MUUUH! GROUP

DIE ANATOMIE EINES CHATBOTS

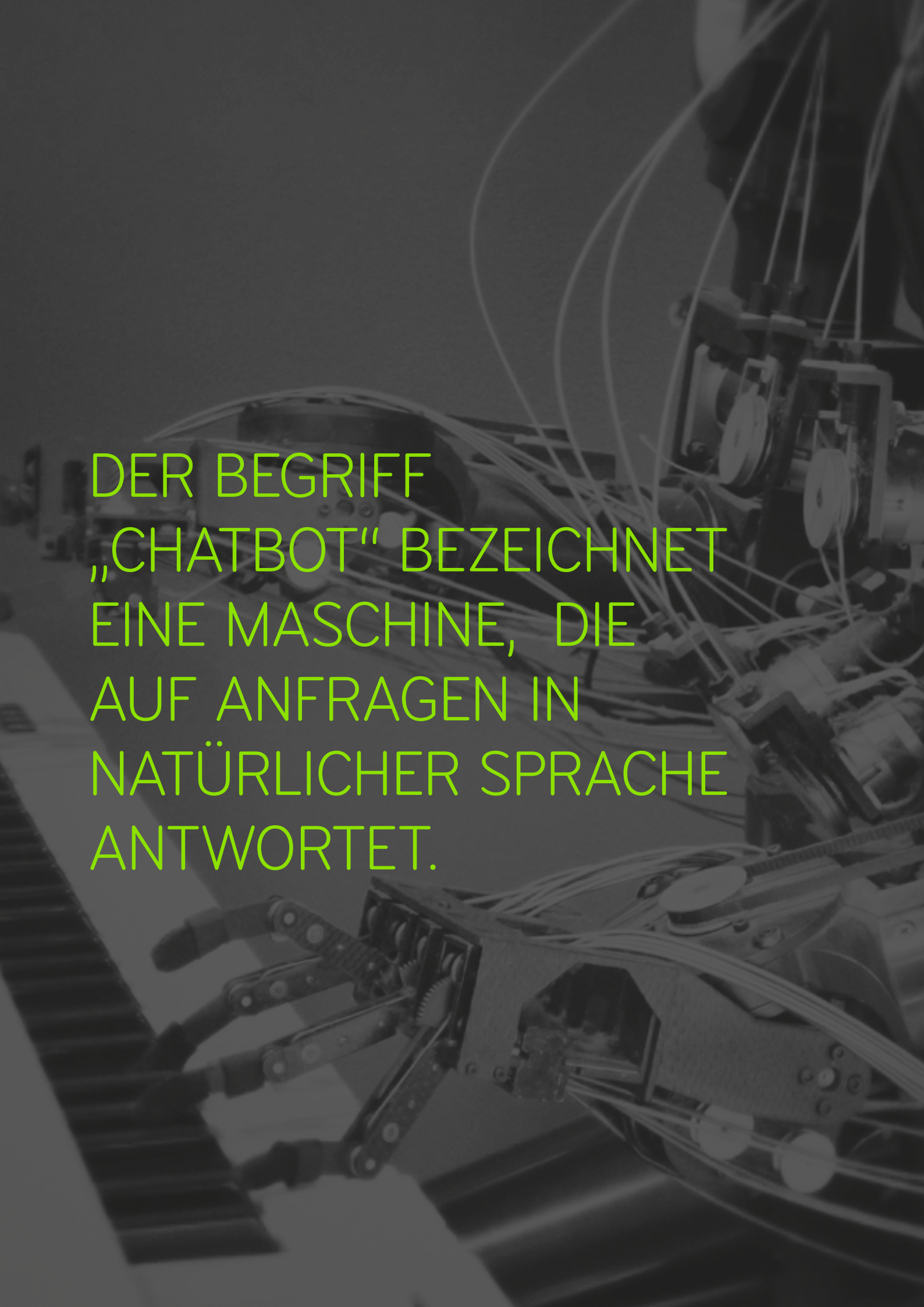
VORWORT

Im Gartner Hype Cycle stehen Conversational User Interfaces kurz vor dem „Gipfel der überzogenen Erwartungen". Da überzogene Erwartungen und harte Enttäuschungen eng beieinander liegen, haben wir es uns zur Aufgabe gemacht aufzuklären: Welche Usecases lassen sich realistisch in Conversational User Interfaces abbilden? Welche nicht? Was geht heute? Was wird morgen gehen? Der wichtigste Schritt für realistische Erwartungen, ist ein Verständnis dafür, was ein Chatbot in einem Conversational User Interface leisten kann.

Grundsätzlich ist ein Chatbot eine Maschine, die in der Lage ist, auf Anfragen in natürlicher Sprache zu antworten. Der Dialog mit Bots kann via Text oder Voice in einer Vielzahl verschiedener Kanäle erfolgen. Beispiele sind Whats App, Slack, Facebook oder Alexa, aber auch Dialogfenster auf der Firmenwebsite oder einer App. All diese Dialogvarianten erfolgen in sogenannten Conversational User Interfaces; also in Oberflächen, bei denen die Mensch-Maschine-Interaktion auf einer Konversation beruht. Conversational User Interfaces orientieren sich an natürlicher Sprache, was sie für Nutzer deutlich intuitiver macht, als klassische Graphical User Interfaces.

In diesem Paper widmen wir uns der Frage wie Chatbots funktionieren und wie sie aufgebaut sind. Anhand unseres Conversational-Layer-Modells führen wir Schicht für Schicht durch die verschiedenen technologischen Elemente und Methoden von Chatbots und erläutern deren Zusammenspiel. Da dieses Paper ein komplexes Thema behandelt, haben wir großen Wert auf klare und verständliche Erklärungen und eingängige Beispiele gelegt.

Eine erkenntnisreiche Lektüre wünscht



DER BEGRIFF „CHATBOT“ BEZEICHNET EINE MASCHINE, DIE AUF ANFRAGEN IN NATÜRLICHER SPRACHE ANTWORTET.

