

Expose zur Masterarbeit „Aufbau einer unabhängigen natürlich-sprachlichen Mensch-Roboter-Interaktion“

Martin Eisoldt

April 2019

1 Ziel

Die aktuelle Entwicklung der Technik weist klar die Tendenz auf, dass in immer mehr Bereichen des Alltags Roboter zum Einsatz kommen. Diese können zunehmend mehr Aufgaben des täglichen Lebens übernehmen und entwickeln sich somit zu echten Assistenzrobotern. Jedoch variieren die Aufgaben, die sich stellen, mit jedem Tag. So ist es nicht ausreichend, dass diese Roboter zu vorgegebenen Zeiten die immer gleichen Tätigkeiten durchführen. Vielmehr müssen sie mit dem Nutzer interagieren und flexibel auf die Anforderungen reagieren. Eine solche Interaktion lässt sich am einfachsten über Sprache umsetzen. Allerdings muss der Roboter dafür menschliche Sprache verstehen können und auch die richtigen Tätigkeiten durchführen.

Aktuell sind große Unternehmen bereits damit beschäftigt, Sprachassistenten zu erstellen. Zu diesen gehört beispielsweise Amazon Alexa oder Google Assistant. Jedoch stellen sich an dieser Stelle immer datenschutzrechtliche Bedenken, da gerade diese beiden Unternehmen ihren Sitz in den USA haben und dortige Gesetze zum Teil stark von europäischen abweichen. So berichtet Pfeifle [20] darüber, dass Strafverfolgungsbehörden in den USA Zugriff auf Aufzeichnungen erhalten haben, die von Alexa erstellt wurden. Dies ist ohne die Zustimmung durch den Nutzer geschehen.

Für die Verarbeitung einer Nutzereingabe wird diese abgeschickt, durch zumeist unternehmenseigene Infrastruktur verarbeitet und das Ergebnis wieder zurück an den Nutzer geschickt. Somit ist nicht klar, auf Basis welcher Gesetze die Verarbeitung stattfindet, denn der Standort der Cloud Infrastruktur ist meistens unbekannt. Des Weiteren stellt Pfeifle [20] heraus, dass durch Alexa erzeugte Aufzeichnungen mindestens teilweise gespeichert werden und für weiteres Training verwendet werden. Lau et al. [18] stellen fest, dass besonders Personen, die keinen Sprachassistenten besitzen, große Bedenken bezüglich der Privatsphäre haben. Besitzer solcher Systeme wiederum sind stark darauf angewiesen, den

Herstellern zu vertrauen. Da sprachgesteuerte Assistenzroboter gerade für die Unterstützung von älteren Menschen, die zumeist noch keinen ähnlichen Systeme verwenden, eingesetzt werden können, sollten solche Bedenken aus dem Weg geräumt werden.

Aus diesem Grund ist lokale Verarbeitung der Eingabe und somit die Hoheit über die eigenen Daten wünschenswert. Dies kann mittels OpenSource Projekten wie *Mycroft.ai* oder *Snips* erreicht werden. In dieser Arbeit sollen mögliche Einsatzszenarien eines solchen Systems betrachtet werden. Dafür ist zunächst eine genauere Betrachtung der Architektur sowie Funktionsweise dieser System nötig. Am Ende des Projekts soll es neben diesen theoretischen Betrachtungen auch zu einer prototypischen Umsetzungen eines solchen Assistenten auf Basis geeigneter Assistenzsoftware kommen. Dieser Prototyp sollte im Anschluss dran auch auf seine Praxistauglichkeit untersucht werden.

2 Theoretischer Hintergrund und Ausgangslage

2.1 Aktuelle Forschung

Aktuelle Forschung im Zusammenhang mit Sprachassistenten beschäftigt sich wie Anke et.al. [11] mit dem Einsatz von Systemen wie Amazon Alexa oder Google Assistant im Bereich des Kundenservices. Gerade in diesem Zusammenhang stellen sich die datenschutzrechtlichen Herausforderungen der Datenverarbeitung in der Cloud, die zumeist in den Vereinigten Staaten lokalisiert ist. In diesem Projekt wurden der Datenschutz durch zwei Ansätze eingehalten. Zum Einen stimmt der Nutzer der Datennutzung zu, ist sich also bewusst, auf welche Art seine Daten verwendet werden. Zum Anderen werden alle Verarbeitungsschritte der Spracheingabe, von der Übersetzung in maschinenverständliche Sprache bis zur Antwort an den Nutzer, auf Hardware des Projektkunden in Deutschland durchgeführt.

