



NürnBOT – smarte Unterstützung für den Einwohnerservice

Künstliche Intelligenz unterstützt bei der telefonischen Terminvereinbarung

November 2021

Whitepaper

Autoren

**Juliana Haas, Thomas Bahlinger,
Roland Zimmermann, John Loutzenhiser**

Kontakt

thomas.bahlinger@th-nuernberg.de
roland.zimmermann@th-nuernberg.de

Chat- und Voicebots bieten vielfältige Potenziale: Sie ermöglichen die Entlastung von Mitarbeitenden in Servicebereichen durch die Übernahme von abgegrenzten Standard-Kommunikationsaufgaben mit anderen natürlichen Personen wie z.B. Bürgerinnen und Bürgern oder anderen Mitarbeitenden. Damit können die entlasteten Fachleute ihren Fokus auf inhaltlich schwierigere Fragestellungen legen. Darüber hinaus sind virtuelle Assistenten im Gegensatz zu menschlichen Mitarbeitenden zu jeder Tages- und Nachtzeit erreichbar und können auch mehrere Gespräche gleichzeitig führen. Organisationen, die Chat- oder Voicebots einsetzen, profitieren zudem von

dem modernen Image eines solchen Systems, können digital-affine Zielgruppen erreichen und bauen gleichzeitig auch noch wertvolle Erfahrung in den Bereichen künstlicher Intelligenz und maschinellen Lernens auf. Es lohnt sich also, einen Blick hinter die Kulissen zu werfen, wie diese smarten „Service-Mitarbeiter“ realisiert werden.

Voicebot für
Nürnberg's
Einwohner-
meldeamt

Die Stadt Nürnberg muss nicht zuletzt aufgrund der Corona-Krise immer mehr Terminvereinbarungen mit Bürgerinnen und Bürgern treffen, damit unnötige Wartezeiten und -schlangen entfallen. Hier setzt ein Kooperationsprojekt mit der Technischen Hochschule Nürnberg Georg Simon Ohm und der MSG Systems AG an: Ein Prototyp für den NürnBOT „Sigena“ wurde als Voicebot für die Terminvereinbarung des Einwohnermeldeamtes realisiert. Dieser virtuelle Assistent kann perspektivisch telefonische Terminvereinbarungen übernehmen und so die Erreichbarkeit der städtischen Einrichtung für ihre Bürgerinnen und Bürger durch eine Erreichbarkeit rund um die Uhr, auch für nicht-digital-affine Personen,

Agiles Kooperations- projekt

verbessern. Gleichzeitig erscheint eine Entlastung der knappen Personalkapazitäten realistisch.

Dieses Projekt, das eine neue Technologie erlebbar machen soll, lebt von enger Kooperation und einer agilen Vorgehensweise. Jeder Projektteilnehmer bringt ganz spezifische Fähigkeiten ein: Das Wissen über die täglichen Terminvereinbarungen durch die Stadt Nürnberg, das fachliche Know-how des Labors für Künstliche Intelligenz in der Betriebswirtschaft der TH Nürnberg, sowie die technische Expertise im Bereich von Voice- und Chatbots im öffentlichen Sektor durch die MSG Systems AG. Gemeinsam sind Antworten auf Fragen zu finden wie etwa „Was ist der Status Quo im Tagesgeschäft?“, „Wer sind die Nutzer und wie kommunizieren diese?“ oder „Wie läuft ein optimales Gespräch mit dem Voicebot ab?“. Diese Fragen werden in mehreren Workshops geklärt, bei denen das Wissen aller Teilnehmer zusammengetragen wird: Interviews mit den Mitarbeitenden im Kundenkontakt helfen, die Nutzergruppen und deren Kommunikationsweise zu verstehen, bedeutende Anwendungsfälle werden analysiert und die Bot-Persönlichkeit wird gemeinsam ausgestaltet.

Das vorliegende Whitepaper gibt einen Hintergrund-Einblick in die Funktionsweise von Voice- und Chatbots und skizziert die wichtigsten Projektschritte, die auf dem Weg zum *NürnBOT Sigena* realisiert wurden.