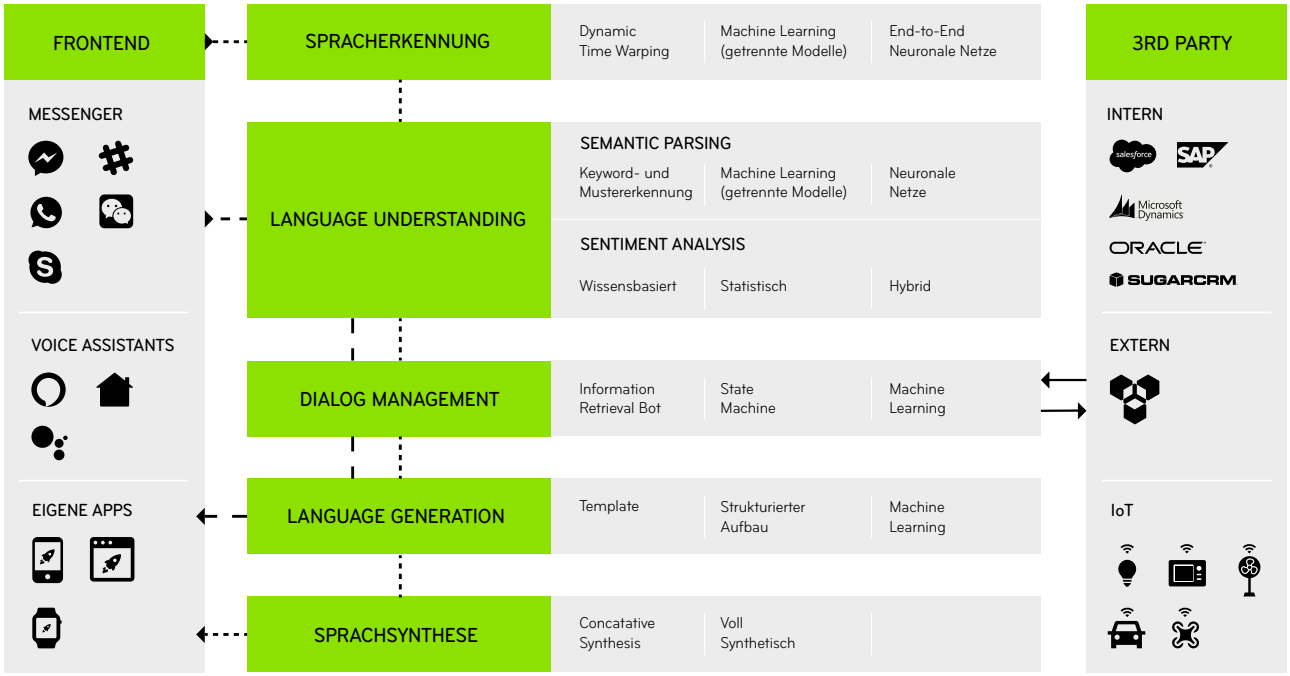
INHALTSVERZEICHNIS

I. Conversational-Layer-Model:

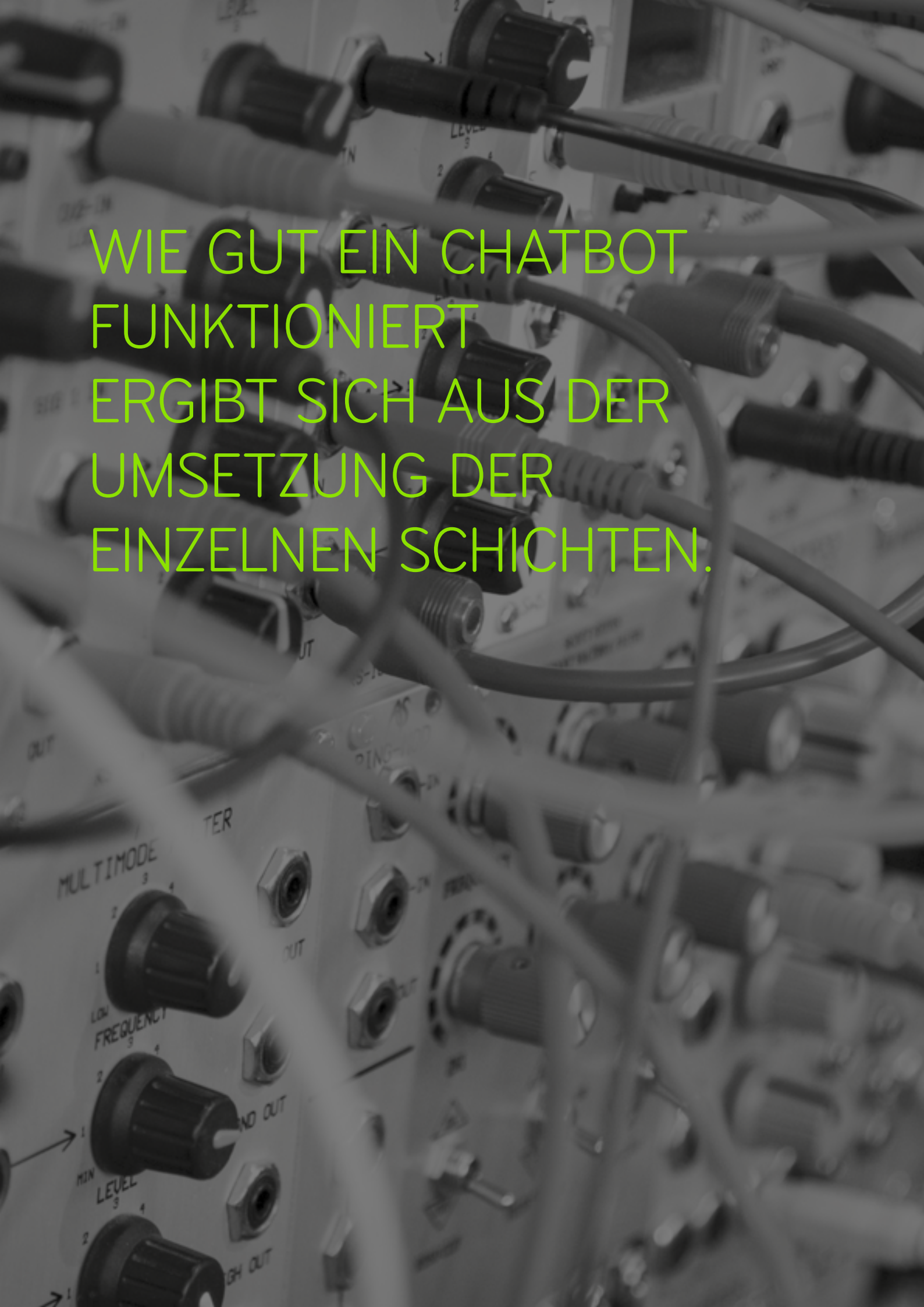
WIE DIE VERSCHIEDENEN TECHNOLOGIEN ZUSAMMEN EINEN CHATBOT ERGEBEN

Im Grunde funktionieren alle Chatbots auf eine sehr ähnliche Art und Weise. Der Bot bekommt aus dem **Frontend** die Eingabe des Nutzers übergeben. Im Falle eines Voicebots geschieht dies in Form einer Tonaufnahme und im Falle eines Textbots, in Form von Text. Bei einem Voicebot werden die gesprochenen Worte von der **Spracherkennung** transkribiert. Dieser Text wird dann von der **Language Understanding** Schicht inhaltlich analysiert und in eine Form gebracht, die vom Dialogmanagement verarbeitet werden kann. Bei einem Textbot bekommt die Language Understanding Schicht den Text direkt aus dem Frontend, zum Beispiel direkt vom Facebook Messenger.