Andere Arbeiten wiederum beschäftigen sich mit dem Einsatz von Sprachassistenten in der Pflege. Der Fokus von Hellwig et al. [16] liegt hierfür auf den Grundlagen für einen kommerziellen Einsatz von Amazon Echo und Google Home in der Pflege. Dabei wird unter anderem betrachtet, wie Senioren davon überzeugt werden können, ein solches System zu nutzen. Wichtig sind dafür die wahrgenommene Nützlichkeit und Benutzerfreundlichkeit. Aber auch Neugierde und Skepsis sind nicht zu vernachlässigen. Als sinnvoll haben sich an dieser Stelle Funktionen erwiesen, die Smart Home-Steuerung ermöglichen. Aber auch Erinnerung an Termine oder alltägliches, wie Medikamenteneinnahme, haben Nutzer von dem System überzeugt. Gleichzeitig können Aufgaben im Bereich der Dokumentation für die Pflegekräfte wesentlich erleichtert werden und somit mehr Zeit für den Patienten aufgewandt werden.

Auch gibt es andere Arbeiten, die sich auf die individuelle Nutzung von Sprachassistentensystemen im Bereich der Steuerung von Smart Homes beschäftigen, beispielsweise Guaman et al. [15]. Dabei werden mittels Alexa Sprachbefehle durch

Amazon Services verarbeitet und die erhaltenen Resultate an einen lokalen Raspberry Pi übertragen, der die Steuerung der einzelnen Geräte übernimmt. Dafür werden Alexa eigene Skills antrainiert, also Fähigkeiten, die eigene Befehle erkennen und entsprechend an ein Endgerät übermitteln. Sridhar und Tolentiono [22] legen ihren Schwerpunkt auf die Verlagerung der Auswertung von Sprachbefehlen weg von der Cloud hin zu lokalen Recheneinheiten. Dabei ist allerdings nicht der Datenschutz Beweggrund, sondern die verstrichene Zeit zwischen Eingabe und Antwort. Für diese Untersuchung werden einzelne Bestandteile von Mycroft genutzt. Beispielweise wurde die Mycroft Skills Engine lokal auf einem Raspberry Pi ausgeführt. Aber auch die Tools der Sprache-zu-Text sowie Text-zu-Sprache Umwandlung sind die gleichen wie bei Mycroft. Dabei sind sie zu dem Ergebnis gekommen, dass, mit einigen Modelloptimierungen, die Verlagerung der Berechnung auf die lokalen Geräte möglich ist. Allerdings benötigen komplexere Verarbeitungen auch mehr Optimierungen. Shaygan et. al. beschäftigen sich wiederum mit einem Vergleich verschiedener Sprachassistentensysteme. Die Vergleichsgeräte sind dabei unter anderem Amazon Echo, Google Home sowie Mycroft Mark I. Neben Faktoren wie Kosten und Design, werden auch die Funktionen sowie Performance verglichen. Für die Leistungsfähigkeit zählen dabei Zuverlässigkeit, Genauigkeit und Sicherheit [21]. Andere Arbeiten wiederum beschäftigen sich mit dem Einsatz von Sprachassistenten zur Kontrolle von Robotern im Bereich der Pflege. So setzen Jungbluth et al. [17] Alexa als Sprachinterface für Roboter ein.

2.2 Verfahren zur Erkennung der Nutzerintention durch Sprachassistenten

Der prinzipielle Aufbau von Sprachassistenten folgt immer einem gleichen Schema. Zuerst muss mittels Weckwort dem Assistenten mitgeteilt werden, dass eine Anfrage zur Interaktion folgen wird. Bei bekannten Assistenten von Google und Amazon sind diese Weckwörter beispielsweise „Siri“ und „Alexa“.

