

# Context Handling in schriftlicher Mensch-Computer-Interaktion

Antonia C. Haslinger



MASTERARBEIT

eingereicht am  
Fachhochschul-Masterstudiengang

Interactive Media

in Hagenberg

im September 2018

© Copyright 2018 Antonia C. Haslinger

Diese Arbeit wird unter den Bedingungen der Creative Commons Lizenz *Attribution-NonCommercial-NoDerivatives 4.0 International* (CC BY-NC-ND 4.0) veröffentlicht – siehe <https://creativecommons.org/licenses/by-nc-nd/4.0/>.

# Erklärung

Ich erkläre eidesstattlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benutzt und die den benutzten Quellen entnommenen Stellen als solche gekennzeichnet habe. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt.

Hagenberg, am 25. September 2018

Antonia C. Haslinger

# Inhaltsverzeichnis

<b>Erklärung</b>	iii
<b>Vorwort</b>	vii
<b>Kurzfassung</b>	viii
<b>Abstract</b>	ix
<b>1 Einleitung</b>	1
1.1 Problemstellung . . . . .	1
1.2 Fragestellung und Zielsetzung . . . . .	1
1.3 Methodik und Aufbau . . . . .	2
<b>2 Chatbots</b>	3
2.1 Definition . . . . .	3
2.2 Geschichte der Chatbots . . . . .	4
2.2.1 Alan Turing . . . . .	5
2.2.2 Turing Test . . . . .	5
2.2.3 Eliza . . . . .	7
2.2.4 Alice . . . . .	8
2.2.5 Heutige Sprachassistenten . . . . .	8
2.3 Kategorien . . . . .	9
2.3.1 Arten . . . . .	9
2.3.2 Use-Cases . . . . .	11
2.4 Plattformen . . . . .	13
2.4.1 Facebook Messenger . . . . .	13
2.4.2 Slack . . . . .	13
2.4.3 Alexa . . . . .	14
2.4.4 Google Assistant . . . . .	15
2.4.5 Weitere . . . . .	15
2.5 Rolle in der Web-Kommunikation . . . . .	15
2.5.1 Kundenanfragen . . . . .	15
2.5.2 Werbung und E-Commerce . . . . .	16
2.5.3 Risiko . . . . .	17

<b>3 Kontext</b>	18
3.1 Definition	18
3.2 Bedeutsamkeit von Kontext	18
3.3 Relevante Aspekte der technischen Entwicklung	19
3.3.1 Variablen	19
3.3.2 Lebensspanne	20
3.3.3 Pronomen	20
3.3.4 Gesprächsübergreifendes Erinnerungsvermögen	22
3.3.5 Context Aware Computing	22
3.4 Schwierigkeiten	22
3.4.1 Bot Amnesia	22
3.4.2 Unerwarteter Input	24
3.4.3 Fehlerhafte Auffassung	25
3.4.4 Unzureichende Verarbeitung	26
3.4.5 Verbindung von unterschiedlichen Kontexten	27
3.5 Kommunikative Maßnahmen	27
3.5.1 Angebote	28
3.5.2 Rich Messages	29
3.5.3 Korrekturen	30
3.5.4 Opt-In	31
3.6 Kontext-Verarbeitung und Datenschutz	33
3.6.1 Relevante Artikel	34
3.6.2 Umsetzung	35
<b>4 Projekt</b>	36
4.1 Aufgabenstellung	36
4.2 Umsetzung	36
4.2.1 Dialogflow	36
4.2.2 Webhook	39
4.2.3 Dynamische Features	40
4.2.4 Facebook Messenger	43
4.3 Kontext	43
4.3.1 Input-Variablen	43
4.3.2 Output-Variablen	44
4.3.3 Lebensspanne der Variablen	44
4.3.4 Verwendung	44
4.4 Schwierigkeiten in der Entwicklung	44
4.4.1 Einschränkung der Speicherkapazitäten	45
4.4.2 Verwendung von Benutzereingaben	45
4.4.3 Löschen der Variablen	46
4.5 Ergebnis	46
4.5.1 Abgedeckte Themen	47
4.5.2 Studiengänge	47
4.5.3 Rich Messages	48
4.5.4 Einbindung des Kalenders	48
4.5.5 Google Maps API	49

Inhaltsverzeichnis	vi
<b>5 Diskussion und Fazit</b>	<b>50</b>
5.1 Beantwortung der Forschungsfrage . . . . .	51
5.2 Ausblick . . . . .	51
5.2.1 Erweiterung der Arbeit . . . . .	51
5.2.2 Weiterführung des Projektes . . . . .	52
<b>A Inhalt der CD-ROM</b>	<b>53</b>
A.1 Hauptverzeichnis . . . . .	53
A.2 Medien . . . . .	53
A.3 Projekt . . . . .	54
A.4 Sicherung Online-Referenzen . . . . .	54
<b>Quellenverzeichnis</b>	<b>55</b>
Literatur . . . . .	55
Online-Quellen . . . . .	55

# Vorwort

Seit Beginn meines Studiums beschäftige ich mich mit der Thematik *Künstliche Intelligenz*. Meine Bachelorarbeit habe ich ebenfalls diesem Thema gewidmet und die Freude an dieser Arbeit hat dazu geführt, dass ich in meinem Master-Studium das selbe Ziel verfolgt habe. Somit ist diese Arbeit entstanden, die sich speziell mit dem Thema *Kontextverarbeitung und deren Bedeutung bei Bots* auseinandersetzt. Ebenfalls Teil dieser Veröffentlichung ist die Beschreibung meines korrespondierenden Projektes, bei dem ich das Grundgerüst eines Chatbots für meine Universität entwickelte. Zusammen dienen sie als Bedingung für meinen positiven Abschluss des Master-Studiengangs *Interactive Media* an der *Fachhochschule Oberösterreich Campus Hagenberg*. Der Zeitraum für Recherche, Entwicklung des Projektes und Verfassen der Arbeit erstreckt sich von August 2017 bis zur Abgabe im September 2018.

Ich möchte meinem Betreuer FH-Prof. Mag. DI Dr. Andreas Stöckl für die Unterstützung bei der Entwicklung des Projektes und dem Verfassen dieser Arbeit danken. Ebenso gilt meine Dankbarkeit der gesamten Dozentenschaft der Fachhochschule, ohne die das Erlernen dieser umfangreichen Fähigkeiten nicht möglich gewesen wäre. Ein Dankeschön auch an meine Familie, die mir das Studium grundsätzlich ermöglicht hat und gemeinsam mit meinen Freunden darauf achtete, mir mental zur Seite zu stehen. Außerordentlichen Dank an Stefan Dumberger, der mich tatkräftig bei dem Projekt und der Arbeit unterstützt hat und meinen Kater Wuschel, der mir mit seinem Schnurren durch einige Blockaden geholfen hat.

Mein herzlichster Dank ist jedoch an meinen jahrelangen Kollegen und Freund Michael Staudinger zu richten. Ohne seinen unermüdlichen Einsatz und Zuwendung wäre ich nicht an das Ende des Studiums und somit an den resultierenden Titel gelangt. Ich empfinde unbeschreibliche Dankbarkeit für die Gestaltung der letzten fünf Jahre in denen wir tagtäglich miteinander gearbeitet, gelernt und gelacht haben. Danke, dass du mir eine erfolgreiche Zukunft mit unzähligen schönen Erinnerungen ermöglicht hast.

Ich hoffe Sie erfreuen sich am Lesen dieser Arbeit und können wichtige Aspekte für Ihre Vorhaben nutzen.

Hagenberg, am 25. September 2018

Antonia Christina Haslinger

# Kurzfassung

Die vorliegende wissenschaftliche Arbeit befasst sich mit technischen und kommunikativen Aspekten zur Entwicklung und Optimierung der Kontext-Verarbeitung bei Chatbots. Die reibungslose Ausführung bereitet den Entwicklern weiterhin Schwierigkeiten, doch deren Optimierung spielt eine maßgebliche Rolle beim Erfolg des Systems, da negative Erfahrungen mit dem Bot die Einstellung des Benutzers gegenüber der jeweiligen repräsentierten Marke negativ beeinflussen können.

Das Ziel dieser Arbeit ist es, einen umfangreichen Einblick in alle Aspekte in Bezug auf Kontext-Verarbeitung bei Chatbots zu ermöglichen. Es wird ein allgemeiner Überblick zu diesen Systemen gegeben und die jeweils relevanten Aspekte in Bezug auf Kontext angeführt. Der Hauptteil der Arbeit besteht aus Erläuterungen und Beispielen zu allen wesentlichen Bestandteilen der technischen und kommunikativen Kontext-Verarbeitung. Abgerundet wird diese Arbeit mit dem Einblick in das korrespondierende Master-Projekt, welches sich mit der Entwicklung eines Chatbots für die *Fachhochschule Oberösterreich Campus Hagenberg* auseinandersetzt. Das gesamte Wissen wurde aus Literaturanalyse, Recherche, Tests von bestehenden Systemen und Erfahrungen aus dem korrespondierenden Projekt zusammengetragen.

Das Ergebnis dieser Analyse der relevanten Bestandteile, Erläuterungen und Beispiele ist, dass die Definition der benötigten Aspekte zwar ausreichend, doch die technische Umsetzung durch steigende Anforderungen und Erwartungen von Benutzern und Auftraggebern noch nicht umfangreich und fehlerfrei möglich ist.



# Abstract

This master thesis investigates technical and communicative aspects of the development and optimization of context-handling in chatbot-systems. It is still hard for developers to reach the ideal benefits, though they are essential for success of the system. Serious problems during conversation can lead to negative experiences and subsequently to unfavorable opinions of the particular brand.

The goal of this thesis is to provide a comprehensive insight of all aspects of context-handling in chatbot-systems. There will be an overview of these systems and appreciable aspects regarding context are discussed within these points. The main part of this thesis contains explanations and examples of all relevant components of technical and communicative context-handling methods. The corresponding project of this work is presented afterwards. Its main purpose was to develop a chatbot for the *University of Applied Sciences Upper-Austria Campus Hagenberg*. The overall knowledge was collected from analysis of literature, research, testing of existing systems and experiences gained during the development of the project.

The result of this analysis of relevant components, explanations and examples is the understanding and estimation of complexity, difficulty and challenges of the technical implementations. There will be more struggles anytime soon due to rising expectations and requests of users and clients, although there are clear definitions of required aspects to implement and take care of.

# Kapitel 1

## Einleitung

### 1.1 Problemstellung

Bei einer Befragung des Technologie-Anbieters *Oracle* im Jahre 2016 gaben 80% der befragten Unternehmen an, dass sie planen, den Kundenservice in spätestens vier Jahren durch Chatbots zu ersetzen [16]. Dieses Ergebnis deutet darauf hin, dass die Beliebtheit von Chatbots und somit die Anforderungen und die erwartete Qualität weiterhin steigen werden. Schwierigkeiten während des Gesprächs, die sich in letzter Instanz auch auf das *Image* des Unternehmens auswirken können, bringen Entwickler und Designer dazu, sich intensiv mit den Feinheiten der Konversations-Gestaltung auseinander zu setzen. Diese beinhaltet unter anderen den wichtigen Aspekt der Kontext-Verarbeitung, die maßgeblich zur Qualität des Systems beitragen kann, wenn die Entwickler technische und kommunikative Maßnahmen treffen.

### 1.2 Fragestellung und Zielsetzung

Die Forschungsfrage der vorliegenden Masterarbeit lautet:

Welche Möglichkeiten stehen den Entwicklern von Chatbots zur Verfügung, um einen optimalen Konversationsverlauf in Bezug auf Kontext zu gewährleisten und welche technischen und kommunikativen Aspekte sind dabei zu beachten?

Der Begriff *optimal* sagt in diesem Zusammenhang aus, dass der Bot eine stabile Konversation führen kann, ohne dass der Benutzer auf Unannehmlichkeiten wie Orientierungsverlust des Chatbots oder *Out-Of-Context*-Situationen stößt, mit dem Ziel einen positiven Eindruck der beworbenen Marke zu erreichen.

Zusätzlich soll im Rahmen dieser Arbeit eine Reihe an untergeordneten Fragen beantwortet werden:

- Was ist Kontext und warum ist dieser bei schriftlichen Konversationen von Bedeutung?
- Wie sah die Entwicklung von Chatbots, besonders in Bezug auf Kontext, aus?
- Welche Arten von Bots gibt es und was ist bezüglich Kontext bei ihnen zu beachten?

- Welche Problematiken können entstehen und welche Ansätze zur Lösung gibt es?
- Wie und in welchem Umfang sollten die vorgestellten Methoden verwendet werden?
- Welche Plattformen stehen für die Veröffentlichung eines Chatbots zur Verfügung?
- Welche Rolle spielen Bots beim *E-Commerce* und warum ist gute kontextuelle Verarbeitung dabei von Bedeutung?

Das Ziel dieser Arbeit ist, einen Überblick zur Entwicklung der Chatbots und deren Kontext-Verarbeitung, Arten und Use-Cases der Systeme, verfügbare Plattformen zur Veröffentlichung und Aspekte in Bezug auf *E-Commerce* zu geben. Vertieft soll auf die gesamte Thematik Kontext in schriftlicher Konversation eingegangen werden. Dies inkludiert Definition, technische Aspekte, kommunikative Maßnahmen, auftretende Schwierigkeiten und Anmerkungen zum Datenschutz.

### 1.3 Methodik und Aufbau

Das verfasste Wissen stammt aus Literatur, in der Entwicklung des Projektes entstandene Erfahrungen und Analyse von bereits bestehenden Systemen wie *Google Assistant* und *Mitsuku*. Zur Verdeutlichung einiger Aspekte wird ein imaginäres Konstrukt eines Bots vorgestellt und in Abbildungen und Erläuterungen verwendet.

Die vorliegende Arbeit gliedert sich in fünf Abschnitte. Nach dieser Einleitung mit Erläuterung der Problemstellung, Festlegung der Forschungsfrage, Definition des Ziels und Anmerkungen zur Methodik, folgt im Kapitel 2 eine Übersicht zum Thema Chatbots. Diese enthält Definitionen des Begriffs an sich, die Entwicklung verschiedener Systeme seit dem Jahr 1966, die verschiedenen Kategorien von Systemen, verfügbare Plattformen für die Veröffentlichung und die Rolle von Chatbots in der Web-Kommunikation. Die dazu wesentlichen Aspekte in Bezug auf Kontext werden jeweils angegeben. Ebenso ist ein kurzer Abschnitt dem Wissenschaftler *Alan Turing* und seinem Konstrukt des *Turing Tests* gewidmet.

In Kapitel 3 wird auf alle Punkte eingegangen, die die Verarbeitung und Verwendung von Kontext in Chatbot-Systemen betreffen. Dieses gliedert sich zuerst in Definition des Begriffes *Kontext* und die Bedeutsamkeit davon in schriftlichen Konversationen. Danach wird auf relevante Aspekte in der technischen Entwicklung, dennoch auftretende Schwierigkeiten und weitere kommunikative Maßnahmen zur Verbesserung eingegangen. Ebenso wird erläutert, welche Bedingungen für die am 25. Mai 2018 in Kraft getretene *Datenschutz-Grundverordnung*, kurz DSGVO, umgesetzt werden müssen, um mit den Vorgaben konform zu sein, trotz Verarbeitung von persönlichen Daten, insbesondere der Kontext-Variablen.

Daraufhin wird in Kapitel 4 das korrespondierende Projekt zu dieser Arbeit vorgestellt. Nach präzisieren der Aufgabenstellung, Planung und Umsetzung werden Methoden zur Kontext-Verarbeitung und entstandene Problematiken angeführt.

Abschließend folgt im Rahmen des Kapitels 5 die zusammenfassende Beantwortung der Forschungsfrage, das allgemeine Fazit und ein Ausblick zu möglichen Weiterführungen der Arbeit und des Projektes. Im Anhang befindet sich das Quellenverzeichnis und Informationen zum Inhalt der CD-Rom.

# Kapitel 2

## Chatbots

Dieses Kapitel bietet kompaktes Hintergrundwissen in Bezug auf *Chatbots*. Nach der Definition dieses Begriffs wird auf die geschichtlichen Entwicklungen eingegangen und einige Veröffentlichungen vorgestellt. Danach werden die unterschiedlichen Arten von Bots und möglichen *Use-Cases* erläutert. Ebenso wird kurz auf die bekanntesten verfügbaren Plattformen für die Veröffentlichung solcher Systeme eingegangen. Zuletzt wird die Rolle von Bots in der Web-Kommunikation bezüglich Werbung und *E-Commerce* zusammengefasst.

### 2.1 Definition

Der Begriff Chatbot setzt sich aus dem englischen *to chat* ‚plaudern, unterhalten‘ und *robot* ‚Roboter, Maschine‘ zusammen. Dies ist auch der gängigste Ausdruck, der sich über die Jahre in diesem Gebiet etabliert hat. Weitere Bezeichnungen sind unter anderen Chatterbot, sozialer Agent und konversationelles Interface. Die klassischen Definitionen von Chatbots beschreiben diese als Programme, welche Eingaben eines Benutzers in natürlicher Sprache empfangen, analysieren, darauf reagieren und eine Ausgabe, ebenso in natürlicher Sprache, an den Benutzer retournieren. Die heutigen Systeme werden optimiert um ebenso andere Eingaben wie beispielsweise Bilder verarbeiten zu können und mit Elementen wie Textdateien, Videoformaten oder Verlinkungen zu reagieren.

Die Definitionen sind zahlreich, aber dennoch grundsätzlich einheitlich. Sie unterscheiden sich in Bezug auf die verschiedenen Verwendungszwecke und der Höhe an Mechanismen und Strategien aus dem technologischen Bereich *Künstliche Intelligenz*, die nötig sind, um die jeweilige Applikation erfolgreich einsetzen zu können. Schlicht [18] sieht Chatbots als einen Service, angetrieben von Regeln und manchmal künstlicher Intelligenz, um mit ihm in einem Chat-Interface zu interagieren. Amir Shevat beschreibt in seiner Veröffentlichung [6] die heutigen Systeme in einer abstrakten Weise folgendermaßen: „At a very basic level, bots are a new user interface. [...] Bots are a new way to expose software services through a conversational interface.“<sup>1</sup>

Die Forschung und Entwicklung dieser Programme startete in den 60er Jahren. Damals war das Ziel vorrangig ein System zu entwickeln, welches eine menschliche Intelli-

---

<sup>1</sup>Übersetzung: ‚Grundsätzlich sind Bots eine neue Art von User-Interface. [...] Bots sind ein neuer Weg, um Software Services über ein konversationelles Interface zu bieten.‘

genz imitiert und somit bei einer geschriebenen Unterhaltung nicht festgestellt werden kann, ob man mit einer Maschine oder einem Menschen kommuniziert. Nach der rapiden Entwicklung der Technologie in den letzten Jahren, vor allem in Bezug auf Maschinen, die unser tägliches Leben bereichern, verschob sich das Hauptaugenmerk der Forschung in die nutzenorientierte Richtung. Chatbots nahmen ihren Platz als interaktive Helfer in diversen Webseiten und Messenger-Diensten wie *Facebook*, *Slack* und *Kik* ein. Mit Hilfe der gängigen Smartphone-Funktionen oder dem Smart-Home-System *Alexa* von *Amazon* findet ebenso die sprachliche Eingabe ihren Nutzen und ermöglicht weitere Interaktionen.

Alexander Braun beschreibt in *Chatbots in der Kundenkommunikation* [1] die Verwendung und das erwünschte Resultat folgendermaßen: Chatbots sollen „[...] Benutzern sowohl bei der Navigation durch komplexe Beschreibungen der Welt helfen als auch Anstoß dazu geben, bereits existierende Vorstellungsbilder automatisch anzuwenden.“ Der Nutzen von Chatbots reicht von Einkauf, Planung, über Steuerung und Information bis zu Unterhaltung. Um Beispiele zu nennen: *Amy Ingram*, ein Service der Firma *x.ai*, ist mit dem Kalender des Nutzers verbunden und regelt automatisch dessen Termine, Meetings und den diesbezüglichen E-Mail-Verkehr nach Einfügen ihrer Adresse im CC-Bereich. *Poncho* hingegen ist ein gesprächiger und humorvoller Bot, der dem Nutzer Wetterinformationen übermittelt. Er ist erreichbar über *Facebook Messenger*, *Kik*, *Slack* und *Viber*.

Trotz dem Trend der Entwicklung von hilfreicher Technik, werden ebenso Chatbots für reine Unterhaltung entwickelt. Diese verfolgen weiterhin das Ziel, eine menschliche Kommunikation störungsfrei zu imitieren. Zu den Spitzenreitern zählt *Mitsuku*, dreimalige Gewinnerin des Loebner Preises in den Jahren 2013, 2016 und 2017. Unabhängig davon, ob ein Chatbot für Amüsement oder spezifische Dienste entwickelt wird, trägt die gelungene Verarbeitung des Kontextes zu der Qualität des Systems und den daraus resultierenden Gesprächen bei.

## 2.2 Geschichte der Chatbots

Chatbots gelten im Allgemeinen in dem Bereich der Kommunikations- und Hilfsmittel als Neuheit des jetzigen Zeitalter. Diese Ansicht wird durch die erhöhten Werbemaßnahmen, auch im TV-Programm wie zum Beispiel bei *Alexa* von *Amazon*, verstärkt und findet somit ihren Weg in alle soziale Schichten. Die Geschichte und Entwicklung von künstlicher Intelligenz und Chatbots reicht jedoch in die 50er Jahre des 20. Jahrhunderts zurück und war Mitbestandteil der frühen Computerentwicklung.

Die Kontext-Verarbeitung wurde anfänglich durch bestimmte Verhaltensweisen umgangen, doch durch die wachsenden Systeme wurde die Ausarbeitung von Lösungen unumgänglich. Die heutigen Sprachassistenten haben ihre Methoden entwickelt, um einen optimalen Gesprächsverlauf zu gewährleisten, doch gelangen die Benutzer über gewisse Funktionen hinaus, kann fehlendes kontextuelles Verständnis des Bots die Erfahrung der Benutzer negativ beeinflussen.

### 2.2.1 Alan Turing

George Dyson unterteilt in seinem Werk *Turings Kathedrale* [3, S. 353] die Anfänge des digitalen Zeitalters in zwei große Abschnitte. Zuerst Gottfried Wilhelm Leibniz, den er als Propheten des Alten Testaments in diesem Zusammenhang bezeichnet, der die logischen Grundlagen erschaffen hat, und John von Neumann, der als Repräsentant des Neuen Testaments die Maschinen dafür baute. Als Bindeglied dieser beiden sehe er Alan Turing.

Alan Mathison Turing wurde 1912 in London geboren. Mit 14 Jahren besuchte er die renommierte *Sherbone-Schule* in Dorset. Ab 1931 studierte er am *Kings College* in Cambridge. 1937 wechselte er auf die *Princeton University*, an welcher er ein Jahr später mit 25 Jahren seinen Dokortitel erwarb. Turing beschäftigte sich in seinen frühen Jahren vor allem mit dem mathematischen Problem der *Berechenbarkeit*. Um der Lösung näher zu kommen, veröffentlichte er die ersten Entwürfe seiner Turing-Maschine.

Während des 2. Weltkrieges war Turing Beauftragter in der *Government Code and Cypher School*, welche von der englischen Regierung als Dechiffrierabteilung ins Leben gerufen wurde. Dort war Turing wesentlich an der Entschlüsselung des deutschen Enigma-Codes beteiligt. Nach dem Krieg arbeitete Turing in verschiedenen Institutionen an der Entwicklung neuer Rechner mit, bevor er sich dem Gebiet der künstlichen Intelligenz zuwandte.

1950 veröffentlichte er seinen wohl berühmtesten Artikel *Computing Machinery and Intelligence* [7], in dem sich nicht nur eine Vorhersage für intelligente Maschinen in der Zukunft befand, sondern auch die Beschreibung des heute noch bedeutsamen Turing Tests, den er selbst als „The Imitation Game“ bezeichnete.

