umzusetzen. Grundsätzlich führt die Kombination verschiedener Technologien zu einem funktionierenden Chatbot. Damit ein Chatbot am Ende zufriedenstellend funktioniert, sind Programmierfähigkeiten notwendig. Unterschiedliche Anbieter stellen Entwicklungsplattformen für Chatbots zur Verfügung, die auch ohne Programmierkenntnisse bedient werden können (zum Beispiel der IBM Watson Assistant), jedoch stellt sich bei solchen Angeboten die Frage, ob eine Abhängigkeit vom Anbieter in Kauf genommen wird. Ebenfalls einzubeziehen bei solchen Überlegungen ist der Datenschutz.

7.2 Erzielte Verbesserungen durch semantische Technologien

Zur Evaluierung von Chatbots gibt es keine Standard-Methode. Je nach Zweck des Chatbots und der Evaluierung werden unterschiedliche Messkriterien und -methoden gewählt. Eine Ausrichtung nach den Masseinheiten der Gebrauchstauglichkeit (Effizienz, Effektivität und Zufriedenstellung) bietet sich an, da so eine umfassende Abdeckung

wichtiger Merkmale gewährleistet werden kann. Die Untersuchung von Casas et al. (2020) hat aber gezeigt, wie unterschiedlich die Merkmale interpretiert werden können. Fehlende Standards benachteiligen auch den übergreifenden Vergleich zwischen verschiedenen Chatbots. Eine Evaluierung verschiedener Design-Methoden hat in der wissenschaftlichen Literatur noch nicht stattgefunden. Da zusätzlich die für diese Arbeit gefundenen Chatbot-Projekte grossteils nicht zugänglich sind, wurde entschieden, einen anderen Ansatz weiterzuverfolgen. Vishwakarma & Pandey (2021) haben häufig auftretende Mängel bei Chatbots dokumentiert. Anhand der beschriebenen Fortschritte in den Chatbot-Projekten aus 6.1 wurde aufgezeigt, dass semantische Technologien die Wissensdomäne von Chatbots erweitern können. Auch im Bereich der Sprachanalyse können Verbesserungen erzielt werden, zum Beispiel in Kombination mit der LSA-Methode. Zu erwarten gewesen wäre eine grössere Verbesserung im Bereich der Semantik, doch auf diese lässt sich aufgrund der gefundenen Chatbot-Projekte nicht schliessen. Weshalb die Bedeutung der Wörter nicht besser genutzt werden kann, bleibt in dieser Arbeit eine offene Frage. Werden die Vorteile semantischer Technologien schlicht nicht voll ausgeschöpft? Dies wäre in weiteren Untersuchungen zu klären.

7.3 Abwägungen für die zukünftige Entwicklung von Viola

Für die Erweiterung des Bibliotheks-Chatbots Viola sind dessen Aufbau und Zweck zu betrachten. Es handelt sich um einen mit AIML erstellten Chatbot, der Auskunft über die Nutzung der Bibliothek gibt. Als besonders hilfreich erachtet, wird eine Erweiterung seiner Funktion durch direkten Zugriff auf den Bestand der Bibliothek. Dazu ist eine Ontologie notwendig, die am besten über ein SPARQL-Endpoint erreicht werden kann und die bibliografische Daten enthält. Linked Swissbib hätte diesen Dienst angeboten, wird aber nicht mehr aktualisiert. Es sieht derzeit nicht danach aus, dass SLSP einen ähnlichen Dienst für Linked-Data zur Verfügung stellen wird. Auch möglich ist die Abfrage der DBpedia, jedoch wäre dies eine Erweiterung der Funktion des Chatbots Viola, sozusagen eine Zugabe.

Damit die AIML-Files von Viola erhalten bleiben können, ist eine Implementierung der Technologien aus den Projekten CyN und ProgramW denkbar. Grundsätzlich kann mit fast jeder Programmiersprache ein Zugang zu RDF-Dokumenten herbeigeführt werden. Allerdings sind die beiden erwähnten Programme bereits auf AIML abgestimmt. Sie stehen online zur Verfügung, müssten aber mithilfe von Programmierkenntnissen angepasst werden, damit auch andere Ontologien abgefragt werden können. Auch Qanary aus dem DBpedia Chatbot kann von Nutzen sein, doch auch für diesen Einsatz

sind Programmierkenntnisse notwendig. Der DBpedia Chatbot bleibt aber als schlechtes Beispiel in Erinnerung, da Faktenabfragen fast nie funktionieren, was ein No-Go für einen Chatbot ist und bei den Nutzenden zu Frustration führt.

