Conversational Agents im Order Management

Jan-Lucas Sachse (5009193), Marc-Alexander Jozef (5009451)

***Abstract* — Conversational Agents können in einem Unternehmen, z. B. einem Restaurant, benutzt werden, um einen modernen Webauftritt zu garantieren. Sie können dabei als eine Art „zweites Ich“ agieren und die Online-Präsenz pflegen und dabei aufkommende Fragen eines Kunden beantworten. Denkbar praktisch können Conversational Agents im Order Management (Restaurant mit Online-Bestellungen) sein, um z. B. einen Bestellvorgang durchzuführen, Fragen zum Essen zu beantworten, einen Tisch zu reservieren oder den Standort des Restaurants schnell über einen Chat durchzugeben. In diesem Paper untersuchen wir diese Use-Cases, indem wir einen Conversational Agent als Chatbot implementieren und im Anschluss mit Hilfe einer Benutzerumfrage evaluieren. Damit soll die Akzeptanz und das Empfinden der Benutzer gegenüber eines Chatbots im Order Management ermittelt werden.**

I. Einleitung

Der Fortschritt bei der Rechenleistung und Rechenkapazität der Computer ermöglicht den Einsatz von künstlicher Intelligenz unter anderem im Feld *Natural Language Processing (NLP)*, *Natural Language Understanding (NLU)* und *Machine Learning (ML)* allgemein. Dadurch wird die Interaktion von Menschen mit Computern durch Austausch von Texten möglich, bei dem der Computer auch die Absicht des Benutzers aus der Anfrage erkennen kann (Pykes, 2021).

Die Interaktion wird mit sogenannten *Chatbots* implementiert. Diese sind in der heutigen Zeit wichtig, um einen modernen Auftritt eines Unternehmens, Restaurants usw. zu gewährleisten und das Benutzererlebnis weiter zu verbessern. Richtig populär wurden diese, seitdem Facebook 2016 ihre Plattform für *Chatbots* geöffnet und damit einer großen Menge an Menschen den Zugang dazu gegeben hat (Lyzanets, 2019).

Chatbots werden oft auch *Conversational Agents*, *virtual assistants* oder *interactive personalities* genannt. Dahinter verbirgt sich allerdings der gleiche *use-case*, die Simulation einer Konversation zwischen einem Benutzer und einem Computerprogramm (Lyzanets, 2019).

Damit die Konversation so positiv wie möglich wahrgenommen wird, ist es notwendig, die Dialoge möglichst gut zu modellieren und den Benutzer durch einen Dialog zu führen (Aneja, McDuff, & Czerwinski, 2020)

Das Paper (Aneja, McDuff, & Czerwinski, 2020) hat die verschiedenen Einflüsse auf die Wahrnehmung eines *Conversational Agents* untersucht und dabei herausgearbeitet, auf welche Fehler man besonders achten soll bzw. welche Fehler einen besonders großen negativen Einfluss auf die Bewertung oder auch Akzeptanz eines *Conversational Agents* haben. Im folgenden Kapitel *Grundlagen* werden diese Fehler herausgearbeitet, weil diese einen wichtigen Beitrag zu unserer Konzeptionierung des *Chatbots* liefern (Aneja, McDuff, & Czerwinski, 2020).

Nach den Grundlagen wird auf die Konzeption des *Chatbots* eingegangen. Dann wird die Nutzerumfrage mit den Ergebnissen vorgestellt und zum Schluss ein Fazit gebildet, dass die Frage beantwortet, ob *Chatbots* im Order Management sinnvoll sind.

II. Grundlagen

Chatbots sind Anwendungen, die mithilfe von künstlicher Intelligenz mit Menschen in natürlicher Sprache kommunizieren können. Dabei können Benutzer Fragen stellen, auf welche das System in einem Chat z. B. antwortet. Die Möglichkeiten der Ein- und Ausgabe sind dabei vielfältig. Mögliche Kanäle sind z. B. Textnachrichten und Audionachrichten. Diese können auch beliebig kombiniert werden (IBM, 2021).

Die Use-Cases von Chatbots sind eher einfacherer Natur und umfassen in der Regel keine komplexen Konversationen. Das bedeutet, dass Antworten bzw. nebensächliche Informationen aus vorherigen Sätzen nicht mit in die Auswertung für den zukünftigen Verlauf der Konversation einfließen (IBM, 2021).

Damit der Chatbot die Sätze des Benutzers erkennen kann, werden Erkennungsfunktionen verwendet. Diese erkennen, was der Benutzer geschrieben bzw. gesagt hat, und geben den Inhalt an weitere Verarbeitungsfunktionen weiter, die sich um den Inhalt der übermittelten Nachricht kümmern. Der Inhalt oder auch die Absicht des Benutzers wird mit Hilfe von Natural Language Understanding nachvollzogen, oder auch verstanden (IBM, 2021).

