



Universität Regensburg

Philosophische Fakultät III
Sprach-, Literatur- und Kulturwissenschaften
Institut für Information und Medien, Sprache und Kultur (I:IMSK)
Lehrstuhl für Informationswissenschaft

Methoden der maschinellen Sprachverarbeitung
Modul: INF-M 31 (M.A.)
WS 2013/14
Leitung: PD Dr. Jürgen Reischer



der Prototyp eines Chatbots für Studenten und Studieninteressierte
der Informationswissenschaft und Medieninformatik

Alexander Bazo
Matrikelnummer: 1409575
Informationswissenschaft
4. Semester M.A.

Michaela Schlesinger
Matrikelnummer: 1457013
Informationswissenschaft
3. Semester M.A.

Jessica Rak
Matrikelnummer: 1455323
Informationswissenschaft
3. Semester M.A.

Abgegeben am 14. März 2014

Inhaltsverzeichnis

Inhaltsverzeichnis.....	2
Abbildungsverzeichnis.....	4
Tabellenverzeichnis.....	5
1 Hinweise	6
2 Einleitung – Vorgehensweise	7
3 State of the Art	9
3.1 Allgemein Chatbots.....	9
3.2 Luzie – Auslandsstudienberatungschatbot	14
3.3 Stella – Chatbot der SUB Hamburg.....	18
3.4 Albot – Chatbot der USB Köln.....	22
3.5 Fazit und Umlegung auf den eigenen Prototyp.....	25
4 Logofindung.....	27
5 Prolog-Schnittstelle	30
5.1 Datenbank.....	30
5.2 Musterstudenten.....	38
5.3 Mächtigkeit des Prolog-Codes.....	39
5.4 Ergänzungen zur formalen Gestaltung des Studiums	41
6 Datenerhebung	42
6.1 Studentische Umfrage im Kursbetrieb	42
6.2 Befragung von Schülern am Regensburger Hochschultag.....	46
6.3 Expertenbefragung im Sprachverarbeitungskurs.....	47
7 Kategorisierung	51
7.1 Kategorisierung durch Excel.....	51
7.1.1 Fragebogen	51
7.1.2 Hochschultag der Universität Regensburg	60
7.2 Kategorisierung durch das Tool Xmind	66
7.2.1 Fragebogen und Hochschultag	66
7.3 Dictionary	70
7.4 Stoppwörter vs. inhaltstragende Wörter.....	73

8	Prototyp	75
8.1	Anforderungen.....	75
8.2	Implementierung	79
8.3	Verwendung	88
8.4	Testergebnisse	91
9	Fazit und mögliche Erweiterungen	93
10	Literatur	97

Abbildungsverzeichnis

Abbildung 1: Luzie der Auslandsstudienberatungschatbot des DAAD	15
Abbildung 2: Luzie Scrollfunktion und Hyperlink am Beispiel Kategorie Student/in....	16
Abbildung 3: Luzie Auswahl an Emotionen des Avatars	17
Abbildung 4: Stella Einbettung in die Webseite	19
Abbildung 5: Stella Auswahl an Emotionen des Avatars	20
Abbildung 6: Stella Beispiel der Informationstiefe.....	21
Abbildung 7: Albot Integrierung in die Webseite und Pop-up-Fenster des Chatbots.....	22
Abbildung 8: Albot Auswahl an Emotionen des Avatars	23
Abbildung 9: Albot Ursprung seiner Lebensgeschichte.....	24
Abbildung 10: Albot Chatverlauf Rückführung.....	24
Abbildung 11: Logo UR Talking	27
Abbildung 12: Webseite Farbpalette	28
Abbildung 13: Datenbankschema Wissensbasierte Systeme	31
Abbildung 14: Datenbankschema Methoden der maschinellen Sprachverarbeitung	35
Abbildung 15 Prolog Student	40
Abbildung 16 Prolog Kursempfehlung.....	40
Abbildung 17: Fragebogen Personenverteilung	44
Abbildung 18: Fragebogen Grundkurse	45
Abbildung 19: Elise User-Interface	46
Abbildung 20: Xmind nach dem Studium	68
Abbildung 21: Dictionary 1 am Beispiel Beruf.....	70
Abbildung 22: Dictionary 1 Sortierung des Wörterbuches	71
Abbildung 23: Dictionary Reduzieren der Frage.....	71
Abbildung 24: Dictionary Formulierung des Endtemplates.....	72
Abbildung 25: Dictionary 2 am Beispiel Beruf.....	72
Abbildung 26: Stoppwort Vokabular allgemein.....	73
Abbildung 27: Stoppwort inhaltstragende Wörter.....	74
Abbildung 28: Prototyp Skizzierte Systemarchitektur des Gesamtsystems. Benutzerschnittstelle, Chatbot und Faktenbasis (Prolog) werden als einzelne und austauschbare Module realisiert, die über entsprechende Brücken miteinander kommunizieren.....	78
Abbildung 29: Prototyp Screenshot der Administrationsschnittstelle.....	81
Abbildung 30: Prototyp Schematischer Ablauf des Präprozessors. Die Benutzereingabe wird von einer beliebigen Anzahl an Modulen verarbeitet, um dann an den AIML- Prozessor übergeben zu werden. Die Reihenfolge der Module kann beliebig festgelegt werden.	86

Tabellenverzeichnis

Tabelle 1: Datenerhebung Allgemein	42
Tabelle 2: Datenerhebung studentische Umfrage im Kursbetrieb	43

1 Hinweise

Das hier beschriebene und prototypisch implementierte Chatbot-System kann über die URL <http://urtalking.ur.de> genutzt werden.

Der Quellcode der fertiggestellten Software-Schnittstelle ist über ein GIT-Repository erreichbar: <https://github.com/alexanderbazo/urtalking>.

Alle relevanten Dokumente, vorangegangenen Seminararbeiten und die Ergebnisse der Datenerfassung finden sich auf der Webseite <http://urtalking.ur.de/documentation.php>.

2 Einleitung – Vorgehensweise

Das Seminar *Methoden der maschinellen Sprachverarbeitung* unter Leitung von HERRN PD DR. JÜRGEN REISCHER beschäftigte sich im Wintersemester 2013/14 neben den Grundlagen der linguistischen Komponenten – Reguläre Ausdrücke, Syntax und Grammatik/Parsing als auch Sprach- und Grammatiktheorie – mit der technischen Komponente am Beispiel von Chatbots.

Die vorliegende Dokumentation *UR Talking – der Prototyp eines Chatbots für Studenten und Studieninteressierte der Informationswissenschaft und Medieninformatik* – setzt hierbei an der Schnittstelle der theoretischen Grundlagen an, um ein prototypisches, automatisiertes Auskunftssystem in Form eines Chatbots für Studenten und Studieninteressierte der Fachrichtungen Informationswissenschaft, Medienwissenschaft und Medieninformatik an der Universität Regensburg zu erstellen.

Als Grundlage werden zu Beginn der Seminararbeit die theoretischen Aspekte in der gängigen Literatur rund um einen Chatbot kompakt erläutert, wobei eine Momentaufnahme durch den State of the Art erstellt wird, um Chatbots im universitären Kontext aufzuzeigen.

Basierend auf diesen theoretischen Abschnitten erfolgt die praktische Umsetzung, woraufhin in dieser Ausarbeitung dargestellt wird, aus und durch welche Bausteine ein prototypischer Chatbot realisiert wird. Im Anschluss wird die Theorie rund um Chatbots eingegrenzt und in den universitären Kontext verlagert.

Als zentraler Baustein dient hierbei die logische Programmiersprache *Prolog*, wobei die beiden Seminararbeiten „Plan in Logik: Ein Studienberater. *Darstellung der Prüfungsordnung der Philosophischen Fakultäten der Universität Regensburg als Regelmenge für Prolog.*“ (BAZO und SPRÖD) und „*Gewichtung und Priorisierung der abhängigen Lehrveranstaltungen eines Empfehlungssystems für Studierende*“ (GROSS, RAK und SCHLESINGER) als Datenbasis fungieren, welche im Sommersemester 2013 im Seminar *Wissensbasierte Systeme* unter

Leitung von HERRN PROF. DR. RAINER HAMMWÖHNER erstellt wurden. Die vorhandene Datenbasis bzw. Datenbank des WBS-Seminars wird im Kontext der vorliegenden technischen Realisierung eines Studienberatungschatbots sinnvoll erweitert und angepasst. *Prolog* als wissensbasiertes System dient für den zu realisierenden Chatbot als Schnittstelle, welche die *Artificial Intelligence Markup Language* unterstützen und verbessern soll.

Für die Themenbearbeitung spielt außerdem die Datenerhebung eine essentielle Rolle, wobei im Rahmen dieser Umsetzung auf zwei verschiedene Arten der Datensammlung stattfindet. Zum einen ist die Erhebung der Daten durch eine Befragung von Schülern im Rahmen des Regensburger Hochschultages erfolgt, zum anderen ergeben sich Daten aus einer studentischen Umfrage während des Kursbetriebs in den drei relevanten Studiengängen Informationswissenschaft, Medienwissenschaft und Medieninformatik.

Die Verarbeitung der gesammelten Daten erfolgt in einem sich daraus ergebenden Schritt der Kategorisierung. Somit dienen die ermittelten Daten sowohl für die Themenfindung, als auch für die möglichen Verbalisierungsvarianten der Thematiken als Ausgangspunkt.

Die technische Umsetzung beschäftigt sich mit den verschiedenen Verarbeitungsmethoden für den Prototyp. Die bereits angesprochenen Themen des theoretischen Abschnittes, als auch der *Prolog*-Schnittstelle und der Datenerhebung tragen zur Erstellung des Prototyps bei. Die Verarbeitung der Daten, als auch eine vorgeschalteten Rechtschreibprüfung, Phrasenersetzung und Stoppwortreduktion dienen als entscheidende Elemente für die Konzeption des Gesamtkonstrukts bzw. des zu erstellenden Chatbots.

Ein Fazit und Ausblick runden die Dokumentation zur vorliegenden Arbeit *UR Talking – der Prototyp eines Chatbots für Studenten und Studieninteressierte der Informationswissenschaft und Medieninformatik* ab.

3 State of the Art

Beginnend mit einem theoretischen Teil werden die Grundlagen rund um einen Chatbot wiedergegeben und drei Chatbots vorgestellt, welche im universitären Kontext genutzt bzw. erstellt wurden. Hierbei handelt es sich um den Auslandsstudienberatungschatbot *Luzie*, die elektronische Informationsassistentin *Stella* und deren Bruderprojekt, der virtuelle Informationsassistent *Albot*.

3.1 Allgemein Chatbots

„Chatbots ermöglichen dem Menschen eine auf natürlicher Sprache basierende Interaktion mit dem Computer, egal ob diese via Tastatur oder Stimmerkennung (voice recognition) erfolgt. Sie greifen dabei auf eine hinterlegte Wissensbasis (Knowledge-Datenbank) zu, in der sie durch das Aufspüren von Übereinstimmungen gestellter Fragen mit dem vorhandenen, von Programmierer erstellten Fragenbestand zugehörige Antworten bzw. Aktionen auswählen und dem Fragenden ausgeben.“

(Braun, 2003, Seite 21)

Der Terminus Chatbot setzt sich aus den englischen Worten *chat* – im Sinne von Plaudern – und dem Wort *bot* – kurz für Robot – zusammen und ist in diesem Sinne selbst erklärend. Im Sprachgebrauch ist unter anderem auch Chatterbots eine übliche Bezeichnung und lt. BRAUN (2003, Seite 21) als Interface-Agents den Software-Agents zuzuordnen.

„Agents provide expertise, skill, and labor. They must of necessity be capable of understanding our needs and goals in relation to them (either explicitly or implicitly), translating those goals into an appropriate set of actions, performing those actions, and delivering the results in a form that we can use. They must also know when further information is needed from us and how to get it.“

(Laurel, 1990, Seite 206)

Chatbots können den Software-Agenten zugeordnet, als auch selbstverständlich einem Teilbereich der Künstlichen Intelligenz anerkannt zu werden.

Im Folgenden werden kompakt die historischen Meilensteine, als auch allgemeine Fakten wiedergegeben, um einen ersten Einblick in die Thematik der Chatbots zu erhalten. Die Geschichte der Chatbots beginnt mit der Erforschung der Künstlichen Intelligenz bzw. Artificial Intelligence (im weiteren KI oder AI) und dem ersten Chatbot *ELIZA*, anschließend nimmt die Arbeit Bezug auf die gegenwärtige Lage von Chatbots.

Ein geschichtlicher Rückblick:

Die Künstliche Intelligenz und die Entstehung des ersten Chatbots ELIZA

Historisch gesehen ist vorerst das Feld der Künstlichen Intelligenz zu nennen, mit welcher sich die Forschung ab Mitte der 1950er Jahre beschäftigte (vgl. Crevier, 1994, Seite 46). Im Rahmen der Forschungsinteressen der neuen Disziplin, als auch in der Blütezeit dieser Ära entstand im Jahre 1966 der erste und wohl bekannteste Chatbot *ELIZA* von JOSEPH WEIZENBAUM. WEIZENBAUM wurde trotz bzw. aufgrund der Popularität und der Euphorie rund um *ELIZA* zu einem der größten Kritiker rund um die AI (vgl. Braun, 2003, Seite 22ff).

„Trotz aller Zweifel in bezug auf die kurzfristigen Aussichten der KI war Weizenbaum bereit zuzugestehen, daß sie letztlich machbar sein werde; doch wandte er sich aus moralischen Gründen gegen die künstliche Intelligenz. Erregt hatte seinen Zorn paradoxerweise seine Erfahrung mit einem der amüsantesten und meistdiskutierten KI-Programme [Programm Weizenbaums, welches das Spiel Fünf hintereinander spielen konnte, in gewisser Weise ein Vorgänger von *ELIZA*], die je geschrieben worden waren.“

(Crevier, 1994, Seite 183)

WEIZENBAUM kritisierte vor allem die nachhaltige Täuschung des humanen Kommunikationspartners und das fehlende, menschlichen Denkvermögen der Maschine/Programmes.

„Der Mensch, erklärte er [Joseph Weizenbaum], lasse sich nicht auf eine informationsverarbeitende Maschine reduzieren, denn ein Organismus sei weitgehend durch die Aufgaben definiert, denen er sich gegenübersteht. Wir bewältigen Probleme, die keine Maschine je lösen könnte. Man-

che Vorstellungen würden Maschinen auf alle Zeiten unverständlich bleiben, weil sie unser Ziele nicht teilen. Er kritisierte die Bemühungen der KI, englische Sätze in semantische Strukturen umzuwandeln, und fragte, wie ein solches Konstrukt je die verzweifelte Sehnsucht eines jungen Mannes nach Liebe ausdrücken vermöchte...”

(Crevier, 1994, Seite 191)

so DANIEL CREVIER über WEIZENBAUMS Ansicht über die Unmoral der KI.

Chatbots heute

ALEXANDER BRAUN bezeichnet die Anfänge der 1980er als Höhepunkt des Forschungsfeldes KI, wobei dessen Boom Mitte des Jahrzehnts abfiel. Erst Anfang der 1990er Jahre erschienen die Chatbots wieder auf den Bildflächen bzw. Bildschirmen. Hierbei nennt BRAUN die Interaktion in natürlicher Sprache, die aktive Gesprächsführung und die Informationsstrukturierung als zentrale Punkte und Eigenschaften, welche die Implementierung eines Chatbots befürworten (vgl. Braun, 2003, Seite 27ff).

„Wenn Personen außerhalb der Computerdomäne in der Lage sein sollen, in relevanter Weise mit Computern zu interagieren, dann müssen entweder sie Computersprachen oder der Computer ihre Sprache lernen.“

(Weizenbaum, 1977, Seite 243)

Die Nutzung eines Chatbots erfolgt über natürliche Sprache, wodurch keine formalisierte, kontextabhängige (softwaretechnische Umgebung) Sprache für den User zu erlernen ist. Diese minimale Erlernungsphase und Einfachheit spricht dem Chatbot bzw. der Effizienz der Interaktion zu. Somit können Fragen in natürlicher Sprache formuliert werden, im Gegensatz zu den sonst bei Suchmaschinen üblichen Suchtermen und Vorschlagwortungen oder auch dem Suchmaschinenoutput in Form einer Ansammlung verschiedener Links. Ein Nutzer erhält im Gespräch mit einem Chatbot auf eine konkrete Frage eine direkte Antwort. Neben der Problematik der Suchmaschine nennt BRAUN die FAQ-Listen, welche zeitaufwendig zu durchforsten sind und einen starken Charakter der Allgemeinheit mit sich bringen (vgl. Braun, 2003, Seite 28f).

Die Eigenschaft der aktiven Gesprächsführung eines Chatbots ist ein weiteres Kriterium, welches BRAUN als positiv anerkennt. Durch die Dialogführung und aktive Teilnahme des Chatbots am Gespräch ist es möglich, den menschlichen Kommunikationspartner zu unterstützen. Dadurch wird das Gespräch durch das System zielführend geleitet, wodurch Chatbots über reine Frage-Antwort-Systeme hinausgehen (vgl. Braun, 2003, Seite 31).

Als drittes Kriterium ist die Informationsstrukturierung zu nennen, wobei BRAUN die trade-off-Problematik von der Breite und Tiefe von Webseiten anspricht. So werden Webseiten mit wenigen Kategorien meist sehr tief und die grobe Kategorienbildung kann unverständlich werden. Jene Informationstiefe impliziert, dass der Nutzer viele Klicks benötigen kann, um an die relevante Information zu gelangen. Abhilfe leisten hierbei zum Beispiel homepage-interne Suchmaschinen, welche in der Webseite eingebettet sind. Suchmaschinen setzen jedoch immer ein gewisses technisches Wissen im Sinne der booleschen Operatoren voraus, um einen Suchraum und die Resultate sinnvoll einzugrenzen. Ein weiteres strukturelles Problem auf Webseiten ist die fehlende oder verschachtelte Definition einer Zielgruppe, wodurch ein spezifisches Informationsangebot gemieden wird. Chatbots hingegen können navigationsunterstützend wirken, insofern diese auf konkrete Fragen antworten und parallel weiterführende Informationen z. B. durch Aufrufen von Webseiten im Hintergrund oder auch zielgerechte Hyperlinks anbieten (vgl. Braun, 2003, Seite 32ff).

Die Verwendbarkeit eines Chatbots impliziert dennoch auch die Akzeptanz des Users, welcher die Nutzbarkeit beurteilt und somit auch entscheidet. Um der Ablehnung des Systems entgegenzuwirken, werden Chatbots häufig durch einen Avatar unterstützt.

„Teilweise sind Chatbots personifiziert, d.h., sie werden als Menschen, Tiere oder Fabelwesen auf dem Bildschirm dargestellt (Agenten-Metapher), teilweise sind sie reine Texteingabeboxen.“

(Braun, 2003, Seite 21)

Der Aspekt der Personifizierung eines Chatbots wird bei RICKEL UND JOHNSON nicht nur aufgegriffen, sondern auch im Sinne der Akzeptanz des Systems befürwortet.

"Virtual reality offers a rich environment for multimodal interaction among agents and humans. Like standard desktop dialogue systems, agents can communicate with humans via speech, using speech synthesis and recognition software. As in previous simulation-based training systems, a simulator controls the behavior of the virtual world. Agents can perceive the state of the virtual world via messages from the simulator. However, an animated agent that cohibits a virtual world with students has a distinct advantage over previous disembodied tutors: the agent can also communicate nonverbally using gestures, gaze, facial expressions, and locomotion."

(Rickel & Johnson, 2000, Seite 95)

Mittlerweile haben die Chatbots Verwendungsmöglichkeiten in verschiedenen Teilbereichen und somit den Einzug in unser Leben gefunden – wie zum Beispiel *Anna*, die virtuelle Mitarbeiterin von IKEA (siehe auch Onlinelinks, Literaturverzeichnis).

Durch die TU Dortmund mit *ASKademicus* (siehe auch Onlinelinks, Literaturverzeichnis), dem ersten Chatterbot einer Universitätsbibliothek, fanden die Agents einen Platz im universitären Leben. Seitdem nutzen immer mehr Bibliotheken virtuelle Mitarbeiter hinsichtlich der Technik von Chatbots als erweiternden Service ihres Online-Angebotes. Chatbot-Projekte wie unter anderem

- *Stella* (,welche wie auch *ASKademicus* im Jahre 2004 veröffentlicht wurde),
- *INA* (Internet-Navigations-Assistent)
(siehe auch Onlinelinks, Literaturverzeichnis)
- oder auch *Albot* (Wissensbasis von *Stella* übernommen)

verfeinerten das Bibliotheksangebot, und somit entstand der erste, indirekte als auch direkte Bezug zu Universitäten (vgl. Christensen, 2008, Seite 18).

Chatbots im universitären Kontext

Bereits vorhandene Studienberatungssysteme an Universitäten und Hochschulen wurden während der Recherchephase nicht gefunden, dennoch gibt es bereits Chatbots,

welche in den universitären Kontext eingebettet sind und genutzt werden. Im Folgenden werden drei ausgewählten Chatterbots – *Luzie*, *Stella* und *Albot* – kompakt erläutert.

3.2 Luzie – Auslandsstudienberatungschatbot

Als erstes wird hierbei der Chatbot *Luzie* im universitären Kontext vorgestellt, bei dem es sich um einen Auslandsstudienberatungschatbot der DAAD – DEUTSCHER AKADEMISCHE AUSTAUSCHDIENST – handelt.

„Der Deutsche Akademische Austauschdienst (DAAD) ist die weltweit größte Förderorganisation für den internationalen Austausch von Studierenden und Wissenschaftlern.“

(DAAD, 2013, entnommen 2014)

Der DAAD folgt dem Leitbild „*Wandel durch Austausch...*“ (vgl. DAAD, 2013, entnommen 2014) und hat sich unter anderem das Ziel der Förderung im Sinne des akademischen Austausches gesetzt. Die virtuelle Beraterin namens *Luzie* ergänzt die Kampagne „*go out! studieren weltweit*“.

„Die Kampagne "go out! studieren weltweit" wurde 2006 vom Bundesministerium für Bildung und Forschung (BMBF) und dem Deutschen Akademischen Austauschdienst (DAAD) initiiert, um die Auslandsmobilität deutscher Studierender zu erhöhen.“

(go-out Studieren weltweit, entnommen 2014)

Die Chatbot-Beratung *Luzie* ging 2007 online und ist auf der Homepage des DAAD eingebettet, als auch über die Webpräsenz der Kampagne „*go-out*“ verlinkt.

Der Gesprächsverlauf selbst findet in einem eigenen Fenster mit der URL <http://bot.daad.de/luziecgi-bin/luzie.cgi> statt. Sie bietet einen virtuellen Beratungsservice – in deutscher Sprache – für Studierende an und steht bei Fragen zum Studium im Ausland dem Chatpartner zur Seite. Der Online-Beratungsservice rund um *Luzie* hilft

primär bei der Vorbereitung und Planung eines Auslandsaufenthaltes und bietet Zugriff im Sinne von Verlinkungen auf das Angebot des DAAD.

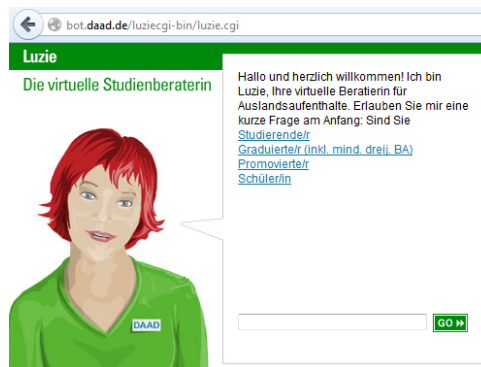


Abbildung 1: Luzie | der Auslandsstudienberatungschatbot des DAAD

Technische Einzelheiten um *Luzie* wurden kaum publiziert, lediglich Presseartikel geben einen kleinen Einblick hinter die Kulissen des Chatterbots. So erklärte lt. FAZ ROLF GIERING, – Sachbearbeiter im DAAD-Informationsreferat und einer der „Väter“ von *Luzie* – deren Verfahren wie folgt:

„Die Antworten werden durch eine Wissensdatenbank generiert, deren Informationsbasis fortlaufend ergänzt und erweitert wird.“

(FAZ, 2007, entnommen 2014)

Laut diesem Bericht belaufen sich die Anfragen an die virtuelle Beraterin einige Wochen nach Veröffentlichung des Chatbots auf durchschnittlich 15.000 pro Monat (vgl. FAZ, 2007, entnommen 2014).

