Technische Universität Berlin

DOCUMENT BUILD DATE: 15th April 2018
DOCUMENT STATUS: Beta

Development and Evaluation of a Service Bot in the e-Government Sector

Bachelor Thesis
am Fachgebiet Agententechnologien in betrieblichen Anwendungen und der Telekommunikation (AOT)
Prof. Dr. Dr. h.c. Şahin Albayrak
Fakultät IV Elektrotechnik und Informatik
Technische Universität Berlin

vorgelegt von
Mohamed Megahed

Betreuer: **Dr. Andreas Lommatzsch**
Gutachter: Prof. Dr. Dr. h.c. Şahin Albayrak
Prof. Dr. Odej Kao

Matrikelnummer: 342655

Erklärung der Urheberschaft

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form in keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Ort, Datum Unterschrift

Berlin, den 15th April 2018

Abstract

Though not a recent phenomenon, chatbots and voice assistants are increasingly gaining unprecedented attention as a successor for mobile and web apps. While still emerging with no defined standards or set protocols, with a hype on the rise, tensions between industry giants with products like Amazon's Alexa, Apple's Siri or the Google Assistant unveil new examples for providing an enriched user experience for consumers and businesses. The surrounding ecosystem also plays a major role in spreading the platforms available while exploring new horizons with alternative approaches and business models. Today voice assistance are already present around indoor spaces, in the car or on the go. They live in smartphones, on fancy TV sets or even as stand-alone devices. Yet, they are still a new terrain to discover and great potential to unleash in numerous contexts.

One such scenarios involves the public sector. In this work, we explore the Alexa Skills Kit by Amazon in combination with respective services to develop a voice assistant for the local city council extending the current chatbot's functionality available on http://service.berlin.de. We touch on the technical challenges and possibilities in implementing a Software as a Service (SaaS) system for e-Government inquiries and analyse its usability as well as its effectiveness in replacing traditional information lookup. We then examine the goals we define for our use case to our achievement with the APIs and SDKs available at the time. With respect to those, we also report on the opportunities, challenges and limitations faced in the development process.

Finally, we study the current state of voice assistants and service bots on the market and the future of this trend from a technical and a social point of view.

[SUPERVISOR NOTE] CONCLUSION: DAS ERGEBNIS DER ARBEIT SOLLTE MAN ZUM SCHLUSS NOCH IM ABSTRACT ERGÄNZEN.]

Zusammenfassung

Auch wenn Chatbots und Sprachassistenten keine Randerscheinung mehr sind, erlangen sie in letzter Zeit eine steigende Bedeutung als Nachfolder von mobilen Anwendungen und Web Applikationen. Trotz des Mangels an definierten Standards oder feste Protokolle, mit einer zunehmenden Aufmerksamkeit auf dem Markt herrscht große Konkurrenz. Ob es durch Amazons Alexa, Apples Siri oder den Google Sprachassistenten hervorgeht, zeigen sich zahlreiche Beispiele, wie die Nutzererfahrung auf privaten oder Geschäftsebene weitgehend bereichert werden kann. Mit Lock-in Effekten spielen Drittanbieter eine große Rolle bei der Verbreitung der einen oder anderen Plattform und sorgen für neue Interaktionsmodelle. Der steigende Nutzen von Sprachassistenten zuhause, im Büro und unterwegs erlaubt viele neue Erkenntnisse und Erfahrungen, die dem Marktumfeld zu verdanken sind.

Auch im öffentlichen Dienst wird dieser Nutzen vertreten und wertvoll eingeschätzt. In der folgenden Recherche untersuchen wir Alexa von Amazon im Zusammenhang mit anderen verfügbaren Plattformen und erweitern die Fähigkeiten des Sprachassistenten durch den Webauftritt des berliner Stadtportals (BerlinOnline) um weitere Dienste sowohl für die vielfältigen und mehrsprachigen Einwohner der Hauptstadt als auch deren Senatsverwaltung. Als Grundlage wird auf die Funktionalität des Chatbots auf dem Stadtportal http://service.berlin.de zurückgegriffen. Anhand dieses Anwendungsfalls werden die prinzipiellen Möglichkeiten und Herausforderungen bei der Implementierung eines Sprachassistenten für die Stadtverwaltung aus technischer und organisatorischer Sicht analysiert. Zusätzlich wird die Usability des Systems als Ergänzung für einen Abfragedienst in Erwägung gezogen. Anschließend werden die erreichten Ziele mit den gesetzten Anforderungen in Perspektive gestellt. Darüber hinaus wird über Systemgrenzen sowie individuellen und systematischen Fehlerquellen berichtet, die bei der Entwicklung behandelt worden sind.

Schließleich nehmen wir den aktuellen Trend von Sprachassistenten und Chatbot-Dienste unter der Lupe und diskutieren eventuelle Zukunftentwicklung mit dem gegeben Stand jenseits des e-Governance Bereichs. **[SUPERVISOR NOTE] CONCLUSION: DAS ERGEBNIS DER ARBEIT SOLLTE MAN ZUM SCHLUSS NOCH IM AB-**

STRACT ERGÄNZEN.]

Acknowledgements

⌈ possibility to make inline notes ⌋ **[CITATION]** when a citation is missing

✍ **TO-DO**: todos

[TO CITE]

[SUPERVISOR NOTE] BITTE BETREUERNOTIZEN AUCH IM FLIESSTEXT MIT \SN{TEXT}]

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With over a third of the world's population projected to own a smartphone in 2018 [15] and a substantial fraction thereof using smarthome gadgets and appliances on a daily basis, AI's role has become more interesting than ever in many disciplines including but not limited to productivity and entertainment. Many technologies we take for granted today, such as dictation and word prediction, recommender systems or other digital analytics depend on Machine Learning and Natural Language Processing techniques that were only made possible thanks to the high computational power shipped in most devices gradually overtaking the consumer market. This transition also facilitated the introduction of a new form of interaction through conversation with the hardware, paving the way to an aspiration modern societies have been striving globally [10]. Whether in blockbuster 60s drama as seen in "Breakfast at Tiffany's" (1961) or in Sci-Fi romance in the movie "Her" (2013), an obsession with voice technologies is featured throughout from an answering machine on tape to a fully personalized but mass-produced voice-based operating system to even become a protagonist.