Wenn der Inhalt korrekt erfasst werden konnte, dann können mithilfe von diesem Vorgehen, Prozesse automatisiert werden. Das bedeutet, dass z. B. bei Absicht X, die Aktion Y ausgelöst werden soll. Das kann Geld sparen, weil z. B. der Support eines Unternehmens unterstützt werden kann und die Bedürfnisse der Kunden vorkategorisiert werden, damit im Anschluss sich ein Spezialist mit dem Anliegen beschäftigen kann (IBM, 2021).

Mittels der vielen Kanäle ist es außerdem möglich, den gleichen Chatbot auf mehreren Plattformen einzubinden. So wäre es denkbar, dass dieser sowohl auf der Homepage eines Unternehmens eingesetzt wird, als auch auf der entsprechenden Facebook-Seite zum Bespiel.

Ist der Chatbot über viele Kanäle ansprechbar, so hat man die Möglichkeit, viele potenzielle Kunden anzuwerben. Beispielsweise kann ein Chatbot für Produktempfehlungen genutzt werden, z. B. zum Empfehlen einer bestimmten Pizza. Weiterhin wäre auch der Einsatz als Kundenservice-Bot denkbar, der einfachere Fragen oder Anliegen von Kunden abwickelt, oder eine Art *Concierge-Service*, der z. B. den Kunden während eines Bestellvorgangs unterstützt und bei Fragen zur Seite steht. Außerhalb von Use-Cases, die nach „außen“ wirken, sind auch Anwendungsfälle möglich, bei

denen der Chatbot intern genutzt wird, um die Mitarbeiter zu unterstützen und Anfragen weiterzuleiten (IBM, 2021).

Wenn die *Use-Cases* implementiert werden, kann es zu vielen unterschiedlichen Fehlern kommen. (Aneja, McDuff, & Czerwinski, 2020) haben versucht, die Fehler zu kategorisieren, um zu prüfen, ob man eine Kategorie an Fehlern mit einer bestimmten Maßnahme beseitigen kann. Dafür haben sie einen *Conversational Agent* erstellt und verschiedene Dialoge zu unterschiedlichen Themengebieten zwischen dem *Conversational Agent* und den Benutzern geführt. Die Dialoge wurden im Anschluss ausgewertet und aufgetretene Fehler wurden kategorisiert. Folgende fünf Kategorien wurden erkannt:

- *Clarify-Errors*: Dieser Fehler tritt auf, wenn der Chatbot auf eine Frage eines Kunden zwar angemessen antwortet, diese Antwort allerdings nicht die nützlichste oder informativste ist. Der Benutzer muss im Anschluss weiter auf seine Frage eingehen, um mehr Informationen vom Chatbot zu erhalten.
- *User-Repitition-Errors*: Dieser Fehler tritt auf, wenn der Benutzer sich wiederholen muss, weil der Chatbot nicht „zugehört" hat. Der Fehler ist eher technischer Natur und bedeutet, dass systembedingt eine Nachricht an den Chatbot nicht zugestellt werden konnte.
- *Agent-Repitition-Errors*: Dieser Fehler tritt auf, wenn der Chatbot zweimal die gleiche Antwort auf eine Frage hintereinander gibt und ist auch technischer Natur.
- *Turn-Taking-Errors*: Bei dieser Art des Fehlers wird ein Benutzer vom Chatbot während der Konversation unterbrochen. Das passiert, wenn der Chatbot plötzlich weitere Informationen liefert oder die Konversation falsch gedeutet hat.
- *Incoherent-Responses-Errors*: Dieser Fehler tritt auf, wenn der Chatbot eine Frage falsch verstanden hat und deswegen eine nicht ganz passende Antwort liefert. Das tritt insbesondere bei zweideutigen Fragen auf.

Nicht jede Fehlerart der genannten Kategorien hat dabei die gleiche Gewichtung. Zu vermeiden sind laut (Aneja, McDuff, & Czerwinski, 2020) besonders die *Turn-Taking-Errors*. Diese haben zur Folge, dass der *Chatbot* als besonders unfreundlich bewertet wird. Außerdem wird dieser als weniger intelligent eingestuft. Bei *Incoherent-Response-Errors* verhält es sich anders. Hier ist z. B. so, dass ein paar Fehler die Bewertung des *Chatbots* sogar positiv beeinflussen können. Der *Chatbot* wird als intelligenter eingestuft, wenn er eine Zweideutigkeit erkennt, an die der Benutzer nicht gedacht hat. Dennoch sollte diese Fehlerart, genau wie die anderen, nicht so häufig auftreten, weil die Wahrnehmung des *Chatbots* darunter leidet.