In den frühen 50er Jahren wurde ein Gerichtsverfahren gegen Alan Turing eingeleitet. Nach damaligen englischen Gesetzen galt Homosexualität noch als Straftat. Als er für schuldig befunden wurde, stellte man ihn vor die Wahl, entweder ins Gefängnis zu gehen oder sich einer Hormonbehandlung mit Östrogen-Injektionen zu unterziehen, von denen bekannt war, dass sie starke Nebenwirkungen hervorrufen können, wie beispielsweise Depression. Im Juni 1954 beging Turing Selbstmord, indem er laut Medien einen mit Zyanid vergifteten Apfel verzehrte. Die Hormontherapie und deren schwere Folgen wurden als Grund angenommen. Am 24. Dezember 2013 sprach Queen Elizabeth posthum ein *Royal Pardon*, also eine königliche Begnadigung, aus, um ihn für seine außerordentlichen Dienste zu ehren [17]. Die biografischen Informationen in diesem Abschnitt stammen aus der Veröffentlichung von Andrew Hodges [5].

### 2.2.2 Turing Test

Der Turing Test wird heute noch verwendet, um die Intelligenz eines Programms zu testen. Eine Testperson unterhält sich schriftlich mit zwei Partnern. Einer dieser Partner ist ein Mensch, der andere eine Maschine. Die Testperson weiß, dass einer der beiden ein Programm ist. Der Turing Test ist bestanden, wenn die Testperson nach Befragung nicht genau sagen kann, wer der Mensch und wer der Computer ist (visualisiert in Abbildung 2.1).

Um ein gutes Ergebnis zu erzielen, müssen sich die Entwickler mit der Thematik der Kontext-Verarbeitung beschäftigen und versuchen gewisse Aspekte in der Gesprächsführung dahingehend zu optimieren. Warum Kontext, nicht nur in festgelegten Funktionen,

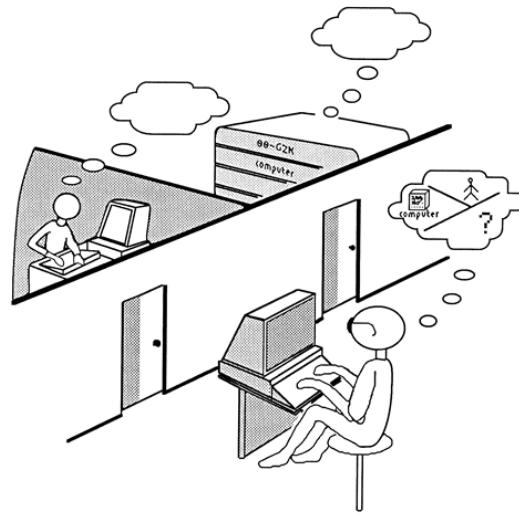


Abbildung 2.1: Darstellung eines Turing Tests [14].

sondern auch bei offenen Konversationen, wie sie von den Bewerbern beim Turing Test angeboten werden, von Bedeutung ist, wird im Abschnitt 3.2 erläutert.

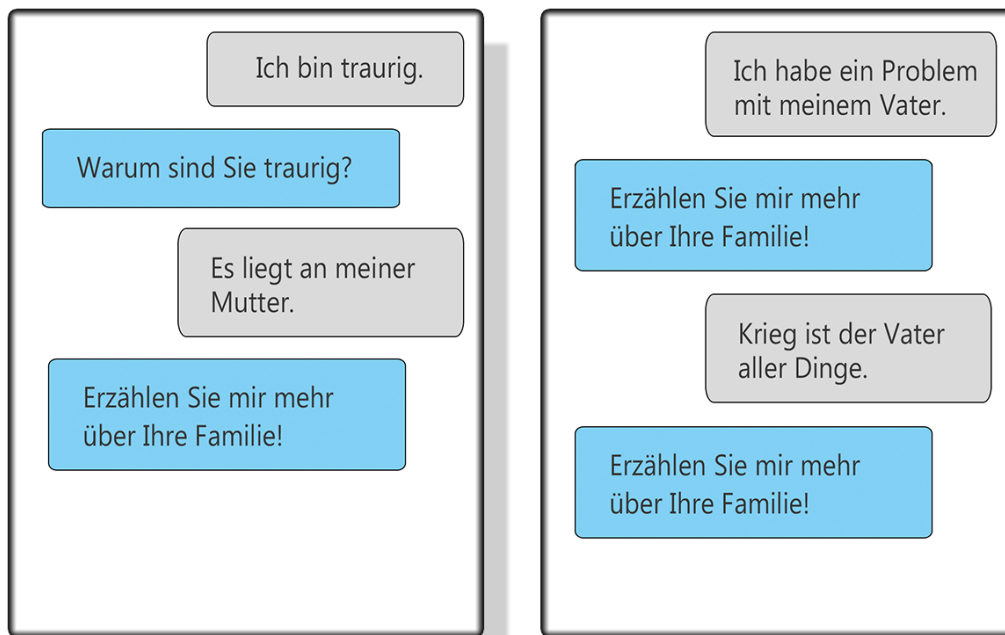
Alan Turing gibt in seinem Paper *Computing Machinery and Intelligence* [7] an, dass er denke, dass es in etwa 50 Jahren (entspreche dem Jahr 2000) möglich sei, einen Computer derart gut zu programmieren, dass sich mindestens 30% der Testpersonen nach einer 5-minütigen Unterhaltung nicht sicher sind, ob sie nun mit einem Menschen oder einem Programm geschrieben haben.

Douglas B. Lenat beschreibt in seiner Arbeit *Building a Maschine Smart Enough to Pass the Turing Test: Could We, Should We, Will We?* [4, Kap. 16], dass 3 Kernaspekte ausschlaggebend sind, um ein Programm aufzubauen, welches den Turing Test besteht.

- Der Bot muss fähig sein, in natürlicher Sprache zu kommunizieren.
- Er muss genügend wissen, um nicht einen Idioten aus sich selbst zu machen.
- Er muss sich unlogisch verhalten wenn und wie ein Mensch es tun würde.

Für das Programm, welches am ehesten menschlich wirkt, wird jährlich der *Loebner Preis* verliehen. Dennoch gibt es bis heute noch kein Programm, welches die Silber oder Goldmedaille errungen hat.

Von Forschern in diesem Gebiet wird der Turing Test als Verfahren zur Feststellung der Imitationsfähigkeit eines Menschen des Programms gesehen – weniger als eine Erprobung der Intelligenz. Bruce Edmonds beschreibt in seinem Beitrag *The Social Embedding of Intelligence* [4, Kap. 14] den Test mit Hilfe der Situation eines Immigranten. Dieser kommt aus einer fremden Kultur und muss nun viele Fähigkeiten erlernen, viel Wissen über die neue Region speichern und die Sprache meistern, um erfolgreich sein zu können. Dennoch würde niemand von ihm verlangen, ebenso würde er es vermutlich nicht anstreben, sich als Einheimischer auszugeben. Das Gegenteil wird von einem Chatbot erwartet, wenn man den Turing Test als Intelligenzprüfung sieht. Das Programm versucht für eine gewisse Zeit sich als Mensch auszugeben. Um allgemeine Intelligenz zu



**Abbildung 2.2:** Darstellung eines Gesprächs mit dem Bot *Eliza*.

bewerten, dient der Test aber nicht. Dieser ist laut Edmonds nur dafür da, die soziale Intelligenz festzustellen. Diese Erkenntnis spiegelt sich im originalen Namen für den Turing-Test wider: *The Imitation Game*. Etwas zu imitieren zeugt von Intelligenz, aber nur in einem bestimmten Bereich.

### 2.2.3 Eliza

Joseph Weizenbaum, ein deutsch-US-amerikanischer Informatiker, zählt als Pionier im Forschungsgebiet der Künstlichen Intelligenz. Von 1964 bis 1966 entwickelte er im *Computer Laboratory* des MIT (*Massachusetts Institute of Technology*) ein Programm, welches ermöglichte, über eine Fernschreibkonsole einen natürlich-sprachigen Dialog in englischer Sprache zu führen. Dieses benannte Weizenbaum, inspiriert von der *Pygmalion* Sage, *Eliza* und veröffentlichte es im Jahre 1966 [8]. *Eliza* gilt als einer der ersten Chatbots. Das System basiert auf einem simplen Schlüsselwort Prinzip, wie im linken Gespräch in der Abbildung 2.2 ersichtlich ist. Die Konversation beginnt mit der Aussage des Benutzers.

*Eliza* hat die Aufgabe einer Psychiaterin, die sich mit ihrem Patienten unterhält und dadurch weitere Reflexion der momentanen Situation erzeugen soll. Findet das Programm in der Eingabe keine verwertbaren Schlüsselwörter im verfügbaren Rahmen, weicht es auf Floskeln wie „Warum glauben Sie das?“ und „Ich weiß nicht, ob ich Sie richtig verstanden habe.“ aus oder greift auf frühere Eingaben zurück. Mit diesem Schlüsselwort Prinzip gehen weitere Einschränkungen einher, welche das rechte Gespräch in der Abbildung 2.2 aufzeigt.



Das System von *Eliza* arbeitet ohne konkrete Kontext-Verarbeitung. Es werden keine personenbezogenen Daten gespeichert oder verwendet. Grundsätzlich werden die Satz-teile grammatikalisch korrekt geändert und bestimmte Schlüsselwörter ersetzt, um Dy-namik zu erwirken. Es wird ein Gefühl von Intelligenz und kontextuelles Verständnis vermittelt, indem *Eliza* nach dem Auffassen eines Themas mit Floskeln wie „Und wei-ter?“ und „Erzählen Sie mehr davon.“ nach den Benutzereingaben reagiert, ohne aus einer Themen-Spezifizierung speziell angepasste Antworten zu retournieren.

#### 2.2.4 Alice

Das Programm *Artificial Linguistic Internet Computer Entity*, besser bekannt als *Alice* [4, Kap. 13], ist einer der größten und bekanntesten Vertreter von AIML, der *Artificial Intelligence Markup Language*. Mit ihrem Begründer Dr. Richard S. Wallace gewann *Alice* in den Jahren 2000, 2001 und 2004 die Bronzemedaille im Loebner-Wettbewerb. *Alice* besitzt ungefähr 41.000 sogenannte *Categories* und wird als Weiterentwicklung von *Eliza* gesehen. Die erste Version von *Alice* wurde 1995 in *SETL* implementiert. *SETL* ist eine höhere Programmiersprache, die auf der mathematischen Mengenlehre aufbaut. Da die Dateien von *Alice* öffentlich zugänglich waren, arbeiteten immer mehr Programmie- rer an ihr, bis 1998 das sogenannte „Program A“, welches in *Java* programmiert wurde, abgeschlossen war. 1999 gelang der Durchbruch mit *Alice* und machte sie zu einem rein AIML basierten Programm. Dieses „Program B“ gewann 2000 die Bronzemedaille des Loebner-Wettbewerbs.

Der grundsätzliche Aufbau eines AIML-Programms besteht aus den sogenannten *Categories*. Diese bestehen aus einem angegebenen *Input*, *Output* und optionalen *Kontext*. Die *Pattern-Sprache* besteht nur aus Buchstaben, Zahlen, *Wildcards*, Leerzeichen und *Tags*. Für die Verarbeitung von Kontext stehen in AIML drei *Tags* zur Verfügung. Das Schlüsselwort *that* bezieht sich auf den letzten *Output* des Programmes. Somit wird die folgende *Category* mit der Bedingung ausgewählt, dass der letzte Output überein- stimmt. Dies findet häufig in Ja- und Nein-Antworten Anwendung. Mit dem *Topic-Tag* kann ein bestimmtes Themengebiet gesetzt werden. Die nachfolgenden *Categories* wer- den dann primär nach diesem Rahmen ausgewählt. Mit Hilfe des *Think-Tags* können Variablen gespeichert werden, ohne den Benutzer darüber zu informieren. Beispielswei- se um Daten wie Name, Alter oder Geschlecht des Gegenübers zu speichern und in die Konversation einzubauen.

#### 2.2.5 Heutige Sprachassistenten

Mit der Markteinführung von *Siri* im Jahr 2010, einem sprachgesteuerten Bot von *Ap- ple*, wurde die Materie in den alltäglichen Gebrauch der Menschen eingeführt. Mit *Siri* kann sich der Benutzer nicht nur über vielfältige Themen dank der Netzwerkanbindung unterhalten, sondern auch Funktionen über sein Smartphone steuern, wie beispielswei- se das Tätigen von Anrufen oder Stellen der Weckfunktionen. 2014 wurde das *Apple HomeKit* für iOS vorgestellt. Per App oder der Sprachsteuerung *Siri* können die Be- nutzer ihre verbundenen Rolläden, Kameras, Thermostate und weitere Haushaltsgeräte bedienen. In den darauf folgenden Jahren veröffentlichten ebenso die weiteren großen IT- Unternehmen wie *Microsoft* (*Cortana*), *Google* (*Google Assistant*) und *Samsung* (*Bixby*) ihre Produkte.

Mit der Entwicklung der Software *Alexa* bietet *Amazon* mit Hilfe der Hardware *Amazon Echo* und weiteren ergänzenden Endprodukten nun auch die Möglichkeit, Smart-Home-Elemente sprachgesteuert zu beeinflussen. Dies reicht vom intelligenten Rasenmäher, über den Backofen bis zum *Connected-Car*. Startups und eingesessene Firmen rüsten ihre Produkte mit Fokus auf sprachgesteuerter Optimierung und weiteren Features nach, um der steigenden Nutzung gerecht zu werden und marktfähig zu bleiben. Durch die steigende Beliebtheit und Faszination dieser Produkte etabliert sich die Benutzung weiterer Bots auch im schriftlichen Bereich.

Die heute bekannten Assistenten haben gemeinsam, dass sie für bestimmte Aufgaben entwickelt wurden. Da ihre Fähigkeiten begrenzt, aber dennoch durch die Nutzung des Internets äußerst umfassend sind, können diese Vorgänge sorgfältig geplant und mit *Fallback-Aktionen* gesichert werden, um ein ideales Ergebnis zu liefern. In ihrer alltäglichen Nutzung bleiben die Benutzer größtenteils im Rahmen der Zwecke, für die diese Bots vermarktet wurden. Somit vermitteln diese Assistenten das solide Gefühl von kontextuellem Verständnis bei den Funktionen, für welche sie ausgelegt sind. Verlassen Benutzer diesen Rahmen, verfallen die Systeme in die selben Schwierigkeiten der unbegrenzte Gespräch-Bots wie *Alice*, *Mitsuku* und *Cleverbot*.

## 2.3 Kategorien

Chatbots unterscheiden sich untereinander durch einige Merkmale, wie die geplante Gesprächsführung, das gewünschte Resultat der Unterhaltung und der erfassten Zielgruppe. Die korrekte Feststellung und Planung der benötigten Art des Systems und die Auswahl der am besten geeignetsten Plattformen dafür ist essentiell für den erfolgreichen Einsatz am Markt. Die angeführten Arten stehen jeweils einer komplementären Anwendung gegenüber. Ebenso wird auf die jeweilige Rolle des Kontextes Bezug genommen. Die nachfolgende Einteilung und Bezeichnungen wurden von Amir Shevat [6] getroffen.

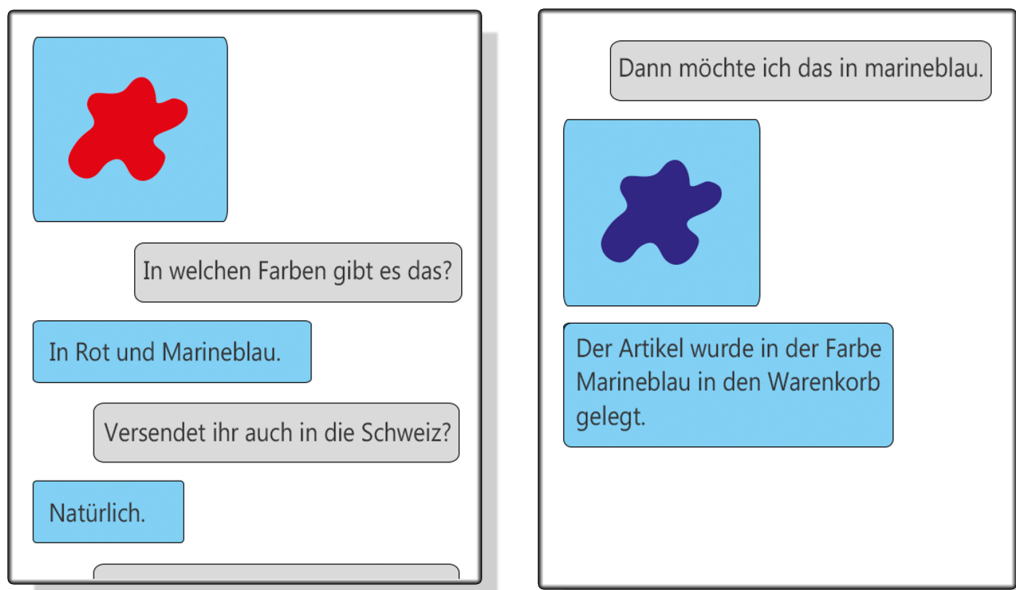
### 2.3.1 Arten

#### Personal- und Team-Bots

Ein Personal-Bot dient dem Benutzer mit direkten Nachrichten als persönlicher Assistent. Das Gespräch findet nur unter dem Beteiligten und dem System statt. Beispiele wären ein Einkaufsberater oder ein Helfer zur Buchung eines Fluges.

Im Gegensatz zum Personal-Bot steht der Team-Bot, der entweder mit einzelnen Personen interagiert oder in einer Gruppenunterhaltung postet. Die Schwierigkeit der Gestaltung liegt in der optimalen Organisation und Synchronisation der eingegebenen Daten der unterschiedlichen User. Beispielsweise wäre der Dienst *Lunch Train* auf *Slack* zu nennen, welcher die gewünschten Lokalisationen für die Mittagspause mit den Teammitgliedern abgleicht. Momentan wird diese Systemform noch nicht von allen Plattformen unterstützt. Durch die rapide Entwicklung in diesem Forschungsgebiet werden jedoch die restlichen Dienste wie *Facebook Messenger* in den nächsten Jahren voraussichtlich nachziehen.

Die wesentlichen Aspekte zur korrekten Verwendung von Kontext sind bei diesen



**Abbildung 2.3:** Beispiel-Konversation zur Darstellung der Bedeutung des Kontextes in Personal-Bots.

beiden Arten überschaubar. Der Bot sollte mit rapiden Themenwechsel umgehen können, um eine problemlose Abhandlung, wie in Abbildung 2.3 ersichtlich, zu ermöglichen. Ein gelungenes Ergebnis kann erreicht werden, wenn die spezifischen Bereiche in Bezug auf Kontext von den Entwicklern durchgeplant und abgedeckt wurden. Dennoch stoßen diese Bots sehr häufig an ihre Grenzen, wenn der Benutzer etwas außerhalb des geplanten Funktionsbereichs anspricht. Das *Opt-In* (siehe Abschnitt 3.5.4) sollte eine klare Beschreibung über programmierte Funktionen und die Art der Nutzung enthalten, um Missverständnisse und Enttäuschungen auf Seiten des Benutzers vorzubeugen.

### Super- und Domain-spezifische Bots

Domain-spezifische Chatbots sind für einen Service zuständig, welcher normalerweise ein Produkt, eine Marke oder eine Firma repräsentiert. Ein Super-Bot hingegen ist für mehrere Dienste universell gestaltet. Beispiele dafür sind zum einen der Chatbot für den Konzern *Pizza-Hut* für die spezifische Variante und auf der anderen Seite *Google Assistant*, welcher mehrere Google Services vereinigt. Der Vorteil für den Benutzer bei einem Super-Bot liegt bei der vielseitigen Verwendung, ohne mehrere verschiedene Dienste in Anspruch nehmen zu müssen. Im Gegensatz dazu verdeutlicht der domain-spezifische Bot die Marke und Aufgabe effizienter, da dieses Programm speziell für diese eine Beschäftigung entwickelt wurde. Ebenso müssen die User nicht durch eventuelle Menüs navigieren, um den gewünschten Service zu erreichen.

In Bezug auf Kontext bergen domain-spezifische Bots ein höheres Risiko für den jeweiligen Konzern. Benutzer, die vom Konversationsverlauf entmutigt sind, können sich nicht nur vom angebotenen Bot-Service, sondern auch von der Marke allgemein

entfernen. Die Entwickler müssen auf die wesentliche Kontext-Verarbeitung achten und genügend *Fallback-Situationen* abhandeln können. Im Gegensatz dazu, bieten Super-Bots eine gewisse Sicherheit. Bei negativen Erfahrungen besteht die Möglichkeit, dass der Benutzer sich nicht generell von diesem Produkt entfernt, sondern lediglich von einem bestimmten Teilaspekt Abstand hält.

### Business- und Konsumenten-Bots

Unternehmen können Bots verwenden, um routinemäßige Arbeiten zu automatisieren und somit den damit verbundenen Arbeitsaufwand zu minimieren. Ebenso kann er als Brücke zwischen den Mitarbeitern dienen. Das Ziel der Business-Bots ist, den User maximal, doch mit so wenig wie möglich benötigten und präzisen Anweisungen zu unterstützen. Konsumenten-Bots hingegen werden entwickelt, um den Nutzer zu unterhalten, ihm Neuigkeiten über seine abonnierte Marke zu liefern oder ihm bei privaten Projekten zu unterstützen (Gesundheit, Ernährung etc.). Die Gesprächsführung ist weniger zielorientiert und richtet sich auf dynamische Interaktionen.

Das Risiko für eine negative Erfahrung der Benutzer, in Bezug auf Kontext, ist bei Business-Bots gering. Diese Programme dienen, wie beschrieben, zur Abhandlung von routinemäßigen Arbeiten. Werden diese Szenarien von den Entwicklern in das Funktion-Schema des Bots aufgenommen, besteht kaum Gefahr, dass die Konversation deswegen unvorteilhaft verläuft. Zusätzlich können die Mitarbeiter, die den Dienst verwenden, zu Beginn ausführlich zu den vorhandenen Funktionen und fehlenden Angeboten gebrieft werden. Konsumenten-Bots hingegen sind grundsätzlich anfälliger für *Bot Amnesia* (siehe Abschnitt 3.4). Das Risiko für einen ungewollten Konversationsverlauf schwankt. Es hängt vom Umfang des angebotenen Themenkatalogs, der Ausführlichkeit der Einführung über Funktionen des Programms und der Art des Angebots ab.

### Sprach- und Text-Bots

Systeme, die für die schriftliche Konversation gestaltet sind, können auf Plattformen wie *Facebook Messenger*, *Slack*, *Kik* und weiteren veröffentlicht werden. Sprachgesteuerte Bots wie *Amazon Alexa*, *Microsoft Cortana* oder *Apple Siri* werden meist durch einen Button oder einen bestimmten Aufruf bei elektronischen Geräten aktiviert. Die Interaktion besteht zumeist aus einer Aufforderung oder Frage und der Bestätigung oder Antwort. Dies dient vor allem für Interaktionen, bei denen der Nutzer nicht mit den Händen agiert, wie an dem Computer oder Smartphone. Auch bei sprachgesteuerten Bots spielt die Verarbeitung des Kontextes eine elementare Rolle. Die Verwendung und Notwendigkeit von Kontext weist Unterschiede zu den von Text-Bots auf, weshalb das Konversations-Design hinsichtlich darauf aufgebaut werden muss.

#### 2.3.2 Use-Cases

##### Handel und Werbung

Für bereits bestehende Unternehmen ist die Verwendung von Bots als Werbemittel und *E-Commerce* ein optimaler Beitrag zum bestehenden Kommunikationsplan. So kann beispielsweise eine Bestellung bei dem Unternehmen *Pizza-Hut* intuitiv in einer Konversation abgegeben werden. Im Nachhinein kann nicht nur eine Bewertung eingeholt,

sondern auch zukünftige Angebote für die präferierten Gerichte oder weitere Werbung übermittelt werden, um die Kundenbindung zu stärken.

Durch die Konversation in der Abbildung 2.3 wird ersichtlich, dass die Bewahrung des Kontextes eine unumgängliche Anforderung im System ist, um den Kunden einen reibungslosen Einkauf zu ermöglichen. In diesem Beispiel wechselt der Benutzer mit der Versands-Frage das Thema. Trotz fehlendes Ansprechen, um welches Objekt es sich handelt, reagiert der Bot korrekt auf die Aussage des Benutzers, dass er die Farbe Blau bevorzugt und fügt das vor dem Themenwechsel im Fokus gestandene Objekt in den Warenkorb hinzu.