Conversational bots were already prevalent since the 80s in the form of Question/Answer systems based on query programming languages like PROLOG and SQL. ELIZA, considered as the world's first chatbot and though quite superficial as an NLP-based programme for psychoanalysis, already at its early stages demonstrated how humans can become emotionally attached to machines, transcending over the anomaly of making conversation *not* with a human [17]. Today, combining ML with retrieval-based approaches allows a more advanced interaction with the system and yields smarter and more personalized conversations between man and machine. Consequently, it is no longer a surprise that chatbots acquire social skills to make Xiaoice, the empathetic bot from China, possibly a new kind of friend made of silicon revoking the fiction element from "Her".

So far, voice assistants represent an additional layer of abstraction from software

beyond the graphical user interface (GUI) and are hence the closest we have come towards human communication. They de-construct another barrier between the user and the hardware as voice communication generally does not require profound computer literacy and the conversational models rely on our inherent ways of expression. As they simulate the human aspect and imitate its behaviour for instance with small-talk abilities [14], voice assistants are regarded as a convenience for daily tasks and are on their way to becoming a de-facto replacement to sophisticated actions we perform on the screen. In fact voice searches now compose a large marget segment ⌈ check quality of statistics ⌋ and are predicted to make up about 30% [7] [16] of all searches by 2020 with currently highest rates coming from the youngest generations based on a survey [19] by Global Web Index. According to the Alpine.ai 2017 Voice Labs report there were 33 million voice-first devices in circulation in 2017 worldwide [1] with tremendous shifts in number of units sold between 2015 and 2017. This and more statistics hint at a revolutionary change towards our use as much as the introduction of the GUI and mouse were to the command line interface. Increasingly, we recognize voice as a *new* user interface, also known as Voice User Interface (VUI), and analyse good practices and design guidelines for it.

Since sound and voice are primal stimuli in the human brain [9], using them becomes more instinctive, making us in turn process our ideas starting with an "inner voice" that translates easiest into words when we speak it before "funneling" it to actions [3]. Arguably, on a market scale, this gives voice assistants today a competitive innovation advantage (CIA) for they are hence more accessible to further demographic groups with a growing wider acceptance.

Meanwhile, statistics [11] show that time spent on messaging apps already surpassed average uptime on social media , which indicates how the former is more desirable as a communication format on mobile platforms and how having a conversation is an easier way to interact with a device instead of downloading an app for every task. ⌈ link ok? ⌋ Further, the speech-to-text/text-to-speech domain has become more powerful with steadily increasing processing power, an effect of Moore's law we only get to understand lately in addition to the gradual lessening of the dominance of Chomskyan theories of linguistics (e.g. transformational grammar) [5]. With an integration in most modern operating systems, speaking to a device has become no longer a absurd novelty. Inadvertently, Google Now, the voice assistant built into the OS currently with the highest usage shares among computer and smartphones [6], supports understanding of multiple languages in the same sentence for multilingual interaction. Moreover, the Echo Dot recorded the peak for best-sellers for 2017 on Amazon with unprecedented numbers showing a high customer retention and satisfaction rate [12]

As we constantly challenge our expectations towards technology, our not too far-fetched imagination makes us question the ability of AI to make a machine able to react to everything we say. Although we are still far from this step, at least for the consumer level, we dedicate a lot of effort to make it happen with examples like IBM's Watson. And as long as we still do not exactly understand human intelligence yet, it is hard to fathom AI as a holistic field. As such, it is therefore more realistic to consider the current works in the field as *intelligence amplification* [3] empowering human take better decisions beyond their normal brain processing power.

1.1 Motivation

With the aforementioned, we try to think how voice assistants could come handy and why we want to invest in such technology, juxtaposing it to available alternatives. If we consider human workforce (i.e. customer service agents) on one hand, it is commonplace that these are more expensive, less available due to restrictive working hours and not always aware of the full circumstances related to an issue they are supposed to fix or a question they are to answer.

Sometimes knowing more about a person could almost become a dangerous tool since it gives room to manipulate them. A client in a shop for instance can have their decision influenced by the seller and eventually get tricked into buying a product based on wrong advice. Although there is practically little the client can normally do to circumvent misinformation if that seller is replaced by an algorithm, having an automated system like a voice assistant step in gives at least a more neutral impression since it are not directly expected to act with malicious intentions.

Since the point of availing voice assistant is to act in a person's interest, we also want an information system not to confuse us or to limit our cognitive abilities. ⌈ talk about cognitive behaviour ⌋ Besides, we can at least ensure that a voice assistant will not become moody and intentionally want to make our lives harder for this reason as opposed to a human. And so though a voice assistant or a chat bot may not potentially answer every question we throw at it, we want to at least presume that it would give us no information and not partial truths or lies. This is why most credible companies elaborate explicitly on their terms, conditions and privacy statements making them accountable on the products they produce and the services they offer. A consumer therefore feels more empowered to assert any faults originating from an automated system than from a human and conditionally has an assurance that they can prevent any violations more systematically.

Eventually a person is more likely to develop a certain kind of trust in a machine

more than in a human once the technology is established and widespread. Cars, email, and other gadgets or services we take for granted today are living proof of how this trust can grow is inevitable, for the better or worse and depending on the degree of affinity, aversion or ignorance in a business sector. Trust in a system can grow once it is certified to have little to minimal exceptions. Besides enriching the value chain, it is a key in setting a technology to become an industry standard. Therefore, if a voice assistant is shown to deliver reproducible results disregarding a person's profile, this definitely contributes towards the credibility of the system. This is however not an easy case, since an advanced voice assistant is not expected to be deterministic in most situations, otherwise it becomes boring! We elaborate later on this thesis how utterances **[]** handle this problem.

On the other hand, information systems range from web pages (e.g. frequently asked questions section (FAQ), forums). These are in most cases even less effective than contacting a human as getting the proper information takes a lot of time, or the level of trustworthiness or participation particularly in a forum is too low, the problem stated is too broad or too specific compared to the answer we are seeking. A user also could come purposefully across false positives in a search and rely on irrelevant information without knowing. Furthermore, some case-related information might be required to have a proper understanding of a situation or a scenario and provide adequate answers. For Example, if a user would like to know if a certain accessory is compatible with their mobile device, they might need to give a model number, which they may or may not know. Consequently, it is of high interest to maintain a system that could determine all these factors autonomously or with the least possible human interference such that a system supersedes the abilities of the classical Q&A approach.