Wie Voicebots funktionieren

NLP und NLU

Ziel eines virtuellen Assistenten ist es, einen natürlichsprachlichen Dialog anzubieten, der ähnlich einem Dialog mit einem menschlichen Gesprächspartner abläuft. Das bedeutet, dass sich der Voicebot von bekannten Telefonleitsystemen oder klickbaren Chatbots abheben soll. Statt aus einem vordefinierten Menü die richtige Option zu wählen – etwa über Aussprechen des Wortes „eins“ – kann der Voicebot die natürliche Sprache seines Gegenübers verarbeiten. Dies geschieht durch das sog. *Natural Language Processing (NLP)* und *Natural Language Understanding (NLU)*. NLP ermöglicht es einem System, mit natürlicher Sprache umzugehen. Es umfasst dabei mehrere aufeinanderfolgende Verarbeitungsschritte: So müssen die gesprochenen Worte, also die Audiosignale, zunächst in Texte überführt werden. Diese Texte werden anschließend z.B. in Wörter zerlegt, die evtl. auf ihre Stammformen reduziert und im Hinblick auf ihre grammatikalische Bedeutung analysiert werden. Erst mit NLU als einem Teilbereich von NLP wird dann ein „Verstehen“ von Sprache erreicht. So besteht eine Aufgabe von NLU darin, von den erkannten Wörtern auf die zugrundeliegende Absicht des Nutzers zu schließen. Dafür muss der Voicebot allerdings mit Varianten ähnlicher Aussagen trainiert werden, um zukünftig auch bislang unbekannte Äußerungen von Benutzern in den richtigen inhaltlichen Kontext zu setzen.

Intent Classification

Dem Voicebot muss beigebracht werden, dass eine bestimmte Kombination von Wörtern eine bestimmte Absicht (einen sog. *Intent*) impliziert. Ein Nutzer kann die selbe Intention aber auf unterschiedlichste Weise kommunizieren. „*Ich brauche einen Termin für ein Führungszeugnis*“ und „*Ich fahre bald in den Urlaub und mein Personalausweis läuft ab*“ liegt der gleiche Gedanke zu Grunde: In beiden Fällen muss das System erkennen, dass der Nutzer einen Termin benötigt. Daher müssen dem Voicebot einige beispielhafte Sätze beigebracht werden, damit er mit der Diversität von Sprache umgehen kann. Wurde das System mit entsprechenden Daten trainiert, kann es auch neue Eingaben erkennen, die von den Trainingsdaten abweichen und diese mit einer gewissen Wahrscheinlichkeit einem Intent zuordnen.

„Ich brauche einen Termin für ein Führungszeugnis.“

Intent: Terminbuchung

Entities: Leistung = Führungszeugnis

„Ich fahre bald in den Urlaub und mein Personalausweis läuft ab.“

Intent: Terminbuchung

Entities: Leistung = Personalausweis

Entity Extraction

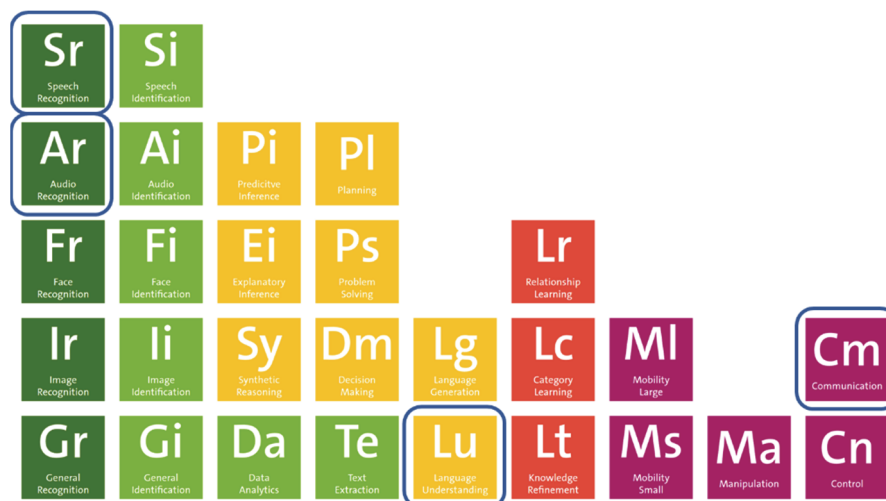
Neben der Nutzerabsicht werden im Zuge von NLU gezielt Informationen zu Objekten (sogenannten *Entities*) aus den Nutzereingaben extrahiert. In den Beispielsätzen fällt auf, dass sich der Grund für die Terminbuchung unterscheidet. Während zunächst von einem Führungszeugnis die Rede ist, benötigt eine andere Person einen neuen Personalausweis. Diese Information gilt es, in einem Dialog zu extrahieren und zu speichern, um im weiteren Verlauf darauf Bezug nehmen zu können.

