


**Exemplarische Untersuchung zur
Konzeption eines künstlich
anthropomorphen Informationsassistenten
am Beispiel Rendezfood**

Bachelorarbeit

im Studiengang
Wirtschaftsinformatik

vorgelegt von

Tom Lorenz Klein

Matr.-Nr.: 

am 07. September 2020
an der Universität Siegen

Erstprüfer/in:
Zweitprüfer/in:

Jun.-Prof. Dr. Thomas Ludwig
Prof. Dr. Volkmar Piepek

Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, insbesondere keine anderen als die angegebenen Informationen aus dem Internet. Diejenigen Paragraphen der für mich geltenden Prüfungsordnungen, die etwaige Betrugsversuche betreffen, habe ich zur Kenntnis genommen. Der Speicherung meiner Bachelorarbeit zum Zweck der Plagiatsprüfung stimme ich zu. Ich versichere, dass die elektronische Version mit der gedruckten Version inhaltlich übereinstimmt.

07.09.2020

(Datum)

(Unterschrift)

Kurzfassung

Chatbots liegen im Trend. In dieser Bachelorarbeit wird eine Konzeption und Implementation eines Chatbot-Prototyps erstellt. Dazu sollen Komponenten empfohlen werden, um künftige Projekte anzuleiten. Das Fundament dieser Forschung ist eine umfangreiche Literaturstudie, aus dessen Erkenntnissen die Konzeption erstellt wird. Im Anschluss wird auf Grundlage des Konzeptes ein Prototyp implementiert, der eine Vorlage für das Projekt Rendezfood darstellt. Zuletzt wird die Forschung durch Nutzerevaluation mit der Thinking-Aloud-Methode bestätigt. Zum Forschungsergebnis lässt sich zum einen sagen, dass keine Komponentenempfehlungen gegeben werden können und der Anwendungskontext in der Planung vermehrt beachtet werden sollte. Zum anderen ist aufgefallen, dass eine modulare Architektur, wie beispielsweise eine API verwendet werden sollte, um das System anpassbar und skalierbar zu halten. Außerdem ist eine datenbankbasierte Umsetzung zu empfehlen, um Nutzerdaten zu speichern, mit welchen der Chatbot anthropomorphisiert werden kann, sodass die Konversationsqualität ansteigt.

Abstract

Chatbots are trendy. In this bachelor thesis a conception and implementation of a chatbot prototype will be created. For this, components should be recommended in order to guide future projects. The foundation of this research is an extensive literature study, from the findings of which a concept was created. A prototype was then implemented on the basis of the concept, which represents a template for the Rendezfood project. Most recently, the research was confirmed by user evaluation using the Thinking Aloud method. Regarding the research result, it can be said, on the one hand, that no component recommendations can be given and that the application context should be considered more closely in planning. On the other hand, it was noticed that a modular architecture, such as an API, should be used to keep the system adaptable and scalable. In addition, a database-based implementation is recommended to save user data with which the chatbot can be anthropomorphized, so that the quality of the conversation increases.

Inhaltsverzeichnis

Erklärung	2
Kurzfassung.....	3
Abstract.....	3
Inhaltsverzeichnis	4
Abbildungsverzeichnis.....	6
Vorwort.....	7
1 Überblick.....	8
2 Ziele.....	10
2.1 Erstellung der Konzeption.....	10
2.2 Implementation des Prototyps	10
2.3 Evaluierung des Prototyps.....	10
2.4 Beantwortung der Forschungsfrage.....	11
3 Grundlagen	12
3.1 Definition: Künstliche Intelligenz	12
3.1.1 Definition: Intelligenz	12
3.1.2 Kontextbasierte Definition: Künstlich.....	12
3.1.3 Gesamtheitliche Definition.....	12
3.1.4 Abgrenzung	12
3.2 Definition: Chatbot.....	12
3.2.1 Definition: Multi-Agent-System (MAS).....	14
3.2.2 Definition: Anthropomorphismus bei Chatbots	14
3.3 Methodik.....	14
3.3.1 Definition: Thinking-Aloud.....	14
3.3.2 Nachteile von Thinking-Aloud.....	14
3.3.3 Vorteile von Thinking-Aloud in besonderen Zeiten	15
3.4 Einführung in genutzte Technologien	15
3.4.1 APIs („application programming interface“) und Webservices.....	15
3.4.2 Webhooks	17
3.4.3 Dialogflow (DF)	17
4 Literaturstudie.....	20
4.1 Bekannte Chatbots.....	20
4.2 Multi-Agent-Chatbot-Systeme	21
4.3 Anthropomorphe Chatbots	22
4.3.1 Vor- und Nachteile anthropomorpher Chatbots	22

4.3.2	Möglichkeiten zur sinnvollen Anthropomorphisierung von Chatbots	23
4.3.3	Empathie als Form der Vermenschlichung	23
5	Konzeption	24
5.1	Framework und Aufbau des Multi-Agent-Systems.....	24
5.1.1	Auswahl des Frameworks.....	24
5.2	Datenbank.....	25
5.2.1	SQL (Standard Query Language).....	25
5.2.2	NoSQL-Datenbanken (Not only SQL).....	26
5.2.3	Auswahl der Datenbank	26
5.2.4	Aufbau der Datenbank.....	26
5.3	Architektur.....	27
5.3.1	User Interface	27
5.3.2	Processing and Network Component (Dialogflow)	28
5.3.3	API.....	28
5.4	Verarbeitungsbeispiel der Anfrage „Zeige mir Gerichte“	29
6	Implementation und Limitationen.....	30
6.1	Implementation.....	30
6.2	Limitationen des Prototyps.....	30
7	Zwischenkolloquium und Evaluation nach der Thinking Aloud- Methode	31
7.1	Zwischenkolloquium am 22.06.2020 über Zoom.....	31
7.2	Einzelne Nutzerevaluation am 22.06.2020 über Zoom	32
7.3	Zusammenfassung der Nutzeraussagen nach Kategorien	33
7.3.1	Begrüßung	33
7.3.2	Verständnis	33
7.3.3	Bilder	34
7.3.4	Dialekte.....	34
7.3.5	Bedienung.....	34
7.3.6	Bedienung des User-Interfaces.....	34
7.3.7	Fehlerbehandlung	34
7.3.8	Weitere Anmerkungen.....	34
8	Zusammenfassung und Fazit.....	36
9	Ausblick.....	38
	Quellenverzeichnis	39

Abbildungsverzeichnis

Abbildung 1: Hierarchie der verschiedenen Dialogsysteme (Radziwill und Benton 2017)	13
Abbildung 2: Single-Agent gegen Multi-Agent-Lösungen (Calvaresi u. a. 2019)..	14
Abbildung 3: Phasen der Erstellung (Calvaresi u. a. 2019).....	22
Abbildung 4: ER-Diagramm der Chatbot Datenbank.....	26
Abbildung 5: Architektur Diagramm der Chatbot Konzeption	27
Abbildung 6: Screenshot vom User Interface.....	28
Abbildung 7: Screenshot der objektorientierten Klasse Meal und deren „read“- Funktion	29
Abbildung 8: Teilausschnitt einer Dialogflow Webhook Request an die API.....	29

Vorwort

Vor Ihnen liegt die Bachelorarbeit „Exemplarische Untersuchung zur Konzeption eines künstlich anthropomorphen Informationsassistenten am Beispiel Rendezfood“. Die Forschungsleistung besteht neben einer Technologie- und Literaturstudie in der Implementierung eines Chatbot-Prototyps, welcher dem Forschungsprojekt Rendezfood als Beispiel dienen soll.

Die Abschlussarbeit wurde im Rahmen meines Bachelorstudiums der Wirtschaftsinformatik an der Fakultät 3 der Universität Siegen und in Zusammenarbeit mit dem Forschungsprojekt Rendezfood am Lehrstuhl für Cyber-Physische Systeme verfasst. Ziel war es, die Aufgabe der Konzeption eines künstlich anthropomorphen Chatbot-Prototyps mithilfe von wissenschaftlicher Fachliteratur und bewährten Technologien zu lösen. Dazu beschäftigte ich mich innerhalb des Sommersemesters 2020 mit der relevanten Literatur und der Implementation des Prototyps.

Zusammen mit meinem Prüfer, Jun.- Prof. Dr. Thomas Ludwig und meinem Betreuer Philip Weber wurde die Fragestellung zu dieser Bachelorarbeit entwickelt. Trotz anfänglicher Schwierigkeiten konnte mir mein Prüfer Jun.- Prof. Dr. Thomas Ludwig durch seine Erfahrung, die Vorgehensweise des wissenschaftlichen Arbeitens doch noch nahelegen. Außerdem unterstützte mich Philip Weber immer wieder, in dem er meine verschwommenen Vorstellungen der Problemlösung fokussierte und stets in die richtige Richtung leitete. Sein zusätzlicher Input und die Hilfe bei methodischen Fragestellungen haben einen großen Teil zum erfolgreichen Verlauf dieser Bachelorarbeit beigetragen.

Daher möchte ich meinen Begleitern für das Einbringen ihrer Erfahrungen, die gute Anleitung und die Unterstützung danken. Zudem möchte ich auch allen Teilnehmern des Zwischenkolloquiums und der anschließenden Befragung danken, denn ohne deren Einsatz wäre eine Beendigung der Arbeit nicht möglich gewesen.

Ich wünsche viel Freude beim Lesen dieser Bachelorarbeit.

1 Überblick

Der Einsatz von virtuellen Informationsassistenten, ist ein anhaltender Trend, welcher sich gleichermaßen auf Unternehmen und die Allgemeinbevölkerung ausbreitet (Zamora 2017). Während Unternehmen meist auf speziell trainierte Assistenten setzen, welche bei der Annahme von Kundenanfragen oder einfachen Supportaufgaben unterstützen, so setzt, die Allgemeinbevölkerung auf ihre smarten Assistenten. Das sorgt dafür, dass pro Quartal mehr als 55 Millionen Stück der sogenannten Smart Speaker vom Typ Amazon Echo, Google Home oder Apple HomePod weltweit verkauft werden (Tenzer 2019). Neben den lokalen Assistenten zuhause, tragen 2019 bereits rund 58 Millionen deutsche Bürger einen intelligenten Informationsassistenten in Form ihres Smartphones mit sich, denn auch ohne spezielle Applikationen haben die führenden Mobilgerätbetriebssysteme einen Sprachassistenten vorinstalliert (Tenzer 2020).

Wie bereits beschrieben, gibt es virtuelle Assistenten in verschiedenen Arten und vor allem für verschiedene Anwendungsbereiche, in der folgenden Arbeit wird jedoch besonders auf die Kategorie der Chatbots eingegangen. Eine Abgrenzung von Chatbots zu anderen virtuellen Assistenten ist im Kapitel 3 erfolgt (Calvaresi u. a. 2019; Dalton, Ajayi, und Main 2018; Følstad und Skjuve 2019).

Chatbots werden bereits für diverse Nutzungsszenarien eingesetzt und können von Smalltalk bis zu informationsgeladenen Konversationen alles abbilden (Angelov und Lazarova 2019; Schrader und Haider 2009). Beispiele sind „Woebot“, ein Bot zur Depressionsprävention und Depressionssymptomerkennung von der Psychologin Alison Darcy, „DoNotPay“, ein juristischer Bot der Nutzern bei kleineren rechtlichen Fragestellungen helfen kann und „Hipmunk“ ein Reisechatbot, welcher die Buchung der Reise vereinfachen soll (Leah 2019).

Chatbots werden aus vielschichtigen Gründen eingesetzt, wobei für Unternehmen einer der Hauptgründe die Kostenoptimierung darstellt, denn Chatbots können mehrere Kunden gleichzeitig betreuen und sind zudem rund um die Uhr erreichbar (Mekni, Bani, und Sulieman 2020). Ein weiterer Grund könnte die schnelle Anbindung an Informationen sein, denn während Menschen Informationen manuell herausuchen müssen, haben Chatbots unmittelbaren Zugang zu Datenbanken und Informationsquellen, wie dem Internet (Roein 2019).

Die folgende Arbeit ist eine exemplarische Untersuchung zur Erstellung einer Chatbot-Architektur für das Forschungsprojekt Rendezfood. Rendezfood ist eine Applikation, welche Kunden bei der Auswahl von Lokalen und Malzeiten helfen kann. Dazu soll ein Chatbot erstellt werden, welcher auf der funktionalen Ebene die folgenden Kriterien erfüllen muss:

1. Das Empfehlen von Lokalen auf Grundlage, der vorangegangenen Konversation und voreingestellter individueller Nutzervorlieben, wie Vegetarismus oder Veganismus
2. Das Empfehlen von Speisen und Getränken auf Grundlage, der vorangegangenen Konversation und voreingestellter individueller Nutzervorlieben, wie Lebensmittelunverträglichkeiten oder subjektivem Ausschluss von Lebensmitteln, aufgrund von beispielsweise religiösen Ansichten.
3. Das Informieren des Kunden über Lokalitäten und Speisen, wie zum Beispiel Herkunft der Grundzutaten oder spezielle Informationen zum Gar- oder Zubereitungsprozesses.

4. Die Emotionalisierung der Bot-Nutzer-Konversation, beispielsweise durch Anpassung oder Nachahmung eines Dialektes, eines Akzentes oder der Wortwahl.

Um diese Vorgaben möglichst optimal abbilden zu können, soll auf Basis einer Literaturstudie ein Konzept erstellt und auf dessen Grundlage ein Chatbot-Prototyp implementiert werden. Dieser soll mit einer Kleingruppe nach der Thinking-Aloud-Methode getestet werden und die Ergebnisse der Evaluation entscheiden über den Erfolg des Konzeptes und der Implementierung. Der Prototyp soll im Anschluss dieser Arbeit dem Team des Projektes Rendezfood zur Verfügung gestellt werden und als beispielhafte Vorlage dienen.

Im nächsten Kapitel soll der Leser detailliert in die Ziele dieser Arbeit eingeführt werden. Um die nötige Durchsicht innerhalb der verwendeten Begriffe und Technologien zu erzeugen, werden wichtige Begriffe im dritten Kapitel, mithilfe von Fachliteratur definiert und abgegrenzt. Um die im fünften Kapitel vorgesehene Konzeption des Vorhabens auf der Grundlage von wissenschaftlichen Arbeiten und dessen Erkenntnissen zu erstellen, wird im vierten Kapitel eine Literaturstudie durchgeführt.