Im Anschluss danach muss das gesprochene Wort in Text umgewandelt werden. Dafür wird eine Sprache-zu-Text Engine verwendet. Dieser Text wird dann darauf untersucht, welcher Befehl ausgeführt werden soll. Anhand dieser Analyse wird danach die gewünschte Funktion ausgeführt, welche einen Skill darstellt. Um dem Nutzer eine gesprochene Antwort geben zu können, muss diese Antwort per Text-zu-Sprache Engine in eine Audio Ausgabe übersetzt werden.

2.3 Architektur von Mycroft

Eines der neusten Sprachassistentensysteme ist *Mycroft.ai*. Dieses Projekt wurde erstmals 2015 auf Kickstarter sowie Indiegogo finanziert. Dabei handelt es sich um einen kompletten Sprachassistenten inklusive Mikrophon und Lautsprecher. Drei Jahre später wurde eine erneuerte Version des Assistenten erneut über Crowdfunding finanziert [4]. Solche Endgeräte sind allerdings für diese Arbeit nicht von Interesse. Vielmehr steht im Fokus, einen eigenen Assistenten mittels der OpenSource Software *Mycroft.ai* zu erstellen.

MyCroft nutzt dabei die Weckwort Engine *Precise* um eine Nutzerinteraktion zu entdecken. Diese basiert dabei auf einem Neuronale Netz, so dass es auch jederzeit möglich ist, eigene Weckwörter zu trainieren. *Precise* benötigt aber keine Internetanbindung, sondern funktioniert auf dem Endgerät, wodurch der Datenschutz verbessert wird [7]. Standardmäßig reagiert Mycroft dabei auf "Hey Mycroft". Gegenüber anderen vergleichbaren Engines hebt sich *Precise* vor allem dadurch ab, dass es OpenSource ist [9]. Aus diesem Grund hat MyCroft die eigene Architektur von *PocketSphinx* auf *Precise* umgestellt. Problem an *PocketSphinx* ist beispielsweise die mangelnde Genauigkeit und dass es nur in Kombination mit englischer Sprache funktioniert [10].

Für die Sprache zu Text Umwandlung verwendet MyCroft bislang noch externe Software. Die verwendete Engine ist *Google STT*. Dies ist eine Cloud Anwendung von Google, mit der bis zu 120 verschiedene Sprachen erkannt werden können. Des Weiteren werden auch *IBM Watson Speech to Text* und *wit.ai Speech to Text* unterstützt. Auch für diese beiden Anwendungen wird ein Internetzugang benötigt, da die Umwandlung in der Cloud stattfindet. Mycroft arbeitet selber an einer Engine zur Konvertierung von Sprache zu Text, *OpenSTT*. Dies geschieht in Zusammenarbeit mit Organisationen wie *Mozilla Voice* [10]. Allerdings ist es dem Mycroft Forum zufolge auch möglich, diese Umwandlung lokal auf dem Gerät ablaufen zu lassen.

Um den Output der Sprache-zu-Text Konvertierung auch auszuführen, wird ein Intentparser benötigt. Dafür kommt aktuell der von Mycroft entwickelte Parser *Adapt* zum Einsatz. Um den gewünschten Befehl zu erkennen wird ein Schlüsselwortmatching durchgeführt. Das Ergebnis sind dann verschiedene Wahrscheinlichkeiten für erkannte Befehle, anschließend wird der Befehl mit der höchsten Wahrscheinlichkeit ausgeführt. Da es sich dabei um ein sehr leichtgewichtiges Framework handelt, kann dies direkt auf dem Gerät ausgeführt werden [1]. Alternativ kann auch *Padatious* eingesetzt werden. Dies ist ein Intent Parser auf Basis eines neuronalen Netzes und wird noch von Mycroft entwickelt. Ziel ist, mit diesem Parser *Adapt* zu ersetzen [10].

Um eine Antwort an den Nutzer zu geben, muss das Ergebnis der Befehlsausführung wieder in Sprache umgewandelt werden. Dafür nutzt MyCroft *Mimic*. Dies kann auch lokal auf dem Gerät ausgeführt werden und zeichnet sich durch eine sehr natürlich klingende Stimme aus. Außerdem können weitere Sprachengines gewählt werden, zum Beispiel *eSpeak*, *MaryTTS*, *Google TTS* und *FATTS* [10].