Speech-to-Text Text-to-Speech

Intent Classification und *Entity Extraction* als Teile von NLU bilden den wichtigsten Aspekt in der Sprachverarbeitung eines virtuellen Assistenten. Beide benötigen jedoch als Grundlage geschriebenen Text, so dass für eine Anbindung an eine Telefonanlage – wie es auch beim NürnBOT der Fall ist – ein weiterer Schritt vor- sowie nachgestellt werden muss. Spricht ein Mensch mit dem System, so gilt es, diese gesprochene Sprache in geschriebene Sprache zu transformieren (*Speech to Text [S2T]*). Liegt dagegen eine schriftliche Antwort des Chatbots vor, muss diese wiederum in gesprochene Sprache umgewandelt werden (*Text to Speech [T2S]*). Die zusätzlichen Funktionen *Spracheingabe* und *Sprachausgabe* eines sprechenden Assistenten stellen also eine Erweiterung eines Chatbotsystems hin zu einem Voicebot dar. (Rasa 2021a, Ellermann 2021)

Periodensystem der KI – Baukasten für Innovation

Das *Periodensystem der KI*¹ (Bitkom 2018) stellt eine Übersicht zentraler KI-Bausteine bereit, die es nicht-IT-Experten ermöglicht, die Anforderungen an innovative Softwarelösungen zu konkretisieren. Auf dieser Basis ist ein greifbarer Dialog möglich, um Design und Pilotierung von KI-Lösungen zu beschleunigen. Beim Bau eines Chat- bzw. Voicebots wie dem NürnBOT-Projekt werden insbesondere die Elemente *Speech Recognition (Sr)*, *Audio Recognition (Ar)*, *Language Understanding (Lu)* und *Communication (Cm)* benötigt.



Anforderungen an einen Bot sammeln

Problemanalyse

Die Analyse des Status quo beginnt damit, die Experten des Tagesgeschäfts zu befragen und deren Erfahrungen einzuholen. So wurden von der TH Nürnberg Interviews mit den Mitarbeitenden der Stadt Nürnberg organisiert, um wichtige Anforderungen an den Chat- bzw. Voicebot identifizieren zu können. Dies beinhaltet neben der Betrachtung des gegenwärtigen Prozesses auch technische Rahmenbedingungen. Folgende bedeutende Punkte inhaltlicher Natur konnten gesammelt werden, die im weiteren Projektablauf berücksichtigen werden müssen:

- Es gibt unterschiedliche Kundengruppen.
- Es existiert eine große Anzahl verschiedener Kundenanliegen.
- Komplexität und Dauer der Telefongespräche können je nach Kundengruppe stark variieren.
- Für manche Termine besteht eine hohe Dringlichkeit.
- Personen sollten bereits bei der Terminreservierung anhand ihres korrekt erfassten Namens und Geburtsdatums identifiziert werden.

¹ <https://periodensystem-ki.de>

Je nach Art des Anliegens ist die Termindauer verschieden und das System soll passende, freie Termine bereitstellen. Durch das Setzen von Filtern kann Einfluss auf die Terminvorschläge genommen werden. Diese Filter können etwa „nur nachmittags“ oder „nur vormittags“ sein oder sich auf einen bestimmten Wochentag beschränken. Auch der virtuelle Assistent soll auf derart unscharfe Wünsche der Nutzer eingehen und Termine nach Suchkriterien ausgeben können.

Daneben wurden auch die Phasen eines typischen Gesprächs analysiert und skizziert. Außerdem bedarf es für das spätere Training des Systems einer Sammlung typischer Formulierungen, die Anrufende regelmäßig bei der Terminvergabe benutzen und exemplarische Folgefragen, die während eines Gesprächs aufkommen. Auch dabei ist die Expertise der Mitarbeitenden der Fachabteilungen gefragt.