In Kapitel 6 wird auf Herausforderungen in der Implementation eingegangen. Im 7. Kapitel beschäftigt sich diese Arbeit mit dem stattgefundenen Zwischenkolloquium, sowie den Ergebnissen der Nutzerevaluation nach der Thinking-Aloud-Methode. Zudem wird über sinnvolle Änderungen am Prototyp diskutiert. In Kapitel 8 werden alle erlangten Kenntnisse und Erkenntnisse zusammengefasst und abschließend ein Fazit zum Konzept erstellt. Zusätzlich wird die Forschungsfrage beantwortet und auf Limitationen dieser Arbeit eingegangen. Im neunten Kapitel folgt der Forschungsausblick.

2 Ziele

Die Ziele dieser Arbeit lassen sich in drei Abschnitte kategorisieren, das erste wissenschaftliche Ziel ist die Erstellung einer Konzeption eines künstlich anthropomorphen Informationsassistenten, auf Basis von den gewonnenen Erkenntnissen aus einer Literaturstudie. Das zweite praxisorientierte Ziel ist die Implementation eines Chatbot-Prototyps, auf Grundlage der erstellten Konzeption. Das dritte Ziel ist die Prüfung der Konzeption durch evaluieren des Prototyps mithilfe der Thinking-Aloud-Methode und abschließender Beantwortung der Forschungsfrage.

2.1 Erstellung der Konzeption

Um einen funktionierenden Chatbot erstellen zu können, müssen zunächst die wissenschaftlichen Grundlagen erarbeitet werden, um danach die einsetzbaren Technologieeingrenzen und verstehen zu können. Zusätzlich sollte eine Literaturstudie, das auf den Grundlagen aufbauende Wissen schaffen, um die Entstehungswahrscheinlichkeit möglicher Konzeptionsfehler frühzeitig zu minimieren.

In der Konzeption sollen nach und nach die einzelnen Komponenten ausgewählt und deren Einsatzgebiet und Zweckmäßigkeit beschrieben werden. Hierbei liegt der Fokus nicht ausschließlich auf der Veranschaulichung der Funktionsweise, sondern vor allem auf der Sinnhaftigkeit im Anwendungskontext.

2.2 Implementation des Prototyps

Die Implementation des Prototyps stellt neben der Konzeption eines der Hauptziele dieser Arbeit dar. Dies ergibt sich zunächst aus der Aufgabenstellung und den Anforderungen an diese Arbeit seitens des Projektleiters als auch durch den Aufbau dieser Arbeit.

Die Aufgabenstellung dieser Bachelorarbeit war neben der wissenschaftlichen Ausarbeitung einer Konzeption, auch die Implementation eines Prototyps für das Forschungsprojekt Rendezfood. Dabei wurde besonderen Wert auf die folgenden Kriterien gelegt.

1. Der Prototyp muss netzwerkfähig sein, da er als Beispiel für einen Chatbot steht, welcher innerhalb einer mobilen App genutzt werden soll.
2. Der Prototyp sollte auf eine Datenbank gestützt sein, da sich die Informationsgrundlage dynamisch verändert und der Chatbot so auch auf ein wechselndes Angebot zurückgreifen kann.
3. Es sollte ein anthropomorpher Chatbot implementiert werden, welcher dynamisch auf bei der Nutzung entstehende Kontexte reagiert.

Die Implementation eines funktionierenden Prototyps ist aus zwei Gründen für diese Arbeit relevant. Erstens schafft sie überhaupt erst die Möglichkeit eine Evaluation mit der Thinking-Aloud-Methode durchzuführen. Zweitens wird aufgezeigt, ob es möglich ist, die auf Basis der Literaturanalyse ausgewählten Komponenten, zu einem funktionierenden System zusammenzuführen.

2.3 Evaluierung des Prototyps

In der Evaluierung des Konzepts wird, der nach dem Konzept erstellte Prototyp, von Nutzern evaluiert. Dabei soll auf die Thinking-Aloud-Methode zurückgegriffen werden, welche im Bereich Grundlagen (Methodik) nochmals ausführlicher beschrieben wird. Die

Evaluation soll im Zusammenhang mit der Bearbeitung von Aufgaben geschehen, welche wiederum im Kapitel „Evaluation mit der Thinking-Aloud-Methode“ detaillierter beschrieben werden.

2.4 Beantwortung der Forschungsfrage

Zuletzt steht die Beantwortung der Forschungsfrage im Fokus, welche lautet: „Welche wissenschaftlichen und technologischen Komponenten sollten zur Erstellung eines künstlich anthropomorphen Chatbots eingesetzt werden?“. Die Beantwortung der Forschungsfrage soll sich dabei nicht ausschließlich auf die Nennung der Komponenten beschränken, sondern vielmehr die Sinnhaftigkeit des Einsatzes darstellen. Zudem muss die Sinnhaftigkeit dabei mit dem Projektbezug und Vorhaben in Relation gesetzt werden, um eine vielschichtige Beantwortung der Forschungsfrage zu garantieren.

3 Grundlagen

Zur Angleichung des Kenntnisstandes folgt eine Definition der wichtigsten Begriffe und Technologien. Im Anschluss wird jeweils die Bedeutung der Begriffe in diesem Kontext von der allgemeinen Definition abgegrenzt, um den konkreten Gegenstand in dieser Arbeit aufzuzeigen.

3.1 Definition: Künstliche Intelligenz

Um den Begriff der „künstlichen Intelligenz“ zu definieren, wird dieser zunächst in seine Bestandteile aufgeteilt, um ihn dann in seiner Gesamtheit zu verstehen. Im Nachgang wird die Bedeutung des Begriffs im Kontext der Arbeit aufgezeigt.

3.1.1 Definition: Intelligenz

Intelligenz wird von Experten und der Bevölkerung unterschiedlich definiert und eingesetzt (Funke und Vaterrodt 2009). Selbst unter den Experten dieses Fachbereiches gibt es keine allgemeingültige Definition des Begriffes, aus diesem Grund möchte ich für diese Arbeit eine Definition verwenden, welche im Rahmen eines 1986 abgehaltenen Symposiums erarbeitet wurde (Funke und Vaterrodt 2009). Der Informationsverarbeitende Ansatz definiert Intelligenz über die „Prozesse geistiger Leistung“, welche die Kompetenzen des „Problemlösens“, „Urteilens“, „Vorstellens“ und „logischen Schlussfolgerns“ beschreiben (Funke und Vaterrodt 2009).

3.1.2 Kontextbasierte Definition: Künstlich

Der Duden definiert, das etwas künstlich ist, wenn es „nicht natürlich, sondern mit chemischen und technischen Mitteln nachgebildet, nach einem natürlichen Vorbild angelegt, gefertigt, geschaffen“ ist (Dudenredaktion (Hrsg.) o. J.).

3.1.3 Gesamtheitliche Definition

Künstliche Intelligenz (KI), oft auch artifizielle Intelligenz (AI) oder englisch „artificial intelligence“ genannt, beschreibt also die Leistung eines nach technischem Vorbild angelegtem Systems, die Kompetenzen, des „Problemlösens“, „Urteilens“, „Vorstellens“ und „logischen Schlussfolgerns“ zu erlernen und anzuwenden (Funke und Vaterrodt 2009; Dudenredaktion (Hrsg.) o. J.).

3.1.4 Abgrenzung

In dieser Arbeit findet die Technologie der künstlichen Intelligenz ausschließlich in der Netzwerk- und Verarbeitungskomponente Anklang. Die genau betroffene Komponente ist das Sprachverarbeitungs-Framework Dialogflow, welches sich der künstlichen Intelligenz bedient um die Nutzerintention einzuordnen, sowie die Satzstrukturen der Nutzereingabe zu verstehen (Hutchens und Alder 1998). Die Funktionsweise der Intentionserkennung und des gesamten Sprachverarbeitungs-Frameworks wird nicht als Fokus dieser Arbeit betrachtet und somit nicht detailliert beschrieben.

3.2 Definition: Chatbot

Das Konzept „Chatbot“ als solches ist keine Überlegung, die in den letzten Jahren aufgekomen ist, so lassen sich die Wurzeln von textbasierter Mensch-Computer-Interaktion bis in die 1960er Jahre zurückführen (B. A. Shawar und Atwell 2007a). Jedoch liegen Chatbots in der jüngsten Vergangenheit wieder im Trend, was sich an der Einbindung von Chatbot-Systemen in Soziale Medien und Messenger-Dienste erkennen lässt (Følstad

u. a. 2018). Der gegenwärtige Trend von Chatbots lässt sich dabei einerseits auf das steigende Angebot, andererseits auf die neuen Möglichkeiten zurückzuführen, welche durch technologische Fortschritte und künstliche Intelligenz entstehen (Khan und Das 2018; Zamora 2017). Mithilfe dieser neuen Möglichkeiten könnten Chatbots, so zumindest den Prozess der manuellen Informationsbeschaffung über Suchfunktionen vollkommen automatisieren und revolutionieren (Brandtzaeg und Følstad 2018). Um die Möglichkeiten des Einsatzes, sowie die Entwicklung der Funktionalitäten und der damit zusammenhängenden Technologien einschätzen zu können, wird in Kapitel 4 ein Einblick über die vergangenen Entwicklungen gegeben. Folgend wird der Begriff „Chatbot“ definiert und von anderen im Trend liegenden Technologien abgegrenzt.

„Chatbots sind Computerprogramme, die mit Benutzern in natürlicher Sprache kommunizieren“ - (B. A. Shawar und Atwell 2007a)

Um Chatbot-Systeme konkreter und einfacher aus der Vielzahl bekannter Technologien identifizieren zu können, werden sie nochmals genauer über ihre Funktionen definiert.

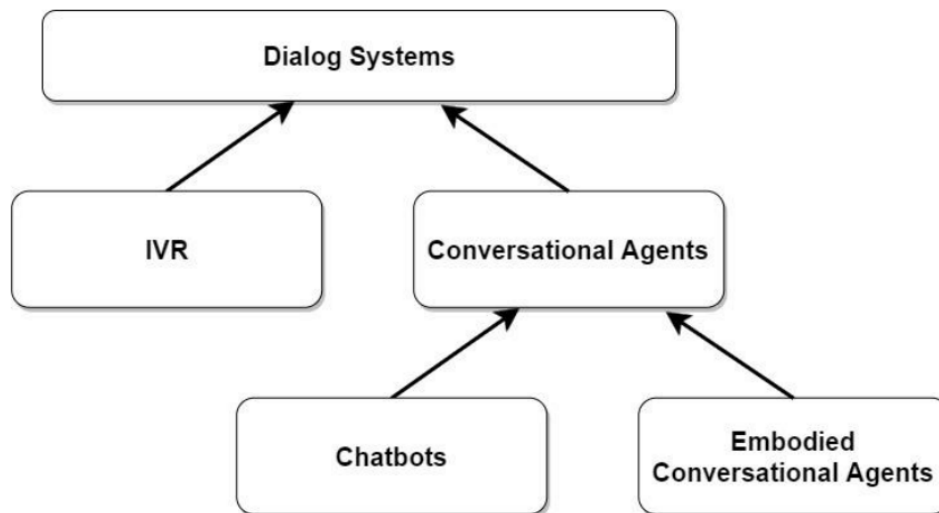


Abbildung 1: Hierarchie der verschiedenen Dialogsysteme (Radziwill und Benton 2017)

Chatbots gehören zu der Gruppe der Dialogsysteme, welche einen Dialog mit dem Nutzer aufbauen (Riccardi u. a. 2012). Dialogsysteme lassen sich dabei in zwei Kategorien unterteilen. Zum Einen, die sogenannten IVR-Systeme („Interactive Voice Response“), welche die Funktion besitzen mit dem Nutzer über Toneingabe und Tonausgabe zu kommunizieren (Asthana, Singh, und Singh 2013; Radziwill und Benton 2017). Zweitens die CA-Systeme („Conversational Agents“), welche ausschließlich mittels Texteingaben funktionieren. CA-Systeme unterscheidet man in Chatbots und ECA-Systeme („Embodied Conversational Agents“). Wobei ECAs einen Avatar bei der Kommunikation zur Verfügung stellen, welcher die Konversation mittels Mimik und Gestik beleben soll (Radziwill und Benton 2017). Diese Arbeit wird sich ausschließlich mit dem Thema Chatbot beschäftigen, somit ist eine Auseinandersetzung mit Technologien zur Erstellung von Avataren nicht zu erwarten.

3.2.1 Definition: Multi-Agent-System (MAS)

Um den Begriff „Multi-Agent“ zu verstehen, sollte man zunächst Single-Agent-Systeme und den Begriff Agent verstanden haben. Single-Agent-Systeme sind Systeme, die sich ausschließlich einer Komponente bedienen. Multi-Agent-Systeme bestehen im Gegensatz zu Single-Agent-Systemen aus mehreren Agenten (Calvaresi u. a. 2019; Prospero u. a. 2017).

Als Agent wird hierbei eine Komponente betrachtet, die Teile der Antwort beeinflusst (Calvaresi u. a. 2019; Prospero u. a. 2017). Dabei werden den verschiedenen Agenten verschiedene Aufgaben zugeteilt. Eine gängige Aufteilung der Agenten ist in „personal-agent“ (Personalagent) und „Gateway-agent“ (Informationsagent) (Calvaresi u. a. 2019). Wobei der Personalagent die Nutzerpräferenzen in die Antworten einfließen lässt und das Gespräch in Bezug auf die Emotionalität und Metaebene steuert (Calvaresi u. a. 2019; Prospero u. a. 2017). Der Informationsagent ist für die Steuerung des Gespräches auf der Informationsebene zuständig (Calvaresi u. a. 2019). Er sorgt dafür, dass die Nutzererwartungen bezüglich des Informationsgehaltes erfüllt werden.

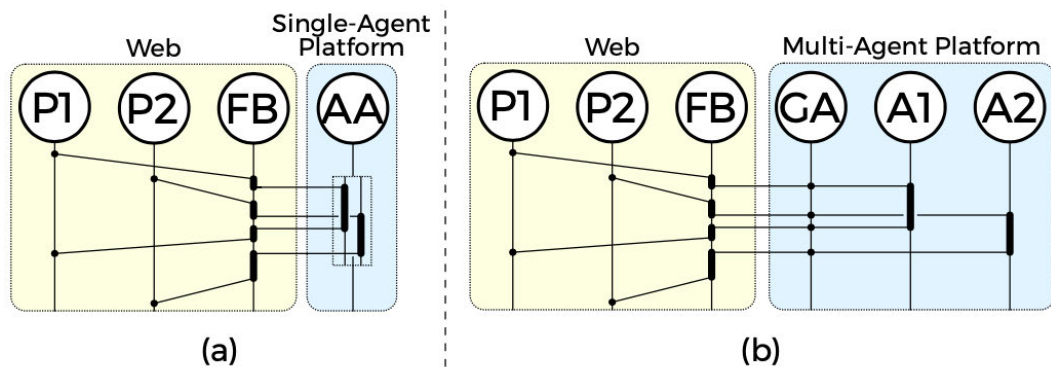


Abbildung 2: Single-Agent gegen Multi-Agent-Lösungen (Calvaresi u. a. 2019).