Damit die genannten Fehler nicht häufig auftreten, wurden *Dialogue Design Principles* entwickelt. Diese sollte man beim Erstellen von *Chatbot*-Dialogen beachten und sind in (Hutapea, 2017) detailliert nachzulesen. Folgend sind die wichtigsten Punkte zusammengefasst, die wir bei der Entwicklung der Dialoge unseres *Chatbots* beachtet haben:

- *Disambiguation*: Bei diesem *Design Principle* sollte der *Chatbot* im Dialog versuchen die Eingaben des Benutzers richtigzustellen, wenn dieser die Frage nicht verstanden hat. Es sollte nicht mit einem niedrigen Score bei der Intentanalyse verfahren werden.
- *Relaxation*: Dieses *Principle* sorgt dafür, dass der *Chatbot* „Ablehnung" im Chat anreichert und z. B. nach Alternativen sucht. Wird z. B. von einem Benutzer nach einem Flug heute gesucht, und es ist keiner verfügbar, so können Vorschläge für den Tag danach gemacht werden, anstatt ein „Nicht verfügbar" in den Chat zu schreiben.
- *Confirmation*: Hier wird versucht, die vom Benutzer übermittelten Informationen zu bestätigen, damit sichergestellt wird, dass keine Missverständnisse oder Fehleingaben aufgetreten sind.
- *Completion*: Bei diesem *Principle* wird der Benutzer vom Chatbot aufgefordert, noch fehlende Angaben auszufüllen, anstatt diese leer zu lassen.

Trotz der *Design Principle* ist es ziemlich wahrscheinlich, dass es zu Fehlern im Dialog bzw. der Konversation kommen wird. Hier ist es immer sinnvoll, als letzte Möglichkeit die Weiterleitung an einen Mitarbeiter einzubauen. Dieser wird dem Benutzer immer weiterhelfen können.

Wenn der *Chatbot* neben bestimmten *Use-Cases* auch *Smalltalk* mit dem Benutzer führen soll oder der Benutzer dazu gebracht werden soll, die Konversation fortzuführen, dann setzen Entwickler oft „*Language Tricks*" ein. Diese werden bei Konversationen benutzt, bei denen die Wahrscheinlichkeit niedrig ist, dass der Chatbot die Absicht des Benutzers richtig verstanden hat. Folgende Tricks könnten bei *Smalltalk* eingesetzt werden (Hutapea, 2017):

- *Switch Topic*: Hier ändert der *Chatbot* das Thema, über das geredet wird, um die Unwissenheit bzw. die nicht verstandene Absicht des Benutzers zu kaschieren. So wäre z. B. eine Änderung auf das Thema Wetter oder Sport möglich.
- *Ask Open Questions*: Anstelle von einfachen Antworten, wie „i don't know" wird „i don't know, could you tell me something interesting?" geantwortet. Beide Antworten gehen zwar in die gleiche Richtung, Letztere animiert den Benutzer allerdings, die Konversation fortzuführen.
- *Tell a joke*: Hier wird die Konversation fortgeführt, indem ein Witz erzählt wird, auf den der Benutzer im besten Fall reagiert.
- *Elicit more infomration*: Bei diesem Trick wird versucht, den Benutzer im Dialog zu halten, indem nach mehr Informationen gefragt wird, falls der *Chatbot* noch nicht gesammelt hat. Z. b. die Interessen oder Hobbys, um Produktempfehlungen zu geben.

In unserem Use-Case haben wir diese Tricks beachtet, wenn auch nur indirekt, weil der *Smalltalk* kein Hauptbestandteil unseres *Chatbots* ist.

III. KONZEPTION

Mithilfe der Grundlagen und des daraus gesammelten Wissens, wie man Dialoge erstellt und Benutzer am besten durch Dialoge führt, haben wir mit der Konzeption des *Chatbots* weitergemacht. Dafür mussten wir eine geeignete Plattform finden, mit der wir den *Chatbot* implementieren und für andere zugänglich machen können. Zur Auswahl standen das Microsoft Bot Framework, RASA und OmniBot. Diese Plattformen haben wir gewählt, weil bereits Erfahrungen im Umgang mit diesen Plattformen gesammelt wurden.
Für unseren Use-Case haben wir uns für OmniBot entschieden, weil wir dort den Chatbot in einem Designer erstellen können und mit einem Skript auch relativ einfach in eine Website einbinden können. Bei den anderen Frameworks müssten wir uns zusätzlich um das Hosting des Backends kümmern. Das wird von OmniBot in unserem Fall übernommen.