Systemumgebung

Auf technischer Ebene wird die Funktionalität bestehender Systeme in Augenschein genommen, in die der Voicebot später eingebunden werden soll. Im Fall des NürnBOTs geht es hier unter anderem um das Terminbuchungssystem² sowie die zentrale Telefonanlage.

Die Persönlichkeit des Bots definieren

Definition der Use Cases

In Workshops wurden aus den gesammelten Kundenanliegen die gefragtesten ausgewählt, auf die sich der Chat- bzw. Voicebot zunächst beschränken soll. Dazu gehörte in Nürnberg etwa das Beantragen eines neuen Ausweises und eines Führungszeugnisses.

Definition der Nutzerpersonas

In einem zweiten Schritt wurde auf die zuvor erwähnten unterschiedlichen Kundengruppen genauer eingegangen. Diese zeichnen sich durch unterschiedliche Eigenschaften und Bedürfnisse aus. Während eine Kundin womöglich sehr gestresst ist und möglichst schnell und ohne Termin beim Bürgeramt vorbeischauchen möchte, ist ein älterer Kunde zeitlich eher flexibel, jedoch unsicher im Umgang mit der Behörde. Um diesen verschiedenen Anforderungen gerecht zu werden, können Personas entworfen werden, die den Hintergrund

Sabine (48)

- Benötigt Führungszeugnis
- Hat es sehr eilig
- Genervt, weil sie den Ämtergang nicht sofort persönlich vornehmen kann

„Kann ich da nicht einfach vorbeikommen?“

² Das Terminbuchungssystem kann auch online über einen Browser genutzt werden: <https://nuernberg.termine-reservieren.de>

und die Ausdrucksweise der häufigsten Kundentypen aufgreifen. Dazu gehörte beim NürnBOT zum Beispiel Sabine (48).

Definition der Chatbotpersona

Zusätzlich zu den Personas auf der Kundenseite gilt es entsprechend des Images eines Unternehmens und dessen Leitwerten eine Persona für den Chat- bzw. Voicebot zu kreieren. Dazu kann etwa das Geschlecht, Charaktereigenschaften, der Name und die Tonalität bestimmt werden. Der Voicebot für die Stadt Nürnberg heißt z.B. Sigena und stellt damit einen Bezug zur Geschichte Nürnbergs her. (Kaiser et al. 2019)

Happy Path

Unter Berücksichtigung der Nutzerpersonas und der Chatbotpersönlichkeit können dann beispielhafte Dialogabläufe für jeden Use Case skizziert werden. Dabei sollten sich die Projektteilnehmenden zunächst auf sog. *Happy Paths* konzentrieren. Happy Paths sind solche Dialogabläufe, die einem optimalen Schema folgen und in denen keine unvorhergesehenen Aktionen vorkommen. Ausgehend von diesen Happy Paths können im späteren Projektverlauf sog. *Unhappy Paths* ergänzt werden, bei denen der Dialog an einer Stelle von diesem optimalen Schema abweicht.



Auf Basis der so gestalteten Happy Paths kann dann eine Sammlung benötigter Intents – also der verschiedenen Absichten der Nutzer – und der in den Intents enthaltenen Entities – also der wichtigen Informationen – erstellt werden. Auf der Seite des Chatbots bedarf es dagegen einer Liste mit passenden Antworten. Diese Sammlung an Gesprächsbausteinen bildet gemeinsam mit den definierten Happy Paths nun den Baukasten für die technische Entwicklung des Chat- bzw. Voicebots.

Den Bot realisieren mit Intents, Responses/Actions und Stories

Training der Intents

Für die technische Umsetzung des Nürnbots hat sich das Projektteam für die Open Source Plattform *Rasa* entschieden.