CONVERSATIONAL-LAYERS



Im **Dialogmanagement** werden die Entscheidungen getroffen. Das heißt, dass der Bot entscheiden muss, welche Aktionen er ausführen und was er antworten soll. In vielen Fällen kann das Dialogmanagement auf **3rd Party** Applikationen zugreifen. Zum Beispiel kann der Chatbot das Wetter vom deutschen Wetterdienst abfragen, die Beleuchtung eines Smart Home Devices aktivieren oder Daten in eine Datenbank schreiben. Wenn das Dialogmanagement eine inhaltlich passende Antwort festgelegt hat, muss diese Antwort noch von der **Language Generation** Schicht in eine für Menschen verständliche Form gebracht werden. In aller Regel heißt das, dass ein Satz formuliert werden muss.



WIE GUT EIN CHATBOT FUNKTIONIERT ERGIBT SICH AUS DER UMSETZUNG DER EINZELNEN SCHICHTEN.

Bei einem Textbot wird dem Nutzer nun dieser Satz im Frontend angezeigt. Bei einem Voicebot wird der Satz an die **Sprachsynthese** übergeben, welche ihn in gesprochene Worte umwandelt und dem Frontend übergibt. Voicebots und Textbots funktionieren also fast identisch, mit dem kleinen Unterschied, dass Voicebots eine Spracherkennung und Sprachsynthese als zusätzliche Schichten benötigen. Obwohl alle Bots ähnlich aufgebaut sind, ergeben sich durch die unterschiedlichen Methoden innerhalb jeder Schicht erhebliche Unterschiede für die umsetzbaren Usecases und die User Experience. Wenn zum Beispiel im Dialogmanagement Machine Learning anstelle eines einfachen Information Retrieval Bots (siehe graue Kästchen im Model) verwendet wird, kann eine bedeutend natürlichere Dialogführung erreicht werden. In den folgenden Kapiteln werden die einzelnen Schichten intensiv beleuchtet und Unterschiede der jeweiligen technischen Methoden herausgearbeitet.

II. Spracherkennung:
MEIN COMPUTER VERSTEHT WAS ICH SAGE

Die Language Understanding Schicht benötigt geschriebenen Text als Eingabe. Bei einem Voicebot muss dieser Text jedoch erst erzeugt werden. Das passiert in der Spracherkennung in der die gesprochenen Worte des Nutzers folgendermaßen transkribiert werden:
Nach der Vorverarbeitung, in welcher das Tonsignal digitalisiert und gefiltert wird, folgt die eigentliche Erkennung. In der Erkennung gibt es zwei nennenswerte Verfahren, welche beide mit Machine Learning Algorithmen arbeiten. Zum einen das Verfahren der Spracherkennung mit getrennten Modellen, das aktuell am weitesten verbreitet ist, sowie das Verfahren der End-to-end-Erkennung mit Neuronalen Netzen. In der Vergangenheit wurde auch Dynamic Time Warping (DTW) als Methode genutzt, was man noch aus alten Telefon-Warteschleifen kennt: „Sagen sie '*Berater*' wenn Sie mit einem Kundenberater sprechen möchten." DTW hat eine bedeutend geringere Präzision als Methoden, welche auf Machine Learning[1] basieren.

Wenn in der Spracherkennung mit Machine Learning und getrennten Modellen gearbeitet wird, werden drei Modelle getrennt voneinander mit Machine Learning trainiert.

1. Das erste Model erkennt Laute. Wenn der Nutzer „Pizza" sagt, wird dieses Model „Pi" und „za" erkennen und dem nächsten Model übergeben.
2. Das zweite Model erkennt Wörter. Es bekommt vom ersten Model die Information, dass der Nutzer wahrscheinlich die Laute „Pi" und „za" gesagt hat. Aus dieser Information schließt das zweite Model, dass der Nutzer vermutlich „Pizza" gesagt hat.
3. Das dritte Model bekommt die einzelnen Wörter aus dem zweiten Model und formt hieraus ganze Sätze.

Man kann sich also die Zusammenarbeit der drei Modelle wie in einer Fertigungsstraße vorstellen. Das erste Model bekommt den Rohstoff (die Tonaufnahme) und fertigt daraus Laute (Pi, Pa, Fi, Fa, Lu...), das zweite Model bekommt die Laute und fertigt daraus Wörter und das dritte bekommt die Wörter und fertigt daraus ganze Sätze. Durch diese Zusammenarbeit der

[1] Mehr Informationen über Machine Learning befinden sich in der Infobox auf der nächsten Seite.

verschiedenen Modelle wird eine hohe Präzision erreicht. Wenn das erste Modell zum Beispiel fälschlicherweise die Laute „Fi" und „za" erkennt, erkennt das zweite Model, dass es das Wort „Fiza" nicht gibt und der Nutzer vermutlich „Pizza" gesagt hat. Das gleiche gilt für das dritte Model. Wenn das dritte Model die Worte „Ich" „möchte" „Pizza" „saufen" vom zweiten Model übergeben bekommt, wird es erkennen, dass das Wort „saufen" an dieser Stelle unwahrscheinlich ist und es durch „kaufen" ersetzen.

Spracherkennung mit getrennten Modellen wird aktuell in den meisten Voicebots verwendet. Sie hat allerdings den Nachteil, dass die getrennten Modelle mit einer Größe von mehreren Gigabyte zu speicherintensiv für den Betrieb auf mobilen Endgeräten sind. Dies zwingt die Spracherkennung noch dazu, die Audioaufnahme des Nutzers in die Cloud zu streamen, wo ein Ergebnis produziert wird und dann wieder zurück gesendet wird. Aus diesem Grund funktioniert Siri nur wenn eine Verbindung zum Internet besteht.

Bei End-to-End Spracherkennung, wird ein Neuronales Netz darauf trainiert, direkt vom digitalisierten Tonsignal die gesprochenen Sätze zu erkennen. Dieser Ansatz ist sehr vielversprechend, da ein trainiertes Neuronales Netz wesentlich weniger Speicherintensiv ist, als die getrennten Modelle. Dies bedeutet, dass eine Spracherkennung ohne aktive Internetverbindung möglich ist. End-to-End Spracherkennung hat sich allerdings noch nicht durchgesetzt, da erst seit etwa 2016 hinreichende Erkennungsgenauigkeit erreicht wird[2].