Dem Jahresbericht 2007 des DAAD konnte lediglich Folgendes entnommen werden:

„Virtueller Beratungsservice „Luzie“ für deutsche Studierende

Seit Ende August beantwortet die virtuelle Beraterin „Luzie“ im Internet Fragen zum Auslandsstudium. Auf der Basis einer Wissensdatenbank berät sie rund um Studium, Praktikum und Sprachkurse im Ausland und nennt die jeweiligen Förderungsmöglichkeiten. Damit unterstützt sie die DAAD- und BMBF-Kampagne „go out! studieren weltweit“.

(DAAD Jahresbericht von 2007, 2008, entnommen 2014, Seite 55)

Luzie selbst erklärt ihre Technik wie folgt:

„Die Software zur Erstellung eines *Virtual Dialogue Agents* arbeitet mit einer großen Anzahl von Regeln, die aus "Erkennungen" und "Antworten" bestehen. Die Benutzereingaben werden mit den Erkennungen verglichen und die Regel, die am besten passt, wird angewendet, um die entsprechende Antwort auszugeben.“

(LUZIE)

Informationen, wie der strukturelle Aufbau und die technische Umsetzung zum Chatbot *Luzie*, sind kaum auffindbar. Dennoch ist festzuhalten, dass die virtuelle Studienberaterin sich großer Beliebtheit erfreut, was aus dem genannten Zahlenmaterial der FAZ ersichtlich ist.

Eine Unterhaltung mit *Luzie* beginnt stets mit der Frage (siehe Abbildung 1: 'Luzie | der Auslandsstudienberatungschatbot des DAAD') des eigenen Status, welche dem Chatpartner bzw. User die Möglichkeit gibt, zwischen den vier vorgegebenen Kategorien Student/in, Graduierte/r (inkl. mind. dreijährigen BA), Promovierte/r oder Schüler/in zu wählen.



Abbildung 2: Luzie | Scrollfunktion und Hyperlink am Beispiel Kategorie Student/in

Während der Nutzung des Chatbots gab es einige Auffälligkeiten, welche im Folgenden kurz benannt, jedoch in diesem Sinne nicht bewertet werden sollen. *Luzie* gibt häufig lange Antworten, welche nur durch scrollen im Textfeld komplett lesbar sind. Des Weiteren bietet *Luzie* stetig Links an, damit der User sich über die eigentliche Antwort der virtuellen Studienberaterin bei Bedarf weiter informieren kann. Jener Scrollbalken, als

auch die Problematik der Fenstergröße ist in Abbildung 2: 'Luzie | Scrollfunktion und Hyperlink am Beispiel Kategorie Student/in' ersichtlich. Außerdem ist festzuhalten, dass das Gespräch nicht wie üblich in einem Verlauf für den Nutzer sichtbar ist, sondern stets die aktuelle Antwort von *Luzie*, sowie auch ein Eingabefeld für die nächste Frage des Users. *Luzie* leitet das Gespräch im Sinne des gewünschten Kontextes im Hauptaugenmerk auf ein Auslandsstudium bzw. -aufenthalt. Nutzereingaben wie das Studienfach oder das angepeilte Land für den Auslandsaufenthalt werden gespeichert und im weiteren Gesprächsverlauf vom Chatbot benutzt. *Luzie* leitet aufgrund dieser Daten den menschlichen Chatpartner häufig auf Webseiten weiter. Der Gesprächsverlauf mit *Luzie* bleibt bestehen, es öffnet sich im Hintergrund jedoch die vorgeschlagene Webseite. Dennoch ist anzumerken, dass *Luzie* nicht direkt auf das Gesprächsthema eingeht bzw. weiterleitet, sondern auf die Hauptseiten, wodurch der Nutzer selbst die thematische Navigation auf der geöffneten Seite durchführt. Sofern *Luzie* während des Gesprächs einen Link anbietet, wird dieser nur durch Klicken des Hyperlinks vom User selbst geöffnet. Die virtuelle Beraterin öffnet den Link – in der aktuellen Version – durch Texteingabe nicht.



Abbildung 3: Luzie | Auswahl an Emotionen des Avatars

Für die Personalisierung wurde für *Luzie* ein weiblicher, rothaariger, blauäugiger Avatar erstellt, welcher ein grünes Tshirt mit einem DAAD-Abzeichen trägt. Die Optik des Avatars wirkt dadurch seriös und menschlich in seiner Beratungsfunktion. Während des Chatverlaufs verändert *Luzie* die Gesichtszüge (siehe Abbildung 3: 'Luzie | Auswahl

an Emotionen des Avatars') ihres Avatars, wobei keine genaue Anzahl ersichtlich ist, wieviel Emotionen der Avatar im Kontext des Gespräches wiedergeben kann.

Abschließend ist festzuhalten, dass *Luzie* im Sinne ihrer Funktionalität den Anforderungen entspricht und dem User in den Grenzen ihrer Möglichkeiten weiterhelfen kann bzw. es versucht. Leider sind kaum technische Daten rund um ihr Projekt greifbar.

Luzie erweist sich im universitären Kontext als ein fähiger und (basierend auf den Nutzerzahlen) auch angenommener Chatbot, welcher seiner Beratungsfähigkeit treu bleibt. Auch Smalltalk mit *Luzie* ist möglich, hält sich jedoch in Grenzen, da die Beraterin den Nutzer immer wieder auf ihre eigentliche Tätigkeit hinweist und den Gesprächsablauf somit in die gewünschten Thematiken führt.

Das Fehlen der Informationen rund um die technischen Hintergründe und Erstellung von *Luzie* ist abschließend erneut zu vermerken.

3.3 Stella – Chatbot der SUB Hamburg

Der Chatbot *Stella* ist das zweite ausgewählte Projekt, welches im Sinne der vorliegenden Arbeit *UR Talking – der Prototyp eines Chatbots für Studenten und Studieninteressierte der Informationswissenschaft und Medieninformatik* vorgestellt wird. Hierbei handelt es sich um eine elektronische Informationsassistentin Stella, ein Onlineangebot der Staats- und Universitätsbibliothek Hamburg. Hauptthematik des Chatbots sind die Stabi - STAATS- UND UNIVERSITÄTSBIBLIOTHEK HAMBURG CARL VON OSSIETZKY - als auch deren Informationsangebote.

„Stella gibt Tipps zur Auswahl von Katalogen und Datenbanken und erläutert Zugangsbedingungen zu elektronischen Volltexten. Damit hilft sie Hamburger Studierenden, eine wichtige Schlüsselqualifikation zu entwickeln: informationskompetent zu werden, also zu wissen, wo man wie nach welcher Information sucht und diese dann auch beschafft.“

(SUB Hamburg, 2004, entnommen 2014)

Das Projekt *Stella* wurde von 2003 bis 2004 entwickelt und ab 2004 bis 2005 sowohl evaluiert als auch optimiert. Die Software ist von *Novomind*, während die technische Entwicklung *Kiwi* und *Scientec* unterlag. Lt. der SUB HAMBURG wurden für den Chatbot 21 Themenmodule mit ca. 2900 Regeln hinterlegt, wovon 1900 bibliotheksbezogen sind und der Rest sich mit der Smalltalkfähigkeit *Stellas* beschäftigt.

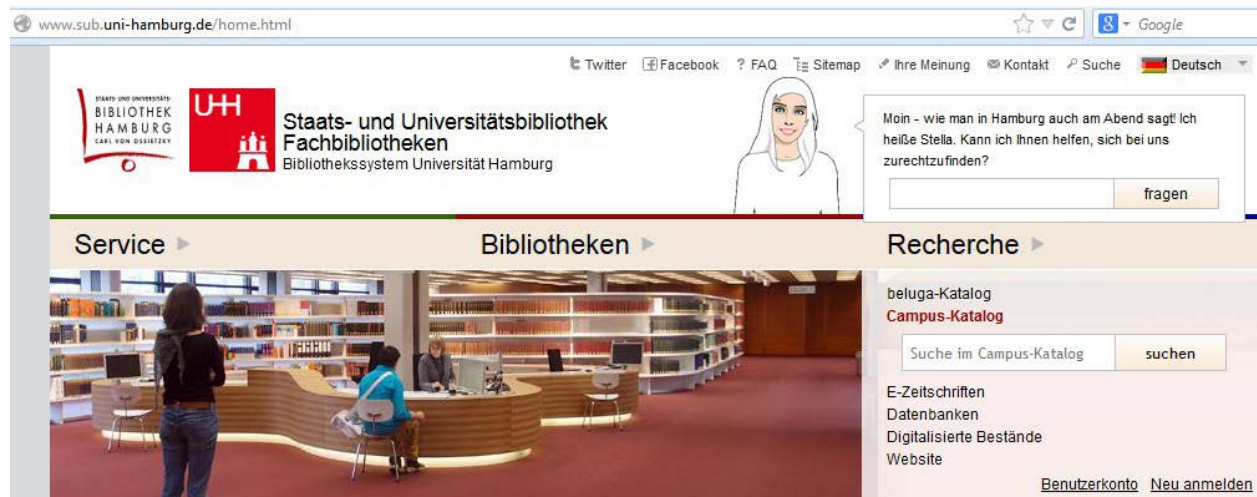


Abbildung 4: Stella | Einbettung in die Webseite

Durchschnittswerte aus einer siebenmonatigen Beobachtungsphase ergaben eine Anzahl von 63100 Dialogen mit *Stella*, wobei ein Gespräch eine durchschnittliche Länge von vier Minuten andauert. Hierbei sind 350 bis 400 Dialoge an Werktagen zu verzeichnen und 120 Gespräche an Wochenenden. Bemerkenswert sind die Nutzerzahlen in Bezug auf den Kontext bzw. den Zweck des Chatbots. So werden in ca. 60 Gesprächen täglich Informationskompetenzen vermittelt und 83 Prozent der geführten Gespräche haben rein bibliotheksbezogene Inhalte. In der Evaluations- und Optimierungsphase konnte die Fehlerquote von über 30 Prozent auf deutlich unter 20 Prozent gesenkt werden (vgl. SUB Hamburg, 2005, entnommen 2014).

Stella wird personalisiert durch einen weiblichen, weißhaarigen, blauäugigen Avatar, welcher zehn verschiedene Stimmungen nachbilden kann.



Abbildung 5: Stella | Auswahl an Emotionen des Avatars

Die Funktionsfähigkeit des Chatbots **Stella** wird im Folgenden anhand eigener Dialoge kurz erläutert, wobei deren festgestellte Auffälligkeiten erneut vorgestellt, jedoch nicht bewertet werden.

Die Begrüßung durch **Stella** beginnt mit einer offenen Frage, wodurch dem Nutzer zu Beginn des Dialogs jegliche Thematik freigestellt wird. Dennoch lenkt die virtuelle Informationsassistentin die Gespräche und Dialoge während eines Smalltalks stets zurück auf den eigentlichen Zweck des Chatbots. Die Antworten von **Stella** sind kompakt, wobei bei längeren Antwortmöglichkeiten der Chatbot dem User die Informationstiefe überlässt (siehe Abbildung 6: 'Stella | Beispiel der Informationstiefe'). **Stella** gibt dem User eine Antwort und bietet dem Chatpartner die Möglichkeit, Weiteres zu erfahren, in dem sie – dialogähnlich – den User nach weiterem Informationsbedarf in der vorliegenden Thematik befragt.



Abbildung 6: Stella | Beispiel der Informationstiefe

Somit vermeidet der Chatbot längere Texte, welche zum Beispiel nur durch Scrollen oder das Betätigen von Links lesbar wäre. Auffällig ist außerdem, dass *Stella* sehr selten Links anbietet. Sie verfügt über ein breites Smalltalkwissen, wodurch ein guter Dialog mit ihr möglich ist. Dennoch ist zu bemerken, dass die virtuelle Hamburgerin stets auf den eigentlichen Zweck des Chatbots zurückführt, den User in der Gesprächsführung unterstützt und somit dem Benutzer Informationskompetenzen lernt. Vermutlich besitzt *Stella* eine Rechtschreibprüfung, welche einerseits sehr nützlich für den User ist, andererseits Wörter, welche nicht vordefiniert wurden und der Rechtschreibprüfung unbekannt sind, in deren Sinn abändert bzw. komplett ersetzt.

Abschließend ist festzuhalten, dass *Stella* in dem Testgespräch sowohl smalltalkfähig war, als auch ihre Schwerpunkte im Sinne von Hinweisen und Nutzung des elektronischen Katalogs, Zeitschriften und Datenbanken vermittelte. Außerdem war der Home-

page zu entnehmen, dass das Konzept von *Stella* an Dresden verkauft wurde und außerdem deren Wissensbasis an die Universität Köln bzw. deren Bibliothek übertragen wurde, was im folgenden Unterkapitel vorgestellt wird.

3.4 Albot – Chatbot der USB Köln

Bei dem dritten und letzten Projekt, welches im Rahmen dieser Ausarbeitung vorgestellt werden soll, handelt es sich um den Chatbot der UNIVERSITÄTS- UND STADTBIBLIOTHEK KÖLN namens *Albot*. Der virtuelle Informationsassistent *Albot* wurde nach eigenen Angaben am 29. September 2009 online gestellt und besitzt als Grundstock die Wissensbasis des Schwesterprojekts *Stella* (vgl. UB KÖLN, entnommen 2014).

„Mit Albot können Sie neben seinem eigentlichen Zweck, Fragen rund um die USB Köln und ihre elektronischen Informationsangebote zu beantworten, aber auch mal einen Plausch über andere Themen halten. Albot hilft Ihnen zu lernen, wo man wie nach welcher Information sucht und diese dann auch beschafft.“

(UB KÖLN, entnommen 2014)

Somit ist festzuhalten, dass *Albot* einen homogenen Zweck wie dessen Schwester *Stella* erfüllt: die Unterstützung und Beratung, als auch die Vermittlung von Informationskompetenzen.

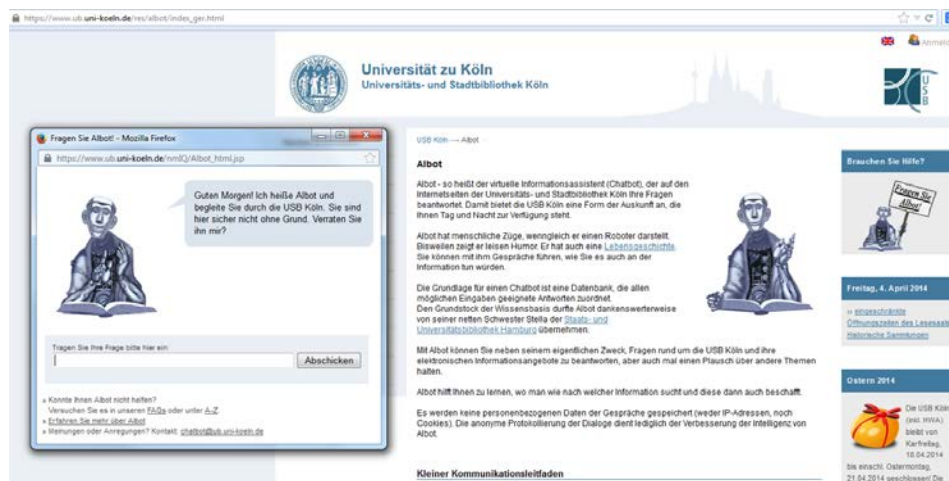


Abbildung 7: Albot|Integrierung in die Webseite und Pop-up-Fenster des Chatbots

Der Chatbot wurde in die Homepage der Universitäts- und Stadtbibliothek Köln integriert, jedoch bei Nutzung des Chatbots öffnet sich ein separates Pop-up-Fenster für einen Gesprächsverlauf. Im Pop-up (siehe Abbildung 7: 'Albot | Integrierung in die Webseite und Pop-up-Fenster des Chatbots') befindet sich der Avatar des Chatbots, dessen Outputfenster, eine Input-Möglichkeit für den Chatpartner, als auch einige Angebote durch Verlinkungen (FAQs, A-Z, Erfahren Sie mehr über Albot und Kontaktdaten).

Albot besitzt außerdem einen Kommunikationsleitfaden, als auch Beispielfragen, welche die Akzeptanz durch den Nutzer vereinfachen und die Fehlerquote des Chatbots vermindern soll. Der Leitfaden für die Kommunikation mit **Albot** beinhaltet unter anderem, dass **Albot** nicht allwissend sein kann, nicht alle Formulierungsvarianten kennt, Antworten in einem Fortsetzungsteil verstecken könnte (weiterführende Links), oder dass er Internetseiten im Hintergrund öffnet, welche die Antwort der Frage beinhalten können. Des Weiteren weist der Leitfaden den Nutzer darauf hin, dass es sich um ein normales Gespräch handelt, dennoch sollte der User zum Beispiel Doppelfragen, Sonderzeichen oder auch Schachtelsätze vermeiden und in einem gewissen Maß auf die Rechtschreibung achten.

Bei dem Avatar handelt es sich um einen Blechroboter, welcher als Gelehrter über einem Buch sitzt. **Albot** kann dem User während des Gesprächs ebenfalls eine ungewisse Zahl an Emotionen (siehe Abbildung 8: 'Albot | Auswahl an Emotionen des Avatars') ausstrahlen, wobei der Avatar weniger seine Mimik verändert, als vielmehr seine Gestik dem Gespräch anpasst.



Abbildung 8: Albot | Auswahl an Emotionen des Avatars

Außerdem ist zu bemerken, dass das Projekt *Albot* im Gegensatz zu den anderen beiden Projekten eine direkte Lebensgeschichte besitzt, welche auch im Chatbot hinterlegt wurde (vgl. UB KÖLN, entnommen 2014).



"[...] Neben seiner Lehrtätigkeit in Philosophie und Theologie beschäftigte sich Albert mit Naturwissenschaften und mit Technik. Einer von den berühmtesten Schülern des hl. Albert hier in Köln war Thomas von Aquin. Thomas war viel weniger an der Technik interessiert als Albert. Damals hatte Albert einen Roboter gebaut. Dieser Roboter stand in seinem Arbeitszimmer. Als Thomas einmal das Arbeitszimmer betrat und diesen Roboter sah, bekam er es mit der Angst zu tun. Als er merkte, dass der Roboter sich nicht bewegte, ging er hin und berührte ihn. Doch da fiel der Roboter zu Boden und war kaputt. Thomas sah nun, was er getan hatte, und ging zu Albert, um sich zu entschuldigen. Diese Szene sehen wir auf unserem Fenster. [...]"

Quelle: [Dominikanerkonvent Sankt Andreas](#)

Abbildung 9: Albot | Ursprung seiner Lebensgeschichte

Auffälligkeiten während eines Testgesprächs mit *Albot* waren – wie bereits bei *Stella* – die Smalltalkfähigkeit des Chatbots, als auch deren Rückführung des Users auf den eigentlichen Zweck und die Hauptthematiken des Informationsassistenten. Außerdem ist festzuhalten, dass der Chatbot (wie bereits die beiden Projekte *Luzie* und *Stella*) keinen Chatverlauf anbietet, jedoch *Albot* eine Zurückfunktion implementiert bekommen hat. Die Funktion dieser Zurückführung wird von *Albot* in Abbildung 10:

‘Albot | Chatverlauf Rückführung’ erläutert.



Leider ist es technisch nicht möglich, einen Zurück-Button einzurichten. Aber es gibt einen Trick, falls Sie die letzte Antwort nur nochmals nachlesen möchten: Halten Sie die Alt-Taste gedrückt und nutzen Sie den Links-Pfeil auf Ihrer Tastatur. Aber bedenken Sie: Ich bin gedanklich schon eine Antwort weiter und kann nur auf die 'richtige' letzte Antwort eingehen.

Abbildung 10: Albot | Chatverlauf Rückführung

Des Weiteren ist *Albot* in der Lage, Verlinkungen nicht nur durch die Klickfunktion des Users zu öffnen, sondern auch durch die textuelle Eingabe des verlinkten Wortes eine Öffnung des Themengebiets herbeizuführen. *Albot* gibt kompakte Informationen, wel-

che durch Hyperlinks angefüllt werden, jedoch die Selbstbestimmung der Informationstiefe durch den Nutzer bestehen bleibt. Somit wird bei *Albot* auch die Scrollfunktion vermieden. Außerdem weist der Chatbot den User auf die Öffnung eines neuen Fensters bezüglich des Chatverlaufs hin.

Der Gesprächsversuch mit *Albot* zeigte resümierend auf, dass der Chatbot den Zweck als Informationsassistent rund um die Thematiken der USB KÖLN erfüllt und auch in einem gewissen Maß Smalltalk führen kann. Der Avatar ist auffällig durch dessen Erscheinungsbild als Roboter und seiner Gestikulationen, wodurch das Gefühl der Menschlichkeit verständlicherweise und selbsterklärend etwas gedrückt wird.

3.5 Fazit und Umlegung auf den eigenen Prototyp

Bei der Betrachtung der drei Projekte ist festzuhalten, dass die Chatbots mittlerweile Einzug in die Bibliothekssysteme gefunden haben und somit auch einen universitären Kontext erlangen. *Luzie* ist hierbei von den beiden anderen Projekten und dem Kontext abzugrenzen, da *Luzie* abseits eines Bibliothekensystems erstellt wurde und deren Zweck dem Auslandsaufenthalt von Studenten gilt. Festzuhalten ist somit die Funktionsfähigkeit und auch Zweckmäßigkeit von Chatbots an den Universitäten.

Essentiell ist jedoch die Abgrenzung des Zwecks eines Chatbots, um klare Anforderungen an den virtuellen Informationsassistenten zu definieren. Dieser Definitionsbedarf soll die Erstellung des Prototypen erleichtern und den Chatbot in den gewünschten Kontext einbetten. Für die vorliegende Seminararbeit ist es deshalb von Bedeutung, den Zweck und die Mächtigkeit des Prototyps – namens *Elise* – von UR TALKING darzulegen, um deren Vokabular, als auch deren Umfang von Antwortmöglichkeiten abzudecken.

Der erste Prototyp von Elise soll somit für die Pioniererstellung ein Vokabular erhalten, welches auf die Datensammlung (siehe Datenerhebung) zurückzuführen ist. Im gegeb-

nen Kontext des Studienberatungssystems soll sich der Chatbot mit Fragen rund um das Studium und dabei speziell rund um die Studiengänge Informationswissenschaft, Medieninformatik und Medienwissenschaft beschäftigen und vor allem die drei Hauptzielgruppen Studenten, Studieninteressierte und Wechsler unterstützen. Der Chatbot von UR TALKING soll hierzu in eine separate URL-Seite eingebunden und somit nicht direkt in die Fakultätshomepage eingebettet werden. Dies liegt darin begründet, dass es sich aktuell um die erste Fassung des Chatbots handelt und außerdem der Vorteil darin gesehen wird, dass dem User der komplette Chatverlauf somit ersichtlich ist. Dadurch wird das Gespräch für den User nachvollziehbarer und menschlicher, da Chats im Allgemeinen den Gesprächsverlauf des aktuellen Dialogs bereitstellen. Durch Hyperlinks soll dem Nutzer selbst die Möglichkeit nach der Informationstiefe überlassen werden, um dessen Information-Need individuell zu befriedigen. Festzuhalten ist außerdem, dass die Personalisierung durch einen Avatar den Rahmen der vorliegenden Dokumentation und des Projektes sprengen würde. Um der Assoziation von klassischen Suchmaschinen zu entgehen, wäre somit ein Avatar wünschenswert, welcher eine Personifizierung verstärkt und der reinen Texteingabebox menschliche Züge verleiht (vgl. Braun, 2003, Seite 31). Die Erstellung eines Avatars wäre somit ein logischer und begrüßenswerter Schritt für die Weiterführung des Projektes.