Internationalization is also another factor to take into account. Since languages differ not only in their vocabulary but also grammar, word and sentence structure, developing a voice-first device requires a flexible infrastructure and software stack both able to accommodate these deviations. In that respect, not only is region-specificity important, but also being able to cater for people in a region who do not speak its official language or are residing there temporarily. Especially in businesses where it's difficult to hire skilled foreign-language speaking personnel, a voice assistant can overcome this challenge as it would communicate more accurately and will not have language problems. A customer is then given the option to avoid an inconvenient experience with a call centre representatives where neither of both parties understands the other. For those who have minimal understanding of the language, there are possibly also options to provide help in the native language if the user is given the options he/she can answer a prompt with. At the very least, a VUI could still give feedback to the user of wether it understands the language or not, since algorithms for detecting a language are not as

complex as answering the question once the language is undertood. ⌈ approaches to localization? (Translators via Wordnet, Stammsprache,...refer to IRS lecture notes) ⌋

In the following, we take all these factors into consideration and narrow down a tailored scenario in the e-Government sector. For this we choose to present our solution using Alexa, since among Apple and Google it hast the voice-first devices best equipped for its platform, a user base larger then its competition and provides the most mature API and SDKs.

1.2 The Analyzed Scenario

Berlin.de is an online one-stop-shop for approximately 3,7 million residents of the German capital [2] with ⌈ hundreds/thousands ⌋ of visitors daily for information lookup, appointment bookings and even access to local news. As part of a federal modernization procedure with the help of the ministry of interior, D115 was launched in 2009 [8] as a phone service to help residents find relevant information about a public service or municipality, something that can be tricky if a person has no overview of the local government structure and still not always easy even with the help of search engines nowadays. To promote information accessibility, D115 continuously aims at expanding its reach and services. It is therefore worth exploring, how to offer D115 services in a fashion that takes advantage of conversational abilities beyond its human personnel.

For now, although local authorities rely heavily on their websites to communicate information to the public, the challenge is mainly finding the right service. In a metropolis with a high influx of incomers, it is also very likely that certain services are frequently pursued, meaning that helping find the right public service or authority is a repetitive task. In this context, thinking of a voice assistant as a public service could have several advantages, like offloading some traffic from the phone service, getting over the language barrier in the case of non-german speakers, expatriates or simply helping native customers formulate the right wording for a query in a more intuitive way than using a search box.

1.3 Approach and Goals

✍ **To-do**: **Aufgabenstellung:**
-AL: Anschließend soll das Ziel der Arbeit formuliert werden: Entwicklung und Evaluation eines Prototypen für den Anwendungsfall.

1- es sollen die Stärken und Schwächen eines solchen System zu analysieren.
2- Es sollte zunächst eine Dienstleitung aus dem Berliner Service-Katalog mit dem Chatbot beauskunftetwerden können.
3- Nuancen beachten (e.g. 10243 / FHain)
4- Smalltalk Fähigkeiten

✍ **To-do**: - ..it would speak as an advantage for bots if they can determine these things automatically..
- currently most tasks revolve around performing tasks like setting an alarm,
- answer suggestions functionality in chatbot equivalent - next step is to get around the user's frustration by making the bot at least more human.
- Alexa Skill will work in Germany in english and german -¿ add english after german

1.4 Structure of the Thesis

This thesis endeavours to shed light on the following: In Chapter 2, we discuss related work as currently available. We first show a few use cases for chat bots and voice assistants and their implementation. For that we consider ⌈ the City of Vienna chat bot service "WienBot" for German and Singapore/ LA/... for English ⌋ and compare them to the current application available on service.berlin.de. We then introduce the Alexa platform and the idea behind Skills in chapter 3 and discuss how it can overcome a few struggles with respect to its counterparts ⌈ like redundant boilerplate code, smalltalk, but also limitations ⌋ . We also present the frameworks we use in our own implementation, the prerequisites and the artefacts provided initially. Chapter 3 and 4 represent a detailled analysis of our implementation. In particular, ⌈ SOME MORE SENTENCES. In Chapter 4, our solution is presented. This solution covers ... (SOME MORE SENTENCES). ⌋ Chapter 5 evaluates our implementation with respect to our target in the defined ⌈ 4-5 ⌋ use cases. ⌈ SOME MORE SENTENCES ⌋ In Chapter 6, we conclude with how the implementation of this e-Government solution is attainable through Alexa and where future works can be directed.

Chapter 2

Background

In this chapter we discuss Amazon's state-of-the-art strategy before moving on to related work in other archetypes as it is important to introduce the implementation scope of voice assistants bottom-up prior to exploring the current context within the same boundaries of voice assistants then compare it to other approaches in the larger context of conversational bots as a whole from a technical and user experience point of view. We start juxtaposition of Voice User Interface (VUI) to the Graphical User interface (GUI) with respect to cognition and behavioural design which gets us to define new terminological foundation that will follow throughout this thesis to then bring implementation requirements to the table in section 2.3.

VUI vs. GUI

etkallem 3an el paradigm shift eli 7asal ma3 el mouse from cli if not already mentioned then 3an el smartphones and the wep from wap etc (deskop version, responsive design, two versions, em instead of pt, relative, starting with a tablet and a phone then anything relative to those then when the standards showed that there won't be an only 15 inch and a 12 inch there will be everything in between, we changed the idea to include continuous units in the spectrum and not only discrete units) [**?**] different screens, different mobile experiences and what does it mean

how do you think of designing voice when you have a mobile mindset how do you think about screens when you have a mobile mindset and thinking about voice subtle differnce

2.0.1 VUI Terminology

✍ **TO-DO**: explain json

intent

utterance

slot

gui in vui *card display* alexa podcast el adima that i heard while running i think imagine you have an intent and the label on th button says ok

2.1 State of the Art

Amazon Web Services (AWS) + Alexa

Getting started with Alexa as a platform for the first time might seem a little overwhelming especially since each component of it is being constantly restructured since its debut release in November 2014 and with Q4 2016 being the beginning of its major market penetration success [7]. As throughout the course of this thesis these major changes occurred, we will go over the workarounds and circumventions to retain a working version of the software we present, discuss

AWS

Amazon Web Services is a Cloud Computing service provider with a complete set of services to help build and run web applications "reliably and securely at a cost and scale" according to one's independent needs. [13] It comes with agile abilities to adjust to various solution at a flexible scheme with benefits like multiple server farms globally (operating under different legal contracts with respect to data security and user privacy), caching, NoSQL, DevOps and so forth.