Die erforderlichen Programmierkenntnisse und das Fehlen einer passenden Ontologie stellen grosse Hürden für den Einsatz semantischer Technologien bei Viola dar. Derzeit hält sich der Aufwand für den Unterhalt des Chatbots in Grenzen. Für Anpassungen müssten die Fähigkeiten, Zeit (und damit Geld) zur Verfügung stehen.

Die Zukunftsaussichten für die auf AIML basierte Viola sind denn auch eher düster: es ist geplant, sie durch einen neuen Chatbot zu ersetzen. Es wird danach gestrebt die Umsetzung an die Firma aiaibot zu delegieren, da diese bereits andere Chatbots für die FHGR zur Verfügung stellt. So wäre es möglich, die Chatbots der Fachhochschule einheitlich zu verwalten und Kenntnisse unter den Mitarbeitenden zu teilen (M. Hanselmann, persönliche Kommunikation, 29. Juni 2021).

8 Fazit

Die Arbeit hat gezeigt, dass bei der Verbesserung von Chatbots unterschiedliche Technologien ausprobiert werden. Aufgrund der systematischen Literaturrecherche konnte eine Liste von Chatbot-Projekten erstellt werden, bei welchen semantische Technologien zum Einsatz kommen. In der Regel wird mittels SPARQL auf eine Ontologie zugegriffen. Beliebt ist die Abfrage von DBpedia, doch kommen je nach Zweck des Chatbots auch andere Ontologien zum Einsatz. Die semantischen Technologien nehmen innerhalb der Chatbots aber in der Regel einen eher kleinen Stellenwert ein, verglichen zu den anderen eingesetzten Technologien. Besonders für das Verständnis der Anfragen der Nutzenden und die Klassifizierung der Absichten sind weitere Technologien (zum Beispiel NLP) im Einsatz. In der Literatur wird auch der Einsatz von Deep-Learning-Techniken besprochen.

Unter den neun gefundenen Projekten wurde nur für einen eine tatsächliche Umsetzung gefunden. Entweder sind die Vorschläge schon etwas älter oder es handelt sich tatsächlich nur um Ansätze, sprich Ideen zur Umsetzung. Daneben bieten drei Projektbeschreibungen Hinweise auf Tools, die weiter genutzt werden können: Canary, CyN und Program W. Alle drei Tools stehen frei zur Verfügung. Für deren Einsatz in Chatbots sind aber Programmierfähigkeiten vorausgesetzt.

Der Einsatz semantischer Technologien bietet in der Chatbot-Entwicklung Vorteile. Die Wissensdomäne der Chatbots lässt sich mithilfe von Ontologien auf schnelle Weise vergrößern. Es können mehrere Ontologien verlinkt werden. Auch zeigte sich in einem Projekt die Verbesserung der Sprachanalysefähigkeit unter Einsatz von WordNet in Kombination mit der LSA-Methode. Eine grössere Verbesserung wurde im Bereich der Semantik erwartet. Technologien des Semantic-Web wären dazu prädestiniert, die eindeutige Begriffszuweisung zu verbessern und die Bedeutung von Wörtern stärker in die Verarbeitungsprozesse des Chatbots einzubinden. Auf eine Verbesserung in diesem Sinne lässt sich in den gefundenen Chatbot-Projekten nicht schliessen. Ob das Potential semantischer Technologien schlicht noch nicht voll ausgeschöpft wurde, bleibt eine offene Frage.

Bezüglich der Übertragung semantischer Technologien auf Chatbots im Bibliotheksbereich wurde Viola von der FHGR Bibliothek betrachtet. Vorgeschlagen wurde eine Erweiterung der Auskunftsdienstleistung um einen direkten Zugriff auf Bestandesinformation der Bibliothek. Der nicht mehr aktive Dienst von Linked Swissbib hätte einen solchen Zugang ermöglicht. Da dieser nicht mehr weitergeführt wird, gibt es in der schweizerischen Bibliothekslandschaft eine Lücke im Bereich Linked-Data. Eine

passende Ontologie für diesen Zweck konnte nicht gefunden werden. Denkbar ist auch eine Erweiterung von Viola über DBpedia, da der Chatbot so Faktenwissen weitergeben könnte. Allerdings würde dies eine Ausweitung des Zwecks von Viola bedeuten. Viola wurde dazu erstellt, Anfragen zur Nutzung der Bibliothek zu beantworten.