4 Logofindung

Um den Chatbot in die universitäre Umgebung einzubinden und das Projekt als universitäres zu kennzeichnen, fiel die Entscheidung, ein Logo zu erstellen. Die Ideenfindung ging hierbei über komplexere Ideen – wie zum Beispiel die Umwandlung des originalen Pacman in einen informationsjagenden Punktesammler – bis hin zu einförmigen Ideen. Begründet in deren Einfachheit und der damit einhergehenden Klarheit entschieden HERR PD DR. REISCHER und das Projektteam sich für die Grundform einer Sprechblase. Aufgrund der Verbundenheit mit der Fakultät der Informationswissenschaft, welche sich seit mehreren Semestern mit der NavigationsApp UR WALKING beschäftigt, übernahm man das Wortspiel. Somit entstand der Grundgedanke des Logos, welches angelehnt an das UR WALKING-Logo (vgl. Rzymkowski, Salomon, Rak, Hamedinger, Schlesinger, 2011) eine UR TALKING-Form erhalten sollte und die Zusammengehörigkeit der Fakultät somit wiedergibt.

Nach mehreren Veränderungen im Design und der Farbwahl konnte ein passendes Logo entwickelt werden.



Abbildung 11: Logo | UR Talking

Die Farbwahl fiel auf ein dezentes Orange und dem typischen Uni-Gräuton aus dem Corporate Design (siehe auch Corporate Design der Universität Regensburg: <http://www.uni-regensburg.de/corporate-design/>).

Erweitert wird die Farbpalette hierbei durch ein Weinrot, welche nicht im Logo, jedoch auf der Webseite verwendet wird.

Farbe	Hexwert	C	M	Y	K-Wert
Grau	#9D9E9E	0	0	0	50
Hellorange	#FFB060	0	39	70	0
Weinrot	#902020	9	100	95	42



Abbildung 12: Webseite | Farbpalette

Während die Schlichtheit als unterstützende Funktion zur Eindeutigkeit und Verständlichkeit des Logos beiträgt und auf die Zweckmäßigkeit des Systems hinweist, erfolgt außerdem durch die Einbettung der Schrift des UR LOGOS (Idee, Konzept und Umsetzung von AKAD. OBERRAT JOSEF MITTLMEIER) ein klarer Kontextbezug des Chatbots zur universitären Umgebung.

Analog zum runden UNI-LOGO trägt die Sprechblase das UR in einer ausgeschnittenen Form, welches durch die Abwesenheit der Buchstaben die Wahrnehmung des menschlichen Auges unterstützt.

Die Sprechblasenspitze wurde so konzipiert, dass es scheint, als würde sie das Wort TALKING tragen, wobei auf eine Berührung der Formen verzichtet wurde. Dennoch wirkt die Sprechblase als tragende Form, auf welchem das TALKING sowohl im visuellen, als auch im metaphorischen Sinn schwebt und somit existiert. Der Abstand zwischen der Sprechblase und dem TALKING wurde hierbei so gewählt, dass das TALKING für sich steht, dennoch im Kontext nicht verloren geht. Aufgrund dessen wurde der Abstand der beiden Formen gering genug gewählt, um das Logo in seiner Gesamtheit zu erfassen. Außerdem schließt die Sprechblasenspitze bündig zur unteren Form des Buchstaben „g“ im Wort TALKING ab, welche die Zusammengehörigkeit des Logos und deren unterschiedlichen Formen in deren Kollektiv bestätigt.

Das Wortspiel des UR TALKING-Logos spiegelt sowohl den Gesprächsbedarf rund um die Thematik der Universität Regensburg wieder, als auch den Selbstbezug zum Nutzer und den Zweck des Chatbots durch die Abkürzung UR TALKING = You are talking. Dabei symbolisiert die Sprechblase – in der runden Form – das Wissen rund um die Universität Regensburg, und die Sprechblasenspitze wird als richtungsweisender, unterstützender Pfad der Dialogführung durch die unendliche Datenbasis wahrgenommen. Das Logo – Idee und Konzeption von HERRN PD DR. JÜRGEN REISCHER und Umsetzung von BAZO ALEXANDER, RAK JESSICA und SCHLESINGER MICHAELA – wurde im Verlauf der Erstellung eines ersten Prototypen für einen Studienberater der Informationswissenschaft und Medieninformatik in Form eines Chatbots in der Webpräsenz eingebettet.

5 Prolog-Schnittstelle

Als zentraler Punkt fungiert wie bereits angesprochen die wissensbasierte Programmiersprache *Prolog*, welche sich als Schnittstelle für den *AIML*-gestützten Prototyp hervorragend eignet. Zurückgegriffen wird hierbei auf die beiden genannten Seminararbeiten des *WBS-Seminars* im Sommersemester 2013.

5.1 Datenbank

Dem *Prolog*-Code zugrunde liegt eine erweiterbare Datenbank – siehe Abbildung 14: ‘Datenbankschema | Methoden der maschinellen Sprachverarbeitung’ –, die aktuell aus sechs Entitäten und acht Beziehungstabellen besteht. Im Rahmen des *Seminars Wissensbasierte Systeme* im SS 2013, geleitet von PROF. DR. RAINER HAMMWÖHNER, entstand ein erster Entwurf einer Datenbank (siehe Abbildung 14: ‘Datenbankschema | Methoden der maschinellen Sprachverarbeitung’), welche als Grundlage eines prologbasierten Recommender-Systems für Studenten dient.

Das konzipierte Datenbankschema spaltet sich im Wesentlichen in zwei Hauptstrukturen – Student und Studium – auf, welche logisch ineinandergreifen. Dabei handelt es sich einerseits um die Daten des Studenten und andererseits um Thematiken rund ums Studium, wie zum Beispiel der Fachkombination, die ein Student studiert oder dem Vorlesungsverzeichnis in Form der Vorlesungen, die aktuell angeboten werden.

Die studentische Seite beschränkt sich auf die Entität *Student*, sowie deren zugehörige Beziehungstabellen *Student_Fachkombi* und *Student_Kurse*. Weiterhin gliedert sich das Datenbankschema in die Entitäten *Fachkombination*, *Module*, *Kurse* und das *Vorlesungsverzeichnis*, welche Informationen rund ums Studium beschreiben. Verbunden sind diese vier Tabellen durch die anknüpfenden Beziehungstabellen *Fachkombi_Modul*, *Kurse_Modul*, *Kurse_Vorlesungsverzeichnis* und *AbhängigkeitKurse*. Jede der genannten Tabellen wird eindeutig durch einen Primärschlüssel – in Form einer ID – identifiziert, wobei Beziehungstabellen durch einen doppelten Primärschlüssel zweier Entitäten gekenn-

zeichnet sind. Im Folgenden wird das Datenbankschema der vorhandenen Tabellen anhand eines Beispiels der Datenbankbefüllung näher erläutert.

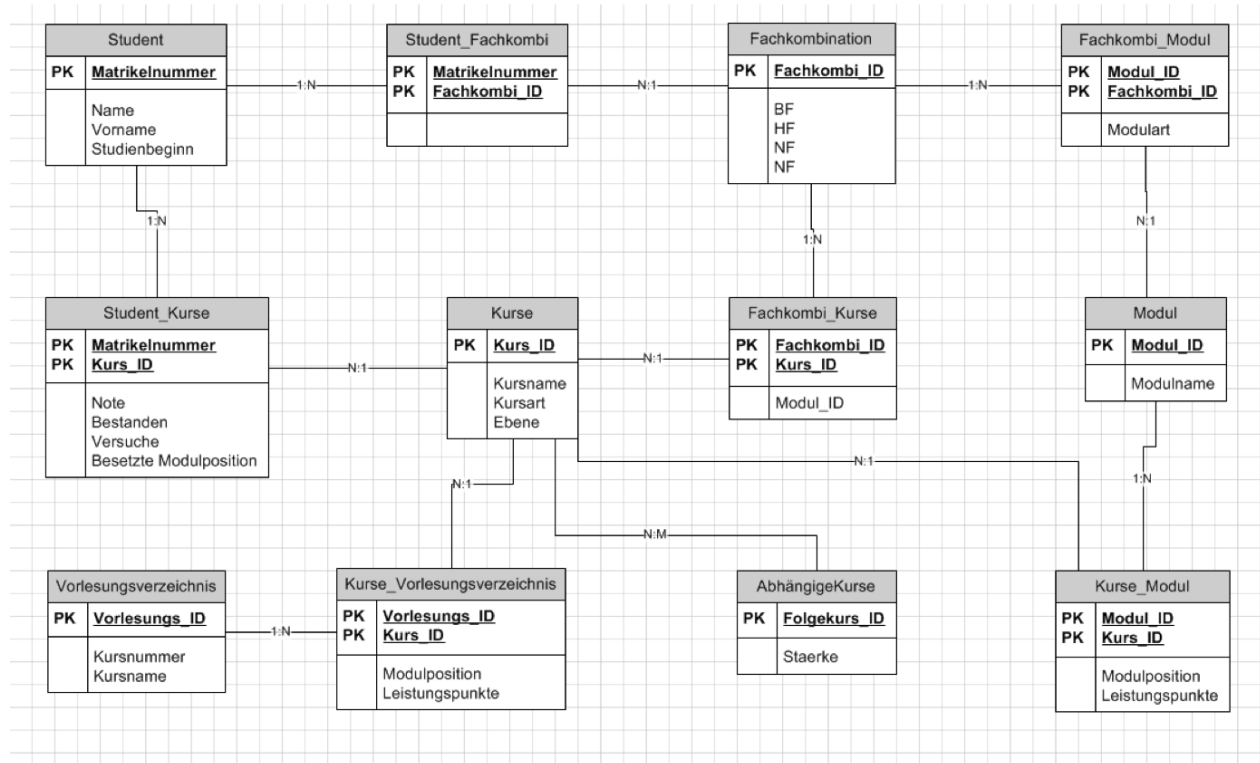


Abbildung 13: Datenbankschema | Wissensbasierte Systeme

Die Entität *Student* enthält den eindeutigen Primärschlüssel der *Matrikelnummer* eines Studenten mit den dazugehörigen Attributen *Name*, *Vorname* und dem jeweiligen *Studienbeginn* eines Studenten.

Beispiel:

```
%student(Matrikelnummer,Name,Vorname,Studienbeginn).
```

```
student (1234321, 'Skywalker', ' Luke', 'WS13').
```

Die *Fachkombination* wird durch die jeweilige *Fachkombi_ID* eindeutig identifiziert und weiterhin durch die Attribute *BF*, *HF* und *NF* spezifiziert. Dadurch, dass sich das Studienberatungssystem auf die drei Studiengänge Informationswissenschaft, Medieninformatik und Medienwissenschaft beschränkt, gibt es neun mögliche Kombinationen. Die Beziehungstabelle *Student_Fachkombi*, welche nur aus dem zusammengesetzten Primärschlüssel beider Tabellen besteht, weist jedem Studenten eine Fachkombination zu.

Beispiel:

```
%fachkombination (Fachkombi_ID, BF, HF, NF, NF).
```

```
fachkombination(001, 'inf', 'mei', 'Null', 'Null').
```

```
% student_fachkombi(Matrikelnummer, Fachkombi_ID).
```

```
student_fachkombi(1234321,001).
```

Die Tabelle *Module* enthält den Primärschlüssel *Modul_ID*, sowie einen dazugehörigen *Modulnamen*. Analog zu der soeben genannten Tabelle, wird die Beziehung zwischen den zu leistenden Modulen einer Fachkombination durch den doppelten Primärschlüssel einer *Modul_ID*, sowie einer *Fachkombi_ID* in der Beziehung *Fachkombi_Modul* abgebildet. Zusätzlich dazu erhält die Beziehungstabelle *Fachkombi_Modul* ein weiteres Attribut, die sogenannte *Modulart*, mit der es möglich ist, zwischen Pflicht- und Wahlmodulen zu unterscheiden.

Beispiel:

```
% module (Modul_ID, Modulname).
```

```
Modul('infm01','Grundlagen der Informationswissenschaft').
```

```
% fachkombi_module(Fachkombi_ID, Modul_ID, Pflichtmodul oder Wahlmodul).
```

```
fachkombi_module(001,'infm01','pflicht').
```


Die vier Attribute – *Kurs_ID*, *Kursname*, *Kursart* und *Ebene* – generieren die Tabelle *Kurse*, die durch das Attribut *Kurs_ID* eindeutig identifizierbar ist. Hierbei entsteht aus den Entitäten *Module* und *Kurse* die Beziehungstabelle *Kurse_Modul*, welche den zusammengesetzten Primärschlüssel *Modul_ID* und *Kurs_ID* zu Identifikation aufweist. Des Weiteren sind in der Relationsentität *Kurse_Modul* die Attribute der *Modulposition* und die *Leistungspunkte* enthalten.

Beispiel:

```
% kurse (Kurs_ID, Kursname, Kursart, Ebene).
```

```
kurse('inf001', 'Einführung in die Informationswissenschaft', 'Einfuehrung', 'E1').
```

```
% modul_kurse(Modul_ID, Kurs_ID, Modulposition, Leistungspunkte).
```

```
modul_kurse('infm01', 'inf001', 'INF - M 01.1', 6).
```

Abhängige Kurse werden durch eine *Kurs_ID* und eine *Folgekurs_ID* eindeutig bestimmt. Somit beschreibt und bestimmt die Beziehungstabelle *AbhängigeKurse*, welche Kurse von anderen Kursen – nach dem Motto Grundkurs und Vertiefungskurs – abhängig voneinander sind. Das Attribut *Staerke* beschreibt aufsteigend den Grad der Abhängigkeit.

Beispiel:

```
% abhangigkecourse(Kurs_ID, Folgekurs_ID, Staerke).
```

```
abhangigkecourse('inf001', 'inf001', 0).
```

Ebenso sind in der Entität *Vorlesungsverzeichnis* die aktuellen Vorlesungen abgespeichert, welche sich jedoch momentan auf das WS 2013/14 beschränken. Vorlesungen enthalten eine eindeutige *Vorlesungs_ID*, sowie die Attribute *Kursnummer* und *Kursname*. Die Beziehung *Kurse_Vorlesungsverzeichnis* beschreibt, welche Pflicht- bzw. Wahlkurse im aktuellen Vorlesungsverzeichnis zu belegen sind und setzt sich aus den beiden Pri-

märschlüsseln *Kurs_ID* und *Vorlesungs_ID* zusammen. Ergänzt wird die Beziehung durch die Attribute *Leistungspunkte* und *Modulposition*.

Beispiel:

```
%vorlesungaktuell(Vorlesungs_ID,Kursnummer, Kursname).  
vorlesungaktuell('infv001', 36650, 'Einführung in die Informationswissenschaft mit  
Tutorium').  
  
%kurse_vorlesungsverzeichnis(Vorlesungs_ID, Kurs_ID, Modulposition, Leis-  
tungspunkte).  
kurs_vorlesungsverzeichnis('infv001', 'inf001', 'INF - M 01.1', 6).
```

Die Beziehungstabelle *Fachkombi_Kurse* setzt sich aus den beiden Schlüsseln *Fachkombi_ID* und *Kurs_ID* zusammen und wird durch das Attribut *Modul_ID* ergänzt. Diese Relation beschreibt, welche *Kurse* in einer bestimmten *Fachkombination* zu belegen sind, die wiederum in *Module* mit einer bestimmten *Modulposition* zusammengefasst sind.

Beispiel:

```
%fachkombi_kurse (Fachkombi_ID, Modul_ID, Kurs_ID).  
fachkombi_kurse (001, 'infm01','inf001').
```

Anschließend dazu bildet die Beziehung *Student_Kurse* die bereits belegten Lehrveranstaltungen ab, die ein bestimmter Student besucht hat. Als Primärschlüssel hierfür sind die Attribute *Matrikelnummer* und *Kurs_ID* tätig, wobei die vier Attribute *Bestanden*, *Note*, *Versuche* und *BesetzteModulposition* die Relationstabelle erweitern.

Beispiel:

```
% student_kurse(Matrikelnummer, Kurs_ID, Bestanden, Note, Versuche, Besetzte  
Modulposition).  
student_kurs (1234567, 'inf001', 'true', 1,3, 1, 'INF - 01.1').
```

Dieses soeben erläuterte Datenbankschema wurde im Rahmen des Seminars „*Methoden der maschinellen Sprachverarbeitung*“, geleitet von PD DR. JÜRGEN REISCHER, um die Entität des Mitarbeiters und um die Beziehungstabelle *Mitarbeiter_Vorlesungsverzeichnis* erweitert (siehe Abbildung 14: ‘Datenbankschema | Methoden der maschinellen Sprachverarbeitung’).

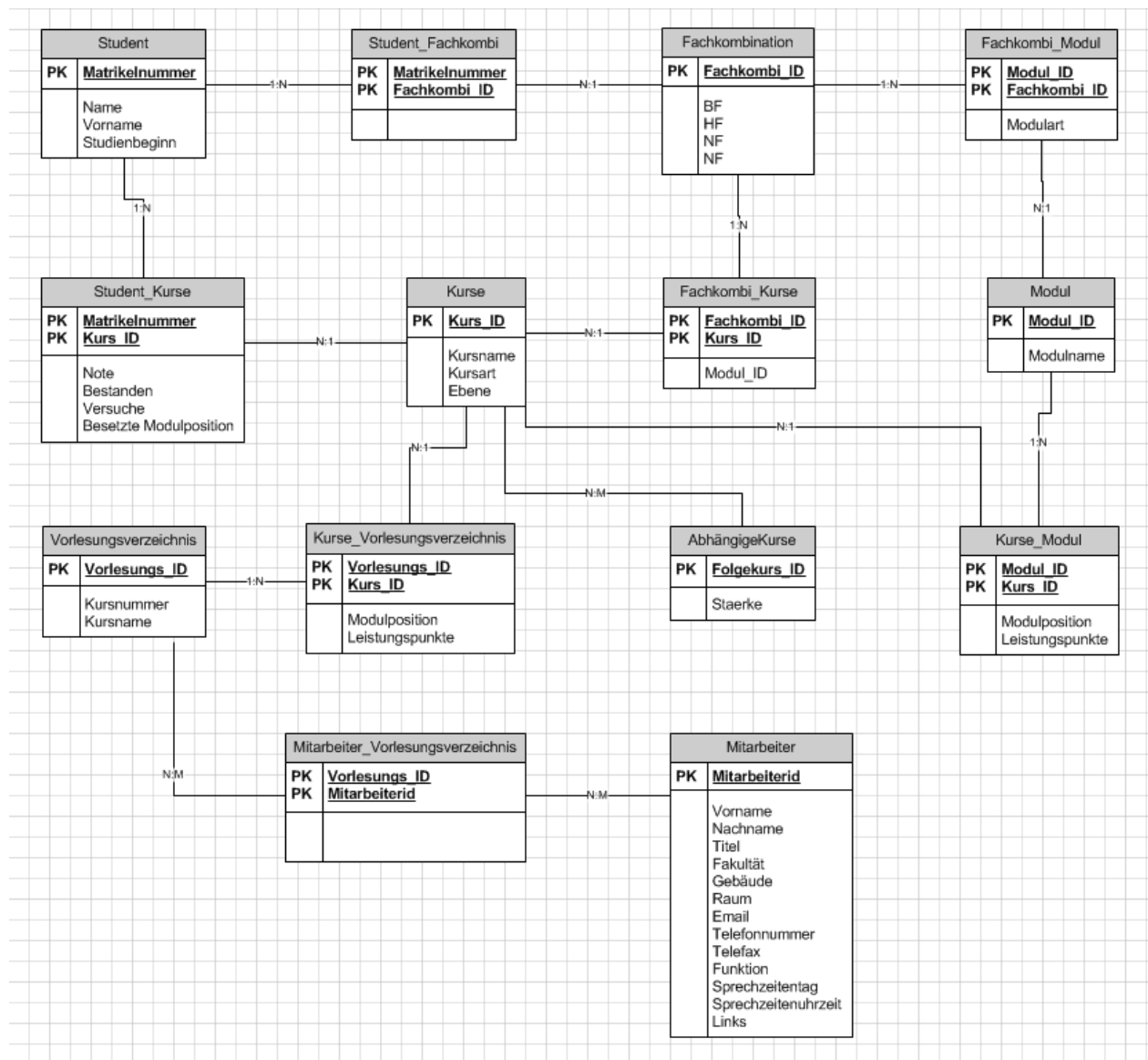


Abbildung 14: Datenbankschema | Methoden der maschinellen Sprachverarbeitung

Zusätzlich wurde die bereits bestehende Datenbank um die Entität des Mitarbeiters erweitert. Dies erwies sich insofern als sinnvoll, da somit Lehrveranstaltungen mit wertvollen Metadaten angereichert werden können.

Die Entität des *Mitarbeiters* umfasst folglich sowohl Professoren, Doktoranden, Dozenten, Studentische Hilfskräfte und das Sekretariat. Eindeutig identifiziert wird diese Tabelle durch eine *Mitarbeiter_ID*, die durch die Attribute *Vorname*, *Nachname*, *Titel*, *Fakultät*, *Gebäude*, *Raum*, *Email*, *Telefonnummer*, *Telefax*, *Funktion*, *Sprechzeitentag*, *Sprechzeiten-
uhrzeit* und den *Link* zu der jeweiligen Person auf der Universitätshomepage ergänzt wird. Durch die Beziehung *Mitarbeiter_Vorlesungsverzeichnis* ist es möglich, die zugeordneten Lehrpersonen einer Lehrveranstaltung auszugeben. Die Relationstabelle wird hierbei eindeutig durch eine *Mitarbeiter_ID* und die jeweilige *Vorlesungs_ID* identifiziert.

Beispiel:

```
% mitarbeiter (Mitarbeiter_ID, Fakultät, Titel, Vorname, Nachname, Telefonnummer, Telefax, Email, Gebäude, Raum, Sprechzeitentag, Sprechzeiten-  
uhrzeit, Funktion, Link)
```

```
mitarbeiter ('ma001', 'Informationswissenschaft', 'Null', 'Renate', 'Vogl', 09419433585,  
09419431954, 'Renate.Vogl@ur.de', 'PT', 'Zimmer 3.0.84b', 'Montag bis Donnerstag',  
'09:00-11:30 Uhr', 'Sekretariat', 'Null').
```

```
% vorlesungsverzeichnis_mitarbeiter (Vorlesungsid, Mitarbeiterid)
```

```
mitarbeiter_vorlesungsverzeichnis ('infv001', 'ma004').
```

Nach der Konzeption des Datenbankschemas erfolgte die Befüllung anhand gegebener Daten des Modulkataloges, der Prüfungsordnung und der Universitäts-Homepage der jeweiligen Studiengänge (Informationswissenschaft, Medieninformatik und Medienwissenschaft). Des Weiteren wurde das aktuelle Vorlesungsverzeichnis des Wintersemes-

ters 2013/14 eingepflegt, wobei es sich bei den Studenten um drei fiktive Musterstudenten handelt.

Die Tabellen stehen in folgenden Beziehungen zueinander:

Student_Fachkombi

Ein Student kann mehrere Fachkombinationen studieren und eine Fachkombination kann von mehreren Studenten studiert werden.

Fachkombi_Modul

Eine Fachkombination enthält mehrere Module und mehrere Module sind in einer Fachkombination.

Fachkombi_Kurse

Eine Fachkombination besteht aus mehreren Kursen und mehrere Kurse bilden eine Fachkombination.

Kurse_Modul

Mehrere Kurse bilden ein Modul und ein Modul besteht aus mehreren Kursen.

Kurse_Vorlesungsverzeichnis

Mehrere Kurse sind in einem Vorlesungsverzeichnis enthalten und ein Vorlesungsverzeichnis besteht aus mehreren Veranstaltungen.

Student_Kurse

Ein Student besucht mehrere Kurse und ein Kurs wird von mehreren Studenten besucht.

Vorlesungsverzeichnis_Mitarbeiter

Mehrere Vorlesungen können von einem Mitarbeiter gehalten werden und mehrere Mitarbeiter können eine Vorlesung halten.

Einen Sonderfall bildet die Tabelle *AbhängigeKurse*, die mit der Tabelle *Kurse* eine selbst-bezogene Beziehung eingeht und einer n:m-Beziehung folgt.

5.2 Musterstudenten

Wie schon erwähnt, wurden zu Testzwecken des Recommender-Systems drei unterschiedliche Musterstudenten angelegt. Sinn und Zweck dieser Musterstudenten ist es, aufgrund der bereits gesammelten Daten zum Beispiel in Form von bestandenen Kursen, prologbasierte Empfehlungen zu generieren, die den weiteren Studienverlauf abbilden.

Hierbei gab es je einen Studienanfänger, der Informationswissenschaft als Bachelorfach und Medieninformatik als zweites Hauptfach studiert, und dabei keinerlei universitäre Vergangenheit bzw. Wissen hat, somit als völlig neuer Nutzer in das System mit aufgenommen wird.