Alexa

Alexa is a cloud-based voice service assistant platform by Amazon powering millions of devices ⌈ with number of requests daily—monthly ⌋ on its own-branded devices like the Echo, Echo Dot, Tap, FireTV as well as cross-platform through mobile apps available through Apple's AppStore for iOS and Google Play Store for Android devices. Unlike Apple's approach with Siri for instance [1], where including support for third-party integration on iOS's Service Development Kit (SDK), Amazon decided with the launch of Alexa to include non-Amazon developers right from the start by introducing a multitude of (SDKs) around the platform as part of AWS, e.g. Alexa Skills Kit SDK discussed further below.

[1] as know from its policy on new software and hardware products like the case with the iPhone, the Mac, etc.

So far, though the separation of AWS and Alexa's environments is not linguistically intuitive with the company's name labelled on every service component [2], Amazon.com, Inc. offers an array of web services through AWS that summarize all building blocks necessary to operate Alexa as a software for end-users. Of course with the introduction of the aforementioned devices, it becomes intuitive to call these 'Alexa devices' since their primary purpose is to operate as Alexa clients. We refer to Alexa here only as the service offered by Amazon to consumers. Consequently, although Alexa comes as a fully packaged service to end-users, we can reduce it to a compilation of many micro-services provided by AWS. As most of these are available separately in the form of Software as a Service (Saas), we conclude that AWS incorporates the following components required to make Alexa come together and becomes a consumer of its own web services platform.

Alexa's AWS Modules ⌈ make logos as minipage objects or set as table ⌋

listed in sequential order of importance, Alexa's service modules include but are not limited to:

Lex [3] *for conversational interfaces using Natural Language Understanding, Text-to-Speech and Speech-to-Text*
"a service for building conversational interfaces into any application using voice and text" [13]. While Lex is the most important backbone to make Alexa possible and can be a main operator of another software package to create a whole new category of voice assistants and conversational bots independent from Alexa, Siri etc., development for Lex was not available in German at the time of this research. We therefore decide to use the Alexa Skills Kit, which take advantage of Lex and the other components described below. Lex and Alexa use the same deep learning techniques for natural language processing with the workflow described with the example in figure 2.1.

Polly [4] *for speech synthesis*
another "service turn[ing] text into lifelike speech, allowing [developers] to create applications that talk" [13] wile harnessing the power of deep learning. It is more or less the mouthpiece of Alexa built on top of speech synthesis algorithms.

[2] see Etymology 6.4

Transcribe [5] *for automatic speech recognition using Speech-to-Text*
which "makes it easy for developers to add speech-to-text capability to [...] applications" [13]. With Transcribe we are able to get text out of the user's voice before passing it into a format Alexa's backend would understand.

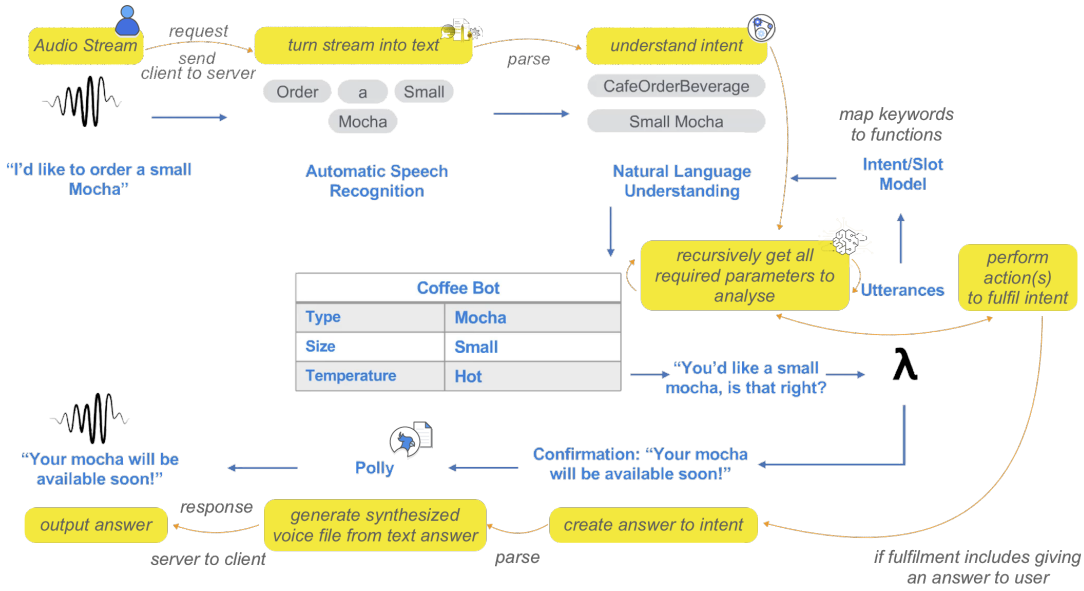
Lambda [6] *for intent fulfilment*
Although Lambda is a versatile "service [built] for a variety of real-time serverless data processing systems" [13], we use it as an integrated server instance to host our back-end code for intent fulfilment 2.2. We might as well use just any other privately or cloud-based server solution to host our code and use it as an endpoint for our software, however choosing Lambda takes away the overhead of linking the front-end with the back-end of the program we develop (an Alexa Skill as we describe below).

CloudWatch [7] *for event logging*
CloudWatch acts like the console for an operating system. It monitors all low-level events happening within the AWS sphere. In combination with Lambda it comes as handy tool to log events resulting at runtime once a Lambda instance is called and its code being executed.

IAM [8] *for identity access management within AWS*
Identity Access Management (IAM) is a secondary service module regulating in the Alexa context in combination with Lambda the routing rights between internal Amazon endpoints. It also ensures compliance policies are enforced within and between AWS modules, as well as between AWS modules and other external services.