Von Interesse ist auch das Erlernen neuer Skills. Standardanwendungen für Sprachassistenten sind beispielsweise das Abspielen bestimmter Musik, allgemeine Informationen über das Wetter und ähnliches. Diese Anwendungen spielen allerdings für diese Arbeit keine Rolle. Vielmehr muss ein Sprachbefehl so in einen Befehl für Loomo umgewandelt werden, dass dieser die gewünschte Tätigkeit ausführt. Für das Training eines neuen Skills stellt Mycroft ein Template im eigenen Repo [8]. Ein solcher Skill besteht aus drei Teilen:

- dialog - die Aussage, die von Mycroft gesprochen wird
- intent - Befehl, der ausgeführt werden soll
- utterance - Phrase, die der Nutzer nach dem Wakeword spricht

2.4 Implementierung von Mycroft in Android Umgebung

Um eine Interaktion mit Loomo zu ermöglichen, ist es nötig, eine Anwendung auf Android Basis zu erstellen. Dafür gibt es auch eine Umsetzung von Mycroft. Diese wird aber nicht offiziell von Hersteller entwickelt, sondern durch die Community. Allerdings verweist der Hersteller direkt auf diese Version und führt sie auch in der Dokumentation. Programmiert ist die App in Kotlin, einer Programmiersprache, die erst seit 2016 als stabile Version existiert. Dabei weist Kotlin eine volle Kompatibilität zu Java auf, beispielsweise können ohne Probleme Java Klassen aufgerufen werden.

Damit die Mycroft App auf einem Android Gerät funktioniert, muss sie mit einem Gerät interagieren, das als mycroft-core Instanz agiert [5]. Diese Instanz ist für die Umwandlung der Sprache in Text und das parsen der Intents verantwortlich. Dafür bietet es sich an, einen Raspberry Pi als eine solche Instanz zu verwenden. Besonders, da dieses Gerät energiesparend ist und somit immer im Hintergrund eingeschaltet sein kann und zeitgleich eine große Menge an persönlichen Einstellungen erlaubt. Offiziell unterstützt werden dabei die Modelle Pi3 B sowie Pi3 B+, mit dem Modell Pi2 ist laut Hersteller nur eine sehr langsame Verarbeitung möglich. Alle anderen Modelle weisen nicht ausreichend Ressourcen auf, um die Eingaben verarbeiten zu können [6].

Für die App wird minimal Android SDK 19 verlangt, somit sollte sie ohne Problem auf Loomo laufen. Dieser hat die SDK Version 21. Des weiteren wird eine Verbindung zu der Instanz benötigt, diese kann nur via Websockets hergestellt werden.

2.5 Alternative OpenSource Sprachassistenten

Aktuell gibt es auch weitere OpenSource Sprachassistenten. Da Projektziel jedoch die Erstellung einer Anwendung ist, die in Zusammenarbeit mit *Loomo* funktioniert, sind nur diese von Interesse, die auch Android unterstützen.

In diesem Zusammenhang taucht mit Snips ein anderes Projekt auf. Dieses Projekt ist im Sommer 2017 gestartet und erlaubt die Erstellung eines eigenen Sprachassistenten über eine Webkonsole. Der prinzipielle Aufbau ähnelt dabei dem von Mycroft. Jedoch folgt das System der Architekturprinzip „Privacy by Design“. Das bedeutet in diesem Fall, dass keine Daten zu einem Server übertragen werden, sondern alles lokal geschieht [12]. Aufgrund dieser Tatsache kann ein Sprachassistent auf Basis von Snips komplett ohne WLAN betrieben werden, lediglich zur initialen Einrichtung wird eine Internetverbindung benötigt.