Stellvertretend für einen fortgeschrittenen Studenten steht der zweite Musterstudent, welcher Informationswissenschaft als Bachelorfach und Medienwissenschaft als zweites Hauptfach im dritten Semester belegt. Als Ausgangsbasis hierbei steht die Absolvierung von ca. 30 Leistungspunkten pro Semester und somit die implizierte Einhaltung der Regelstudienzeit von sechs Fachsemestern. Anhand dieser Daten ist es dem System möglich, priorisierte Empfehlungen für den weiteren Studienverlauf im nächsten Semester zu generieren.

Der letzte und dritte Musterstudent studiert im fünften Semester Informationswissenschaft als Bachelorfach und sowohl Medieninformatik als auch Medienwissenschaft als Nebenfach. Das Profil dieses Studenten steht für einen chaotischen Studienverlauf, bei dem zum Beispiel unter anderem grundlegende Abhängigkeiten von Grund- und Vertiefungskursen außer Acht gelassen werden. Die unstrukturierte Belegung und Absolvierung der Lehrveranstaltungen spiegeln sich bei dem Studenten in Drittversuchen

und zurückstehenden Kursen wider. Des Weiteren befindet er sich nicht in der Regelstudienzeit.

Die drei Musterstudenten des WBS-Seminars im SS 2013 zeigten auf, dass es individuelle Studienverläufe gibt, welche nach einer individuellen Beratung und Empfehlung von Lehrveranstaltungen verlangen. Um diese Daten für den Chatbot zu nutzen, wäre eine Verbindung zwischen der *Prolog*-Datenbank und FlexNow unabdingbar. Die personalisierten Daten könnten durch eine Hinterlegung außerdem ein individualisiertes Gespräch mit dem Chatbot ermöglichen, da dem Chatbot direktes Wissen über den menschlichen Kommunikationspartner zur Verfügung stehen würde.

Da es sich hierbei jedoch um den ersten Prototyp des Chatbots handelt, wird im Rahmen der vorliegenden Arbeit nicht weiter auf diese Thematik eingegangen. Bei einer Weiterführung des Projektes wäre ein Aufleben dieses Gedankens wertvoll.

5.3 Mächtigkeit des Prolog-Codes

Auf Grundlage der hinterlegten Datenbasis ist es möglich, den Nutzern das aktuelle Vorlesungsverzeichnis auszugeben, welches mit Metadaten – ebenfalls in der Datenbank hinterlegt – angereichert ist.

Wie schon im vorherigen Kapitel beschrieben, wäre es möglich, das System für jeden Studenten anhand seiner Matrikelnummer – siehe Abbildung 15 ‘Prolog|Student’ – zu personalisieren. Dies erfolgt durch die eindeutige Identifikation des jeweiligen Studenten durch die zugewiesene Identifikationsnummer. Voraussetzungen für die personalisierte Nutzung des Chatbots wäre der Zugriff auf studentische Daten durch das System FlexNow, welches alle Daten des Studenten, wie zum Beispiel bestandene und nicht bestandene Kurse, in sich vereint.

```
?- start.  
Bitte gib deine Matrikelnummer ein.: 1234321  
  
Hallo Luke Skywalker,  
  
Achtung: In der Kombination mit Medieninformatik als zweites Hauptfach  
muss INF-M 03 durch einen zusätzlichen Anwendungsschwerpunkt aus MEI-M  
10 mit Seminararbeit oder eine analoge fachliche Vertiefung aus dem  
Bereich Informationswissenschaft ersetzt werden.  
  
Bitte beachte, dass für die Grundlagen- und Orientierungsprüfung bis  
zum Ende des 2.ten Semesters jeweils ein Kurs aus den Module INF-  
M01, INF-M02 und INF-M04 und die Module MEI-M01 und MEI-M03 erfolgreich  
belegt sein müssen.
```

Abbildung 15 Prolog|Student

Anhand der hinterlegten Daten wäre es somit möglich, dem jeweiligen Studenten eine priorisierte und personalisierte Empfehlung von Lehrveranstaltungen auszugeben, welche nach bestimmten Kriterien sortiert worden ist. Unter diesen Kriterien sind zum Beispiel Drittversuche, die Abhängigkeit von Grund- und Vertiefungskursen und bereits belegte bzw. freie Modulpositionen aufzuführen. Dabei gilt die Regel, je höher der Faktor (links aufgeführt z.B. 3.241), desto höher ist auch die Priorität des Kurses im aktuellen Studienverlauf.

```
? - abschluss.  
  
3.241- - Einführung in die Programmierung und Programmiersprachen  
6-LP MEI - M 03.1  
3.241- - Einführung in die Informationswissenschaft  
6-LP INF - M 01.1  
3.034- - Einführung in die Informatik und Medieninformatik  
5-LP MEI - M 01.1  
2.241- - Algorithmen und Datenstrukturen  
6-LP MEI - M 03.2  
1.828- - Mathematische Grundlagen  
4-LP INF - M 02.1  
1.828- - Einführung in die Informationslinguistik  
4-LP INF - M 02.3  
1.241- - Übung zu Mathematik für Medieninformatik I  
6-LP MEI - M 02.1  
1.241- - Mathematik für Medieninformatiker I  
6-LP MEI - M 02.1  
  
[...]
```

Abbildung 16 Prolog|Kursempfehlung

5.4 Ergänzungen zur formalen Gestaltung des Studiums

Durch die Ergänzungen der *Prolog*-Implementierung von Bazo und Spröd sind einige zusätzliche Funktionen möglich, die für das konzipierte Chatbot-System relevant sein können. Abgebildet sind unter anderem die möglichen Studiengänge der Fakultät sowie die validen Kombinationsmöglichkeiten dieser Fächer. Einzelne Fachkombinationen – sowohl für Hauptfachkombinationen sowie Haupt- und Nebenfachkombinationen – können auf ihre Studierbarkeit hinsichtlich der Prüfungsordnung überprüft werden. Ebenfalls können für einzelne Fächer mögliche Kombinationspartner generiert werden. Für die Fächer Informationswissenschaft, Medieninformatik und Medienwissenschaft können, bei entsprechender Eingabe der absolvierten Module, vorläufige Abschlussnoten berechnet werden. Dabei werden die Berechnungsvorschriften der Prüfungsordnung berücksichtigt. Möglich sind dabei – unter Einbezug unterschiedlicher Interpretationen noch nicht abgeschlossener Modulpositionen – sowohl die Rückgabe einer optimistischen als auch einer pessimistischen Schätzung.

Die konkreten Implementierungen dieser Ergänzungen sowie weiterer, für das Botsystem weniger relevanter Möglichkeiten, können in der entsprechenden Seminararbeit eingesehen werden.

Um den Chatbot ELISE ein Innenleben zu verschaffen, wurden in einem weiteren Schritt – welcher in den nächsten Kapiteln näher erläutert wird – mögliche Fragestellungen an das System gesammelt.

6 Datenerhebung

Obwohl die aufbereiteten Daten und Regeln des *WBS-Seminars*, als auch deren sinnvolle Erweiterung im universitären Kontext des Chatbots UR TALKING bereits eine mächtige Datenbasis darstellen, ist eine separate Datenerhebung unabdingbar.

Im Folgenden werden die drei angewandten Methoden zur Datensammlung – Befragung von Schülern am Regensburger Hochschultag, studentische Umfrage im Kursbetrieb und Expertenbefragung im Sprachverarbeitungskurs – vorgestellt, wobei anzumerken ist, dass nur zwei dieser Erhebungen zu sinnvollen bzw. brauchbaren Daten geführt haben.

DATENERHEBUNG			
	Zielgruppe	Anzahl der Datensätze	Brauchbare Datensätze
Kursbetrieb	Studenten	515	515
Hochschultag	Schüler	74	71
Sprachverarbeitungskurs	Experten	unbrauchbar	0

Tabelle 1: Datenerhebung | Allgemein

6.1 Studentische Umfrage im Kursbetrieb

Wie bereits angesprochen, zeichnet sich das Nutzerprofil eines Beratungschatbots im universitären Kontext durch die beiden Hauptgruppen Studenten und Schüler aus. Um die Zielgruppe der Studenten zu erreichen, fiel die Entscheidung, eine Umfrage im laufenden Kursbetrieb der Informationswissenschaft und der Medieninformatik durchzuführen. Hierbei wurden Kursteilnehmer der Einführungsveranstaltungen priorisiert, da Studenten des Grundstudiums bereits einen gewissen Wissensstand(ard) zum Studentischen Leben haben, dennoch geeignet sind, um die Interessensgebiete und aufkommende Fragen rund um das Studium zu formulieren. Folglich beschränkte sich die Befra-

gung auf die Kurse *Einführung in die Informationswissenschaft* von PROF. DR. LUDWIG HITZENBERGER; *Mathematische Grundlagen*, geleitet von PROF. DR. BERND LUDWIG; *Einführung in die Informatik und Medieninformatik*, geleitet von RAPHAEL WIMMER, VICTORIA BÖHM und TIM SCHNEIDERMEIER und *Einführung in die Informationslinguistik* von PD DR. JÜRGEN REISCHER.

Insgesamt haben 101 Personen an der studentischen Umfrage teilgenommen und es wurden 515 Fragen an den Chatbot formuliert.

Kursname	Kursleiter	Teilnehmer
36650 Einführung in die Informationswissenschaft	HITZENBERGER	18
36600 Einführung in die Informatik und Medieninformatik	WIMMER, SCHNEIDER, BÖHM	59
36653 Mathematische Grundlagen	LUDWIG	6
35018 Einführung in die Informationslinguistik	REISCHER	18
Gesamt		101

Tabelle 2: Datenerhebung | studentische Umfrage im Kursbetrieb

Die unausgewogene Verteilung – siehe Abbildung 17: ‘Fragebogen | Personenverteilung’ – ergab sich aus dem Faktum, dass es sich bei den vier Lehrveranstaltungen um Grundkurse handelt und diese von den Studenten auch parallel besucht werden.

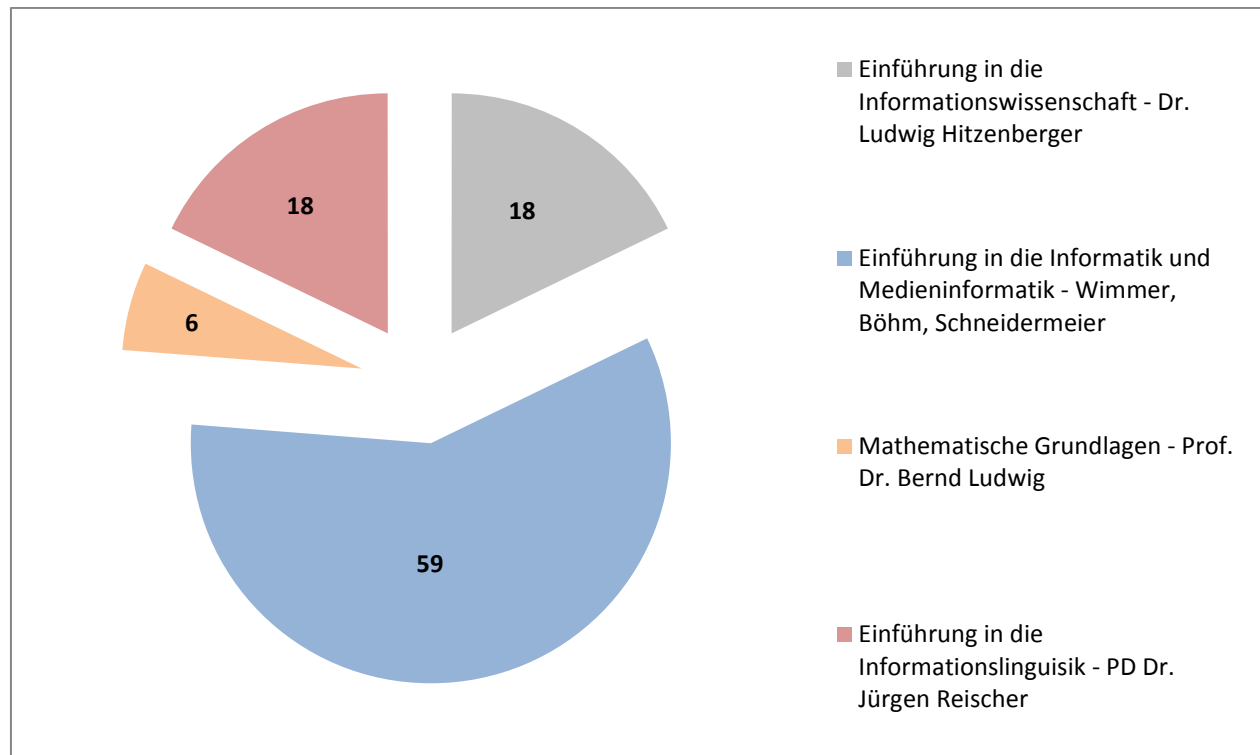


Abbildung 17: Fragebogen | Personenverteilung

Die Umfrage erfolgte durch Verteilung eines Evaluationsbogens – siehe Abbildung 18: ‘Fragebogen | Grundkurse’ – während der genannten Kurse. Aufgabenstellung hierbei war es, mindestens fünf Fragen zu notieren, welche die Studenten an die Studiengänge Informationswissenschaft, Medienwissenschaft und Medieninformatik haben. Die Zielgruppe der Studierenden erwies sich als teilnahmefreudig, wodurch weitgefächert Fragen und Auskunftsbedarf der Studenten gesammelt werden konnten. Neben der breit abgedeckten Generierung von Thematiken und Formulierungsvarianten konnten bei der Sichtung der Datensätze bereits Themengebiete entdeckt werden. Generell teilten sich die Daten in die drei Kategorien *vor dem Studium*, *während des Studiums* und *nach dem Studium*. Ein klarer Trend zeigte auf, dass die Kategorie *während des Studiums* die umfangreichste Klasse sei, welche sowohl in den Fragen als auch in den Antworten den größten Informationsbedarf aufweist.

Im Anschluss zur Befragung der Studenten erfolgte – zur Abdeckung der zweiten Zielgruppe – eine Befragung von Schülern am Regensburger Hochschultag.



Philosophische Fakultät III

Sprach-, Literatur- und Kulturwissenschaften

Institut für Information und Medien, Sprache und Kultur (I:IMSK)

Lehrstuhl für Informationswissenschaft

36679 Methoden der maschinellen Sprachverarbeitung

Modul: INF-M 31.1: VS/VP Methoden der maschinellen Sprachverarbeitung

WS 2013/14

Leitung: PD Dr. Jürgen Reischer

Gruppe: Bazo, Rak, Schlesinger

Im Rahmen des Masterseminars 36679 Methoden der maschinellen Sprachverarbeitung wird ein Chatbot konzipiert, welcher als Studienberatungssystem fungieren soll.

Im Fokus stehen hierbei vor allem Fragen, welche Studienanfänger an den Studiengang Informationswissenschaft/Medieninformatik/Medienwissenschaft und dessen Kombinationsmöglichkeiten beschäftigen.

Bitte formulieren Sie mindestens **fünf relevante Fragen**, welche sich an den Studiengang richten bzw. Fragen, welche sich rund um das Ihnen bevorstehende Studium drehen.

Abbildung 18: Fragebogen | Grundkurse

6.2 Befragung von Schülern am Regensburger Hochschultag

Die Befragung von Schülern und Studieninteressierten hat am 02. Februar 2014 während des Regensburger Hochschultages stattgefunden. Hierbei erfolgte die Sammlung von Fragen, welche sich für Studienanfänger, Schüler, als auch potentielle Studenten an das Studienfach richten. Hierbei wurden insgesamt 71 Fragen – von insgesamt 74 Fragen, wobei drei Fragen lediglich dem Demonstrationszweck dienten und deshalb für die Gesamtwertung entfernt wurden – geloggt, welche teilweise von den Schülern selbst in das System – siehe Abbildung 19: 'Elise | User-Interface' – eingetippt wurden und/oder während der Beratungsgespräche von den Fachkräften notiert wurden.

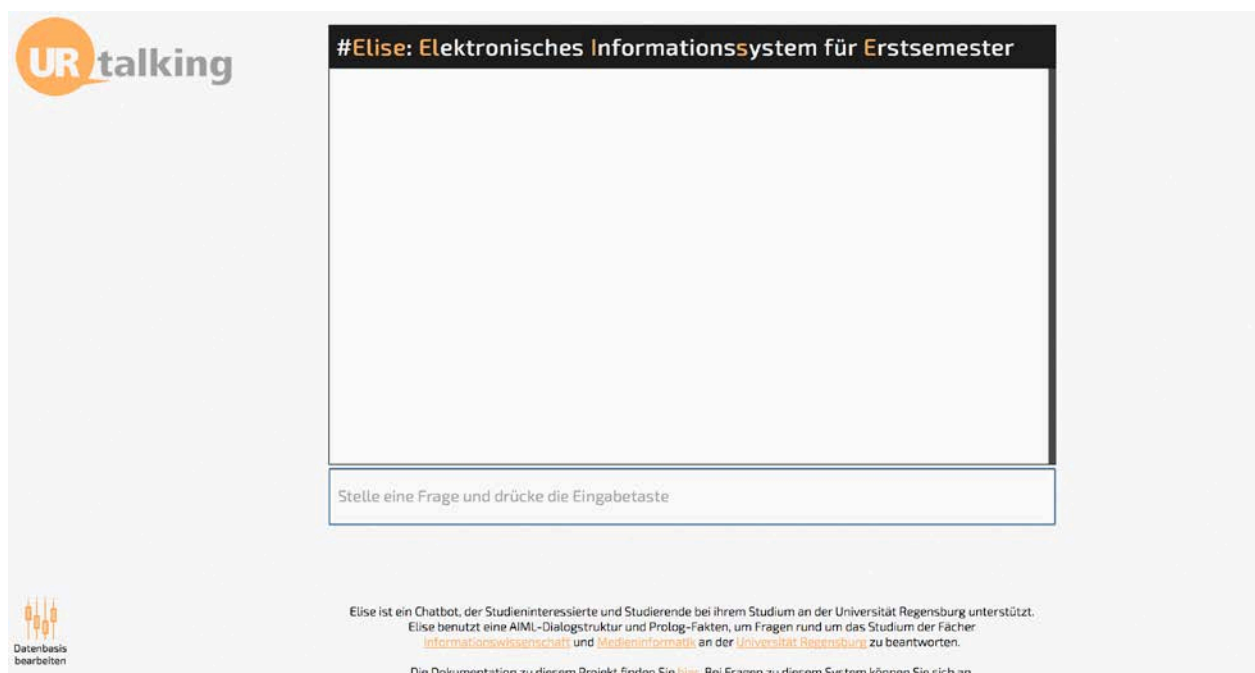


Abbildung 19: Elise | User-Interface

Ein klarer Trend zeigte sich in der Frage nach dem Inhalt des Studienganges, was durch die Veranstaltung des Informationstages bedingt ist. Außerdem stellte die Zielgruppe am Hochschultag Fragen nach den Kombinationsmöglichkeiten, den Lerninhalten bzw. Überschneidungen der Studiengänge, den Voraussetzungen/Vorkenntnissen und den beruflichen Möglichkeiten. Diese Art von Fragestellungen und Themenschwerpunkten

liegen dem abweichenden Kenntnisstand von Schülern und Studenten zu Grunde. Die Zielgruppe des Hochschultages repräsentiert hierbei das Knowhow von studienfremden Personen, welche in das universitäre Leben noch keinen oder nur geringen Einblick genommen haben. Thematiken wie zum Beispiel Praktika und Ausland, Allgemeines Studentenleben, Klausuren und Note, Bachelor und Master als auch Kontextfrei und Scherzfragen wurden gar nicht oder nur in Einzelfällen angesprochen. Die Zielgruppe beschäftigt sich somit wie erwartet mit Fragen vor dem Studium und nach dem Studium, sowie der Einzelthematiken des Inhaltes und Kombinationsmöglichkeiten während des Studiums. Vertiefende Auskünfte bzw. Themencluster in der Rubrik *während des Studiums* sind (im Gegensatz zur studentischen Zielgruppe) kaum angesprochen worden.

6.3 Expertenbefragung im Sprachverarbeitungskurs

In der letzten Sitzung des Sprachverarbeitungskurses von HERRN PD DR. REISCHER wurde die Thematik *Wissensressourcen für Chatbots II – Informationswissenschaft und Medieninformatik* behandelt. Hierbei bat sich die Gelegenheit, die Kursteilnehmer – welche sich als Experten für studienkundige bzw. studienaffine Personen qualifizieren – als Fachkräfte eine ausgewählte Anzahl der gesammelten Fragen aus den Datensätzen der studentischen Umfrage im Kursbetrieb beantworten zu lassen.

Hierzu wurde das *Etherpad* – ein *collaborative real-time editor* – verwendet, aufrufbar durch die URL <http://goo.gl/qtT2w0>.

Interaktiv sollten hierbei Frage-Antwort-Paare generiert werden, wobei folgende Themen bzw. Fragen vorgegeben wurden:

Beruf und Karriere

Welche Berufsmöglichkeiten gibt es mit Infowissenschaft?

Welche Studienfachkombination ist sinnvoll für welchen Beruf?

Welche Beschäftigungsmöglichkeiten habe ich mit abgeschlossenem Bachelor-Studium?

In welcher Branche werde ich nach dem Studium sehr gute Jobchancen haben?

Unterschiede der Studiengänge

Wie überschneiden sich Medieninformatik und Informationswissenschaft in der Praxis?

Behandeln Informationswissenschaft und Medieninformatik fast nur verwandte Themen?

Wie praxisorientiert ist der Studiengang?

Was sind die Hauptthemen meines Studiengangs?

Bauen spätere Veranstaltungen auf die anfangs erworbenen Lerninhalte auf?

Stundenplangestaltung

Wie kann man seinen Stundenplan sinnvoll erstellen?

Worauf sollte ich bei meinem Stundenplan achten?

Gibt es einen Beispiel-Stundenplan für mein erstes Semester?

Wer kann mir helfen, den Stundenplan zusammen zustellen?

Studienverlauf

Wie ist das Studium aufgebaut?

Welche Kurse sollte ich zu Beginn wählen?

Was wäre die optimale Einteilung (Kurse pro Semester)?

Gibt es Kurse, die ich bis zu einem bestimmten Semester bestanden haben muss?

Gibt es sich überschneidende Veranstaltungen?

Prüfungen und Klausuren

Wo finde ich alte Klausuren?

Kann ich Noten auch noch im Nachhinein verbessern, wenn ich die Veranstaltung wiederhole?

Wie oft kann man eine Prüfung wiederholen (2x/3x)?

Wie melde ich mich für die Prüfungen an?

Wo kann ich Prüfungstermine nachschauen?

Leistungspunkte

Welche Möglichkeiten habe ich, um die Leistungspunkte im Wahlbereich zu bekommen?

Wie viele ECTS-Punkte bekommt man im Schnitt pro Kurs?

Wie funktioniert das Punktesystem des Bachelors?

Wie viele Punkte sind pro Semester optimal?

Wie viele LPs bekomme ich für welche Veranstaltung?

Brauche ich bis zu einem bestimmten Semester eine Mindestanzahl von ECTS?

Orientierung auf dem Campus

Wo ist die Mensa?

Wie finde ich meine Hörsäle?

Wo finde ich einen Campus-Plan?

Wo gibt es ruhige Plätze zum Lernen an der Uni?

Praktika und Auslandssemester

Gibt es während des Studiums die Möglichkeit, Praxiserfahrung zu sammeln – wenn ja, wie?

Was gilt als Praktikum für Modul 4 von MeWi?

Gibt es verpflichtende Praktika?

Welche studienbegleitenden Angebote gibt es (Ausland, Praktika etc.)?

Wie gut lässt sich ein Auslandsaufenthalt in den Studienverlauf integrieren?

Kann man ein Auslandssemester machen, ohne dass man die Regelstudienzeit von 6 Semestern verlängern muss?

Vorkenntnisse

Welche Voraussetzungen/Interessen sollte man mitbringen?

Welche Vorlesungen soll ich im ersten und zweiten Semester auf alle Fälle besuchen/besucht haben, um gute Voraussetzungen für den weiteren Verlauf zu haben?

Sollte ich Programmierung/Mathe im Vorfeld vorbereiten?