Cognito [9] *for identity access management beyond AWS*
like the rest of AWS's platform, Cognito is a scalable service module to perform user authentication and can be used in combination with an external endpoint instead of Lambda.

Figure 2.1: Interaction between AWS modules in the use case of a “Coffee Bot” based on reference [4]



putting different combinations of these and other building blocks interactively together generates the model for Alexa. In a world increasingly operated by Internet of Things (IoT), we describe a possible interaction in the example below for a use case of a hypothetical chatbot that operates a coffee machine using some of these service modules (Figure 2.1). Although this graphic describes how an end user would combine these modules to set up their own chatbot, this is the same workflow that Alexa uses. Hence, with Amazon’s use of its own micro-services we infer that takes advantage of these to build a whole new ecosystem putting Alexa Skill developers as producers, end-users as consumers, and the Amazon website and Alexa App as a marketplace to mediate between the products (Skills) the developers produce to their target customers (end-users, country-specific or worldwide). From a marketing perspective Amazon achieves through Alexa a vertical diversification of its product programme (where existing AWS services result in a new product expanding the value chain) while simultaneously offering an extention of its aggregator-model marketplace model by playing as a mediator between the developers’ role and that of end-users

we can therefore describe the meta-model for Alexa similar as one of an application with several front-end and back-end components. Unlike in most GUI-based scenarios with an MVC design pattern [18], where the user uses the `controller` to manipulate the `model`, which in turn updates the `view` appearing to the user, in a VUI scenario, we need to consider that the user’s paradigm to the `view` component is quite different. Before we

dive into the VUI paradigm, we introduce Alexa's own implementation of it for third-party applications. These are broken down into the following elements for users and developers to comprise a holistic ecosystem:

- Alexa Skills Kit: although it is hard to define it as a complete SDK for Alexa, it is responsible for compiling the loosely coupled tools provided by AWS and others to act as an interface for the skill from a developer point of view. This fits into the rest of Amazon's scheme of focusing on interoperable micro-services that fit multiple purposes.
  - Alexa Developer Console
  - ASK Command-Line Interface
  - Amazon Voice Interface
- Alexa Skills Store for users: where a user can preview the skills before they install it, know if they want it or not and by installing them, the own instance of alexa becomes "smarter" um einen weiteren skill [10] for now the user does not need to think about updates since these happen in the backend. approval makes amazon decide wether versions are good and make the skill better or worse or change things to worse partly

Alexa Skills Kit

2.1.1 Amazon Voice Service

just like other common chat bot constructs, Alexa Skills Kit (ASK) divides the building model into intents **??**, utterances and slots. The Fulfilment part is taken care of through Amazon Lambda contains the programming ⌈ and business ⌋ logic to the interface

- Intents

Alexa Interfaces

hardware ⌈ just make a comparison table, which one has a screen, which one has which capabilities with alexa ⌋ Echo, Echo Dot, Tap, FireTV

software Alexa app is not an alexa interface

while these are possibilities for testing, too, they are primarily developed with testing in mind

[10] das mündet into something where AI becomes smarter then maybe in 5 years the concept of alexa skills won't exist anymore and would be integrated into Alexa's own brain without the user knowing and it would just be a hit or miss kind of thing.

Table 2.1: Alexa Devices in Comparision

	Echo Dot/Echo/Echo Plus/Tap	Echo Spot	Echo Show	FireTV
screen	n	2.5” round	7.0”	via HDMI Display
line out	y	y	via Bluetooth	via HDMI

EchoSim.io

Alexa Simulator

Reverb

✍ **TO-DO**: move / remove what is irrelevant In this section, we conduct a short survey of problems we can face with natural language

topology of bots

by platform: -API.ai / Facebook Messenger Chatbots / -wit.ai / -motion.ai / Flask **by category**
- leisure / - fun bots / - productivity / - more (graph from voicelabs report)
- what are classic use cases for their use with prominent examples?
Booking tickets (e.g. airline bot)
- quick survey of respective ’AppStores’
by purpose
-physical locations (home, office, car, phone, in a business)
Information bots
- mention available service types (information system as a ”webpage/database”)
- vs an interactive bot that gives you customized information on demand hier soll der D115 Anwendungsfall ”Beauskunftung” kurz erläutert werden
social bots
- with advantages / disadvantages
- fake news / online reviews
more on AI in bots (optional)
- use of ML
Handyversicherungsbeispiel
- from business perspective, the bot is aiming to sell more polices,
- the bot tries to determine if there is a nuance in the user’s answer (machine acting as a judge!) - e.g. “how did the phone fall off“ - MKTG - Aufwand
eingehen auf
- Wienbot
- Singapore / LA / Ask Georgia...

2.2 Related Work

wienbot, singapore etc, d115

✍ **TO-DO:**

While we have no access to AskGeorgia

✍ **TO-DO:** 2 ¶

- Chatbot vs. human: Analyze differences between bot and human response
-disadvantage: a bot wants a sentence broken down in small pieces to avoid errors in lengthy interpretation
- **Why can't robots understand us:** language ambiguities - the need to understand context
–Syntactical: Homonyme
–Semantic: Methaphors, sarcasm, and puns
–dialects: enunciations
–underlying grammar
–underlying sentiment
-**NLP Progress:** How does it help in enriching the bot experience
–neural networks: help understanding language patterns and get better over time
–thought vectors: helps connect different words with related meaingns
- **wrap-up:** can bots replace serivces offered by humans? – mention transition from facets (Altavista) to metasearches to all-in-one (Google).
– chatbots as enablers in customer service industry
– conclusion: Although not impossible, it is a bit too far-fetched at this stage.