### Notification

Diese Bots können E-Mail und App-interne Benachrichtigungen ersetzen. Beispielsweise können Nachrichten, Preisschwankungen oder weitere Analysen aktualisiert gesendet werden. Diese Abhandlung kann ebenso in Kombination mit anderen Use-Cases verwendet werden und diese dadurch bestärken.

### Coaching

Den eigenen Tagesablauf und die Gesundheit zu optimieren nimmt einen großen Stellenwert in den digitalen Medien ein. Seit Anbeginn dieser Einstellung stehen den Interessenten Tools mit Hilfestellungen und Erinnerungen zur Verfügung. Ziel ist die Steigerung der Produktivität in den individuellen Bereichen. Chatbots können auch in diesem Gebiet interaktiv mit dem Nutzer agieren, dessen eingegebenen Information analysieren und Erinnerungen direkt als Nachricht schicken.

### Unternehmen

Die Kommunikation unter Team-Mitgliedern kann durch den Einsatz von Bots zu einer gesteigerten Produktivität führen. Ebenso können sie routinemäßige Kleinarbeit erledigen, wie Meetings zu passenden Zeiten zu arrangieren und standardisierte Verträge vorzubereiten.

### Unterhaltung

Neben der Verwendung eines Bots mit konkretem Ziel, gibt es auch etliche auf dem Markt, die der Unterhaltung dienen. Mit dem *Mitsuku* Bot, dreifacher Gewinner des Loebner Preises, kann man sich frei über ein Thema seiner Wahl unterhalten und über die generierten Antworten staunen. Ebenso ist es möglich kleine Spiele über diverse Plattformen anzubieten.

### FAQ

Viele Chatbots übernehmen die Funktion als Kundenservice. Diese überzeugen durch ihre Fähigkeiten der unbegrenzten Anzahl an Kommunikationspartnern und der unumschränkten Erreichbarkeit. Dies führt zu einer Kostenersparnis und Entlastung der Mitarbeiter. Die Entwicklung ist meist von geringerem Aufwand, da die Gespräche größtenteils

aus einem typischen Frage-Antwort-Prinzip bestehen, die durch häufiger Wiederholung ideal abgedeckt werden können.

### System-Integration

Es kann sich vorteilhaft auf bestimmte Arbeitsabläufe auswirken, wenn ein System eines Drittanbieters in den bestehenden Service eines Bots eingebaut wird. So können beispielsweise Statistiken oder Zusammenfassungen angezeigt werden.

### Bindeglied

Ein Bot kann nicht nur selbst als Service dienen, sondern auch als ein Bindeglied zwischen zwei Personen fungieren. Das Prinzip ist auf die bekannten Systeme von Telefonaten mit voriger Einteilung und Weiterleitung zurückzuführen. Ein populärer Dienst mit dieser Funktion ist der Bot *Sensay*. Dieser verbindet den Nutzer durch eine Aussage oder Frage mit einem anderen Menschen, der laut System diese Erfahrung aufweist, um Rat anzubieten.

## 2.4 Plattformen

Die Auswahl an möglichen Plattformen für die Integration eines Bots ist riesig. Obwohl sie oberflächlich gesehen die selben Funktionen anbieten, ermöglichen manche von ihnen gewisse Vorteile für die individuelle Bot-Art (siehe Abschnitt 2.3). Viele Entwickler veröffentlichen ihre Programme auf mehreren Plattformen simultan, um weitere Zielgruppen durch die unterschiedlichen Kanäle zu erreichen. Dafür sind auch meist nur geringe technische Änderungen, häufig bei der Einbindung von *Rich Messages*, nötig. In diesem Abschnitt werden die bekanntesten und gängigsten Plattformen vorgestellt, die sich für einen Chatbot eignen.

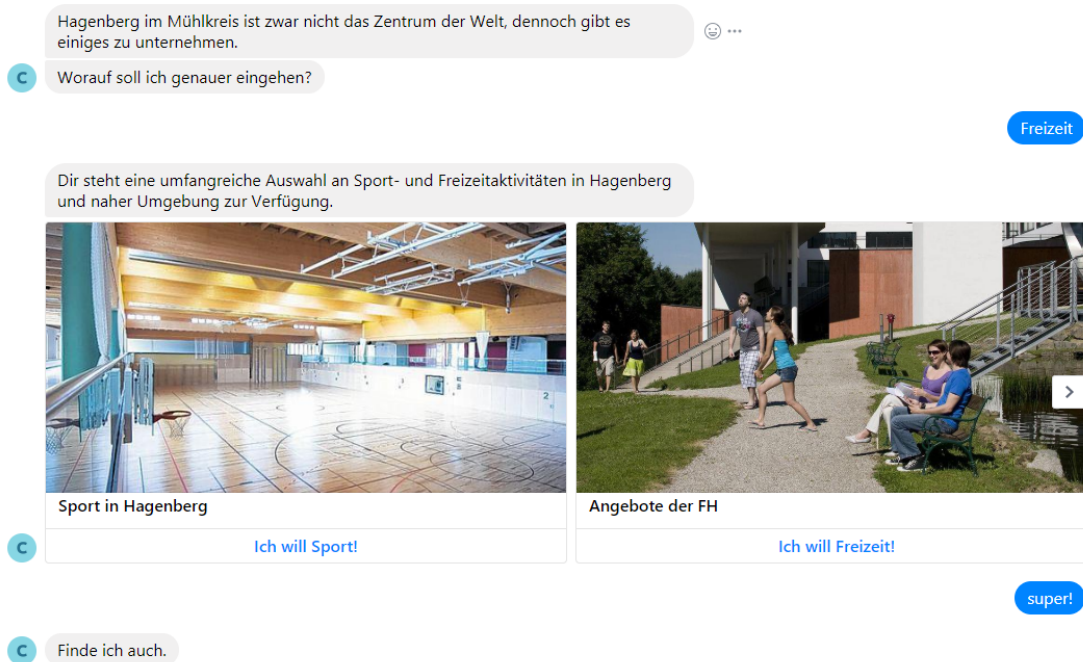
### 2.4.1 Facebook Messenger

*Facebook Messenger* erfreut sich bei vielen Entwicklern großer Beliebtheit und wird häufig als Plattform für Personal-Bots (siehe 2.3.1), mit den unterschiedlichsten Use-Cases, genutzt. Die Plattform punktet mit einer einfachen Integration, kostenfreier Verwaltung bei *Facebook for Developers* und Einbinden des Bots, umfangreicher Auskunft zu analytischen Daten, ansehnlichen *Rich Messages* wie Buttons und den sogenannten *Karussells* (siehe Abbildung 2.4) und einer ausführlichen API. Die weiteren Vorteile werden in der folgenden Auflistung ersichtlich:

- Der Bot kann feststellen, ob die Nachricht angekommen ist.
- Eine Ansicht von externen Seiten ist möglich.
- *Geolocation* ist mit nur einem Klick ermittelbar.

### 2.4.2 Slack

Die Plattform *Slack* bietet Business-Bots eine optimale Umgebung. Die Bots können in Gruppen und mit Einzelpersonen interagieren. Bots auf *Slack* werden somit häufig von



**Abbildung 2.4:** Anzeige eines *Karussells* in *Facebook Messenger*, für die grafische Auflistung von mehreren Komponenten.

Teams benutzt, die gewisse Abläufe automatisieren und ihre Koordination untereinander optimieren wollen. Neben dem Versenden und Empfang von Nachrichten und *Rich Messages* werden folgende Vorteile von der API ermöglicht:

- Mit einem Schrägstrich können *Commands* ausgeführt werden.
- Mit der *Event* API können Bots über verschiedene Ereignisse informiert werden, wie zum Beispiel, wenn ein Benutzer dem Channel hinzugefügt wird.
- *Slack* und dessen *Identity Provider* kann genutzt werden, um die Identität eines Benutzers festzustellen.
- Bots können administrative Aktionen ausführen.

### 2.4.3 Alexa

*Alexa* von *Amazon* ist eine gängige Plattform für sprachgesteuerte Bots. Anzumerken ist, dass der Bot nicht als individueller Teil eingebaut ist, sondern über einen sogenannten *Skill* gesteuert wird. Beispielsweise wäre eine Anfrage: „Alexa, frage Wetter-Bot nach den heutigen Temperaturen.“, wobei *Wetter-Bot* den implementierten Service darstellt. Das *Alexa Skill Kit* bietet den Entwicklern einige Möglichkeiten, die gefolgt aufgelistet sind:

- Ein *Aufrufsname* kann erstellt werden, mit dem der Service angesprochen wird.
- *Alexa* transkribiert die Aussage und schickt sie weiter. Ebenso liest sie den Text der Antwort vor.
- Das *Kit* ermöglicht die Arbeit mit *Smart-Home*-Elementen.

#### 2.4.4 Google Assistant

*Google Assistant* ist ein *Super-Bot* der Firma *Google Inc.* Mit der Veröffentlichung von *Google Allo* wird eine passende Messenger-Anwendung für die Dienste hinzugefügt. Die *Rich Messages* haben einen ähnlichen Umfang und vergleichbares Design wie bei *Facebook Messenger*. Ist der Service auf einem elektronischen Gerät mit Mikrofon aktiviert, kann der Benutzer Anfragen mit der Aussage „OK Google“ starten. Der Einbau und die Benutzung eines externen Bots wird wie bei *Amazon Alexa* geregelt, mit dem Unterschied, dass der Aufruf beim *Google-Service* offiziell als *Aktion* bezeichnet wird, anstatt den Begriff *Skill* zu verwenden.

#### 2.4.5 Weitere

Mit *Kik*, *Skype*, *WeChat* und vielen weiteren, stehen den Entwicklern einige Türen offen, um ihren Bot zu verbreiten. Aus Sicht des Marketing ist es ratsam das selbe Programm auf mehreren Plattformen anzubieten, damit die Kunden den Dienst in ihrer bevorzugten Umgebung benutzen können. Ebenso sind Email und SMS als Plattformen für Bots zu erwähnen. Trotz der Tatsache, dass ihre Möglichkeiten beschränkt sind, bieten sie für manche Use-Cases die geeignete Anwendungs-Umgebung.

### 2.5 Rolle in der Web-Kommunikation

Unternehmen arbeiteten jeher an Strategien, um Kosten zu sparen, die Kundenzufriedenheit zu steigern und in den Köpfen der Zielgruppe präsent zu sein. In den frühen Jahren des 21. Jahrhunderts fokussierte sich das Online-Marketing auf Suchmaschinen-Optimierung. Nach der gesteigerten Popularität von *Facebook* und co. ging der Trend klar zu Social Media Präsenz. Nach dem deutlichen Anstieg von *Mobile Marketing* und dem jetzigen Plateau aufgrund des Rückgangs an App-Downloads, dreht sich nun alles um Messenger Dienste [19]. Chatbots können die genannten Bereiche durch Optimierung des Kundenservices, direkte und zielgerichtete Werbung, Unterstützung bei Buchungen und interaktive Verkaufsberatungen wesentlich verbessern und die Marke durch Präsenz und Aktualität stärken.

#### 2.5.1 Kundenanfragen

In der Informationsflut, der Nutzer des Internets ausgesetzt sind, können gewünschte Aspekte und gesuchte Daten untergehen. Trotz der digitalisierten Welt ist es nicht unüblich geworden, dass Menschen direkten Kontakt zu einem Unternehmen suchen. Die Beweggründe reichen von Fragen, die auf der Webseite nicht klar ersichtlich waren, bis zu Unklarheiten und Problemstellungen bezüglich diversen Handlungsabläufen. Chatbots können diese Anfragen übernehmen und im Zweifelsfall auch an einen menschlichen Mitarbeiter weiterleiten. So wird für Entlastung des Kundenservices gesorgt und anfallende Fragen gefiltert. Zusammengefasst bieten diese Systeme folgende Vorteile für das Unternehmen selbst und ihre Kunden:

- Erreichbarkeit: Ein Chatbot ist, ausgenommen von Systemausfällen, 24 Stunden an 7 Tagen die Woche erreichbar. Dies ermöglicht es den Kunden unabhängiger von ausgewählten Servicezeiten zu interagieren.



- **Anfragen:** Besonders bei großen Konzernen können die Kundenservice-Mitarbeiter entlastet werden, da ein Chatbot beliebig viele Anfragen zeitgleich abklären kann.
- **Reaktionszeit:** Anstatt mehrere Stunden oder Tage auf eine Antwort zu warten oder in der Warteschleife eines Call-Centers zu verweilen, bekommen die Nutzer innerhalb von Sekunden die gewünschte Information.

Wenn das Erlebnis mit dem interaktiven System erfolgreich gestaltet und geführt wird, kann dies Dank der verringerten Wartezeit, minimalem Aufwand bei der Suche und unabhängigen Erreichbarkeit zu einer erhöhten Kundenzufriedenheit führen. Ebenso können die Kosten durch die niedrigere investierte Zeitspanne im Bereich des telefonischen und schriftlichen Kundenservice verringert werden.

### 2.5.2 Werbung und E-Commerce

Heutige Werbemaßnahmen überfluten die Webseiten, Straßenränder, Fernsehen und Postwürfe, welche Endkunden dazu veranlasst sich um Gegenmaßnahmen zu bemühen, wie beispielsweise diverse Block-Systeme für den Browser zu aktivieren, das Umschalten auf einen anderen Sender oder das Anbringen von Aufklebern, dass keine Werbung im Postkasten erwünscht ist. Umso schwieriger gestaltet sich effektive Werbung für die Unternehmen. Durch die hohe und häufige Nutzung sind *Messenger* Plattformen ein idealer und direkter Zugang zu den Kunden und stärkt bestehende Kontakte. Nach Zustimmung des Nutzers können auf diese Weise Werbung und Neuigkeiten durch ein zentrales System gesendet werden. Diese werden auf dieselbe Art und Weise von anderen Usern als Benachrichtigung geschickt und angezeigt. Somit entsteht eine hohe Erreichbarkeit und direkter Bezug zum potenziellen Kunden.

Traditionell wird Online-Werbung angezeigt und durch einen Klick aktiviert. Dies wird durch die enorme Häufigkeit als aufdringlich angesehen und größtenteils ignoriert. Mit Chatbots ist eine aktive Konversation und dynamischer Einsatz von Werbemitteln mit dem Benutzer möglich. Durch personenbezogene Fragen, die im Laufe eines Gesprächs gestellt werden können, ist es möglich Daten zu dem jeweiligen User zu analysieren und die Werbung darauf individuell anzupassen. So kann die nächste Anzeige von Kleidungsstücken in der Lieblingsfarbe des Users gestaltet sein oder Ankündigungen zu Konzerten von Bands aus dem präferierten Musikstil gesendet werden.

Chatbots werden neben der Verwendung als Kundenservice, Unterhaltung und Werbemittel auch für Verkaufsberatungen und Buchungen eingesetzt und ermöglichen somit eine interaktive Shopping-Tour durch das Sortiment oder unterstützen den Benutzer bei der Reservierung eines Fluges. Da der aktuelle Trend dazu neigt, dass die Nutzer keine zusätzlichen Apps herunterladen, außer bei unabdingbarer Notwendigkeit, stellen Bots eine effektive Alternative dar, da die *Messenger-Dienste* zum festen Bestandteil vieler Handynutzer gehören und keine Umwege nötig sind, um einen Dienst zu nutzen. Amerikanische Bürger können sich zum Beispiel mittels *Facebook Messenger* ein *Uber-Car* bestellen und die Route festlegen oder ihre Lieblings Speisen von *Domino's Pizza* anfordern, ohne dass sie eine spezifische App von diesen Anbietern benötigen.

Um die Kundenzufriedenheit weiter zu steigern, sind individuelle Abhandlungen nach der Bestellung ein effizientes Mittel. Der Bot kann über weitere ähnliche Produkte informieren, oft auch *Kunden kauften auch* genannt, oder nach Erhalt der Lieferung nach der Zufriedenheit der Zustellung fragen. Durch die Konto-Daten des Nutzers kann

beispielsweise auch ein persönlicher Geburtstags-Gruß mit Gutschein gemeinsam mit Kaufempfehlungen anhand der letzten Bestellungen gesendet werden, um stets wieder auf die Marke aufmerksam zu machen.

Doch Firmen können nicht nur aktiv mit einem eigenen Dienst ihre Produkte bewerben, sondern auch bei bereits existierenden Systemen anschließen. Bei einem Bot, der über das Wetter informiert, könnte bei sonnigen Aussichten eine Empfehlung für eine bestimmte Sonnenschutz-Creme angehängt werden. Ebenso wäre bei einem Gespräch über eine bestimmte Stadt oder Region eine Werbeschaltung der lokalen Autovermietung möglich. Somit kann die ideale Zielgruppe angesprochen werden.

### 2.5.3 Risiko

Chatbots sind nicht am Ende ihrer Entwicklung angelangt. Häufig sind Verwechslungen und Missverständnisse während eines Gespräches möglich und in drastischen Fällen gibt der Nutzer die Konversation auf und deaktiviert die Benachrichtigungen. Es muss sichergestellt werden, dass die Aufgaben und Fähigkeiten des Systems klar dargestellt werden. Ebenso sollten diese Möglichkeiten limitiert sein, um eine optimal ausgearbeitete und komplikationsfreie Ausführung zu gewährleisten. Die zukünftigen Entwicklungen werden verbessertes Auffassungsvermögen der Situation bieten, um einen gezielten Einsatz von einem menschlichen Mitarbeiter in heiklen Situationen zu ermöglichen.

Nach einer Befragung [12] würden 73% der Nutzer nach einer negativen Erfahrung nicht mehr mit dem Chatbot interagieren. Dies bedeutet nicht zwangsweise, dass sich der User auch von der dazugehörigen Marke entfernt. Für sich allein stehende Chatbots ist diese Entscheidung jedoch fatal, da eine negative Bewertung auch zukünftige Benutzer abhält und prägt.

Neben einem allgemein nicht zufriedenstellenden Service sind ungeplante Verläufe und erzwungene erneute Starts des Vorgangs Hauptgründe für vorzeitige Beendigung einer Konversation mit einem Chatbot. Abgesehen vom generellen Service spielt die korrekte Verarbeitung und Verwendung des Kontextes eine maßgebliche Rolle bei diesen Problematiken. Doch diese Schwierigkeiten wirken sich bei Chatbots, die für das Unternehmen automatisierte Verkaufs-Arbeit leisten, drastisch aus. Im Gegensatz dazu werden Bots, die für bloße Unterhaltungszwecke geschaffen wurden, schneller ein Fehler verzeihen, da die Benutzer sich keinen konkreten Service erwarten, den sie abschließen möchten. Im nachfolgenden Kapitel wird auf die Fragestellungen, welche Probleme auftreten, welche Maßnahmen in Folge dessen getätigt werden können und welche Mittel zur Verfügung stehen, um Schwierigkeiten im Vorhinein zu vermeiden, eingegangen.

# Kapitel 3

## Kontext

In diesem Kapitel werden die Haupt-Aspekte dieser Arbeit vorgestellt. Nach der Definition des Begriffs *Kontext* wird erläutert, warum dieser bei Konversationen eine bedeutende Rolle spielt. Danach werden mögliche Punkte in der technischen Entwicklung der Kontext-Verarbeitung angeführt. Nach Auflistung und Erklärung auftretender Schwierigkeiten, werden kommunikative Maßnahmen zur Vorbeugung eben dieser zusammengefasst. Zuletzt werden noch relevante Artikel der *Datenschutz-Grundverordnung* angegeben und deren mögliche Umsetzung im System angeboten.

### 3.1 Definition

Der Begriff Kontext stammt vom lateinischen Wort *contexere* ab, welches als ‚zusammenweben‘ übersetzt werden kann. Im Bereich der Linguistik (Sprachwissenschaft) werden alle Elemente einer Kommunikationssituation, die das Verständnis einer Aussage bestimmen und beeinflussen, als Kontext bezeichnet. Der Begriff wird häufig in sprachlichen, verbalen und situativen, non-verbalen Kontext unterteilt. Die deutsche Sprachwissenschaftlerin Hadumod Bußmann nähert sich, in ihrer Veröffentlichung *Lexikon der Sprachwissenschaft* [2], detailliert der Thematik und unterteilt Kontext in drei Bereiche. Sie unterscheidet zwischen allgemeinem, sozialem und sprachlichem Kontext. Ort, Zeit und Handlung der Situation bilden den allgemeinen Kontext. Als sozialer Kontext gilt die Beziehung der Kommunikationspartner, deren Verhältnis zueinander, ihre Interessen und Umfang an Wissen. Als sprachlicher Kontext, um Bußmann [2] konkret zu zitieren, gilt der Zusammenhang „der die Ausdrücke grammatisch und semantisch verknüpft und gleichzeitig durch *Deixis* oder pragmatische Indikatoren wie Modalpartikel in den situativen K. einbettet, [...]“. *Deixis* wird umgangssprachlich auch Zeigewort genannt. Deiktische Ausdrücken gewinnen erst ihre Bedeutung, wenn sie in einer Sprechsituation verwendet werden, wie beispielsweise die Wörter *hier*, *dort*, *drüben*, und *sie*.

### 3.2 Bedeutsamkeit von Kontext

Die Mehrzahl an Bots konzentrieren sich darauf, ein gutes Ergebnis zu einer spezifischen Anfrage zu liefern. Was davor oder danach passiert, ist im System noch nicht relevant. Doch durch die Einbindung von Kontext können nicht nur technische Funktionen, son-

dern auch die Kommunikation zwischen Chatbot und Benutzer verbessert werden. Dies kann zu einer positiven Erfahrung führen, welche sich auf die Empfindung des Users gegenüber der vertretenen Marke auswirkt. Wird der Kontext zu einer Situation und Person verstanden, können Aspekte wie Unterbreiten von Angeboten, Gewinnen des Vertrauens und steigende Zufriedenheit verbessert werden.

Viele Teile eines Gespräches gewinnen erst durch den bestehenden Kontext an Bedeutung. Unabhängig davon, ob es sich um Pronomen (siehe Abschnitt 3.3.3) handelt, die ohne vorige Anmerkungen nicht einzuordnen wären, oder komplette Aussagen eines Users, die in unterschiedlichen Situationen verschiedene Zusammenhänge eröffnen.

Kontext und das zugehörige Erinnerungsvermögen können automatisierte Vorgänge verbessern. Zum Beispiel kann ein Chatbot, der für Bestellungen bei einer Pizzeria zuständig ist, gespeicherte Daten wie Adresse und ältere Lieferungen aufrufen und somit den Vorgang verkürzen.

### 3.3 Relevante Aspekte der technischen Entwicklung

In diesem Abschnitt werden einige Themen zur Entwicklung und Optimierung der Kontext-Verarbeitung erläutert. Um verständliche Beispiele aufzuzeigen wird, das imaginäre Konstrukt eines Chatbots verwendet, der sich um alle Belange rund um eine Hotel-Buchung, inklusive angebotene Ausflüge und Behandlungen, kümmert. Im Abschnitt 3.5 werden mögliche Modifikationen des Kommunikation-Designs zur Steigerung der Qualität in Bezug auf Kontext präzisiert.

#### 3.3.1 Variablen

Die korrekte Erhebung, Speicherung und Verwendung von Variablen ist unumgänglich, um eine, in Bezug auf den Kontext, gelungene Konversation zu führen und dem Benutzer die gewünschten Dienste zu leisten. Grundsätzlich kann man zwischen gebundene und globale Variablen unterscheiden. Die gebundenen Variablen sind zeitlich begrenzt und meist gekoppelt mit dem jeweiligen Intent. Wird bei dem Hotel-Bot beispielsweise über die Buchung eines Ausfluges gesprochen, dann ist der gebundene Kontext zu dieser Situation *Ausflüge*.