In einem ersten Schritt wird das System mit den definierten Intents trainiert. Dazu werden die in den Interviews gesammelten typischen Formulierungen der entsprechenden Absicht zugeordnet. Dabei ist es notwendig, darin enthaltene Entitäten zu kennzeichnen, so dass der Chatbot diese später auch in den Nutzereingaben erkennen kann. Wie bereits erwähnt, kann das System nach dem Training auch neue Aussagen korrekt einem Intent zuordnen, sofern diese eine gewisse Ähnlichkeit mit den hinterlegten Trainingsdaten aufweisen. Daher ist es essenziell, Beispieldaten zu wählen, die einen breiten Wortschatz nutzen.

intent: terminbuchung

examples: |

- Ich brauche einen Termin für ein [Führungszeugnis]{"entity": "Leistung"}
- Ich fahre bald in den Urlaub und mein [Personalausweis]{"entity": "Leistung"} läuft ab
- Brauche neuen [Perso]{"entity": "Leistung"}
- Ich benötige einen Termin für ein [Führungszeugnis] {"entity": "Leistung"}

Gestaltung der Responses und Custom Actions

Für die Rückmeldungen des Chatbots werden sog. *Responses* angelegt. Dabei können für eine Response auch mehrere Antwortmöglichkeiten angegeben werden, aus denen im Dialog zufällig eine ausgewählt wird. So hinterlässt der virtuelle Assistent bei mehrmaliger Nutzung einen menschlicheren Eindruck. Bei Responses handelt es sich um statische Antworten, die nicht mit weiteren Informationen angereichert werden. Im Gegensatz dazu können *Custom Actions* eingesetzt werden, um Antworten dynamisch zu gestalten. Dies ist etwa bei den Terminvorschlägen erforderlich, wenn das System auf Basis der Angaben des Benutzers einen passenden Termin aus einer Datenbank abfragen muss.

responses:

utter_vorstellung:

- text: Hi, ich bin Sigena, Ihr virtueller Assistent

utter_wie_kann_ich_helfen:

- text: Was kann ich für Sie tun?
- text: Wie kann ich Ihnen weiterhelfen?
- text: Nennen Sie mir Ihr Anliegen, dann kann ich Ihnen weiterhelfen!

Gestaltung von Stories

Damit auf einen erkannten Intent auch eine korrekte Antwort folgt, werden sog. *Stories* erstellt. Stories bilden den Dialogablauf Schritt für Schritt nach, so dass immer eine Response oder Custom Action auf einen Intent folgt. Für mögliche Wendungen im Gespräch wird jeweils eine entsprechende Story erstellt, damit der Chatbot weiß, wie er in welchem Fall reagieren muss.

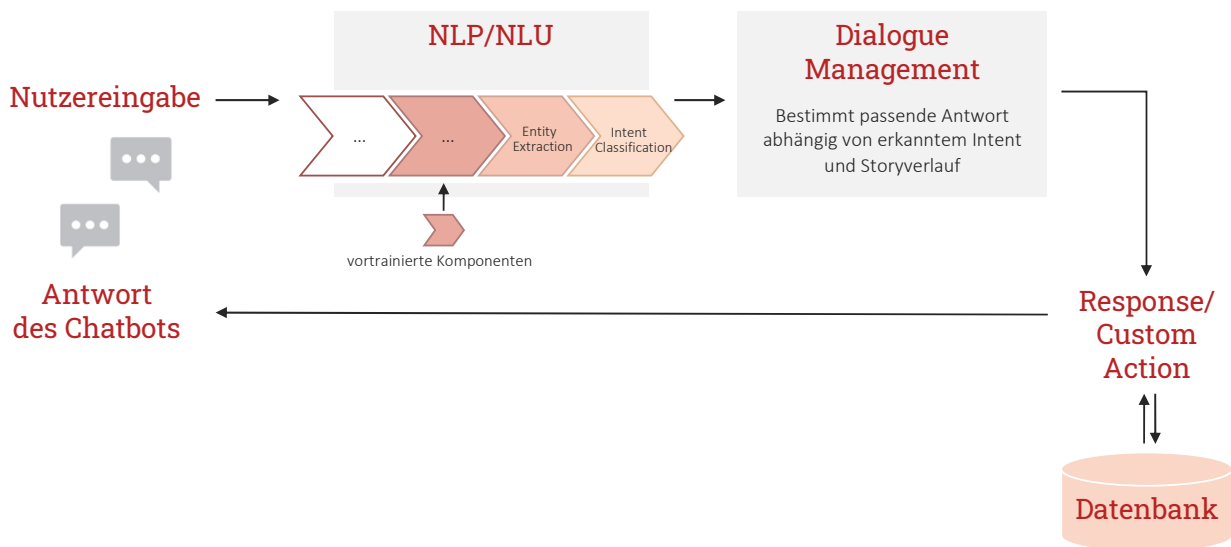
story: begruessung

steps:

- intent: begruessung
- action: utter_vorstellung
- action: utter_wie_kann_ich_helfen
- intent: terminbuchung
- ...