2.6 Entwicklung von Apps zur Steuerung von Loomo

Für die Steuerung von Loomo gibt es ein eigenes, durch Segway bereitgestelltes, SDK. Dieses befindet sich noch in der Entwicklung, kann aber ohne Probleme mittels Gradle in eine Android App eingebunden werden. Allerdings hat der Umstand, dass die Software noch in der Entwicklung ist, eine lückenhafte Dokumentation zur Folge. Teilweise ist es einfacher, sich an offiziellen Beispielapps zu orientieren, die die Grundfunktionen des Roboters implementieren. Daraus geht hervor, dass es für Loomo eine Liste mit verschiedenem Verhalten gibt, das bei Bedarf eingesetzt werden kann. Neben Kopfbewegungen umfasst dies auch das Drehen um die eigene Achse [3].

2.7 Datenschutz

Im Rahmen der Datenschutzgrundverordnung (DSGVO) wurde definiert, welche Daten als persönliche Daten gelten und unter welchen Umständen sie verarbeitet werden dürfen. Unter anderem zählen Sprachaufnahmen als persönliche Daten. Solche Daten dürfen erarbeitet werden, allerdings muss für den Nutzer transparent sein, auf welche Art und zu welchem Zweck diese Daten verarbeitet werden [19]. Allerdings gibt es auch das Schlupfloch der "Verbesserung der Nutzererfahrung" [14].

Ein Problem der Nutzung von Sprachassistenten ist, dass die Verarbeitung der Daten zumeist in der Cloud des jeweiligen Anbieters stattfindet. Dieser speichert den Sprachbefehl auch nach seiner Verarbeitung, sofern die Löschung nicht durch den Nutzer vorgenommen wird [13]. Von Interesse sind in diesem Zusammenhang, welche Daten verarbeitet werden und wo die Speicherung dieser geschieht. Außerdem ist es nötig, dass die Datenübertragung sowie Verarbeitung sicher ablaufen [13]. Ein weiteres Risiko besteht darin, dass auch erst dann Daten verarbeitet werden, sobald das Weckwort gesagt wurde. Dieses Problem hat [13] genauer untersucht. Denn gerade die Sensitivität auf das Weckwort kann zu fehlerhafter Aktivierung und damit verbundener Verarbeitung der Daten führen. Kritisch wird es in diesem Zusammenhang vor allem dann, wenn diese Verarbeitung nicht mehr vollautomatisch geschieht. So wurden Mitschnitte durch Alexa von Amazon Mitarbeitern von Hand transkribiert [2]. Diese Daten umfassen teilweise hoch sensible Daten, wie Kontodaten, ohne dass der Nutzer von der Aufnahme weiß.

3 Fragestellungen/Zielsetzungen

3.1 Untersuchungsziel

Ist es möglich, einen Assistenzroboter auf Basis von OpenSource Sprachassistenzsystemen zu steuern?

Teilziele:

- Analyse:
 - Wie unterscheiden sich OpenSource Sprachassistenten von Amazon Alexa? (im Hinblick auf Funktionalität, Architektur, Individualisierbarkeit)
 - Welche möglichen Einsatzszenarien gibt es für OpenSource Sprachassistenten für Assistenzroboter?
- Konzept:
 - Welche dieser Szenarien sind mit der gegebenen Hardware umsetzbar?
 - Welches der in Frage kommenden Sprachassistenzsysteme ist am besten für die Umsetzung geeignet?
- Evaluation:
 - Können Nutzer ohne größeres Vorwissen die Anwendung problemlos verwenden um die zuvor erstellten Einsatzszenarien zu bewältigen?

3.2 Vorgehen

Zu Beginn der Arbeit werden die beiden OpenSource Tools Mycroft und Snips eingehend miteinander verglichen. Außerdem soll eines der aktuell marktführenden Systeme, Amazon Alexa, in diesen Vergleich mit einbezogen werden. Dieser Vergleich soll dabei den Fokus auf Privatsphäre sowie Leistungsfähigkeit des Assistenzsystem in Zusammenarbeit mit einer Android App legen. Außerdem ist von großer Relevanz, wie eine Steuerung Loomo umsetzbar ist.

Der Prototyp wird auf Basis von zuvor erstellten Einsatzszenarien umgesetzt. Dieser basiert dabei auf Android, allerdings ist von einer Anpassung von Mycroft auszugehen. Die Umsetzung umfasst die Erstellung neuer Skills, die dann entsprechende Befehle für Loomo geben sowie möglicherweise der Einsatz unterschiedlicher Tools für einzelne Schritte der Verarbeitungspipeline.