Machine Learning ist der Oberbegriff für alle Methoden bei denen ein Computer aus Erfahrung lernt. Der folgende Absatz erklärt Machine Learning am Beispiel einer einfachen Spamerkennung:

Das System wird mit Daten trainiert, welche bereits markiert sind. Das heißt es hat bereits ein Mensch alle E-Mails durchgeschaut und als "Spam" oder "Sinnvoll" markiert. Im zweiten Schritt werden (ebenfalls durch einen Menschen) die Eigenschaften (Features) extrahiert, welche für die Spamerkennung wichtig sind. Dies könnte bei der Spamerkennung zum Beispiel die Häufigkeit an komplett GROßGESCHRIEBENEN Wörtern, die Häufigkeit des Wortes Viagra und für sinnvolle E-Mails die persönliche Ansprache sein. Danach erfolgt das Training. Der Computer stellt eine Formel auf : spamfaktor = x * GROßSCHREIBUNG + y * Viagra - z * Ansprache. Die Variablen x, y und z sind die Gewichtungen der drei Faktoren. Nun muss der Computer festlegen wie wichtig die drei Faktoren sind. Dafür nimmt er zuerst an, dass die Großschreibung, Viagra, und die persönliche Ansprache gleich wichtig sind, also x = 1, y = 1, z = 1. Mit dieser Formel geht der Computer nun durch alle markierten E-Mails und schaut wieviele der E-Mails er mit der Formel korrekt sortiert. Danach verändert er die Faktoren zum Beispiel zu x = 1, y = 1, z = 0,5 und schaut ob mit den neuen Gewichten mehr E-Mails korrekt sortiert. Wenn das Training abgeschlossen ist, hat der Computer eine Formel die zum Beispiel so aussehen könnte: spamfaktor = 0,7386 * GROßSCHREIBUNG + 0,9122 * Viagra - 0,4876 * Ansprache. Wenn diese Formel nun im E-Mail Programm verwendet wird und der Nutzer selber E-Mails, die falsch sortiert wurden neu als "Sinnvoll" oder "Spam" markiert, kann der Computer das Training mit den neuen Daten wiederholen und im Laufe der Zeit immer genauer werden.

[2] https://arxiv.org/abs/1612.02695

III. Language Understanding:
MEIN COMPUTER VERSTEHT WAS ICH SCHREIBE

Sobald die Aussage des Nutzers in Textform vorliegt, wird sie der Language Understanding Schicht übergeben. Entweder bekommt sie den Text von der Spracherkennung, oder direkt als Eingabe des Nutzers (bei einem Textbot). In dieser Schicht wird dem Text die Bedeutung entnommen.
Language Understanding teilt sich in zwei Aufgaben: Zum einen wird extrahiert WAS der Nutzer sagt. Das heißt dass der Bot aus dem Satz "Ich möchte eine Pizza Tonno nach Hause bestellen" die Intention **pizza_bestellen**, die Sorte **pizza_tonno** und den Ort **zuhause** extrahiert.
Zum anderen ist es möglich nicht nur zu analysieren WAS der Nutzer sagt, sondern auch WIE er es sagt. Hier wird die Stimmung des Nutzers analysiert.

> Diese Schicht wird oft "Natural Language Understanding (NLU)" genannt. Der Begriff NLU wird allerdings ebenfalls für eine Untermethode, also das Language Understanding mittels Machine Learning verwendet. Um sprachlich präzise zu bleiben, nutzen wir den etwas breiteren Begriff „Language Understanding".