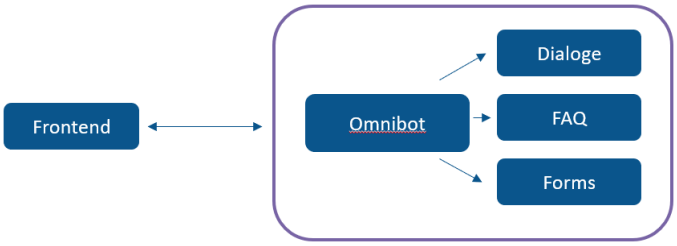

Abbildung 1 Systemaufbau des Chatbots

A. *Softwareumgebung*

Da es in diesem Projekt primär um Chatbots ging, wurde ein passendes Frontend auf GitHub gesucht. Der gewählte *One Pager* kann hier gefunden werden. Die Webseite wurde mithilfe des Apache HTTP Server bereitgestellt. An der Webseite selbst wurden lediglich kleine Änderungen, wie z. B. die Einbindung des Chatbots, vorgenommen. Zur Realisierung des Backends, also der Logik der Chatbots, wurde das Framework OmniBot verwendet. Die gleichnamige Firma ist in Oldenburg ansässig und war eines der ersten Unternehmen in Europa, welches eine komplette Voice und Conversational AI Plattform, auf Basis eigener Technologien, zur Verfügung stellen konnte (Omnibot, 2021). Zur Implementierung der Chatbots wird der OmniBot Designer, eine Art IDE für Chatbots, verwendet. Im folgenden Abschnitt wird erklärt, wie die unterschiedlichen Funktionalitäten unseres Chatbots durch Verwendung des Designers realisiert wurden.

B. *Modellierung der Dialoge*

Der OmniBot Designer, gezeigt in Abbildung 9 im Anhang, bietet im Wesentlichen drei verschiedene Konzepte, um mit einem Nutzer zu kommunizieren: Dialoge, *Knowledge Bases* und Formen (Abbildung 1). Dialoge dienen hauptsächlich dazu, den Nutzer durch ein Gespräch zu führen. Zum Starten eines Dialoges können verschiedene Events, wie z. B. das Initialisieren des Chatbots, oder Filter verwendet werden. Diese Filter reagieren auf bestimmte Intentionen, in unserem Fall beispielsweise das Fragen nach Pizza. Zur Steuerung des Gesprächverlaufs können Fragen, Verzweigungen oder eigene PHP-Blöcke verwendet werden. Wie auf Abbildung 9 zu erkennen ist, ähneln die Dialoge Flussdiagrammen. Des Weiteren können *events* und *client commands* eingebracht werden, welche das Frontend steuern oder auf dieses reagieren können. Beim Anbieten einer Pizza wird so z. B. automatisch zum Menü gescrollt.
Knowledge Bases hingegen dienen der Bereitstellung von Antworten auf Fragen, welche in einem kurzen Absatz beantwortet und in Kategorien sortiert werden können. Ein Anwendungsbeispiel dafür sind FAQs, also Antworten auf häufig gestellten Fragen. Abbildung 2 zeigt die Sicht auf unsere *Knowledge Base*.

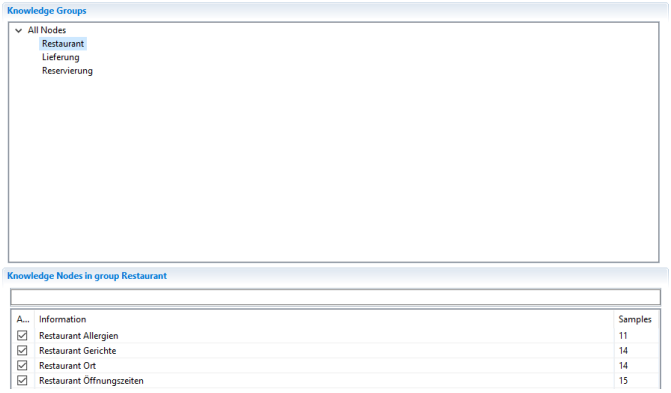

Abbildung 2 Knowledge Base FAQ

Die letzte Komponente, Formen, werden in Kombination mit Dialogen verwendet. Durch Formen können Informationen, z. B. die des Nutzers, einfacher abgefragt werden als mit puren Dialogen. Der Chatbot erfragt selbstständig alle Felder der Form, das kann zum Beispiel die E-Mail-Adresse des Nutzers sein, und validiert die Antworten. Das Verhalten bei fehlerhaften Eingaben wird ebenfalls in der Form selbst spezifiziert. In Abbildung 3 ist die Form „order“ unserer Implementation zu erkennen.