Um die Praxistauglichkeit zu beurteilen, werden sowohl quantitative als auch qualitative Tests durchgeführt. Für die quantitative Bewertung sind dabei Faktoren wie Verarbeitungszeit aber auch Genauigkeit von Interesse. Die qualitativen Tests hingegen sollen mit mehreren Probanden durchgeführt werden. Dabei soll bewertet werden, wie gut Nutzer sich mit dem System zurecht finden. Außerdem können in dem Zuge auch weitere quantitative Daten erhoben werden,

da unterschiedliche Personen verschieden sprechen. Somit kann eine Beurteilung der Zuverlässigkeit mit einer ausreichenden Datenmenge getroffen werden. Abschließend lassen sich anhand der eingangs durchgeführten Vergleiche Rückschlüsse darauf anstellen, ob Mycroft für dieses System die richtige Wahl war oder die gewählten Szenarien mit einer anderen Assistenzsoftware besser durchführbar sind.

4 Zeitplan

Aufgabe	Start	Ende	Zeit in Wochen
Masterarbeit	06.05.	07.10.	22
Vergleich OpenSource Sprachassistenten mit Alexa	06.05.	26.05.	3
Erstellung Einsatzszenarien	27.05.	02.06.	1
Auswahl geeigneter Szenarien für Prototyp			
Konzepterstellung für App	03.06.	09.06.	1
Auswahl geeigneter Systeme (Roboter, Sprachsoftware)			
Implementierung der App	10.06.	04.08.	8
Evaluation der App durch Nutzer	05.08.	11.08.	1
Auswertung der Nutzerevaluation	12.08.	18.08.	1
Schreiben der Arbeit	19.08.	15.09.	4
Korrektur der schriftlichen Ausarbeitung	16.09.	22.09.	1
allgemeiner Puffer	23.09.	07.10.	2

Tabelle 1: Zeitplan Pufferzeiten von 30 % bereits in die Punkte eingerechnet

5 Inhaltsverzeichnis

1. Einleitung (5 %)
 - (a) Motivation
 - (b) Aufbau der Arbeit
2. Vergleich von OpenSource Sprachassistenzsystemen mit Alexa (15 %)
 - (a) Betrachtung möglicher Sprachassistenzsysteme
 - (b) Architektur der Systeme
 - (c) Betrachtung der Systeme unter dem Aspekt des Datenschutzes
 - (d) Wie gut sind die Systeme für eigene Anforderungen individualisierbar?
 - (e) Welche Kosten entstehen bei der Nutzung (Anschaffung, nötige Infrastruktur, Betrieb)
 - (f) Besonderheiten der einzelnen Systeme
3. Umsetzung eines Prototyps auf Basis von Mycroft zur Steuerung von Loomo (60 %)
 - (a) Einsatzszenarien für Assistenzroboter mit Sprachsteuerung
 - (b) Auswahl von geeigneten Szenarien für den Prototyp
 - (c) Konzept für Umsetzung eines Prototyps
 - (d) Übersicht über die Bordmittel von Loomo und Begründung der nötigen Erweiterung
 - (e) Umsetzung des Prototyps
4. Bewertung des Prototyps (15 %)
 - (a) Erstellung von Aufgaben für die Probanden ausgehend von den Einsatzszenarien
 - (b) Wie erfolgreich konnten die Aufgaben bewältigt werden?
 - (c) Wie sicher haben sich die Probanden bei der Bedienung des Prototyps gefühlt?
5. Fazit (5 %)
 - (a) Zusammenfassung der Arbeitsergebnisse
 - (b) Diskussion
 - (c) Ausblick