Sollte ein Nutzer Eingaben tätigen, die zu keiner hinterlegten Story passen, wird der Chat- bzw. Voicebot dennoch auf Basis seines trainierten Wissens versuchen, eine passende Antwort auszuwählen und auszugeben. Um aber eine möglichst gute Leistung des Systems zu gewährleisten, ist es von großer Bedeutung, zunächst nicht berücksichtigte aber häufig vorkommende Dialogabläufe ebenfalls als Stories zu speichern und damit den Chatbot robuster zu machen. Dies erfordert, dass das entwickelte System auch während der Betriebsphase (oder einer Pilotphase) anhand echter Nutzerdialoge weiter hinzulernt: Seine Wissensbasis also mit Hilfe zusätzlicher Trainingsdaten erweitert.

Vereinfacht ist die Architektur eines Chatbots ohne Spracheingabe und Sprachausgabe wie folgt darstellbar:



Unschärfe in der Terminfindung mit einem Voicebot

Für einen Voicebot stellt die Terminfindung eine besondere Herausforderung dar: Es muss ein passender Termin zwischen Nutzer und Bot „verhandelt“ werden. Dazu steht lediglich die gesprochene Sprache als „Verhandlungswerkzeug“ zur Verfügung, denn bei einem rein sprachbasierten Chatbot bzw. Voicebot besteht für Benutzer keine Möglichkeit, auf bekannte grafische Oberflächen wie einen Kalender zurückzugreifen. Über eine Kalendendarstellung können Benutzer schnell einen präzisen Überblick über freie Termine erhalten und einen passenden auswählen. Diese Option entfällt bei einem Voicebot: Die Terminfindung muss in einem Gespräch erfolgen, das unscharfe Aussagen enthalten kann und in dem beide Gesprächspartner durch Sprecherwechsel im Dialog Termine vorschlagen, ablehnen, und verfeinern, bis ein passender Termin gefunden wurde. Das stellt für ein Voicebot-System folgende Herausforderungen dar:

Wie wird die Terminfindung eingeleitet?

Wie bzw. durch wen soll die Terminfindung eingeleitet werden? Sollen Initialvorschläge vom Voicebot kommen oder soll lieber der Benutzer einen Erstvorschlag für einen Termin machen? In der Praxis hängt diese Entscheidung unter anderem von der Menge freier Termine im Terminalsystem ab. Sind eher viele freie Termine vorhanden, so ist es wenig sinnvoll, dass der Voicebot immer wieder eine veränderte Auswahl daraus anbietet, bis ein Vorschlag angenommen wird. Die Folge wäre ein eher monotoner, unnatürlicher und damit irritierender Gesprächsverlauf. Andererseits, wenn nur noch wenige Termine im Terminalsystem frei sind, dann ist es weniger sinnvoll, den Benutzer zu fragen, ob er einen Wunschtermin hat. In diesem Fall sollte der Voicebot sofort freie Termine aktiv vorschlagen.

Für den NürnBOT-POC wird die Annahme getroffen, dass eher viele freie Termine vorliegen. Deswegen wird die Terminfindung mit der Frage vom Voicebot an den Benutzer eingeleitet „*Haben Sie einen Wunschtermin?*“. Zukünftig kann eine Entscheidung basierend auf der Menge freier Termine im Terminalsystem ergänzt werden.

Unschärfe Äußerungen zu Zeit und Datum zuverlässig interpretieren?