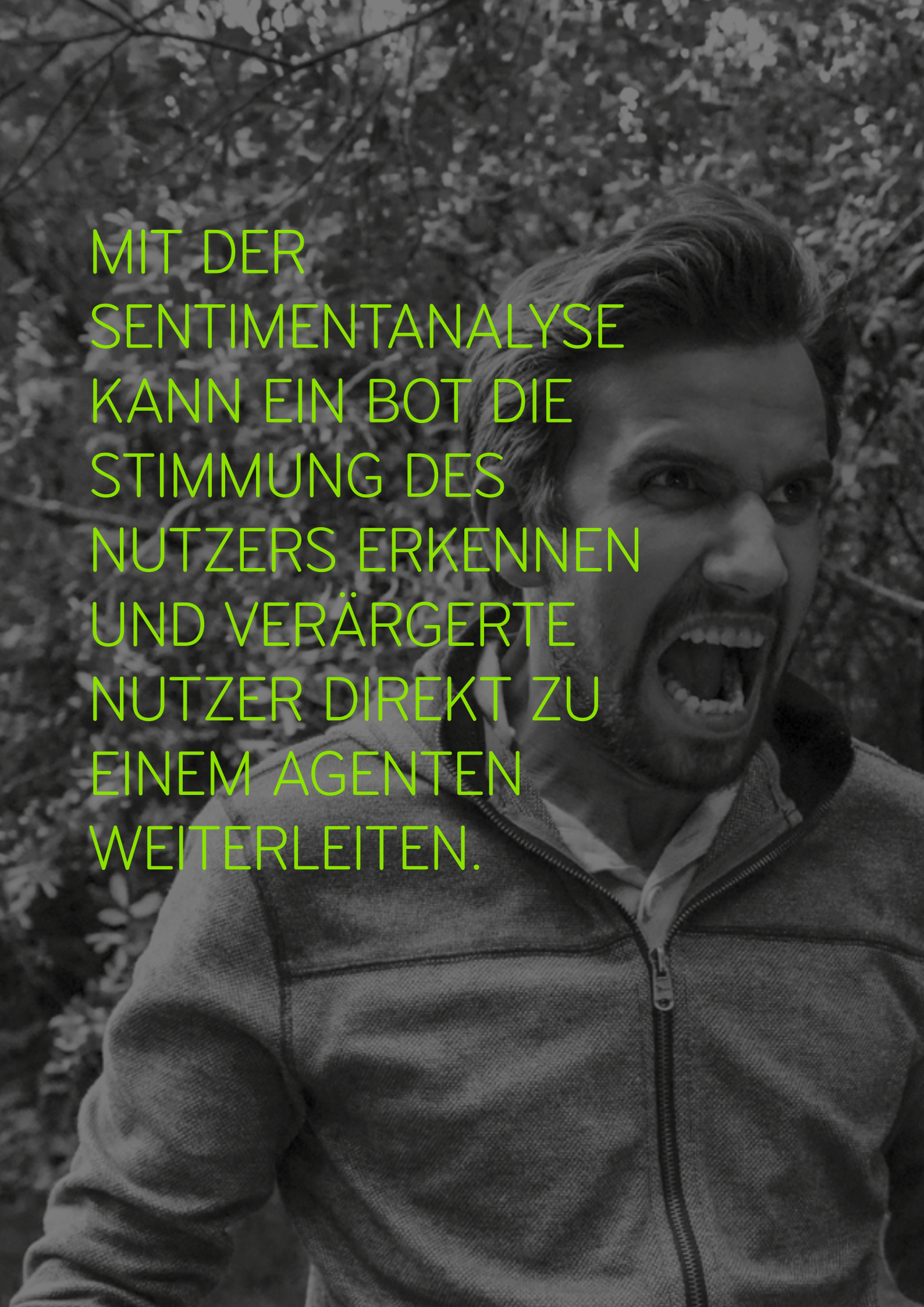
DAS "WAS" ERKENNEN

In der semantischen Analyse werden verschiedene Methoden genutzt, um zwei Informationsarten zu erkennen. Die erste ist die "Named Entity Recognition" (NER). NER extrahiert Eigennamen und markiert sie. Aus dem Satz "Ich möchte eine Pizza Tonno nach Hause bestellen.", wird zum Beispiel der Eigenname "Pizza Tonno" erkannt und mit dem Label **PIZZA_SORTE** markiert, sowie "nach Hause" mit dem Label **ORT**.
Die zweite Informationsart ist die Intention des Nutzers (Intent recognition). Hier werden im Kontrast zur NER nicht einzelne Wörter markiert, sondern der gesamten Aussage eine Intention zugewiesen. Dem Satz „Ich möchte Pizza bestellen." würde zum Beispiel die Intention **pizza_bestellen** zugewiesen. Die Präzision der semantischen Analyse hängt stark von den genutzten Methoden ab. Wenn zum Beispiel eine einfache Keyworderkennung genutzt wird, wird der Bot die Antwort „Ja ich möchte die Pizza bestellen" verstehen, aber „Sehr gerne möchte ich die Pizza bestellen" nicht, da er nach der Antwort „Ja" oder „Nein" sucht. Je weiter die semantische Analyse entwickelt ist, desto erfolgreicher kann der Bot natürliche und abwechslungsreiche Sprache erkennen.

DAS "WIE" ERKENNEN

Die Analyse der Stimmung des Nutzers, auch Sentimentanalyse genannt, ist der Versuch "zwischen den Zeilen zu lesen". Der am weitesten verbreitete Fall ist die dichotomische Sentiment Analyse. Das heißt, die durch die Sentimentanalyse bewerteten Aussagen werden mit dem Label "Positive Emotion" oder "Negative Emotion" (und manchmal "Neutral") versehen. Eine Schwierigkeit ist die mangelnde Objektivität. Wenn verschiedene Menschen eine größere Anzahl Aussagen als positiv oder negativ markieren, stimmen diese Markierungen nur in 80% der Fälle überein. Zudem ist die Klassifizierung in positive und negative Emotionen eine starke Vereinfachung der Realität. Aktuelle Forschungsarbeiten weiten die Sentimentanalyse auf tatsächliche Emotionen aus[3] (Wütend, Traurig, Fröhlich...), um Verbesserungen zu erzielen.

[3] Ali Yadollahi, Ameneh Gholipour Shahraki, and Osmar R. Zaiane. 2017. Current State of Text Sentiment Analysis from Opinion to Emotion Mining. ACM Comput. Surv. 50, 2, Article 25 (May 2017), 33 pages. DOI: https://doi.org/10.1145/3057270



MIT DER SENTIMENTANALYSE KANN EIN BOT DIE STIMMUNG DES NUTZERS ERKENNEN UND VERÄRGERTE NUTZER DIREKT ZU EINEM AGENTEN WEITERLEITEN.

IV. Dialog Management:
MEIN COMPUTER WEISS WAS ICH VOR 5 MINUTEN GESAGT HABE

Wenn die Language Understanding Schicht das WAS und eventuell das WIE aus der Aussage des Nutzers extrahiert hat, muss der Bot darauf reagieren. Er muss also entscheiden, was als nächstes zu tun ist.

Im vorherigen Beispiel der Pizzabestellung könnte der Bot, nachdem er erkannt hat, dass der Nutzer Pizza bestellen möchte, auf ein CRM zugreifen, um auf die Adresse des Nutzers zuzugreifen, um dann in einem ERP-System eine Bestellung auszulösen. Alternativ könnte der Bot versuchen weitere Produkte wie Wein oder Nachtisch zu verkaufen. Welche Aktion der Bot ausführt hängt vom Stand der Konversation sowie, dem Use Case des Bots ab.

INFORMATION RETRIEVAL BOT

Die einfachste Form des Dialogmanagements ist der Information Retrieval Bot, der immer nur die letzte Nachricht des Nutzers betrachtet. Die Schwäche dieser Form des Dialogmanagements wird im folgenden Beispiel klar:

Nutzer: Welche Pizzasorten gibt es?
Bot: Es gibt Pizza Salami, Hawai und Tonno.
Nutzer Was ist auf der ersten alles drauf?
Bot: Entschuldige, ich verstehe dich nicht.

Hätte der Nutzer gefragt: „Was ist auf der Pizza Salami?" Hätte der Bot eine sinnvolle Antwort geben können. Da er aber zur Beantwortung der zweiten Frage den Kontext aus seiner ersten Antwort brauchte, konnte der Bot die Anfrage nicht verarbeiten. Diese Art des Dialogmanagements ist einfach zu implementieren und funktioniert insbesondere gut zur Beantwortung von Standartanfragen.

STATEMACHINE

Das Dialogmanagement mittels Statemachine ist ein Ansatz mit variabler Komplexität. Zusammenfassend kann das Prinzip folgendermaßen beschrieben werden: Der Chatbot befindet sich zu jedem Zeitpunkt in einem definierten Zustand. Dieser Zustand kann z.B. sein auf_eingabe_warten, pizza_sorte_abfragen oder bestellung_bestätigen. Es gibt klare Regeln wie der Bot in die jeweiligen Zustände gelangt, was er tun muss, wenn er sich in einem Zustand befindet und welche Folgezustände möglich sind.

Die einfachste Form der Statemachine ist die Baumstruktur. Hierbei wird der komplette Dialogverlauf vordefiniert. Das bedeutet, dass die Konversation immer entlang des so genannten „Happy Path" geführt werden muss. Wenn der Nutzer anders handelt als von den Botentwicklern antizipiert, kann der Bot nicht mehr sinnvoll reagieren. Da Nutzer häufig anders reagieren als angenommen, muss der Bot den Dialog stark führen, um dem Nutzer möglichst wenige Möglichkeiten zum Verlassen des „Happy Path" zu geben.

Um eine intelligentere Form der Statemachine handelt es sich, wenn sich der Bot alle Informationen rund um den aktuellen Dialog merkt. Das heißt er speichert zum einen Informationen, welche der Nutzer bereits gegeben hat und die aktuelle Situation des Dialogs..

Nutzer: „Ich möchte eine Pizza zur Berlinerstr. 134 zu 20:30 Uhr bestellen."
Bot: „Sehr gerne, welche Pizza möchten Sie bestellen?"
Nutzer: „Welche Pizzen sind vegetarisch?"
Bot: „Pizza Broccoli, Pizza Vegetala und Pizza Spinachi."
Bot: „Welche Pizza möchten Sie bestellen?"
Nutzer: „Was ist auf der Pizza Vegetala?"
Bot: „Broccoli, Artischocken, Zwiebeln, Spargel, Paprika, Pilze, Tomatensauce und Käse."
Bot: „Welche Pizza möchten Sie bestellen?"
Nutzer: „Eine Pizza Salami bitte."
Bot: „Ok, Ihre Pizza Salami wird um 20:30 zur Berlinerstr. 134 geliefert. Kann ich sonst noch etwas für Sie tun?"