3.2.2 Definition: Anthropomorphismus bei Chatbots

Anthropomorphismus (*gebildet aus den griechischen Wörtern „anthropos“ (Mensch) und „morphē“ (Gestalt)*) steht für die Übertragung menschlicher Eigenschaften auf etwas Nichtmenschliches (hier: Chatbot) (Dudenredaktion (Hrsg.) o. J.). Wie diese Übertragung umgesetzt werden soll, ist in der Fachliteratur nicht weiter definiert. Um dennoch einen Einblick in die Umsetzungsmöglichkeiten zu geben, werden in der Literaturstudie in Kapitel 4 zwei beispielhafte Umsetzungen dargestellt.

3.3 Methodik

3.3.1 Definition: Thinking-Aloud

Die Thinking-Aloud-Methode stammt Ursprünglich aus der Problemlöseforschung und erlaubt die Einsicht in die mentalen Prozesse (Holzinger 2006). Bei dieser Methode ist es üblich Nutzern eine Aufgabe zu stellen und diese zu beten, während dessen Erfüllung ihre Gedankengänge laut mitzuteilen. Dabei ist es sinnvoll, die Aussagen der Nutzer aufzuzeichnen, um wichtige Aussagen erneut ansehen zu können (Holzinger 2006).

3.3.2 Nachteile von Thinking-Aloud

Die Nachteile dieser Methode halten sich gewissermaßen in Grenzen, sind jedoch im Sinne der Vollständigkeit zu nennen. Ein Nachteil ist, dass viele Nutzer gehemmt sind

ihre Gedanken auszusprechen und somit die Aussagekraft der Methode vermindert wird (Holzinger 2006). Zudem setzt die Methode im Fall dieses Projektes einen lauffähigen Prototypen voraus.

3.3.3 Vorteile von Thinking-Aloud in besonderen Zeiten

Der Forscher Andreas Holzinger bezeichnet die Thinking-Aloud-Methode als „Königsmethode im Usability Engineering“, da sie seiner Meinung nach, bereits aussagekräftige Ergebnisse bei einer kleinen Anzahl an Endbenutzern bringt (Holzinger 2006). Ein weiterer Aspekt, welcher zur Auswahl der Methode geführt hat, war das diese Methode in Zeiten von Corona Online ohne Ergebnisverlust durchgeführt werden konnte.

3.4 Einführung in genutzte Technologien

3.4.1 APIs („application programming interface“) und Webservices

Das Internet ist für einen Großteil der Bevölkerung zu einem Alltagsgut geworden. Doch nur kleine Bevölkerungsteile kennen die Technologien, Normen, Standards und Abkommen, die das Internet weltweit verfügbar und nutzbar machen. In dieser Arbeit wird das bedeutendste Protokoll, sowie auch die wichtigsten der genutzten Web-Technologien kurz erläutert, um mithilfe dieser Kenntnisse Webservices und APIs („application programming interface“) verstehen zu können.

3.4.1.1 Hypertext Transfer Protocol (HTTP)

Das Hypertext Transfer Protocol (HTTP) ist mittlerweile nicht mehr nur ein Netzwerkprotokoll, sondern auch die Basis eines sehr großen Systems, welches nicht mehr weg zu denken ist (Mogul 2002). Das Protokoll wurde von Tim Berners-Lee und seinem Team am europäischen Kernforschungszentrum (CERN) im Jahr 1989 entwickelt und 1991 eingeführt (Berners-lee, Cailliau, und Groff 1992; Myrach und Zwahlen 2008).

Der Zweck des HTTP ist, Regularien für die Kommunikation zwischen Computern aufzustellen, dabei wird in zwei Typen von Computern unterschieden.

1. Der Client („Kunde“), welcher Dienste oder Dienstleistungen in Anspruch nimmt oder benötigt (Fielding 1999).
2. Der Server („Diener oder Bediensteter“), welcher Dienste zur Verfügung stellt (Fielding 1999).

Die Kommunikation zwischen den beiden Computern erfolgt über Messages („Nachrichten“), welche jeweils in Requests und Response eingeordnet werden (Fielding 1999). Requests („Anfragen“) werden meist von dem Client an den Server gesendet, um einen bestimmten Dienst des Servers zu nutzen, beispielsweise Informationen anzufragen (Fielding 1999). Der Server verarbeitet die empfangene Request und antwortet dem Client mit einer Response („Antwort“) (Fielding 1999). Dabei ist zu beachten, dass der Server sich meist in einem passiven, wartenden Zustand befindet und der Client den Nachrichtenaustausch beginnt (Fielding 1999).

Neben der Unterscheidung der Nachrichten in Request und Response ist der Aufbau einer HTTP-Message für diese Arbeit von Bedeutung. HTTP-Nachrichten sind immer gleich aufgebaut, sie bestehen aus den Message-Headers und dem Message-Body (Fielding 1999).

Die „Headers“ der Nachricht geben dem Server vor, wie der Body („Inhalt“) der Nachricht zu interpretieren ist (Fielding 1999). Beispiel: „content-length“ (Repräsentiert die Länge des Bodys).

Der Body („Körper beziehungsweise Inhalt“) der Nachricht gibt den Inhalt der Nachricht an (Fielding 1999). Die Form, in welcher sich der Inhalt befindet, wird dem Empfänger über den Header „content-type“ aufgezeigt, damit dieser den Inhalt verstehen kann (Fielding 1999).

Um dem Sender der Request Auskunft über die fehlerfreie Verarbeitung der Nachricht zu geben, definiert das Protokoll sogenannte Statuscodes (Fielding 1999). Die genaue Bedeutung dieser ist in der HTTP-Definition nachzulesen und für das Verstehen dieser Arbeit nicht notwendig.

3.4.1.2 REpresentational State Transfer (REST)

REST beschreibt ein Programmierparadigma von verteilten Systemen, insbesondere Webservices. Der Architekturstil wurde im Jahr 2000 von Roy T. Fielding entwickelt und in seinem Werk „Architectural styles and the design of network-based software architectures“ beschreiben (Ed-douibi u. a. 2016; Fielding 2000a). Die Architektur baut auf dem Hypertext Transfer Protocol auf, welche Fielding in seiner Dissertation um folgende Grundprinzipien erweiterte (Fielding 2000a; 2000b).

1. Client-Server

Einordnung der Komponenten in nutzernahe Komponenten und Datenspeicherung (Fielding 2000b).

2. Statelessness („Zustandslosigkeit“)

Alle zur Verarbeitung benötigten Informationen sind in einer Nachricht erhalten. Die verarbeitende Komponente benötigt keine Kontext- oder Session-Informationen zur Verarbeitung (Fielding 2000b).

3. Cache

Die Zwischenspeicherung von großen Datenpaketen zu Minimierung des Datenstroms soll genutzt werden (Fielding 2000b).

4. Uniform Interface („Einheitliche Schnittstelle“)

a. Addressable resources („Adressierbare Ressourcen“)

Jede Ressource muss mittels „Uniform Ressource Identifier“ (URI) adressierbar sein (Ed-douibi u. a. 2016).

b. Repräsentationsorientierung („Representation-oriented“)

Alle Dienste müssen von einer Adresse erreichbar sein und ihre Dienste und Dokumentationen in verschiedenen Formen (JSON, XML oder HTML) ausliefern (Ed-douibi u. a. 2016).

c. Zustandslosigkeit („Statelessness“)

Siehe 2.

d. Uniform and Constrained Interface („Selbstbeschreibende Methoden“)

Die REST-Schnittstelle muss die Standard-HTTP-Methoden („GET, PUT, POST, DELETE, ...“) beinhalten und bei weiteren Diensten ist auf eine selbstbeschreibende Benennung zu achten (Ed-douibi u. a. 2016).

5. Layered System („Mehrschichtige Systeme“)

Die Architektur sollte als mehrschichtiges System angelegt werden, sodass der Anwender nur die Wurzel des Systems ansprechen muss, aber auch nur über die Wurzel an die „tieferliegenden“ Dienste gelangt.

6. Code-On-Demand

Im Bedarfsfall können Codeteile übertragen werden, damit sollen dem Client Funktionen zur Verfügung gestellt werden, welche den Client-Quellcode vereinfachen soll.

3.4.1.3 Application Programming Interfaces (API)

APIs sind Schnittstellen, welche es Programmierern erlauben externe Programme und vor allem deren Prozeduren, Routinen, Werkzeuge und Funktionen zu nutzen (Minnaert, Sawyer, und Thayer 2002). Diese externen Programmteile können dann im eigenen Quellcode genutzt werden (Minnaert, Sawyer, und Thayer 2002). Wie die Schnittstelle funktioniert beziehungsweise genutzt werden kann, ist in der Regel in einer Dokumentation beschrieben (Minnaert, Sawyer, und Thayer 2002).

3.4.1.4 Webservices

Wie ein Webservice definiert wird, wurde 2004 von dem World Wide Web Consortium wie folgt festgelegt:

„A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards“ (W3C 2004).

3.4.1.5 Unterscheidung von APIs und Webservices

Webservices nutzen festgelegte Nachrichtenformate und Spezifikationen, wie SOAP oder WSDL, dadurch unterscheiden sie sich von APIs, da diese an keine Spezifikationen gebunden sind (Bush 2019; W3C 2004). Außerdem sind Webservices an die Nutzung innerhalb eines Netzwerkes gebunden, wo hingegen APIs sowohl Online als auch Offline eingesetzt werden können (Bush 2019). Abschließend ist jeder Webservice eine API aber nicht jede API ein Webservice (Bush 2019).

3.4.2 Webhooks

Webhooks sind nutzerdefinierte Callbacks, welche durch Events ausgelöst werden. Webhooks nutzen ebenfalls das Hypertext Transfer Protocol und benachrichtigen einen bestimmten Server mit Hilfe des Sendens eines Requests über ein Ereignis (Krishnan, Pal, und Venkatasubramanian 2018).

3.4.3 Dialogflow (DF)

Im Jahr 2012 bekam die Firma Speaktioit Kapital zur Entwicklung eines Natural Language User Interface, welches als Api.ai in 2014 veröffentlicht wurde (Lee 2018). In 2016 kaufte Google Api.ai und benannte es 2017 in Dialogflow um (Lee 2018). Seit dem wird DF von Google weiterentwickelt und bildet die Grundlage der Google Assistenten Google Home und Google Auto, sowie von Amazon Alexa (Lee 2018).

DF ist also einerseits ein Programm, welches mithilfe von maschinellem Lernen und künstlicher Intelligenz natürliche Sprache erkennen kann, andererseits ein Framework zur Erstellung von Dialogsystemen (Lee 2018; „Dialogflow“ o. J.).

In dieser Arbeit werden ausschließlich die für den Nutzer interessanten Prozesse und Funktionen von DF thematisiert werden. Die Funktion des Natural Language Processing (Spracherkennung und Verarbeitung), sowie die Erkennung der Nutzerintention (Intents) werden stehen nicht im Fokus dieser Arbeit.

3.4.3.1 Aufbau von DF

Beim Start von DF muss der Nutzer ein Projekt anlegen, welches ebenfalls einem Agenten (Agent) entspricht. Hierbei kann zwischen einem Main bzw. Single-Agent-Projekt und einem Multi-Agent-Projekt (Agent als Mega-Agent anlegen) entschieden werden.

Intents (Intentionen)

Für jeden Agent lassen sich sogenannten Nutzerintentionen (Intents) erstellen. Wobei jeder Agent die Standardintentionen Default-Welcome-Intent und Default-Fallback-Intent enthält.

Jeder Intent enthält dabei Contexts (Kontext), welche in Input- und Output-Kontexte unterteilt sind. Der Default-Welcome-Intent enthält keinen Input-Kontext, da er der erste Intent in jeder Konversation ist und noch kein Kontext besteht. Bei allen folgenden Intents sollten Input-Kontexte hinzugefügt werden, um einen Konversationsfluss zu generieren. Output-Kontexte werden wie Variablen vom Ersteller definiert und während der Konversation automatisch von DF befüllt.

Entities (Entitäten)

Entitäten dienen DF, wie Kategorien, welche in Kontexten beziehungsweise in Parametern abgespeichert werden. Folgend wird eine Entität für unseren Beispielagenten erstellt.

Entitätsname: Origin

- Asiatisch: chinesisch, China, thailändisch, Thai, Thailand, Japan, japanisch, Korea, koreanisch, Vietnam, vietnamesisch, asiatisch
- Amerikanisch: BBQ, Amerika, amerikanisch, Fastfood
- Osteuropäisch: russisch, Russland, romanisch, ungarisch, Ungarn, Rumänien, polnisch, Polen, Litauen, Tschechien, tschechisch
- Mediterran: Spanien, Italien, Frankreich, Portugal, Griechenland, Türkei, türkisch, griechisch, italienisch, spanisch, portugiesisch, französisch
- Skandinavisch: Schweden, Norwegen, Finnland, Dänemark, dänisch, finnisch, norwegisch, skandinavisch, schwedisch

Trainingphrases (Trainingssätze)

Um DF aufzuzeigen, welche Intention der Nutzer mit einer Frage oder Aussage hat, muss der Ersteller Trainingssätze eingeben. Hierbei handelt es sich um beispielhafte Aussagen.

Beispiel

Wir möchten dem Nutzer die Möglichkeit geben ein Restaurant anhand der Herkunft des Speisenangebotes auszuwählen. Dafür definieren wir eine Nutzerintention recognizeOrigin und trainieren diese mit folgenden Trainingssätzen:

- ich möchte chinesisch essen
- ich mag Thailand
- ich habe Hunger auf koreanisches Essen
- ich mag deutsche Küche
- ich liebe Norwegen
- ich mag Schweden
- ich möchte skandinavisch probieren
- ich möchte russisch essen
- ich mag Mittelmeerküche
- ich möchte etwas Mediterranes
- ich möchte BBQ
- ich habe Lust auf Fastfood

DF versucht nun einzelne Worte in den Sätzen zu markieren und somit eine Entität zu identifizieren. Wenn der Ersteller keine DF systemeigenen Entitäten, wie sys.location, sys.url, ... nutzen möchte kann er bestimmte Worte in den Trainingssätzen markieren und eigenen Entitäten zuordnen, um so DF auf die Auswahl der eigenen Entitäten zu konfigurieren.

Parameter

Parameter sind Variablen, welche dem Nutzer in der Konfiguration und Erstellung von benutzerdefinierten Antworten zur Verfügung stehen.