Im Gegensatz dazu wäre das dahinterstehende Datum des gesamten Urlaubes eine globale Variable, die nicht nur bei der Zimmer-Reservierung, sondern auch bei der Buchung eines Kurztrips benötigt wird. Nach Fixierung des Reiseterrines, ermöglicht dies somit den gemütlichen und realistischen Dienst, dass der Bot den Benutzer auf die Anfrage nach Ausflügen nicht nach dem gewünschten Datum fragt, sondern eine Auflistung von möglichen Unternehmungen im bereits angegebenen Zeitraum sendet. Der Kontext *Ausflüge* hingegen, wird nach dem Wechsel zum Thema *Behandlungen* oder sonstiges, nicht mehr gebraucht. Eine weitere mögliche globale Variable ist der Name des Benutzers, der bei Zusammenfassungen oder Verabschiedung benutzt werden kann. Ebenso ergeben sich mit Hilfe der Abfrage der momentanen Uhrzeit weitere Möglichkeiten zur Personalisierung. Der Tag könnte in vier Abschnitte unterteilt, die jeweiligen Variablen zugeteilt und in der Konversation verwendet werden. Je nach Tageszeit kann dann die angebrachte Begrüßung, Verabschiedung und weiteres aufgerufen werden.

### 3.3.2 Lebensspanne

Ein sensibler Punkt der Qualität eines Bots ist die Planung der Zeitspanne, die die Variablen gespeichert werden. Kein Benutzer wäre erfreut, besonders bei Kauf oder Buchungs-Vorgängen, wenn nach einer kurzen Unterbrechung der Konversation, um zum Beispiel weitere Details mit den Mitreisenden abzusprechen, der Kontext verloren gegangen ist und die komplette Prozedur erneut durchlaufen werden muss. Diese Thematik spielt vor allem in schriftlichen Konversationen eine Rolle, da diese darauf ausgelegt sind, zeitliche Abstände zwischen den Nachrichten zu ermöglichen. Bei eigenständig programmierten Systemen muss lediglich eine Textdatei oder Datenbank, je nach Anzahl der aktiven Nutzer, und Funktionen, die diese verwalten und nach einer festgelegten Zeit auch aussortieren und säubern erstellt werden.

Da die angebotenen Plattformen für Bot-Entwicklung, besonders in Bezug auf *Natural Language Processing*, kurz NLP, gute Dienste und Hilfestellungen leisten, steigt ihre Beliebtheit bei Personen mit unterschiedlichem technischen Wissen. Meist bieten diese Systeme nur eine relativ kurze Zeitspanne für die Speicherung der Variablen an. Bei *Dialogflow*<sup>1</sup>, ein Angebot von *Google*, haben die Variablen eine äußerst kurze Lebensdauer von 20 Minuten. Diese kurze Unterbrechung findet häufig in schriftlichen Konversationen statt. Eine Lösung für diese enorme aber verständliche Einschränkung bei den angebotenen Systemen zu finden, gestaltet sich schwierig, wie im Abschnitt 4.4.1 erläutert wird.

### 3.3.3 Pronomen

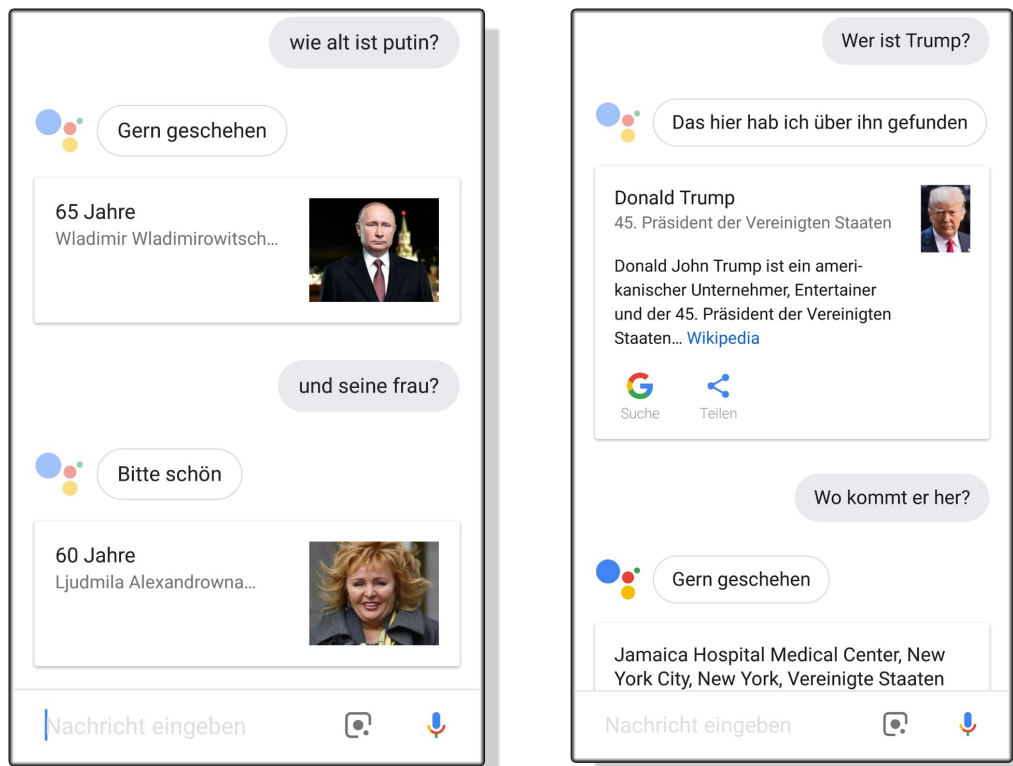
Pronomen, im Deutschen auch Fürwörter genannt, sind ein wichtiger Bestandteil der Grammatik und werden häufig bei Konversationen verwendet. Sie ersetzen ein bestimmtes Nomen und erlangen erst ihre korrekte Bedeutung, wenn sie einem gewissen Kontext zugrunde liegen. Dies deutet darauf hin, dass Pronomen zu den *deiktischen* Begriffen zählen (näher erläutert im Abschnitt 3.1). Allerdings ist dies in Fachkreisen umstritten, da sie ebenfalls der Gruppierung der *Anaphern* zugehörig sind. Linguisten sprechen bei Pronomen in der ersten und zweiten Person von einer *Personaldeixis*. Pronomen werden in mehrere Kategorien mitsamt den jeweiligen Deklinationen unterteilt, welche von einem Bot-System erkannt und zugeordnet werden müssen, um den Kontext korrekt weiterführen zu können.

- *Personalpronomen* sind *ich, du, er, sie, es, wir, ihr, sie* und ihre deklinierten Formen wie *mir, dir, etc.*
- *Possessivpronomen* zeigen eine Zugehörigkeit an. Es gibt begleitende Possessiva (Das ist *meine* Masterarbeit.) und als Ersatz dienende (Das ist *meine*).
- Mit *Relativpronomen* werden Relativsätze eingeleitet (Die Universität sucht die Studentin, *die* diese Masterarbeit geschrieben hat.).
- Ebenfalls zu nennen sind *Reflexivpronomen, Interrogativpronomen, Indefinitpronomen, und Demonstrativpronomen*, welche aber in der technischen Entwicklung eines Chatbots nur eine untergeordnete Rolle spielen.

Die Definition im ersten Satz dieses Abschnittes lässt erkennen, dass der Chatbot

---

<sup>1</sup><https://dialogflow.com/>



**Abbildung 3.1:** Beispiel für Kontext-Verständnis in Bezug auf Personal- und Possessivpronomen bei *Google Assistant*.

dazu gezwungen ist kontextuelle Gegebenheiten miteinzubeziehen, um die korrekte Interpretation solcher Pronomen zu erheben. Die für Menschen selbstverständliche Verwendung ist in Abbildung 3.1 ersichtlich. Ebenso ist auf Verwechslungen in diesem Gebiet zu achten. Der folgende Satz soll als Beispiel dafür dienen: „Robert hatte einen Hamster. *Er* ist gestorben.“ Nun stellt sich natürlich die Frage, auf was sich das Personalpronomen *er* bezieht. Dies ist aber grundsätzlich eine Ungenauigkeit des Benutzers, wo auch bei einer Mensch zu Mensch Konversation Fragen und Missverständnisse entstehen können.

Es sei darauf verwiesen, dass die Ausführung, wie Pronomen aus der Nachricht extrahiert und Problematiken der Art des vorigen Absatzes gelöst werden, Bestandteile des Themengebietes *Natural Language Processing*, kurz NLP, insbesondere des Bereiches *Natural Language Understanding*, kurz NLU, sind. Die meisten AI-Frameworks enthalten Ansätze in ihrem Angebot. Dennoch bieten sie noch keinen optimalen Service. Ab einer gewissen Komplexität entstehen Schwierigkeiten, welche im Abschnitt 3.4 angeführt werden.

### 3.3.4 Gesprächsübergreifendes Erinnerungsvermögen

Dem Chatbot ein Gedächtnis zu implementieren, um Zugriff auf ältere Intents, oder gar komplette Konversationen zu ermöglichen, ist ein noch äußerst unerprobtes Gebiet. Wenn dies auch aus kontextueller und programmiertechnischer Sicht noch nicht möglich ist und zuvor andere Problematiken gelöst werden müssen, so gibt es dennoch Möglichkeiten einen Ansatz eines Erinnerungsvermögens zu schaffen. Die einfachste Funktion eines Chatbot-Gedächtnisses wäre eine automatische Beglückwünschung am Geburtstag des Benutzers, sofern er diesen angegeben hat. Solch übergreifenden Aussagen können das Vertrauen und die Frequenz der Nutzung seitens des Users erhöhen. Dieses System lässt sich auch ideal bei dem vorgestellten Hotel-Bot umsetzen. Die Abfrage und Auflistung von bereits gebuchten Aufenthalten, geplanten Behandlungen oder deren Bewertungen wäre eine vorteilhafte Ergänzung in der Konversation. Dies wäre eine Weiterführung der globalen Variablen, die in Abschnitt 3.3.1 vorgestellt wurden.

### 3.3.5 Context Aware Computing

*Context Aware Computing*, also die Entwicklung von Kontext-sensitiven Systemen, erhielt über die letzten Jahre einen immer höheren Stellenwert in der IT-Branche. Die Qualität des Programms soll mit Hilfe von Informationen über den Benutzer, dessen Umgebung und Aktivitäten gesteigert werden. Dies kann von einfachen Methoden wie Shopping-Angebote in der Nähe des Users, bis zu komplexen Systemen mit vielen Elementen des situationsabhängigen Kontextes bei einer Chatbot-Konversation reichen. In den Begriff *Context Aware Computing* fallen auch Systeme, die wissen „was“ sie sind und wie sie verwendet werden. Beispiele dafür wären, dass sich die Orientierung der Seite ändert, wenn das Tablet gedreht wird, oder sich eine Navigationsanzeige je nach Geschwindigkeit und Licht-Situation umstellt.

## 3.4 Schwierigkeiten

In Laufe einer Konversation führen viele Eingaben zu Schwierigkeiten, die störende Folgen für die Kontext-Verarbeitung und das weitere Gespräch mit sich ziehen können. Neben dem Versuch sie durch technische und kommunikative Maßnahmen (siehe Abschnitt 3.5) vorbeugend zu umgehen, ist es für einen Entwickler notwendig, sich mit den möglichen Problematiken vertraut zu machen und gezielte Szenarien zu planen, um von den Schwierigkeiten zurück zu dem geplanten Konversationsverlauf zu gelangen. Es werden in diesem Kapitel die gängigsten Erschwernisse bei der Auffassung und Verarbeitung von Kontext dargestellt.

### 3.4.1 Bot Amnesia

Die Mehrzahl an Bots konzentrieren sich auf ein typisches Anfrage/Antwort-Schema. Dabei erhält jede neue Anfrage einen frischen Kontext und oftmals werden die alten Variablen gelöscht. Menschen sind dazu fähig sich in Konversationen über ein Thema zu unterhalten, zwischendurch zu einem komplett anderen zu wechseln und danach ohne Probleme zum Ersten zurückzukehren.

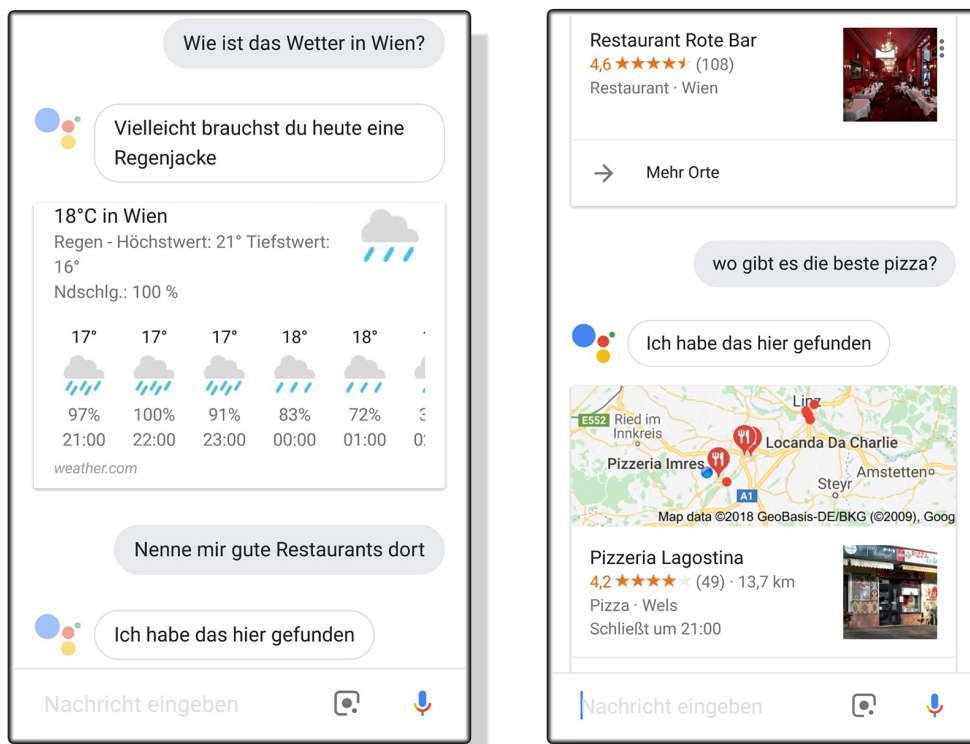


Abbildung 3.2: Beispiel für *Bot Amnesia* bei *Google Assistant*.

Im Gespräch der Abbildung 3.2 wird der Dienst *Google Assistant* nach dem Wetter in Wien gefragt. Dies gibt er dann mit einer anschaulichen Anzeige zurück. In der nächsten Anfrage wird der deiktische Begriff „dort“ verwendet, der sich auf die Stadt Wien bezieht. Auch dies meistert der Chatbot mit einer Auflistung von verschiedenen Restaurants. Geht man jedoch eine Stufe weiter lässt sich der typische *Context-Loss* erkennen. Grundsätzlich ist die Aussage nicht komplett unpassend. Er gibt die Pizzerien rund um die *Geolocation* an. Da die Frage jedoch direkt nach der Anzeige der Restaurants in Wien gestellt wurde und sich klar auf diese Lokalitäten bezog, ist es ein Fall von *Bot Amnesia*.

Robert Hoffer bezeichnet Systeme, die dermaßen agieren, als *Dory-Bots* [13], benannt nach der Fischdame aus dem Film *Findet Nemo*, welche an einer Schwäche des Kurzzeitgedächtnisses leidet. Hoffer ist der Meinung, dass die meisten der, zur Zeit der Veröffentlichung des Artikels existierenden Bots, dieses Phänomen vorweisen. Anzumerken ist, dass der Artikel im Juli 2016 online gestellt wurde. Laut Beispielen von Hoffer konnten die getesteten Bots nicht mal die erste Aufgabe lösen, die im vorigen Absatz beschrieben wurde. Dies lässt somit denn Schluss zu, dass es eine starke Entwicklung im Bereich der *Anaphorik-Verarbeitung* seit der Veröffentlichung gab.



### 3.4.2 Unerwarteter Input

Ein wichtiger Aspekt, der den Chatbots häufig Schwierigkeiten bereitet, ist das Aufkommen von unerwarteten Eingaben des Benutzers. In dieser Arbeit sind unerwartete Eingaben definiert als Inputs, die nicht in der Planung des optimalen Konversationsverlauf enthalten sind. Beispiele dafür sind:

- Danksagung oder Beleidigungen,
- Bilder, Videos oder Audio-Dateien,
- einzelne Emojis,
- Buttons aus früheren Abschnitten,
- Änderung der Meinung und
- Smalltalk.

Zusammenfassend ist zu diesen Punkten zu sagen, unabhängig von der Art ihrer Verarbeitung, dass darauf geachtet werden muss, dass der momentane Kontext und dessen Variablen erhalten bleiben und die Konversation nach einer solchen Unterbrechung ohne Rückschritte fortgeführt werden kann. Natürlich ist es möglich, dass die in der Liste angegebenen Aspekte nicht als unerwartete Eingaben zählen, da für die jeweiligen Zwecke Systeme und Lösungen eingebaut wurden und diese Nachrichten verarbeiten können.

#### Zwischenrufe, Smalltalk und Buttons

Die Problematik mit Danksagungen, Beleidigungen und Smalltalk ist am einfachsten durch ein Set an Intents, die diese Szenarien abdecken, zu lösen. Für diese stehen auch Templates und Sammlungen in verschiedenen Sprachen online zur Verfügung. Manche Anbieter zur Chatbot-Entwicklung, wie *Dialogflow*, führen eine umfangreiche Intent-Auflistung zum Thema Smalltalk im System.

Wird ein Button aus einem früheren Abschnitt mit anderem Kontext aktiviert und dessen Text in die Konversation gesendet, entstehen nachvollziehbare Probleme, welche auch dementsprechend schwierig entgegenzuwirken sind. Dies fällt dann größtenteils in die selbe Kategorie der sinnlose Eingaben seitens des Benutzers wie eine wahllose Kette von Buchstaben und endet im Zweifelsfall mit einer *Default-Antwort*.

#### Dateien und Emojis

Bei Dateien jeglicher Art und Nachrichten, die nur einen Emoji enthalten ist Vorsicht geboten. Ist der Bot nicht auf die Analyse des Mediums ausgelegt, können Missverständnisse und ungewollte Gespräche die Folge sein. Die Ansicht der Bedeutung von Emojis unterscheidet sich von Mensch zu Mensch. Zusätzlich spielen Alter und Herkunft eine Rolle, was die Untersuchung von *HighSpeedInternet* [11] ergibt und in der Abbildung 3.3 aufgezeigt wird. Es werden die drei höchstens Prozentzahlen der angegebenen Bedeutungen aufgelistet. Da kein einheitliche Abbildung der jeweiligen Emotion oder Gegenstands festgelegt wurde, unterscheiden sich Emojis je nach Endgerät und Messenger-Dienst, was zu weiteren verschiedenen Ansichten der Benutzer führt.

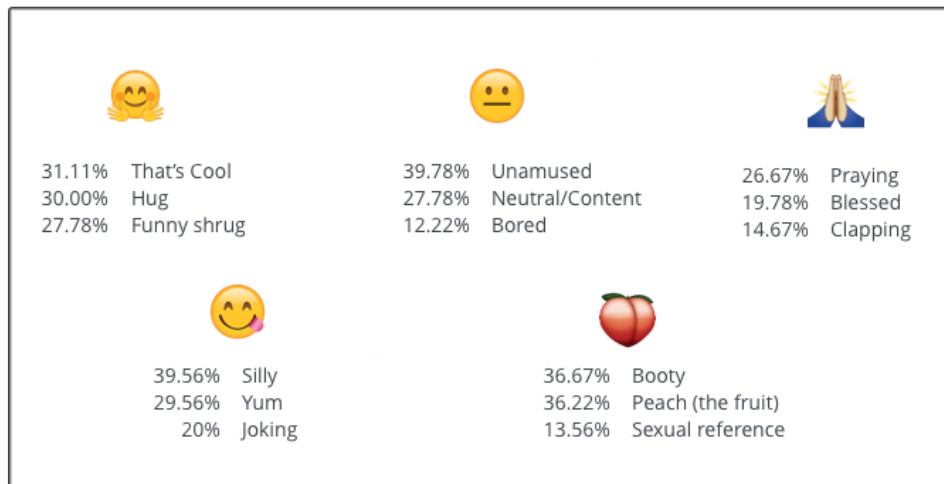


Abbildung 3.3: Analyse von HighSpeedInternet.com über Bedeutung von Emojis [11].

### Änderungswünsche

Änderungswünsche des Benutzers können den Chatbot ebenfalls vor eine Herausforderung stellen. Man nehme bei dem vorgestellten Hotel-Bot an, dass der Kunde über das System die Bestellung für sein Frühstück, welches in sein Zimmer geliefert wird, aufgibt. Auf die Frage, welchen Kaffee er möchte, antwortet er beispielsweise mit der Nachricht „Espresso“. Danach werden die Feinheiten zu Wurst und Käse besprochen. Für den Chatbot scheint alles verständlich zu sein, nur dann folgt zum Beispiel diese Aussage: „Eigentlich hätte ich doch lieber einen Cappuccino.“. Werden keine Funktionen für diesen Fall implementiert, ist der Benutzer gezwungen den Bestellvorgang neu zu starten, falls er auf die Änderung besteht.

Bei derart mehrstufigen Vorgängen gibt es zwei Ansätze zur Lösung. Vor den ersten Besprechungen der Bestellungen informiert der Bot den User darüber, dass er mit Hilfe eines *Keywords* wie „Änderung“ ein Szenario starten kann, das ihm ermöglicht einzelne Abschnitte der Bestellung schrittweise zu modifizieren. Dabei ist natürlich ausschlaggebend, dass nach der erfolgreichen Änderungen der Vorgang ungestört fortgeführt werden kann, ohne dass bereits fixierte Aspekte verloren gegangen sind. Ein anderer Ansatz wäre es, für alle Punkte der Bestellung Intents anzulegen, die diese Fälle von Aussagen abdecken und verarbeiten. Diese Methode ist verständlicherweise nicht derart simpel zu warten und erweitern, wie die davor beschriebene Art, dennoch kann sie, ordentlich ausgeführt, durchaus effektiv sein.