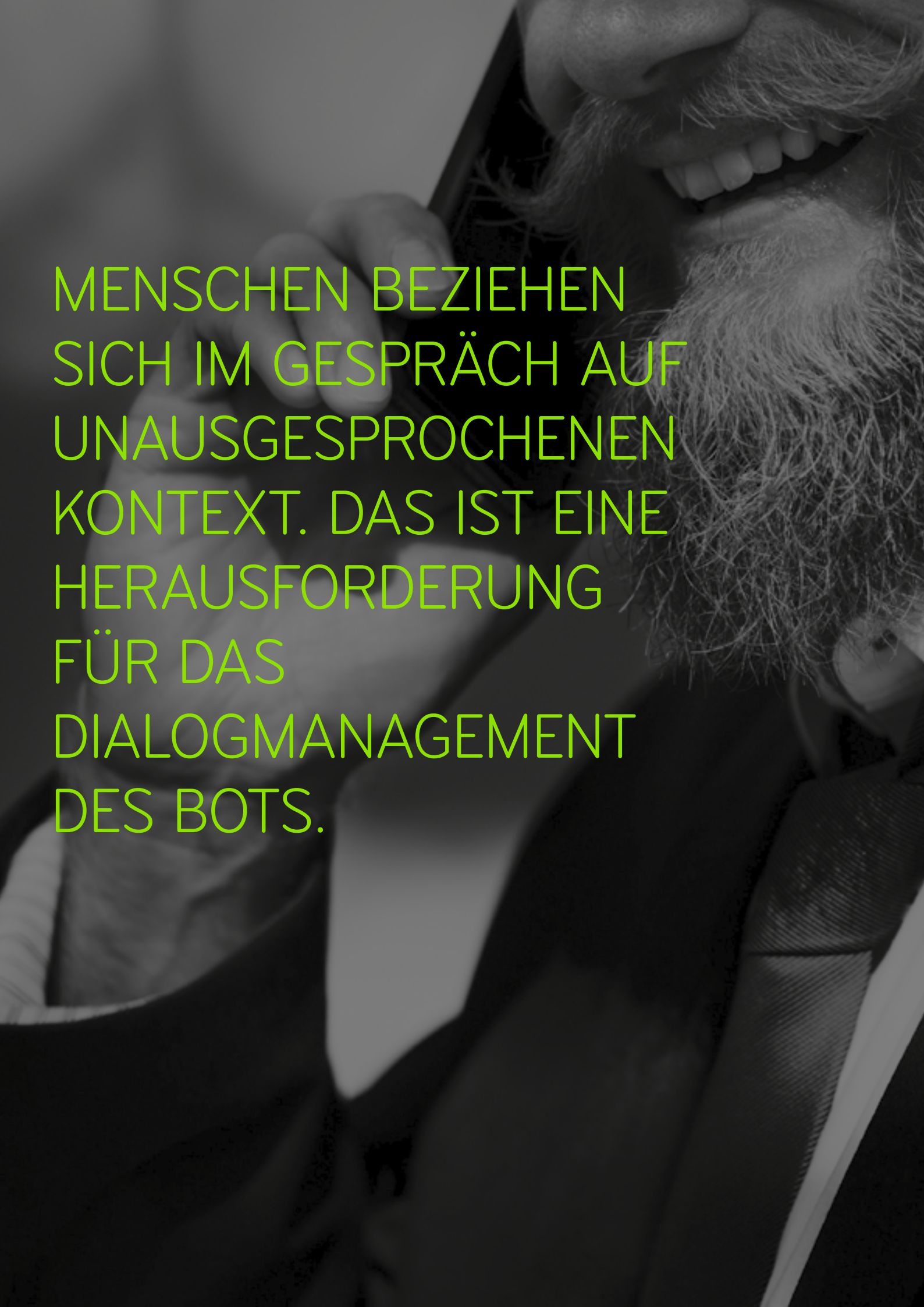
In diesem Dialog hat sich der Bot die Informationen über die Uhrzeit und die Adresse während des gesamten Dialogs gemerkt. Außerdem ist er aus dem Zustand pizza_bestellen in den Zustand pizza_auskunft gewechselt ohne den ursprünglichen Zustand zu vergessen. Hierfür wurden die einzelnen Zustände "gestapelt" und dann nacheinander abgearbeitet. Diese Form des Dialogmanagements ist bedeutend flexibler als die einfache Baumstruktur, da dem Nutzer nicht ein „Happy Path" vorgegeben wird, sondern man ihm die Möglichkeit bietet, auch Zwischenfragen zu stellen. Gleichzeitig kann dies im Vergleich zum Mensch-Mensch Dialog zu unnatürlichen Gesprächsverläufen führen. Der Nutzer möchte nicht unbedingt zu jedem Zustand des Chatbots zurückkehren, welchen er initiiert hat. Insbesondere wenn die Konversation viele „Kurven" nimmt, wirkt diese Form des Dialogmanagements schnell unnatürlich.

MACHINE LEARNING

Wird im Dialogmanagement mit Machine Learning[4] gearbeitet, werden die Zustände nicht mehr benötigt. Das Dialogmanagement speichert zu jedem Zeitpunkt die bisherige Konversation, eventuelle Daten zum Nutzer und sonstige relevante Daten, um diese zusammen mit der aktuellen Eingabe des Nutzers in einen Machine-Learning Algorithmus zu geben, um die nächste Aktion zu bestimmen. In diesem Kontext ist der Begriff „Aktion" sehr weit gefasst. Aktionen können zum Beispiel sein, Daten aus einer Datenbank abzurufen, dem Nutzer zu antworten oder auch auf eine Eingabe des Nutzers zu warten. Nach dem Abschluss jeder Aktion werden die aktuellen Daten dem Machine Learning Algorithmus zur Verfügung gestellt, um wieder die nächste Aktion zu bestimmen. Das Dialogmanagement befindet sich also kontinuierlich im Wechsel vom Bestimmen der nächsten Aktion und dem Durchführen der aktuellen Aktion. In diesem Zusammenhang ist es wichtig zu betonen, dass auch das Warten auf eine Nutzereingabe eine Aktion darstellt, welche das Dialogmanagement durchführt. Entsprechend würde eine Pizzabestellung zum Beispiel folgende Aktionen benötigen:

pizzasorte_abfragen → eingabe_abwarten → adresse_abfragen → eingabe_abwarten → bestellung_bestätigen

[4] Mehr Informationen über Machine Learning befinden sich in der Infobox auf Seite 8.



MENSCHEN BEZIEHEN SICH IM GESPRÄCH AUF UNAUSGESPROCHENEN KONTEXT. DAS IST EINE HERAUSFORDERUNG FÜR DAS DIALOGMANAGEMENT DES BOTS.

V. Language generation:
ICH VERSTEHE WAS MEIN COMPUTER SCHREIBT

Das Dialogmanagement gibt der Language Generation Schicht den Auftrag, dem Nutzer zu antworten. Auf Basis von Informationen über den Inhalt der Antwort wird in dieser Schicht eine lesbare Antwort formuliert.