Neben der nicht vorhandenen Ontologie stellt auch der Einsatz der vorgestellten Technologien eine Hürde dar. Für die Umsetzung sind Programmierkenntnisse erforderlich. Zeit und Geld müsste investiert werden. Die FHGR Bibliothek überlegt sich momentan, die auf AIML-basierte Viola durch einen neuen Chatbot zu ersetzen. Dieser würde durch eine externe Firma erstellt. Generell stellt sich für Firmen und Institutionen die Frage, ob eine eigenständige Implementierung oder Auslagerung des Chatbot-Projekts an eine spezialisierte Firma sinnvoller ist. Bei der Auslagerung an Dritte sind die Abhängigkeit von der externen Dienstleisterin sowie der Datenschutz in Betracht zu ziehen.

Die Frage im Titel der Arbeit, ob Semantic-Web und Chatbots ein Dream-Team sind, lässt sich mit «Ja» beantworten. Semantische Technologien erleichtern die Erschliessung grosser Wissensbasen für Chatbots. Aber die semantischen Technologien allein machen noch keinen Chatbot. Die Kombination mit anderen Technologien ist unabdingbar für das gute Funktionieren eines Chatbots.

9 Quellenverzeichnis

- Abdul-Kader, S. A., & Woods, J. (2015). Survey on chatbot design techniques in speech conversation systems. *International Journal of Advanced Computer Science and Applications*, 6(7), 72–80. <https://doi.org/10.14569/IJACSA.2015.060712>
- Adamopoulou, E., & Moussiades, L. (2020). An overview of chatbot technology. In I. Maglogiannis, L. Iliadis, & E. Pimenidis (Hrsg.), *Artificial Intelligence Applications and Innovations* (S. 373–383). Springer International Publishing. https://doi.org/10.1007/978-3-030-49186-4_31
- AIML Foundation. (2018). *AIML Docs*. <http://www.aiml.foundation/doc.html>
- Ait-Mlouk, A., & Jiang, L. (2020). KBot: A knowledge graph based chatbot for natural language understanding over linked data. *IEEE Access*, 8, 149220–149230. <https://doi.org/10.1109/ACCESS.2020.3016142>
- Al-Zubaide, H., & Issa, A. A. (2011). OntBot: Ontology based chatbot. *International Symposium on Innovations in Information and Communications Technology*, 7–12. <https://doi.org/10.1109/ISIICT.2011.6149594>
- Athreya, R. G., Ngonga Ngomo, A.-C., & Usbeck, R. (2018). Enhancing community interactions with data-driven chatbots: The DBpedia Chatbot. In P.-A. Champin, F. Gandon, & L. Médini (Hrsg.), *Companion Proceedings of the The Web Conference 2018* (S. 143–146). International World Wide Web Conferences Steering Committee. <https://doi.org/10.1145/3184558.3186964>
- Augello, A., Gentile, M., & Dignum, F. (2017). An overview of open-source chatbots social skills. In S. Diplaris, A. Satsiou, A. Følstad, M. Vafopoulos, & T. Vilarinho (Hrsg.), *Internet Science* (S. 236–248). Springer International Publishing. https://doi.org/10.1007/978-3-319-77547-0_18
- Augello, A., Pilato, G., Machi', A., & Gaglio, S. (2012). An approach to enhance chatbot semantic power and maintainability: Experiences within the FRASI project. *2012 IEEE Sixth International Conference on Semantic Computing*, 186–193. <https://doi.org/10.1109/ICSC.2012.26>
- Augello, A., Pilato, G., Vassallo, G., & Gaglio, S. (2014). Chatbots as interface to ontologies. In S. Gaglio & G. Lo Re (Hrsg.), *Advances onto the internet of things: How ontologies make the internet of things meaningful* (S. 285–299). Springer International Publishing. https://doi.org/10.1007/978-3-319-03992-3_20
- Bagchi, M. (2020). Conceptualising a library chatbot using open source conversational artificial intelligence. *DESIDOC Journal of Library & Information Technology*, 40(6), 329–333. <https://doi.org/10.14429/djlit.40.06.15611>
- Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2), 157–166. <https://doi.org/10.1109/72.279181>