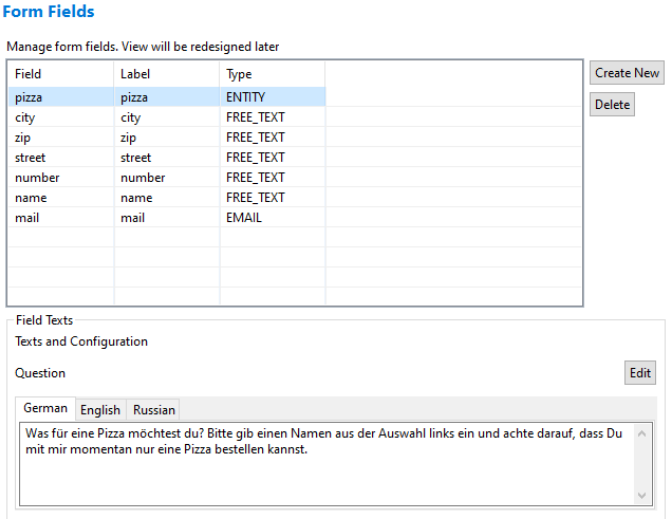

Abbildung 3 Form "order"

Unser Prototyp verfügt über verschiedene Dialoge, Formen und *Knowledge Bases*, welche in Tabelle 1**Fehler! Verweisquelle konnte nicht gefunden werden.** erklärt werden.

Komponente	Nutzen
Dialog „escalation“	Behandlung von nicht erkannten Intents.
Dialog „greeting“	Begrüßung des Nutzers und Einführung in die Verwendung eines Chatbots.
Dialog „ordering“	Dialog zum Bestellen einer Pizza, greift auf die Form „order“ zurück.
Dialog „scheduling“	Dialog zum Reservieren eines Tisches, greift auf die Form „reservation“ zurück.
Form „order“	Enthält alle Felder, welche nötig sind, um eine Pizza zu bestellen (Adresse etc.).
Form „reservation“	Enthält alle Felder, welche nötig sind, um einen Tisch zu reservieren (Name etc.).
Knowledge Base „FaQ“	Enthält einige Antworten auf häufig gestellte Fragen.

Tabelle 1 Komponenten des Chatbots und ihr Nutzen

Ein weiteres wichtiges Konzept sind die Entitäten. Entitäten bezeichnen wichtige Textbestandteile, wie z. B. Namen, Orte oder Zeitangaben. Der OmniBot Designer bietet Zugriff auf vordefinierte Entitäten, unter anderem Namen oder E-Mail-Adressen, und ermöglicht das Einbinden eigener neuer Entitäten. In unserem Fall haben wir weitere Entitäten, in Form verschiedener Pizzen, eingepflegt, um die Eingaben der Nutzer zu validieren.
Der fertige Chatbot kann auf das OmniBot-Backend hochgeladen und per Skript in eine Webseite eingebettet werden.

IV. Evaluation

A. *Testablauf*

Der Test bestand aus zwei Phasen. In der ersten Phase sollten die Teilnehmer die Pizza-Webseite aufrufen und dort mit dem Chatbot interagieren. Sie wurden angewiesen, eine Pizza zu bestellen, einen Tisch zu reservieren und allgemeine Fragen in Bezug auf das Restaurant selbst zu stellen. Im Anschluss sollten die Teilnehmer eine Umfrage auf Google Forms auszufüllen. Diese wurde abschließend durch die integrierten Funktionen und in einem Jupyter Notebook ausgewertet.

B. *Aufbau der Umfrage*

Die Umfrage wurde in drei verschiedene Abschnitte gegliedert. Im ersten Abschnitt wurden Informationen zu den teilnehmenden Personen gesammelt. Das umfasste Fragen zum Alter, zur technischen Versiertheit und bisherige Erfahrungen mit Chatbots. Im zweiten Teil der Umfrage ging es um die Bewertung des Chatbots. In diesem Fall konnte die Intelligenz, die Sympathie und die Menschlichkeit auf einer Skala von eins (niedrig) bis sechs (hoch) bewertet werden. Der letzte Teil beinhaltet Fragen zur Interaktivität, also z. B. zum automatischen Scrollen. Zudem gab es ein Freitextfeld für generelles Feedback.

C. *Auswertung der Ergebnisse*

In diesem Kapitel werden die Ergebnisse der Umfrage dargestellt und diskutiert. Zudem werden einige Probleme bei der Kommunikation mit Nutzern anhand von Logs exemplarisch aufgezeigt.
In Log 1 entscheidet sich ein Teilnehmer in Bezug auf die initiale Frage „Wir können dir eine Pizza anbieten. Hast du Lust?“ um. Der Chatbot hat zu diesem Zeitpunkt keinen Bezug mehr und kann den Input des Teilnehmers dementsprechend nicht behandeln. In Log 2 fragt ein Nutzer nach Hilfe, worauf der Chatbot keine Antwort hat. Um die Kommunikationsqualität mit dem Chatbot zu erhöhen, sollten also Lücken in dieser ausfindig gemacht und gefüllt werden. Dabei ist es besonders wichtig, dass der Chatbot von mehreren Personen implementiert und getestet wird, damit dieser möglichst viele Intentionen abdeckt. Zudem sollten die Dialoge persistenter gestaltet werden, sodass sich Nutzer auch im Nachhinein umentscheiden können.