### 3.4.3 Fehlerhafte Auffassung

In seltensten Fällen sind Systeme fehlerfrei. Bei komplexen Anfragen oder gesamten Konversationen kann es zu fehlerhaften Auffassungen des Kontextes kommen, die das Gespräch vom *Happy Path* entfernen. Im linken Abschnitt der Abbildung 3.4 ist eine der genannten Problematik zu erkennen. Der Bot *Mitsuku* nimmt an, dass sich beide Informationen auf „my mother“ beziehen und speichert die zweite Information in einer

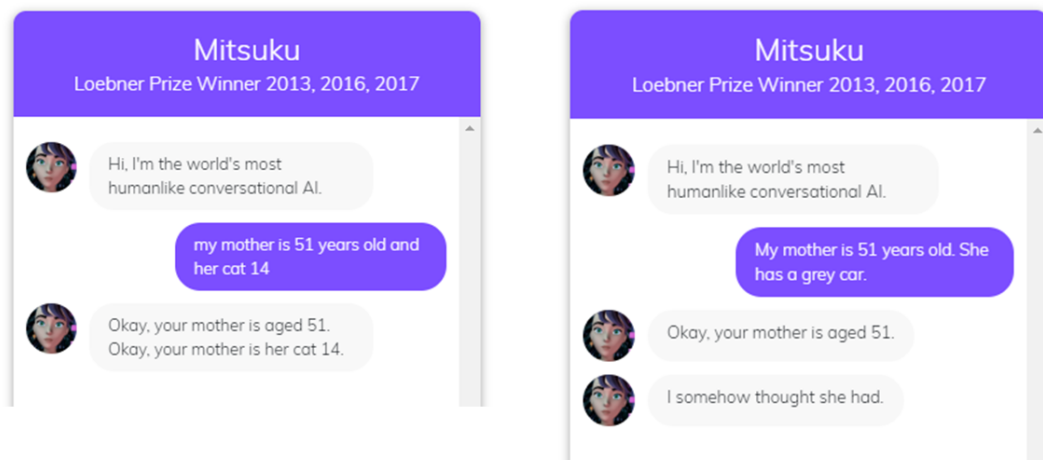
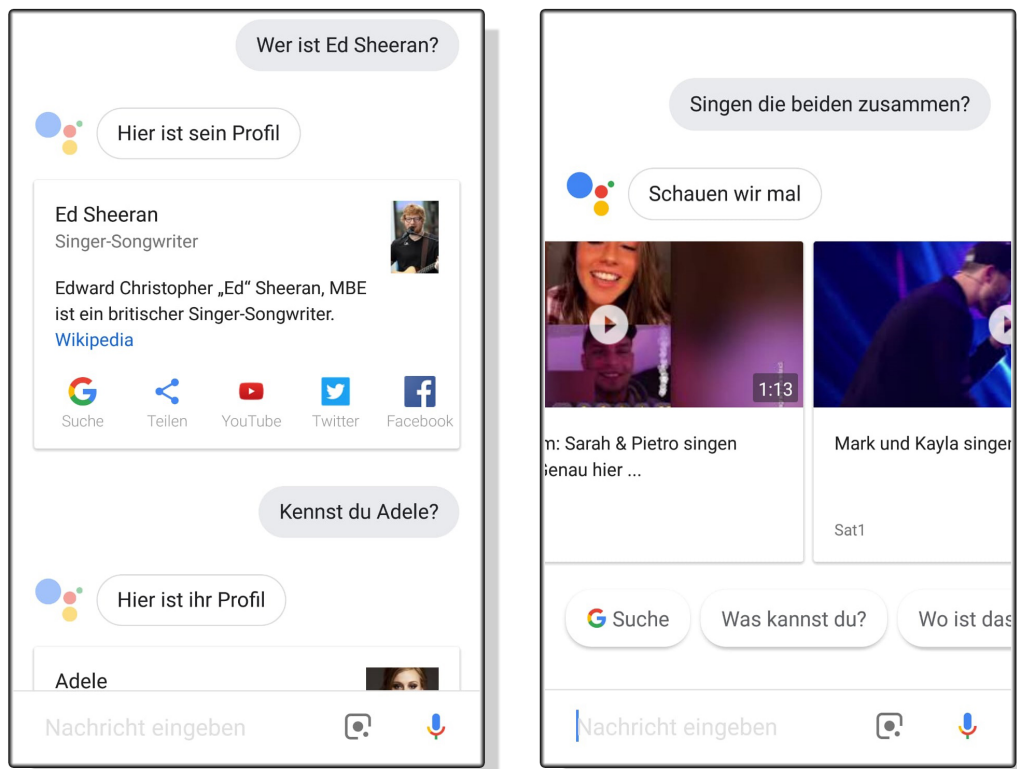


Abbildung 3.4: Konversationen mit dem Chatbot *Mitsuku*.

unpassenden Weise. An den meisten Situationen, wo ein falscher Kontext aufgefasst wird, sind zu lange Sätze mit mehreren Aspekten ausschlaggebend. Im Falle des Bots *Mitsuku* ist es möglich, mehrere Informationen in einer Nachricht zu senden und dennoch eine korrekte Antwort zu erwarten, wenn diese mit einem Punkt getrennt sind. Das System registriert diese dann als zwei Aussagen und antwortet auch dementsprechend unabhängig darauf. Es wird kein Kontext zwischen den beiden geschlossen. Wird der Bot daraufhin gefragt, wer ein graues Auto besäße, wird kein Schluss auf den Teil „my mother“ gezogen.

#### 3.4.4 Unzureichende Verarbeitung

Die Verarbeitung des Kontextes bezieht sich nicht rein auf das korrekte Erheben, sondern auch auf die weiteren Funktionen und der zeitlich abgestimmten Löschung. Werden unterschiedliche Kontext-Variablen nicht geordnet und rechtzeitig entfernt, kann das System fehlerhafte Informationen zur Verfügung stellen. Angenommen es wird bei dem vorgestellten Hotel-Bot eine „Frühstück-Aufs-Zimmer-Service“-Variable erstellt, die anzeigt, dass sich die Konversation momentan zu diesem Thema befasst und die eingegebenen Anfragen daraufhin gelenkt werden sollen. Der Benutzer wechselt darauf zum Thema „Betthupferl-Service“ und das System speichert wiederum diese Variable als Kontext ein. Möchte das Programm den vorigen Kontext noch nicht vergessen, weil beispielsweise der Vorgang noch nicht abgeschlossen war, dann ist es unbedingt von Nöten eine Priorität der Variablen zu arrangieren. Fragt der Benutzer als nächsten Schritt indirekt nach den Kosten des „Betthupferls“, darf nicht der Preis des vorher besprochenen Services angezeigt werden. Unterstützt der Bot keinen weitgehenden Kontext, dass auf den vorigen Vorgang zurück gelenkt werden kann, dann muss das System fähig sein, nicht gebrauchte Variablen zeitgemäß aus dem Vorgehen zu entfernen, um ungewollte Überschneidungen zu vermeiden.



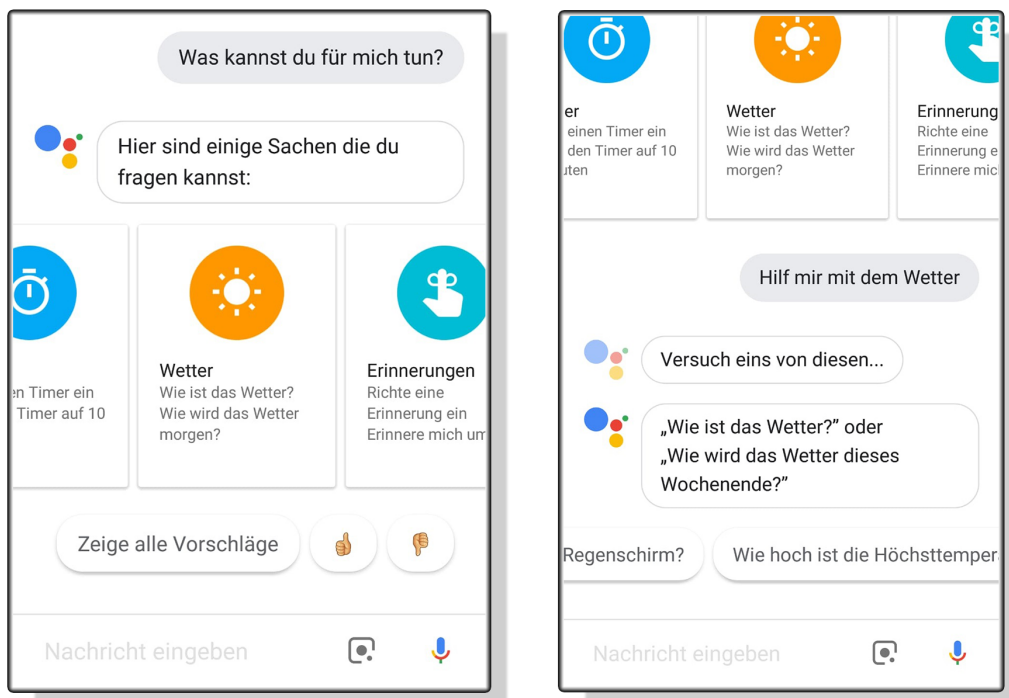
**Abbildung 3.5:** Beispiel für Fehlschlag der Verbindung von zwei unterschiedlichen Kontexten bei *Google Assistant*.

### 3.4.5 Verbindung von unterschiedlichen Kontexten

Selbst die bekanntesten und fortgeschrittensten Systeme scheitern noch größtenteils an der Verbindung von zwei Kontexten. Meist wird keiner der beiden in der folgenden Antwort erwähnt, oder das System bezieht sich nur auf den zuletzt aufgerufenen Aspekt. Das Beispiel in Abbildung 3.5 soll diese Problematik verdeutlichen. Zuerst wird das System nach dem Musiker *Ed Sheeran* gefragt. Darauf folgt eine Anfrage zu der Sängerin *Adele*. Bei einem Gespräch von Mensch zu Mensch wären die Aspekte der weiteren Frage „Singen die beiden zusammen?“ geklärt. Das Programm sieht dies dennoch als neuen Kontext, reagiert mit Treffern auf „singen zusammen“ und ignoriert den, in diesem Fall bestimmenden und maßgeblichen Teil „die Beide“.

## 3.5 Kommunikative Maßnahmen

Bei der Planung von möglichen Verläufen der Konversation und der unterschiedlichen Situationen können neben den technischen Implementierungen auch kommunikative Maßnahmen eingesetzt werden, um dem Benutzer ein höheres Maß an kontextuellem Verständnis vorzuspielen. Dies wird vorrangig damit erreicht, dass der Entwickler Situationen zu vermeiden versucht, in denen Schwierigkeiten wie *Bot Amnesia* auftreten



**Abbildung 3.6:** Beispiel für Vorschläge zur Verwendung bei *Google Assistant*.

können oder vermehrt Probleme in der Testphase festgestellt wurden.

Die Qualität und der Erfolg eines Chatbots hängt unter anderen Aspekten davon ab, wie viel Aufwand beim Design der Konversationen geleistet wird. Es spielt eine maßgebliche Rolle, wie das System dem Benutzer die Informationen darbringt. Der bestmögliche Verlauf wäre, wenn der *Happy Flow* ohne gröbere Abweichungen und Umwege durchlaufen wird und der Benutzer dabei trotzdem das Gefühl hat, dass er die Konversation geleitet hat. Im Hintergrund wurden aber geschickte Schritte durchgeführt, um dieses Szenario zu erzielen. Diese Mittel, besonders in Bezug auf Kontext, werden in diesem Abschnitt erläutert.

### 3.5.1 Angebote

Die offensichtlichste Methode einem Benutzer Angebote zu unterbreiten, besteht darin ihm Aussagen direkt als *Quick-Reply-Buttons* anzubieten, wie in Abbildung 3.6 ersichtlich. Wegen dem großen Umfang an Features, die der Dienst *Google Assistant* anbietet, ermöglicht er dem Benutzer zuerst eine Übersicht an Möglichkeiten mit Hilfe eines interaktiven Karussells. Nachdem der Benutzer einen dieser Vorschläge aktiviert hat, bietet der Bot direkte Aussagen an, mit welchen Möglichkeiten dieses Feature beispielsweise genutzt werden kann. Indem das System somit eine ungefähre Richtung vorgibt, wie die Anfragen formuliert sein sollten, können eventuelle Schwierigkeiten mit missverständlichen Anweisungen und folgenden kontextuellen Fehlschritten, abgefangen werden. Diese

Methode ist auch eine ideale Art um das Gespräch am Laufen zu halten.

Eine weitere Möglichkeit um *Dead Ends* der Konversation, die oftmals erst zu den Problemen führen, zu vermeiden ist, bei nicht weiterführenden Aussagen eine Frage anzufügen. Beispielsweise könnte an die Aussage „Ich hoffe ich konnte Ihnen behilflich sein.“ ein Zusatz wie „Möchten Sie vielleicht wissen welche Sightseeing-Möglichkeiten sich in der Nähe des Hotels befinden?“ angefügt werden, um das Gespräch fortzusetzen. Falls eine umfangreiche Hilfestellung, die ebenfalls bei solchen Situationen zielführend und kompensierend sein kann, implementiert wurde, sollte dies dem User gleich zu Beginn der Konversation, am Besten im *Opt-In*, mitgeteilt werden. Die Hilfe sollte mit den vielfältigsten Aussagen aktiviert werden können. Ebenso ist ein fixierter Punkt im Menü vorteilhaft.

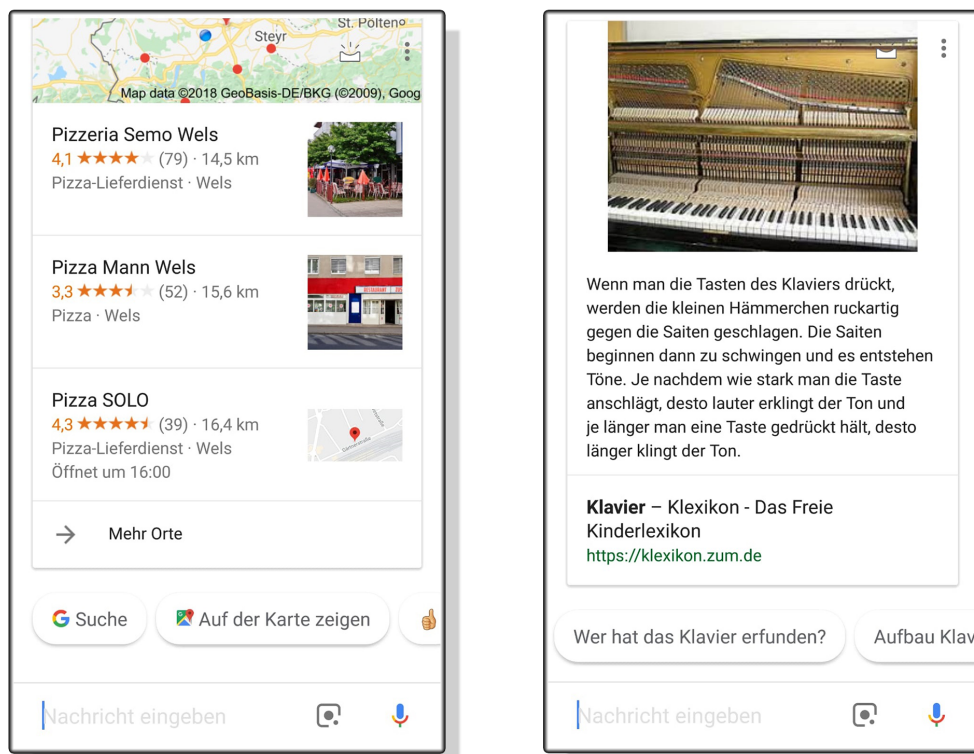
### 3.5.2 Rich Messages

*Rich Messages* sind alle Elemente in einer schriftlichen Konversation mit Hilfe eines technischen Gerätes, die über eine textuelle Nachricht hinausgehen. Dazu gehören Buttons, Dateien jeder Art, Bildergalerien und die Einbindung von externen Seiten. Diese können nicht nur das Konversations-Design eines Chatbots verbessern, sondern auch dafür sorgen, dass gewisse folgende kontextbezogene Anfragen nach einem *Request* vorweg angezeigt werden, um Problematiken wie die sogenannte *Bot Amnesia* (siehe Abschnitt 3.4) vorzubeugen.

Im linken Verlauf der Abbildung 3.7 wurde dem Service *Google Assistant* die Anfrage „ich möchte Essen bestellen“ geschickt. Daraufhin wird keine Aussage wie „Wo oder Was möchten Sie essen?“ retourniert, die eventuell Schwierigkeiten hervorrufen könnte, sondern sofort eine Karte mit markierten möglichen Lieferanten in der Umgebung geschickt. Darunter werden diese Punkte der Anzeige auch in interaktiven Kacheln aufgelistet, die aktiviert werden können, um weitere Informationen zu erhalten. Die Konversation wird ebenso weiter vorangetrieben, indem unter der Ausgabe auch Buttons mit hilfreichen Vorschlägen für die nächste Anfrage angeboten werden. Somit wird versucht etwaige Probleme mit unverständlichen, kontextuell auf die momentane Situation bezogene Fragen zu vermeiden. Dieses Vorgehen dient somit als Prävention für ungewollte und negativ wirkende fehlerhafte Ausgaben.

Im rechten Verlauf der Abbildung 3.7 werden die möglichen weiteren, kontextuell abhängigen Anfragen deutlicher aufgelistet. Dem Service *Google Assistant* wurde die Frage „wie funktioniert ein Klavier?“ geschickt. Daraufhin antwortet der Bot mit einem Abschnitt einer Internetseite, die diese Frage beantwortet. Darunter bietet er mit Hilfe von sogenannten *Quick Replies* weitere Beispiele wie „Wer hat das Klavier erfunden?“ und „Aufbau Klavier“ an.

Welche Arten von *Rich Messages* verwendet werden, ist dem Entwickler überlassen. Dennoch ist eine gewisse Mäßigung der Häufigkeit gefragt, da eine Überstimulierung kontraproduktive Ergebnisse nach sich ziehen kann. Die optimale Häufigkeit der Nutzung hängt von dem Zweck des Chatbots ab. Steht die abwechselnde schriftliche Kommunikation im Vordergrund, können zu viele Dateien und Anzeigen diese ungewollt unterbrechen. Sollte es sich, wie bei *Google Assistant*, um ein Anfrage-Antwort Schema mit ständig wechselnden Themengebieten handeln, dann ist ein häufiger Einbau von *Rich Messages* vorteilhaft, um eine umfangreiche Antwort zu liefern.

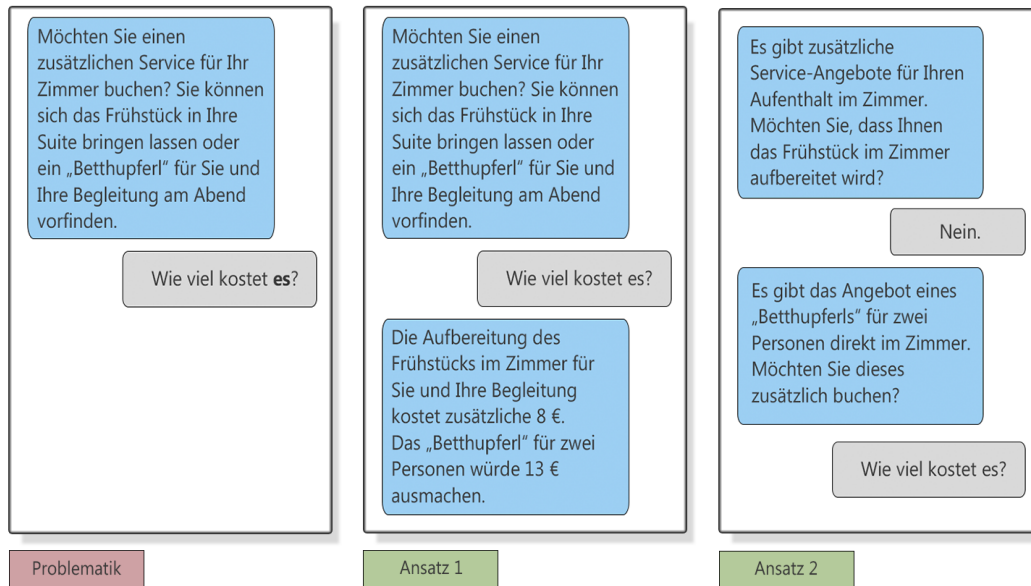


**Abbildung 3.7:** Beispiel für Verwendung von *Rich Messages* zum Vorantreiben der Konversation und Abdeckung von nächsten Kontext-abhängigen Anfragen bei *Google Assistant*.

### 3.5.3 Korrekturen

Es kann zu Situationen kommen, wo das System den Kontext fehlerhaft auffasst und die Konversation in eine vom Benutzer ungewollte Richtung lenkt und eventuell ein belastender Neustart erforderlich wird. Dies kann durch eine unzureichende Formulierung des Benutzers oder einer falschen Auffassung des Systems passieren. Solche Situationen kann man mit kommunikativen Maßnahmen im Design vorbeugen.

In Abbildung 3.8 ist eine solche Problematik dargestellt. Es ist unklar, worauf sich das Pronomen *es* bezieht. In *Ansatz 1* ist eine Lösungsstrategie angeführt. Zur Sicherheit werden keine Aktionen, wie beispielsweise das Nachfragen des Pronomens, durchgeführt, sondern direkt alle möglichen Komponenten erläutert. Im *Ansatz 2* wird diese Strategie weiter verfeinert. Es wird vorausschauend geplant, dass diese Hürde nicht entstehen kann. Es wird zuerst die Möglichkeit des Frühstücks geliefert, auf eine Antwort gewartet und danach der zweite Service vorgeschlagen. Dies ist natürlich nur möglich, wenn der Umfang der Auflistung nicht mehr als drei Einträge beträgt. Sollte dies der Fall sein, ist es sinnvoll diese Konversation entweder mit *Ansatz 1* durchzuführen, oder Einzelheiten erstmals komplett zu verbergen und dem Benutzer eine Nachricht wie „Möchten Sie einen zusätzlichen Service für Ihr Zimmer buchen?“ zu senden. Dies führt höchstwah-



**Abbildung 3.8:** Beispiel zur Vorbeugung von kontextuellen Schwierigkeiten mit dem vorgestellten Beispiel eines Hotel-Bots.

scheinlich zu einer Verneinung oder der Anfrage, welche es denn gibt. Auf die zusätzliche Frage der Kosten, kann noch immer auf *Ansatz 1* zurückgegriffen werden, falls der Benutzer keine klare Aussage gibt.

Solche Problematiken sollten in der Planung eines Chatbots, der mit dem Benutzer einen gewissen Dienst durchläuft, bei dem Störungen und ein Neustart äußerst unangenehm wären, berücksichtigt werden. Im Gegensatz dazu können Systeme, die für eher lockere Unterhaltungszwecke geschaffen wurden, auf eine andere Methode zurückgreifen. Da bei den meisten riesigen Umfängen dieser Bots Ansätze, wie im vorigen Absatz erläutert, kaum für jeden Aspekt umsetzbar sind, sollten diese auf eine Entschuldigung oder ein Einverständnis mit Bitte um Korrektur zurückgreifen. Diese Handlung wird auch von der 3-fachen Loebner-Preis-Trägerin *Mitsuku* verwendet, wie in Abbildung 3.9 ersichtlich. Sie ist erreichbar in *Facebook Messenger*, *Kik* und weiteren Messenger-Diensten. Man kann sich ebenso im Browser<sup>2</sup> mit ihr unterhalten.

### 3.5.4 Opt-In

Die erste Nachricht des Chatbots an den Benutzer, in dieser Arbeit als *Opt-In* bezeichnet, ist eines der wichtigsten Mittel für den Erfolg eines Bots. Eine Vielzahl der Benutzer wird nur dieses Statement lesen. Nach dieser Mitteilung entscheiden viele User bereits, ob sie den Dienst benutzen wollen, oder die Interaktion vorzeitig abbrechen. Ebenso ist das *Opt-In* seit Änderung der Datenschutz-Verordnung (siehe Abschnitt 3.6) der ideale Ort für die Klärung von rechtlichen Aspekten. Außerdem können durch konkrete und klare Angaben darüber, wofür dieser Bot programmiert wurde, im Vorhinein Miss-

<sup>2</sup>Der Chatbot *Mitsuku* ist unter folgendem Link erreichbar: <https://www.pandorabots.com/mitsuku/>





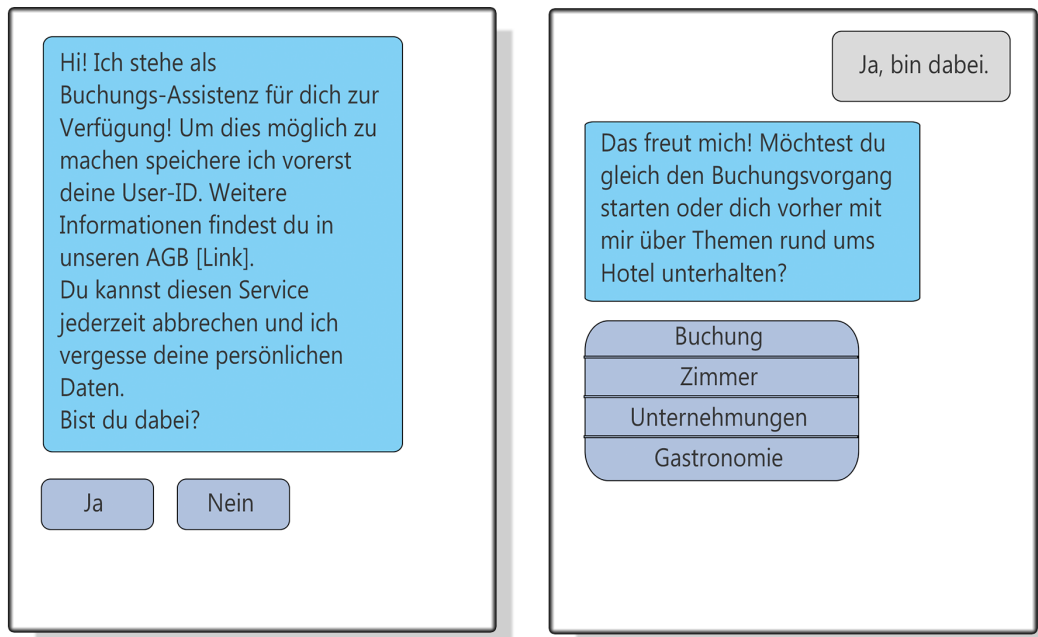
**Abbildung 3.9:** Beispiel für das Eingestehen einer Verwechslung des Chatbots *Mitsuku*.

verständnisse und mögliche Fehlentwicklungen der Konversation verhindert werden. Es gibt mehrere Aspekte, die in der Begrüßungs-Nachricht enthalten sein sollten, dennoch muss darauf geachtet werden, dass der Benutzer nicht von Unmengen an Text überflutet wird.