- Both, A., Diefenbach, D., Singh, K., Shekarpour, S., Cherix, D., & Lange, C. (2016). Canary: A methodology for vocabulary-driven open question answering systems. In H. Sack, E. Blomqvist, M. d'Aquin, C. Ghidini, S. P. Ponzetto, & C. Lange (Hrsg.), *The Semantic Web. Latest Advances and New Domains. ESWC 2016*. (S. 625–641). Springer International Publishing. https://doi.org/10.1007/978-3-319-34129-3_38
- Bruno Marietto, M. das G., Aguiar, R. V., Barbosa, G. de O., Botelho, W. T., Pimentel, E., Franca, R. dos S., & da Silva, V. L. (2013). Artificial Intelligence Markup Language: A brief tutorial. *International Journal of Computer Science & Engineering Survey*, 4(3), 1–20. <https://doi.org/10.5121/ijcses.2013.4301>
- Butterfield, A. B., Ngondi, G. E. N. E., & Kerr, A. K. (2016). Trie search. In A. Butterfield, G. E. Ngondi, & A. Kerr (Hrsg.), *A dictionary of computer science*. Oxford University Press.
- Cambridge Dictionary. (o. J.). *Dream team*. Cambridge Dictionary. Abgerufen 1. Juli 2021, von <https://dictionary.cambridge.org/dictionary/english/dream-team>
- Caronia, A., Pilato, G., Augello, A., & Gaglio, S. (o. J.). *Program W*. Getting started with ProgramW. Abgerufen 27. Juni 2021, von <http://programw.sourceforge.net/>
- Casas, J., Tricot, M.-O., Abou Khaled, O., Mugellini, E., & Cudré-Mauroux, P. (2020). Trends & Methods in Chatbot Evaluation. *Companion Publication of the 2020 International Conference on Multimodal Interaction*, 280–286. <https://doi.org/10.1145/3395035.3425319>
- Cox, C. (2005). *Assessing the utility of ResearchCyc in recognizing textual entailment*. Stanford University. <https://nlp.stanford.edu/projects/cycErrorAnalysis.html>
- Dale, R. (2016). The return of the chatbots. *Natural Language Engineering*, 22(5), 811–817. <https://doi.org/10.1017/S1351324916000243>
- Danowski, P., & Pohl, A. (2013). *(Open) Linked Data in Bibliotheken* (Bd. 50). De Gruyter Saur. <https://doi.org/10.1515/9783110278736>
- Daxtron Laboratories. (2010). *Project CyN*. Daxtron Laboratories. <https://www.daxtron.com/CyNAIML.html>
- DBpedia Association. (2021). *About DBpedia*. DBpedia Association. <https://www.dbpedia.org/about/>
- Deutsche Nationalbibliothek. (2019). *Linked-Data-Service*. Deutsche Nationalbibliothek. https://www.dnb.de/DE/Professionell/Metadatendienste/Datenbezug/LDS/lds_node.html
- Di Blas, N., Lodi, L., Paolini, P., Pernici, B., Renzi, F., & Rooein, D. (2019). *Data driven chatbots: A new approach to conversational applications*. 27th Italian Symposium on Advanced Database, Castiglione della Pescaia, Italy. https://www.researchgate.net/publication/338833712_Data_Driven_Chatbots_a_New_Approach_to_Conversational_Applications