These include lex for.... so-called 'Skills' It comes with
Intent fulfilment

Figure 2.2: text

✍ **TO-DO:** - Alexa Skills Kit+ Amazon Voice Services
- mention how ability to react to everything is centralized at alexa somewhere
talk about SKILLS
- ability to retain sessions (explain requests/responses - GET/POST)
- fullfilling intents
- nested handlers

skill service: code - business logic - handles json requests skil interface: configuration (developer portal)
- difference to Lex & Polly
diff alexa lex
- Major prob: lex is not in german - Alexa Documentation

2.3 Frameworks and Data Structures

-AL: Ich würde erst etwas die Algorithmen und Datenstrukturen (Textanalyse, JSON, ggf. Graphen beschreiben. -AL: Anschließend die Frameworks vorstellen
-AL: Wichtig ist: Aus den Beschreibungen eine Schlussfolgerung ableiten, welche Art von Lösung entwickelt werden soll.
for current bot:
- Lucene **as the golden standard**: spell check, unscharfe suche, Tika / detect language / ...
- Solr - explain what's an intent, whats a slot `https://service.berlin.de/virtueller-assistent/virtueller-assistent-606279.php` `https://www.itdz-berlin.de/`

Node.js

AWS Lambda supports multiple runtime environments including Python, Java, C# and Go. We decide to use Node.js due to its event-driven nature and to take advantage of its non-blocking I/O model. Being single-threaded, Node.js guarantees high performance at large scale with large volumes of requests considered. With its JavaScript (ECMAScript) foundation, no wonder it is becoming a standard in web-apps. Hence, the decision also comes due to the richness of develpers' experience with the implementation for Alexa Skills.

✍ **To-do**: - Server-side, browser side (Chrome V8), App layer, data layer
- because it can read our JSON easily and fast - talk about Methodenaufbau (syntax) and firing events

Solr

w lucene w tika w nutch wel habal da if required

2.4 The Analysed Scenario as a Use Case

D115

- summarize infobroschuere_ BMI08324_screen_barrierefrei.pdf
-Use case im Detail
-Welche Daten gibt es?
-Was sind die Erwartungen?
- wie kann man die Güte des Systems beurteilen?

- Meist sollte man in diesem Kapitel die Lösung schon im Auge haben, um die Erwartungen so zu formulieren, dass die Lösung auch geeignet ist?

currently deployed bot

- dienstleistungen.json structure (finding the info through hierarchical nodes)
- interpreting the nodes as intents - traversing the nodes (one level up then to next node)
- no session/no persistence

OnLaunch IntentHandler intent is triggered by utterence account verlinkungen etc
provided in JSON for value lookup, there are
explain how the json nodes map to intents and cores in solr etc

Figure 2.3: `Dienstleistungen.json` - Primary Nodes

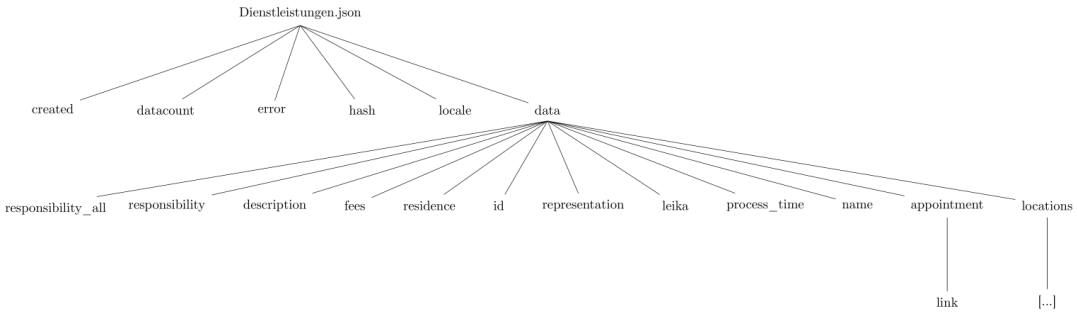
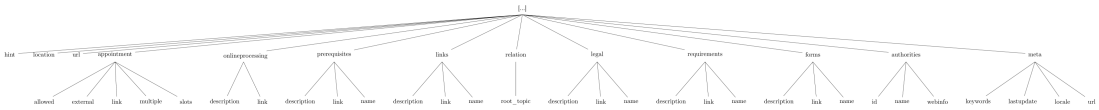


Figure 2.4: `Dienstleistungen.json` - secondary Nodes



- 616 Intents as `data`, each containing

> ✍ **TO-DO**: missing variables e.g. are required papers, flag: persönliche Vorsprache ja nein, ...

  - `<string> responsibility` denoting in which city halls a service is available
  - `<boolean> responsibility_all` a flag set to true in case the service is available in all local authority offices / service points

- `<HTML list string> description` not unified and includes text
- `<string> not unified and might need to have an \lstinline`int— added to it and set to 0 in case service is free
- `<int>residence`
- `<int>id`
- `representation`
- `<long>leika`
- `<string> process_time` need to derive minimum, average and maximum service times instead of a string, as well as conditions
- `<string> name` the name of the service that would make sense to a human
- `<node> appointment` with
  * `link` (Key value with URL to /terminveinbarung page) - check if orphan or if it is for each behörde and in that case how it gets the right one
- `<node> locations`
  * `hint`
  * `<int> location` one of the 12 authorities
  * `url` of that service at that authority
  * `<node> appointment` (a second one)

    .
- `<node> onlineprocessing`
- `<node> prerequisites`
- `<node> links`
- `<node> relation`
- `<node> legal`
- `<node> requirements`
- `<node> forms`
- `<node> authorities`
- `<node> meta`

2.5 Implementation Possibilities

- structure of Hitlist on berlin.de is provided by ITDZ - as opposed to Versicherungsfirma z.B (ML tries to detect irregular patterns in case customer is lying). -

unfortunately forums vs. FAQs did not work. if i want assistance, i want the customer to tell me the model number - and forums have mostly Schrott!
what the bot curently achieved is at least not give wrong answers, sometimes says idk but it doesnt confuse u. same attitude like in german shops (nur unpassende antworten sind frustrierend!

-Vorgehensweise: XML -¿ index über Lucene - ¿solr knoten...based on sth like when i say äm 10. augustït gets me masalan events..aha august ist ein monat, monat relates to calendar, calendar relates to events

Chapter 3

Skill Implementation: ⌈ Dienstabfrage ⌋

- as an example for voice
-System Specifications
-System Structure
-UML Diagrams
-Design Choices
-scopes and granularity

3.1 All about Alexa

https://en.wikipedia.org/wiki/Amazon_Alexa https://medium.com/@robinjewsbury/how-to-create-bots-and-skills-for-facebook-messenger-and-amazon-echo-4
- Alexa Appstore had over 5,000 functions (”skills”) available for users to download,[18] up from 1,000 functions in June 2016. McLaughlin, Kevin (16 November 2016). ”Bezos Ordered Alexa App Push”Paid subscription required. The Information. Retrieved 20 November 2016.