Laut den Psychologen Janine Willis und Alexander Todorov bilden sich Menschen nach einem Zehntel einer Sekunden den ersten Eindruck [9]. Wie auch in der menschlichen Kommunikation ist es für den Erfolg eines Chatbots von Nöten, einen stabilen und vorteilhaften ersten Eindruck zu hinterlassen. Durch die Maßnahmen, die im Rahmen der DSGVO (siehe Abschnitt 3.6) erforderlich sind, wird das *Opt-In* in zwei Abschnitte unterteilt. Grundsätzlich sollten die folgenden Aspekte umgesetzt und sich auch auf diese beschränkt werden. Die einzelnen Punkte sollten ebenfalls nicht mehr als einen Satz umfassen, da sonst die gesamte Länge ein kritisches Maß erreicht. Außerdem sollten sie, wenn möglich, in einem Absatz geschrieben werden, da mehrere, hintereinander folgende Nachrichten häufig als ungewollter Spam angesehen werden. Der erste Abschnitt umfasst folgende Punkte:

- Die Aufgabe des Chatbots.
- Welche Daten erhoben und wie diese verwendet werden.
- Ein Link zu den Allgemeinen Geschäftsbedingungen.
- Die Rechte in Bezug auf Datenschutz.
- Eine Einwilligung des Benutzers.

Nach Bestätigung des Users folgt der zweite Abschnitt mit einem Angebot an möglichen nächsten Schritten. Dieses kann mit Hilfe von Text und Buttons eingebaut werden. Ebenso sollte ein Hinweis gegeben werden, falls dies implementiert wurde, wie eine Hilfestellung während der Konversation verlangt werden kann (erläutert in Abschnitt 3.5.1). In der Abbildung 3.10 ist ein mögliches *Opt-In* des vorgestellten imaginären Hotel-Bots



**Abbildung 3.10:** Beispiel für ein *Opt-In* des vorgestellten Hotel-Bots.

zu sehen. Der Bot stellt sich innerhalb einer Nachricht vor, erläutert die Gegebenheiten bezüglich Datenschutz und fragt nach dem Einverständnis. Nach der Einwilligung bietet er dem Benutzer Informationen an, wie die Konversation weitergeführt werden kann. Zusätzlich werden *Rich Messages* in Form von Buttons angeboten.

### 3.6 Kontext-Verarbeitung und Datenschutz

Durch das Inkrafttreten der Datenschutz-Grundverordnung am 25. Mai 2018, im Folgenden als DSGVO bezeichnet, ändern sich weltweit die Vorgänge der Verarbeitung von personenbezogenen Daten. Dies betrifft wie erwähnt nicht nur, wie häufig angenommen, die Konzerne und Vereine der Europäischen Union, sondern auch alle Unternehmen auf den verschiedensten Kontinenten, die Daten von europäischen Bürgern verarbeiten. Dieser Beschluss ist unabhängig von Bekanntheit, Anzahl der Mitarbeiter und Einkommen. Bei Missachtung und Verstoß drohen hohe Geldbußen, die bis zu 10 Millionen Euro betragen können [15]. Die DSGVO wirkt sich auch auf die Entwicklung von Chatbots und deren Handhabung von Datensätzen, die zur Verbesserung des kontextuellen Verständnisses von Bedeutung sind, aus. Die dazu relevanten Artikel in der Verordnung und deren Umsetzung im System werden in den folgenden Abschnitten erläutert.

### 3.6.1 Relevante Artikel

Die nachfolgenden Zitate und Inhalt der Zusammenfassungen werden von der Veröffentlichung des Unternehmens *intersoft consulting services* AG [10] bezogen.

#### Definition von Ausdrücken

Artikel 4 der DSGVO beinhaltet eine Reihe an Definitionen von Begriffen, die in der Verordnung regelmäßig verwendet werden. Für die relevanten Artikel in dieser Arbeit sind die Ausdrücke *personenbezogene Daten*, *Verarbeitung* und *Verantwortlicher* von Bedeutung.

- „*Personenbezogene Daten* sind alle Informationen, die sich auf eine identifizierte oder identifizierbare natürliche Person beziehen.“
- Mit *Verarbeitung* sind alle Vorgänge, die, mit oder ohne Hilfe von automatisierten Verfahren, ausgeführt werden und im Zusammenhang mit *personenbezogenen Daten* stehen, wie unter anderen das Erheben, Speichern und Auslesen.
- Der *Verantwortliche* ist eine natürliche oder juristische Person, Behörde, Einrichtung oder andere Stelle, die allein oder mit weiteren *Verantwortlichen* über die Art und den Zweck der *Verarbeitung* von *personenbezogener Daten* entscheidet.

#### Grundsätze der Verarbeitung und Einwilligung

In Artikel 5 werden einige Prinzipien zur Verarbeitung von personenbezogenen Daten erläutert, welche in diesem Abschnitt aufgelistet sind:

- „Personenbezogene Daten müssen auf rechtmäßige Weise, nach Treu und Glauben und in einer für die betroffene Person nachvollziehbaren Weise verarbeitet werden.“
- Die Daten müssen für festgelegte und eindeutige Zwecke erhoben werden und dürfen nicht für andere Vorhaben, die nicht vereinbart wurden, zur Weiterverarbeitung gereicht werden.
- Sie dürfen nur so lange wie benötigt gespeichert werden.
- Ebenso ist auf ein hohes Maß an Sicherheit für die Daten während der Speicherung und Verarbeitung zu achten.
- *Datenminimierung* hat hohe Priorität. Es soll nur das notwendige Maß an Daten gesammelt werden, die für diesen Zweck benötigt werden.

Der Verantwortliche ist für die Einhaltung dieser Regeln verantwortlich und muss diese Erfüllung auch nachweisen können. In Artikel 6 und 7 wird erläutert, dass die Verarbeitung erst ihre Rechtmäßigkeit erhält, wenn der Benutzer eine klare Zustimmung dafür gibt. Diese Zustimmung muss der Verantwortliche ebenfalls nachweisen können.

#### Informationspflicht

Laut Artikel 13 besteht nun die Pflicht einen Benutzer ausführlich zu informieren wann und welche personenbezogenen Daten erhoben werden. Bei der Entwicklung von Chatbots erläutert man einige Aspekte am besten im Opt-In (siehe Abschnitt 3.5.4). Die verbliebenen Pflichten sollten über die Verlinkung der AGB erfolgen.

Dem Benutzer muss der Name und die Kontaktdaten des Verantwortlichen, die Zwecke der Erhebung und die Dauer der Speicherung mitgeteilt werden. Ebenso besteht die Pflicht ihn auf die Rechte in Bezug auf seine persönlichen Daten hinzuweisen. Der User hat das Recht auf Auskunft darüber, welche Daten von ihm erhoben worden sind. Ebenso, dass diese Daten zu jeder Zeit nach Aufforderung gelöscht werden können (Artikel 17) und er die benötigten Einwilligungen für die Speicherung widerrufen kann (Artikel 21).

#### Datenübertragbarkeit

Der Benutzer hat durch Artikel 20 jederzeit das Recht seine personenbezogenen Daten, die für die Verarbeitung erhoben wurden, anzufordern. Dies muss in einer strukturierten, gängigen und maschinenlesbaren Form zur Verfügung gestellt werden.

#### Verletzung des Schutzes

Der Verantwortliche muss laut Artikel 33, im Falle einer Verletzung des Schutzes der gespeicherten Daten, die Gegebenheiten unverzüglich der Aufsichtsbehörde melden. Dies ist keine Verpflichtung, wenn die Verletzung voraussichtlich nicht zu einem Risiko für die Rechte des betroffenen Nutzers führt. Im Nachfolgenden Artikel 34 wird der Verantwortliche ebenso dazu verpflichtet, die betroffenen Personen über die Verletzung des Schutzes zu informieren, insofern ein hohes Risiko für ihre Rechte und Freiheit besteht.

### 3.6.2 Umsetzung

Bereits bestehende Systeme müssen daraufhin angepasst werden, um konform mit der Verordnung zu sein. Der erste Schritt für Entwickler ist, sich intensiv damit zu beschäftigen, welche Funktionen das System haben soll, beziehungsweise bereits hat. Daraufhin wird ersichtlich, welche Daten von Benutzern erhoben und für wie lange diese gespeichert werden müssen, um den Zweck zu erfüllen.

Das Opt-In (siehe Abschnitt 3.5.4) sollte mit kompakten Erläuterungen ergänzt werden. Diese sollten die Auskunft darüber, welche Daten wofür erhoben werden und das Recht, dass die Daten jederzeit gelöscht werden können, enthalten. Die weiteren Verpflichtungen, die im Bereich *Relevante Artikel* (siehe Abschnitt 3.6.1) erläutert wurden, finden am besten ihren Platz in den klar formulierten AGB, dessen Link ebenfalls im Opt-In gesendet werden sollte. Nach diesen Erklärungen und dem Senden des Links zu den AGB, ist die klare Zustimmung des Benutzers notwendig. Im Falle eines Chatbots kann dies mit einer kurzen Frage wie „Bist du damit einverstanden?“ und einer Ja oder Nein Antwortmöglichkeit als Buttons umgesetzt werden. Die Bedingung aus Artikel 20 der DSGVO (siehe Abschnitt 3.6.1), welche verlangt, dass eine Datei mit Auflistung der gespeicherten Daten bereitzustellen ist, kann als fixer Punkt im Menü angeboten werden.

Sollte sich der Nutzer im Laufe der Konversation oder zu einem späteren Moment dazu entschließen, seine Daten nicht mehr für die Verarbeitung zur Verfügung stellen zu wollen, muss die Möglichkeit gegeben sein, diese Daten im System zu löschen. Ebenso sollte ein erneutes Opt-In für bestehende Benutzer erfolgen und auch deren Zustimmung eingeholt werden.

# Kapitel 4

## Projekt

Dieses Kapitel behandelt alle Aspekte des korrespondierenden Projektes. Es wird die Aufgabenstellung vorgestellt und die Methoden der Umsetzung ausführlich erläutert. Ebenso werden die relevanten Teile des Projektes in Bezug auf Kontext und dieser schriftlichen Arbeit zusammengefasst. Vor dem Anführen der endgültigen Ergebnisse, werden die erwähnenswerten Schwierigkeiten und Problematiken während der Entwicklung behandelt.

### 4.1 Aufgabenstellung

Da die Popularität von Chatbots in allen Sektoren der Wirtschaft und Medien stark steigt, wäre ein System dieser Art auch eine Bereicherung für die *Fachhochschule Oberösterreich Campus Hagenberg*. Der Bot soll Interessierte am Studium, Studienbeginner und schon länger inskribierte Studenten eine Unterstützung sein. Es soll eine solide Konversation über die Studiengänge, den Alltag, die Infrastruktur und den Unterricht geführt werden können. Ebenso sollen mit Hilfe des im *Levis* hinterlegten ICS-Files Fragen wie „Was und wo ist meine nächste Stunde?“ beantwortet werden können. Zusätzlich sollte als weiterer dynamischer Inhalt die *Google Maps* API verwendet werden, um Details zur Route von einem beliebigen Standpunkt zum Campus angeben zu können.

### 4.2 Umsetzung

Das Projekt wurde im Rahmen der Lehrveranstaltungen *Thesis Project 1* und *2* im dritten und vierten Semesters des Master-Studiengangs *Interactive Media* an erwähnter Universität umgesetzt. Für die konkrete Aufgabenstellung, Recherche, Planung, Umsetzung und Überarbeitung wurde die Zeitspanne von August 2017 bis Juni 2018 benötigt.

#### 4.2.1 Dialogflow

*Dialogflow* ist eine 2014 veröffentlichte und 2016 von *Google* gekaufte Plattform, die die Entwicklung von Chatbots kostenlos anbietet. Das Framework wird über eine web-basierte Entwicklungsumgebung konfiguriert. Die Systeme können in 16 verschiedenen Sprachen aufgebaut und problemlos bei 17 unterschiedlichen Anbietern, wie beispiels-

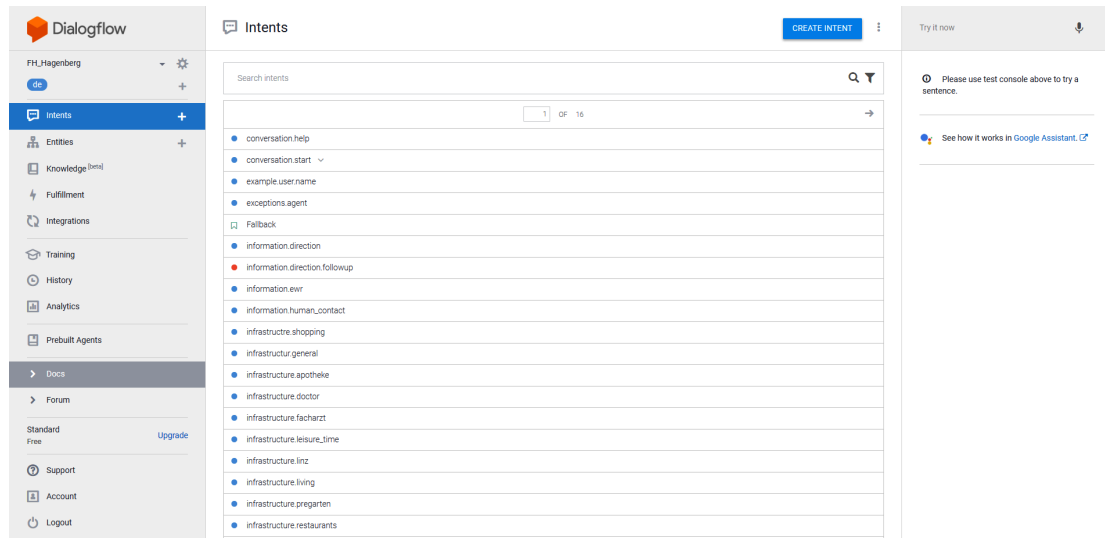


Abbildung 4.1: Aufbau der Konsole in *Dialogflow*. Stand September 2018.

weise *Facebook Messenger*, *Slack* und *Kik*, integriert werden. Zusätzlich steht eine Dokumentation und ein Forum für Benutzer zur Verfügung. Die Systeme im Hintergrund ermöglichen und nutzen *Machine-Learning* und *Natural Language Processing*. *Dialogflow*, vor der Übernahme als *API.AI* bekannt, wurde vom betreuenden Professor empfohlen und nach eingehender Recherche und Abwägung der verschiedenen angebotenen Features und Einschränkungen für die Umsetzung ausgewählt.

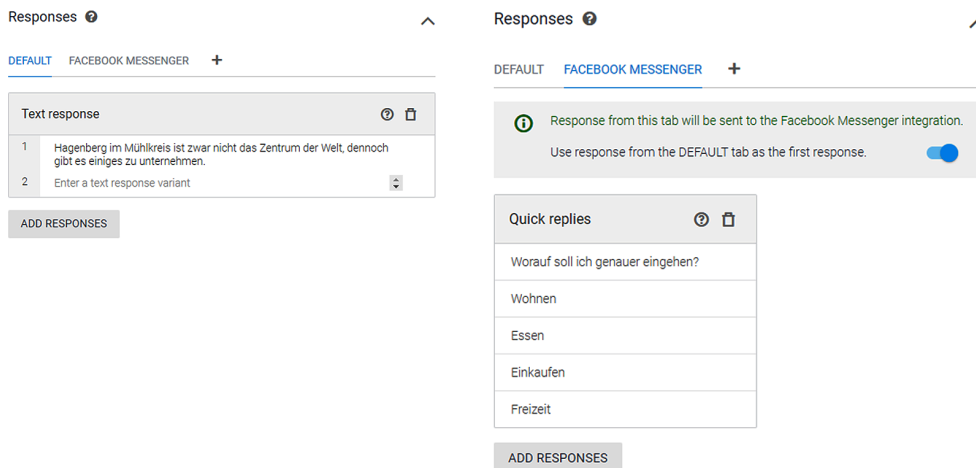
### Aufbau der Konsole

In der linken Leiste befinden sich alle relevanten Bereiche der Konsole. Angefangen mit Auswahl des Projekts, da die Möglichkeit besteht, mehrere Bots in einem Account zu erstellen, gefolgt mit der verwendeten Sprache, die als Abkürzung angegeben ist. Wichtige Aspekte für die Entwicklung befinden sich in den Punkten *Intents*, *Entities*, *Fulfillment* und *Integrations*. Auf das in Beta befindende Feature *Knowledge Bases* wird in dieser Arbeit nicht näher eingegangen, da es zum Zeitpunkt der Entwicklung noch nicht zur Verfügung stand. Danach befinden sich die Abschnitte *Training* und *History*, die ermöglichen Konversationen, auch solche, die auf integrierten Plattformen geführt werden, einzusehen und *Intents* und weitere Einzelheiten anzupassen. Die weiteren Ausführungen sind für organisatorische Zwecke und weitgehendst selbsterklärend.

### Intents

*Intents* bilden die Basis eines Chatbot-Systems. Es wird für jeweils eine mögliche Eingabe des Benutzers ein *Intent* angelegt. Bei *Dialogflow* bestehen diese aus folgenden Teilen:

- Benötigte *Input-Kontext-Variablen*, damit dieser *Intent* aufgerufen wird,
- *Output-Kontext-Variablen*, die nach dem *Intent* eingespeichert werden,



**Abbildung 4.2:** Verwalten von Antworten in einem *Intent* in *Dialogflow*.

- *Events*, die es ermöglichen, einen *Intent* ohne übereinstimmenden User-Input aufzurufen,
- Phrasen, die den *Intent* triggern und mit Hilfe von NLP ebenso abgewandelt aufgegriffen werden,
- Aktionen und Parameter, die angegeben werden können,
- Antworten des Chatbots und
- der mögliche Aufruf eines externen Webhooks für diesen *Intent*.

Welche Wertung diese *Intents* in einer mehrfachen Übereinstimmung einer Eingabe erhalten, hängt von den angegebenen *Input-Kontext-Variablen* und *Events* ab. Auf Ersteres wird im Abschnitt 4.3 näher eingegangen.

Die Antworten, die der Chatbot geben soll, werden im Abschnitt *Responses* verwaltet, der standardmäßig mit einem Fenster *Default* ausgestattet ist. Es können mehrere Antworten erstellt werden, die im Chat als einzelne Nachrichten geschickt werden. In jeder Text-Antwort können mehrere Ausführungen angegeben werden, die dann zufällig ausgewählt werden. Die *Default-Ausgabe* wird immer verwendet, wenn für die benutzte Plattform keine spezifische Antwort angegeben wird. Für diese Plattformen können zusätzliche Elemente erstellt und mit unterstützten *Rich Messages* befüllt werden. Ebenso ist es mit Hilfe einer angegebenen *Checkbox* möglich, die Antwort im *Default-Bereich* zuerst zu verwenden und die spezifische Ausgabe als zusätzliche Nachricht zu versenden, wie in Abbildung 4.2 ersichtlich ist.

## Entities

Mit Hilfe von *Entities* kann die Erkennung eines passenden *Intents* vereinfacht und verbessert werden. Eine *Entity* besteht aus mehreren Einträgen, die in *Dialogflow* Synonyme genannt werden. Diese Einträge bestimmen, welche Eingaben diese *Entity* auslösen. Die Verwendung von Uhrzeit ist beispielsweise eine gängige Verwendung von *Entities*.

*Dialogflow* definiert mehrere, häufig verwendete *Entities* standardmäßig. Es ermöglicht beispielsweise das Erkennen von allen möglichen Zeitangaben und die unterschiedlichen Schreibweisen. Ebenso können in *Dialogflow* eigene *Entities* erstellt werden, was folgendes Beispiel aufzeigt.

Im Falle des vorgestellten Projektes wurde eine *Entity* mit der Bezeichnung *courses* erstellt. Diese beinhaltet alle möglichen Abkürzungen und Termini der unterschiedlichen Studiengänge an der Fachhochschule. Wird in einer Trainingsphrase im *Intent* ein Wort als *Entity courses* markiert, werden alle Benutzer-Eingaben, die mit einem Eintrag aus der definierten *Entity* übereinstimmen, mit diesem *Intent* verarbeitet. Diese Methode wird häufig gewählt, um unnötige Arbeit einzusparen. Verständlicherweise geben Benutzer unterschiedliche Dinge verschieden ein und mit Hilfe von *Entities* können diese Szenarien adäquat verarbeitet werden. Im Projekt wird die *Entity courses* im Webhook verwendet, um zu den spezifischen *Intents* für den gewünschten Studiengang weiterzuleiten. Ebenso sind *Entities* sinnvoll, wenn eine bestimmte Gruppe in mehreren verschiedenen *Intents* ausführlich in den Trainingsphrasen mit den unterschiedlichsten Eingabe-Möglichkeiten behandelt werden müsste.

#### 4.2.2 Webhook

Als *Webhook* wird ein Verfahren zur Kommunikation zwischen Servern bezeichnet. Ein Aufruf besteht dabei aus einer *HTTP-POST* Anfrage, mit der Daten an den *Webhook* gesendet werden. Nachdem dieser die Daten verarbeitet hat, sendet er sein Ergebnis über die Antwort zurück. Die verwendete Programmiersprache hängt grundsätzlich von der Plattform ab, auf der der *Webhook* gehostet wurde. Die gängigsten Sprachen sind *JavaScript*, *PHP* und *Python*. In *Dialogflow* ist die Einbindung eines *Webhooks* möglich, um vom Benutzer eingegebene Daten weiter zu verarbeiten, beziehungsweise um Funktionalitäten durchzuführen, die durch die Möglichkeiten von *Dialogflow* selbst nicht abgebildet werden können.

Für den *Webhook* im Projekt wurde *JavaScript* verwendet. Die entwickelten Funktionen wurden auf *Google Cloud Platform* gehostet, die diesen Dienst bei geringen Datenraten kostenlos anbietet. Ab einer gewissen Menge von monatlichen Aufrufen wird ein Entgelt pro Zugriff berechnet. Die Plattform verwendet *Node.js* als *Framework*. *Dialogflow* stellt die Anforderungen an den *Webhook*, dass die Antwort eine maximale Datenmenge von 64 Kilobyte nicht überschreiten darf. Des Weiteren wird ein Aufruf auf maximal 5 Sekunden Länge beschränkt und bei Überschreitung als ungültig behandelt.

*Dialogflow* verwendet für die Kommunikation ein klar definiertes Format in Form eines JSON-Objektes. Die wichtigsten Elemente in der Anfrage sind der Name des *Intents*, der den Aufruf durchführt, eine Liste der gesetzten *Output-Kontext-Variablen* und eine *ID*, über die der Benutzer identifiziert werden kann. Als Antwort erwartet *Dialogflow* ebenfalls ein JSON-Objekt, mit welchem es möglich ist, Antworttexte und *Rich Messages* an den Benutzer zu senden, sowie sogenannte *Follow-up Intents* aufzurufen.

In der *Webhook-Datei index.js* des Projektes befinden sich zusammengefasst folgende Funktionen:

- Kompilierung des ICS-Files,
- Funktionen für unterschiedlichen Anfragen des Benutzers bezüglich diesen Files,
- Einbindung und Verwendung der *Google Maps API* und



- Weiterleitung zu kontextuell benötigten *Intents*.

### 4.2.3 Dynamische Features

Zur Erhöhung der Qualität des Services und Angebot von interessanten *Features* wurde die Verarbeitung des Stundenplans der Studierenden und die Berechnung der Route von einem beliebigen Standort nach Hagenberg im Mühlkreis entwickelt.

#### Stundenplan

Der Stundenplan an der *Fachhochschule Oberösterreich Campus Hagenberg* wird auf der Online-Plattform *Levis* angeboten. Hier ist es möglich den eigenen Kalender als ICS-File zu exportieren. Das File wurde im *Webhook* verwendet, um Fragen wie beispielsweise „Was und wo ist meine nächste Stunde?“ dynamisch und spezifisch zu beantworten.