Am Pizzabeispiel: Das Dialogmanagement übergibt der Language Generation Schicht die Informationen "nachricht: bestellung_erfolgreich; artikel: pizza_tonno; zeit:45" aus diesem Datensatz formuliert die Language Generation Schicht die Nachricht: "Ich habe eine gute Nachricht für Sie: Ihre Pizza Tonno ist auf dem Weg zu Ihnen! Der Bote braucht 45 Minuten bis er bei Ihnen ist. Kann ich sonst noch etwas für dich tun?" Wie Language Understanding häufig "Natural Language Understanding" genannt wird, wird auch Language Generation häufig "Natural Language Generation (NLG)" genannt. Auch hier wird das „Natural" häufig mit der Verwendung von Machine Learning Algorithmen assoziiert. Language Generation mittels Machine Learning ist aktuell allerdings hauptsächlich im Forschungsumfeld zu finden[5]. In der Praxis ist Language Generation mittels Templates der übliche Fall, bei dem der Bot für jede mögliche Antwort einen vorformulierten Satz verwendet, der potentiell mit Variablen individualisiert wird. Für das Pizzabeispiel würde das Template folgendermaßen aussehen:

"Ich habe eine gute Nachricht für Sie: Deine <pizza_sorte> ist auf dem Weg zu Ihnen! Der Bote ist in <lieferzeit> bei Ihnen. Kann ich sonst noch etwas für Sie tun?"

pizza_sorte und lieferzeit sind die Variablen, welche der Language Generation Schicht vom Dialogmanagement übergeben wurden. Um etwas Abwechslung in die Gesprächsverläufe zu bringen, werden häufig für eine Aussage mehrere Templates vorgehalten, welche dann zufällig gewählt werden. Ein Bot könnte zum Beispiel für begrüßung_vormittag folgende Sätze vorhalten "Guten Morgen, Pizza zum Frühstück?", "Guten Morgen, haben Sie auch von Pizza geträumt?", "Guten Morgen, möchten sie Pizza für heute Mittag bestellen?" und diese zufällig abwechselnd nutzen.

VI. Sprachsynthese:
ICH VERSTEHE WAS MEIN COMPUTER SAGT

Wenn der Nutzer mit dem Chatbot spricht, muss das, was die Language Generation Schicht formuliert hat noch ausgesprochen werden. Hierfür kommt die Sprachsynthese zum Einsatz, für die es aktuell zwei praktikable Ansätze gibt.

CONCATATIVE SYNTHESIS

Hier werden einzelne Laute (Phoneme / Silben / Wörter / Sätze) in einer Datenbank gespeichert und bei Bedarf aneinandergehängt. Die jeweiligen Laute werden mehrfach mit unterschiedlichen Eigenschaften gespeichert, also in unterschiedlichen Geschwindigkeiten, mit unterschiedlichen Grundfrequenzen, unterschiedlichen Melodien, etc. Diese Eigenschaften der jeweiligen gespeicherten Tonaufnamen werden zusammen mit den Aufnahmen gespeichert. Wenn nun ein Satz erstellt wird, muss

[5] https://arxiv.org/pdf/1506.05869.pdf

die Sprachsynthese entscheiden, welche Form des jeweiligen Lauts es nutzt um eine möglichst natürliche Sprachausgabe zu erreichen.

VOLL SYNTHETISCH

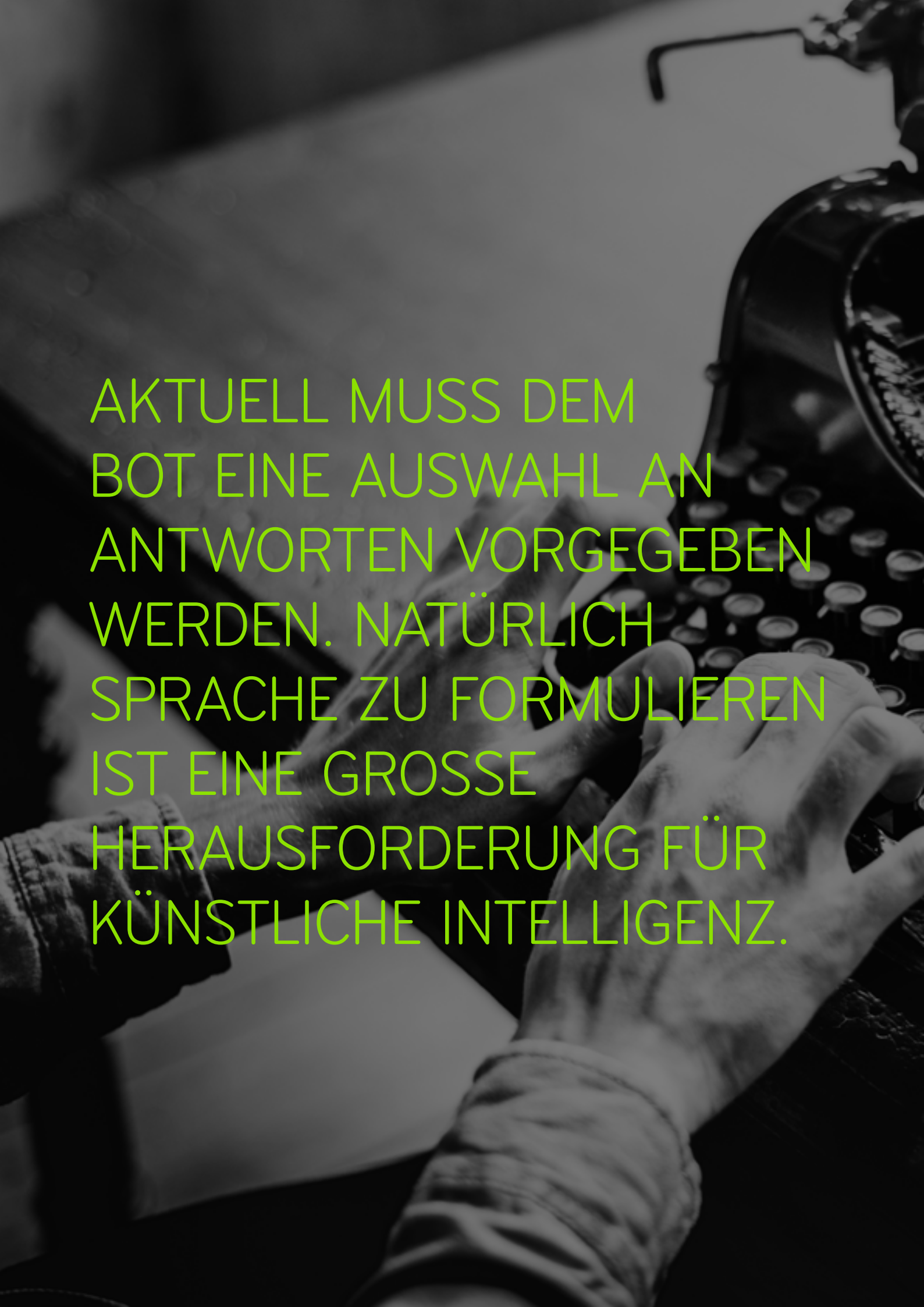
Wenn Sprache voll synthetisch generiert wird, basiert dies immer auf mathematischen Modellen, welche entweder Sprache als solche mathematisch beschreiben oder den menschlichen Vokaltrakt mathematisch nachbilden.