Perez, Sarah (3 June 2016). ”Amazon Alexa now has over 1,000 Functions, up from 135 in January”. TechCrunch. Retrieved 5 August 2016.

3.2 Difference Between Lex and Alexa Skills

https://stackoverflow.com/questions/42982159/differences-between-using-lex-and-alexa#URL
https://aws.amazon.com/lex/faqs/

https://aws.amazon.com/about-aws/whats-new/2017/09/export-your-amazon-lex-chatbot-to-the-alexa-skills-kit/
Amazon Lex is a service for building conversational interfaces using voice and text. Powered by the same conversational engine as Alexa, Amazon Lex provides high quality speech recognition and language understanding capabilities, enabling addition of sophisticated, natural language ćhatbotsźo new and existing applications. Amazon Lex reduces multi-platform development effort, allowing you to easily publish your speech or text chatbots to mobile devices and multiple chat services, like Facebook Messenger, Slack, Kik, or Twilio SMS. Native interoperability with AWS Lambda, AWS MobileHub and Amazon CloudWatch and easy integration with many other services on the AWS platform including Amazon Cognito, and Amazon DynamoDB makes bot development effortless.

3.3 APIs and SDKs

- swagger for handling JSON requests?
- https://github.com/alexa/alexa-skills-kit-sdk-for-nodejs

3.4 challenges

- und Lösungen dafür
- eine Überführung in Alexa, not writing everything new in alexa. such that when you want to do it in another system what do u want to integrate?
- use external web service maybe? in case that helps instead of alexa doing everything..
- konten hosting to be on alexa
- wo hilft mir alexa, was mach ich lieber woanders?
- Ähnlichkeitsmaße -levenstein-distanz, IFTTT

Error: There was a problem with your request: ”werden?” in the sample utterance ”TestIntent was soll aus dieser Skill werden?” is invalid. Sample utterances can consist of only unicode characters, spaces, periods for abbreviations, underscores, possessive apostrophes, and hyphens.

do not use ”?”

Chapter 4

Skill Implementation: ⌈ Kosten-/Terminabfrage ⌋

- as an example for text
- implementing the answer suggestions as buttons
- passing data to the Bürgeramt terminseite
https://console.dialogflow.com/api-client/
https://console.actions.google.com

Hey:

do this set up as prerequisites first as a deployment script

- install aws cli - install ask cli - set up a profile on AWS - ask init etc.

https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-getting-started.html

There is no AWS credential setup yet, do you want to continue the initialization?

https://developer.amazon.com/docs/smapi/quick-start-alexa-skills-kit-command-line-interface.html

Manage IAM roles (User, groups, rights, policies etc.)

all logs go onto cloudwatch including when the request is not right etc...can be helpful to know what users want, beyond testing

amazon does not disclose avaliable values for their slot types (lists), but they give you examples here

===

finally, we had to resolve to the information publicly available to tailor custom scenarios

=== once you upload the new lambda, the old one is gone, unless you version it like this: https://docs.aws.amazon.com/lambda/latest/dg/versioning-aliases.html

====

first i got screwed over with this Big nerd ranch then this happened: alexa-skill module - so don't use this tutorial, although very helpful for a beginner and makes you

understand what happens under the hood, it has been abstracted in many other function and the logic is no longer the same. - which explains why no one starred it

https://github.com/matt-kruse/alexa-app

https://www.bignerdranch.com/blog/developing-alexa-skills-locally-with-nodejs-deploying-your-skill-to-staging/

Chapter 5

Evaluation

✍ **TO-DO**: NPR
time series a-b testing weighting use of correlations in the legal system (got, defaulting) search map predicts ur personality fear of judging, fear,. . . interdependence with robots likelihood of having different social cirlces for a lasting relationship
- see how many use the skill after publishing

-benchmarks
-strengths and weaknesses
-challenges
-performance
-usability
-feasibility of using the studied agents
- node.js?
- amazon's system testing options (incl. Betas)

- system usability scales (ISO, DIN)
- Con: Alexa skills are listed in the amazon shop page. Sehr unübersichtlich
just like prime
- impression: Amazon collects data and makes something "intuitive out of it for you". e.g. fire stick setup already had account linked before connecting to the internet! scary/funny/ but then it could be counterintuitive at some point if u want to do ur own customizations.
- removing bias in recriutment of participants (diversify based on what categories?)

- EVAL: AUC/ROC, true positives, false...no of utterances to text
- compare with Wiener Stadportal as a benchmark for a bot
https://www.wien.gv.at/bot/ http://www.vienna.at/wienbot-chatbot-der-stadt-

wien-informiert-als-virtueller-beamter/5590853 https://digitalcity.wien/wienbot-auszeichnung-fuer-chatbot-der-stadt-wien/ singaporebot

5.1 Results

usability metrics: - heuristic eval - guidelines **(jakob nielsen, ralf molich whitepaper)**
– biggest usability flaw
- cognitive walkthrough
– step-by-step approach
– questions..wil the user tr and achive
- pluralistic walkthrough
– panel method
- hallway testing
- A/B Test
- speed and Bottlnecks

- clientele: census / SOEP, who can use the bot
- make a small prediction (Bus Analytics)
- this Hassloch thing from MKTG

5.2 Discussions

- Evaluate the system:
- is it trivial to build such a bot or not / what is the aufwand
- how does it react with longer sentences? some service names are long
- what does levenstein distanz cause
- wie leicht kann ich eine antwort finden auf das was ich suche?
- how am i going to classify my tests?