Brauche ich Vorwissen in Bezug auf Programmierung oder Informatik?

Wieviel Aufwand bringt das Studium mit sich?

Wie hoch sind die Anforderungen in diesem Studiengang?

Studentisches Leben

Wo ist die Mensa?

Woher bekomme ich eine Mensa-Karte?

Wann ist vorlesungsfreie Zeit/Vorlesungszeit?

Wie finde ich in der ersten Woche meine Räume?

Wie finde ich mich schnell auf dem Unigelände zurecht?

Wie bekomme ich einen Sportausweis?

Wie viele Leistungspunkte brauche ich, um das Studium erfolgreich zu beenden, und wie sind sie unterteilt?

Bachelor- und Masterstudiengang

Gibt es einen Masterstudiengang?

Ist es empfehlenswert einen Master in Medieninformatik zu machen?

Welche Angebote nach den Bachelor (Master, ...)?

Ich habe keine Ahnung, was ich in Abschlussarbeit schreiben soll und wie.

Wie setzt sich die Abschlussnote zusammen?

Das Expertenteam (in Form der Kursteilnehmer) erhielt die Aufgabe, passende Antworten auf die vorgegebenen Fragen zu formulieren. Sofern die Experten der Meinung waren, dass sie weitere Informationen vom User benötigen, um eine Frage zu beantworten, so sollte dies ebenfalls verbalisiert werden (in Form von Zwischenfragen).

Das Expertenteam sollte das Gegenstück zu den beiden vorherigen Zielgruppen – Schüler und Studenten – darstellen und nicht aus der Sicht des Nutzers agieren, sondern sich wie ein Chatpartner bzw. wie der zu erstellende Chatbot verhalten. Dieser Perspektivenwechsel sollte die Datenerhebung komplettieren, jedoch aufgrund der geringen Teilnahme an der Umfrage bzw. den stichpunktartigen, knappen oder gar fehlenden Antworten und dem Zeitmangel am Ende der Sitzung fiel die Entscheidung, die Datensätze der Expertenbefragung im Sprachverarbeitungskurs nicht zu verwenden.

Die gesammelten Fragen wurden anschließend automatisch – durch Lemmatisierung – und per Hand – in einer *Excel*-Tabelle und über das Tool *XMind* – kategorisiert und geclustert. Dieser Vorgang wird im Folgenden unter dem Punkt Kategorisierung näher erläutert.

7 Kategorisierung

Nach dem Sammeln relevanter Fragestellungen an die Studiengänge der Informationswissenschaft und Medieninformatik, erfolgte im nächsten Schritt das Clustern bzw. Kategorisieren der gesammelten Daten.

7.1 Kategorisierung durch Excel

Im ersten Schritt wurden die Daten tabellarisch in *Excel* in logische Einheiten geclustert. Insgesamt handelt es sich dabei um 515 Fragen, die durch den Fragebogen erhoben und um 71 Fragen, die am Hochschultag der Universität Regensburg gesammelt wurden. Dabei ist noch zu erwähnen, dass die gesammelten Fragen unverändert in Originalform – zum Beispiel mit Rechtschreibfehlern – in die Liste eingetragen wurden, um die Daten nicht zu verfälschen.

7.1.1 Fragebogen

Durch die erste Betrachtung der gesammelten Fragen ergaben sich neun grobe Kategorien, die iterativ verfeinert worden sind. Vorab fällt jedoch auf, dass es sich hierbei um drei Phasen handelt, die Fragen vor dem Studium abdecken – wie zum Beispiel Einschreibungsfristen –, Informationen während des Studiums betreffen – zum Beispiel zu belegende Kurse – und zuletzt Perspektiven nach dem Studium, wie zum Beispiel berufliche Aussichten.

Folgende Hauptkategorien wurden definiert:

- Beruf
- Vorkenntnisse
- Praktika und Ausland
- Fachkombination und Wechsel
- Allgemeines Studentenleben
- Klausur und Note
- Bachelor und Master
- Individuelles Studium und
- Kontextfreie Scherzfragen.

Die genannten Hauptkategorien wurden bei weiterer Bearbeitung durch einige Unterkategorien ergänzt, die im Folgenden näher erläutert werden:

- Beruf
 - Berufliche Aussichten und Wirtschaftliche Fragen
 - Branchen und Tätigkeitsfelder
- Vorkenntnisse
 - Voraussetzungen und Kenntnisse
 - Voraussetzungen und Kenntnisse für einen bestimmten Studiengang
 - Voraussetzungen und Kenntnisse für bestimmte Kurse in Hinblick auf deren Abhängigkeiten zueinander
 - Programmier- und Mathekenntnisse
 - Anforderungen und Aufwand
- Praktika und Ausland
 - Praktikum
 - Auslandsaufenthalt
- Fachkombination und Wechsel
 - Fachkombination im Allgemeinen
 - Fachkombination in Hinblick auf die Medieninformatik
 - Fachkombination in Hinblick auf die Informationswissenschaft
 - Fachkombination in Hinblick auf die Medienwissenschaft
 - Informationswissenschaft, Medieninformatik und Medienwissenschaft als Fachkombination
 - Medieninformatik und Medienwissenschaft als Fachkombination
 - Informationswissenschaft und Medienwissenschaft als Fachkombination
 - Wechsel
 - Fakultätsübergreifende Fachkombinationen

- Frei kombinierbares Nebenfach (FKN)
- Allgemeines Studentenleben
 - Mensa
 - Semesterkalender
 - Räumlichkeiten
 - Sport
 - Leistungspunkte und Credits im Allgemeinen
 - Leistungspunkte und Credits für Vorlesungen
 - Leistungspunkte und Credits pro Semester
 - Leistungspunkte und Credits für den Wahlbereich
 - Außerstudentisches Leben
 - Allgemeines und Erstsemesterfragen
 - Beratung und Ansprechpartner
 - Navigation
 - Urlaubssemester
 - Bibliothek
- Klausur und Note
 - Termine
 - Klausurvorbereitung
 - Anmeldung und Wiederholung
 - Prüfung
 - Fragen bezüglich der Webseite der Universität Regensburg
- Bachelor und Master
 - Master
 - Bachelor
 - Bachelorarbeit und Abschlussnote

- Anmeldung und Einschreibung
- Individuelles Studium
 - Inhalte und Lernziele
 - Kurse und Wahlbereich
 - Module
 - Studienverlauf und Planung
 - Dozenten
 - Dauer des Studiums
 - Numerus Clausus
 - Schwerpunkte des Studiums
 - Abgrenzung bzw. Überschneidung der Studiengänge
 - Stundenplan
 - Spezielle Fragen zur Informationswissenschaft
 - Spezielle Fragen zur Medieninformatik
 - Grundlagen- und Orientierungsprüfung
 - Praxisorientiertheit

und

- Kontextfreie Scherzfragen
 - Verbesserungsvorschläge und Kritik
 - Statistische Fragen

Somit ergeben sich letztlich neun Hauptkategorien und insgesamt 59 Unterpunkte.

Beginnend mit der Hauptkategorie BERUF handelt es sich bei dem Teilcluster *Berufliche Aussichten und Wirtschaftliche Fragen* pauschal darum, welche beruflichen Aussichten ein Student nach erfolgreichem Abschluss des Studiums hat. Der wirtschaftliche Aspekt beläuft sich darauf, dass nach Chancen auf dem Arbeitsmarkt gefragt wird, welche je-

doch zusätzlich noch von einigen anderen Faktoren – wie zum Beispiel von der Konjunktur eines Landes – abhängig sind. Im zweiten Teilcluster *Branchen und Tätigkeitsfelder* handelt es sich ableitbar darum, in welchen Branchen man mit dem jeweiligen Studiengang tätig werden kann bzw. um eine weitere Ebene runtergebrochen, welchem Jobprofil man entspricht und welche spezifische Tätigkeit mit dem jeweiligen Studiengang ausübbar ist.

Die zweite Überkategorie VORKENNTNISSE umfasst fünf Unterkategorien. Erstere Teilkategorie *Voraussetzungen und Kenntnisse* bezieht sich allgemein auf die Frage, welche Voraussetzungen bzw. Vorkenntnisse vor dem Studium wünschenswert wären. Weiterhin handelt es sich bei dem Punkt *Voraussetzungen und Kenntnisse für einen bestimmten Studiengang* um denselben Ausgangspunkt, jedoch auf einen bestimmten Studiengang beschränkt. In dem Cluster *Voraussetzungen und Kenntnisse für bestimmte Kurse in Hinblick auf deren Abhängigkeiten*, beziehen sich die Fragen auf Vorkenntnisse, die bestimmte Kurse betreffen bzw. welche Kurse von anderen abhängig sind. Häufig wurde nach *Programmier- und Mathekenntnissen* gefragt, weshalb dieser Punkt einen eigenständigen Cluster bildet. Inhaltlich handelt es sich hierbei darum, ob das Programmieren oder die Mathematik einen sehr relevanten Teil innerhalb des Studienganges einnehmen und ob man nach Abschluss des Studiums die Kunst des Programmierens beherrscht. Als letzter Punkt unter *Anforderungen und Aufwand*, wurde pauschal gefragt, wie viel Aufwand bzw. Lernaufwand das Studium mit sich bringt und wie hoch die Anforderungen an den Studiengang sind.

Der Punkt PRAKTIKA UND AUSLAND umfasst unter dem Cluster *Praktikum* Fragen, welche Möglichkeiten es im Laufe des Studiums gibt, ein Praktikum zu absolvieren bzw. ob ein bestimmter Studiengang Pflichtpraktika im Studienverlauf vorgesehen hat. Die Unter-

kategorie *Auslandsaufenthalt* deckt die Thematik nach möglichen Auslandssemestern und der Vereinbarung zwischen dem Studiengang und einem Auslandsaufenthalt ab.

In der Kategorie FACHKOMBINATION UND WECHSEL – die aus zehn Unterkategorien besteht – wird mit einer allgemeinen Betrachtung auf mögliche Fachkombinationen begonnen im Cluster *Fachkombination Allgemein*. Hierbei beschränken sich die Fragen vor allem darauf, welche Fachkombinationen empfohlen werden und überhaupt möglich bzw. (sinnvoll) studierbar sind. Das allgemeine Konzept wird dann auf die einzelnen Studiengänge bzw. deren Kombinationen aufgebrochen. Beginnend mit dem Cluster *Medieninformatik als Fachkombination*, bei dem mögliche Fächerkombinationen mit diesem Fach abgefragt werden bzw. ob die Medieninformatik auch als einzelnes Fach studiert werden kann. Äquivalent und inhaltsübergreifend dazu die *Informationswissenschaft als Fachkombination*, sowie die *Medienwissenschaft als Fachkombination*. In dem Cluster *Informationswissenschaft, Medieninformatik und Medienwissenschaft als Fachkombination* wird nach Nutzen bzw. Sinn gefragt, diese Dreierkombination zu studieren. Bei der *Medieninformatik und Medienwissenschaft als Fachkombi* interessierten sich die Studenten dafür, ob man die genannten Fächer auch als erstes und zweites Hauptfach belegen könnte. Unter der Teilkategorie *Wechsel* sind inhaltlich Fragen über die Möglichkeit eines Fächerwechsels während des Studiums eingeordnet. Weiterhin wurde in dem Cluster *Fakultätsübergreifende Fachkombinationen* gefragt, ob es möglich sei, eines der Fächer – Informationswissenschaft, Medieninformatik oder Medienwissenschaft – mit einem Fach außerhalb der Sprach-, Literatur- und Kulturwissenschaftlichen Fakultät zu kombinieren. Als letzter Unterpunkt wurde das *Frei Kombinierbare Nebenfach (FKN)* aufgelistet, bei dem es um Fragen bezüglich des Ablaufs eines FKNs ging, welche FKN-Fächer das Studium sinnvoll ergänzen und ob es definierte Pflichtkurse bei dieser Art von Nebenfach gibt.

Der nächste große Punkt ALLGEMEINES STUDENTENLEBEN umfasst 14 Unterkategorien. Beginnend mit dem Teilcluster *Mensa*, der hauptsächlich Fragen über die Mensakarte umfasst, darauffolgend die Kategorie *Semesterkalender*, deren Hauptthemen sich um die Semesterferien, die Dauer eines Semesters und um die Zeitspanne der vorlesungsfreien bzw. Vorlesungszeit drehen. Betreffend der *Räumlichkeiten*, handelt es sich um Fragen, wie in welchen Räumen eine Vorlesung stattfindet bzw. welche Klausur in welchen Räumen geschrieben wird. Der Punkt *Sport* dreht sich um Informationen um den Sportausweis und Fragen rund um das Sport- und Freizeitangebot der Universität Regensburg. Ein weiterer großer Bereich deckt das Informationsbedürfnis in Hinblick auf *Leistungspunkte bzw. Credits* ab. Vorerst wird im Allgemeinen das Punktesystem des Bachelors behandelt. Dabei geht es um die rudimentäre Frage, was überhaupt ein Leistungspunkt ist und wie viele Leistungspunkte man insgesamt braucht um das Studium erfolgreich absolvieren zu können. Eine Ebene darunter wird die Fragen in nach den Punkteanzahlen für eine bestimmte Vorlesung bzw. einen bestimmten Kurs behandelt. Fortführend wurde auch nach der empfohlenen Punkteanzahl pro Semester gefragt, und was es für Möglichkeiten gibt, seinen Wahlbereich punktemäßig zu füllen. Ein weiterer Punkt betrifft das *außerstudentische Leben*, gesammelt wurden hierbei Fragen bezüglich der Wohnungssuche, dem Semesterticket, den Parkmöglichkeiten und Informationen zu einem Teilzeitstudium gewünscht. Unter der Teilkategorie *Allgemeines und Erstsemesterfragen*, befinden sich Fragen, die zum Beispiel die größten Studienirrtümer oder den Cafetenschein, sowie auch Einsatzmöglichkeiten in studentischen Organisationen betreffen. Das Topic *Beratung und Ansprechpartner* umfasst Fragen rund um die studentische Beratung durch fakultätsinterne Ansprechpartner. Dadurch, dass die Universität Regensburg eine Campus-Hochschule ist, stellt sich natürlich auch die Frage nach der *Navigation*, wodurch sich ein großes Informationsbedürfnis nach der Lokalisierung von Hörsälen, Räumen und allgemeiner Orientierung auf dem Unigelände ergibt.

Ein weiterer Teilcluster *Urlaubssemester* dreht sich um die grundsätzliche Frage nach der Begriffsklärung und ob ein Urlaubssemester erlaubt sei. Die letzte Unterkategorie *Bibliothek* beschäftigt sich mit der Funktionsweise der Bücherausleihe und des Bibliothekssystems sowie mit der Frage ob relevante Bücher gekauft werden müssen oder von der Bibliothek bereitgestellt werden.

Die sechste Hauptkategorie KLAUSUR UND NOTE unterteilt sich in fünf Unterkategorien, die als erstes die Termine umfasst. In dem Topic *Termine* befinden sich Fragestellungen, die vor allem die Prüfungstermine umfassen. Unter der Teilkategorie *Klausurvorbereitung* sind Fragen rund um das Lernen für Klausuren bis hin zu vorhandenen Unterlagen zu finden. Ein weiterer Punkt beschäftigt sich mit der *Anmeldung und Wiederholung* von Prüfungen bzw. Klausuren. Der Punkt *Prüfungen* beschäftigt sich unter anderem mit der Dauer von Prüfungen und der Art von Prüfungen bis hin zur Anmeldung. In der letzten Teilkategorie *Webseite* bestehen Fragen zur Universitätshomepage in Bezug auf die fakultätsinterne sowie fakultätsfremde Navigation.

Die nächste definierte Überkategorie behandelt Fragen rund um den BACHELOR- UND MASTERSTUDIENGANG. Beginnend mit der Unterkategorie *Master*, welche sich um das Vorhandensein eines Masterstudienganges und um diverse Masterangebote dreht, ergänzend dazu der Teilcluster des *Bachelors*, der sich damit beschäftigt, welche Fächer bzw. Lehrveranstaltungen in die Endnote des Bachelors miteinfließen, sowie ob bestimmte Studiengänge zulassungsbeschränkt sind. Thematisch anschließend dazu Fragestellungen zur *Bachelorarbeit und Abschlussnote*, welche von Abschlussarbeitsthemen bis hin zur Zusammensetzung der Abschlussnote reichen. Das letzte Topic dieser Hauptkategorie *Anmeldung und Einschreibung*, deckt Fragen ab, welche sich mit Einschreibungsterminen und Voraussetzung für die Einschreibung beschäftigen.

Eine weitere wichtige Kategorie bildet der Punkt INDIVIDUELLES STUDIUM, welcher aus 14 Unterkategorien besteht. Erstere, *Inhalte und Lernziele* bezieht sich auf Fragen, welche die Inhaltsebene des Studiums betreffen, sowie die Hauptthemen und Interessengebiete, die während des Studiums abgedeckt werden. In der nächsten Unterkategorie *Kurse und Wahlbereich* handelt es sich um Themengebiete, die den Wahlbereich und das Kursangebot betreffen, vor allem im Hinblick auf den erfolgreichen Abschluss des Studiums. Unter dem Topic *Module* werden Fragen aufgelistet, welche hauptsächlich Informationen über Pflicht- und Wahlmodule abdecken. Die Unterkategorie *Studienverlauf und Planung* bezieht sich auf Fragen, die Bezug auf den Aufbau des Studiums nehmen, überschneidende Veranstaltungen thematisieren, die optimale Wahl und Reihenfolge von Lehrveranstaltungen während des Studiums abdecken und auch welche Individualisierungsmöglichkeiten im Studiengang geboten werden. Der Punkt *Dozenten* umfasst den Zuständigkeitsbereich der jeweiligen Lehrpersonen, sowie persönliche Fragen z.B. zum Thema der Beliebtheit. Eine weitere Unterkategorie bildet das Topic *Dauer des Studiums*, welche die Regelstudienzeit sowie den maximalen Zeitraum umfasst, den ein Student zum erfolgreichen Abschluss des Studiums beziehen kann. Der Cluster *Numerus Clausus* beantwortet Fragen in Hinblick auf individuelle Zulassungsbeschränkungen bezüglich eines Studienganges und die Höhe bzw. den Wert des NCs. Unter dem Punkt *Schwerpunkte des Studiums* werden selbsterklärend Schwerpunkte des jeweiligen Studiengangs bzw. der Studiengänge in Kombination aufgelistet. Folglich der Cluster *Abgrenzung und Überschneidungen der Studiengänge*, welcher sich mit den Unterschieden der verschiedenen Kombinationsmöglichkeiten der Studiengänge Informationswissenschaft, Medieninformatik und Medienwissenschaft, auf Ebene der Module, sowie auf rudimentärer Ebene beschäftigt. Die Unterkategorie *Stundenplan* deckt Fragen ab, welche sich mit der Erstellung eines Stundenplans auseinandersetzen. Informationsbedarf besteht auch in Hinblick auf die *Informationswissenschaft*, in Form von grundsätzlichem Aufklärungsbe-

darf über den Studiengang, sowie über die Lerninhalte. Äquivalent dazu gibt es Fragen zum Studiengang der *Medieninformatik*, welche sich mit den thematisch gelegenen Schwerpunkten des Studienganges und der Bedeutung der Begrifflichkeit beschäftigen. Der Cluster *Grundlagen- und Orientierungsprüfung* thematisiert vor allem, zu welchem Zeitpunkt welche Pflichtveranstaltungen bereits abgelegt werden müssen bzw. welche Module zu welchem Zeitpunkt abgeschlossen sein müssen. Als letzter Punkt unter der umfangreichen Hauptkategorie INDIVIDUELLES STUDIUM bildet selbsterschließend die *Praxisorientiertheit* eines bestimmten Studienganges.

Die letzte Oberkategorie, welche die sonstigen Fragen beinhaltet, ist der Cluster KONTEXTFREI UND SCHERZFRAGEN. Unter diesem Punkt werden *Scherzfragen* aufgelistet, welche nichts mit Informationen bezüglich des Studiums bzw. der Studiengänge zu tun haben. Dabei handelt es sich um Fragen, wie zum Beispiel den Flirtfaktor im Studienfach der Informationswissenschaft, oder um subjektive Fragen, ob der Studiengang Spaß mache. Ein weiteres Untertopic bilden *Verbesserungsvorschläge und Kritik*, welcher die Zufriedenheit bezüglich des Studiums betrifft und Anregungen zur Verbesserung des Studentenlebens umfasst. Der letzte Punkt *statistische Fragen* umfasst Auflistungen zur Geschlechterverteilung eines bestimmten Studienganges sowie das Durchschnittsalter der Studenten.

Diese Aufteilung in die genannten Hauptkategorien und Unterkategorien wurde auch beim Clustering der Fragen des Hochschultages beibehalten, um die Konsistenz und Übertragbarkeit der Daten zu bewahren.

7.1.2 Hochschultag der Universität Regensburg

Im Gegensatz zu der umfangreichen Datensammlung, die sich durch die Auswertung des Fragebogens ergab, fallen beim Clustering der Fragen des Hochschultages der Universität Regensburg einige Kategorien und Unterkategorien heraus. Dabei handelt es

sich um inhaltliche Einbußen in den Hauptkategorien VORKENNTNISSE, FACHKOMBINATION UND WECHSEL, ALLGEMEINES STUDENTENLEBEN, BACHELOR UND ABSCHLUSSNOTE, INDIVIDUELLES STUDIUM sowie die Kategorie KONTEXTFREI UND SCHERZFRAGEN.

Der Hauptcluster KLAUSUR UND NOTE als auch PRAKTIKA UND AUSLAND fällt bei dieser Datenerhebung komplett heraus.

Zusammenfassend ist zu sagen, dass von den neun Hauptclustern sich insgesamt sieben Kategorien inhaltlich lückenhaft gestalten, während zwei Kategorien komplett aus der Auswertung ausgeschlossen werden. Bei den Unterkategorien handelt es sich um folgende Themengebiete:

- Vorkenntnisse
 - Voraussetzungen und Kenntnisse bezüglich eines bestimmten Studienganges
 - Voraussetzungen und Kenntnisse für bestimmte Kurse hinsichtlich ihrer Abhängigkeiten
- Fachkombination und Wechsel
 - Medieninformatik als Fachkombination
 - Medienwissenschaft als Fachkombination
 - Medieninformatik, Medienwissenschaft und Informationswissenschaft als Fachkombination
 - Medieninformatik und Medienwissenschaft als Fachkombination
 - Medienwissenschaft und Informationswissenschaft als Fachkombination
 - Wechsel
 - Fakultätsübergreifende Fachkombinationen
 - Frei Kombinierbares Nebenfach (FKN)

Das Fehlen des Informationsbedürfnisses – zum Beispiel nach den Studiengängen Medieninformatik und Medienwissenschaft – in diesem Cluster ist aus dem Faktum, dass es sich beim Hochschultag der Universität Regensburg um Schüler handelte, welche sich ausschließlich über die Informationswissenschaft informieren wollten abzuleiten. Der Ausgangspunkt, dass die Besucher der Zielgruppe von Schülern entsprechen, spiegelt sich auch im Informationsbedürfnis der folgenden Cluster wider insofern, dass es sich bei den Fragen eher um ein rudimentäres Interesse bezüglich des Studienganges handelt und spezifische Fragen erst im Laufe des Studiums bzw. des Studentenalltages auftauchen. Als vorangehendes Beispiel kann die Hauptkategorie INDIVIDUELLES STUDIUM genannt werden, welche die Unterkategorie *Grundlagen und Orientierungsprüfung* besitzt. An dem Beispiel ist deutlich zu erkennen, dass ein Informationsbedürfnis in Hinblick auf dieses Interessengebiet erst in den ersten Semestern einer Studentenlaufbahn entstehen kann, da es sich hierbei um eine sehr individuelle Frage handelt.