Wir hinterlegen den Parameter Origin welcher auf die Entität Origin referenziert und den Wert \$origin beinhaltet.

Responses

Responses in DF sind analog zur HTTP-Response die Antworten die DF zurückgibt. Sie können benutzerdefiniert erstellt und mit Parametern gefüllt werden.

Für unser Beispiel ändern wir die Standard Response in der Default-Welcome-Intent in "Hallo, nach welcher Art Restaurants soll ich für dich suchen?" um. Ebenfalls fügen wir dem recognizeOrigin-Intent die Response "Ok, ich suche für dich nach Restaurants der Kategorie \$origin." hinzu, um dem Nutzer eine kontextabhängige Antwort und das Gefühl zu geben, das wir seine Anfrage verstanden haben.

3.4.3.2 Funktionsweise von DF und Beispiel Konversation

In diesem Abschnitt der Arbeit wird mithilfe des aufgestellten Beispiel Agenten ein Gesprächsfluss und die dahinterliegenden Verknüpfungen von Entitäten, Intentionen, Trainingssätzen, Parameter und zuletzt Responses aufgezeigt.

U: Hallo

DF findet keinen Kontext und wählt aufgrund fehlender Input-Kontext-Parameter den Default-Welcome-Intent aus. DF gibt benutzerdefinierte Response aus.

DF: Hallo, nach welcher Art Restaurants soll ich für dich suchen?

U: Ich möchte japanisch essen.

DF erkennt die Nutzerintention und aktiviert unser recognizeOrigin-Intent. Dabei erkennt DF „japanisch“ als Wert und sucht in seinen Entitäten nach japanisch, dabei findet DF die Kategorie „Asiatisch“ und setzt dies in eine benutzerdefinierte Response mit dem Parameter Origin ein und gibt diese aus.

DF: Ok, ich suche für dich nach Restaurants der Kategorie Asiatisch.

4 Literaturstudie

In diesem Kapitel werden die Anfänge, Entwicklungen und der derzeitige Forschungsstand anhand von erarbeiteter Fachliteratur dargestellt. Zu Beginn wird die Entwicklung von Chatbots an dem Hauptkriterium dem Loebner Preis angegeben. Daraufhin wird auf Multi-Agent-Chatbot-Systeme und anthropomorphe Chatbots eingegangen.

4.1 Bekannte Chatbots

Im Januar 1966 veröffentlichte Joseph Weizenbaum den Chatbot „ELIZA“, welcher Konversationen zwischen Menschen und Computern möglich machte (Weizenbaum 1966). ELIZA konnte auf Walisisch, Deutsch und Englisch mit dem Nutzer kommunizieren und bereits auf einfache Fragen in ganzen Sätzen antworten (Weizenbaum 1966).

Im Jahr 1991 startete der Loebner Preis, ein Preis, der von Hugh Gene Loebner ausgeschrieben wird. Der Loebner Preis entwickelte sich schnell zu einer der Hauptauszeichnung in der Chatbot-Entwicklung. Er zeichnet sich durch eine Weiterentwicklung des Turing-Tests aus. Der Turing-Test ist eine von Alan Turing im Jahr 1950 entwickelte Idee, zur Feststellung, ob ein Computer dem menschlichen Denkvermögen gewachsen ist. In der Prüfung zum Loebner Preis wird das Computerprogramm einem 25-minütigen Test unterzogen (Mauldin 1994).

In den Jahren 1991, 1992, 1993, 1995 gewann der von Joseph Weintraub implementierte PC Therapist, welcher aus sieben Modulen bestand. Eines zielte darauf ab, Kindern und Jugendlichen den Umgang mit dem Computer näher zu bringen (Weintraub 1999).

Im Jahr 1996 veröffentlichte Jason L. Hutchens seinen Chatbot HeX und gewann mit ihm den Loebner Kontest. HeX wurde von Hutchens innerhalb eines Monats entwickelt und hatte viele Gemeinsamkeiten mit ELIZA (Hutchens und Alder 1998; B. A. Shawar und Atwell 2007a).

Ein Meilenstein in der Chatbot-Entwicklung gelang Hutchens im Jahr 1998 mit MegaHAL (B. A. Shawar und Atwell 2007a). MegaHAL bediente sich neben den Strukturen aus HeX (somit auch ELIZA) auch an dem von Andrei Andrejewitsch Markov entwickeltem Markov-Modell (Hutchens und Alder 1998; Shawar und Atwell 2007a).

Im den Jahren 1998 und 1999 gewann der Chatbot „Albert One“, welcher seit 1997 von Robby Garner in DOS entwickelt wurde (Lun 1998; Müller 2019; Verduijn o. J.). Albert One überzeugte besonders durch sein menschenähnliches Verhalten (Verduijn o. J.).

In den Folgejahren 2000, 2001 und im Jahr 2004 konnte sich Richard Wallace den Loebner Preis sichern (Loebner 2009b). A.L.I.C.E (Artificial Linguistic Internet Computer Entity) wurde seit 1995 von Wallace entwickelt und war der erste webbasierte Chatbot der beim Loebner Kontest vorgestellt wurde (Shawar und Atwell 2007a; Verduijn o. J.). 1998 wurde A.L.I.C.E in die Programmiersprache Java übersetzt (B. Shawar und Atwell 2004).

Der Chatbot Jabberwacky von Rollo Carpenter konnte sich den Preis in den Jahren 2005 und 2006 sichern (Loebner 2009a). Seine Versionen George und Joan überzeugten vor allem durch ihre Lernfähigkeit während der Nutzerinteraktion (Loebner 2009a).

In den Jahren 2010, 2011 gewannen die Chatbots Suzette und Rosette von Bruce Wilcox (Loebner 2009c). Das erstaunliche war dabei, dass Hugh Loebner 2010 den Preis an Suzette verlieh obwohl die Weiterentwicklung von A.L.I.C.E besser abschnitt (Loebner 2009c). Loebner begründet dies damit, dass Suzette einen der Richter zum Narren hielt (Loebner 2009c).

Bruce Wilcox konnte mit seinem Chatbot „Rose“ den Kontest in den Jahren 2014 und 2015 nochmals für sich entscheiden („AISB - The Society for the Study of Artificial Intelligence and Simulation of Behaviour - Loebner Prize“ 2015).

Seit dem Jahr 2015 gewinnt Steve Worswick mit seinem Chatbot „Mitsuku“ den Loebner Preis. Worswicks Bot überzeugte 2019 erneut mit seinem erstaunlich menschenähnlichen Verhalten (Müller 2019).

4.2 Multi-Agent-Chatbot-Systeme

In der Literatur werden zwei Arten von Multi-Agent-Chatbot-Systemen beschrieben, wobei eines über mehrere Agenten mit gleichem Funktionsumfang verfügt, welche sich nur durch die Charaktereigenschaften unterscheiden (Calvaresi u. a. 2019; Hettige und Karunananda 2015). Das andere System verfügt nur über einen Agenten, welcher mit dem Kunden kommuniziert, jedoch werden die Antworten im Hintergrund dynamisch von einer Emotions-Komponente angepasst, so dass der Single-Agent mehrere Persönlichkeiten erhält (Hettige und Karunananda 2015). Im folgenden Abschnitt wird zu jeder Multi-Agent-Technologie ein Beispiel vorgestellt.

In dem Konferenz-Paper „Social Network Chatbots for Smoking Cessation“ wurden Single-Agent-Chatbot-Systeme ohne emotionale Anpassung in ihrer Wirkung zu Multi-Agent-Chatbot Systemen mit mehreren Agenten verglichen. Dabei liegt der Fokus dieser Arbeit auf der Umsetzung des Multi-Agent-Chatbots SMAG. SMAG besteht aus einem „Gateway-agent“, welche die emotionale Situation des Nutzers einschätzt und einem der drei weiteren Agenten zuordnet. Die drei Agenten sind darauf trainiert bestimmte Stimmungen und personalisierte Ziele in der Nutzerkommunikation zu verarbeiten und wirken demnach menschlich auf den Nutzer (Calvaresi u. a. 2019).

Die Vorteile dieses Systems sind, dass auf jeden Nutzer individuell eingegangen werden kann und gerade in der Suchtbehandlung eine Art persönlicher Coach zur Verfügung gestellt wird. Ebenfalls können die Gründe für eine Kommunikation mit dem Chatbot zusammen gefasst werden und SMAG kann je nach Situation einen emphatischeren Agenten oder einen optimistischeren Agenten auswählen (Calvaresi u. a. 2019).

Die Nachteile des Systems sind, dass die Nutzer viele persönliche Informationen über sich preisgeben, dies ist zwar bei der Suchtberatung üblich, jedoch ist dies nicht für jeden Kontext denkbar. Außerdem werden mehrere Phasen zur Erstellung des Systems verwendet, sodass zumindest ein releasefähiger Single-Agent-Chatbot benötigt wird um Nutzerdaten zu sammeln (Calvaresi u. a. 2019). Der Vorteil, dass die Auswahl des Agenten durch das Nutzerverhalten beeinflusst wird, kann ebenfalls in anderen Kontexten zum Nachteil werden, wenn beispielsweise die Entwickler auf die Emotionalität der Antworten Einfluss nehmen möchten.

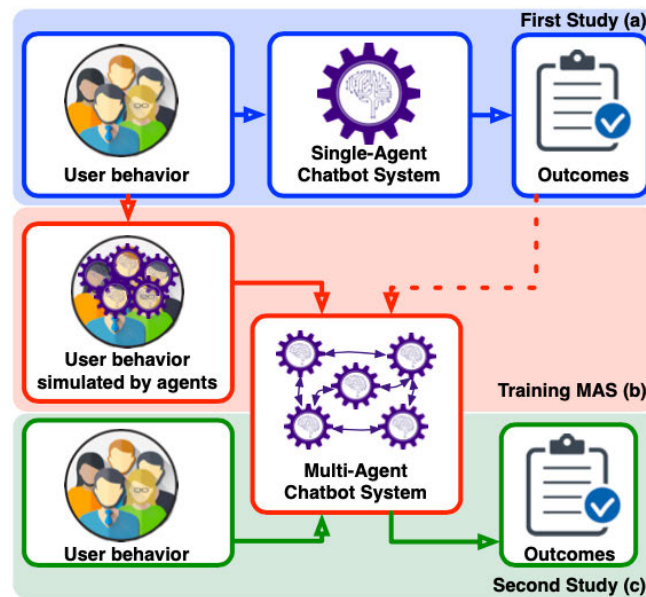


Abbildung 3: Phasen der Erstellung (Calvaresi u. a. 2019).

Eine weitere Möglichkeit ein Multi-Agent-Chatbot System zu entwerfen, zeigt das Projekt Octopus, welches den Nutzer ausschließlich mit dem Manager-Agent kommunizieren lässt. Octopus besitzt folgende acht Unteragenten, das Kernsystem, die Nutzerschnittstelle, ein Ontologie System, die NLP-Einheit, das Lernsystem, das Aktionssystem, das Kommunikationssystem, ein Suchsystem und ein System für den Datenzugriff, welche durch den Kern verwaltet werden (Hettige und Karunananda 2015).

Die Vorteile dieses Systems sind, dass jede Kommunikation aus Informationen und Emotionen bestehen und diese dynamisch verwaltet werden können. So können Entwickler Einfluss auf die Antworten des Chatbots nehmen (Hettige und Karunananda 2015).

Die Nachteile des Systems sind, dass der Nutzer keinen Einfluss auf die Emotionen des Chatbots nehmen kann. Ein weiterer Nachteil ist der Umfang an Modulen und die Menge an Regeln, die bei einer releasefähigen Version anfallen würden, dies lässt sich jedoch durch die Feinheiten der Emotionalität und der Menge der Entitäten regulieren.

4.3 Anthropomorphe Chatbots

Wie bereits im Kapitel Grundlagen behandelt, sind anthropomorphe Chatbots, Informationsassistenten, welche dem Nutzer menschliche Eigenschaften vorspielen oder diese tatsächlich aufweisen. Im folgenden Abschnitt sollen Vor- und Nachteile von anthropomorphen Chatbots und die Möglichkeiten der Anthropomorphisierung eines Chatbots aufgezeigt werden.

4.3.1 Vor- und Nachteile anthropomorpher Chatbots

In der Wissenschaft wird es meist als Ziel angesehen, ein möglichst menschenähnliches System zu entwickeln und zu implementieren, dabei ist jedoch der Grad der Anthropomorphisierung entscheidend. So beschreibt Masahiro Mori (1982) den Uncanny-Valley-Effekt, bei dem viele Nutzer gerade bei einem hohen Grad der Anthropomorphisierung Furcht verspüren und somit die Interaktion negativ erfahren wird (Lotze 2016).

Ein anderes Beispiel in dem ein durch künstliche Intelligenz anthropomorphisierter Chatbot Furcht verbreitete ist „Tay“, ein von Microsoft entwickelter Chatbot zur Unterhaltung. Der Bot wandelte seine Aussagen durch Nutzer Kommentare zu rassistischen Aussagen und löste regelrechte Furcht vor künstlicher Intelligenz auf Twitter aus (Steiner 2016).

Jedoch kann die Vermenschlichung des Chatbotverhaltens auch positiv Einfluss nehmen. So generieren vermenschlichte Systeme eine höhere Aufmerksamkeit bei den Nutzern und die Gesprächsabbruchrate sinkt (Lotze 2016). Außerdem sollen die Interaktionen auf ein qualitativ höheres Level gehoben werden, als bei allein textbasierten Systemen (Lotze 2016).

4.3.2 Möglichkeiten zur sinnvollen Anthropomorphisierung von Chatbots

Wie bereits im vorherigen Abschnitt beschrieben ist der Grad der Vermenschlichung bei der Umsetzung entscheidend. So ist ein vollkommen vermenschlichtes System nicht wünschenswert, da die Interaktion, durch kurzfristiges wechseln des Dialogziels oder auch durch Stimmungsschwankungen seitens des Systems sehr schwerfällig und ineffizient wird (Lotze 2016).

Die Möglichkeiten der zur sinnvollen Anthropomorphisierung sind also begrenzt. Dennoch sollte der Chatbot anthropomorphe Elemente enthalten, um beispielsweise durch Stilmittel, wie Humor dem Nutzer im Gedächtnis zu bleiben (Lotze 2016).

4.3.3 Empathie als Form der Vermenschlichung

Empathie als Charakterzug beschreibt die Möglichkeit sich in andere Menschen und deren emotionale Lage hinein zu versetzen. Emphatisches Verhalten ist nach einigen Experten im Kundenkontakt und in der Gastronomie sehr wichtig (Wehr 2019). Für einen Chatbot, welcher meistens mit Kunden im Kontakt steht kann Empathie also genau so wichtig sein, wie für die Mitarbeiter der Kundenbetreuung.