Zusammengefasst wurde eine simple ICS-Verarbeitung implementiert, die die benötigten Daten aus dem File extrahiert. In *Dialogflow* wurden mehrere *Intents* erstellt, die unterschiedliche Benutzer-Eingaben erfassen und den *Webhook* aufrufen. Mit Hilfe von *If-Bedingungen* werden die jeweiligen *Intents* angesprochen und die extrahierten Daten des ICS-Files für die generierte Antwort des Chatbots verwendet. Im Folgenden wird auf die Implementierung im *Webhook* näher eingegangen und zusätzlich werden Code-Ausschnitte für besseres technisches Verständnis eingebaut.

Um die benötigten Daten aus dem ICS-File zu extrahieren, wird es zuerst nach `BEGIN:VEVENT` *gesplittet*. Dieser Begriff zeigt im ICS-File den Beginn des nächsten Eintrags an. Als Ergebnis erhält man alle Daten zum jeweiligen Event. Im Code wird das `Array entries` erstellt, das verwendet wird, um die extrahierten Daten zu speichern und in den folgenden Funktionen für die Ausgabe zu verwenden. Zusätzlich werden die Variablen für die benötigten Punkte angelegt und jeder Eintrag weiter verarbeitet, begonnen mit dem *splitten* bei jedem Zeilenumbruch. Danach werden die Zeilen mit *Split-Befehlen* geteilt und die Daten den entsprechenden Variablen zugeteilt. Zuletzt werden diese in das `Array entries` mit aufsteigenden Indexposition gespeichert.

```
function icsFileToEntries (content) {
  var eventTexts = content.split("BEGIN:VEVENT");

  var entries = new Array();
  var nextIndex = 0;

  var start, end, location, category, summary;

  eventTexts.forEach(function(eventText) {
    var lines = eventText.split("\n");
    ...
  })
  ...
}
```

Die Verwendung der generierten Daten wird mit Hilfe von Ausschnitten der Funktion `fhHagenberg (req, res)` erläutert. Diese besteht in erster Instanz aus mehreren *If-Bedingungen*, die jeweils einen *Intent* in Bezug auf eine Stundenplan-Anfrage abdecken. Die folgenden Erklärungen und Code-Teile stammen aus dem Abschnitt, der den

*Intent* `university.ical.nextlesson` verarbeitet und die Antwort für den nächsten verfügbaren Termin generiert.

Die Daten bezüglich des Datums und der Zeit des jeweiligen Eintrags des *Arrays* werden extrahiert und gespeichert. Mit Hilfe der aktuellen Zeit werden Vergleiche an- gestellt, um herauszufinden, ob der momentan bearbeitete Eintrag der nächste Termin vom jetzigen Datum aus ist.

```
if (yearNow < year) found = true;
if (yearNow == year && monthNow < month) found = true;
if (yearNow == year && monthNow == month && dayNow < day) found = true;
if (yearNow == year && monthNow == month && dayNow == day && hourNow < hour) found =
    true;
if (yearNow == year && monthNow == month && dayNow == day && hourNow == hour &&
    minuteNow < minute) found = true;
```

Die Antwort an den Benutzer wird aus den gewonnenen Daten und weiteren Funk- tionen zusammengestellt. Die Funktion `icalTimeStampToDay (date)` verwendet die Aspekte in Bezug auf das Datum und bringt sie in eine verwendbare Form.

```
answer = icalTimeStampToDay(foundEntry.start) + " hast du von " +
start + " bis " + end + " " +
icalCategoryArticleAkku(foundEntry.category) +
foundEntry.category + " " + foundEntry.summary +
" " + foundEntry.location + ".";
```

`icalCategoryArticleAkku (category)` gibt die jeweiligen Artikel der unterschiedli- chen Kategorien von Terminen zurück, wie im folgenden Programmtext ersichtlich ist.

```
if(category.includes("Termin") || category.includes("Vortrag") || category.includes
("Prüfungstermin")) {
    return "den ";
}
if(category.includes("Lehrveranstaltung") || category.includes("externe Nutzung")) {
    return "die ";
}
if(category.includes("Festival") || category.includes("Seminar")) {
    return "das ";
}
return "ein ";
```

In Abbildung 4.4 ist eine solche Ausgabe im *Facebook Messenger* zu sehen. Es wurden zwei unterschiedliche Formen von Antworten generiert, um mehr Variation im Gespräch zu ermöglichen, da es geplant ist, dass diese Aufrufe vom Benutzer mehrmals ausgeführt werden. Diese werden mit Hilfe von `Math.random()` der Ausgabe zugeteilt.

Abgesehen von der Schwierigkeit der richtigen Verwendung der Artikel und Zeitfor- men in der deutschen Sprache, ergab sich die Problematik der unterschiedlichen Anzahl an Tagen in den jeweiligen Monaten. Ist der letzte Tag eines Monats in der Verar- beitung betroffen, müssen Schritte bei der Analyse der folgenden Termine und in der Ausgabe eingeleitet werden. In der Funktion `isLastDayOfMonth (year, month, day)` wurde die jeweilige Anzahl der Tage festgelegt. Ebenso wurde die Möglichkeit eines Schaltjahrs berücksichtigt.

## Route

Als weiteres dynamisches *Feature* wurde die *Google Maps* API im *Webhook* verwendet, um die Distanz und Fahrzeit von einem angegebenen Standort aus nach Hagenberg im Mühlkreis zu berechnen. Zusätzlich wurde ein *Button* eingebaut, der den Link zur *Google Maps Seite* enthält und dort die eingegebene Route anzeigt. Um diese API verwenden zu können, muss das Projekt auf der *Google Cloud Platform* registriert sein. Daraufhin bekommt man einen *Key*, den man im Projekt angeben muss, um die Funktionen nutzen zu können. Die folgenden Abschnitte dienen zur Erläuterung der Implementierung dieses Features in der *JavaScript*-Datei des *Webhooks*.

```

if(req.body.result.metadata.intentName == "information.direction.followup") {

    var apiKey = "AIzaSyCpbl2YFr0wlqJt0GVMS48_QsH7rwpvd54";

    var startingPlace = req.body.result.resolvedQuery;
    startingPlace = startingPlace.replace("ß", "ss");
    startingPlace = startingPlace.replace("ä", "ae");
    startingPlace = startingPlace.replace("ö", "oe");
    startingPlace = startingPlace.replace("ü", "ue");

    var apiRequest = "https://maps.googleapis.com/maps/api/directions/json?origin=" +
        startingPlace + "&destination=FH+Hagenberg&key=" + apiKey;

    ...
}

```

Zuerst wird mit einer *If-Bedingung* ermittelt, ob der *Webhook* durch den dazugehörigen *Intent* `information.direction.followup` aufgerufen wurde. Dann enthält der vom User eingegebene Text die Informationen, von welchem Standpunkt aus der Benutzer nach Hagenberg im Mühlkreis fahren möchte. Ebenso wird der benötigte *API-Key* angegeben. In die Variable `startingPlace` wird die von *Dialogflow* erhobene Startposition gespeichert. Zusätzlich ist es nötig die Umlaute und das scharfe *ß* zu ersetzen. Daraufhin wird der zu sendende *Request* mit der benötigten URL, bestehend aus den Seiten-Angaben, dem Startpunkt, dem Ziel und dem *API-Key*, befüllt. Anschließend werden die zurückgelieferte Daten extrahiert, mit `split()` und `substr()` verarbeitet und als Ausgabe abgeschickt. Zusätzlich wird der Link zu der externen Webseite mit Hilfe eines Buttons angeboten, wie in der letzten Zeile des folgenden Code-Ausschnittes ersichtlich ist.

```

answer = "Bis zur FH Hagenberg benötigst du mit dem Auto im Normalfall " + time + ".
        Die Strecke beträgt " + distanceInformation + ".";

while(startAddress.includes(" ")) {
    startAddress = startAddress.replace(" ", "%20");
}

var webUrl = "https://www.google.com/maps/dir/?api=1&origin=" + startAddress + "&
            destination=FH%20Hagenberg";

```

#### 4.2.4 Facebook Messenger

Das Projekt wurde für die Interaktion in *Facebook Messenger* optimiert und somit dort getestet und zur Verfügung gestellt. Um das System zu integrieren, sind einige Schritte nötig. Zuerst muss eine beliebige *Facebook-Seite* erstellt werden, die vorerst auch nicht öffentlich sein muss, was sich ideal für Testzwecke eignet. Durch diese Seite verfügt man über ein *Facebook Page Access Token*, welches in *Dialogflow* im Bereich *Integrations* eingefügt werden kann. Nach der Eingabe eines beliebigen Passwortes erhält man eine *Callback URL*. Mit diesem Link und dem ausgewählten Passwort kann auf der offiziellen Seite *Facebook For Developers* ein Projekt erstellt und mit der *Facebook-Seite* verbunden werden. Ebenso ist es möglich auf dieser Plattform einige verwaltungstechnische Schritte durchzuführen, wie das Festlegen von Rollen, die endgültige Veröffentlichung der Bot-Funktion auf der jeweiligen Seite und das Einsehen von analytischen Daten. Für genauere Angaben über Features und Möglichkeiten des Messenger-Dienstes sei auf den Abschnitt 2.4.1 referenziert.

### 4.3 Kontext

In diesem Abschnitt wird die Kontext-Verarbeitung im Projekt erläutert und dazu benötigtes Hintergrundwissen zusammengefasst. In *Dialogflow* befinden sich die wichtigsten Aspekte bezüglich Kontext in jedem *Intent* an oberster Stelle im *Kontext-Bereich*. Dieser kann aus einer Reihe von *Input- und Output-Kontext-Variablen* bestehen. Diese werden mit einer numerisch dargestellten Lebensspanne versehen. Diese gibt an, wie viele weitere *Intents* lang die Variablen gültig sind.

#### 4.3.1 Input-Variablen

*Input-Variablen* werden verwendet, um den Zugriff auf den *Intent* im Falle einer Übereinstimmung zu spezialisieren. Stimmt die Eingabe mit einer der angegebenen Trainingsphrasen überein, überprüft das System, ob die benötigten *Input-Bedingungen* durch die momentan gespeicherten und aktiven Variablen, die in vorigen *Intents* als *Output-Variablen* angegeben wurden, erfüllt sind. Somit wird der *Intent* nur verarbeitet, wenn alle geforderten *Input-Variablen* im Verlauf des Gespräches aktiviert wurden und ihre Lebensspanne nicht abgelaufen ist. Welches Vorgehen im Falle einer mehrfachen Übereinstimmung einer Eingabe des Benutzers erfolgt, wird im folgenden Beispiel zum besseren Verständnis erläutert.

Nach Erhalt der Nachricht stehen möglicherweise drei *Intents* zur Verfügung, die alle eine passende Trainingsphrase enthalten. Einer enthält keine *Input-Bedingung*, der Zweite fordert eine Variable und der Dritte zwei Bedingungen. Alle drei *Intents* wurden in diese Runde der Verarbeitung aufgenommen, da bei jedem alle Bedingungen zutreffen. Nun gilt allgemein die Regel, dass zuerst der spezifischste *Intent* gewählt wird, der die meisten *Input-Bedingungen* fordert. Diese werden generell einem *Intent* vorgezogen, der keine Bedingungen enthält. Dies kann bei komplexeren Systemen für die Gesprächsgestaltung in der jeweiligen Tiefe des Themas genutzt werden.

### 4.3.2 Output-Variablen

Diese Variablen werden gesetzt, um das weitere Gespräch zu beeinflussen. Sie spielen in Mechanismen, wie im Abschnitt 4.3.1 beschrieben, eine maßgebende Rolle. Jede Variable bekommt eine Zahl aus  $\mathbb{N}$ , einschließlich Null, zugewiesen. Dies stellt die Lebensspanne dar, die sich in Anzahl an folgenden *Intents* misst.

Technisch anzumerken ist, dass die Variablen erst nach allen Schritten der Verarbeitung vom System gesetzt werden. Dies macht es in dieser Zusammensetzung unmöglich die angegebenen *Output-Variablen* im Webhook zu bearbeiten oder anderweitig zu beeinflussen.

### 4.3.3 Lebensspanne der Variablen

Standardmäßig haben Variablen in *Dialogflow* eine Lebensspanne von fünf Eingaben des Benutzers oder 20 Minuten nach der Aktivierung. Die numerisch angegebene Lebensspanne für folgende Eingaben lässt sich im jeweiligen *Intent* umschreiben. Wird eine bereits aktivierte Variable in einem weiteren *Intent* erneut angegeben, wird die Lebensspanne daran angepasst. Ebenso ist es möglich die Lebensspanne auf Null zu setzen, was bewirkt, dass die Variable nach Verarbeitung des *Intents* gelöscht wird.

### 4.3.4 Verwendung

Im Rahmen des Projektes werden die Kontext-Variablen benutzt, um zu den jeweiligen Studiengängen identische Fragen beantworten zu können. Das System speichert den Studiengang als Variable, auf den im Gespräch eingegangen wird. Somit können grammatikalisch allgemeine Fragen wie beispielsweise „Wo liegen die Schwerpunkte?“ korrekt beantwortet werden.

Der Chatbot erkennt mit Hilfe der erstellten *Entity courses* einen Themenwechsel zu einen anderen Studiengang. Daraufhin werden die bestehenden Kontext-Variablen gelöscht und die benötigten neuen eingespeichert. Dies geschieht im *Webhook*, der *Intent university.courses.recognition\_and\_contextclear* aufgerufen wird. Dieser wird für die Verarbeitung und Weiterleitung zum jeweiligen Studiengang benötigt und retourniert keine zusätzliche Antwort (siehe Abschnitt 4.4).

Die numerische Angabe der Lebensspanne wurde in beinahe allen Fällen zwischen 5 und 10 gewählt. Diese hohe Zahl dient jedoch nur als Absicherung für eine genügend lange Dauer der Aktivierung. Die eigentliche Verarbeitung der Variablen findet im Falle des vorgestellten Projektes im *Webhook* statt. Dies ist durch Einschränkungen in *Dialogflow* erforderlich. Im nächsten Abschnitt werden dazu nähere Angaben erläutert.

## 4.4 Schwierigkeiten in der Entwicklung

Während des Projektes entstanden einige Schwierigkeiten, die meist durch Einschränkungen von *Dialogflow* verursacht wurden. Die nennenswerten und relevanten Problematiken werden in diesem Abschnitt erläutert.

#### 4.4.1 Einschränkung der Speicherkapazitäten

Trotz einiger positiver Aspekte birgt *Dialogflow* eine massive Einschränkung für die Qualität des Chatbots. Variablen werden nur maximal 20 Minuten lang im System gespeichert. Dieses Hindernis ist dennoch durchaus verständlich, da die kostenlose Plattform Speicherplatz für alle Projekte zur Verfügung stellen muss und je nach Beliebtheit und Verwendung eines Bots, dadurch große Datenmengen entstehen können, würde man keine Begrenzung einführen. Jedoch ist es aus Sicht der Entwicklung enttäuschend, dass selbst im bezahlten *Business Dialogflow* die Lebensspanne nicht verlängert wird. Somit besteht keine Möglichkeit, unabhängig davon ob kostenlos oder kostenpflichtig, diese Problematik intern zu lösen.

Durch die Restriktion auf 20 Minuten leiden einige mögliche Szenarien der Benutzung. Beispielsweise ist es dem Benutzer nicht möglich eine kurze Pause einzulegen, sei es für den Gang ins Badezimmer oder zur weiteren Recherche und Beratschlagung mit Mitreisenden bei Buchungsvorgängen. Die schriftliche Konversation ist, im Gegensatz zu einem Telefonat, darauf ausgelegt Pausen und Abstände zu ermöglichen. Anzumerken ist, dass diese Problematik während der Projekt-Entwicklung deutlich gravierender war. Erst nach einigen Aufrufen der Entwickler im Forum von *Dialogflow* wurde die Lebensspanne von nur kurzen 10 Minuten im Sommer 2018 verdoppelt.

In der Entwicklung des Projektes wurde angenommen, dass die schnellste und effektivste Lösung eine Datenbank oder eine simple Datei für Testzwecke wäre. Mit geringem Aufwand wurde eine Datenbank angelegt und im *Webhook* eingebaut. Daraufhin folgte die Erkenntnis der nächsten Schwierigkeit mit der Arbeit in *Dialogflow*, welche im folgenden Abschnitt erläutert wird.

#### 4.4.2 Verwendung von Benutzereingaben

Die Datenbank brachte den erwarteten Erfolg, dass die benötigten Variablen zu der dazugehörigen *Facebook*-ID gespeichert werden. Es stellte sich die Schwierigkeit heraus, dass es im *Webhook* zwar möglich ist, an einen anderen *Intent* mit Hilfe der Angabe seiner Bezeichnung weiterzuleiten und dessen Antwort zu benutzen, jedoch nicht, dass diese Weiterleitung aufgrund der stattgefundenen Benutzer-Eingabe als Parameter getätigt wird.

Beispielsweise wird angenommen, dass der Benutzer nach einer Pause von über 20 Minuten, was der Lebensspanne der Variablen in *Dialogflow* entspricht, wieder mit dem Chatbot interagiert. Der User stellt seine nächste Frage und erwartet eine Antwort im dazugehörigen Kontext. Um dies zu ermöglichen, müsste im System ein spezieller *Intent* aufgerufen werden, der keine Antwort, sondern nur den Aufruf des *Webhooks* enthält. Im *Webhook* können die Variablen des Benutzers aus der Datenbank geladen werden. Daraufhin müsste auf den benötigten *Intent* weitergeleitet werden, der aus der Verbindung der Benutzer-Eingabe und des davor genutzten Kontextes erforderlich wäre. Jedoch besteht die Einschränkung der *Dialogflow* API, dass Benutzer-Eingaben, die in diesem Fall gebraucht würden, um zu den passenden *Intent* weiterzuleiten, nicht als Bedingungen für jene Übermittlung verwendet werden können.

Zusammengefasst ist es möglich die Variablen zu speichern und wieder im Gespräch zu verwenden, jedoch gilt das nicht für die erste Eingabe des Benutzers nach seiner Rückkehr zur Konversation. Ab der nächsten Nachricht des Users hatte das System

bereits davor die Möglichkeit die Variablen erneut zu laden und zu verwenden, doch es ist nicht der gewünschte Effekt im Gespräch, dass die erste Nachricht des Benutzers nach erneut aufgenommenener Interaktion falsch interpretiert oder ignoriert wird.

Im zeitlich begrenzten Rahmen des Projektes wurde diese Problematik ausgeforscht, beschrieben und aus Zeitmangel zur Seite gelegt. Dennoch wurden zwei mögliche Lösungen erörtert. Es wäre möglich, dem Benutzer, nach Ablauf der vorgegebenen 20 Minuten eine unabhängige Nachricht zukommen zu lassen, die beispielsweise einen *Quick-Reply-Button* mit der Aufschrift „Ich bin wieder da!“ enthält. Wird dieser vom Benutzer aktiviert oder eine andere triviale Nachricht gesendet, können im *Webhook* am Ende des dazugehörigen *Intents* die Kontext-Variablen geladen werden und sind daraufhin für weitere Eingaben wieder einsatzbereit. Jedoch besteht natürlich die Möglichkeit, dass der User den Button ignoriert und sein Gespräch mit einer essentiellen Frage fortsetzt. Die klassische Alternative wäre, das System der Variablen in *Dialogflow* auszulagern und vollständig selbst zu verwalten. Dies benötigt verständlicherweise einen dementsprechenden Mehraufwand.

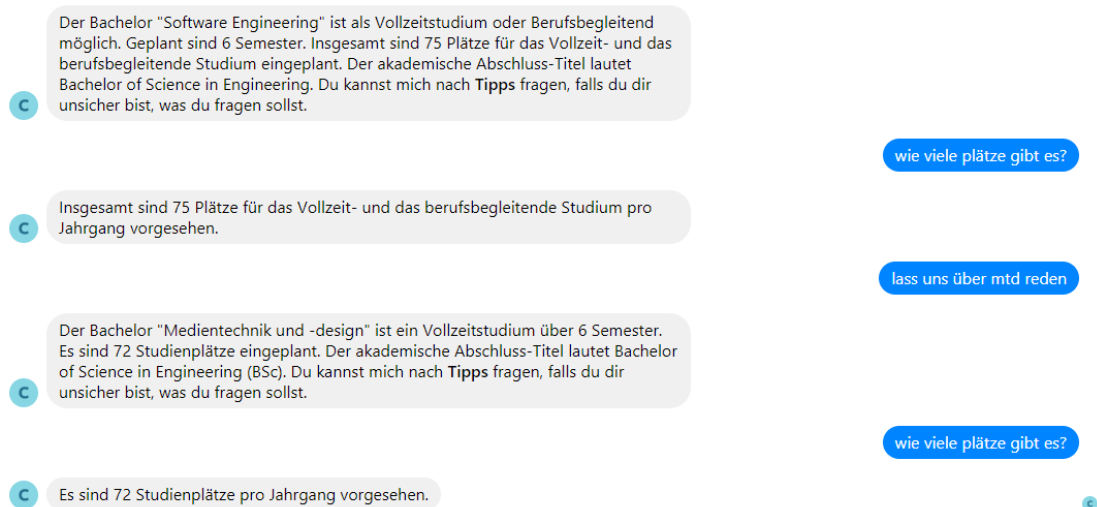
#### 4.4.3 Löschen der Variablen

Im entwickelten Chatbot ist es, wie bereits im Abschnitt 4.3.4 erläutert, möglich allgemeine Fragen wie „Wo liegen die Schwerpunkte?“ zu stellen. Mit Hilfe der Kontext-Variablen verarbeitet das System diese Anfrage dennoch korrekt und leitet zu dem *Intent* weiter, der den dazugehörigen Studiengang zu dieser Frage abdeckt. Sollte der Benutzer jedoch im Gespräch einen anderen Studiengang ansprechen, ist es von Nöten, dass das System *vor* der nächsten Benutzung eines *Intents* die vorhandenen Variablen zu dem davor besprochenen Studiengang löscht und die Neuen speichert. Ein solches Vorgehen findet im *Webhook* statt. Dieser wird jedoch, wie in diesem Kapitel häufiger erwähnt, erst am Ende der Verarbeitung eines *Intents* aufgerufen. Somit könnte im beschriebenen Szenario der *Intent* aufgerufen werden, der zu den vorigen Variablen passt und erst am Ende von diesem die verfälschenden Kontext-Aspekte mit Hilfe des *Webhooks* gelöscht werden.

Da dies ungewollte Schwierigkeiten hervorrufen kann, wurde im Projekt ein Übergang mit Hilfe eines *Zwischen-Intents* erstellt, der für die Erkennung des Studiengangs, das Löschen der Variablen und die Weiterleitung an den benötigten *Intent* zuständig ist. Dieser trägt die Bezeichnung `university.courses.recognition_and_contextclear` im System. Somit ist es möglich, das Gespräch ohne ungewollte Variablen im korrekten Kontext fortzusetzen. Der *Intent* ist ebenfalls nötig, um den richtigen Kontext bei Namensgleichheit von Bachelor- und Master-Studiengängen zu gewährleisten. Dies geschieht mit Hilfe der weiteren Variablen `courses` und `courses_master`, die während des Gesprächs gewechselt werden können und das weitere Geschehen beeinflussen.

## 4.5 Ergebnis

In diesem Abschnitt werden die Ergebnisse und *Features* des Projektes zusammengefasst. Die erforderlichen und möglichen weiterführenden Aspekte und Verbesserungen der Entwicklung werden in *Diskussion und Fazit* im Abschnitt 5.2.2 angeführt.



**Abbildung 4.3:** Darstellung der gelungenen Kontext-Verarbeitung des Projektes.

#### 4.5.1 Abgedeckte Themen

Es ist möglich, sich mit dem entwickelten Chatbot über folgende Themen zu unterhalten:

- Infrastruktur von Hagenberg im Mühlkreis und den benachbarten Gemeinden,
- Studiengänge an der Fachhochschule,
- Allgemeines zur Fachhochschule und
- Smalltalk.