text_out	Wir können dir eine Pizza anbieten. Hast du Lust?	11:41:45
text_in	Nein	11:41:54
text_out	Okay, schade...	11:41:55
text_in	Oder doch!	11:42:03
event_in	voice_input_not_handled	11:42:03
event_out	voice_input_not_handled	11:42:03
text_out	Ich habe Dich leider nicht verstanden. Bitte versuche es noch einmal.	11:42:03
text_in	Doch	11:42:09
event_in	voice_input_not_handled	11:42:10
event_out	voice_input_not_handled	11:42:10
text_out	Ich habe Dich noch immer nicht verstanden. Versuche bitte Deinen Text anders zu formulieren.	11:42:10
text_in	Ich möchte doch eine Pizza	11:42:23
php_call	get_dish_value(...) (gen_functions.php), duration: 132ms	11:42:23
text_out	Wir können dir eine Pizza anbieten. Hast du Lust?	11:42:23
text_in	Ja	11:42:29

Log 1 Ausschnitt einer Konversation

text_in	ich brauche hilfe	10:08:08
php_call	get_dish_value(...) (gen_functions.php), duration: 260ms	10:08:09
text_out	Wir können dir eine Pizza anbieten. Hast du Lust?	10:08:09

Log 2 Ausschnitt einer Konversation 2

Abbildung 4 zeigt die diskrete Altersverteilung der Teilnehmer. Insgesamt haben 15 Personen an der Umfrage teilgenommen. Wie zu erkennen ist, war ein Großteil der Teilnehmenden 20 bis 30 Jahre alt. Die restlichen drei Teilnehmer befanden sich in der Altersspanne 50-55. Ein Defizit gab es also bei Kindern, Jugendlichen und Personen des mittleren Alters (30-50).

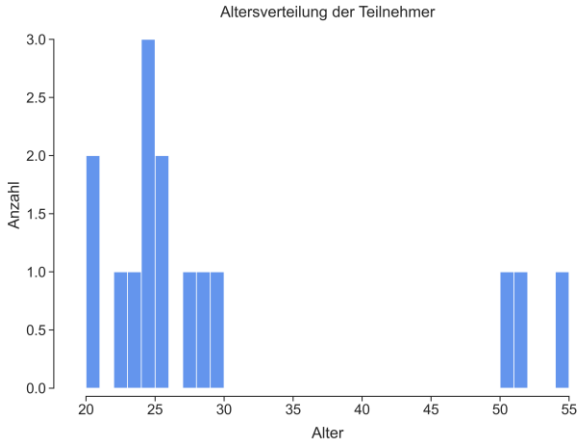

Abbildung 4 Altersverteilung der Benutzer unserer Umfrage

Abbildung 5 ist die Auswertung der Bewertung verschiedener Eigenschaften des Chatbots. Die Bewertungskategorien, Menschlichkeit, Intelligenz und Sympathie, befinden sich auf der x-Achse der Darstellung. Außerdem ist jede Kategorie in die „technische Versiertheit“, in diesem Fall von den Teilnehmern selbst bewertet, aufgeteilt, um ggfs. auftretende Abweichungen zwischen diesen Gruppen zu visualisieren. Die Legende in der oberen rechten Ecke der Abbildung beschreibt die Farbgebung: Orange steht für technisch nicht versiert, Grün für das Gegenteil. Die y-Achse stellt die Bewertungsskala von eins bis sechs dar. Diskrete Werte, also Antworten aller Nutzer, werden durch die schwarzen Punkte repräsentiert. Die diskreten Verteilungen der einzelnen Kategorien werden zudem durch eine kontinuierliche Verteilung per *kernel density estimation*

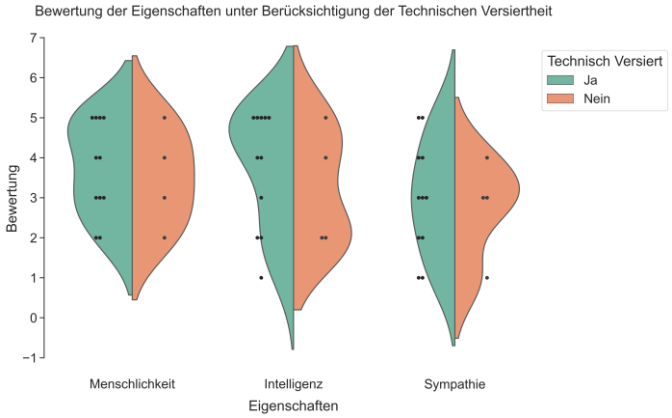

Abbildung 5 Bewertung der Eigenschaften des Chatbots unter Berücksichtigung der technischen Versiertheit

approximiert, wodurch die Schiefe und Multimodalität der Verteilungen hervorgehoben werden. Auffällig ist, dass die Bewertungen jeder Kategorie sehr stark gestreut sind. Das könnte sowohl an der kleinen *sample size* von 15, aber auch an unterschiedlichen Erwartungshaltungen und Interpretationen der Bewertungskategorien liegen. Die Kategorien Menschlichkeit und Sympathie weisen keine signifikante Schiefe auf. Lediglich bei der Bewertung der Intelligenz, unter den technisch versierten Teilnehmern, ist zu erkennen, dass diese als eher hoch eingeschätzt wird.