5 Konzeption

In diesem Kapitel der Arbeit wird näher auf die Konzeption eingegangen. Die Konzeption beginnt mit der Auswahl des Chatbot-Frameworks, da dies Komponente die Hauptkomponente des Projektes darstellt und mitunter die meisten Einschränkungen mit sich bringen kann.

5.1 Framework und Aufbau des Multi-Agent-Systems

Die Ansprüche, welche an das Framework gestellt werden, sind zunächst einmal eine reibungslose Anbindung an einen Datenspeicher, die Möglichkeit das Framework als Multi-Agent-System zu betreiben, sowie die Unterstützung der Netzwerkfähigkeit, da das Framework innerhalb einer App genutzt werden soll.

5.1.1 Auswahl des Frameworks

Die Entscheidung Dialogflow als Chatbot Framework zu nutzen basierte auf zwei Punkten. Erstens konnte Dialogflow die Anforderungen, welche an das Framework gestellt wurden, alle erfüllen. Beispielsweise ist die Möglichkeit Dialogflow mit Informationen externer Datenquellen zu versorgen, direkt durch die Funktion Fullfillment und die Anbindung von Webhooks vorgesehen und vergleichsweise schnell umzusetzen. Außerdem ist Dialogflow ein Teil der Google Cloud Plattform, woraus sich ergibt das der Agent die Netzwerknutzung unterstützt. Zweitens Dialogflow-Klassenbibliotheken wurden bereits bei der Erstellung des App-Prototypen genutzt, somit hätte eine Implementation mit einem projektfremden Framework zu einer nicht notwendigen und arbeitsintensiven Anpassung seitens des Projektes geführt.

Aufgrund dessen, dass emotionales Verhalten durch ein Single-Agent-System nicht angemessen abgebildet werden kann, wurde sich im Falle dieses Projektes für ein Multi-Agent-System und gegen ein Single-Agent-Chatbot entschieden. Wie bereits in Kapitel 4 herausgestellt, stehen hierbei ein Multi-Agent-System mit mehreren Agenten oder ein Multi-Agent-System mit Emotional Agent zur Verfügung. Aufgrund der in der Literaturstudie erarbeiteten Erkenntnisse, zu den Anpassungsmöglichkeiten für Entwickler ist die Wahl auf ein Multi-Agent-System mit Emotional Agent gefallen. Die Entwicklung wird also aus verschiedenen Komponenten bestehen, welche im folgenden Abschnitt einzeln erläutert und evaluiert werden.

1. User Interface

Die Nutzerschnittstelle dient der Kommunikation des Nutzers mit dem Chatbot, über Sie werden neben den Textnachrichten, auch die MealCards (kleinst Steckbriefe der vorgeschlagenen Gerichte mit Gerichtsbezeichnung, Bild und Preis) gesendet und empfangen.

Im weiteren Projektverlauf kann als Userinterface auch der Facebook Messenger oder eine App dienen.

2. Processing and Network Component (Verarbeitungs- und Netzwerkkomponente)

Die Verarbeitungs- und Netzwerkkomponente besteht aus der Plattform Dialogflow, welche mithilfe von künstlicher Intelligenz und Natural Language Processing die Nutzerintentionen, die Auswahl von Speisen oder die Festlegung der Speisenherkunft erkennt und über das Netzwerk zum Kern des Systems überträgt.

3. Core (Kern)

Der Kern des Chatbots ist eine API, welche die formatierten Informationen der Verarbeitungs- und Netzwerkkomponente empfängt und die Informationen des Information Agents abfragt und diese mithilfe des Emotional Agents

emotionalisiert. Sind alle Verarbeitungsschritte des Kerns durchlaufen, so stellt er die Antwort über die Netzwerkkomponente der Nutzerschnittstelle zur Verfügung.

4. Information Agent (Informationsagent oder auch Datenbankschnittstelle)
Der Information Agent wird mithilfe der Objektklassen in die API eingebunden und gibt der API die Möglichkeit mittels Datenbank-Lese-, -Schreibe- oder -Lösch-Befehlen den Informationsumfang des Chatbots zu erweitern, zu verringern oder zur Verfügung zu stellen.
5. Emotional Agent (Anthropomorphisierungs- oder Emotionalisierungskomponente)
Der Emotional Agent dient der nachträglichen Verarbeitung der Nachrichten, dazu bedient er sich vordefinierten Regeln und Stilmitteln, mit denen er die bereitgestellten Informationen emotionalisiert und verändert. Danach stellt er diese dem Kern wieder zur Verfügung.

5.2 Datenbank

Ein wichtiger Aspekt bei der Konzeption eines Chatbots, ist das zur Verfügung stellen von Informationen. Dabei ist es bei statischen Informationen möglich dies ausschließlich innerhalb des Frameworks zu realisieren. Für Such- und Empfehlungs-Chatbots ist es jedoch üblich eine Datenbank als Informationsquelle zu nutzen, da die Informationen sich dynamisch verhalten. Beispiele für die Veränderung des Datenbestandes im Kontext Rendezvous sind, das Wechseln von Speisen oder Herkunft der Zutaten, sowie Anpassung der Öffnungszeiten.

Bevor jedoch die Anbindung des Chatbots an eine Datenbank thematisiert werden kann, sollte sich für eine Datenbanktechnologie entschieden werden. Zur Auswahl stehen folgende Technologien. Erstens eine herkömmliche SQL-Datenbank, welche auf der Theorie von relationalen Datenbanken basiert (Silva, Almeida, und Queiroz 2016; Spertus und Stein 2000). Zweitens eine NoSQL-Datenbank (Not only SQL), welche sich neben der relationalen Datenbankkomponenten auch anderen Speichermethoden bedient (Rautmare und Bhalerao 2016; Silva, Almeida, und Queiroz 2016). Folgend sollen die beiden Technologien kurz beschrieben werden, um im Anschluss Vor- und Nachteile im Bezug auf die Implementation heraus zu stellen.

5.2.1 SQL (Standard Query Language)

SQL ist eine Programmiersprache zum Verwalten Relationalen Datenbanken (Silva, Almeida, und Queiroz 2016). Sie arbeitet mit Tabellen, welche aus Spalten und Zeilen bestehen.

Nachteile von SQL-Datenbanken:

- SQL-Datenbanken leiden innerhalb von produktiv Umgebungen unter Performanceproblemen.
- Können bei vielen verschiedenen Objekten (Tabellen) unübersichtlich werden.

Vorteile von SQL-Datenbanken:

- Objekte der objektorientierten Programmierung lassen sich leicht in Tabellen (Objekte) mit Spalten (Attribute) übersetzen.
- Hohe Übersicht bei wenigen Datenobjekttypen.
- Die Entitätsstruktur von Dialogflow lässt sich gleichermaßen in SQL-Tabellen transferieren (Projektrelevant).

5.2.2 NoSQL-Datenbanken (Not only SQL)

NoSQL-Datenbanken basieren nicht auf der Theorie von relationalen Datenbanken, sondern sind für eine besonders hohe Verfügbarkeit und Skalierbarkeit entworfen (Silva, Almeida, und Queiroz 2016). Auf eine tiefgehendere Funktionsbeschreibung wird aus Gründen der Projektrelevanz verzichtet.

Vorteile von NoSQL-Datenbanken:

- Objekte benötigen keine Bearbeitung, sondern können ohne Bearbeitungsaufwand gespeichert werden.
- Wesentlich höhere Performance bei großen Datenmengen (Yassien und Desouky 2016).

Nachteile von NoSQL-Datenbanken:

- Abfrage durch Queries ist in vielen Fällen nicht möglich, somit ist die Programmierung einer weiteren API oftmals erforderlich.
- Keine Optimierung hingehend großer Datenmengen erforderlich, da der Prototyp nur mit Beispieldaten arbeitet (Projektrelevant)

5.2.3 Auswahl der Datenbank

Aufgrund der genannten Vor- und Nachteile in den letzten Abschnitten wäre der Einsatz von sowohl einer SQL- als auch einer NoSQL-Datenbank denkbar, jedoch wurde sich aus den folgenden Gründen für eine SQL-Datenbank entschieden.

1. Vorbereitete Datenbank für das Projekt Rendezfood ist eine SQL-Datenbank.
2. Sehr gute Erfahrung mit SQL-Datenbanken in vorrausgehenden Projekten
3. SQL-Datenbank ist bereits in Komponenten der Testumgebung integriert (MAMP MySQL)

5.2.4 Aufbau der Datenbank

In diesem Abschnitt wird der Aufbau der Prototypen Datenbank aufgezeigt und kurz erläutert. Dieser Aufbau ist relevant für den objektorientierten Ansatz, welcher bei der Implementation des Information Agents gewählt wurde.

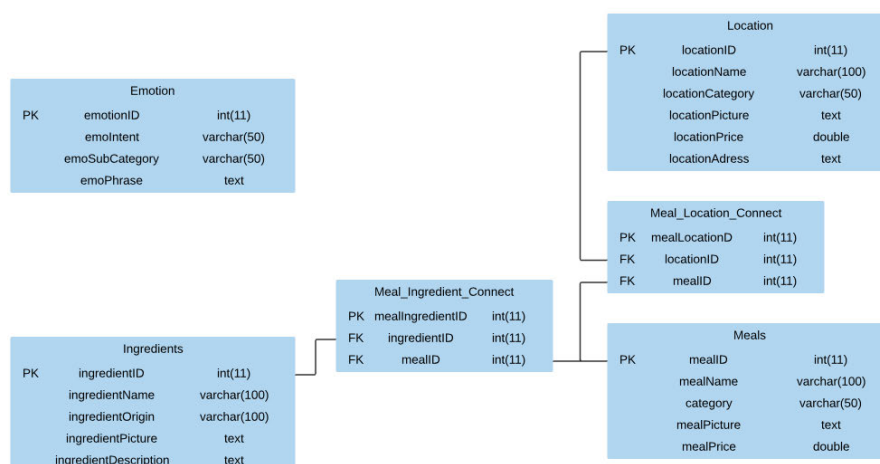


Abbildung 4: ER-Diagramm der Chatbot Datenbank

5.3 Architektur

Für die Architektur von Chatbots werden in der Literatur mehrere Systeme genutzt. Erstens lokale Systeme, welche ohne eine Internetverbindung auskommen. Zweitens netzwerk-basierte Systeme, welche bestimmte Aufgaben auslagern (Mekni, Baani, und Suliman 2020; B. Shawar und Atwell 2004). Aufgrund dessen, dass der Chatbot von Rendezfood ein Teil einer App für Mobileendgeräte darstellen und bei seiner Nutzung Daten aus einer Datenbank beziehen soll, wurde sich bei der Umsetzung für ein netzwerk-basiertes System entschieden.

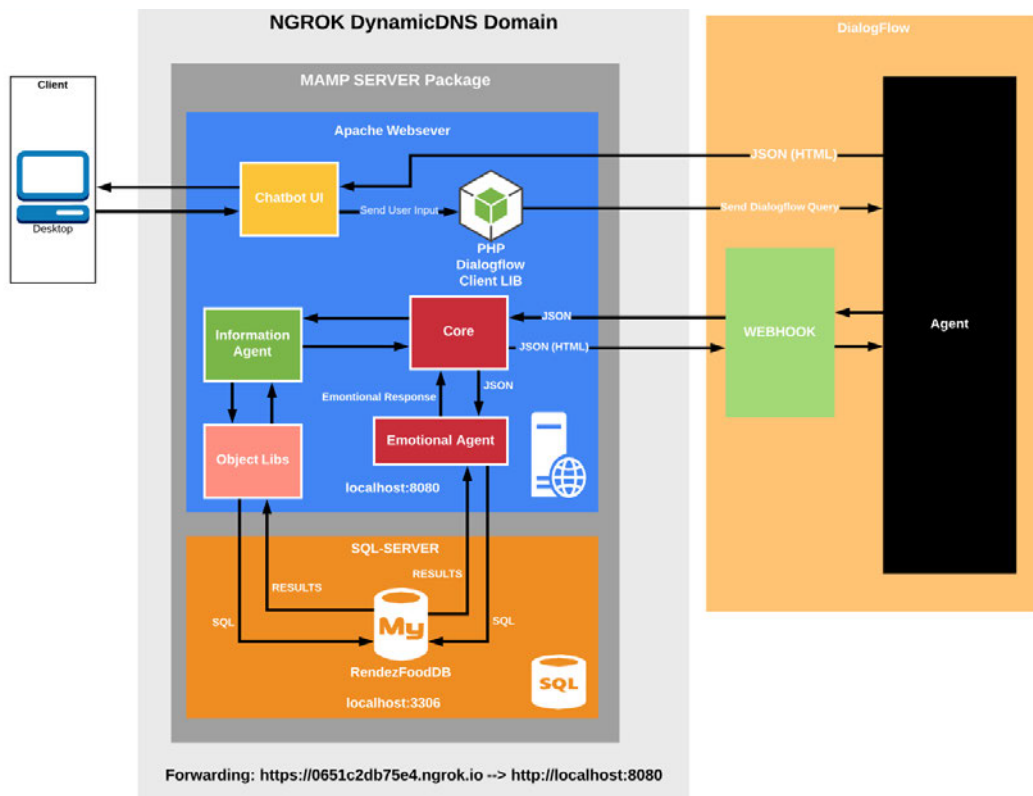


Abbildung 5: Architektur Diagramm der Chatbot Konzeption

5.3.1 User Interface

Das User Interface benötigt der Nutzer, um den Chatbot zu bedienen und gleichzeitig werden hier auch die Antworten des Chatbots dargestellt. Dazu wird eine PHP-Seite, welche sich auf einem Webserver befindet, genutzt. Die PHP-Seite ist neben dem Einlesen von Nutzereingaben und dem Ausgeben der Antworten auch in der Lage die Dialogflow Client Libarys einzubinden und zu verwenden. Mithilfe der Client Libarys kann die PHP-Seite eine Session mit Dialogflow aufbauen, sowie die Nutzeranfrage an die Processing and Network Component (Dialogflow) stellen.

Ist die Antwort vom Kern fertig gestellt worden, wird sie von Dialogflow als Antwort an das User Interface gesendet. Hierbei wird HTML-Code, welcher in eine JSON-Struktur eingebaut wurde, verwendet, sodass die PHP-Seite auch mitgesendete Formatierungs- und Designoptionen anzeigen kann.