Ein weiterer inhaltsleerer Cluster ist unter dem Begriff *Allgemeines und Studentenleben* zusammengefasst und umfasst folgende leere Topics:

- Mensa
- Semesterkalender
- Sport
- Leistungspunkte und Credits Allgemein
- Leistungspunkte und Credits Vorlesung
- Leistungspunkte und Credits Semester
- Leistungspunkte und Credits Wahlbereich
- Außerstudentisches Leben

- Beratung und Ansprechpartner
 - Navigation
 - Urlaubssemester
 - Bibliothek
 - Bachelor und Master
 - Bachelor
 - Bachelorarbeit und Abschlussnote
 - Anmeldung bzw. Einschreibung
 - Individuelles Studium/Fakultätsbezogen
 - Module
 - Stundenplan
 - Grundlagen und Orientierungsprüfung
 - Praxisorientiertheit
- und als letztes der Cluster
- Kontextfrei und Scherzfragen
 - Verbesserungsvorschläge und Kritik
 - Statistische Fragen

Inhaltlich betrachtet handelt es sich wie bereits im Kapitel 7.1.1 beschrieben um dieselben Grundfragen, die in verschiedenen Kombinationsmöglichkeiten auftauchen. Beginnend mit der Hauptkategorie BERUF, welche aus drei Unterkategorien besteht, handelt es sich bei dem Topic *Berufliche Aussichten und Wirtschaftliche Fragen* pauschal um die Frage, was man mit einem bestimmten Studiengang, in dem Fall mit der Informationswissenschaft, später werden kann. Der Punkt *Branchen und Tätigkeitsfelder* deckt selbsterklärend Fragen ab, welche sich mit dem zukünftigen Arbeitsgebiet beschäftigen. In der letzten Unterkategorie *Berufschancen* werden die Chancen auf dem Arbeitsmarkt angesprochen.

Das Cluster VORKENNTNISSE umfasst in dem Topic *Voraussetzungen und Kenntnisse* Informationen darüber, welche Voraussetzungen für den Studiengang Informationswissenschaft mitgebracht werden müssen. Weiterhin handelt es sich um Fragen, welche das Vorhandensein von *Programmier- und Mathekenntnissen* und Informationen bezüglich des Arbeitsaufwandes bzw. Anforderungen an das Studium betreffen.

Fortführend unter dem Punkt FACHKOMBINATION UND WECHSEL wird unter dem Teilcluster *Fachkombination Allgemein* nach möglichen Sprachen gefragt, welche in Kombination studiert werden können. Außerdem wurde aufgrund der Intention des Informationstages oft nach Fachkombinationen gefragt, welche mit der Informationswissenschaft kompatibel sind.

Unter der Hauptkategorie ALLGEMEINES STUDENTENLEBEN wird unter dem Topic *Räumlichkeit* die Frage nach dem Raum D003 aufgelistet. Weiterhin unter dem Punkt *Allgemeines und Erstsemesterfragen* das Bedürfnis nach Informationen über diverse Einschreibungstermine.

Als nächste Kategorie, benannt als BACHELOR UND MASTER, befindet sich unter dem Punkt *Master* die Frage, ob es sinnvoll wäre, einen Masterstudiengang nach dem Bachelor zu erwägen.

Wie schon im Kapitel davor bildet die Kategorie INDIVIDUELLES STUDIUM den größten Cluster mit der höchsten Anzahl an Fragen. Beginnend mit dem Teilcluster *Inhalte und Lernziele*, welche die Frage nach einem allgemeinen Informationsbedürfnis bezüglich der Informationswissenschaft beinhaltet, über die Kategorie *Kurse und Wahlbereich*, welche Fragen abdeckt, die den Schwierigkeitsgrad bestimmter Lehrveranstaltungen betrifft, sowie die Höhe bzw. Anzahl der Leistungspunkte für bestimmte Kurse. Unter dem

Punkt *Studienverlauf und Planung* ist die Frage nach den vermittelten Inhalten im ersten Semester aufgelistet. Die Teilkategorie *Dozenten* beschäftigt sich mit der individuellen Frage, wer Manuel Burghardt sei. Weiterhin findet man unter dem Cluster *Dauer des Studiums* Fragen bezüglich der Regelstudienzeit. Der Punkt *Numerus Clausus* umfasst die Frage, ob der Studiengang der Informationswissenschaft zulassungsbeschränkt – in Form eines festgelegten NCs – oder zulassungsfrei ist. Die Frage nach den inhaltlichen Schwerpunkten des Studiums der Informationswissenschaft wird unter der Teilkategorie *Schwerpunkte des Studiums* eingeordnet. Der Cluster *Abgrenzung bzw. Überschneidungen der Studiengänge* beschäftigt sich mit den verschiedenen Kombinationsmöglichkeiten sowie der inhaltlich vermittelten Unterschiede der Studiengänge Medieninformatik, Medienwissenschaft und Informationswissenschaft. Unter dem Punkt *Informationswissenschaft* wurden Fragen eingeordnet, welche sich um die Begriffserläuterung des Studienganges drehen, um die grundsätzliche Frage, was die Informationswissenschaft überhaupt sei und mit was sie sich beschäftigt, sowie Informationen über den technischen Anteil zum Beispiel in Form der Erlernung verschiedener Programmiersprachen während des Studiums. Das Topic *Medieninformatik* beschäftigt sich mit grundsätzlichen Informationen zum Studiengang. Eine neue Kategorie, welche während des Hochschultages der Universität Regensburg aufgrund einer hohen Fragenfrequenz generiert wurde, ist die *Informatik*. Diese Unterkategorie behandelt Fragen, die den informatischen Anteil der Informationswissenschaft abdecken bzw. einen Ähnlichkeitsvergleich zu der klassischen Informatik anstellen.

Als letzte und neunte Kategorie der Cluster KONTEXTFREI UND SCHERZFRAGEN, der insgesamt zwei Fragen beinhaltet. Aufgrund der einerseits prototypischen Gestaltung des Chatbots *Elise* und andererseits der fehlenden Datenbasis, um spezifische Fragen bereits beantworten zu können, wurde bei einer inkompatiblen Frage die Meldung „Daran hat

der Autor der *AIML*-Datei nicht gedacht“ ausgegeben. Aufgrund dessen wurde dem Chatbot die Frage gestellt, woran der Autor der *AIML*-Datei nicht gedacht habe, gestellt. Als letzte Frage unter einer neu generierten Teilkategorie *Allgemein Smalltalk* eingeordnet die Frage danach, was Information überhaupt sei.

Nach dem Prozess der Datensammlung sowohl in Form eines Fragebogens, als auch am Hochschultag der Universität Regensburg, wurde in Hinblick auf die Mächtigkeit der zukünftigen *AIML*-Struktur des Chatbots *Elise* ein weiterer Schritt – nämlich die Kategorisierung durch das Tool *Xmind* –, welche im Folgenden näher erläutert wird, durchgeführt.

7.2 Kategorisierung durch das Tool Xmind

Um noch einen Schritt weiter zu gehen, wurden die bereits geclusterten Fragen ein weiteres mal überarbeitet, vor allem im Hinblick darauf, verschiedene Kombinationsmöglichkeiten – in Form von Synonymen – eines Themenbereiches später in der Useranfrage an den Chatbot abfangen zu können. Dies erwies sich in Hinsicht auf die Erstellung des Dictionaries – siehe Kapitel Dictionary – als besonders sinnvoll und hilfreich. Als allgemeine Vorgehensweise wurde sowohl bei der Bearbeitung der Daten des Fragebogens, als auch des Hochschultages versucht, inhaltstragende Wörter zu extrahieren und diese dann mit einem zusammenhängenden Verb bzw. einer passenden ergänzenden Wortstruktur zu verknüpfen.

7.2.1 Fragebogen und Hochschultag

Wie schon erwähnt, dient die bereits erstellte und kategorisierte *Excel*-Liste als Ausgangspunkt für die weitere Extraktion von Synonymen und zusammenhängenden Wortstrukturen. Dabei ergaben sich folgende Themencluster:

- Abschlüsse
 - Individuelles Studium
 - Leistungsnachweise
 - Praktischer Einsatz und Ausland
 - Skills
 - Studentenleben
 - Studiengänge und Kombinationen
 - Sonstiges
- und
- nach dem Studium

Da das Vorgehen in jeder Kategorie denselben Mustern und Regeln folgt, wird im Folgenden exemplarisch die Datei *nachdemStudium* – siehe Abbildung 20: ‘Xmind | nach dem Studium’ –, welche die Kategorie *Beruf* bearbeitet, aufgezeigt.

Keywords, welche dem Chatbot ein Indiz dafür sein können, dass es sich um Fragen rund um den Themenbereich Beruf handelt, wurden in ihrer Grundform bzw. in ihrer Flexionsform extrahiert. Dabei konnten 13 Begriffskombinationen herausgefiltert werden.



Abbildung 20: Xmind | nach dem Studium

Der blau visualisierte Knoten bildet allgemein in dem Tool *Xmind* die Hauptkategorie bzw. das behandelte Themengebiet ab. Eine Ebene darunter sind die nachfolgenden Knoten abgebildet, welche die verschiedenen Kombinationsmöglichkeiten – hier des Clusters *Beruf* – abbilden. Diese werden wiederum eine Ebene darunter von möglichen Ergänzungen oder durch ein begleitendes Verb vervollständigt.

Im Folgenden werden alle inhaltstragenden Begriffe in ihrer Grundform aufgelistet und durch jeweils ein Beispiel untermalt.

- Beschäftigungsmöglichkeit
 - Welche *Beschäftigungsmöglichkeiten* habe ich mit abgeschlossenem Bachelor-Studium?

- Arbeiten
 - In welchen Tätigkeitsfeldern kann ich *arbeiten*?
- Tätigkeitsfeld
 - In welchen *Tätigkeitsfeldern* kann ich arbeiten?
- Branche
 - In welcher *Branche* werde ich nach dem Studium sehr gute Jobchancen haben?
- Job
 - Welche *Jobperspektiven* habe ich nach Abschluss des Studiums?
- Beruf
 - Welche *Berufsfelder* stehen mir nach dem Studium offen?
- Nach Abschluss bzw. nach (dem) Studium
 - Welche *Jobperspektiven* habe ich nach *Abschluss* des *Studiums*?
- Zukunft
 - Welche *Zukunftsmöglichkeiten* bieten diese Studiengänge heutzutage?
- Arbeitsmarkt
 - Welche Chancen habe ich nach meinem Studium (Medienwissenschaft) am *Arbeitsmarkt*?
- Beruflich
 - *Berufliche* Aussichten?
- Später
 - Inwiefern bereitet das Studium auf die *späteren* Berufe vor?

und
- Anfangen
 - Was kann ich mit meinem Studium *anfangen*?

Die weiteren bearbeiteten *Xmind*-Dateien sind im Anhang einsehbar.

7.3 Dictionary

Bereits während der Kategorisierung der gesammelten Fragen – sowohl in der *Excel*-Tabelle, als auch durch das Tool *Xmind* –, fielen immer wieder Wortwiederholungen und ähnliche Begriffe auf. Diese Terme erwiesen sich für die Verständlichkeit des Satzes bzw. der Frage als inhaltstragend. Durch diese Erkenntnis erwies es sich als folgerichtig, ein Dictionary anzulegen, welches zu allererst die Synonymie aller Begriffe in sich vereint. Dabei wurden die gesammelten Wörter auf einen Hauptterm reduziert, welcher alle Terme in sich vereint. Der größte Hauptterm – in Abbildung 21: ‘Dictionary 1 | am Beispiel Beruf’ dargestellt – bildete dabei der Begriff des Berufes.

BERUF:arbeitsleben(anfangen[5],abschluss[5],abschlüsse[5]),berufliche aussichten(abschlussstudium[5],abschlussstudiums[5]),berufsfelder(nach studium[5]),chancen(arbeitsmarkt[5],arbeitsleben[5],beruf[5],nach studium[5],abschluss[5]), nach studium(arbeitsmarkt[5],arbeitsmärkte[5],arbeiten[5],machen[5],anstreben[5],tun[5],ausüben[5],be kommen[5],chance[5],chancen[5],arbeitsplatz[5],arbeitsplätze[5],arbeitsleben[5],berufsfeld[5],beruf sfelder[5],beruf[5],berufe[5],werden[5],berufsmöglichkeit[5],berufsmöglichkeiten[5],berufschance[5],berufschancen[5],arbeitsstelle[5],gewerbe[5],tätigkeit[5],tätigkeitsfeld[5],tätigkeitsfelder[5],profess ion[5],stellenangebot[5],stelle[5],berufsweg[5],jobperspektive[5],jobperspektiven[5])

Abbildung 21: Dictionary 1 | am Beispiel Beruf

Die Syntax WORT(NACHBAR[ENTFERNUNG],NACHBAR[ENTFERNUNG],...) definiert dabei einen zusammenhängenden Term, welcher in Abhängigkeit zu seinen Nachbarwörtern betrachtet wird. Taucht also zum Beispiel die Kombination ARBEITSLEBEN ANFANGEN[5] in dem Satz „Was kann ich später mit dem Studiengang im Arbeitsleben anfangen?“, so wird der Inhalt dieses Satzes im ersten Schritt auf die Kategorie BERUF reduziert. Dabei kann der Abstand der beiden Begriffe ARBEITSLEBEN und ANFANGEN in einem maximalen ABSTAND VON FÜNF WÖRTERN voneinander entfernt sein. Mit diesem ersten Auswahlverfahren ist es möglich, einen ersten thematischen Zusammenhang in

Bezug auf die Frage herzustellen. Da die Verarbeitung des Dictionarys prozedural abläuft, wurde bei der Sortierung des Wörterbuches – siehe Abbildung 22: ‘Dictionary 1 | Sortierung des Wörterbuches’ – darauf geachtet, vom Detailreichen ins Grobe zu ordnen. Der Grundgedanke hierbei war, dass sich verschiedene reduzierte Kategorien überschneiden können und deshalb der Match einer Kombination von Wörtern eine höhere und sichere Trefferquote bezüglich des thematischen Kontextes liefert, als nur ein Schlagwort. Am Beispiel der Kategorie KLAUSUR sieht man, dass erst nach den verschiedenen Kombinationsmöglichkeiten synonyme Schlagwörter wie zum Beispiel PRÜFUNG oder EXAMEN aufgeführt werden.

```
KLAUSUR:statt(klausur[5],klausuren[5],prüfung[5],prüfungen[5],test[5],tests[5],examen[5],examina[5],modulklausur[5],modulklausuren[5]),stattfinden(klausur[5],klausuren[5],prüfung[5],prüfungen[5],test[5],tests[5],examen[5],examina[5],modulklausur[5],modulklausuren[5]),leistung,leistungen,studienleistung,studienleistungen,prüfung,prüfungen,test,tests,examen,examina,modulklausur,modulklausuren
```

Abbildung 22: Dictionary 1 | Sortierung des Wörterbuches

In einem weiteren Schritt wurde die bereits kategorisierte ursprüngliche Frage auf ihren Grundkern reduziert, sodass Stopwörter und selbstdefinierte inhaltsleere Wörter entfernt wurden. So wurde zum Beispiel aus der Frage „Welche beruflichen Aussichten hat man?“ → „Welche berufliche Aussichten“. Die Fragewörter wurden in diesem Schritt noch mitgezogen, da eine Unsicherheit bezüglich ihrer Aussagekraft bestand. Letztendlich wurden ausschließlich die Fragewörter Wo, Wann und Wer in die Liste der inhalts-tragenden Wörter eingereiht.

Beruf	Reduzieren / Stopwörter und Fragen
	Berufliche Aussichten und Wirtschaftliche Fragen
Welche beruflichen Aussichten hat man?	Welche beruflichen Aussichten
Bei welcher Kombination bestehen die besten Berufsaussichten?	welcher Kombination bestehen Berufsaussichten
Wie stehen die Berufschancen?	Wie stehen Berufschancen
Welche Zukunftsmöglichkeiten bieten diese Studiengänge heutzutage?	Welche Zukunftsmöglichkeiten bieten Studiengänge
Mit welchem Studiengang habe ich später bessere Chancen einen Arbeitsplatz in der freien Wirtschaft zu bekommen?	welchem Studiengang später Chancen Arbeitsplatz freien Wirtschaft bekommen
Wie sind die Berufsaussichten?	Wie Berufsaussichten
Welche berufliche Aussichten hab ich mit dem Abschluss des Studiums?	Welche berufliche Aussichten Abschluss Studiums
Berufliche Aussichten?	Berufliche Aussichten

Abbildung 23: Dictionary | Reduzieren der Frage

Der zweite Schritt bestand darin, die übriggebliebenen Begriffe – siehe Abbildung 24: ‘Dictionary | Formulierung des Endtemplates’ – in Verbindung mit dem ersten Dictionary zu bringen. Folglich wurde der Begriff „beruflichen Aussichten“ auf den Hauptterm BERUF reduziert. Die sukzessive Reduktion der Fragen führte letzten Endes zu der Benennung eines Endtemplates, in diesem Fall BERUFALLGEMEIN. Sinn und Zweck dieses Clustering-Prozesses war es, Templates zu definieren, welche dann mit der richtigen Antwort befüllt werden können.

Reduzieren / Stopwörter und Fragen	erste Liste	zweite Liste
Berufliche Aussichten und Wirtschaftliche Fragen		
Welche beruflichen Aussichten	BERUF	BERUFALLGEMEIN
welcher Kombination bestehen Berufsaussichten	FACHKOMBI bestehen BERUF	BERUFALLGEMEIN
Wie stehen Berufschancen	stehen BERUF	BERUFALLGEMEIN
Welche Zukunftsmöglichkeiten bieten Studiengänge	BERUF bieten STUDIENGANG	BERUFALLGEMEIN
welchem Studiengang später Chancen Arbeitsplatz freien Wirtschaft bekommen	STUDIENGANG BERUF BERUF bekommen	BERUFALLGEMEIN
Wie Berufsaussichten	BERUF	BERUFALLGEMEIN
Welche berufliche Aussichten Abschluss Studiums	BERUF BERUF	BERUFALLGEMEIN
Berufliche Aussichten	BERUF	BERUFALLGEMEIN

Abbildung 24: Dictionary | Formulierung des Endtemplates

Im Laufe der weiteren Bearbeitung der umfangreichen Datensammlung wurde die Notwendigkeit eines zweiten Dictionarys – siehe Abbildung 25: ‘Dictionary 2 | am Beispiel Beruf’ – unumgänglich. Dies fiel vor allem bei der Formulierung der in AIML zu befüllenden Templates auf, welche sich nach der weiteren Bearbeitung der bereits kategorisierten Datensammlung ergaben.

BERUF

BERUFALLGEMEIN:

BERUF(eröffnen[5],bestehen[5],besteht[5],stehen[5],steht[5],qualifikation[5],qualifikationen[5],qualifiziert[5],studiengang[5],existieren[5],sehen[5],möglichkeit[5],möglichkeiten[5],mögliche[5],mögliche r[5],offen[5],anstreben[5],gibt[5],geben[5],gibts[5])

BERUF

Abbildung 25: Dictionary 2 | am Beispiel Beruf

Nach dem Prozess der weiteren Reduzierung der Frage und die daraus resultierende Erstellung eines Endtemplates stellte sich die inhaltstragende Funktion des Verbes immer mehr heraus, sodass ein zweites Dictionary angelegt wurde. Dieses beinhaltet die erste Reduktion auf einen Hauptterm – erstes Dictionary – in Verbindung mit übrigge-

bliebenen inhaltstragenden Begriffen der zweiten Reduktion. Somit entstehen vorgefertigte Strukturen, welche die Anfrage des Nutzers abfangen, bearbeiten und dem richtigen Antwort-Template zuordnen.

7.4 Stoppwörter vs. inhaltstragende Wörter



Die Verarbeitung der Fragen durch die Dictionaries bzw. die Bildung Endtemplate erforderte die Kategorisierung der Fragen in Stoppwörter und inhaltstragende Wörter. Obwohl es verschiedene Stoppwort-Listen für die Deutsche Sprache gibt, entschied sich die Projektgruppe für die Erstellung einer eigenen Liste, welche auf das genutzte Vokabular der Datenerhebung zurückgreift.

Abbildung 26: Stoppwort | Vokabular allgemein

Durch und während der Kategorisierung konnte das Vokabular der Studenten und Studieninteressierten in Stoppwörter, Fragewörter und inhaltstragende Wörter im Kontext des Prototypen *Elise* zugeordnet werden.

Wie bereits angesprochen wurde während der Abarbeitung der verschiedenen Fragen und deren Fragestellung bzw. Formalisierungsvariante immer deutlicher, dass auf den Großteil der Fragewörter verzichtet werden kann. Inhaltstragende Fragewörter sind somit in der vorliegenden Arbeit und deren technische Umsetzung lediglich *wo*, *wer* und *wann*. Die restlichen Fragewörter – wie zum Beispiel *welche*, *was*, *wie*, *warum* etc. – wurden der Stoppwortliste hinzugefügt. Hingegen wurden die drei relevanten Fragewörter



8 Prototyp

Zur Umsetzung und Erprobung der durch die Datenerhebung und Kategorisierung gewonnenen Erkenntnisse wurde ein Prototyp entwickelt, der, aufbauend auf der vorhandenen und erweiterten Datenbasis, als einfacher Chatbot fungieren sollte. Dafür wurde ein servergestütztes System implementiert, das über eine Webschnittstelle die Nutzung des Bot ermöglicht. Dieser kann im prototypischen Umfang Fragen zum Studium der Medieninformatik und Informationswissenschaft beantworten. Die folgende Dokumentation stellt die grundsätzlichen Anforderungen und Systemstrukturen vor, erläutert die Implementierung der einzelnen Module und beschreibt die Nutzung und mögliche Erweiterung der entstandenen Software. Die aktuelle und hier beschriebene Version des Systems kann über die URL urtalking.ur.de abgerufen werden. Der Quellcode ist über ein GIT-Repository frei zugänglich: <https://github.com/alexanderbazo/urtalking>.

8.1 Anforderungen

Benutzerschnittstelle

Das prototypische System sollte als zentrale Benutzerschnittstelle über eine web-basierte Eingabemaske verfügen, über die die Kommunikation zwischen Nutzer und Chatbot durch die Eingabe natürlich-sprachiger Texte ermöglicht wird. Dabei stellt diese Implementierung lediglich eine mögliche Ausformung der Benutzerschnittstelle dar; die interne Dialogverarbeitung und das zugrundeliegende Chat-System wurden explizit als eigenständige Module konzipiert und umgesetzt. Diese Unabhängigkeit der Komponenten sollte eine spätere Ersetzung der Web-Schnittstelle durch andere Formen der Benutzereingabe - wie etwa externe Applikationen oder sprach-erkennende Systeme - erleichtern. Dazu wurde die Verbindung zwischen der serverseitigen Komponente und der Prototyp als PHP-gestützte Zwischenstelle realisiert, die über das Hypertext Transfer Protocol angesteuert werden kann. Das Resultat einer textuellen (Benutzer-)Anfrage

wird ebenfalls über dieses Protokoll zurückgeliefert. Diese Lösung ermöglicht eine einfache Übertragung der Funktionalität auf andere Anwendungen, da die zugrundeliegende Mechanik, das Senden von HTTP-Requests, von allen gängigen Programmiersprachen und Systemarchitekturen unterstützt wird. Die technischen Anforderungen an eine mögliche Schnittstelle für das hier vorgestellte System umfassen damit im Wesentlichen nur die Möglichkeit der Kommunikation über das HTTP sowie die Erfassung bzw. Umwandlung einer Benutzereingabe in eine textuell repräsentierte Anfrage.

Datenbasis

Die implementierte Software sollte die vorhandene Datenbasis der *Prolog*-Fakten nutzen können und diese bei der Beantwortung möglicher Nutzeranfragen berücksichtigen. Die relevanten Komponenten des Systems sollten dabei über die Möglichkeit verfügen, die entsprechenden Daten ohne vorherige Konvertierung zu nutzen. Das heißt, die vorhandenen bzw. erweiterten *Prolog*-Fakten sollten dabei als solche gespeichert und genutzt werden. Dazu wurde ein entsprechendes, *Prolog*-interpretierendes Modul als zusätzliche technische Anforderung festgelegt. Dieses sollte zur Laufzeit des Systems über entsprechende Anfragen auf die *Prolog*-formatierten Daten zugreifen können und die Ergebnisse dieser Anfragen für den Nutzer verständlich zurückgeben oder an eine weitere Systemkomponente übergeben.

Chatsystem

Das zu implementierende Chatsystem sollte zur Dialogführung eine Wissensbasis nutzen, die auf der *Artificial Intelligence Markup Language (AIML)* basiert. Die relevanten Systemkomponenten sollten daher über die Möglichkeit verfügen, auf entsprechende *AIML*-formatierte Strukturen zuzugreifen und Benutzeranfragen hinsichtlich der dort definierten Regeln zu verarbeiten.