Benutzeräußerungen zu Zeitangaben müssen zuverlässig interpretiert werden. Äußerungen wie „*Nachmittags*“, „*Mittwoch in zwei Wochen*“ oder „*am ersten September um zehn Uhr*“ sind für Menschen leicht zu interpretieren, und wir wissen sofort, um welches Datum bzw. welche Uhrzeit es sich handelt. Letztlich sind sie aber im Vergleich zu einem exakten Zeitstempel unschärfer und vom Kontext abhängig. Das Voicebot-System muss diese unscharfen Angaben in ein eindeutiges Datums- und Zeitformat umwandeln, damit passende Termine in der Terminiendatenbank gefunden werden können. „*Nachmittags*“ muss z.B. in ein Format wie „*Montag-Freitag, 12-18 Uhr*“ transformiert werden. „*Mittwoch in zwei Wochen*“ soll z.B. als „*15.12.2021*“ interpretiert werden, wenn heute Freitag der 3.12.2021 ist. Diese

Herausforderungen bestehen für Chat- und Voicebots gleichermaßen. Ein Voicebot muss jedoch zusätzlich in der Lage sein, die sprachlichen „Zeit-Wörter“ in konkrete Zeitangaben zu wandeln. So wird „am ersten September um zehn Uhr“ zunächst zu „am 01. September um 10 Uhr“ und weiter zu „11.09.2022,10:00“.

Aushandeln eines passenden Termins?

Wie kann im Dialog die Menge möglicher Termine möglichst rasch eingegrenzt werden, so dass ein passender Termin gefunden wird?

Gibt der Benutzer z.B. an, dass er „nächsten Montag“ gerne einen Termin hätte, der Voicebot aber keinen passenden Termin findet, so kann der Voicebot entweder einen alternativen Vorschlag machen, oder den Benutzer auffordern, selber einen alternativen Wunschtermin anzugeben. Gibt der Benutzer daraufhin z.B. an „dann nächsten Mittwoch nachmittags“, und das System findet für Mittwoch um 17 Uhr eine Option, so fragt der Voicebot zurück: „passt Ihnen nächsten Mittwoch um 17 Uhr?“. Wenn der Benutzer darauf jedoch erwidert: „Nein dann lieber Freitag“, so soll der Voicebot nur noch Termine am darauffolgenden Freitag vorschlagen. Dieses Szenario macht deutlich, dass der Voicebot sehr viel „Gedächtnis“ zu schon artikulierten Wunschterminen und abgelehnten Terminen mitführen muss, um die Menge möglicher Termine im Dialog möglichst rasch und zuverlässig einzugrenzen, damit ein passender Termin mit dem Benutzer erfolgreich verhandelt wird.

Eigennamen zuverlässig erkennen

Eine weitere Herausforderung für den Voicebot ist die richtige Erkennung von Eigennamen. Im Gegensatz zu der beispielhaften Entität *Leistung* mit den Ausprägungen *Führungszeugnis* oder *Ausweis* gibt es eine fast unbegrenzte Menge verschiedener Namen. Diese müssen von Speech-to-Text-(S2T)-Komponenten zuverlässig interpretiert werden. Sehr häufige Namen wie z.B. „Christian Müller“ werden in der Regel von S2T ohne weiteres gut erkannt. Die Erkennung von selteneren oder Namen aus kleineren Kulturkreisen erfordert aber zusätzliches Training der S2T-Komponenten.

Für die NürnBOT-Entwicklung wurden sowohl fertige Komponenten aus dem Open-Source-Bereich als auch Eigenentwicklungen eingesetzt, um den genannten Herausforderungen zu begegnen.

Vom Baby-Bot zu einem respektierten Bot

Ist ein erster Prototyp eines Chat- bzw. Voicebots fertig und eine Benutzeroberfläche zur Interaktion bereitgestellt, sind regelmäßige Testsitzungen mit Personen außerhalb der Entwicklergruppe unbedingt nötig und maßgeblich für den Erfolg des Projekts. Die ersten Testpersonen sind optimalerweise zunächst die Mitarbeitenden der auftraggebenden Organisation, da sie die Kundinnen und Kunden sowie deren Verhalten am besten einschätzen können. Später erfolgen sukzessive Tests mit tatsächlichen Nutzern. Ziel des Testens ist es, iterativ zu evaluieren, ob die entwickelten Dialogabläufe auch in der Realität funktionieren und wo

Verbesserungsbedarf besteht. Dabei werden Intents, Responses, CustomActions und Stories schrittweise angepasst, erweitert und auf Inkonsistenzen geprüft.