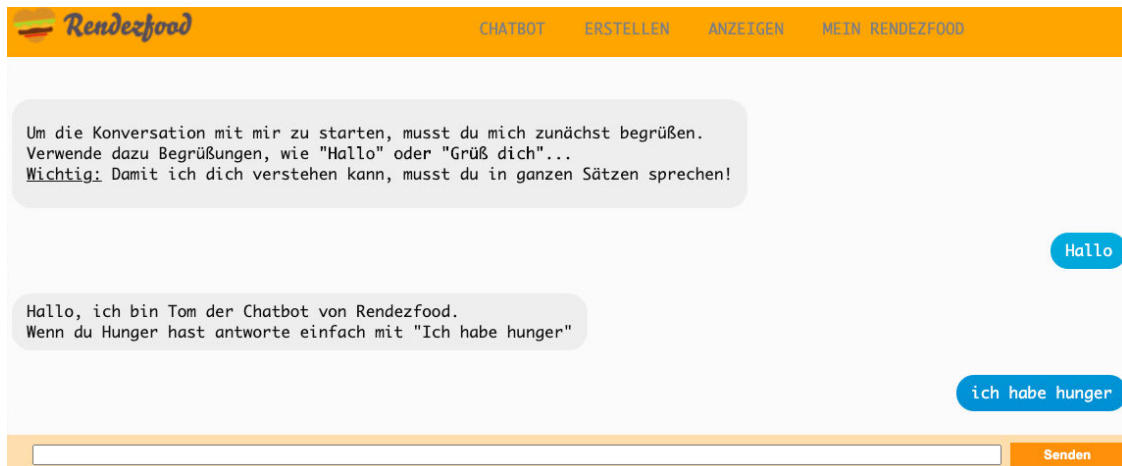


Abbildung 6: Screenshot vom User Interface

5.3.2 Processing and Network Component (Dialogflow)

Dialogflow erhält über die Client Library eine Anfrage, diese wird dann verarbeitet. Zur Verarbeitung gehört das Einordnen der Aussage in einen Kontext, das Erkennen der Nutzerintention und ggf. das Abspeichern und Zuordnen von Parametern. Danach bringt Dialogflow diese Information in eine JSON-Struktur und stellt mittels Webhook eine HTTPS-Request an einen hinterlegten Server. Der hinterlegte Server antwortet auf die Webhook HTTPS-Request mit einer Response, diese sendet Dialogflow, dann als Antwort auf die von der Client Library gestellte Anfrage.

5.3.3 API

Zur Kommunikation hätten in der Theorie eine API oder ein Webservice verwendet werden können, jedoch musste sich für eine API entschieden werden, da Dialogflow beim Senden der Webhooks weder SOAP noch WSDL unterstützt. Die API ist in der Entwicklungsphase ebenfalls auf einem lokal installierten Webserver abgelegt, dessen Pfad Dialogflow zur Verfügung gestellt werden muss.

5.3.3.1 Core

Wie bereits erwähnt besteht die API aus mehreren Komponenten. Zunächst der Core, welcher die Methoden zur Datenveränderung zur Verfügung stellen soll. In der Entwicklungsphase steht jedoch nur die Read-Methode, zum Abfragen von Daten zur Verfügung. Die Aufgabe des Cores ist es die ankommende HTTPS-Request zu verarbeiten und die benötigten weiteren Komponenten zu verwalten. Dazu ließt er das gesendete JSON aus und ordnet die Anfrage einer zur erkannten Nutzerintention passenden Methode zu. In dieser Methode wird dann mithilfe des Information Agents und des Emotional Agents eine Antwort im HTML-Format erstellt und in eine von Dialogflow geforderte JSON-Struktur gebracht. Abschließend wird die erstellte Struktur als Response auf die gestellte Dialogflow Request gesendet.

5.3.3.2 Information Agent

Der Information Agent ist eine Bibliothek, welche sich innerhalb des API Ordners auf dem lokalen Webserver befindet. Diese Bibliothek stellt dem Core die jeweiligen Objekte zur Verfügung, welche die Lese- und Schreibe-Befehle für die Datenbank beinhalten. In der Entwicklungsphase bestand der Information Agent aus den Objekten Ingredient, Meal und Location. Wobei jedes Objekt verschiedene Funktionen zur Verfügung stellt.

```

1 <?php
2 class Meal{
3
4     // database connection and table name
5     private $conn;
6     // constructor with $db as database connection
7     public function __construct($db){
8         $this->conn = $db;
9     }
10
11     // read products
12     function read($category){
13         // select all query
14         $query = "SELECT * FROM Meals WHERE mealMainCategory LIKE ' ".$category." '";
15         // prepare query statement
16         $stmt = $this->conn->prepare($query);
17         // execute query
18         $stmt->execute();
19         return $stmt;
20     }

```

Abbildung 7: Screenshot der objektorientierten Klasse Meal und deren „read“-Funktion

5.3.3.3 Emotional Agent

Der Aufbau des Emotional Agent ist an den des Information Agents angelehnt. Jedoch ist dieser nicht für die Wiedergabe der Daten, wie Speisen und Lokale, sondern für das Verändern und Austauschen der Beschreibungen und Sätze zuständig. Dabei wird der zurück gegebene Wert des Emotional Agents in die HTML-Response des Cores integriert.

5.4 Verarbeitungsbeispiel der Anfrage „Zeig mir Gerichte“

1. Nutzereingabe „Zeig mir Gerichte“ und absenden des Formulars.
2. Das mittels POST-Methode abgesendete Formular wird abgesendet und das User Interface legt eine Session an.
Außerdem wird die Dialogflow Client Library eingebunden, eine Dialogflow Session und eine Request an Dialogflow erstellt.
3. Dialogflow verarbeitet die Anfrage und erkennt die MealsV1-Intention. Daraufhin erstellt Dialogflow eine Webhook Anfrage, welche die Intention (displayName), sowie die bereits gesetzte Speisenkategorie enthält.

```

1 {
2   "responseId": "af33ca2c-60b9-472b-b42b-63445b45862f-83ffff32",
3   "queryResult": {
4     "queryText": "zeig mir Gerichte",
5     "parameters": {},
6     "allRequiredParamsPresent": true,
7     "fulfillmentText": "OK. Hier eine Auswahl an Gerichten die dir gefallen k\u00f6nnten.",
8     "fulfillmentMessages": [
9       {
10        "text": {
11          "text": [
12            "OK. Hier eine Auswahl an Gerichten die dir gefallen k\u00f6nnten."
13          ]
14        }
15      }
16    ],
17    "outputContexts": [
18      {
19        "name": "projects/testagentv1-gxckfu/agent/intents/2d8619be-12fb-4fea-92e1-892a42d195ac",
20        "displayName": "MealV1"
21      }
22    ],
23    "intent": {
24      "name": "projects/testagentv1-gxckfu/agent/intents/2d8619be-12fb-4fea-92e1-892a42d195ac",
25      "displayName": "MealV1"
26    },
27    "intentDetectionConfidence": 1,
28    "languageCode": "de"
29  },
30  "originalDetectIntentRequest": {
31    "payload": {}
32  },
33  "session": "projects/testagentv1-gxckfu/agent/sessions/123456"
34 }

```

Abbildung 8: Teilausschnitt einer Dialogflow Webhook Request an die API

6 Implementation und Limitationen

In diesem Kapitel wird der Prozess der Implementation, sowie dabei aufgetretene Probleme beschrieben. Außerdem werden Limitationen des Programms aufgezeigt.

6.1 Implementation

Während der Implementation des Prototyps sind immer wieder Probleme aufgetreten. Die größte Herausforderung dabei war bei der Vielzahl der Komponenten die Komponente zu ermitteln, welche die Fehlfunktion verursacht.

Zu Beginn der Programmierung musste zunächst einmal die Architektur aufgestellt werden. Dazu wurde eine Test-Intention in Dialogflow erstellt, sowie die Webhook und das Fulfillment aktiviert. Die auftretende Fragestellung war nun, welche Adresse ist bei der Webhook einzutragen, um die lokale Entwicklungsumgebung und den lokalen MAMP-Server zugreifbar zu machen? Die Lösung war NGROK, eine Drittanbieter-Software, welche eine Portweiterleitung auf unsere lokale Umgebung einrichtet.

Eine weitere Herausforderung stellte das Erkennen des Datenverkehrs da, denn durch die Anzahl an Schnittstellen im Programm mussten der Sendeweg der Client Library, der Empfang der Dialogflow Request, der Sendeweg der Response, sowie auch die Response von Dialogflow überwacht und nachvollzogen werden.

Ein zu lösendes Problem lag außerdem in der Einarbeitung in die Dialogflow Client Library, da das Parsen der JSON-Response von der Library einiger Schritte und Überlegungen bedurfte. Insgesamt war der Umgang mit den JSON-Messages im Zusammenhang mit Dialogflow komplizierter, da eine spezielle Struktur eingehalten werden musste und das Encoding nicht frei wählbar war.

6.2 Limitationen des Prototyps

In diesem Abschnitt möchte ich kurz auf die Limitationen im Hinblick auf den Prototyp eingehen. Den aufgrund der oben genannten Herausforderungen konnten nicht alle anfangs geplanten Features umgesetzt werden.

Zunächst ist bei den Limitationen der Emotionale Agent zu nennen, dessen Umfang durchaus als ausbaufähig beschrieben werden kann. Zunächst war es hier geplant auf künstliche Intelligenz zurück zu greifen, jedoch benötigt jede künstliche Intelligenz Trainingsdaten, welche in benötigtem Umfang nicht erhoben werden konnten.

Außerdem war es geplant speziell auf den Nutzer einzugehen, dies hätte jedoch eine sehr aufwendige Veränderung der Datenbank, sowie ein umfangreiches User Interface vorausgesetzt, sodass Nutzer ihre Vorlieben dem System hätten mitteilen können.

Zuletzt ist jedoch zu nennen, dass alle oben genannten Limitationen sich ausschließlich auf den Prototypen beziehen. Der Prototyp ist bewusst Modular gebaut worden, sodass die Anbindung einer künstlichen Intelligenz oder die Erstellung und Verwaltung eines Nutzerprofils ohne große Eingriffe in die Struktur möglich sind. Somit stellt dieser Prototyp ein solides Fundament für das Team von Rendezfood und deren Chatbot-Projekt

7 Zwischenkolloquium und Evaluation nach der Thinking Aloud-Methode

In diesem Kapitel soll das am 22.06.2020 stattgefundenene Zwischenkolloquium mit anschließender Nutzerevaluation beleuchtet werden. Aufgrund des zusätzlichen Implementationsauftrages zu dieser Arbeit wurde sich für ein Zwischenkolloquium entschieden, da so die Möglichkeit zur Überarbeitung des Prototyps besteht.

7.1 Zwischenkolloquium am 22.06.2020 über Zoom

Das Zwischenkolloquium bestand aus einer 20-minütigen Präsentations- und einer 40-minütigen Diskussionsphase. In der Präsentationsphase wurde zunächst der Prozess der Themenfindung, die gesellschaftliche Relevanz von Chatbots, sowie die benötigten wissenschaftlichen und technologischen Grundlagen besprochen. Danach wurden die ersten Erkenntnisse, sowie die Konzeption vorgestellt. Zuletzt wurde, die Evaluationsmethodik und der Evaluationsablauf erläutert. In der Diskussionsphase wurde den Teilnehmern des Zwischenkolloquiums die Möglichkeit gegeben Fragen zu stellen und Kritik an Teilpunkten der Arbeit zu äußern.

Diskussionsphase

Folgend möchte ich die aufgetretenen Kritikpunkte der Diskussionsphase wiedergeben, und die daraus resultierenden Veränderungen aufzeigen.

Ein Kritikpunkt, war die mangelnde Auseinandersetzung mit einem emphatischen Charakterzug bei der Konzeption des Chatbots. Hierbei ging es vor allem um die Aussage, dass ein emphatischer Charakterzug im Anwendungskontext Rendezfood weniger Relevanz hätte, wie im Anwendungskontext der Raucherentwöhnungsstudie.

Tatsächlich scheint Empathie im Umgang mit dem Kunden ein Thema zu sein, so existieren mehrere Bücher, Magazinartikel und Blogbeiträge von Gründern, Gastronomen und Service Experten, welche Customer Empathy und deren positiven Einflüsse beschreiben (Wulf 2018; Rödiger 2017; Wehr 2019). Somit soll im Abschnitt 4.3.1 Empathie von Chatbots der Charakterzug Empathie nochmal genauer beleuchtet werden, sowie empathisches Verhalten ein Feature im Ausblick darstellen.

Ein weiterer Kritikpunkt war die Konzeption der Evaluation, welche in sich nicht konsistent und schlüssig erschien. Zunächst war eine Expertenevaluation geplant, welche die Frage aufwarf, was in diesem Kontext Experten ausmachen würden. Da diese Frage bei der Evaluationskonzeption unbeachtet geblieben war, wurde anstatt dessen eine Nutzerevaluation mit den im Kolloquium Anwesenden durchgeführt.

Weitergehend wurde an dem Aufbau der Evaluation kritisiert, dass eine unbeaufsichtigte Nutzerevaluation geplant war, was eine Abwandlung des Standardvorgehens darstellt und eine tiefere Auseinandersetzung mit der Thinking-Aloud-Methodik fordern würde. Dies würde jedoch das Ziel des Forschungsauftrags verfehlen, somit sei eine einzeln beaufsichtigte, bzw. betreute Nutzerevaluation vorteilhafter.

Zur Auswertung der Evaluation war ein quotenbasiertes System gewählt worden, welches demnach eine Quantitative Auswertung mit sich bringt. Aufgrund der Anzahl der Evaluationsteilnehmer wäre diese jedoch nicht aussagekräftig gewesen. Aus diesem Grund wurde sich für eine Qualitative Auswertungsmethode entschieden. Dies bedeutet konkret, dass die einzelnen beaufsichtigten Nutzerevaluationen über Zoom aufgezeichnet wurden und im nachhinein Kommentare zum Prototyp in Kategorien zusammengefasst werden.

Zuletzt wurde noch nach einer Zielgruppe des Chatbots gefragt, welche kein Bestandteil der Konzeption darstellt. Dies wird jedoch auch weiterhin der Fall sein, da die Weiterentwicklung innerhalb der App von Rendezfood stattfindet und somit die Zielgruppe der App gilt.

7.2 Einzelne Nutzerevaluation am 22.06.2020 über Zoom

Wie bereits im vorherigen Abschnitt angekündigt, wurde der Ablauf und die Form der Evaluation stark diskutiert. Demnach wurde mit jedem Teilnehmer ein betreuter Prototyp-Test durchgeführt, welcher über Zoom aufgezeichnet wurde. Die Evaluationsdauer pro Tester lag zwischen 20 und 30 Minuten und wurde ausschließlich durch folgende Aufgaben eingeschränkt.