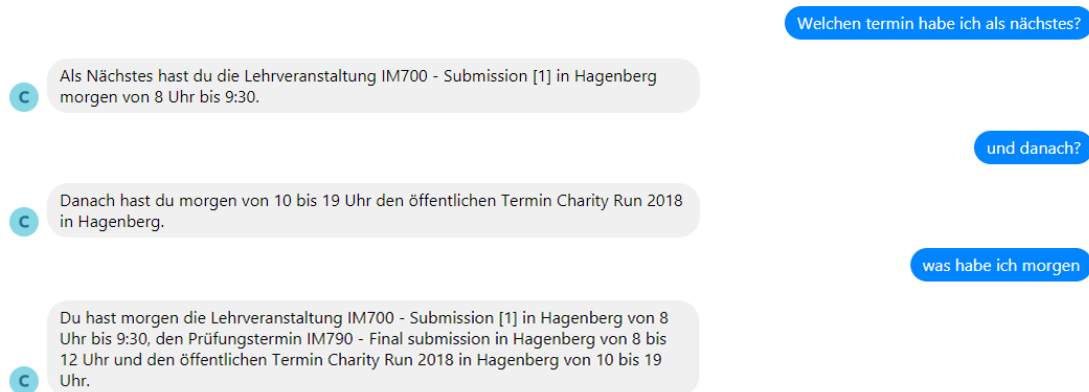
Der Begriff *Infrastruktur* deckt in diesem Zusammenhang die Informationen über ansässige Ärzte, Einkaufsmöglichkeiten, Gastronomie, Wohnmöglichkeiten und Freizeitgestaltung ab. Zur Umsetzung von grundsätzlichen Smalltalk wurde ein Paket von *Dialogflow* benutzt, welches vorgefertigte *Intents* beinhaltet und eine Auswahl an Trainingsphrasen. In der deutschen Sprache sind jedoch die dazugehörigen Antworten manuell einzugeben.

#### 4.5.2 Studiengänge

Es ist für den Benutzer möglich sich ausführlich über die unterschiedlichen Studiengänge zu informieren. Dies gilt für Bachelor- und Master-Studien. Neben allgemeinen Informationen zu Dauer, verfügbaren Plätzen und zu erwartenden akademischen Titeln, kann der User grundsätzlich folgende Themen vertieft besprechen:

- Zeitaufwand des Studiums,
- Aufnahmekriterien,
- Studienplan,
- Schwerpunkte,
- weiterführende Studien-Möglichkeiten,
- mögliche Arbeitsplätze und wenn vorhanden,
- Vertiefungen während des Studiums.





**Abbildung 4.4:** Mögliche Eingaben, um die Funktion der Kalender-Verarbeitung zu nutzen.

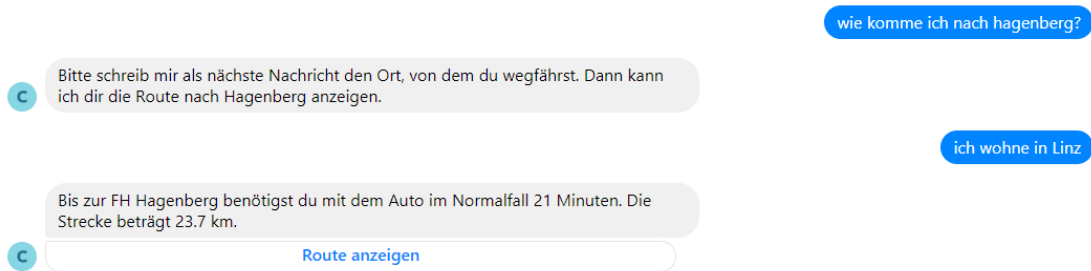
In diesen Szenarien spielt der Kontext eine maßgebliche Rolle. Die Schwierigkeiten der Verarbeitung, besonders bezüglich dieser Problematiken, werden im Abschnitt 4.4.3 erläutert. Die korrekte Umsetzung davon ist in Abbildung 4.3 ersichtlich. Der Chatbot gibt die gewünschten Informationen zum *Software Engineering Bachelor*. Wird nach der Anzahl der verfügbaren Plätze in diesem Studium gefragt, wird die korrekte Anzahl von 75 genannt, was mit der darüber liegenden Zusammenfassung übereinstimmt. Der Benutzer wechselt daraufhin das Thema und möchte nun Informationen zum Studiengang *Medientechnik und -design* erhalten. Nach der kurzen Erklärung schickt er erneut die Anfrage zu der Anzahl an möglichen Studierenden. Der Kontext wurde korrekt geändert und die angepasste Zahl 72 ausgegeben.

#### 4.5.3 Rich Messages

Im Projekt wurde eine Vielzahl der Arten von *Rich Messages* verwendet. Buttons und *Quick Replies* wurden vorrangig dafür verwendet, um auf eine externe Ressource weiterzuleiten und im Gesprächsverlauf die nächsten möglichen Punkte zu einem spezifischen Thema vorzustellen und anzuführen. Karusselle wurden in gemäßigter Weise bei passenden Aspekten verwendet, wie beispielsweise die Auflistung der möglichen Studiengänge und Freizeitmöglichkeiten in der Umgebung (siehe Abbildung 2.4). Wenn es einer Beschreibung zweckdienlich war, wurde ein Bild in die Konversation eingebunden. Ebenso kann der Benutzer bei bestimmten Studiengängen ein PDF-File anfordern, welches ausführliche Informationen des Studiums beinhaltet. Weitere Dateien wie Videos oder Gifs wurden nicht verwendet. Anzumerken ist, dass keine Files oder *Emojis*, die vom Benutzer geschickt werden, verarbeitet werden können.

#### 4.5.4 Einbindung des Kalenders

Durch die Bereitstellung des dynamischen *Features*, können dem Benutzer Fragen zum eigenen Stundenplan beantwortet werden. Wie in der Abbildung 4.4 ersichtlich ist, kann nicht nur nach beispielsweise der nächsten Stunde gefragt werden, sondern auch eine Zusammenfassung des heutigen oder morgigen Tages generiert werden. Ebenso ist es



**Abbildung 4.5:** Darstellung von möglichen Anfragen, um sich die Route anzeigen zu lassen.

möglich nach Erhalt der Information über den nächsten Termin, nach dem darauf zu fragen. Die Formulierung der Termine umfasst deren Art, die Bezeichnung, das Datum, die Uhrzeit und wenn vorhanden, den Ort oder Raum.

#### 4.5.5 Google Maps API

Wie in Abschnitt 4.2.3 erläutert wurde, umfasst die Verwendung der *Google Maps API* im Projekt die Berechnungen der Route und die vorausgefüllte Webseite des Anbieters. Der Benutzer wird nach Anfrage der Route um Bekanntgabe des Start-Standortes gebeten. Daraufhin verarbeitet das System die Daten und gibt die geplante Dauer der Fahrt mit dem Auto und die Kilometeranzahl dieser Strecke an. Ebenso ist mit Hilfe eines Buttons der Link zur Webseite von *Google Maps* angegeben. Wird dieser aktiviert, öffnet sich ein zusätzlicher *Tab* und der Benutzer bekommt die vorausgefüllte Routen-Ansicht. Dieses *Feature* funktioniert mit allen Standorten und Routen, die auch von *Google Maps* unterstützt werden. Zur Veranschaulichung dieses Szenarios dient die Abbildung 4.5.

## Kapitel 5

# Diskussion und Fazit

Zusammenfassend kann gesagt werden, dass zur Entwicklung eines Chatbots mit optimaler Kontext-Verarbeitung Kenntnisse aus dem Bereich der Linguistik, Gesprächsführung und Programmierung notwendig sind. Seit Beginn der Entwicklung von solchen Systemen in den frühen 60er Jahren bis zur heutigen modernen Auffassung von Bots spielte das Thema *Kontext* eine maßgebliche Rolle, um die Qualität des Gespräches zu verbessern. Was bei *Eliza* von Joseph Weizenbaum im Jahre 1966 noch per Hand in AIML ausgearbeitet werden musste, übernimmt heute größtenteils ein aufwendiger und umfassender *Natural Language Processor*. Ebenfalls konnte die Qualität der Gespräche in Bezug auf Kontext gesteigert werden, indem die unterschiedlichen Arten von Bots auch dementsprechend verschieden aufgebaut und der Kontext angepasst verarbeitet und angewandt wurde. Es stehen den Entwicklern und Firmen eine Vielzahl an Plattformen zur Verfügung, um ihre Marke mit einem Chatbot optimal zu repräsentieren. Nach dem Inkrafttreten der neuen *Datenschutz-Grundverordnung* sind zusätzliche Aspekte bei der Verarbeitung von personenbezogenen Daten während der Entwicklung zu beachten.

Die Bedeutsamkeit von Kontext ist in Bezug auf die Gesprächsführung unerlässlich. Trotz klarer Definitionen von technischen Aspekten, die umgesetzt werden müssen, wie beispielsweise die Verarbeitung von Variablen und Pronomen, entstehen Schwierigkeiten, die das Empfinden und die Reaktion des Benutzers negativ beeinflussen können. Häufig werden diese durch einen geringen Aufwand bei der Planung der Verarbeitung unerwarteter und hochgradig komplizierter Eingaben und fehlerhafte Auffassungen des Systems hervorgerufen. Mit Hilfe von kommunikativen Maßnahmen wie *Rich Messages* und einem detaillierten *Opt-In* können derartige Problematiken abgefangen oder im Falle des Geschehens daraufhin ausgebessert werden.

Das korrespondierende Projekt dieser Arbeit umfasst die Entwicklung eines Chatbots für die *Fachhochschule Oberösterreich Campus Hagenberg*, welcher fähig sein sollte, Fragen über das Studium und die Infrastruktur zu beantworten. Zusätzlich sollte es für die Studierenden möglich sein, den eigenen Kalender zu verbinden, um bevorstehende Termine aufgelistet zu bekommen und die Route von einem beliebigen Standpunkt aus zur Fachhochschule abzufragen. Das Projekt wurde mit Hilfe der Plattform *Dialogflow* und einen in *JavaScript* geschriebenen *Webhook* realisiert. Für die Veröffentlichung wurde der Dienst *Facebook Messenger* gewählt.

## 5.1 Beantwortung der Forschungsfrage

Den Entwicklern stehen einige definierte technische Aspekte zur Verfügung, die während der Entwicklung umgesetzt werden sollten. Diese beinhalten Erläuterungen zur korrekten Verarbeitung von Variablen, das Setzen derer Lebensspanne und die Verwendung von gesprächsübergreifenden Informationen. Ebenso ist die Erkennung der verwendeten Pronomen und die dementsprechende Reaktion darauf in der Entwicklung zu berücksichtigen. Diese Aufgabenstellung gehört zum größten Teil in den Bereich von NLP und ist bei den Plattformen zur Chatbot-Entwicklung, wie beispielsweise *Dialogflow*, standardmäßig enthalten.

Zusätzlich können die Entwickler auf kommunikative Maßnahmen zurückgreifen, um einige Szenarien, wie unerwartete oder zu komplexe Eingaben, die zu Schwierigkeiten und einer negativen Erfahrung des Benutzers führen können, zu vermeiden. Dies reicht von der Einbindung von *Rich Messages*, über Angebote für den Verlauf des weiteren Gesprächs, bis zum Senden eines detaillierten *Opt-Ins*, welches von vornherein klarstellt, für welche Zwecke das System entwickelt wurde.

## 5.2 Ausblick

Die Verwendung von Chatbots, unabhängig ob diese sprach- oder textgesteuert werden, wird laut Prognosen der Marktforschung und Experten im Bereich der Kundenkommunikation weiterhin ansteigen. Ebenso werden sich die Erwartungen an Qualität und Service daraufhin erhöhen. Diese Entwicklung fordert eine technische und kommunikative Steigerung der Funktionen und Anwendung in Bezug auf die Kontext-Verarbeitung. Da für eine optimale Gesprächsführung, die Schwierigkeiten wie beispielsweise *Bot Amnesia* vermeiden soll, bereits ausreichend Möglichkeiten erläutert werden und zur Verfügung stehen, sollte sich die weitere Entwicklung auf die Verbesserung im Bereich NLP und Verwendung von Variablen fokussieren, um nachhaltig die Verarbeitung des Kontextes in Bot-Systemen zu optimieren.

### 5.2.1 Erweiterung der Arbeit

Im Rahmen dieser Arbeit wurden die relevanten Aspekte bezüglich der Entwicklung und Verbesserungen der Kontext-Verarbeitung in Chatbot-Systemen erläutert. Umfangreicher und praxisorientierter wären diese Angaben, wenn zusätzlich weitere und konkretere Beispiele gegeben werden, wie diese in bekannten Systemen umgesetzt wurden. Ebenso wären technische Abschnitte der Systems aufschlussreich und könnten bei umfangreicher Analyse weitere Erkenntnis bringen. Die im Abschnitt 3.3.3 beschriebenen Aspekte der Pronomen in Bezug auf die Kontext-Verarbeitung könnten mit Hilfe eines Einblicks in deren Erhebung und Zuordnung in NLP-Systemen detaillierter erfasst werden. Diese möglichen Erweiterungen sind für die Beantwortung der angegebenen Forschungsfrage nicht zwingend notwendig, sondern würden nur zu einem tieferen Verständnis der jeweiligen Thematik führen. Ebenso wären für diese Aspekte größtenteils zur vollständigen Darstellung sehr detaillierte Erläuterungen von Nöten, die jedoch den Rahmen dieser Arbeit sprengen würden.

### 5.2.2 Weiterführung des Projektes

Die Möglichkeiten für die Weiterführung des vorgestellten Projektes gliedern sich in zwei unterschiedliche Bereiche. Die folgende Auflistung beinhaltet Aspekte, die zwingend erforderlich sind, um das System erfolgreich zu veröffentlichen und für die *Fachhochschule Oberösterreich Campus Hagenberg* zu benutzen.

- Erweiterung der Themengebiete,
- weitere Anhäufung von Trainingsphrasen in den einzelnen *Intents*,
- eine funktionierende Datenbank, die das ICS-File des Benutzers speichert und
- Maßnahmen, um mit der DSGVO konform zu sein.

Im Rahmen des Projektes wurden manche Themen nur oberflächlich eingebaut, wie beispielsweise die Infrastruktur und die unterschiedlichen Arten des Unterrichts. Diese sollten vor der Veröffentlichung ausgiebig erweitert werden. Zusätzlich sollte die Anzahl und Varianz der Trainingsphrasen der jeweiligen *Intents* ausgebaut werden, um eine bessere Erkennung der Anfragen zu gewährleisten. Um den Service der Abfrage des Stundenplans zu erhalten, ist das Aufsetzen und Einbinden einer Datenbank nötig, die das File des Benutzers mit seiner spezifische *User-ID* speichert, um nicht bei jedem Gespräch die Daten neu anfordern zu müssen. Ebenso müssen Maßnahmen ergriffen werden, um die Aspekte der *Datenschutz-Grundverordnung* zu erfüllen.

Die nächste Auflistung beinhaltet Möglichkeiten, die kein Muss sind, aber jedoch zur Qualität des Systems beitragen können.

- Erweiterung der möglichen Anfragen bezüglich des Stundenplans,
- Erstellung unterschiedlicher möglicher Antworten innerhalb eines *Intents*,
- Verarbeitung der Kontext-Variablen selbst übernehmen, um die begrenzte Lebensspanne von 20 Minuten in *Dialogflow* zu umgehen,
- Einbindung weiterer dynamischer Inhalte,
- Abspeichern und Analyse der Gespräche und
- weitere Campus hinzufügen.

Die Verarbeitung der Anfragen des Stundenplans können erweitert und verbessert werden. Beispielsweise wäre die Abfrage des Stundenplans für einen bestimmten Tag vorteilhaft für den Benutzer. Gibt der Chatbot für die selbe Nachricht unterschiedliche Antworten mit gleicher Information, aber geänderter Struktur wider, kann die Variation des Gespräches gesteigert werden. Ein bedeutsamer Aspekt in dieser Auflistung ist der Vorschlag zur Erstellung einer Datenbank und die eigene Verarbeitung der Variablen darin. Dies ermöglicht die eingeschränkte Lebensspanne der Plattform *Dialogflow* zu umgehen. Verständlicherweise ist bei dieser Entwicklung mit einem erheblichen Mehraufwand zu rechnen. Weitere dynamische *Features*, wie beispielsweise das Einbinden der wechselnden Menüs in den Gastronomie-Betrieben, wäre eine zusätzliche Bereicherung. Abschließend wäre das Speichern und die Analyse der Gespräche ein idealer Weg, um das System kontinuierlich zu verbessern. Anzumerken ist, dass der Benutzer aufgrund der DSGVO dafür um Erlaubnis gebeten werden muss und ein klares Einverständnis zu geben hat. Sollte der Chatbot voll funktionstüchtig und einsatzfähig sein, wäre eine ideale Erweiterung, dass er für die Studierenden von allen Standorten der *Fachhochschule Oberösterreich* verwendbar ist und die jeweiligen Informationen angepasst werden.

# Anhang A

## Inhalt der CD-ROM

### A.1 Hauptverzeichnis

Pfad: /

Thesis.pdf . . . . . Diese Arbeit als *PDF-Datei*

### A.2 Medien

Pfad: /medien

amnesia.png . . . . . Beispiel für *Bot Amnesia*  
angebote.png . . . . . Beispiele dem Benutzer Angebote zu unterbreiten  
Carousel.PNG . . . . . Verwendung eines Karussells im Projekt  
context1.png . . . . . Beispiel für gelungene Verarbeitung der Pronomen  
context2.png . . . . . Verwendung von zwei Kontexten  
context3.png . . . . . Verwendung von *Rich Messages*  
courses.PNG . . . . . Wechseln des Themas zu einen anderen Studiengang  
dialogflow\_aufbau.PNG Aufbau der *Dialogflow-Konsole*  
eliza.png . . . . . Darstellung eines Gesprächs mit dem Chatbot *Eliza*  
emoji.png . . . . . Analyse zur Bedeutung von Emojis  
error.png . . . . . Umgehen eines häufig auftretenden Problems  
ics.png . . . . . Anfragen zum Stundenplan  
logo.pdf . . . . . Logo für Arbeiten der Fachhochschule Oberösterreich  
mitsuku.PNG . . . . . Gesprächsverlauf mit dem Bot *Mitsuku*  
mitsuku\_b.png . . . . . Fehlerhafte Auffassung von dem Bot *Mitsuku*  
optin.png . . . . . Beispiel für ein ausführliches *Opt-In*  
personalbot.png . . . . . Bedeutung von Kontext bei Bestellvorgängen  
responses.png . . . . . Aufbau der Antworten in *Dialogflow*  
route.PNG . . . . . Anfrage der Route  
turingtest.png . . . . . Darstellung des *Turing-Test*

### A.3 Projekt

Pfad: /projekt

- dialogflow.zip . . . . . Daten des Projektes von *Dialogflow*
- index.js . . . . . Webhook

### A.4 Sicherung Online-Referenzen

Pfad: /onlineref

- dsgvo.pdf . . . . . Online-Quelle [10]
- emojis.pdf . . . . . Online-Quelle [11]
- fakten.pdf . . . . . Online-Quelle [12]
- future.pdf . . . . . Online-Quelle [19]
- guide.pdf . . . . . Online-Quelle [18]
- hoffer.pdf . . . . . Online-Quelle [13]
- oracle.pdf . . . . . Online-Quelle [16]
- pardon.pdf . . . . . Online-Quelle [17]
- wko.pdf . . . . . Online-Quelle [15]

# Quellenverzeichnis

## Literatur

- [1] A. Braun und P. Glotz. *Chatbots in der Kundenkommunikation*. Xpert. press Series. Springer, 2003 (siehe S. 4).
- [2] H. Bussmann. *Lexikon der Sprachwissenschaft*. Kröner, 2002 (siehe S. 18).
- [3] G. Dyson und K. H. Siber. *Turings Kathedrale: Die Ursprünge des digitalen Zeitalters*. Ullstein eBooks, 2014 (siehe S. 5).
- [4] Robert Epstein, Gary Roberts und Grace Beber. *Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer*. Springer Publishing Company, Incorporated, 2008 (siehe S. 6, 8).
- [5] A. Hodges. *Alan Turing: The Enigma*. Vintage, 1992 (siehe S. 5).
- [6] Amir Shevat. *Designing Bots—Creating Conversational Experiences*. O’Reilly Media, Inc., 2017 (siehe S. 3, 9).
- [7] A. M. Turing. „Computing Machinery and Intelligence“. *Mind*. New Series 59.236 (1950), S. 433–460 (siehe S. 5, 6).
- [8] Joseph Weizenbaum. „ELIZA—A Computer Program for the Study of Natural Language Communication Between Man and Machine“. *ACM* 9.1 (Jan. 1966), S. 36–45 (siehe S. 7).
- [9] Janine Willis und Alexander Todorov. „First Impressions: Making Up Your Mind After a 100-Ms Exposure to a Face“. *Psychological Science* 17.7 (2006), S. 592–598 (siehe S. 32).

## Online-Quellen

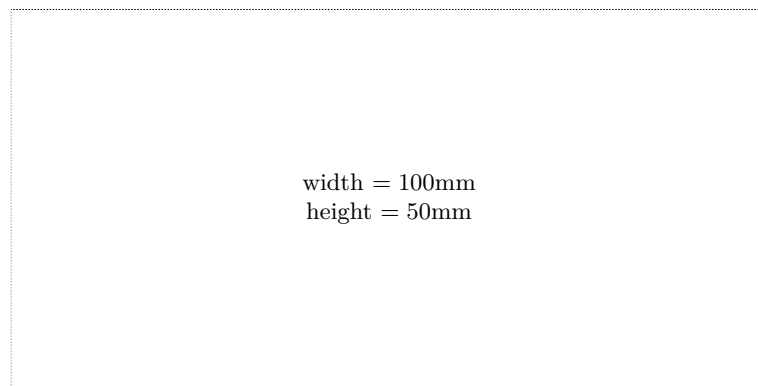
- [10] Intersoft consulting services AG. *Datenschutz-Grundverordnung (DSGVO)*. [Online; accessed 29-August-2018]. 2018. URL: <https://dsgvo-gesetz.de/> (siehe S. 34, 54).
- [11] Social News Daily. *What Does Your Favorite Emoji Mean Around the World?* [Online; accessed 01-September-2018]. 2017. URL: <https://socialnewsdaily.com/66932/emoji-meaning-around-the-world/> (siehe S. 24, 25, 54).



- [12] Matthias Großkopf. *10 Fakten, die du noch nicht über Chatbots kennst*. [Online; accessed 8-August-2018]. 2018. URL: <http://www.magronet.de/11-fakten-ueber-chatbots/> (siehe S. 17, 54).
- [13] Robert Hoffer. *The trouble with bots: A parent's musings on SmarterChild*. [Online; accessed 01-September-2018]. 2016. URL: <https://venturebeat.com/2016/06/15/the-trouble-with-bots-a-parents-musings-on-smarterchild/> (siehe S. 23, 54).
- [14] Gregor Lüdi und Martin Lüscher. *Abbildung Turing Test*. [Online; accessed 8-August-2018]. 2007. URL: <http://www.natur-struktur.ch/ai/images/turingtest.gif/> (siehe S. 6).
- [15] Wirtschaftskammer Österreich. *EU-Datenschutz-Grundverordnung: Verantwortlicher und Auftragsverarbeiter*. [Online; accessed 29-August-2018]. 2018. URL: <https://www.wko.at/service/wirtschaftsrecht-gewerberecht/EU-Datenschutz-Grundverordnung:-Verantwortlicher-und-Aufr.html/> (siehe S. 33, 54).
- [16] T3N digital pioneers. *Oracle-Studie: Bis 2020 erledigen Chatbots den Kundendienst*. [Online; accessed 03-September-2018]. 2016. URL: <https://t3n.de/news/chatbots-uebernehmen-kundendienst-bis-2020-775998/> (siehe S. 1, 54).
- [17] *Royal pardon for codebreaker Alan Turing*. [Online; accessed 8-August-2018]. 2013. URL: <https://www.bbc.com/news/technology-25495315/> (siehe S. 5, 54).
- [18] Matt Schlicht. *The Complete Beginner's Guide To Chatbots*. [Online; accessed 8-August-2018]. 2016. URL: <https://chatbotsmagazine.com/the-complete-beginner-s-guide-to-chatbots-8280b7b906ca/> (siehe S. 3, 54).
- [19] Adelyn Zhou. *Chatbots Are The Future Of Marketing*. [Online; accessed 8-August-2018]. 2017. URL: <https://thinkgrowth.org/chatbots-are-the-future-of-marketing-67d21bd557db/> (siehe S. 15, 54).

# Messbox zur Druckkontrolle

— Druckgröße kontrollieren! —



— Diese Seite nach dem Druck entfernen! —