- Eichenauer, H., Hentschke, J., Rupp, J., & Svensson, L. G. (2018). *Der Linked-Data-Service der Deutschen Nationalbibliothek: Auslieferung der Metadaten*. Deutsche Nationalbibliothek.
https://www.dnb.de/SharedDocs/Downloads/DE/Professionell/Metadatendienste/linkedDataZugriff.pdf?__blob=publicationFile&v=3
- Freese, E. (2007). *Enhancing AIML bots using semantic web technologies*. *Proceedings of Extreme Markup Languages*. <http://web-old.archive.org/web/20070917184351/http://www.idealliance.org/papers/extreme/proceedings/html/2007/Freese01/EML2007Freese01.html>
- Galitsky, B. (2019). *Developing Enterprise Chatbots: Learning Linguistic Structures*. Springer International Publishing. <https://doi.org/10.1007/978-3-030-04299-8>
- Gartner Group. (2019). *Chatbots will appeal to modern workers*. Smarter With Gartner. <http://www.gartner.com/smarterwithgartner/chatbots-will-appeal-to-modern-workers/>
- GeeksforGeeks. (2021). *Trie | (Insert and Search)*. GeeksforGeeks. <https://www.geeksforgeeks.org/trie-insert-and-search/>
- Gygax, R. (2009). *Chatbots in Bibliotheken: Theoretische Überlegungen und Umsetzung eines Chatbots für die Bibliotheken der HTW Chur* [Unveröffentlichte Bachelor-Thesis]. Hochschule für Technik und Wirtschaft Chur.
- Hanselmann, M. (2020). *Chatbots in Bibliotheken: Eine aktuelle Bestandesaufnahme* [Unveröffentlichte Master-Thesis]. Fachhochschule Graubünden.
- Hitzler, P., Krötzsch, M., Rudolph, S., & Sure, Y. (2008). *Semantic Web: Grundlagen*. Springer. <https://doi.org/10.1007/978-3-540-33994-6>
- Jadeja, M., & Varia, N. (2017). Perspectives for evaluating conversational AI. *arXiv:1709.04734 [cs.AI]*. <http://arxiv.org/abs/1709.04734>
- Kona Connect. (2020). *Sophia: Der Chatbot*. Kona Connect. <https://kona-connect.org/de/sophia/>
- Kondylakis, H., Tsirigotakis, D., Fragkiadakis, G., Panteri, E., Papadakis, A., Fragkakis, A., Tzagkarakis, E., Rallis, I., Saridakis, Z., Trampas, A., Pirounakis, G., & Papadakis, N. (2020). R2D2: A DBpedia chatbot using triple-pattern like queries. *Algorithms*, 13(9), 217. <https://doi.org/10.3390/a13090217>
- Koutamanis, A. (2002). AZILE: an intelligent verbal interface for the retrieval of pictorial architectural information. In K. Agger (Hrsg.), *Distributing knowledge in building* (S. 8). <http://itc.scix.net/paper/w78-2002-51>
- La Bua, S. (2021). LSA-Bot. *SLB Labs*. <https://www.slblabs.com/projects/l sabot/>
- Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes*, 25(2–3), 259–284. <https://doi.org/10.1080/01638539809545028>

- Lee, J. H., Yang, H., Shin, D., & Kim, H. (2020). Chatbots. *ELT Journal*, 74(3), 338–344. <https://doi.org/10.1093/elt/ccaa035>
- Märki, M. (2020). Manager aus Bits und Bytes. *Globe - das Magazin der ETH Zürich und der ETH Alumni*, 3, 29. https://ethz.ch/content/dam/ethz/common/docs/media/globe/2020/Globe2003_Alles-KI_DE.pdf
- Maroengsit, W., Piyakulpinyo, T., Phonyiam, K., Pongnumkul, S., Chaovalit, P., & Theeramunkong, T. (2019). A survey on evaluation methods for chatbots. *Proceedings of the 2019 7th International Conference on Information and Education Technology*, 111–119. <https://doi.org/10.1145/3323771.3323824>
- Mckie, I. A. S., & Narayan, B. (2019). Enhancing the academic library experience with chatbots: An exploration of research and implications for practice. *Journal of the Australian Library and Information Association*, 68(3), 268–277. <https://doi.org/10.1080/24750158.2019.1611694>
- Mori, K., Jatowt, A., & Ishizuka, M. (2003). Enhancing conversational flexibility in multimodal interactions with embodied lifelike agent. In *Proceedings of the 8th international conference on Intelligent user interfaces* (S. 270–272). Association for Computing Machinery. <https://doi.org/10.1145/604045.604096>
- Nazir, A., Khan, M. Y., Ahmed, T., Syed, I. J., & Wasi, S. (2019). A novel approach for ontology-driven information retrieving chatbot for fashion brands. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 10(9). <https://doi.org/10.14569/IJACSA.2019.0100972>
- Nielsen, J. (2020). *10 usability heuristics for user interface design*. Nielsen Norman Group. <https://www.nngroup.com/articles/ten-usability-heuristics/>
- Nimavat, K., & Champaneria, T. (2017). Chatbots: An overview. types, architecture, tools and future possibilities. *IJSRD - International Journal for Scientific Research & Development*, 5(7), 1019–1026.
- opendata.swiss. (2017). *Bibliographic Data from Swiss libraries, archives and other institutions*. opendata.swiss. <https://opendata.swiss/de/dataset/bibliographic-data-from-swiss-libraries-archives-and-other-institutions>
- Panda, S., & Chakravarty, R. (2021). *Implementing conversational AI in libraries: A practical approach*. 124–145. <https://doi.org/10.31235/osf.io/werz6>
- Pilato, G., Augello, A., Vassallo, G., & Gaglio, S. (2008). EHeBby: An evocative humorist Chat-Bot. *Mobile Information Systems*, 4(3), 165–181. <https://doi.org/10.1155/2008/312319>
- Porreca, S., Leotta, F., Mecella, M., Vassos, S., & Catarci, T. (2018). Accessing government open data through chatbots. In I. Garrigós & M. Wimmer (Hrsg.), *Current*