1. Informier dich über BBQ Restaurants
2. Lass dir die Gerichte von Toms BBQ Bude zeigen und dich von einem „echten“ Texaner begrüßen.
3. Lasse dir chinesische Gerichte zeigen
4. Finde heraus, was in der Sushi Variation ist und woher der Lachs kommt
5. Lass dir bayrische Restaurants zeigen und dich zum Hofbräuhaus München navigieren

Folgend werde ich die gefallenen Nutzerkommentare tabellarisch darstellen und in Kategorien einordnen. Im Anschluss werde ich das Feedback zu jeder Kategorie abwägen, sowie Handlungsbedarf und Handlungsmöglichkeiten ermitteln.

Kategorie	N1	N2	N3	N4
Position	Chatbot unerfahren	Chatbotentwickler	Projektleiter	Hauptentwickler
Vorstellung	k. A.	Findet Begrüßung ansprechend	Findet Begrüßung ansprechend	k. A.
Verständnis	Findet Sprachvariabilität zu unausgereift	Findet Sprachvariabilität gut, jedoch noch ausbaufähig	k. A.	Findet Sprachvariabilität bereits sehr Ausgereift
Bilder	Bilder sollten interaktiv „klickbar“ gemacht werden	Bilder sollten interaktiv „klickbar“ gemacht werden	Bilder sollten interaktiv „klickbar“ gemacht werden	Bilder sollten interaktiv „klickbar“ gemacht werden
Dialekte	Findet Dialekte nicht notwendig	Sieht Dialekte als angenehme Abwechslung	Findet Dialekte einfach zu verstehen und als angenehme Abwechslung	Findet Dialekte schwer verständlich und als nicht essenziell
Bedienung	Findet die Navigation mithilfe von Eingaben unpraktikabel und wünscht sich Buttons	Erwägt das Nutzen von Buttons bei Entscheidungen	Über Buttons zur Auswahl sollte nachgedacht werden	k. A.

Bedienung der UI	Findet Bedienung der UI durch „ENTER“ gut	Findet Bedienung der UI durch „ENTER“ gut	Findet Bedienung der UI durch „ENTER“ gut	Findet Bedienung der UI durch „ENTER“ gut
Fehlerbehandlung	Wünscht sich detaillierte Vorschläge	Vermisst Hilfe oder Tutorial	Wünscht sich detailliertere Vorschläge zur Nutzung	Vermisst „Rücksprung“ und Hilfe
Weitere Anmerkungen	Würde Chatbot nicht nutzen wollen	Wünscht Informationen zum Funktionsumfang Wünscht sich leichteren Konversationseinstieg	Wünscht sich mehr klickbare Interaktionen	Wünscht sich Smalltalk

Abbildung 6: Tabellarische Darstellung der Nutzeraussagen nach Kategorien

7.3 Zusammenfassung der Nutzeraussagen nach Kategorien

In diesem Abschnitt sollen die Nutzeraussagen nach den Kategorien zusammengefasst und gewichtet werden. Außerdem sollen konkrete Beispiele zur Verbesserung der Kritikpunkte durchdacht und vorgestellt werden.

7.3.1 Begrüßung

Die Nutzer 2 und 3 waren von der Begrüßung als Start der Konversation überzeugt, da dies bei einer realen Konversation ebenfalls der Fall ist. Da dieser Punkt nur zwei von vier Testern positiv aufgefallen ist, könnte man diesen Punkt noch mehr einbringen, um den Anschein eines realen Konversationsstarts zu verstärken. Dies könnte beispielsweise durch immer personalisiertere Begrüßungen nach Nutzungshäufigkeit oder durch Bezugnahme auf vorhergegangene Konversationen umgesetzt werden. Zur Umsetzung der genannten Vorschläge wäre neben den Anpassungen des Chatbots auch die Ergänzung der Nutzer- und Session-Entitäten umzusetzen, welche jedoch den Rahmen dieses Prototyps sprengen würde.

7.3.2 Verständnis

Bei der Variabilität der Sprache gab es verschiedene Ansichten der Nutzer, so nahm Nutzer 1 die Variabilität des Verständnisses als gering, Nutzer 2 und Nutzer 4 die Variabilität als gut bis sehr gut war. Aufgrund der Differenzen in der Einschätzung wird zur Beurteilung dieser Kategorie die Erfahrung der Nutzer in die Gewichtung einbezogen. Nach Einbezug des Erfahrungsgrades lässt sich darauf schließen, dass der Chatbot als Prototyp durchaus eine befriedigende Sprachvariabilität aufweist (Entwickler-Perspektive), diese jedoch für den Unerfahrenen Endnutzer (Stellvertreter ist Nutzer 1) nicht ausreichend ist. Aufgrund der Situation, dass es sich ausschließlich um einen Prototyp handelt, wird die Variabilität nur in kritischen Bereichen (Nutzer gaben vermehrt gleiche Befehle ein, welche nicht verstanden wurden) angepasst.

7.3.3 Bilder

In der Kategorie Bilder waren sich alle Nutzer einig. Der Chatbot zeigte Auswahlmenüs, welche Bilder von Gerichten, Lokalen und Zutaten beinhalteten. Die Nutzer versuchten alle die Bilder anzuklicken um somit eine Auswahl zutreffen. Dadurch, dass das Anklick-Verhalten ausnahmslos bei allen Nutzern zu erkennen war, macht diese Änderung zu einem der wichtigsten Kritikpunkte für die gewohnte Navigation. Somit sollten die Bilder in der zweiten Version des Prototyps klickbar gemacht werden und der Nutzer durch klicken eine Auswahl treffen können.

7.3.4 Dialekte

Bei den Dialekten sind die Nutzer geteilter Meinung. Außerdem lässt sich in dieser Kategorie keine Gewichtung durch einbeziehen der Erfahrung vornehmen. Da jedoch das Dialekt-Feature eines der wesentlichen Merkmale dieses Prototyps und dieser Arbeit darstellt, wird keine Veränderung an dieser Kategorie vorgenommen.

7.3.5 Bedienung

Bei der Bedienung des Chatbots geben die Nutzeraussagen unabhängig von deren Kenntnisstand eine klare Richtung vor. So sollte die Bedienung des Chatbots durch Buttons, ähnlich wie in Punkt 7.3.3 unterstützt werden, so könnte die Nutzung beschleunigt und erleichtert werden. Dieser Kritikpunkt sollte ebenfalls in dem nächsten Prototyp umgesetzt werden. Beispielsweise bei der Entscheidung, ob nach Lokalen oder Gerichten gesucht werden soll, könnte die Auswahl mithilfe eines Buttons einen Vorteil bringen.

7.3.6 Bedienung des User-Interfaces

Die Bedienung des Prototyps mithilfe des eigens entwickelten Nutzerinterfaces kam sehr positiv bei den Nutzern an. Dabei ist besonders, das Nutzen der Eingabetaste („Enter“) zum Senden der Antworten aufgefallen. Hier könnten sicherlich noch weitere Tastenkombinationen, wie „Strg + Z“ als Rücksprung oder die Funktionstasten zur Schnellauswahl, der in 7.3.3 und 7.3.5 angesprochenen Veränderungen integriert werden.

7.3.7 Fehlerbehandlung

Ein sehr wichtiger Kritikpunkt, welcher den Nutzern beim Testen des Prototyps aufgefallen ist, ist die fehlende Fehlerbehandlung. Der Nutzer bekommt bei Falscheingabe lediglich eine Standardantwort, wie beispielsweise „Ich verstehe nicht was du möchtest“. Die Nutzer wünschten sich im Test einheitlich eine allgemeine Nutzungsanleitung, welche sich beispielsweise durch einen Befehl, wie „HILFE“ öffnen lässt. Außerdem wurde der Wunsch nach einer situationsabhängigen Fehlerbehandlung geäußert, welche dem Nutzer Eingabevorschläge anzeigt. Diese Funktion könnte ebenfalls mithilfe von Komponenten aus den Abschnitten 7.3.3, 7.3.5 und 7.3.6 umgesetzt werden.

7.3.8 Weitere Anmerkungen

In der Kategorie weitere Anmerkungen wurden Anmerkungen der Nutzer zum Prototypen zusammengefasst, welche sich nicht in die anderen Kategorien einordnen ließen oder besonders betont wurden. So zeigt die Aussage des Nutzers 1, dass der Prototyp für den Endnutzergebrauch noch weiter ausgebaut werden muss. Nutzer 4 wünscht sich noch mehr Smalltalk mit dem Chatbot, diesem Wunsch kann mit der Aktivierung der Smalltalk-Funktion des Dialog-Frameworks entgegengekommen werden. Die Wünsche des 2. Nutzers spiegeln sich teilweise in der Hilfe-Funktion wieder in welcher der Nutzer über die Funktionen aufgeklärt wird. Jedoch könnte dies bei Release zusätzlich mit einem geführten Tutorial umgesetzt werden. Der besonders betonte Interaktive Umgang mit

Bildern und die Auswahl mithilfe von Buttons, die Nutzer 3 äußert wurde umfänglich beschrieben und somit sollte die Anpassung diesem Wunsch ebenfalls nachkommen.

8 Zusammenfassung und Fazit

In diesem vorletzten Kapitel dieser Bachelorarbeit sollen nochmal alle wichtigen Erkenntnisse der durchgeführten Literaturanalyse und der Forschung dargestellt werden. Zusätzlich stellt die Beantwortung der Forschungsfrage einen Teil dieses Kapitels dar. Außerdem soll ein Abschnitt verwendet werden, um die Limitationen dieser Arbeit aufzuzeigen.

8.1 Forschung

Im Verlauf dieser Arbeit wurden viele verschiedene Chatbots vorgestellt, welche besondere Stärken hatten und auch speziell für eine besondere Aufgabe konzipiert und entwickelt wurden. Jedoch lassen sich einige Gemeinsamkeiten unter den vorgestellten Systemen erkennen. Bei den meisten Systemen handelt es sich um Architekturen mit mehreren Komponenten bzw. Ebenen der Verarbeitung (Calvaresi u. a. 2019; Hettige und Karunananda 2015). Die Systeme der letzten Jahre war außerdem meist netzwerkfähig, um sie von einem Punkt möglichst vielen Nutzern zur Verfügung zu stellen (B. Shawar und Atwell 2004). Orientiert an diesen Erkenntnissen wurde in dieser Arbeit ein Konzept für einen künstlich anthropomorphen Informationsassistenten entworfen und ein Prototyp nach dieser Konzeption entwickelt. Dabei ist aufgefallen, dass die Implementierung eines so komplexen Systems mit einer derartigen Vielzahl an Komponenten, selbst für geübte Entwickler eine Herausforderung darstellen kann.

Um die Implementation als erfolgreich bewerten zu können wird der Prototypen kurz mit den zum Anfang aufgestellten Zielen abgeglichen.

1. Es ist ein netzwerkfähiger Prototyp entstanden, der aufgrund seiner Modularität auch von einer App aus genutzt werden kann.
2. Der Prototyp arbeitet datenbankbasiert.
3. Der Prototyp kann auf Kontexte innerhalb der Konversation mit sprachlichen Eigenheiten reagieren.

Damit sind die aufgestellten Ziele erfüllt. Außerdem ist das Ergebnis der Nutzerevaluation positiv einzuordnen, sodass letztendlich von einer erfolgreichen Implementation gesprochen werden kann.

8.2 Forschungsfrage

Weitergehend ist die Forschungsfrage wie folgt zu beantworten. Es lässt sich schlussfolgern, dass es nicht möglich ist eine allgemeine Empfehlung für die Auswahl der Komponenten abzugeben, sondern bei der Auswahl, der Anwendungszweck eine Hauptrolle spielen sollte. Dennoch ist als Ergebnis dieser Arbeit anzusehen, dass bei der Architektur eines so komplexen Systems eine modulare Architektur gewählt werden sollte. Dies hilft nicht nur bei der Umsetzung der Anforderungen, da jede Komponente unabhängig funktioniert und auch unabhängig getestet werden kann, sondern bietet auch den Aspekt der Skalierbarkeit. Sollte das System überarbeitet werden, so kann dies Schritt für Schritt erfolgen, ohne in den Betrieb eingreifen zu müssen. Des Weiteren bietet eine modulare Architektur den Vorteil, eine Komponente, wie die zur Anthropomorphisierung auf künstliche Intelligenz umzustellen oder auch komplett neu zu überdenken.

Eine weitere Erkenntnis ist es, auch bei nicht-netzwerkfähigen Systemen mit einer API zu arbeiten, um wie bereits beschrieben den Schritt zur Netzwerkfähigkeit leichter angehen zu können.

Ebenso ist es sinnvoll den Chatbot individuell und persönlich auf den Nutzer eingehen zu lassen, denn wie in Abschnitt 4.3.1 beschrieben, kann dies nicht nur die Nutzeraufmerksamkeit, sondern auch die Gesprächsqualität ansteigen lassen. Aus diesem Grund ist es sehr empfehlenswert eine datenbankbasierte Architektur zu verwenden, da das Abspeichern der Nutzerinformationen mit einer Datenbank leichter ist.

8.3 Limitationen

Zuletzt soll auf die Limitationen dieser Arbeit eingegangen werden, um im Anschluss einen Ausblick auf die weitere Forschung an Chatbots geben zu können

Als eine Limitation ist zu nennen, dass es in dieser Arbeit nicht gelungen ist die gesamte Fachliteratur zum Thema anthropomorphe Chatbots einzubeziehen, da es den Rahmen einer Bachelorarbeit gesprengt hätte. Außerdem ist zu beachten, dass sich die Forschung in dieser Arbeit stark auf einen Prototyp beschränkt hat, sodass keine allgemeingültigen Empfehlungen gegeben werden können.

Wünschenswert wäre es außerdem gewesen, wenn bei der Nutzerevaluation noch mehr Daten hätten gesammelt werden können, beispielsweise durch eine Erhöhung der Gruppengröße oder durch eine umfangreichere Aufgabenstellung.

Im Nachhinein betrachtet, wäre ein abweichender Projektstart sinnvoll gewesen. Mithilfe einer Probandenbefragung, welche Features ein Chatbot in diesem Anwendungskontext aufweisen sollte, hätte man die benötigten Features besser abgrenzen können. Im Anschluss hätte man den Chatbot konzipiert und implementiert. Zum Schluss hätte man den Chatbot mit der befragten Gruppe und einer neutralen Gruppe evaluieren können. Durch dieses Vorgehen wäre der Erfolg der Implementation besser beweisbar gewesen.

9 Ausblick

Im Ausblick möchte ich weiteren Forschungsbedarf in diesem Themengebiet aufzeigen, sowie auf die Relevanz meiner Ergebnisse eingehen.