Literatur

- [1] Adapt.
<https://mycroft.ai/documentation/adapt/>. Accessed: 2019-04-17.
- [2] Amazon-mitarbeiter hören sich privatgespräche mit alexa an.
<https://www.spiegel.de/netzwelt/gadgets/amazon-mitarbeiter-hoeren-sich-tausende-privatgespraeche-mit-alexa-an-a-1262315.html>. Accessed: 2019-04-22.
- [3] Loomo sdk api reference.
<https://developer.segwayrobotics.com/developer/documents/java-doc.html>.
- [4] Mycroft ai – opensource alternative zu alexa und co.
<https://www.panbachi.de/mycroft-ai-opensource-alternative-zu-alexa-und-co-350/>. Accessed: 2019-04-11.
- [5] Mycroft-android github repository.
<https://github.com/MycroftAI/Mycroft-Android>. Accessed: 2019-04-11.
- [6] Mycroft hardware requirements and compatibility chart.
<https://mycroft.ai/documentation/picroft/#hardware-requirements-and-compatibility-chart>. Accessed: 2019-04-11.
- [7] Mycroft precise.
<https://github.com/MycroftAI/mycroft-precise>. Accessed: 2019-04-17.
- [8] Mycroft skills repo.
<https://github.com/MycroftAI/mycroft-skills/>. Accessed: 2019-04-11.
- [9] Wakewordengine vergleich.
<https://github.com/MycroftAI/mycroft-precise/wiki/Software-Comparison>. Accessed: 2019-04-17.
- [10] What is mycroft.
<https://mycroft.ai/documentation/mycroft-software-hardware/>. Accessed: 2019-04-17.
- [11] Jürgen Anke, Uwe Fischer, and René Lemke. Integration digitaler sprachassistenten in den kundenservice am beispiel der stadtwerke leipzig. In Michael Räckers, Sebastian Halsbenning, Detlef Rätz, David Richter, and Erich Schweighofer, editors, *Digitalisierung von Staat und Verwaltung*, pages 25–36, Bonn, 2019. Gesellschaft für Informatik e.V.

- [12] Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*, 2018.
- [13] Verbraucherzentrale NRW e.V. Amazon alexa: Wann ist der sprachassistent ganz ohr? ein reaktions-check. kurzuntersuchung der verbraucherzentralen, 2017.
- [14] Eoghan Furey and Juanita Blue. Alexa, emotions, privacy and gdpr. In *Proceedings of the 32nd International BCS Human Computer Interaction Conference*, page 212. BCS Learning & Development Ltd., 2018.
- [15] Steven Guamán, Adrián Calvopiña, Pamela Orta, Freddy Tapia, and Sang Guun Yoo. Device control system for a smart home using voice commands: A practical case. In *Proceedings of the 2018 10th International Conference on Information Management and Engineering*, pages 86–89. ACM, 2018.
- [16] Andre Hellwig, Caroline Schneider, Sven Meister, and Wolfgang Deiters. Sprachassistenten in der pflege-potentiale und voraussetzungen zur unterstützung von senioren. *Mensch und Computer 2018-Tagungsband*, 2018.
- [17] Jan Jungbluth, Rolf Krieger, Wolfgang Gerke, and Peter Plapper. Combining virtual and robot assistants-a case study about integrating amazon’s alexa as a voice interface in robotics. In *Robotix-Academy Conference for Industrial Robotics (RACIR) 2018*, page 5. Shaker, 2018.
- [18] Josephine Lau, Benjamin Zimmerman, and Florian Schaub. Alexa, are you listening?: Privacy perceptions, concerns and privacy-seeking behaviors with smart speakers. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW):102, 2018.
- [19] Europäisches Parlament. Verordnung (eu) 2016/679 des europäischen parlaments und des rates vom 27. april 2016 zum schutz natürlicher personen bei der verarbeitung personenbezogener daten, zum freien datenverkehr und zur aufhebung der richtlinie 95/46/eg (datenschutz-grundverordnung). Technical report, Tech. Rep. Amtsblatt der Europäischen Union, 2016.
- [20] Anne Pfeifle. Alexa, what should we do about privacy: Protecting privacy for users of voice-activated devices. *Wash. L. Rev.*, 93:421, 2018.
- [21] Amir Shaygan, Gaya Hacıane, Rassaniya Lerdphayakkarat, Juliana Miaira Kutch, Jonathan Roschke, and Tugrul Daim. Decision making model for choosing voice-operated intelligent speakers for graduate students. In *2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, pages 1–9. IEEE, 2018.
- [22] Smruthi Sridhar and Matthew E Tolentino. Evaluating the impact of pushing voice-driven interaction pipelines to the edge. 2018.