Abbildung 6 bis Abbildung 8 zeigen die Antworten aller Teilnehmer auf verschiedene Ja/Nein-Fragen als Kuchendiagramme. Die Farbe Blau steht dabei für „Ja“, während Rot „Nein“ repräsentiert. Über 93% der Teilnehmer empfinden das interaktive Scrollen als sinnvoll und könnten sich vorstellen, nicht nur das Chatfenster, sondern auch die Webseite selbst als Input für den Dialog mit dem Chatbot zu nutzen. Zwei Drittel der Nutzer würden den Chatbot als Methode zum Bestellen von Pizza in Anbetracht ziehen. Als Gründe dagegen wurden ein zu hoher Aufwand, Kommunikationsprobleme und ein zu langer Bestellvorgang genannt. Um diesen Faktoren entgegenzuwirken, sollten Wege gesucht werden, die Kommunikation effizienter und kurzweiliger zu gestalten. Zudem sollte es in jedem Fall eine Alternative zum Bestellen und Reservieren, zum Beispiel per Telefon oder Webseite, geben.

Abbildung 6 Bewertung von interaktiven Elementen

Abbildung 7 Einschätzung der Benutzer über weitere interaktive Elemente

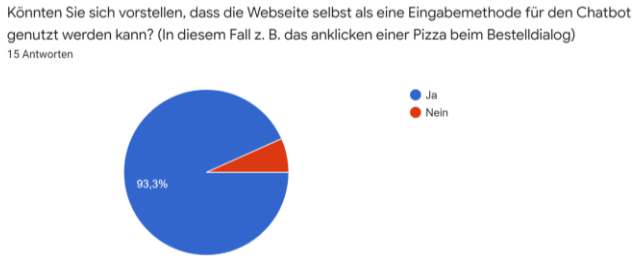

Abbildung 8 Bewertung über den generellen Einsatz von Chatbots im Order-Management

V. Fazit

In unserem Projekt haben wir gelernt, dass bei der Erstellung von Dialogen und beim Trainieren der Erkennungsfunktionen auf möglichst diverse Beispielsätze geachtet werden muss. Es ist sehr schwer, die unterschiedlichen Anfragemöglichkeiten der Benutzer abzudecken bzw. zu Versuchen, mögliche Fragen anderer Benutzer vorherzusehen. Für diesen Schritt muss im Vorfeld eine eigenständige Evaluation von Fragemöglichkeiten der Benutzer getätigt werden, um die unterschiedlichen Arten zu sammeln. Als Beispiel könnte es eine Umfrage geben, bei denen Nutzer angeben sollen, wie sie nach einem Standort eines Restaurants fragen würden.

Außerdem ist es wichtig, die Dialoge robust zu gestalten, und mit allen möglichen Antworten von Benutzern zu rechnen. Wir haben bei der Auswertung erkannt, dass der Wille der Benutzer sehr hoch ist, den Chatbot auf die Probe zu stellen und möglichst komplexe Anfragen zu machen.

Zusätzlich sollte man viele Interaktionsmöglichkeiten zwischen Chatbot und Website einbauen. Das bedeutet, dass man z. B. die Pizzen beim Bestellvorgang auch auf der Website anklicken können sollte, anstatt den Namen der Pizza im Chat zu nennen. Bei diesen Interaktionsmöglichkeiten muss man allerdings abwägen, dass ein Warenkorbsystem beim Bestellvorgang zum Teil als effizienter angesehen und von einigen Benutzern als

besser empfunden wird. Hier würde der Chatbot sehr gut als *Concierge-Service* dem Benutzer zur Seite stehen und alle möglichen Fragen zum Restaurant beantworten.
Dennoch war der „WOW-Effekt“, gerade bei den Benutzern höheren Alters hoch, weil der Chatbot oft unerwartet die Absicht verstanden hat und Informationen richtig aufgenommen hat.
Die eingangs gestellt Frage, wie sinnvoll ein Chatbot im Order Management ist, lässt sich mit „ist sinnvoll“ beantworten. Wenn der Chatbot ausgereift ist, eine gute Grundlage an Beispielsätzen besitzt und die Benutzer angemessen und mit etwas Humor durch Dialoge leitet, dann ist dieser für das Marketing des Restaurants förderlich und hilft bei auftretenden Fragen von potenziellen Kunden zum Restaurant.