Auch im NürnBOT-Projekt wurde eine Reihe von Testsitzungen mit Beteiligten der Stadt Nürnberg organisiert. Das fachliche Feedback floss in die Implementierung direkt ein und zudem wurde auch die Plattform *Rasa X* vorgestellt. Rasa X macht es auch Nichtfachkundigen einfach möglich, kleinere inhaltliche Verbesserungen vorzunehmen, den Chatbot weiter zu trainieren und Einblicke in die Entwicklungsarbeit zu bekommen. Rasa X bietet eine Benutzeroberfläche auf der etwa Responses angepasst, Intents weitertrainiert und von Testern geführte Dialoge bestätigt oder abgelehnt werden können (Rasa 2021b). Rasa X stellt damit ein Produktivitätswerkzeug bereit, um Entwicklerteams zu entlasten und zugleich weiteres fachliches Wissen ohne Umwege in einen Chatbot zu integrieren.

Ausblick

Die grundsätzliche Eignung der aktuell verfügbaren Chatbot-Technologie konnte im Rahmen des beschriebenen Proof-of-Concept nachgewiesen werden. Hinsichtlich der weiteren Entwicklung stellt sich daher für die Stadt Nürnberg die Frage, wie der entwickelte Prototyp zukünftig in ein praktisch einsatzfähiges System überführt werden kann. Aufgrund der kritischen Anforderungen an den Datenschutz insbesondere für Behörden sollte eine Implementierung auf zentralen Rechensystemen der Stadt Nürnberg erfolgen (on premise), weshalb die Projektbeteiligten diesbezüglich einen klaren Vorteil von Rasa gegenüber proprietären Lösungen sehen. Insbesondere zum Terminverwaltungssystem ist zudem eine Schnittstelle zu konzipieren. Darüber hinaus ist die Erweiterung der Test- und Evaluationsschritte notwendig, bevor an einen Echteininsatz gedacht werden kann.

Quellen

Bitkom (2018): Digitalisierung gestalten mit dem Periodensystem der Künstlichen Intelligenz. Ein Navigationssystem für Entscheider. Hg. v. Bitkom Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e. V.

Online verfügbar unter https://www.bitkom.org/sites/default/files/2018-12/181204_LF_Periodensystem_online_0.pdf.

Ellermann, Ben (2021): Chatbots – Operative und strategische Facetten von Text- und Sprachautomatisierung. In: Heinrich Holland (Hg.): Digitales Dialogmarketing. Grundlagen, Strategien, Instrumente. 2. Aufl. Wiesbaden: Springer Gabler, S. 681–699. Online verfügbar unter <https://link.springer.com/>. S.692

Kaiser, Markus; Buttkereit, Aline-Florence; Hagenauer, Johanna (2019): Journalistische Praxis: Chatbots. Automatisierte Kommunikation im Journalismus und in der Public Relation. S. 21/22

Rasa (2021a): <https://rasa.com/blog/nlp-vs-nlu-whats-the-difference/>, 05.10.2021.

Rasa (2021b): <https://rasa.com/docs/rasa-x/>, 05.10.2021

Kontakt

Prof. Dr. Thomas Bahlinger, thomas.bahlinger@th-nuernberg.de

Prof. Dr. Roland Zimmermann, roland.zimmermann@th-nuernberg.de

Important Disclosures

© 2021 Labor für Künstliche Intelligenz in der Betriebswirtschaft. Alle Rechte vorbehalten.

Vorbehalt bei Prospektaussagen und Zukunftsaussagen, Haftung

Soweit wir in den Aussagen unserer Studien sowie in sonstigen Verlautbarungen Tatsachenbehauptungen, Prognosen oder Erwartungen äußern oder unsere Aussagen die Zukunft betreffen, können diese Aussagen mit bekannten und unbekanntem Risiken und Ungewissheiten verbunden sein. Die tatsächlichen Ergebnisse und Entwicklungen können daher wesentlich von den geäußerten Erwartungen und Annahmen abweichen. Die Autoren sowie das Labor übernehmen keine Garantie für Fehlerfreiheit und Aktualität in ihren Studien und Verlautbarungen und übernehmen keine Verpflichtung, die Ausführungen in Studien und anderen Verlautbarungen zu aktualisieren. Das Material dieser Studie dient nur zu Informationszwecken.