Die Ergebnisse dieser exemplarischen Untersuchung sollen dem untersuchten Projekt als Vorlage dienen, sodass die Erkenntnisse in deren Konzeption übernommen werden. Für andere in dieser Arbeit nicht untersuchte Projekte, kann diese Bachelorarbeit dazu dienen, als Beispiel-Architektur in diesem Anwendungskontext gesehen zu werden.

Ein Forschungsbedarf besteht meiner Meinung nach vor allem im Bereich der Anthropomorphisierung und wie dieses Feature konkret in Chatbots umgesetzt werden kann. Zudem ist diese Beispiel Architektur mithilfe weiterer exemplarischer Untersuchungen zu belegen und weitergehend durch Austausch der beschriebenen Komponenten zu verifizieren.

Zuletzt könnte es sinnvoll sein, die Architektur in anderen Anwendungskontexten von Chatbots zu testen, Beispiele dafür könnten die Suchberatung oder ein regelchatbot für kompliziertere Brettspiele sein.

Quellenverzeichnis

- „AISB - The Society for the Study of Artificial Intelligence and Simulation of Behaviour - Loebner Prize“. 2015. 19. Oktober 2015. <https://web.archive.org/web/20151019030117/http://www.aisb.org.uk/events/loebner-prize#contest2014>.
- Angelov, Stefan, und Milena Lazarova. 2019. „E-Commerce Distributed Chatbot System“. In *Proceedings of the 9th Balkan Conference on Informatics*, 1–8. Sofia Bulgaria: ACM. <https://doi.org/10.1145/3351556.3351587>.
- Asthana, Siddhartha, Pushpendra Singh, und Amarjeet Singh. 2013. „Design and Evaluation of Adaptive Interfaces for IVR Systems“. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems on - CHI EA '13*, 1713. Paris, France: ACM Press. <https://doi.org/10.1145/2468356.2468663>.
- Berners-lee, Tim, Robert Cailliau, und Jean-françois Groff. 1992. „World-wide web: the information universe“. *Electron. Network*.
- Brandtzaeg, Petter Bae, und Asbjørn Følstad. 2018. „Chatbots: Changing User Needs and Motivations“. *Interactions* 25 (5): 38–43. <https://doi.org/10.1145/3236669>.
- Bush, Thomas. 2019. „What Is The Difference Between Web Services and APIs? | Nordic APIs |“. *Nordic APIs*. 16. Juli 2019. <https://nordicapis.com/what-is-the-difference-between-web-services-and-apis/>.
- Calvaresi, Davide, Jean-Paul Calbimonte, Fabien Debusson, Amro Najjar, und Michael Schumacher. 2019. „Social Network Chatbots for Smoking Cessation: Agent and Multi-Agent Frameworks“. In *IEEE/WIC/ACM International Conference on Web Intelligence on - WI '19*, 286–92. Thessaloniki, Greece: ACM Press. <https://doi.org/10.1145/3350546.3352532>.
- Dalton, Jeffrey, Victor Ajayi, und Richard Main. 2018. „Vote Goat: Conversational Movie Recommendation“. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval - SIGIR '18*, 1285–88. Ann Arbor, MI, USA: ACM Press. <https://doi.org/10.1145/3209978.3210168>.
- „Dialogflow“. o. J. Dialogflow. Zugegriffen 30. Mai 2020. <https://dialogflow.com/>.
- Dudenredaktion (Hrsg.). o. J. „Duden | Anthropomorphismus | Rechtschreibung, Bedeutung, Definition, Herkunft“. Zugegriffen 30. Mai 2020a. <https://www.duden.de/rechtschreibung/Anthropomorphismus>.
- . o. J. „Duden | künstlich | Rechtschreibung, Bedeutung, Definition, Herkunft“. Zugegriffen 9. Mai 2020b. <https://www.duden.de/rechtschreibung/kuenstlich>.
- Ed-douibi, Hamza, Javier Luis Cánovas Izquierdo, Abel Gómez, Massimo Tisi, und Jordi Cabot. 2016. „EMF-REST: Generation of RESTful APIs from Models“. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing - SAC '16*, 1446–53. Pisa, Italy: ACM Press. <https://doi.org/10.1145/2851613.2851782>.
- Fielding, Roy Thomas. 1999. „HTTP/1.1: Introduction“. 1999. <https://www.w3.org/Protocols/rfc2616/rfc2616-sec1.html#sec1>.
- . 2000a. „Architectural styles and the design of network-based software architectures“. Phd, University of California, Irvine.
- . 2000b. „Fielding Dissertation: CHAPTER 5: Representational State Transfer (REST)“. 2000. https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm#sec_5_1.
- Følstad, Asbjørn, Petter Bae Brandtzaeg, Tom Feltwell, Effie L-C. Law, Manfred Tscheligi, und Ewa A. Luger. 2018. „SIG: Chatbots for Social Good“. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, 1–4. Montreal QC Canada: ACM. <https://doi.org/10.1145/3170427.3185372>.
- Følstad, Asbjørn, und Marita Skjuve. 2019. „Chatbots for Customer Service: User Experience and Motivation“. In *Proceedings of the 1st International Conference on*

- Conversational User Interfaces - CUI '19*, 1–9. Dublin, Ireland: ACM Press. <https://doi.org/10.1145/3342775.3342784>.
- Funke, Joachim, und Bianca Vaterrodt. 2009. *Was ist Intelligenz?* C.H.Beck.
- Hettige, B, und AS Karunananda. 2015. „Octopus: A Multi Agent Chatbot“, 7.
- Holzinger, Andreas. 2006. „Thinking-aloud eine Königsmethode im Usability Engineering“. 2006. [http://user.medunigraz.at/andreas.holzinger/holzinger%20de/papers%20de/G49_HOLZINGER%20\(2006\)%20Thinking%20aloud.pdf](http://user.medunigraz.at/andreas.holzinger/holzinger%20de/papers%20de/G49_HOLZINGER%20(2006)%20Thinking%20aloud.pdf).
- Hutchens, Jason L., und Michael D. Alder. 1998. „Introducing MegaHAL“. In *Proceedings of the Joint Conferences on New Methods in Language Processing and Computational Natural Language Learning - NeMLaP3/CoNLL '98*, 271. Sydney, Australia: Association for Computational Linguistics. <https://doi.org/10.3115/1603899.1603945>.
- Khan, Rashid, und Abu Das. 2018. *Build Better Chatbots*. Apress.
- Krishnan, Subramanian, Varun Pal, und Bharathi Venkatasubramanian. 2018. Generic and configurable technique for webhook validation with arbitrary applications. United States US20180115595A1, filed 26. Oktober 2016, und issued 26. April 2018. <https://patents.google.com/patent/US20180115595A1/en>.
- Leah. 2019. „Die 7 besten Chatbots 2019“. 26. April 2019. <https://www.user-like.com/de/blog/die-besten-chatbots>.
- Lee, Henry. 2018. *Voice User Interface Projects: Build Voice-Enabled Applications Using Dialogflow for Google Home and Alexa Skills Kit for Amazon Echo*. Packt Publishing Ltd.
- Loebner, Hugh. 2009a. „Joan | Loebner Prize 2006“. Chatbots.Org. 5. Juli 2009. https://www.chatbots.org/awards/winner_nomination/joan/.
- . 2009b. „Loebner Prize 2000“. Chatbots.Org. 16. Juli 2009. https://www.chatbots.org/awards/loebner_prize/year_2000/.
- . 2009c. „Loebner Prize 2010“. Chatbots.Org. 13. Oktober 2009. https://www.chatbots.org/awards/loebner_prize/year_2010/.
- Lotze, Netaya. 2016. „Chatbots: Eine linguistische Analyse - 1002758.pdf“. 2016. <https://library.oapen.org/viewer/web/viewer.html?file=/bitstream/handle/20.500.12657/27255/1002758.pdf?sequence=1&isAllowed=y>.
- Lun, Erwin van. 1998. „Albert One | 1998“. Chatbots.Org. 1998. https://www.chatbots.org/awards/winner_nomination/albert_one/.
- Mauldin, Michael L. 1994. „ChatterBots, TinyMuds, and the Turing Test: Entering the Loebner Prize Competition“, 6.
- Mekni, Mehdi, Zakaria Baani, und Dalia Sulieman. 2020. „A Smart Virtual Assistant for Students“. In *Proceedings of the 3rd International Conference on Applications of Intelligent Systems*, 1–6. Las Palmas de Gran Canaria Spain: ACM. <https://doi.org/10.1145/3378184.3378199>.
- Minnaert, Gary, Ayanna Sawyer, und Alexander Thayer. 2002. Internet-based application program interface (API) documentation interface. United States US6405216B1, filed 17. September 1999, und issued 11. Juni 2002. <https://patents.google.com/patent/US6405216B1/en>.
- Mogul, Jeffery C. 2002. „Clarifying the fundamentals of HTTP“. In *Proceedings of the 11th international conference on World Wide Web*, 25–36. WWW '02. Honolulu, Hawaii, USA: Association for Computing Machinery. <https://doi.org/10.1145/511446.511450>.
- Müller, Bertie. 2019. „Mitsuku Wins 2019 Loebner Prize and Best Overall Chatbot at AISB X“. AISB – The Society for the Study of Artificial Intelligence and Simulation of Behaviour. 15. September 2019. https://aisb.org.uk/new_site/?p=350.

- Myrach, Thomas, und Sara Margarita Zwahlen. 2008. *Virtuelle Welten?: die Realität des Internets*. Peter Lang.
- Prospero, Adam Di, Nojan Norouzi, Marios Fokaefs, und Marin Litoiu. 2017. „Chatbots as Assistants: An Architectural Framework“, 11.
- Radziwill, Nicole, und Morgan Benton. 2017. „Evaluating Quality of Chatbots and Intelligent Conversational Agents“, Januar, 21.
- Rautmare, Sharvari, und D. M. Bhalerao. 2016. „MySQL and NoSQL database comparison for IoT application“. In *2016 IEEE International Conference on Advances in Computer Applications (ICACA)*, 235–38. <https://doi.org/10.1109/ICACA.2016.7887957>.
- Riccardi, Giuseppe, Philipp Cimiano, Alexandros Potamianos, und Christina Unger. 2012. „Up from Limited Dialog Systems!“, Januar, 2.
- Rödiger, Kai. 2017. *Empathie im Kundenkontakt: Einflussfaktoren, Erfolgsauswirkungen und Altersunterschiede als Kontextfaktoren*. Springer-Verlag.
- Rooein, Donya. 2019. „Data-Driven Edu Chatbots“. In *Companion Proceedings of The 2019 World Wide Web Conference on - WWW '19*, 46–49. San Francisco, USA: ACM Press. <https://doi.org/10.1145/3308560.3314191>.
- Schrader, Thomas J., und Christian Haider. 2009. „Complex Values in Smalltalk“. In *Proceedings of the International Workshop on Smalltalk Technologies - IWST '09*, 126. Brest, France: ACM Press. <https://doi.org/10.1145/1735935.1735957>.
- Shawar, Bayan Abu, und Eric Atwell. 2007a. „Chatbots: Are They Really Useful?“, 21.
- . 2007b. „Chatbots: Are They Really Useful?“, 21.
- . 2007c. „Different measurements metrics to evaluate a chatbot system“. In *Proceedings of the Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technologies*, 89–96. NAACL-HLT-Dialog '07. Rochester, New York: Association for Computational Linguistics.
- Shawar, Bayan, und Eric Atwell. 2004. *A Comparison Between Alice and Elizabeth Chatbot Systems*. University of Leeds, School of Computing research report 2002.19.
- Silva, Yasin N., Isadora Almeida, und Michell Queiroz. 2016. „SQL: From Traditional Databases to Big Data“. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education - SIGCSE '16*, 413–18. Memphis, Tennessee, USA: ACM Press. <https://doi.org/10.1145/2839509.2844560>.
- Spertus, Ellen, und Lynn Andrea Stein. 2000. „Squeal: A Structured Query Language for the Web“. *Computer Networks* 33 (1): 95–103. [https://doi.org/10.1016/S1389-1286\(00\)00074-8](https://doi.org/10.1016/S1389-1286(00)00074-8).
- Steiner, Anna. 2016. „Künstliche Intelligenz: Zum Nazi und Sexisten in 24 Stunden“. *FAZ.NET*, 24. März 2016. <https://www.faz.net/1.4144019>.
- Tenzer. 2019. „Statistiken zum Thema Smart Speakers“. 6. Dezember 2019. <https://de.statista.com/themen/4662/smart-speakers/>.
- Tenzer, F. 2020. „Anzahl der Smartphone-Nutzer in Deutschland in den Jahren 2009 bis 2019“. 1. April 2020. <https://de.statista.com/statistik/daten/studie/198959/umfrage/anzahl-der-smartphonenuutzer-in-deutschland-seit-2010/>.
- Verduijn, Xander. o. J. „Chatbot Albert One, Robby Garner | Virtual Assistant Albert One | Virtual Agent Albert One | Chat Bot Albert One | Conversational Agent Albert One | (4537)“. Chatbots.Org. Zugegriffen 25. Mai 2020. https://www.chatbots.org/chatbot/albert_one/.
- W3C. 2004. „Web Services Architecture“. 2004. <https://www.w3.org/TR/ws-arch/#whatis>.
- Wehr, Andre`. 2019. „Customer Empathy: Einfühlungsvermögen mit Kunden entscheidet“. *tractionwise* (blog). 4. Januar 2019. <https://www.tractionwise.com/magazine/customer-empathy-einfuehlungsvermoegen-kunde/>.

- Weintraub, Joseph. 1999. „HISTORY OF THE PC THERAPIST“, März, 3.
- Weizenbaum, Joseph. 1966. „ELIZA—a Computer Program for the Study of Natural Language Communication between Man and Machine“, Januar, 10.
- Wulf, Jan-Peter. 2018. „Empathie, Achtsamkeit und Co.: 5 Tipps für die Gastronomie im Jahr 2018“. nomy. 26. Januar 2018. <https://www.nomyblog.de/nomyblog/gastronomie/tipps-gastronomie-2018/>.
- Yassien, Amal W., und Amr F. Desouky. 2016. „RDBMS, NoSQL, Hadoop: A Performance-Based Empirical Analysis“. In *Proceedings of the 2nd Africa and Middle East Conference on Software Engineering*, 52–59. AMECSE '16. Cairo, Egypt: Association for Computing Machinery. <https://doi.org/10.1145/2944165.2944174>.
- Zamora, Jennifer. 2017. „Rise of the Chatbots: Finding A Place for Artificial Intelligence in India and US“. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces Companion - IUI '17 Companion*, 109–12. Limassol, Cyprus: ACM Press. <https://doi.org/10.1145/3030024.3040201>.