- Trends in Web Engineering* (S. 156–165). Springer International Publishing. https://doi.org/10.1007/978-3-319-74433-9_14
- Princeton University. (2021). *WordNet: A lexical database for English*. <https://wordnet.princeton.edu/>
- Radziwill, N., & Benton, M. C. (2017). Evaluating quality of chatbots and intelligent conversational agents. *ArXiv*. <https://arxiv.org/abs/1704.04579>
- Ramesh, K., Ravishankaran, S., Joshi, A., & Chandrasekaran, K. (2017). A survey of design techniques for conversational agents. In S. Kaushik, D. Gupta, L. Kharb, & D. Chahal (Hrsg.), *Information, Communication and Computing Technology: Second International Conference, ICICCT 2017, New Delhi, India, May 13, 2017, Revised Selected Papers* (S. 336–350). Springer. https://doi.org/10.1007/978-981-10-6544-6_31
- Stock, O. (2001). Language-based interfaces and their application for cultural tourism. *AI Magazine, Spring*, 85–97.
- Stoner, D. J., Ford, L., & Ricci, M. (2003). *Simulating military radio communications using speech recognition and chat-bot technology*. <https://docplayer.net/39136593-Simulating-military-radio-communications-using-speech-recognition-and-chat-bot-technology.html>
- Stucki, T., D'Onofrio, S., & Portmann, E. (2020). *Chatbots gestalten mit Praxisbeispielen der Schweizerischen Post: HMD Best Paper Award 2018*. Springer Fachmedien. <https://doi.org/10.1007/978-3-658-28586-9>
- Unger, C., Böhmann, L., Lehmann, J., Ngonga Ngomo, A.-C., Gerber, D., & Cimiano, P. (2012). Template-based question answering over RDF data. *Proceedings of the 21st International Conference on World Wide Web - WWW '12*, 639–648. <https://doi.org/10.1145/2187836.2187923>
- Valtolina, S., Barricelli, B. R., Di Gaetano, S., & Diliberto, P. (2018). Chatbots and conversational interfaces: Three domains of use. *Proceedings of Fifth International Workshop on Cultures of Participation in the Digital Age (CoPDA 2018)*, 62–70. <http://ceur-ws.org/Vol-2101/paper8.pdf>
- Vegesna, A., Jain, P., & Porwal, D. (2018). Ontology based chatbot (for e-commerce website). *International Journal of Computer Applications*, 179(14), 51–55. <https://doi.org/10.5120/ijca2018916215>
- Venkatesh, A., Khatri, C., Ram, A., Guo, F., Gabriel, R., Nagar, A., Prasad, R., Cheng, M., Hedayatnia, B., Metallinou, A., Goel, R., Yang, S., & Raju, A. (2018). On evaluating and comparing open domain dialog systems. *arXiv:1801.03625 [cs.CL]*. <http://arxiv.org/abs/1801.03625>
- Vincze, J. (2017). Virtual reference librarians (Chatbots). *Library Hi Tech News*, 34(4), 5–8. <https://doi.org/10.1108/LHTN-03-2017-0016>