VI. LITERATURVERZEICHNIS

Aneja, D., McDuff, D., & Czerwinski, M. (2020). *Conversational Error Analysis in Human-Agent Interaction.* NY, USA: ACM.

Hutapea, A. (2017). *CHATBOT: Architecure, Design & Development.* Pennsylvania: University of Pennsylvania.

IBM. (05. 08 2021). *Was ist ein Chatbot*. Von https://www.ibm.com/de-de/campaign/was-ist-ein-chatbot abgerufen

Lyzanets, K. (22. 12 2019). *Everyone Needs Their Own Chatbot*. Von https://towardsdatascience.com/everyone-needs-their-own-chatbot-72cb210c0161 abgerufen

Omnibot. (05. 08 2021). *Omnibot Homepage*. Von https://omnibot.ai/ abgerufen

Pykes, K. (31. 03 2021). *Build A Simple Chatbot In Python With Deep Learning*. Von https://towardsdatascience.com/a-simple-chatbot-in-python-with-deep-learning-3e8669997758 abgerufen

VII. ANHANG

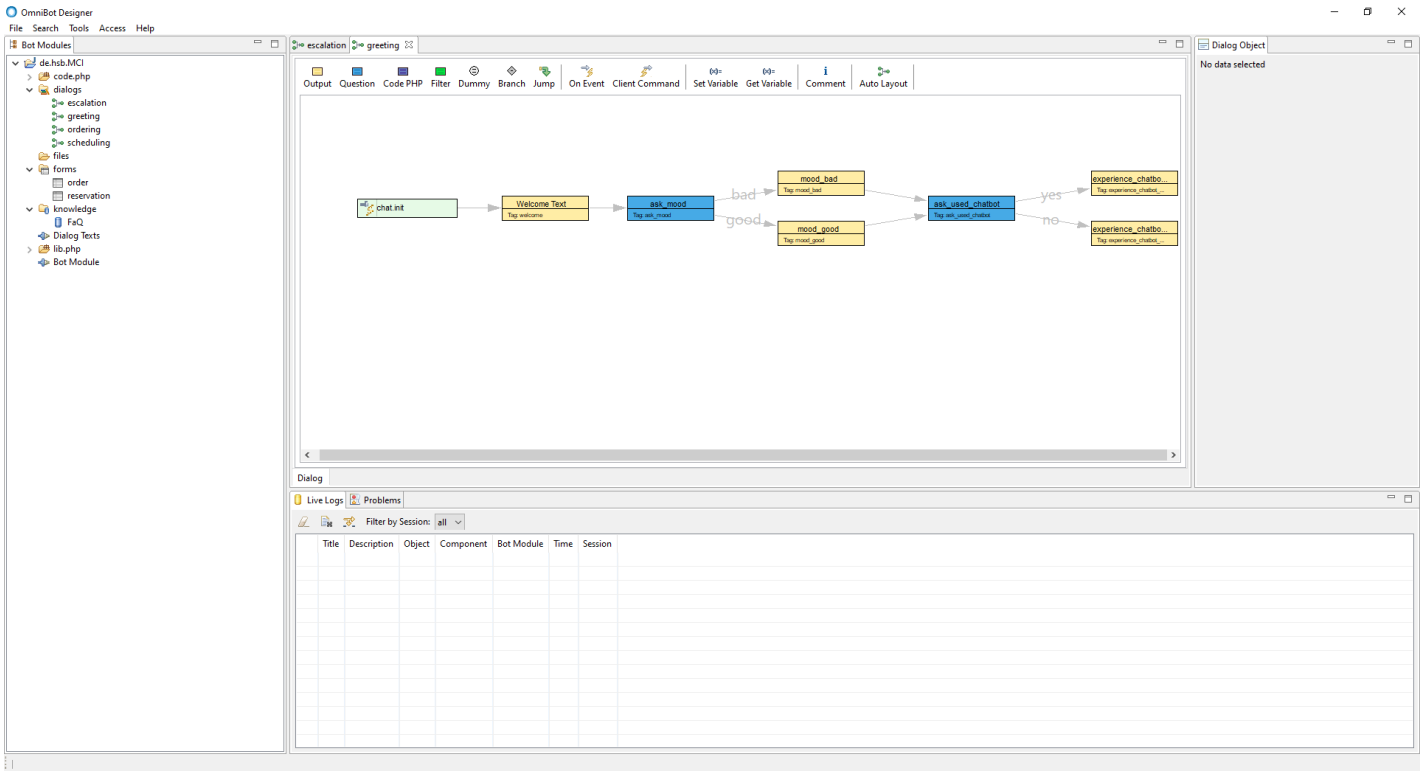

Abbildung 9 Omnibot-Designer