- are chatbots being pushed on the market or is there a demand? (kleine Umfrage basteln?)
- how easy or difficult it is to make a bot: planing poker - varianz anschauen zw. leicht und schwer und iterativ darüber sprechen
- wo kann der Kunde (Sawa2 kan el end user or the senat in our case) help optimize the bot masalan bürgeramt beyektebo, welche Rechtsgrundlage k̈eine- auffällige Probleme masalan zay Perso, PA, personalausweis, how to introduce ëxpert modeso that if u add it with a special character it knows what u want, just like alexa knows when u rename the lamp - refer again to use cases and exper vs personal field

Chapter 6

Conclusion and Future Work

6.1 Summary

6.2 Conclusion

6.3 Future Work

- use machine learning to rank higher demands for more popular services.
- matkhoshesh fel 7etta di awi - for now hitlist already given.
- future of bots. deren Einsatz. roles (As judges, catereres in hotels (that hotel botler)

We are going to build neurons so they can bridge together (Alexa Podcast 18) Otherwise they will forget Branches stop growing when they reach another branch Intelligenceaggreation - amplification We are voice-first approach (as said earlier: Inner voice) We do not know much about human brain /intelligence- -the point is not in taking over but making Internet more powerful (Podcast) information will appear as you need it: Voice first not voice only - less time to look at screen.

Bibliography

[1] fka VoiceLabs Alpine.ai, jan 2017. accessed on 16.02.2018.

[2] Amt für Statistik Berlin-Brandenburg. Statistischer Bericht - A I 5 hj 1. https://www.statistik-berlin-brandenburg.de/publikationen/stat_berichte/2017/SB_A01-05-00_2017h01_BE.pdf, August 2017. Accessed on 06.02.2018.

[3] Brian Roemmele, Dave Isbitski. Episode 018 - A Voice First Future with Brian Roemmele. *Alexa Dev Chat Podcast*, 11 2017. Accessed on 13/02/2018.

[4] Amazon Web Services Webinar Channel. Introducing amazon lex: Service for building voice/text chatbots - march 2017 aws online tech talks.

[5] Wikipedia contributors. History of natural language processing — wikipedia, the free encyclopedia, 2017. [Accessed 14 February 2018].

[6] Wikipedia contributors. Usage share of operating systems — wikipedia, the free encyclopedia, 2018. [Accessed 14 February 2018].

[7] Gartner. Gartner reveals top predictions for it organizations and users in 2017 and beyond. Oct 2017. accessed on 15.02.2018.

[8] Geschäfts- und Koordinierungsstelle 115 im Bundesministerium des Innern. {https://www.115.de/SharedDocs/Publikationen/Service_Publikationen/Infomaterialien/infobroschuere_%20BMI08324_screen_barrierefrei.pdf?__blob=publicationFile&v=8}, November 2013.

[9] Tobias Grossmann, Regine Oberecker, Stefan Koch, and Angela D Friederici. The developmental origins of voice processing in the human brain. *Elsevier*, 65:852–8, 03 2010.

[10] Judy Goldsmith and Nicholas Mattei. Science fiction as an introduction to ai research, 2011.

[11] M. Ballve. Messaging apps are now bigger than social networks, 2016.

[12] NBC. Amazon's alexa had a breakout holiday — people even used echoes to buy more echoes. Accessed on 15/02/2018.

[13] Amazon Web Services.

[14] Shankar Vedantam. Radio Replay: I, Robot. `https://www.npr.org/podcasts/510308/hidden-brain`, 2018. Accessed on 25.01.2018.

[15] Statista. Smartphone user penetration as percentage of total global population from 2014 to 2020, November 2016. Accessed on 06.02.2018.

[16] Laurie Sullivan. Gartner predicts 30 12 2017. accessed on 12.2.2018.

[17] Joseph Weizenbaum. *Computer Power and Human Reason: From Judgment to Calculation*. W. H. Freeman & Co., New York, 1976.

[18] Wikipedia. Model-view-controller.

[19] Katie Young, June 2016. accessed on 16.02.2018.

Appendices

6.4 Excursus: On Etymology

Product naming is a branding problem which companies deal with differently. In the case of Apple, with exception of the 'Apple Watch' and on their consumer products line, they opt for a clever version using the letter 'i' in front of simple common words, like iBook, iPad, iPhone. though this naming convention originally arose from 'i' to stand for 'interactive' celebrating the early adoption of the internet on the Macintosh product line, it has turned into a naming breakthrough since it had two effects: 1) it gives the user of Apple software or hardware product a feeling of personalisation for the letter 'i' would make something sound as belonging to oneself, namely " 'I', as a person" as a prefix for the name of the object create the name of the product, and 2) it is a marketing strategy to associate all products and services carrying the prefix 'i' with Apple without the need to name the company's name. This applies to products and services that came to life even long after people's adoption of the internet and the widespread use of Apple products, like with iCloud as a web service. Another example is with professional software products like AutoCAD combining the company's name (AutoDesk) with the product's functionality (Computer Aided Design). Companies like Adobe clearly prefer to name their products and services with their brand's full name included, as the case with Adobe Photoshop, Adobe Flash (formely know as MacroMedia Flash). Similarly, as the youngest of them, Amazon adopts a combined approach by stating the full name of the company on its consumer line like with 'Amazon Prime', 'Amazon Alexa' and a mix between 'AWS' and 'Amazon' like with 'Amazon SageMaker', 'AWS Firewall Manager' etc. on its developer B2B line. amazon transcri

Appendix A: Abbreviations

AWS	Amazon Web Serivces
ASK	Alexa Skills Kit
AVS	Alexa Voice Service
ARN	Amazon Resource Name
MVP	Minimum Viable Product
AI	Artificial Intelligence
NLP	Natural Language Processing
ML	Machine Learning
GUI	Graphical User Interface
VUI	Voice User Interface
ACID	Atomicity, Identity, .. criteria
appx	approximately

Appendix B: Glossary

Intent	erklärung
Slot	erklärung
Utterance	erklärung
Alexa	erklärung
Alexa Skill	erklärung
Lambda Function	erklärung
Alexa Skills Kit	erklärung
Amazon Developer Console	erklärung
AWS Lambda	erklärung
Amazon Lex	erklärung
Amazon Polly	erklärung
Amazon Transcribe	erklärung
ElasticSearch	erklärung
node.js	Framework built on top of JavaScript
Interaction Model	erklärung
Service	bot, AWS, Berlin.de
https://docs.aws.amazon.com/general/latest/gr/glos-chap.html	
Application ID	erklärung
Skill ID	erklärung
Bot	Unless otherwise mentioned, yeb2a Cha
Hitlist	erklärung
Voice-first	an always-on, intelligent piece of hardwa
B2B	business-to-business
B2C	business-to-consumer
Inc.	incorporation