- Vishwakarma, A., & Pandey, A. (2021). A review & comparative analysis on various chatbots design. *International Journal of Computer Science and Mobile Computing*, 10(2), 72–78. <https://doi.org/10.47760/ijcsmc.2021.v10i02.011>
- W3C. (2015). *Semantic Web*. W3C. <https://www.w3.org/standards/semanticweb/>
- W3Schools. (o. J.). *Introduction to XML*. W3Schools. Abgerufen 4. August 2021, von https://www.w3schools.com/xml/xml_what_is.asp
- Ward, L. (2021, März 27). Can artificial intelligence replace human therapists? *Wall Street Journal*. <https://www.wsj.com/articles/can-artificial-intelligence-replace-human-therapists-11616857200>
- WDAqua. (2020). *Qanary* [Java]. WDAqua. <https://github.com/WDAqua/Qanary>
- Wikidata. (2019). *Wikidata*. https://www.wikidata.org/wiki/Wikidata:Main_Page
- Wilcox, B. (2019). *ChatScript*. GitHub. <https://github.com/ChatScript/ChatScript>
- Zdravkova, K. (2000). Conceptual framework for an intelligent chatterbot. *Proceedings of the 22nd International Conference on Information Technology Interfaces*, 189–194. <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=915905>
- Zou, L. (2018). SPARQL. In L. Liu & M. T. Özsu (Hrsg.), *Encyclopedia of Database Systems* (S. 3554–3558). Springer. https://doi.org/10.1007/978-1-4614-8265-9_80803

Bisher erschienene Schriften

Ergebnisse von Forschungsprojekten erscheinen jeweils in Form von Arbeitsberichten in Reihen.
Sonstige Publikationen erscheinen in Form von alleinstehenden Schriften.

Derzeit gibt es in den Churer Schriften zur Informationswissenschaft folgende Reihen:
Reihe Berufsmarktforschung

Weitere Publikationen

Churer Schriften zur Informationswissenschaft – Schrift 118
Herausgegeben von Wolfgang Semar
Rebekka Hirsbrunner
Conversation Design für textbasierte Conversational Agents
Umsetzung eines Dialogkonzepts am Beispiel eines Studienberatungs-Chatbox einer Hochschule
Chur, 2020
ISSN 1660-945X

Churer Schriften zur Informationswissenschaft – Schrift 119
Herausgegeben von Wolfgang Semar
Nicole Zimmermann
Elektronische Patientenaufklärung in Schweizer Spitälern
Eine Machbarkeitsstudie
Chur, 2020
ISSN 1660-945X

Churer Schriften zur Informationswissenschaft – Schrift 120
Herausgegeben von Wolfgang Semar
Sarah Stalder
Wie können Barrieren im organisationalen Wissenstransfer abgebaut werden?
Barrieren bei der Teilung und Nutzung von Wissen und Lösungsansätze für die
Unternehmenspraxis
Chur, 2020
ISSN 1660-945X

Churer Schriften zur Informationswissenschaft – Schrift 121
Herausgegeben von Wolfgang Semar
Saskia Müller
Mit humanoiden Robotern Bewerbungsgespräche trainieren
Chur, 2020
ISSN 1660-945X

Churer Schriften zur Informationswissenschaft – Schrift 122
Herausgegeben von Wolfgang Semar
Christian Arthur Müller
Archivöffnung: Zugänglichkeit der SRG-Sendungsarchive am Beispiel von SRF
Chur, 2020
ISSN 1660-945X

Churer Schriften zur Informationswissenschaft – Schrift 123
Herausgegeben von Wolfgang Semar
Susanne Grieder
Archive: Infrastruktur- und Bestandesnutzung durch Menschen mit Sehbehinderung oder Blindheit
Chur, 2020
ISSN 1660-945X

Churer Schriften zur Informationswissenschaft – Schrift 124
Herausgegeben von Wolfgang Semar
Sophia Zimmerer
Digital Nudging im Pre-Purchase Kontext
Einfluss des Social Norm Nudge im Social-Media-Advertising während der Need Recognition
Phase
Chur, 2021
ISSN 1660-945X

Churer Schriften zur Informationswissenschaft – Schrift 125
Herausgegeben von Wolfgang Semar
Nadine Christinger
Medienpädagogik in Schulbibliotheken
Zukünftige Rolle von Schul- und Gemeindebibliotheken im Bereich der Medienpädagogik am
Beispiel des Kantons St. Gallen
Chur, 2021
ISSN 1660-945X