Chatsystem – Administration

Die Inhalte der zugrundeliegenden *AIML*-Struktur sollten problemlos bearbeitet und erweitert werden können. Änderungen an dieser Struktur sollten dabei möglichst fehlerfrei und zeitnah in das Produktivsystem integriert werden können. Eine entsprechende administrative Oberfläche für die relevanten Systemkomponenten sollte dabei leicht zugänglich sein und alle relevanten Funktionalitäten der *AIML* unterstützen.

Chatsystem – Sprachverarbeitender Präprozessor

Die serverseitige Komponente des Systems sollte über die Möglichkeit verfügen, die textuelle Eingabe des Benutzers für die Verwendung durch das Chatsystem aufzubereiten, um die Erkennung und Interpretation zu erleichtern sowie die nötige Ausnahmebehandlung durch den *AIML*-Interpreter zu minimieren. Diese Komponente sollte dabei modular aufgebaut sein und die tatsächliche Funktionalität der Vorverarbeitung beliebig erweiterbar gestalten. Die Eingabe des Benutzers sollte von einer, in der Reihenfolge festlegbaren, Kette an Verarbeitungsmodulen umgeformt werden. Am Ende der Kette sollte der sprachverarbeitende Präprozessor das entstandene Ergebnis für die Verwendung durch das eigentliche Chatsystem weitergeben.

Unterstützung individueller Nutzer

Das Gesamtsystem sollte die gleichzeitige Nutzung durch mehrere Benutzer unterstützen. Dabei sollten Anfragen und Dialogverlauf der verschiedenen Nutzer eindeutig unterschieden und zugeordnet werden können. Insbesondere sollten die *AIML*-Funktionalitäten zur kontinuierlichen Dialogführung unterstützt werden, d.h. System und Benutzer sollten auf vorangegangene Elemente des Dialoges Bezug nehmen können und das System sollte diese zur Verarbeitung der Benutzeranfragen berücksichtigen.

Systemarchitektur und weitere Anforderungen

Zusammenfassend stellt sich die Systemarchitektur des Gesamtsystems dadurch dreiteilig dar. Die eigentliche Benutzerschnittstelle kommuniziert mit dem Chabot, der Zugriff auf das *Prolog*-formatierte Faktenwissen hat. Die Teilmodule werden dabei durch geeignete Brücken miteinander verknüpft. Zusätzlich verfügt der Chatbot über eine Komponente zur sprachlichen Vorverarbeitung der Benutzereingabe. Die administrative Sicht auf die verwendeten *AIML*-Strukturen kann unabhängig implementiert werden. Für den Prototypen wird sie parallel zur Benutzerschnittstelle als webbasierter Dienst realisiert.

Als zusätzliche Anforderungen sollten alle Systeme weitmöglichst auf etablierte Standards und Dienste zurückgreifen. Zentrale Komponenten der Module sollten dabei durch den Einsatz vorhandener Lösungen und Programme realisiert werden und funktionierende und bekannte Klassen-Bibliotheken verwenden. Abbildung 28 zeigt die skizzierte Systemarchitektur mit den wesentlichen Komponenten.

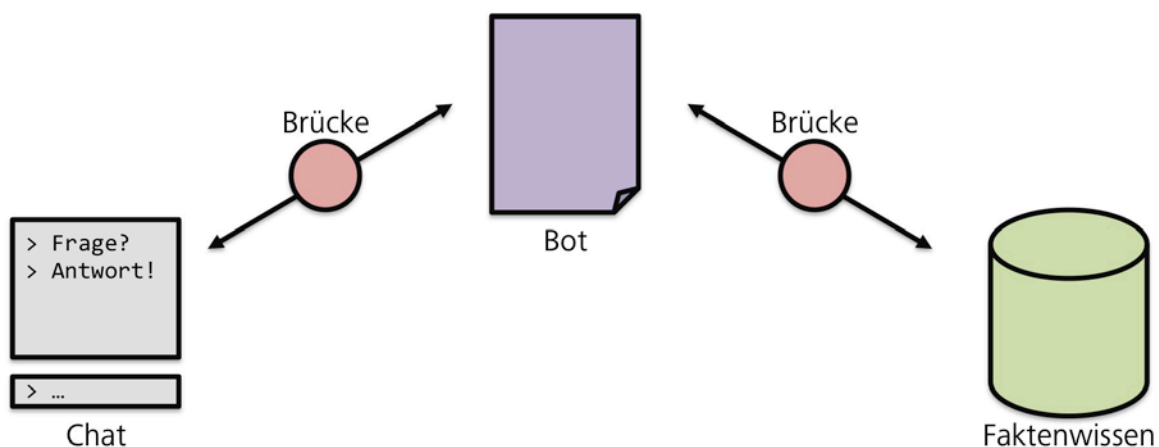


Abbildung 28: Prototyp | Skizzierte Systemarchitektur des Gesamtsystems.

Benutzerschnittstelle, Chatbot und Faktenbasis (Prolog) werden als einzelne und austauschbare Module realisiert, die über entsprechende Brücken miteinander kommunizieren.

8.2 Implementierung

Webschnittstelle

Die Webschnittstelle wurde als einfache HTML-Seite implementiert und umgesetzt. Die dynamischen Komponenten des Chatsystems und die Anbindung an die serverseitigen Mechanismen erfolgten durch JavaScript und AJAX (Asynchronous JavaScript and XML).

Das Benutzerinterface besteht aus einem DIV-Container, der den vergangenen Dialogverlauf - chronologisch geordnet - anzeigt. Die Beiträge des Benutzers und des Botsystems werden dabei visuell unterschiedlich dargestellt. Eine Eingabemaske erlaubt die Nutzung durch die Besucher der Website. Zur Verarbeitung der Benutzereingaben wurde ein clientseitiges JavaScript-Modul (bundle.js) über die JS-Implementierung des jeweiligen Browsers mit dem Chatfenster verknüpft. Das Skript lauscht auf Veränderungen in der Eingabemaske und verarbeitet diese, sobald der Nutzer die Eingabetaste betätigt. Clientseitig wird die Eingabe des Nutzers in den Dialogverlauf eingetragen. Zusätzlich wird über einen AJAX-Aufruf die serverseitige PHP-Brücke (api.php) angesprochen. Die Rückgabe dieser Anfrage wird anschließend wieder durch das clientseitige JS-Modul verarbeitet. Dabei wird die Antwort des Bots in der Benutzerschnittstelle angezeigt. An dieser Stelle werden abschließend Formatierungsangaben für mögliche Serverantworten gesetzt. Unter anderem werden hier, falls der Server mit einer Liste an Kursen oder Modulen antwortet, Hyperlinks zur Anzeige zusätzlicher Metainformation in der Darstellung des Dialogverlaufs ergänzt.

Administration

Zur erleichterten Administration der *AIML*-Daten wurde ein einfaches Webinterface implementiert, das die XML-Struktur der verwendeten *AIML*-Datei übersichtlich in Menüstrukturen überführt und editierbar macht. Die *AIML*-Dateien können dabei ohne

Veränderung des Schemas genutzt werden. Die einzige Ergänzung zum definierten *AIML*-Syntax ist ein zusätzliches Attribut `topic` in jedem `category`-Knoten:

```
<category topic="Smalltalk">
```

Diese Ergänzung dient zur Gliederung der *AIML*-Datei selbst sowie zur Verbesserung der Lesbarkeit der Menüstrukturen in der Administrationsansicht. Grundsätzlich erlaubt bereits die *AIML*-Syntax die Nutzung eines entsprechenden *XML*-Knotens zur Gruppierung zusammenhängender Kategorien. Allerdings wird dieser exklusiv interpretiert: Gruppierte Kategorien werden nur dann vom Parser berücksichtigt, wenn im Vorfeld ein entsprechendes Konversationsthema identifiziert worden ist. Die hier gewählte Vorgehensweise erlaubt die Strukturierung der *AIML*-Datei, ohne die technische Verarbeitung zu komplizieren und die Dialoggestaltung unnötig komplexer zu machen. Ausgehend von den ergänzten Attributen, stellt die Administrationsansicht ein Menü mit allen existierenden Themen (`topic`) dar. Der Benutzer kann einzelne Themen auswählen und die jeweiligen Kategorien, die diesem Thema zugeordnet worden sind, über ein weiteres Listenmenü editieren. Dabei können alle Komponenten der *AIML*-Kategorie (*Pattern*, *that-Referenz* und *Templates*) frei verändert bzw. ergänzt werden. Ebenso ist es dem Nutzer möglich, neue Themen zu erstellen und in bestehenden Themen weitere Kategorien zu ergänzen.

Die Administrationsschnittstelle kommuniziert über *AJAX* mit der serverseitigen *API*. Dabei übergibt der Server die aktuelle *AIML*-Datei als *XML*-String an die Schnittstelle. Dieser String wird lokal von entsprechenden *JavaScript*-Bibliotheken geparkt. Aus der interpretierten Struktur wird anschließend die Menüstruktur aufgebaut. Veränderungen am Datenbestand werden zuerst lokal im Browser zwischengespeichert. Zur Übernahme der Änderungen auf dem Server benötigt der Nutzer ein Passwort. Über einen er-

neuten *AJAX*-Aufruf wird die veränderte Fassung der *AIML*-Struktur zusammen mit dem Passwort übertragen. Die veränderte Datenbasis wird dabei als *JSON*-formatierter Text versendet und auf dem Server in eine entsprechende *XML*-Darstellung überführt. Sobald die Anfrage verifiziert worden ist, wird die neue Version auf dem Server gespeichert. Anschließend wird diese Version automatisch dem Chatsystem bereit gestellt. Zur Vermeidung von Datenverlust wird vor jeder Speicheroperation eine Sicherheitskopie der aktuellen *AIML*-Datei angelegt.

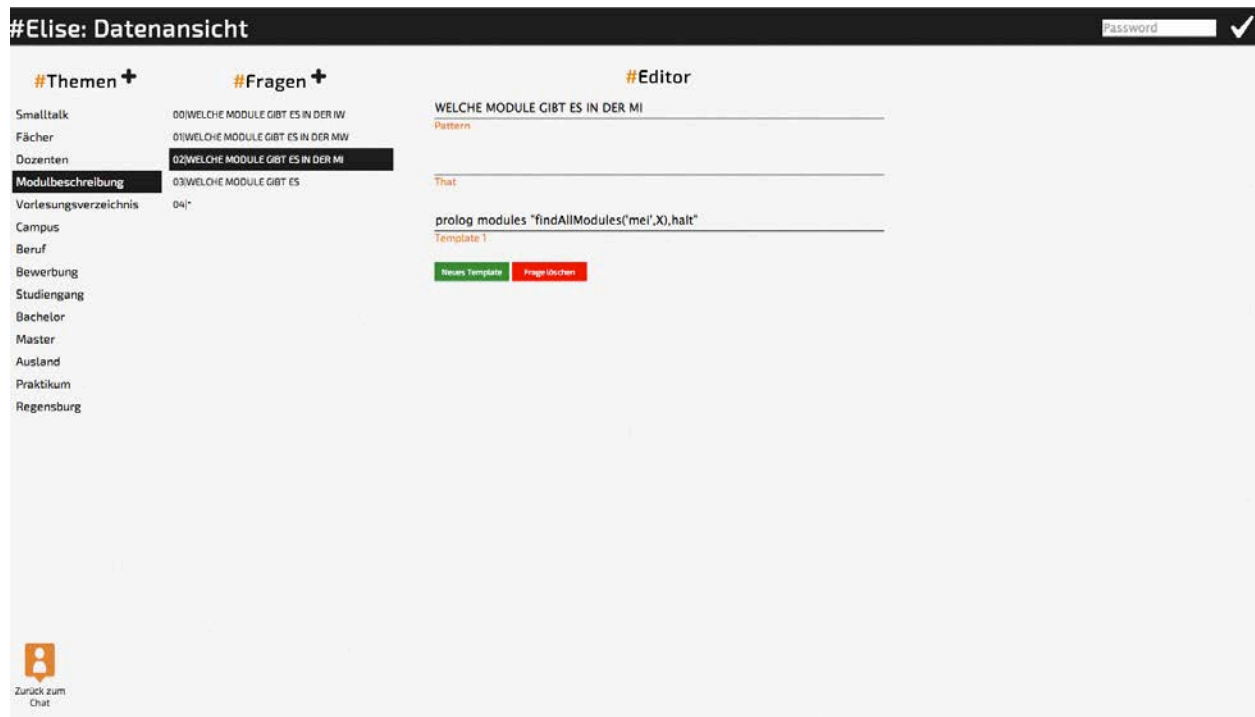


Abbildung 29: Prototyp | Screenshot der Administrationsschnittstelle.

PHP-Brücke

Zentraler Bestandteil und Einstiegspunkt für die Verarbeitung der Benutzeranfragen ist die serverseitige *PHP*-Brücke. Dieses Skript kann von der Benutzerschnittstelle über *AJAX* angesprochen werden. Dazu werden beim Aufruf der zentralen Datei `api.php` übergebene *POST*-Werte ausgelesen. Diese Parameter sind Bestandteil des *AJAX*-Aufrufs und können dazu benutzt werden, um die Anfrage an die Brücke zu spezifizieren. Je nach übergebenen *POST*-Werten werden unterschiedliche Funktionen ausge-

führt. Aktuell besteht die Möglichkeit, eine Anfrage an den *AIML*-Interpreter zu stellen (`$_POST['request'] = 'aiml'`), die aktuelle *AIML*-Struktur als *XML*-Dokument anzufordern (`xml`), sowie eine veränderte Version der Struktur auf den Server zu laden (`update`). Je nach verwendeter Funktion ist die Übergabe weiterer *POST*-Werte nötig. Wichtig ist dabei vor allem `$_POST['query']` zur Spezifizierung der Anfrage. Die *PHP*-Brücke steuert die Anfragen und ruft, in Abhängigkeit von der Anfrage, unterschiedliche Subsysteme auf. Diese sind zur besseren Gliederung des Programmcodes auf unterschiedliche *PHP*-Dateien aufgeteilt:

aiml.php | Diese *PHP*-Datei wird zur Steuerung der *AIML-Schnittstelle* verwendet.

prolog.php | Hier wird der *Prolog-Interpreter* angesprochen.

xml.php | Diese Datei ermöglicht sowohl das Herunterladen der aktuellen *AIML*-Datei, sowie auch das Heraufladen einer aktualisierten Fassung.

Die *PHP*-Brücke überprüft nach der Aktivierung eines solchen Subsystems dessen Rückgabe. Das Resultat wird entweder direkt an den Nutzer weitergegeben oder durch ein anderes Subsystem weiterverarbeitet. Wichtig ist dabei vor allem die Zusammenführung der *AIML*- und *Prolog*-Komponenten. Durch die Interpretation der *AIML*-Rückgabe kann die Brücke entscheiden, ob diese direkt an den Benutzer weitergeben werden kann (im Falle einer natürlich-sprachigen Antwort) oder erst durch das *Prolog*-System weiterverarbeitet werden muss.

Python-Präprozessor

Eine wichtige Komponente des implementierten Prototypen stellt der Python-Präprozessor dar. Dieses System ist Bestandteil der Python-basierten *AIML*-Schnittstelle und umfasst mehrere Module, die Benutzeranfragen vor der Weiterverarbeitung durch

den *AIML*-Interpreter aufbereiten. Ziel ist es dabei, die textuelle Anfrage des Nutzers durch verschiedene Verfahren so aufzubereiten, dass sie leichter vom *AIML*-Prozessor interpretiert werden kann – damit soll die Möglichkeit geschaffen werden, die eigentliche *AIML*-Struktur so einfach wie möglich gestalten zu können. Die auftretenden sprachlichen Variationen sollen nicht durch parallele Phrasenabbildung innerhalb des *AIML*-Dokuments abgefangen werden. Vielmehr soll – durch den Einsatz des Präprozessors – die Anfrage des Nutzers sprachlich reduziert und vereinfacht und damit der Darstellung innerhalb der *AIML*-Datei angepasst werden. Dazu durchläuft die Query eine Reihe von Modulen, die jeweils einen Eingabestring auf unterschiedliche Art und Weise verarbeiten und das Ergebnis dieser Transformation an das nächste Modul weitergeben. Zur Realisierung dieser Idee wurden zwei wesentliche Komponenten implementiert:

Rahmenprogramm | Das Rahmenprogramm des Präprozessors steuert die Verarbeitung. Zu Beginn werden die unterschiedlichen *Aimlizer*-Module geladen und deren Verarbeitungsreihenfolge festgelegt. Anschließend wird der Eingabe-String in die Modulkette injiziert. Nach dem Durchlauf durch die Kette ruft das Rahmenprogramm den *AIML*-Interpreter auf und übergibt, als Anfrage, das Resultat der Modulkette.

Aimlizer-Module | Der Präprozessor unterstützt eine beliebige Anzahl von Modulen, die die Eingabe des Benutzers nacheinander verarbeiten. Diese Module basieren auf der Klasse `AimlizerModule`:

```
class AimlizerModule:
    def process(self, string)
    return string
```

Jedes Modul muss die Funktion **process** implementieren bzw. erben, die innerhalb der Verarbeitungskette aufgerufen wird und den zu verarbeitenden *String* an das Modul übergibt. In dieser Funktion werden die spezifischen Verarbeitungsschritte des jeweiligen Moduls implementiert. Zusätzlich kann im Konstruktor der Klasse eine individuelle Anpassung durch weitere Parameter erfolgen. Einige Module benötigen etwa Wortlisten oder Wörterbücher.

Der aktuelle Prototyp beinhaltet vier verschiedene *Aimlizer*-Module.

Rechtschreibkorrektur | Der implementierte *Spell-Checker* nutzt einen Algorithmus von Perte Norvig (<http://norvig.com/spell-correct.html>) zur probabilistischen Korrektur eines Eingabe-*Strings*. Dazu werden die übergebenen Wörter mit eine Liste vorhandener, korrekter Wörter verglichen. Anhand von Ähnlichkeitswerten werden ungrammatikalische, weil nicht in der Wortliste vorhandene, Wörter korrigiert. Dazu ist eine umfassende Liste von Wörtern der jeweils verwendeten Sprache nötig. Der aktuelle Prototyp arbeitet mit einer Liste der 10000 häufigsten Wörter der deutschen Sprache, die dem Wortschatz Leipzig Projekt (<http://wortschatz.uni-leipzig.de/>) entnommen worden ist.

Stoppwortreduktion | Anhand einer beliebigen Liste von Stoppwörtern kann dieses Modul die Anfrage weiter reduzieren. Dabei wird jedes Wort auf sein Vorhandensein in der Liste überprüft und gegebenenfalls aus der Anfrage entfernt.

Normalisierungs-Modul | Dieses Modul entfernt Satzzeichen aus dem übergebenen *String*. Zusätzlich werden überflüssige Leerzeichen entfernt.

Ersetzungsmodul | Das Ersetzungsmodul arbeitet mit einer Wörterbuchliste, um synonyme oder gleichwertige Wörter auf einen gemeinsamen Begriff zu reduzieren.

Dadurch kann die Anzahl der abzubildenden Phrasen in der *AIML*-Datei maßgeblich reduziert werden, da keine sprachlichen Variationen berücksichtigt werden müssen. Die verwendete Wörterbuchliste muss dabei zeilenweise nach einer festgelegten Syntax aufgebaut sein:

HALLO:hi,hey,guten tag

Alle Wörter der Liste auf der rechten Seite des Doppelpunktes werden auf das vorangehende, linke Wort reduziert. Auf der rechten Seite sind dabei, unter Berücksichtigung des Kommas als Trennzeichen, auch Leerzeichen-getrennte Mehrwortgruppen möglich. Zusätzlich besteht die Möglichkeit, eine minimale Nähe zu anderen Wörtern anzugeben:

LP:punkte(bekommt[2];bekomme[2])

Das Wort „Punkte“ wird nur dann zu LP reduziert, wenn es innerhalb der Anfrage nicht weiter als 2 Wörter von „bekommt“ oder „bekomme“ entfernt ist. Durch Lemmatisierung oder eine geeignete Stammformreduktion könnte dieses System verbessert werden; momentan müssen alle gewünschten Variationen der relevanten Nachbarn angegeben werden.

Durch die Festlegung einer gemeinsamen Elternklasse für alle Module können beliebige weitere *Aimlizer*-Module ergänzt werden, ohne Änderungen am Rahmenprogramm nötig zu machen.

Nach Abschluss der Vorverarbeitung generiert das Rahmenprogramm mögliche Kombinationen aus dem reduzierten Anfragestring. Diese Permutationen aus den Term- Bestandteilen (Wörtern) werden sukzessive an den *AIML*-Interpreter weitergegeben, um eine positive Rückgabe zu bekommen. Der Hintergrund für dieses Verfahren begründet sich in der Feststellung, dass durch die Reduzierung der Anfrage Terme entstehen können, die aus den gleichen Wörtern - in unterschiedlicher Reihenfolge - bestehen. Um in der *AIML*-Datei nur eine dieser Kombinationen abbilden zu müssen und trotzdem die unterschiedlichen Varianten abfangen zu können, werden zusätzlich Permutationen der Anfrage verwendet.

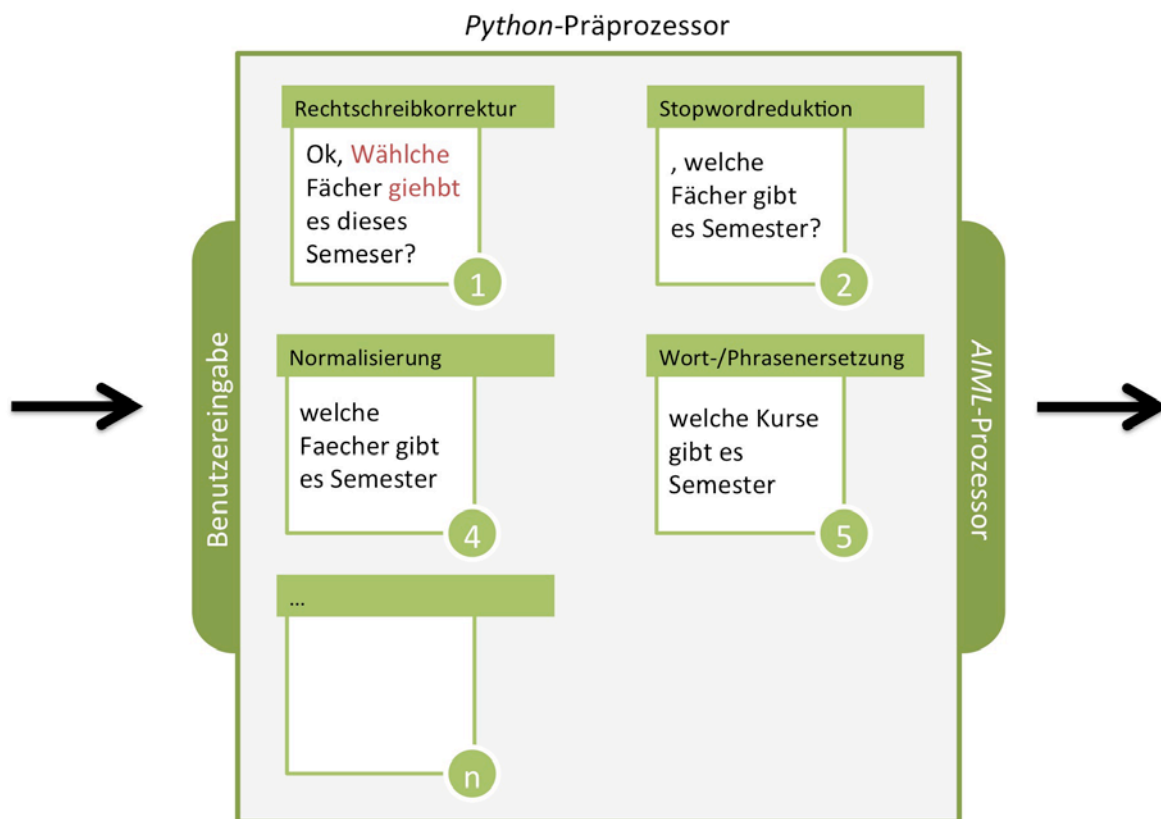


Abbildung 30: Prototyp | Schematischer Ablauf des Präprozessors.