Der Vorteil von voll synthetisch generierter Sprache ist eine bedeutend höhere Flexibilität. Wenn ein System auf Concatative Synthesis basiert und der Bot eine andere Stimme bekommen soll (zum Beispiel männlich statt weiblich) müssen alle aufgenommenen Laute neu eingesprochen werden. Bei voll synthetisch generierter Sprache muss die Sprachsynthese nur neu trainiert werden, was bedeutend schneller möglich ist, da nicht sämtliche Wörter neu eingespochen werden müssen, sondern der Sprachsynthese ein Sample an Tonaufnahmen genügt.

Aktuell sind noch sehr wenige Voicebots mit einer voll synthetischen Sprachsynthese ausgestattet, da diese Methode erst vor kurzem (~2016) auf ein gut nutzbares Niveau gebracht wurde[6]. Insbesondere Entwicklungen im Bereich Deep Learning waren hier ausschlaggebend. Es ist abzusehen, dass voll synthetische Sprachsynthese auch beliebige Stimmen imitieren können wird. So könnte ein Bot mit der Stimme von Donald Trump oder Barack Obama antworten[7].

[6] https://deepmind.com/blog/wavenet-generative-model-raw-audio/

[7] https://www.scientificamerican.com/article/new-ai-tech-can-mimic-any-voice/

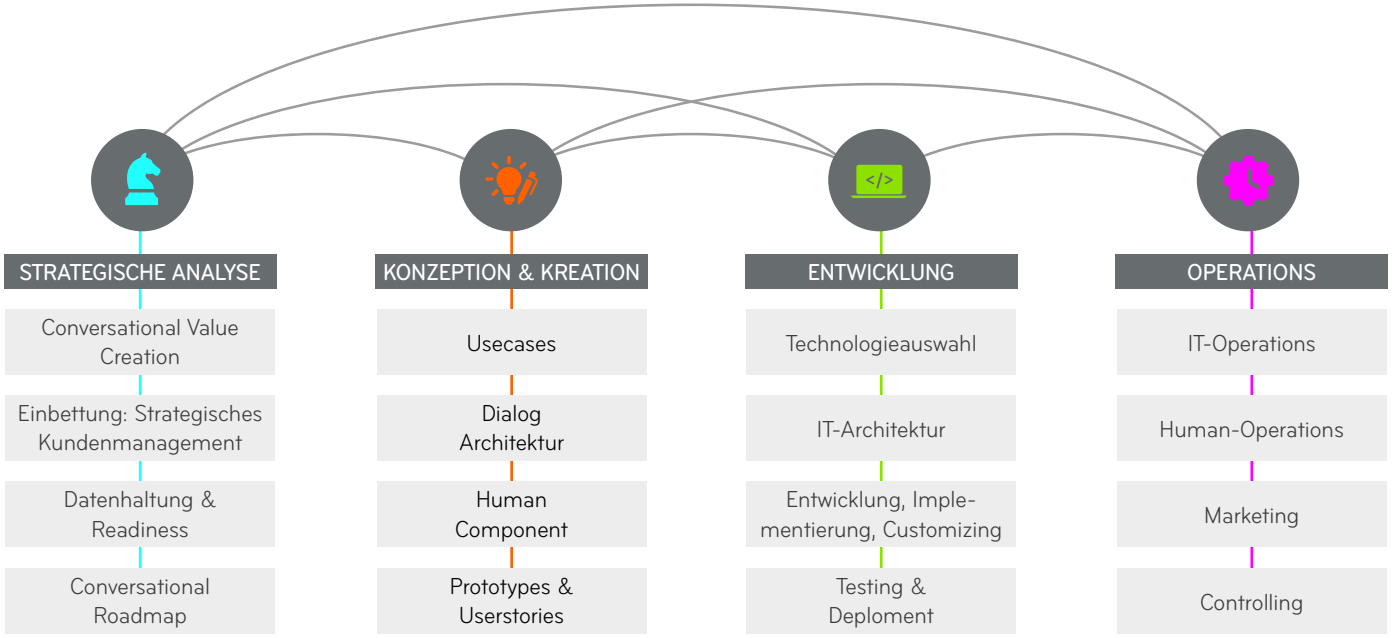


AKTUELL MUSS DEM BOT EINE AUSWAHL AN ANTWORTEN VORGEGEBEN WERDEN. NATÜRLICH SPRACHE ZU FORMULIEREN IST EINE GROSSE HERAUSFORDERUNG FÜR KÜNSTLICHE INTELLIGENZ.

VII. Fazit

Vor der eigentlichen Entwicklung eines Conversational-Projects bzw. eines Bots, sollte der jeweilige Usecase mit einem hohen Detailgrad definiert werden. Die Bandbreite an denkbaren Usecases ist ebenso groß, wie die Bandbreite an zur Verfügung stehenden Technologien. Hinzu kommen Rahmenbedingungen, wie der gewünschte Grad an Accessibility und Skalierbarkeit. Besonders wichtig ist auch die sogenannte „Human Component", also die Rolle, die Menschen bei der automatisierten Bot-Dialogführung einnehmen müssen.

Um in diesem Dschungel eine Technologieanbieter unabhängige Orientierung zu geben, haben wir das **Conversational Project Framework** entwickelt, um Projekte von der Idee, über die Konzeption und Entwicklung bis hin zum Betrieb zu entwickeln.



MUUUH! ist zugleich das erfahrenste und innovativste Start-up im Kundenmanagement. Hervorgegangen aus Europas größtem inhabergeführten Kundenmanagement-Dienstleister buw, kombinieren wir 25 Jahre Erfahrung in der Kundengewinnung, Kundenbindung und Kundenpotenzialausschöpfung mit einer radikal neuen Sichtweise auf die Herausforderungen modernen Kundenmanagements.

Wir handeln mutig, unbequem und herausragend, um den Spaß am gemeinsamen Erfolg zu erreichen. Kunden finden in uns einen Partner, der Veränderung als Chance sieht, komfortable Standpunkte hinterfragt und nur die beste aller möglichen Lösungen akzeptiert.

Diskutieren Sie jetzt mit uns, wie Sie ihr Kundenmanagement mit Conversational Customer Management evolutionieren können.

Autoren:
Marcus Hülsdau
Ben Ellermann
Elena Morawin

MUUUH! GmbH
Heger-Tor-Wall 19
49078 Osnabrück
Deutschland

Mobil +49 (0) 170 3736 - 770
E-Mail bots@muuuh.de
Web www.muuuh.de

MUUUH! GROUP