Churer Schriften zur Informationswissenschaft – Schrift 126
Herausgegeben von Wolfgang Semar
Mirjam Nydegger
Unterrichtskonzept eines Forschungsdatenmanagement-Kurses für Mediziner im Masterstudium an
der Universität Bern
Chur, 2021
ISSN 1660-945X

Churer Schriften zur Informationswissenschaft – Schrift 127
Herausgegeben von Wolfgang Semar
Meret Stocker
Erlesene Räume
Eine Analyse zur Nutzungsauslastung von Lesesälen wissenschaftlicher Bibliotheken von 1990 bis
heute
Chur, 2021
ISSN 1660-945X

Churer Schriften zur Informationswissenschaft – Schrift 128
Herausgegeben von Wolfgang Semar
Ramona Blum
Das Medien-Image der "Grossen Vier" (Google, Apple, Facebook und Amazon - GAFA)
Chur, 2021
ISSN 1660-945X

Churer Schriften zur Informationswissenschaft – Schrift 129
Herausgegeben von Wolfgang Semar
Linus Niederhauser
Digital Nudging im Pre-Purchase-Kontext der Customer Journey unter Berücksichtigung des
Umweltbewusstseins
Chur, 2021
ISSN 1660-945X

Churer Schriften zur Informationswissenschaft – Schrift 130
Herausgegeben von Wolfgang Semar
Colin Bolli
Impact of Digital Payment Methods on Traditional Payment Transactions
An Analysis of the Effects on the Swiss Financial Market
Chur, 2021
ISSN 1660-945X

Churer Schriften zur Informationswissenschaft – Schrift 131
Herausgegeben von Wolfgang Semar
Patrik Jurkovic
Erfolgsgarant Lean-Startup Approach?
Chur, 2021
ISSN 1660-945X

Churer Schriften zur Informationswissenschaft – Schrift 132
Herausgegeben von Wolfgang Semar
Sandra Rumiz
Firmenarchive in Wikimedia-Projekten
Wie Bestände von Schweizer Textilunternehmen über Wikipedia und Wikidata auffindbar werden
Chur, 2021
ISSN 1660-945X

Über die Informationswissenschaft der Fachhochschule Graubünden

Die Informationswissenschaft ist in der Schweiz noch ein relativ junger Lehr- und Forschungsbereich. International weist diese Disziplin aber vor allem im anglo-amerikanischen Bereich eine jahrzehntelange Tradition auf. Die klassischen Bezeichnungen dort sind Information Science, Library Science oder Information Studies. Die Grundfragestellung der Informationswissenschaft liegt in der Betrachtung der Rolle und des Umgangs mit Information in allen ihren Ausprägungen und Medien sowohl in Wirtschaft und Gesellschaft. Die Informationswissenschaft wird in Chur integriert betrachtet.

Diese Sicht umfasst nicht nur die Teildisziplinen Bibliothekswissenschaft, Archivwissenschaft und Dokumentationswissenschaft. Auch neue Entwicklungen im Bereich Medienwirtschaft, Informations- und Wissensmanagement und Big Data werden gezielt aufgegriffen und im Lehr- und Forschungsprogramm berücksichtigt.

Der Studiengang Informationswissenschaft wird seit 1998 als Vollzeitstudiengang in Chur angeboten und seit 2002 als Teilzeit-Studiengang in Zürich. Seit 2010 rundet der Master of Science in Business Administration das Lehrangebot ab.

Der Arbeitsbereich Informationswissenschaft vereinigt Cluster von Forschungs-, Entwicklungs- und Dienstleistungspotenzialen in unterschiedlichen Kompetenzzentren:

- Information Management & Competitive Intelligence
- Collaborative Knowledge Management
- Information and Data Management
- Records Management
- Library Consulting
- Information Laboratory
- Digital Education

Diese Kompetenzzentren werden im Swiss Institute for Information Science (SII) zusammengefasst.

Impressum

Impressum

FHGR - Fachhochschule
Graubünden
Information Science
Pulvermühlestrasse 57
CH-7000 Chur

www.informationsscience.ch

www.fhgr.ch

ISSN 1660-945X

Institutsleitung

Prof. Dr. Ingo Barkow

Telefon: +41 81 286 24 61

Email: ingo.barkow@fhgr.ch

Sekretariat

Telefon: +41 81 286 24 24

Fax: +41 81 286 24 00

Email: clarita.decurtins@fhgr.ch