Die Benutzereingabe wird von einer beliebigen Anzahl an Modulen verarbeitet, um dann an den AIML-Prozessor übergeben zu werden. Die Reihenfolge der Module kann beliebig festgelegt werden.

AIML-Schnittstelle

Zur Verarbeitung der AIML-Anfrage wird *PyAIML* (<http://pyaiml.sourceforge.net/>) genutzt. Dieser *AIML*-Interpreter für *Python* erlaubt es, *AIML*-Daten einzulesen. Auf der Basis der eingelesenen Dateien können Anfragen beantwortet werden. Der Interpreter fungiert dabei als Chatbot. Die Verwendung innerhalb eines eigenen *Python*-Programmes erfolgt über das Laden des `aiml`-Pakets. Anschließend kann ein `kernel`-Objekt erzeugt werden, das für das Laden und Nutzen vorhandenen *AIML*-Quellen verwendet wird. Im implementierten Prototypen wird zuerst eine spezifizierte *AIML*-Datei geladen. Das `kernel`-Objekt erlaubt es, individuelle Dialogverläufe zwischen dem Benutzer und dem Interpreter zu speichern. Die relevanten Informationen werden dabei in separaten Dateien gespeichert und können jederzeit wieder geladen werden. Hier geschieht die Unterscheidung der Nutzer über eine entsprechende Variable, die der aufrufenden *Python*-Anwendung übergeben wird. Anhand dieser Variable wird in der *AIML*-Schnittstelle die relevante Session-Datei geladen. Nach diesem Initialisierungsvorgang wird die Benutzeranfrage, die beim Aufruf der *AIML*-Schnittstelle übergeben worden ist, an den Interpreter weitergegeben. Die Rückgabe wird anschließend auf der Standardausgabe des Systems ausgegeben. Dadurch kann sie von Programmen, die diese Schnittstelle nutzen, abgefangen werden.

Prolog-Interpreter

Prolog wird über die Nutzung von *SWI-Prolog* (<http://www.swi-prolog.org/>) integriert. Das entsprechende Programmpaket wurde auf dem verwendeten Server installiert und kann über den Befehl `swipl` von beliebigen anderen Anwendungen oder über eine Eingabeaufforderung angesprochen werden. Im implementierten Prototyp wird die Nutzung der *Prolog*-Datenbasis über *PHP* ermöglicht. Das *PHP*-Skript der Brücke ruft, wenn nötig, den *Prolog*-Interpreter auf. Dazu wird die Funktion `shell_exec` verwendet, die

es *PHP*-Skripten erlaubt, auf serverseitige Anwendungen zuzugreifen. Die nötige *Query* für den *Prolog*-Interpreter wird von der *AIML*-Schnittstelle geliefert. Neben natürlich-sprachigen Antworten enthält die verwendete *AIML*-Datei auch *Templates* mit *Prolog*-Anfragen, die über ein entsprechendes Präfix von „normalen“ Antworten unterschieden werden können. Wird eine solche Anfrage an die *PHP*-Brücke zurückgegeben, kann sie durch das Präfix identifiziert und an *Prolog* weitergeben werden. `swipl` erwartet eine Datei mit *Prolog*-Fakten sowie eine entsprechende *Query* als Parameter. Die Rückgabe erfolgt dabei auf der Standardausgabe des Systems und wird im Falle des Prototypen in eine *PHP*-Variable umgeleitet. Diese Variable erhält daher nach dem Aufruf das Resultat der *Prolog*-Anfrage und kann weiter verwendet werden.

8.3 Verwendung

AIML

Die *AIML*-Schnittstelle wird über ein zentrales Python-Skript angesprochen, das sowohl den Präprozessor als auch das eigentliche Interface zur *AIML*-Datei beinhaltet. Das Skript wird über verschiedene Optionsparameter gesteuert und kann über eine Eingabeaufforderung oder durch den Aufruf innerhalb eines anderen Programmes genutzt werden. Die Ergebnissrückgabe erfolgt dabei über die Standardausgabe des Systems. Die Optionen unterteilen sich in zwei Gruppen, die zum einen die *AIML*-Schnittstelle betreffen und zum anderen nötige Parameter für die Vorverarbeitung der Benutzeranfrage definieren. Ein möglicher Aufruf des Skriptes stellt sich wie folgt dar:

```
ask.py -q QUERY -a FILE -s SESSION -d DICTIONARY -c SPELLCHECKER -l STOPWORDS -p PERMUTATIONS
```

QUERY | Über die Option `q` wird dabei die Benutzeranfrage übermittelt. Besteht dieser *String* dabei aus mehreren Wörtern, muss er durch Anführungszeichen eingeschlossen

werden. Zusätzliche Anführungszeichen innerhalb der Query müssen ausgezeichnet werden (\").

AIML | Die zu verwendende *AIML*-Datei wird über die Option **a** definiert. Hierbei kann entweder der absolute Pfad oder der relative, ausgehend vom Ort des Skriptaufrufs, angegeben werden.

SESSION | Über die Option **s** wird ein beliebiger Identifikator übermittelt, der den Benutzer eindeutig definiert. Diese ID dient der Unterscheidung und benutzerspezifischen Verarbeitung der Anfrage und sollte zur Laufzeit für gleiche Nutzer durchgehend gleichbleibend verwendet werden.

DICTIONARY | Über die Option **d** kann der Pfad (relativ oder absolut) zu einem Wörterbuchverzeichnis angegeben werden. Alle darin enthaltenen Dateien werden für das Ersetzungsmodul des Präprozessors verwendet und müssen in dem definierten Format strukturiert sein.

SPELLCHECKER | Über die Option **c** wird eine Textdatei zur Rechtschreibkorrektur angegeben (absolut oder relativ). Diese Datei enthält eine Liste an Wörtern, die zur probabilistischen Rechtschreibkorrektur der Benutzereingabe verwendet werden. Diese Wörter müssen dabei zeilenweise (ein Wort pro Zeile) in die Datei eingetragen werden.

STOPWORDS | Über die Option **l** kann der Pfad (absolut oder relativ) zu einer Textdatei angegeben werden, die eine Liste (zeilenweise) mit Stoppwörtern enthält. Diese werden aus der jeweiligen Benutzeranfrage extrahiert und nicht für die weitere Verarbeitung durch den *AIML*-Interpreter genutzt.

PERMUTATIONS | Über die Option **p** kann die maximale Anzahl an Query-Permutationen angegeben werden, die nach der Vorverarbeitung des Eingabestrings angewendet wird. Die Benutzeranfrage wird durch die vorgeschalteten Module weitmöglichst reduziert, um den inhalts-relevanten Pattern in der *AIML*-Datei zu entsprechen. Dabei werden jeweils nur eine mögliche Kombination aus Endtermen (komplett reduzierten Query-Bestandteilen) abgebildet. Da Benutzeranfragen unter Umständen auf verschiedene Kombinationen von gleichen Termen reduziert werden können, werden auch die Permutationen der reduzierten Query getestet. Der PERMUTATION-Wert gibt dabei an, wie lang ein reduzierter Term maximal sein darf, damit seine Permutationen für die *AIML*-Anfrage berücksichtigt werden. Diese Option dient der Reduzierung der Bearbeitungsdauer, da bei unvollständig reduzierten Termen mit vielen Wörtern die Anzahl der Kombinationen exponentiell steigt.

Prolog

Für die Nutzung der *Prolog*-Datenbasis wird der vorhandene *SWI*-Interpreter (<http://www.swi-prolog.org/>) verwendet, der für alle gängigen Systemarchitekturen verfügbar ist. In dem implementierten System wird serverseitig die *Debian/Linux*-Version verwendet, die über die Paketverwaltung bereitgestellt wird. Der Interpreter wird dabei über die Eingabeaufforderung angesprochen. Der dazu nötige Befehl

```
swipl -f ../prolog/facts/mi_iw.pl -g QUERY
```

kann über die Umgebungsvariable des Systems erreicht werden und befindet sich auf dem verwendeten Server unter `/opt/local/bin/swipl`. Von den möglichen Funktionen und Argumenten des Interpreters werden für den hier implementierten Einsatz nur die Optionen für die Parametrisierung der *Prolog*-Quelle (**f**) sowie der Benutzeranfrage (**g**) genutzt. Die Rückgabe erfolgt über die jeweilige Standardausgabe des Systems.

8.4 Testergebnisse

Der aktuelle Stand des Prototypen wurde mit dem Fragenkatalog getestet, der durch die Erstsemesterbefragung erhoben worden ist. Dazu wurden die transkribierten Fragen in einer einzelnen Datei gebündelt. Pro Zeile enthält diese Testdatei eine Frage. Das Python-Skript wurde angepasst und kann über eine zusätzliche Option `t` mit einer Referenz auf diese Datei gestartet werden. Die Datei wird dabei zeilenweise eingelesen und jede Frage an den *AIML*-Interpreter weitergegeben. Die Verarbeitung erfolgt dabei genauso wie im eigentlichen Programm: Die Anfragen werden durch die verschiedenen Module vorverarbeitet und schließlich durch *AIML* interpretiert. Vor jeder Anfrage wird der Interpreter zurückgesetzt, um eine mögliche Beeinflussung durch den Dialogverlauf (vorherige Frage) auszuschließen.

Insgesamt wurden 508 Fragen erfasst und für den Test verwendet. 385 (75%) der Fragen konnten erfolgreich reduziert werden. 214 (42%) Fragen wurden durch die Vorverarbeitung in ein einzelnes End-Template umgeformt, das von dem *AIML*-Interpreter verarbeitet werden kann. 100 dieser End-Templates (46%) wurden durch das Bot-System korrekt beantwortet.

Zwei Problemfelder führten zu der unzureichenden Abdeckung der Eingaben. Zum einen wurden nicht alle festgestellten Inhaltskategorien mit Antworten verknüpft. Dieses ist auf die großen Masse an vorhandenen Daten zurückzuführen, die für die prototypische Implementierung nicht komplett verarbeitet werden konnte. Eine entsprechende inhaltliche Ergänzung der *AIML*-Grundlage durch Hinzufügen weitere End-Templates und entsprechender Antworten ist aber technisch und praktisch problemlos möglich. Kritisch ist hierbei der immense zeitliche Aufwand, der für die Recherche der Antworten nötig ist.

Ein weiteres Problem betrifft die technische Umsetzung des Ersetzungswörterbuchs. Die verwendete Syntax kann maschinell leicht eingelesen werden. Die Editierung durch die Autoren ist auf Grund der strikten Syntax-Regeln jedoch schwierig. Ohne zusätzliche Software-Unterstützung, die entsprechende Fehler direkt bei der Eingabe deutlich macht, ist eine Wartung der Datenbestände nur schwer handhabbar. Die beiden verwendeten Wörterbuchdateien verfügen zusammen über ca. 300 Einträge. Jedem Eintrag sind mehrere, in vielen Fällen über 30, Terme zu geordnet. Jeder Term wird bei der Verarbeitung durch den entsprechenden Eintrag ersetzt. Dabei wird ebenfalls der Kontext berücksichtigt, der in der Datei zusammen mit dem Term aufgeführt wird. Diese Dateien wurden manuell in einer Textdatei angelegt. Die Masse an Daten machte diesen Prozess unübersichtlich und führte zu verschiedenen Syntax-Fehlern. Die eingesetzte Software fängt entsprechende Fehler beim Parsing ab, ignoriert aber die betreffenden Zeilen. Für die weitere Nutzung ist daher zum einen eine nachgiebigere Parser-Implementierung sowie eine entsprechende Tool-Unterstützung der Autoren notwendig.

9 Fazit und mögliche Erweiterungen

Der finale Stand des implementierten Prototypen gibt einen Ausblick auf die Möglichkeiten, die sich durch die intensive Erfassung und Aufarbeitung der erhobenen Datenbasis, ergeben. Die gesammelten Anfragen der möglichen Nutzer wurden analysiert, inhaltlich geordnet und aufbereitet. Dadurch entstand eine umfangreiche Basis für die Verwendung im Prototypen. Durch diese manuelle Vorverarbeitung konnte ein technisches Konzept zur Erkennung und Beantwortung von ähnlichen Fragemustern vorgeschlagen und konzipiert werden. Entsprechende Tests zeigen, dass sich die Datenanalyse gewinnbringend anwenden lässt. Die vorgeschlagene Systemarchitektur kann für die konkrete Problemstellung adaptiert und eingesetzt werden. Dabei ist die tatsächliche Implementierung jedoch prototypisch und muss entsprechend diskutiert und kritisiert werden. Abschließend seien daher eine Auswahl an möglichen Verbesserungsvorschlägen aufgeführt, die bei einer Weiterentwicklung des Systems berücksichtigt werden müssen.

Systeminterner Datenaustausch

Die Kommunikation zwischen den implementierten Modulen erfolgt grundsätzlich über die Weiter- bzw. Rückgabe einzelner *Strings*. Trotz der Parametrisierungsmöglichkeiten der verwendeten Einzelkomponenten ist eine Verbesserung dieser Übergabeart denkbar. Zusätzliche Parameter, die die Benutzereingabe bzw. Systemrückgabe genauer beschreiben, könnten bei der Fehlerkorrektur oder Ergebnispräsentation durch die Benutzerschnittstelle hilfreich sein. Dazu wäre es erforderlich, das aktuell nur die eigentliche Ein- bzw. Ausgabe umfassende Austauschformat von der monolithischen Textrepräsentation in ein komplexeres Format zu überführen. Denkbar wäre dabei, vor allem für die Rückgabekanäle, ein auf *JSON (JavaScript Object Notation)* basierendes System. Ein solches Format könnte neben dem verarbeiteten String zusätzliche Meta-Informationen wie

etwa aufgetretene Fehler während der (Vor-)Verarbeitung enthalten. Zusätzlich wären, durch ein einheitliches und protokolliertes Format die einfache Integration von komplexeren Rückgabewerten – etwa sortierte Listen von möglichen Antworten – möglich.

Sprachverarbeitende Module

Die beschriebenen Ansätze zur maschinellen Vorverarbeitung könnten im Zuge einer möglichen Überarbeitung des Prototypen erweitert und verbessert werden. Einige der vorgeschlagenen und implementierten Module müssen für den Einsatz im *Live*-System korrigiert werden. Die eingesetzte Methodik zur Rechtschreibkorrektur weist ein problematisches Verhalten bei der Verarbeitung unbekannter Wörter auf. Diese werden nicht ignoriert, sondern zu ähnlichen, aber kontextuell falschen Wörtern korrigiert. Zur Vermeidung dieses Verhaltens ist es notwendig, dass geeignet Referenzdokumente ergänzt werden. Zusätzlich ist eine Erweiterung der sprachverarbeitenden Module denkbar. Mögliche Ansätze wären dabei etwa Versuche zur inhaltlichen Erschließung der Benutzeranfragen. Dabei könnte etwa zwischen fachlichen Fragen und akuten Problem (Raumfindung, etc.) unterschieden werden und entsprechende, unterschiedliche Lösungsvorschläge präsentiert werden.

Fehlerkompensation durch Information-Retrieval-Ansätze

Die aktuelle Implementierung ermöglicht nur einen *Eins-zu-Eins*-Abgleich zwischen Nutzeranfrage und Ergebnis. Im Falle der Nichtabdeckung der Anfrage durch die *AIML*-Struktur wird ein entsprechender Fehler bzw. eine entsprechende Notiz an den Nutzer zurückgegeben. Dieses Verhalten spiegelt die grundsätzliche Verfahrensweise vorhandener *AIML*-Interpreter wider. Dabei wird der Versuch unternommen, eine passende, bekannte Fragestruktur zu ermitteln, die deckungsgleich mit der Nutzeranfrage ist. Wird eine solche gefunden, wird diese zur Ermittlung der passenden Antwort genutzt. Es werden keine Backtracking-Möglichkeiten zur Ermittlung weiterer passender

Lösungen berücksichtigt. Die *AIML*-Definition erlaubt nicht die Definition ambiger oder nicht-eindeutiger Frage-Antwortpaare. Um bei einem misslungenen Abgleich einer Nutzeranfrage trotzdem einen Antwortversuch zu unternehmen, könnte ein Abgleich der vorhandenen und bekannten Fragemuster – die in der *AIML*-Struktur definiert sind – und der eigentlichen Benutzeranfrage berücksichtigt werden. Dazu kann, durch Methoden und Techniken der maschinellen Sprachverarbeitung und des *Information-Retrievals*, ein Ähnlichkeitsvergleich durchgeführt werden. Voraussetzung dafür wäre eine entsprechende Abbildung der Antwortmuster und eine Integration eines solchen Abgleichmoduls in die bestehende Systemstruktur. Möglich wäre dies etwa durch die *PHP*-Schnittstelle zwischen Benutzer und *AIML-Bot*, die bei einer negativen Rückgabe den beschriebenen Ähnlichkeitsvergleich durchführt und die ermittelte, am besten passende Frage für eine erneute Anfrage an den Chatbot nutzt. Vorangehend müssten passende Schwellenwerte für die Ähnlichkeitswahrscheinlichkeit ermittelt und getestet werden, um einen sinnvollen und hilfreichen Dialogverlauf zu gewährleisten.

Identifikation von Kursnamen

Der aktuelle Prototyp arbeitet bei der Erschließung und Reduzierung der Anfragen durch Ersetzungs-Anweisungen. Eine inhaltliche Interpretation der syntaktischen Struktur geschieht noch nicht. Für viele der möglichen Anfragen ist ein solches Vorgehen jedoch ein vielversprechender Ansatz. Mit der Hilfe einer entsprechend aufbereiteten Kursliste könnten etwa Bezeichner oder Titel von Lehrveranstaltungen identifiziert werden. Diese könnten dann, zusammen mit dem Satz-Kontext zur Beantwortung der Benutzeranfrage verwendet werden. So wäre es dann möglich, eine Frage bezüglich eines konkreten Kurses mit einem direkten Bezug auf diesen zu beantworten und beispielsweise gezielt im Vorlesungsverzeichnis nach Information zu der Veranstaltung zu

suchen. Hinweise auf das Auftreten eines Kursnamen können aus dem syntaktischen Kontext erschlossen werden.

Ergänzung von Dialogstrukturen und nicht-inhaltlichen Äußerungen

Der aktuelle Prototyp wurde darauf optimiert, die inhaltlichen Fragen, die sich aus der Datenerhebung ergeben haben, zu beantworten. Für ein komplettes Chatbot-System ist es erforderlich, zusätzliche Möglichkeiten der Dialogführung zu ergänzen. Dieser Teil stand jedoch nicht im Fokus dieser Arbeit und wurde daher nicht berücksichtigt. Für die Ergänzung einer entsprechenden Komponente kann eine zweite *AIML*-Datei sowie ein mehrstufiges Anfrage-System konzipiert werden. Das vorliegende System besteht aus einer Vorverarbeitungskomponente sowie einer *AIML/PROLOG*-Datenbasis, die die Beantwortung konkreter, inhaltlicher Fragen erlaubt. Diesem Prototypen kann eine weitere Komponente zur Seite gestellt werden, die – ähnlich aufgebaut – eine weniger konkrete Dialogführung („Small-Talk“) erlaubt. Dabei kann die eingesetzte Software leicht so erweitert werden, dass beide Systeme parallel betrieben werden und die Anfrage des Benutzers durch beide Module verarbeitet wird. Kann eine der Komponenten keine Antwort generieren, kann die Rückgabe der anderen an den Benutzer weiter gegeben werden.

Ergänzung der Administrationssoftware

Als wesentliche Komponente wird in dieser Arbeit eine sprachverarbeitende Vorverarbeitung der Benutzeranfrage vorgeschlagen. Dieses modular aufgebaute System ist abhängig von einer Reihe von Datenquellen und Dokumenten. Die prototypische Implementierung hat gezeigt, dass die manuelle Pflege der entsprechenden Dateien und Listen mit wachsender Komplexität des Prototypen schwierig wird. Daher ist eine Ergänzung des Systems nötig, die den zukünftigen Autoren entsprechende Werkzeuge bietet, um diese Quellen langfristig zu pflegen.

10 Literatur

Bazo, Alexander; Spröd, Thomas (2013): *Plan in Logik: Ein Studienberater. Darstellung der Prüfungsordnung der Philosophischen Fakultäten der Universität Regensburg als Regelmenge für Prolog*, Regensburg.

Braun, Alexander; Glotz, Peter (2003): *Chatbots in der Kundenkommunikation*. Berlin [u.a.]: Springer (Xpert.press).

Crevier, Daniel (1994): *Eine schöne neue Welt? Die aufregende Geschichte der künstlichen Intelligenz*. Düsseldorf, Wien, Moskau, New York: ECON-Verl.

Laurel, Brenda; Mountford, S. Joy (1990): *The Art of human-computer interface design*. Reading, Mass: Addison-Wesley Pub. Co.

Pflamminger Melanie; Spies Oliver; Rank Martin (2014): *Analytische Ableitung von Konversationsstrukturen aus Chatbot-Mensch-Interaktionen*, Regensburg.

Rak, Jessica; Gross, Beate & Schlesinger, Michaela (2013): *Gewichtung und Priorisierung der abhängigen Lehrveranstaltungen eines Empfehlungssystems für Studierende*, Regensburg.

Rickel, Jeff; Johnson, W. Lewis: *Task-Oriented Collaboration with Embodied Agents in Virtual Worlds*, In: **Cassell, Justine** (2000): *Embodied conversational agents*. Cambridge, Mass: MIT Press.

Rzymkowski, Agathe; Salomon, Elisabeth; Rak, Jessica, Hamedinger, Beate & Schlesinger, Michaela (2011): *Dokumentation Gruppe Landmarken Outdoor – Forschungsseminar MMI: Aktuelle Trends in der Spracherkennung*, Regensburg.

Turkle, Sherry (1998): *Leben im Netz. Identität in Zeiten des Internet*. 1. Aufl. Reinbek bei Hamburg: Rowohlt.

Weizenbaum, Joseph (1977): *Die Macht der Computer und die Ohnmacht der Vernunft*. 1. Aufl. Frankfurt am Main: Suhrkamp.

Onlinequellen

DAAD (2013):

<https://www.daad.de/portrait/wer-wir-sind/kurzportrait/08940.de.html>

<https://www.daad.de/portrait/wer-wir-sind/leitbild/23115.de.html>

DAAD Jahresbericht von 2007 (2008):

<http://d-nb.info/993702252/34>

Christensen, Anne (2008): *Virtuelle Auskunft mit Mehrwert : Chatbots in Bibliotheken* / von Berlin : Institut für Bibliotheks- und Informationswissenschaft der Humboldt-Universität zu Berlin, 2008. – 62 S. - (Berliner Handreichungen zur Bibliotheks- und Informationswissenschaft; Heft 222)

<http://www.ib.hu-berlin.de/~kumlau/handreichungen/h222/h222.pdf>

Coporate Design der Universität Regensburg:

<http://www.uni-regensburg.de/corporate-design/>

FAZ (2007):

www.faz.net/s/Rub8D4D94F71B3A43159C78FB196DED6A94/Doc~E955FF21E67DF4ABE9553BE73201D4157~ATpl~Ecommon~Scontent.html

go-out Studieren weltweit:

<http://www.go-out.de/portrait/de/>

SUB Hamburg (2004):

<https://www.sub.uni-hamburg.de/bibliotheken/projekte/chatbot-stella.html>

SUB Hamburg (2004):

https://www.sub.uni-hamburg.de/fileadmin/redaktion/Bibliotheken/factsheet_stella.pdf

UB Köln:

https://www.ub.uni-koeln.de/res/albot/index_ger.html

https://www.ub.uni-koeln.de/res/albot/geschichte/index_ger.html

Links der vorgestellten Chatbots

Luzie:

<http://bot.daad.de/luziecgi-bin/luzie.cgi>

Stella:

<http://www.sub.uni-hamburg.de/bibliotheken/projekte/chatbot-stella.html>

Albot:

http://www.ub.uni-koeln.de/res/albot/index_ger.html

Links der erwähnten Chatbots

ASKacademicus:

<https://www.ub.uni-dortmund.de/chatterbot/>

INA:

<http://www.buecherhallen.de/ina/>

Anna:

http://www.ikea.com/ms/de_DE/campaigns/services/kontakt.html?icid=de|itl|fy14_service|teaser|kw3|237#Frag_Anna

Letzter Zugriff auf alle Onlinequellen: 09. April 2014

Anlagen: <http://urtalking.ur.de/documentation.php>